Ph.D. DISSERTATION

Enriching Seed Information for Robust Interactive Image Segmentation

강인한 대화형 영상 분할 알고리즘을 위한 시드 정보 확장 기법에 대한 연구

BY

GWANGMO SONG

February 2021

DEPARTMENT OF ELECTRICAL ENGINEERING AND
COMPUTER SCIENCE
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Enriching Seed Information for Robust Interactive Image Segmentation

강인한 대화형 영상 분할 알고리즘을 위한 시드 정보 확장 기법에 대한 연구

지도교수 이경무

이 논문을 공학박사 학위논문으로 제출함

2021 년 2 월

서울대학교 대학원

전기컴퓨터공학부

송광모

송광모의 공학박사 학위논문을 인준함

2021 년 2 월

위원장 조남익 (인) NamIk Cho

부위원장 이경무 (인)

위원 윤일동 (인)

위원 한보형 (인) Bohyung

위원 이수찬 (인) 이수찬

Abstract

Segmentation of an area corresponding to a desired object in an image is essential to computer vision problems. This is because most algorithms are performed in semantic units when interpreting or analyzing images. However, segmenting the desired object from a given image is an ambiguous issue. The target object varies depending on user and purpose. To solve this problem, an interactive segmentation technique has been proposed. In this approach, segmentation was performed in the desired direction according to interaction with the user. In this case, seed information provided by the user plays an important role. If the seed provided by a user contain abundant information, the accuracy of segmentation increases. However, providing rich seed information places much burden on the users. Therefore, the main goal of the present study was to obtain satisfactory segmentation results using simple seed information.

We primarily focused on converting the provided sparse seed information to a rich state so that accurate segmentation results can be derived. To this end, a minimum user input was taken and enriched it through various seed enrichment techniques. A total of three interactive segmentation techniques was proposed based on: (1) *Seed Expansion*, (2) *Seed Generation*, (3) *Seed Attention*. Our seed enriching type comprised expansion of area around a seed, generation of new seed in a new position, and attention to semantic information.

First, in *seed expansion*, we expanded the scope of the seed. We integrated reliable pixels around the initial seed into the seed set through an expansion step composed of two stages. Through the extended seed covering a wider area than the initial seed, the seed's scarcity and imbalance problems was resolved. Next, in *seed generation*, we created a seed at a new point, but not around the seed. We trained the system by imitating the user behavior through providing a new seed point in the erroneous region. By learning the user's intention, our model could efficiently create a new seed point. The generated seed helped segmentation and could be used as additional information for weakly supervised learning. Finally, through *seed attention*, we put semantic information in the seed. Unlike the previous models, we integrated both the segmentation process and seed enrichment process. We reinforced the seed information by adding semantic information to the seed instead of spatial expansion. The seed information was enriched through mutual attention with feature maps generated during the segmentation process.

The proposed models show superiority compared to the existing techniques through various experiments. To note, even with sparse seed information, our proposed seed enrichment technique gave by far more accurate segmentation results than the other existing methods.

Key words: Interactive image segmentation, Seed expansion, Reinforcement learning, Attention module, Deep neural network

Student number: 2012-20795

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Most of the images and videos we encounter in our daily life contain various objects and backgrounds. To extract the information contained in the images, it is necessary to separate the objects and the backgrounds. Such separation also helps to solve computer vision application problems, such as deblurring or tracking. In this context, segmentation of an object of interest in an image is one of the fundamental computer vision problems. However, without knowing the user's intention, if object selection is done automatically, certain issues arise in most cases. This is mainly because the objects to be extracted vary from user to user. For example, some people may want to cut out an entire object, while others may want to focus on only a part of the object. To solve this problem, an interactive segmentation approach, which gathers information on a desired object from a user in the form of a scribble or a bounding box and performs segmentation, is widely used to extract the object from an image or video.

The general operation of interactive segmentation has been shown in Figure 1.1. In this procedure, a user first provides an initial seed corresponding to the desired

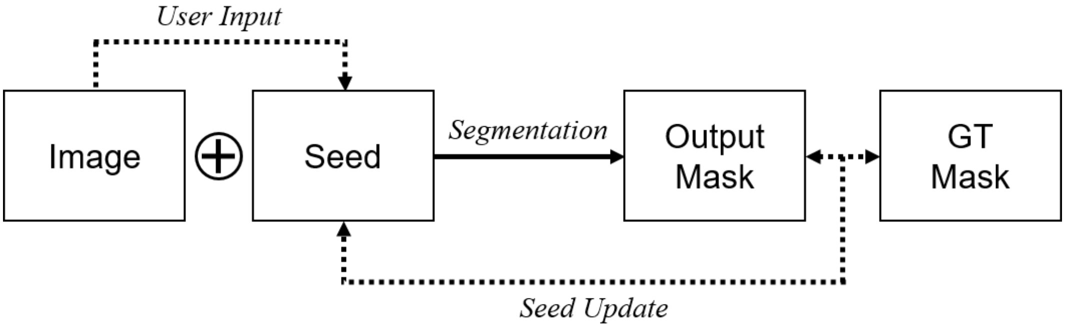


Figure 1.1: Flow chart of interactive segmentation. The accuracy of segmentation is improved via repeated interaction with the user through seed.

object in a given image. Typically, most seeds convey location information of both the foreground and background. The generated seed serves to express the intention of the user. Next, segmentation is performed based on the image and seed information. In this step, if the segmentation result is not satisfactory, the user provides an additional seed. A new seed is then proposed by comparing the obtained mask with the ground truth mask, i.e. the mask that the user expects to obtain. The additional seeds thus focus on correcting the erroneous regions. The user obtains a new segmentation result with the updated seed information. The entire process is repeated until the user is satisfied. Since this procedure involves gradual improvement of segmentation results through interaction with the user, it is called interactive segmentation. Applying multi-label seeds to multiple objects is also possible, but we have only considered binary segmentation in this dissertation.

1.1 Previous Works

As one of the major problems in computer vision, interactive segmentation has been studied for a long time. Many interactive segmentation algorithms have tried to segment the desired object with various user inputs such as contour, scribble, or bounding box. Various algorithms are present in the MRF optimization method. The most famous method is the GrabCut algorithm [3], which repeats the graph cut optimization for segmentation task. Also, the geodesic star convexity (GSC) algorithm [4], which uses shape prior, shows promising results.

Other successful studies were conducted using the Random Walk (RW) algorithm. The RW model was successfully applied to the interactive segmentation problem in [2]. Furthermore, the RWR model [1] which applies the restart probability, successfully solved the segmentation problem. In addition, various algorithms such as watershed model [5], geodesic matting [6], and shape prior model [7] have been suggested. The Laplace coordinate (LC) algorithm [8] using Laplace coordinate-based energy function showed excellent performance.

Meanwhile, various approaches to the interactive segmentation problem of the learning method have attracted attention. Several algorithms have been proposed by interpreting the interactive segmentation problem as a weakly supervised learning problem. In [9], sweeping line multiple instance learning (MIL) technique was presented. The MIL-based classifier is trained with foreground and background bags from the user-annotated bounding box. Santner *et al.* [10] also treated the interactive segmentation problem in a weakly supervised learning manner. [10] showed that HoG descriptors learned with random forests successfully segment out a textured object. Kuang *et al.* [11] trained optimal parameters for a single image. The weights for color, texture, and smoothing terms are tuned during the iteration process.

With the recent development of deep learning, the convolutional neural network (CNN) has been successfully used for interactive segmentation. The first work that applied the deep learning frame to the interactive segmentation problem is

DIOS [12]. The FCN [13] network structure that solved the semantic segmentation problem was modified and used. They constructed an input map by adding a seed map as an additional channel to the input image. Changing the input part of the semantic segmentation network has been used in various interactive segmentation algorithms. RIS-Net [14] improves segmentation accuracy by separating global and local branches. LD [15] increased diversity by focusing on the ambiguity of segmentation and suggesting multiple possible masks from a single input. In the FCTSFN [16], they did not combine RGB images and seed information at the input, but proceeded to each network stream and then fused. CMG [17] has improved performance by additionally using a guidance map containing context information as input. MultiSeg [18] improved their original DIOS method and designed a network that produces various segmentation results according to scale.

1.2 Proposed Methods

The factor that differentiates between general segmentation and interactive segmentation is the presence of seed. The performance of the algorithm dramatically varies depending on the type of seed information provided. According to [19], the minimum seed information required for each image is different. For relatively simple images, satisfactory segmentation can be obtained with only a little information. However, for images where several objects are complexly mixed, more detailed seed information is required. As shown in Figure 1.2, seeds can be divided into 4 types, namely point, scribble, bounding box, and tri-map. The amount of information in the seed and the burden of the user are in a trade-off relationship. For example, for tri-map, the only unlabeled area is the light gray region. Therefore, it provides reasonably

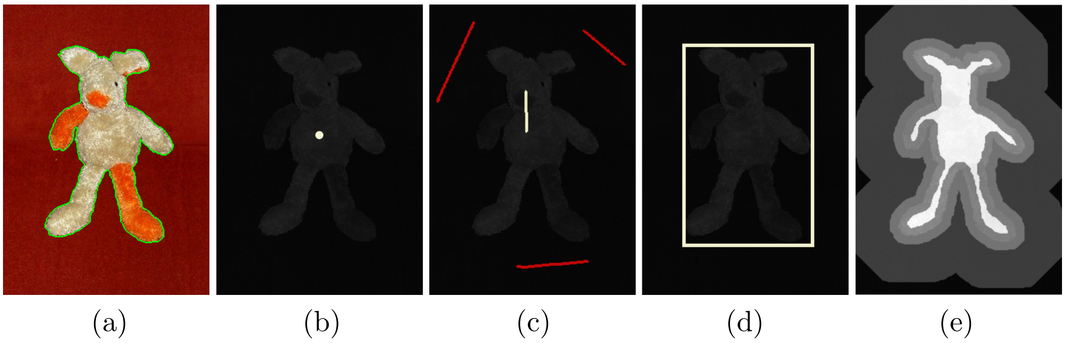


Figure 1.2: Types of user-provided seeds. The object area is weakly marked for visibility. (a) The input image and object boundary (green contour) (b) *Point*. the white dot denotes foreground. (c) *Scribble*. white line denotes foreground, and red lines denote background. (d) *Bounding box*. It contains objects inside the box. (e) *Tri-map*. The white region denotes foreground, and light gray denotes an unknown region, dark gray and black regions denote background.

accurate information about object mask, but it takes much effort to create a whole map. In recent situations where a lot of data are required, interactive segmentation has been under the spotlight as an annotation tool. However, if the burden of the user increases, its utility as an annotation tool is significantly reduced. In that case, it is necessary to perform segmentation robustly with minimal information.

In this dissertation, a method of using a seed with a small amount of information has been dealt with. In particular, three seed enrichment techniques had been proposed that could transform a seed with sparse information into a sufficient seed for segmentation. Using these techniques, segmentation accuracy could be improved by performing the seed transform procedure. The three types of proposed seed enrichment algorithms were based on: (1) *Seed Expansion*, (2) *Seed Generation*, and (3) *Seed Attention*.

The structure of this dissertation and key ideas are summarized in Table 1.1. All of the proposed techniques aim to produce better segmentation results by transforming seed information. However, the direction of seed enrichment is slightly different as well as the methodology used. *Seed expansion* enriches seed information in a spatial

Table 1.1: Summary of the seed enrichment methods proposed in the present dissertation.

Proposed Methods	**Enriching Type**	**Main Idea**
Interactive segmentation with *seed expansion* (Chapter 2)	Expansion of seed to a wider area	Expansion of training set using label propagation
Interactive segmentation with *seed generation* (Chapter 3)	Provide seed in a new position	Simulate correction step by applying reinforcement learning
Interactive segmentation with *seed attention* (Chapter 4)	Reflect semantic information	Update and emphasis of seed using a bi-directional attention module

sense. A new seed was proposed by expanding the area around the given seed. *Seed generation* reinforces seed information by creating a seed point in a new area, not near a given seed. Finally, *seed attention* extends the seed to the semantic domain. By reflecting the semantic information of the image, the existing seed was converted to a seed type suitable for segmentation. The algorithm proposed in each chapter improved the shortcomings of the algorithm introduced in the previous chapter. The algorithm in Chapter 2, limited to continuous expansion, was improved by discrete expansion as shown in Chapter 3. Furthermore, the algorithms in Chapter 2 and 3, which only considered spatial expansion, were extended to the semantic domain in Chapter 4. A brief introduction and contribution of the techniques proposed in each chapter are summarized below.

In Chapter 2, the *seed expansion* technique [20] will be introduced. The seed that a user can easily provide is mainly point or line type, which has very sparse information. In general, the existing interactive segmentation techniques work well for rich seeds but show weakness when the seed information is insufficient or unbalanced. Therefore, we widened the seed region by incorporating a reliable area around the given sparse seed. We proposed a novel seed expansion process in two stages. For each stage, we gradually widened the seed area to obtain an extended seed. The extended seed provided sufficient information for segmentation and generated robust results for initial seed distribution by resolving unbalanced seed information. We additionally used global information to improve the accuracy of the segmentation algorithm which operated based on local information.

The spatial expansion of seeds described in chapter 2 has a limitation that expansion occurs only around the seed. In Chapter 3, a new *seed generation* algorithm [21] will be developed which can generate seeds in new regions. After understanding the intention of the user using Convolutional Neural Network (CNN), a new seed point was created based on the network. We proposed a technique for sequentially generating seed points using a reinforcement learning algorithm. As if playing a game, we can obtain improved segmentation results through the process of picking the correct seed points one by one. Moreover, in this approach, the seed information obtained is much richer than that obtained from expanding the region around a seed.

Lastly, in Chapter 4, seed transformation through *seed attention* [22] will be dealt with. The techniques covered in chapters 2 and 3 require separate segmentation modules. By contrast, in chapter 4, we will introduce a technique that simultaneously performs interactive segmentation and seed transformation. In this approach, the seed information evolves into richer information based on the semantic information of the image. The expanded seeds discussed in the previous chapters involved spatial expansion only and were used only as input to the segmentation module. However, as will be shown in chapter 4, the seeds interact with the features of the segmentation network and help develop each other's information. To this end, a network was constructed using a novel bi-directional attention module. The newly developed network shows better segmentation results than the baseline through a single step.

The final conclusions and summary are presented in Chapter 5. Additionally, the direction of improvement and future work will also be highlighted in this chapter.

Chapter 2

Interactive Segmentation with Seed Expansion

2.1 Introduction

Over the past decades, image segmentation, which decomposes a scene into meaningful objects, has been one of the most significant issues in computer vision fields. Other than the application itself, image segmentation has a practical value as a fundamental technique in other vision problems, such as object detection or scene understanding. However, local appearance ambiguities intrinsically limit the performance in unsupervised image segmentation, which segments a region without any prior. In order to resolve this ambiguity, semi-supervised image segmentation techniques that use additional information have been introduced. A typical technique is an interactive segmentation, in which ambiguity can be reduced by interactively receiving information from the user regarding the label of the region.

In several previous works, the interactive image segmentation problem is formulated as an MRF (Markov Random Field) framework [23, 24, 3, 2, 1]. These algorithms show promising performance for the dataset that contains rich seed information. For example, in the case of the GrabCut dataset [3], most areas are labeled except for the border area between the object and the background (Fig 1.2(e)). However, providing rich seed information places a lot of input burden on users and is not a practical case. It is similar to the actual user's environment that seed information in the form of stroke is provided like the GSC dataset [4] (Fig 1.2(c)). In this situation, existing segmentation algorithms show unsatisfactory results due to a lack of seed information and an imbalance of foreground and background seeds.

We focused on seed information management from the fact that existing algorithms work well with rich seed information. When sparse seed information is given, we transform it into rich seed information to improve segmentation performance. That is, we proposed a technique for transforming seed information through the novel seed expansion step.

The Random Walk with Restart (RWR)-based algorithm [1] demonstrates an impressive performance but is not robust and consistent with sparse seed. We designed a seed expansion technique based on the RWR segmentation technique. Initially given seeds are the minimum human input with only a small amount of information; such information is insufficient to classify the object using the RWR model. We carefully design the seed expansion step such that the seed is expanded without losing the intention of the user. In Figure 2.1, RWR segmentation shows unsatisfactory result on the provided sparse seed information. Due to the lack of given seed information, the predicted segmentation boundary does not match the ground truth object boundary. Instead, our expanded seed information can help to capture the object boundary.

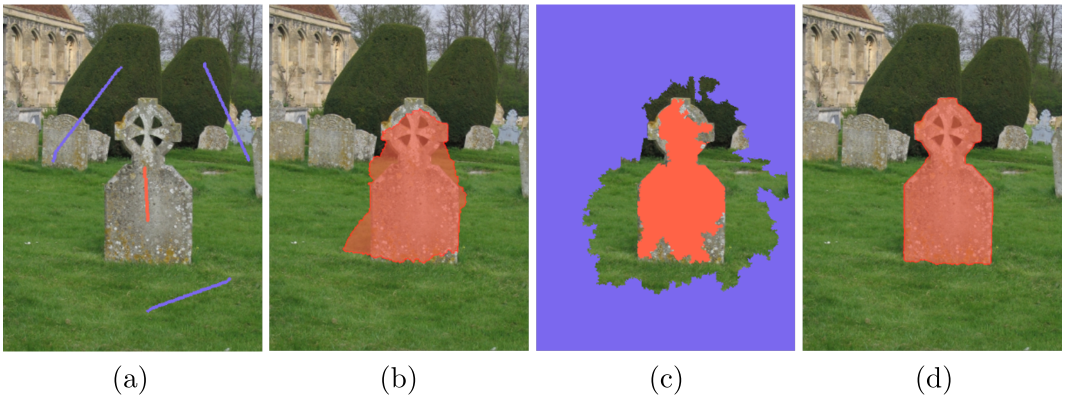


Figure 2.1: Interactive segmentation results. (a) The input image and its foreground and background seeds. An orange line denotes the foreground, and purple lines denote the background. (b) RWR segmentation results [1]. (c) Expanded seed through the proposed algorithm. (d) Segmentation result with our expanded seed.

Our seed expansion step is affected by the label propagation [25, 26]. The label propagation strategy is a method to increase the accuracy of learning on a sparse training set. This strategy adds valid samples from an unknown region into a training set and repeatedly trains the model. The label propagation concept is also used in interactive segmentation as a seeded region growing (SRG) segmentation algorithm [27, 28, 29]. In interactive segmentation, an area with seeds can be interpreted as a labeled sample, and a rest area can be interpreted as an unlabeled sample. By adding a reliable area among the unknown samples to the training sample, we can obtain more training samples. And the acquisition of more training samples leads to better estimation results. A representative work is the SRG technique [27], which starts from the initial seed set and expands the seed repeatedly through the steps. Each step of SRG, the most similar pixel among adjacent pixels is taken as an additional seed point. Growcut [29] also uses a similar algorithm concept, where cellular automaton is used as an image model. Automata evolution models the segmentation process. In each step, a labeled cell attempts to attack its neighbors. If the strength of the defender cell is lower than the attack force, then the label of the defender cell is changed to the label of the attacker cell. Inspired by the label propagation concept, we propose a novel seed expansion step comprising various types of region-enlarging methods for each stage.

Also, we adopt an image saliency model to improve the segmentation accuracy further. The RWR-based algorithm, which considers the relationships between adjacent pixels, can effectively manage local information. However, RWR-based algorithm cannot utilize global structure information. Therefore, integrating high-level cues to catch global information increases the performance of the segmentation. Many works have been published on high-level cues such as objectness or saliency [30, 31, 32, 33]. In this chapter, we use a saliency map [31, 34] to capture global information.

Finally, we boosted the speed of the segmentation algorithm. The RWR-based algorithm is difficult to apply in real life due to its slow operation speed. We applied the RWR algorithm to the coarse-to-fine structure to significantly improve the speed without losing the accuracy of segmentation.

2.2 Proposed Method

The purpose of our work is to increase the accuracy of interactive segmentation by enriching a seed that has scarce information. The proposed overall flow of the algorithm is shown in Figure 2.2. We called our proposed algorithm as **RWRexp**, which performs the RWR on *expanded* seeds. If an image and corresponding seed information are provided, then we can obtain an oversegmented superpixel map [35] and a saliency map [31, 34] that represents the image. With this information, given sparse

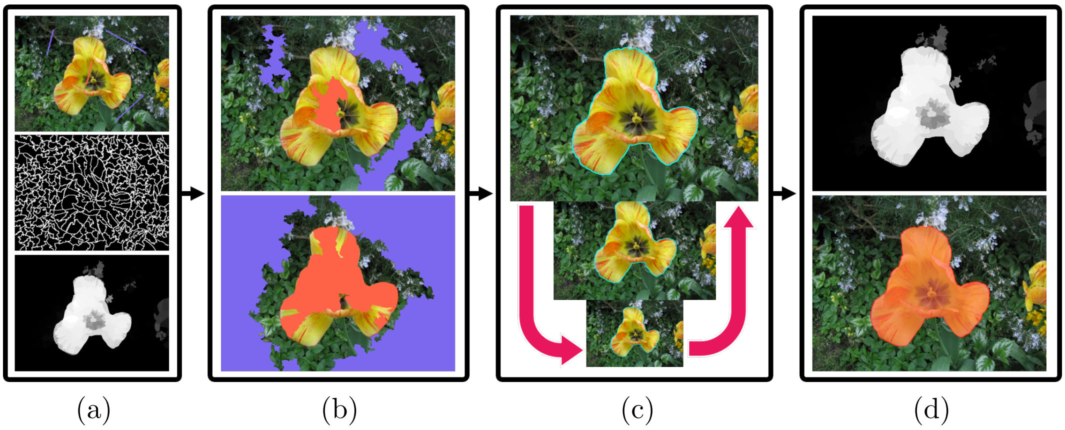


Figure 2.2: Overall pipeline of our algorithm. (a) Input information. Image and given seeds (*top*), superpixel (*middle*) and saliency map (*bottom*). (b) Two-step seed expansion process. *Step* 1 (*top*) and *Step* 2 (*bottom*). (c) RWR segmentation on a coarse-to-fine pyramidal structure. (d) Refinement step with saliency map

seeds are expanded through a two-step expansion procedure. We obtain a binary segmentation solution using these expanded seeds with the coarse-to-fine framework RWR segmentation. Finally, the final segmentation result is obtained through the refinement step using the saliency map. We will briefly review the main segmentation algorithm, RWR. Then, we explain how to improve the processing speed of the algorithm using a coarse-to-fine framework. Next, we will explain in detail the proposed two-step seed expansion process. Finally, we will cover the operation of the post-processing procedure using a saliency map.

2.2.1 Background

We briefly explain the RWR method for the interactive segmentation, which is our baseline algorithm [1]. The RWR-based method is similar to the interactive segmentation method that is based on the RW [2]. In RW, random walkers start from the given foreground and background seed pixels and move to the adjacent pixel. Herein, the probability of the movement direction depends on the color difference between the pixels. In the RWR algorithm, the restarting probability is added. According to the restarting probability c, a random walker returns to its user-supplied seed pixels at a constant probability. In other words, a random walker transitions to its closest pixel at a probability of $1 - c$ or restarts from its seed pixels at a probability of c. Finally, repeated transitions and restarts establish a steady-state distribution of a random walker.

Then, we next discuss how the RWR is imported to the interactive segmentation problem. Segmentation is a labeling problem in which each pixel $x_i \in X = \{x_1, ..., x_N\}$ is assigned a label $l \in L = \{foreground, background\}$. X is a given image that has N number of pixels. In a generative approach, the posterior probability is obtained using Bayesian rules.

$$p(l|x_i) = \frac{p(x_i|l)p(l)}{\sum_l p(x_i|l)p(l)}, \tag{2.1}$$

where the sum in the denominator is taken over all labels. The likelihood can be estimated as follows:

$$p(x_i|l) = \frac{1}{Z \times |M_l|} \sum_{m=1}^{M_l} p(x_i|x_m^l, l), \tag{2.2}$$

where Z is a normalizing constant and M_l is the total number of seeds of label l. This algorithm is not affected by the number of seed pixels due to the normalizing factor. The likelihood of each pixel is modeled by a mixture of distribution from each seed. Herein, the steady-state distribution of each seed is defined by the RWR.

An image should be represented as a graph to apply the RWR to image segmentation. The image segmentation is generally modeled by an undirected graph $G = (V, E)$ with nodes $v_i \in V$ and edges $e_{ij} \in E$. Each node v_i corresponds to a pixel x_i of a given image. The edge E between two nodes is determined by the neighborhood system. The edge weight w_{ij} represents the strength of edge e_{ij} or the similarity of two neighboring nodes, which is given by the simple Gaussian weighting function.

$$w_{ij} = exp\left(\frac{-|g(x_i) - g(x_j)|^2}{\sigma}\right). \tag{2.3}$$

The function $g(\cdot)$ represents the image colors of each pixel in Lab color space. Suppose a random walker starts from a m-th seed pixel x^l_m of label l in this graph G. The random walker iteratively transmits to its neighborhood with a probability that is proportional to the edge weight between them. The RWR model is formulated by defining an adjacency matrix $\mathbf{W} = [w_{ij}]_{N\times N}$ as follows:

$$\mathbf{r}^l_m = (1-c)\mathbf{P}\mathbf{r}^l_m + c\mathbf{b}^l_m, \tag{2.4}$$

where $\mathbf{r}^l_m$ represents the steady-state probability for the m-th seed of each label l that indicates either the foreground or background, c stands for the restarting probability, transition matrix $\mathbf{P}$ is the row-normalized version of the adjacency matrix $\mathbf{W}$, and $\mathbf{b}$ (indicating vector) contains the information on the locations of the seeds.

Notably, the $\mathbf{r}^l_m$ component that corresponds to pixel x_i can be regarded as the likelihood of a single seed at pixel x_i, $p(x_i|x^l_m, l)$. Therefore, we can calculate the total likelihood of a label l at a pixel x_i for all seed point x^l_m. We also obtain the posterior probability of each label by assuming the uniform distribution for prior. Comparisons of the posterior probability from each label directly assigned the foreground and background labels to each pixel x_i.

2.2.2 Pyramidal RWR

In practical use, an interactive segmentation system should have a rapid processing speed as well as accurate results. The existing RWR segmentation algorithm produces quite accurate results. However, the problem is the slow speed of the RWR segmentation algorithm. Thus, simultaneously catching accuracy and speed is an important issue. Hence, we analyze the RWR formulation and compose an effective structure to solve this issue. The RWR formulation can be analyzed into two types; **power iteration**, **matrix inversion**. We combine both strategies to form a pyramid structure that encompasses the accuracy and speed of the interactive segmentation system.

2.2.2.1 Power Iteration Formulation

The two types of formulation are classified according to the method of obtaining a steady-state solution. As in 2.4, the steady-state solution is obtained by converging the seed distribution. Intuitively, it can be formulated as follows:

$$\mathbf{r}^{(t)} = (1-c)\mathbf{P}\mathbf{r}^{(t-1)} + c\mathbf{b}. \tag{2.5}$$

We simplified the notations for convenience. As iteration t increases, $\mathbf{r}$ becomes a converged value, thereby obtaining a steady-state solution using this *power iteration.* However, one of the problems in using this strategy is that the number of iterations for convergence is unknown. Insufficient iteration causes an inaccurate steady-state solution, and excessive iteration causes time loss. However, the capability of handling iteration number can also be an advantage. If we can change the number of iterations, then the trade-off between time and accuracy can be controlled.

2.2.2.2 Matrix Inversion Formulation

Another method for obtaining the steady-state solution is *matrix inversion*. This method is used in a baseline RWR segmentation algorithm. After a simple matrix calculation, Formula 2.4 can be written as follows:

$$\mathbf{r} = c(I - (1-c)\mathbf{P})^{-1}\mathbf{b}. \tag{2.6}$$

If we use the preceding formula, we obtain a steady-state solution through a direct matrix inversion without an iteration process. We immediately obtain a more accurate solution than that obtained using a power iteration method. However, a large-size matrix inversion needs a heavy calculation. Therefore, it is accurate yet time-consuming. Another issue of matrix inversion is that the trade-off between time and accuracy cannot be controlled.

2.2.2.3 Pyramidal Structure

We used a pyramidal structure to combine the two methods. The image pyramid strategy of the coarse-to-fine framework has been used in many computer vision works [36, 37, 38]. The basic operation process is simple. First, we change the size of the given image to a coarse level to rapidly obtain the solution. Then, we bring the coarse level result into a fine level to obtain the ultimate solution.

Figure 2.3 shows the overall procedure of our coarse-to-fine pyramidal RWR segmentation. We used three steps of the Gaussian pyramid. For convenience, we name each step as the $P1$-layer, $P2$-layer, and $P3$-layer. $P1$-layer is the original scale image. $P2$-layer is the half-scaled image of the $P1$-layer, and the $P3$-layer is the half-scaled image of the $P2$-layer.

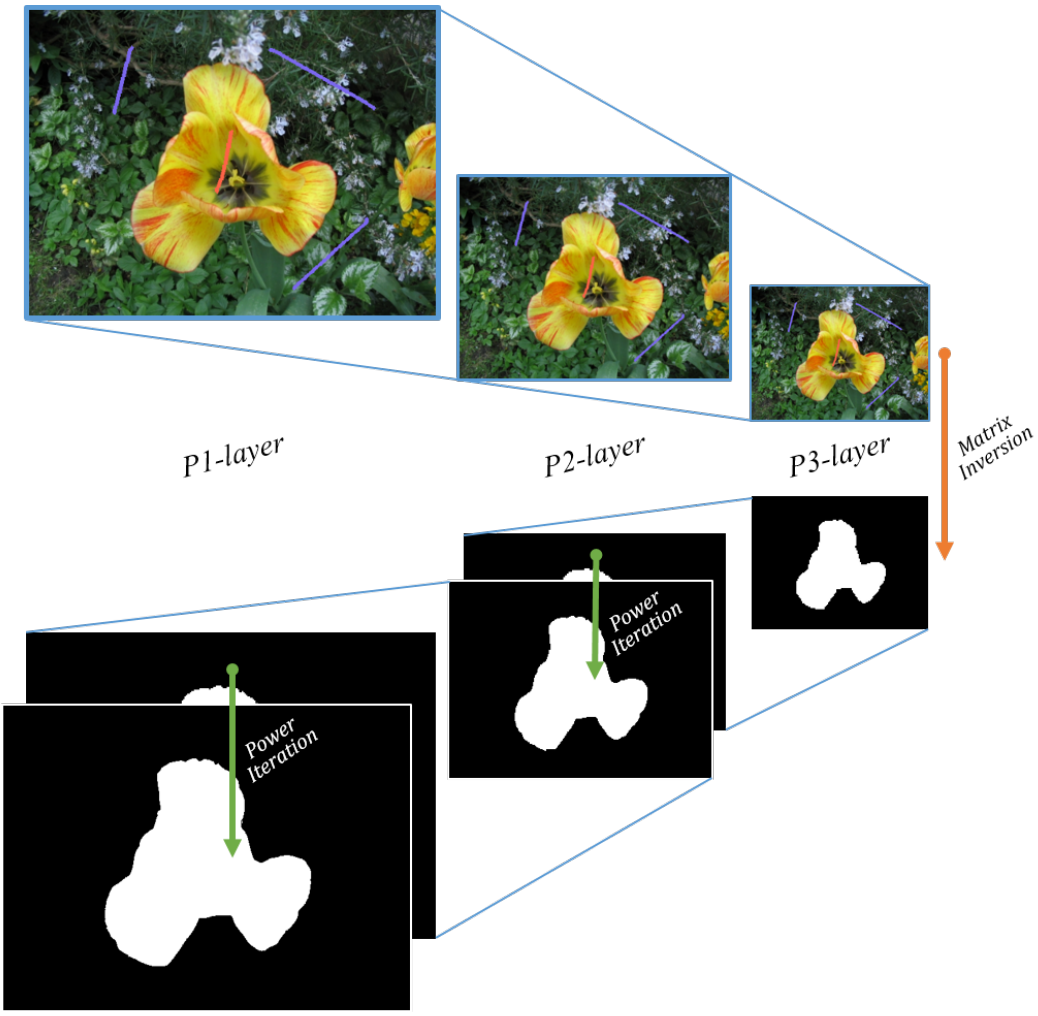


Figure 2.3: Pyramidal RWR segmentation procedure. Blue lines between pyramid layers denote a half-size scale change. Two types of RWR (Matrix inversion, Power iteration) are used for each layer.

An initial solution is obtained through accurate RWR segmentation with the matrix inversion in the $P3$-layer, which is the coarsest level. In this step, since the image size is sufficiently small, fast calculation of matrix inversion is possible. Next, we scale up the solution to fit the size of the following $P2$-layer. Our obtained solution can be regarded as the initial distribution to calculate the steady-state solution through power iteration. We assume that the initial distribution will not differ significantly from the final steady-state solution. Thus, the steady-state solution can be obtained with a small number of power iterations. In other words, it is more of a refinement step rather than finding a new solution. The same work goes to the last $P1$-layer. We obtain a final solution through scaling and refinement step. The overall operation time is greatly reduced through the pyramidal structure.

The novelty of pyramidal structure lies not only on its rapid speed but also on the employment of a large region information. Currently, the RWR method uses the values of the adjacent neighbor pixel. Thus, it depends only on local information. However, the pixel values in the coarse level contain information on the patch in the fine level. Hence, we can use a large region information because adjacent pixels in coarse level contain broad information. Using this coarse-to-fine strategy, we considerably enhance the convergence rate and robustness to local appearance variations by considering a broad region context in the image.

2.2.3 Seed Expansion

One of the problems of the RWR algorithm is that the segmentation boundary strongly depends on the positions of the foreground and background seeds. This dependence usually causes two problems; spreading and shrinking. The spreading problem indicates that the segmentation boundary can either be opened or expanded beyond the image boundary, whereas the shrinking problem prevents the full expansion of the segment region. These problems are the severe drawbacks of the RWR method.

We analyze the existing RWR segmentation and examine why the boundary of the segmentation is influenced by the seed location. When the random walker moves, its direction depends on the difference in information between pixels. Thus, the RWR captures the boundary between the foreground and the background in which the characteristics of pixel significantly change. The situation is quite different in practical application. When calculating the distribution of a random walker, unexpected situations occur. For example, a random walker can exhibit a low probability of existence due to the large distance from the seed pixel, not the difference in characteristics. This phenomenon means that the random walker cannot distinguish between the immediate characteristics change and accumulated one.

Therefore, the RWR successfully captures the boundary only when the accumulated change in characteristic can be ignored or cancelled out. It means the distance to the seed point is sufficiently small, or the distance to the foreground and background seed are nearly equal. Otherwise, the RWR might fail. Therefore, the RWR algorithm is influenced by the distance of the seed from the boundary. Thus, the RWR is sensitive to the initial seed distribution.

Figure 2.4 is shown as an example. The white and black dots represent a seed region, and we segment the blue and yellow regions. When each seeds are located around the boundary and the distances between the boundary and each seed are similar, the RWR segmentation works well. However, if the seed is separately located at the end of each side, then the RWR sets the boundary in the middle of the image due to the accumulating effect. Despite this drawback, we use the RWR segmentation

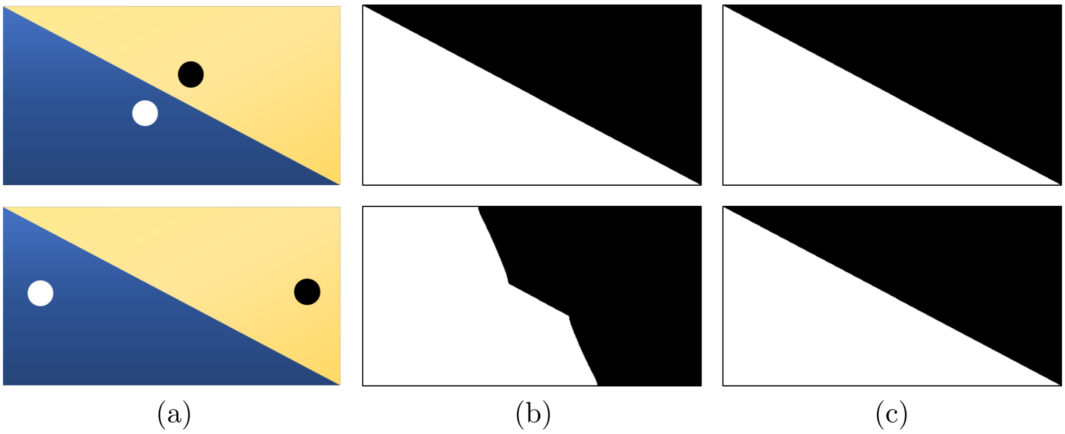


Figure 2.4: Synthetic experiment for the limitation of the RWR segmentation. (a) Synthetic image with seed points. White dot and black dot represent each seed point. (b) RWR segmentation results for different seed positions. (c) Our segmentation results for different seed positions.

as our baseline algorithm because satisfying segmentation results can be obtained if the seed information is sufficiently rich. Therefore we readjusted the seed location through our seed expansion step to overcome the shortcomings of RWR method.

2.2.3.1 Seed Expansion with Superpixel

We modify the seed information instead of the random walk algorithm itself to solve the seed dependency problem. We suggest region expansion of the seeds to place the modified foreground and background seeds near the boundary of the object.

Our proposed seed expansion takes place in two steps. The first step is expanding the seed from the pixel level to the *superpixel* level. The given image is oversegmented through the superpixel algorithm [35]. Then, if each superpixel contains a labeled seed pixel, we assign the label of the seed to all the pixels belong to the corresponding superpixel. Hence, seed information is expanded into the region level of the superpixel. It is a simple process, but effectively expanded the seed information. We call this step *Seed Expansion 1*. However, the transformed seed was not expanded enough to reach the object boundary. We will obtain a more expanded seed on the following stage.

Meanwhile, we use the superpixel information only in the seed expansion stage and not in the segmentation process. Many segmentation algorithms process their work based on the superpixel [39, 40]. If we process segmentation using the superpixel information, then we can rapidly get segmentation result. However, pixel-level segmentation accuracy may decrease. Furthermore, in an interactive segmentation system, the user can provide additional seeds if the result is unsatisfactory. However, if the superpixel does not capture the boundary of the object, then similar incorrect results will be obtained for the additional seed. Therefore, our segmentation is processed on the pixel level.

2.2.3.2 Seed Expansion with RWR

We further expand our seed in the second stage. As discussed above, the RWR segmentation cannot properly operate in areas far from the seed, but robustly operate near the seed. Therefore, the RWR segmentation results near the seed point are reliable. Using this feature of RWR, we extend the seed one step further. First, we bring the seeds obtained in the first expansion stage to the coarsest level of the pyramidal structure and obtain the likelihoods by RWR segmentation. At this point, we use a strong restart probability to obtain a reliable region only. Then, broad and reliable seeds are obtained by thresholding the likelihood probability with a conservative threshold value; T_F and T_B. The seeds obtained through this process differ slightly in the degree of expansion for each image. Overall, expanded seeds nearly stretch to the boundary of the object. Even if the expanded seed is not sufficiently extended to the boundary of the object, there is no problem if the seeds of each label are evenly

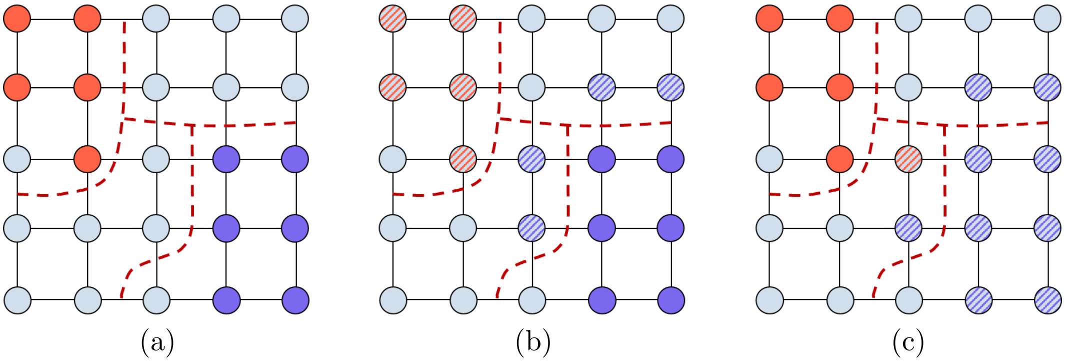


Figure 2.5: Case of seed configuration. (a) No adjacency between different types of seeds. (b) foreground seed superpixel and background seed superpixel are adjacent. The pixel with a diagonal line is removed in the seed reduction step. (c) Foreground seed and background seed are under the same superpixel. The pixel with a diagonal line will be removed.

distributed. The effect of each seed by the distance is canceled, and satisfactory segmentation results can be obtained.

However, the seed may possibly expand beyond the object boundary if it excessively expands. It is a difficult problem to determine the extent of expansion because the exact boundary location is not known. We suggest a few heuristic methods. The first way is to adjust the parameters. We delicately control the threshold value and the RWR restart probability to prevent the seed from expanding beyond the object boundary. Since excess is less than insufficient, We set parameters conservatively. These parameters could depend on image dataset. However, we use one fixed value for all datasets in this work.

Another method is the *seed reduction* process. This is a process applied to remove seed information when the seed is too expanded. That is, the foreground seed and the background seed are too extended to be adjacent to each other. In this case, we assume that the adjacent part is near the boundary of the object and reduce the nearby seed. As shown in Figure 2.5, (a) is the distribution we aim to achieve in the foreground and background seed distribution. Through the seed reduction process,

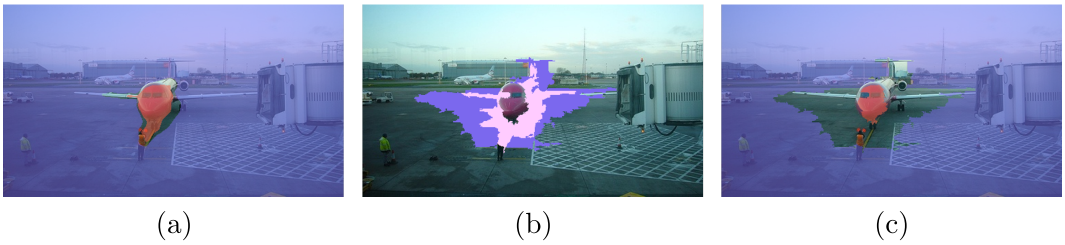


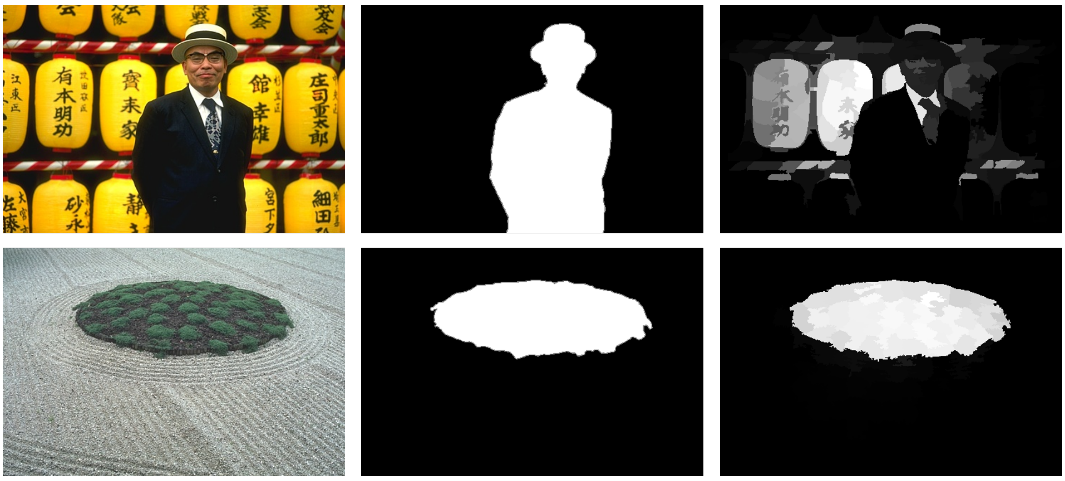
Figure 2.6: Seed reduction process. (a) Expanded seed before the seed reduction step. (b) Superpixels of the reduction target. The purple region is for the adjacent background superpixel case. The pink region is for the shared superpixel case. (c) The results of seed reduction.

we want to penalize cases (b) and (c). In (b), superpixels containing different types of seeds are adjacent. In this case, we remove all seed information belonging to the adjacent superpixel. In (c), one superpixel contains the foreground and background seeds. We first removed the information of the corresponding superpixel, and also increased the stability by removing the information of the adjacent background superpixel. Through this reduction strategy, we prevent the seed from invading into the object boundary. After the seed reduction step, we obtain a final expanded seed, which will be used in the segmentation process (Figure 2.6).

Segmentation results are achieved using expanded seeds at the coarsest level of pyramidal structure and are then recursively projected and refined at fine levels. The proposed seed expansion method produces robust results on the changes of the initial seed, as configured by the users. In Figure 2.4, our suggested algorithm locates the boundary of the color regardless of the location change in the seeds.

2.2.4 Refinement with Global Information

We use a wider information where we consider the neighbor in the coarsest level in the previous pyramidal structure. However, the patch-level information still remains at the local level. Therefore, we use additional information on the overall image. We



(a) (b) (c)

Figure 2.7: An example of a saliency map. (a) Input image (b) Ground truth image (c) Saliency map.

improved the accuracy of segmentation by introducing global cues into the refinement step. As global information for the image, we use the saliency map information. The saliency map represents the level of saliency for visual attention. The saliency map can set apart objects at different levels of attention. Thus, the saliency map increases the accuracy of the segmentation results because the saliency map can differentiate objects.

In this work, we use the results of the saliency map [31, 34] as a high-level cue to include global information. In several previous works, the saliency map and segmentation algorithm are jointly used [41, 42]. However, they used the saliency map as a direct labeling procedure that classifies salient foreground and background pixels. We attempt to use the saliency map as the global feature of the segmentation algorithm. We applied the saliency global information to the original algorithm in two ways.

First, the saliency map is used when calculating the weights between neighboring

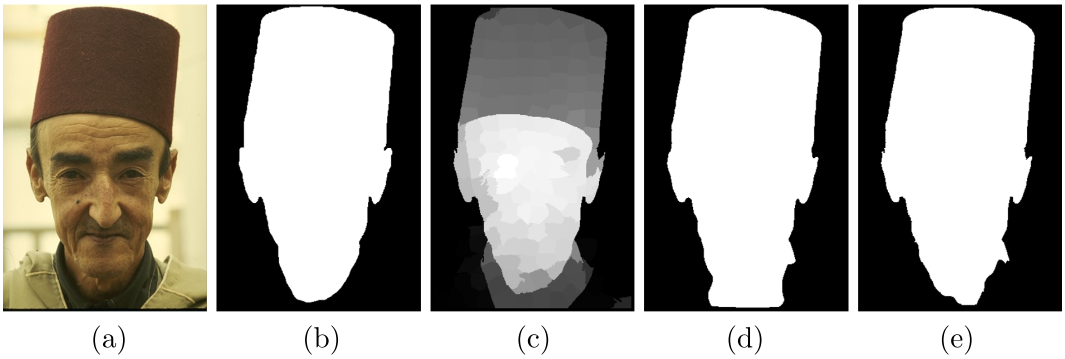


Figure 2.8: Saliency-based postprocessing. (a) Input image (b) Ground truth image (c) Saliency map (d) Segmentation mask before postprocessing (e) Segmentation mask after postprocessing

pixels. A new space containing saliency information is added to the existing *lab* color space when calculating the edge weight. Therefore, objects with similar attention may have similar labels. For saliency map X^S, we set weight function as follows:

$$w_{ij} = exp\left(- \frac{|g(x_i) - g(x_j)|^2 + \alpha|x_i^S - x_j^S|^2}{\sigma} \right), \tag{2.7}$$

for each pixel $x_i^S \in X_S = \{x_1^S, ..., x_N^S\}$. The salient region is not forced to be the foreground object. We only focus on the objectness property of the saliency map.

As a second way, we add a post-processing procedure to the final segmentation step. Usually, the areas highlighted in the saliency map are the most visible objects. However, in reality, the salient region and the region that users want to segment may not always be identical. As in the second row of Figure 2.7, the salient region and the ground truth mask are not always identical. Therefore, we first compare the segmentation results with the saliency map of the image to decide whether the salient region and the segmentation region are identical. We analyze the inner and exterior regions of the segmentation mask with respect to that of the saliency map. If the user wants to segment an object of high attention in the saliency map, then the saliency map probability is directly added to the segmentation probability to retrieve the final segmentation result, as shown in Figure 2.8. In other words, if the saliency map is certainly useful in segmentation, then the saliency map helps to post-process the ambiguous region. We propose an interactive segmentation algorithm with global information through this postprocessing step. Algorithm 1 represents the postprocessing process, and F and B represent the foreground and background labels, respectively.

Algorithm 1 Saliency-based postprocessing

```
Input: saliency map $X_S$, likelihood $P_F$, $P_B$, segmentation mask $X_M$
Output: segmentation mask $X'_M$
$F_{cross} = sum((X_M == True) \cdot X_S)$, $F_{num} = sum(X_M == True)$,
$B_{cross} = sum((X_M == False) \cdot X_S)$, $B_{num} = sum(X_M == False)$
$F_{match} = F_{cross}/F_{num}$, $B_{match} = B_{cross}/B_{num}$
if $F_{match} > thres_F$ and $B_{match} < thres_B$ then
    $P'_F = P_F + \beta \cdot X_S$, $P'_B = P_B + \beta \cdot (1 - X_S)$
    Decide label of $X'_M$ based on $P'_F, P'_B$
else
    $X'_M = X_M$
end if
```

2.3 Experiments

2.3.1 Dataset

We mainly conducted an experiment on the dataset, which is known to be complex and difficult, to evaluate the algorithm. Gulshan *et al.* [4] suggested an intricate dataset (GSC dataset) by collecting contemporary datasets, which comprise

49 sheets of GrabCut dataset images [3], 99 sheets of PASCAL VOC 09 segmentation challenge dataset [43], and 3 sheets of alpha-matting dataset images [44]. Each dataset contains an RGB and a seed image as well as a ground truth image. Each seed image consists of one line of foreground seeds and three lines of background seeds. In addition, we experimented with another dataset to verify the versatility of our algorithm. We use Weizmann single object dataset [45] which comprises 100 RGB images and a ground truth mask for a single object.

2.3.2 Implement Details

Only two parameters exist in the original RWR interactive segmentation; one for color variance σ, and the other for restart probability c. However, additional parameters must be set in our system. Since we adopted a pyramid structure, σ and c were set for each layer. We set $\sigma_1 = 1$, $c_1 = 0.5$ for $P1$-layer, $\sigma_2 = 1/20$, $c_2 = 0.5$ for $P2$-layer, and $\sigma_3 = 1/40$, $c_3 = 0.9$ for $P3$-layer. Notably, the coarsest layer has a high probability of restarting. The neighborhood system is also constructed differently in each layer. The $P1$-layer and $P2$-layer both have an 8 neighborhood system for fast refinement, but $P3$-layer uses a basic 4 neighborhood system. For the RWR refinement step, the power iteration is done for $P1$-layer and $P2$-layer. In this step, the number of iteration is crucial to the speed and accuracy of the algorithm. For the experiment, we set the iteration number as 10 for $P1$-layer and 30 for $P2$-layer. In preprocessing, the superpixel algorithm needs a target number of superpixels. We use 300 superpixels for each image regardless of its size. In the RWR seed expansion, $T_F = 0.5$ and $T_B = 0.8$ are used to prevent the excessive spread of the seed. Finally, in the saliency postprocessing step, we set $\alpha = 0.2$, $thres_F = 0.7$, $thres_B = 0.3$, and $\beta = 1$. With these parameters, 56 of the 151 images in the GSC dataset utilize the saliency map for postprocessing. Moreover, we use the same parameters for the seed expansion and the RWR segmentation steps.

2.3.3 Performance

2.3.3.1 GSC dataset

Our proposed algorithm (RWRexp) is compared with the four state-of-the-art seed-based interactive segmentation algorithms; GSC [4], LC [8], RW [2], and RWR algorithms [1]. We conducted the evaluation using the intersection over union (IoU) metric widely used in the segmentation field. IoU evaluates the degree of overlap between the segmentation output and the ground truth mask and is expressed in the following equation:

$$IoU = \frac{area(Output \cap GT\ Mask)}{area(Output \cup GT\ Mask)}(\%). \tag{2.8}$$

The segmentation output and GT mask are binary masks, and the *area* represents the number of pixels in the region. Table 2.1 shows the accuracy of each algorithm. The LC algorithm shows the best results among the existing algorithms. And RWR, the baseline algorithm, records the second-highest IoU score. We experimented with the proposed RWRexp modifying the baseline RWR and obtained the best results. Through this, we can confirm that the proposed seed expansion algorithm worked effectively. Only the change for a given seed led to better results under the same RWR algorithm.

The qualitative results are shown in Figure 2.10. The first column shows the input image and its corresponding seed location, the second column is the ground truth image, and the last column shows our expanded seed location. Our proposed

Table 2.1: GSC dataset result. The intersection over union result for each method.

Method	GSC [4]	LC [8]	RW [2]	RWR [1]	*RWRexp*
IoU (%)	57.94	62.01	52.49	60.27	**67.09**

Table 2.2: Weizmann dataset result. The intersection over union result for each method.

Method	GSC [4]	LC [8]	RW [2]	RWR [1]	*RWRexp*
IoU (%)	76.76	70.98	58.88	68.21	**79.13**

algorithm presents a remarkable performance in a scarce seed test.

2.3.3.2 Weizmann dataset

We compared segmentation results of various algorithms using Weizmann dataset [45] to show the robustness of our algorithm. Since there is no data containing seed information in the Weizmann dataset, we artificially created the seed information for the experiment. Similar to GSC dataset, we created annotation information, which comprises one foreground scribble and three background scribbles. Each scribble forms a straight line. We experimented with state-of-the-art algorithms as in the GSC dataset case. Table 2.2 shows the IoU for the experimental results. Although our baseline RWR algorithm does not provide the best result, the proposed algorithm evidently shows the best result. Figure 2.11 presents the qualitative results for the Weizmann dataset.

2.3.4 Contribution of Each Part

We conducted additional experiments on the GSC dataset. Table 2.3 shows the contribution of each component to the algorithm. When we apply the basic RWR to

Table 2.3: Contribution of each part. It shows the result of changing one part at a time in the baseline algorithm. *Expansion* 1 denotes superpixel expansion and *Expansion* 2 denotes RWR expansion process. *Saliency* denotes the saliency-based postprocessing step.

Algorithm	*IoU* (%)
RWR [1]	60.27
RWR Pyramid	58.42
RWR P. + *Expansion* 1	63.73
RWR P. + *Expansion* 2	60.71
RWR P. + *Exp.* 1 + *Exp.* 2	64.92
RWR P. + *Exp.* + *Saliency*	67.09

the pyramidal structure, the accuracy is lower than the baseline RWR. We traded some precision for fast algorithm speed. Accuracy is significantly enhanced when we use the expanded seed through superpixel seed expansion. Furthermore, accuracy is also increased if we apply the RWR seed expansion without the superpixel seed expansion. Therefore, accuracy increases if we simultaneously use both seed expansion modes. Finally, we achieve the highest level of accuracy when we apply a saliency map into the weight and the postprocessing step for global information.

2.3.5 Seed Consistency

We experimented on seed consistency for the suggested algorithm and baseline. Interactive segmentation algorithms should be designed to provide reliable segmentation results in diverse seed distribution. The baseline RWR is sensitive to the position of seeds. We tested our algorithm to determine whether our algorithm could overcome the seed sensitivity of the RWR not only in synthetic experiments but in real data. In Figure 2.9, when seeds are located around the object which we attempt to segment, RWR and RWRexp work well (first and third rows). However, when

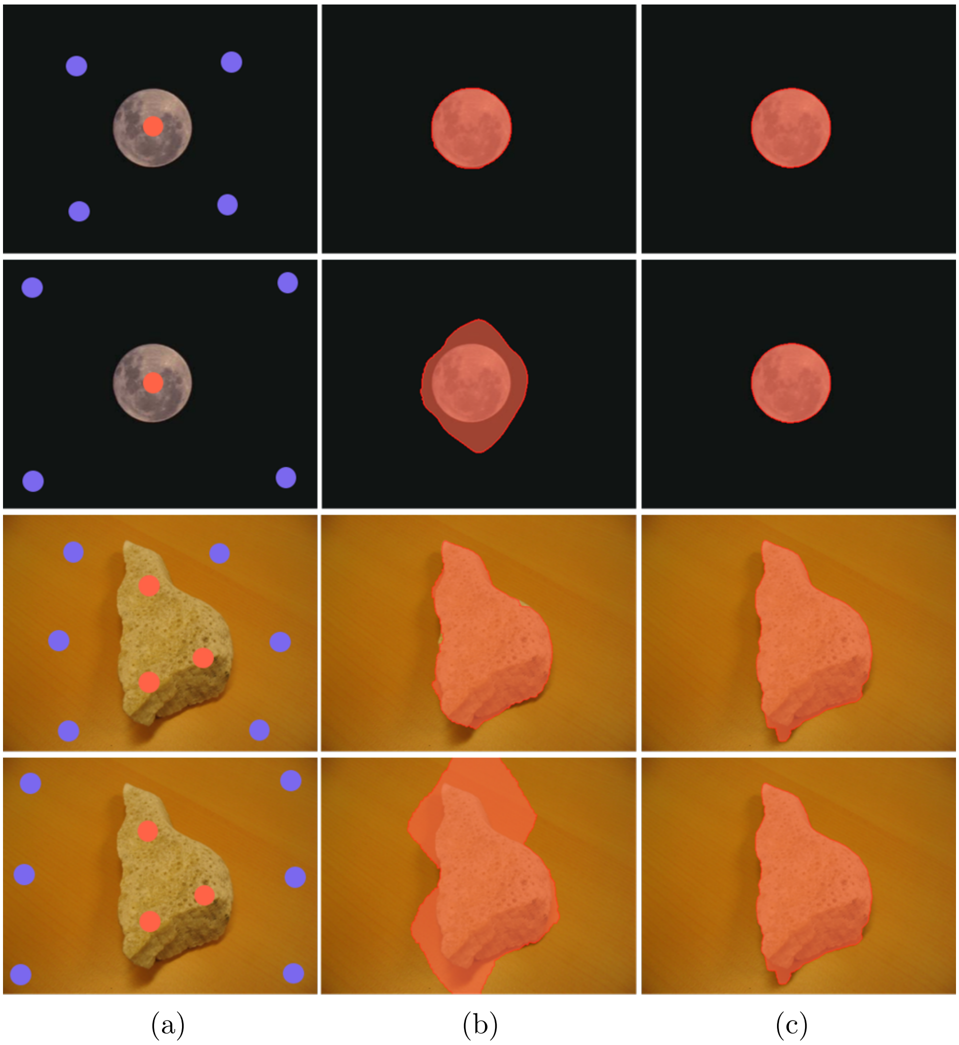


Figure 2.9: Seed location consistency. Each rows show individual experiment results. (a) Input image and seed location. Orange dot denotes the foreground seed and purple dot denotes the background seed. (b) Results of RWR. (c) Results of RWRexp.

seeds move apart from the object, the RWRexp performs well, whereas the RWR fails (second and fourth rows). We observe that the suggested algorithm is robust for location changes of seeds.

2.3.6 Running Time

We cannot ignore the running time in an interactive segmentation system. The running time of interactive segmentation can be divided into the offline and the online running time. The former is the period for preprocessing a given image, whereas the latter is the period for producing output with the provision of the user seed information. Normally, we cannot immediately obtain a satisfactory result at once. Thus, the user usually provides the additional seed until the result is satisfactory. In this respect, we insist that the online running time of the algorithm is more important than the offline running time. Our proposed algorithm aims to decrease the online running time.

In our system, the preprocessing period, which is a part of the offline running time, includes the time taken to obtain the superpixel and saliency map. Thus, preprocessing time is dependent on the each algorithms. However, once preprocessing information is obtained, we can reuse the information on the additional seed input of the user. This preprocessing factor is irrelevant to the online running time. Therefore, we compared the online running time of the proposed interactive segmentation with that of the baseline RWR. The average running time of the proposed algorithm is **0.661s**, which is faster than the **2.312s** average running time of the RWR on a 4.0GHz Intel quadcore i7 CPU and 32GB RAM.

2.4 Summary

We have proposed a new interactive image segmentation framework via a pyramidal RWR with seed expansion. The proposed algorithm produces highly stable and accurate segmentation results by using reliable expanded seeds. In addition, the proposed algorithm allows fast convergence through the coarse-to-fine strategy. Also, the segmentation performance was improved by introducing global information using a saliency map. The experimental results demonstrate that the proposed algorithm produces a superior performance compared with the existing algorithms.

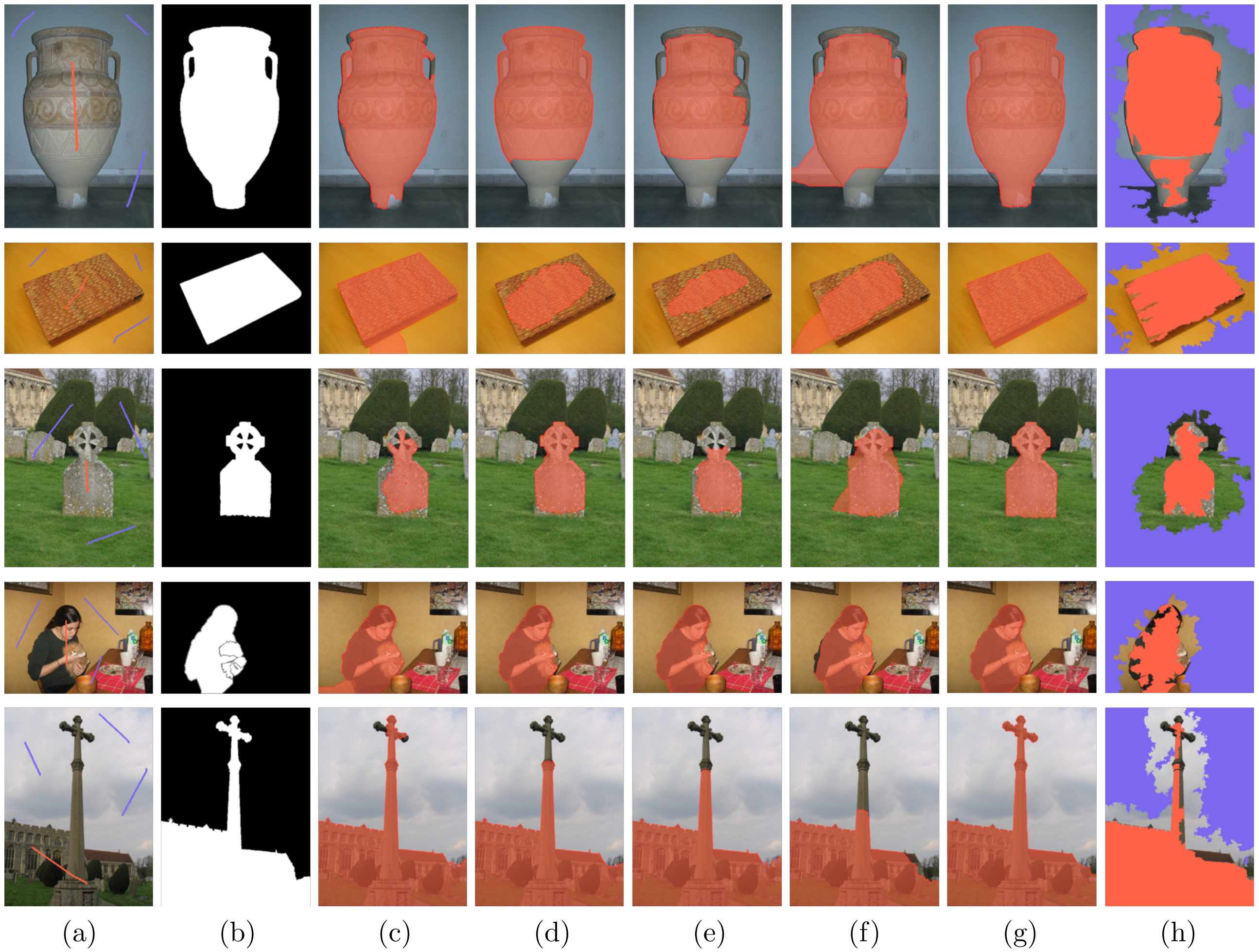


Figure 2.10: Segmentation results for GSC dataset (a) Input image (b) Ground truth (c) Geodesic star convexity (GSC) (d) Laplacian Coordinates (LC) (e) Random walk (RW) (f) Random walk with restart (RWR) (g) Proposed algorithm (RWRexp) (h) Our expanded seed

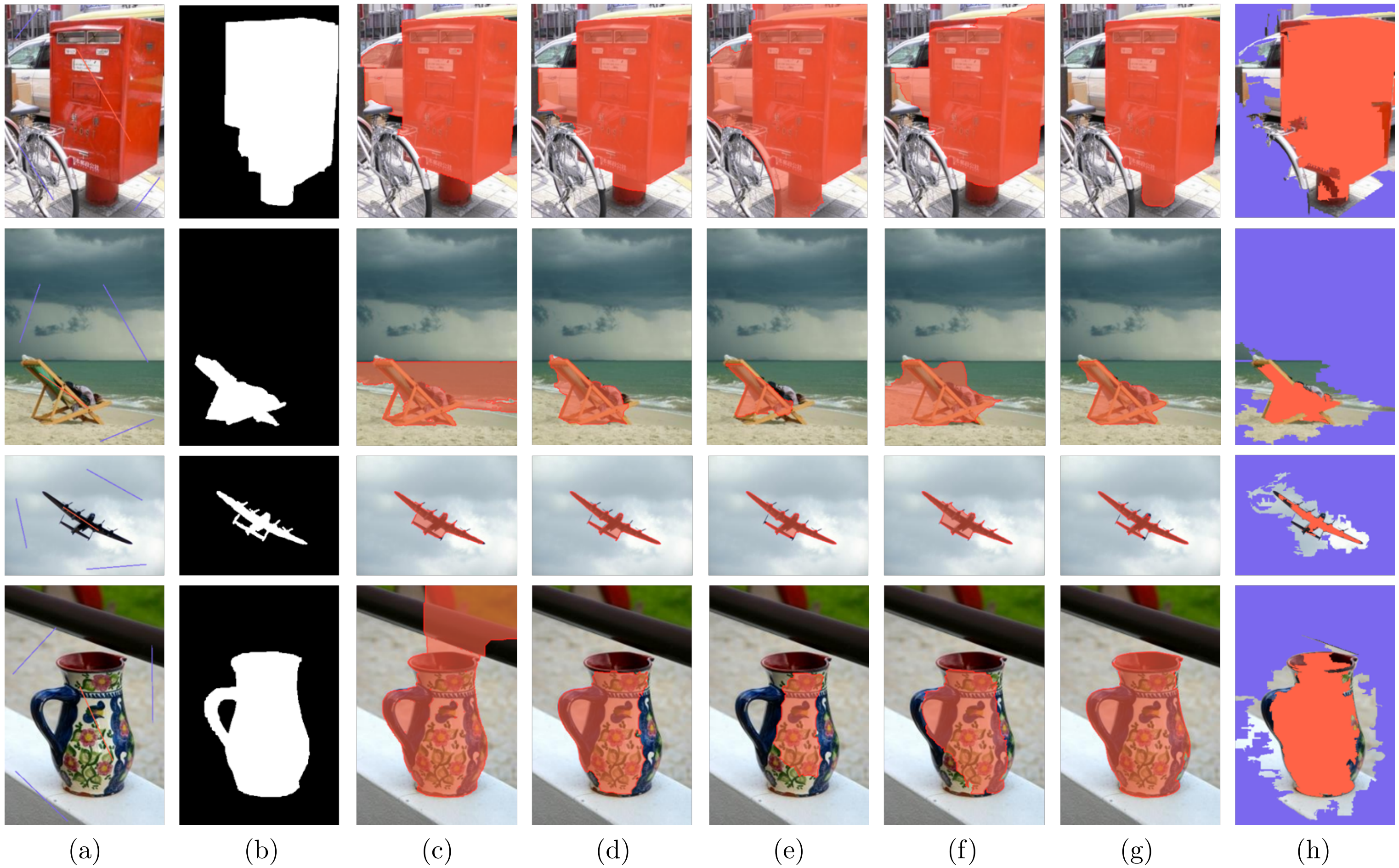


Figure 2.11: Segmentation results for Weizmann dataset. (a) Input image (b) Ground truth (c) Geodesic star convexity (GSC) (d) Laplacian Coordinates (LC) (e) Random walk (RW) (f) Random walk with restart (RWR) (g) Proposed algorithm (RWRexp) (h) Our expanded seed

Chapter 3

Interactive Segmentation with Seed Generation

3.1 Introduction

In the previous chapter, we introduced the seed expansion technique using the RWR model. Despite its simple operation, it brought a good performance improvement. However, there are some limitations in RWR expansion. First of all, the operation of the algorithm varies depending on the threshold value. Since each image data has different properties, it is challenging to expect sufficient seed expansion with a fixed value. Another is that the expanded range of the seed is limited around the seed region. If the seed is concentrated on one side, it is not easy to expand to the other side. In this chapter, we propose a more advanced seed enrichment method by generating a new seed point.

One of the critical components of the interactive segmentation algorithm is robust object extraction while matching the human intention. For many objects with a complex background, the user often has to spend much effort to refine the results obtained from the algorithm. In this regard, how to reduce human effort while maintaining the performance in interactive segmentation is very important. In [4], the number of additional efforts by users is used as a measure of system performance. In this chapter, we propose a novel technique to simulate the human process of guiding the interactive segmentation system to obtain the desired object. When the user enters a point on the desired object and a point on the background, our system automatically generates the sequence of artificial user input to accurately localize the target object of interest, as illustrated in Figure 3.1. The proposed system is designed to achieve high performance while significantly reducing the number of user inputs.

In this chapter, we formulate the automatic seed generation problem as a sequential decision-making problem and train the seed generation agent with deep reinforcement learning. Our agent starts by analyzing the image and the foreground/background segmentation produced with the initial seeds by the user, and then determines a new foreground or background seed. After creating a new segmentation by combining the created seed with the initial seeds, our agent uses this segmentation as a next input and repeats the process of creating seeds. Deep reinforcement learning is suitable for our task because we cannot define globally optimal seed at some stage of interactive segmentation. Additionally, for effective learning, we propose a novel reward function depending on the intersection-over-union (IoU) score. The advantage of the proposed system is that consistent performance has been achieved in images in unobserved datasets as well as in previously observed datasets.

The contributions of this chapter include (1) the introduction of a Markov Decision Process (MDP) formulation for the interactive segmentation task where an

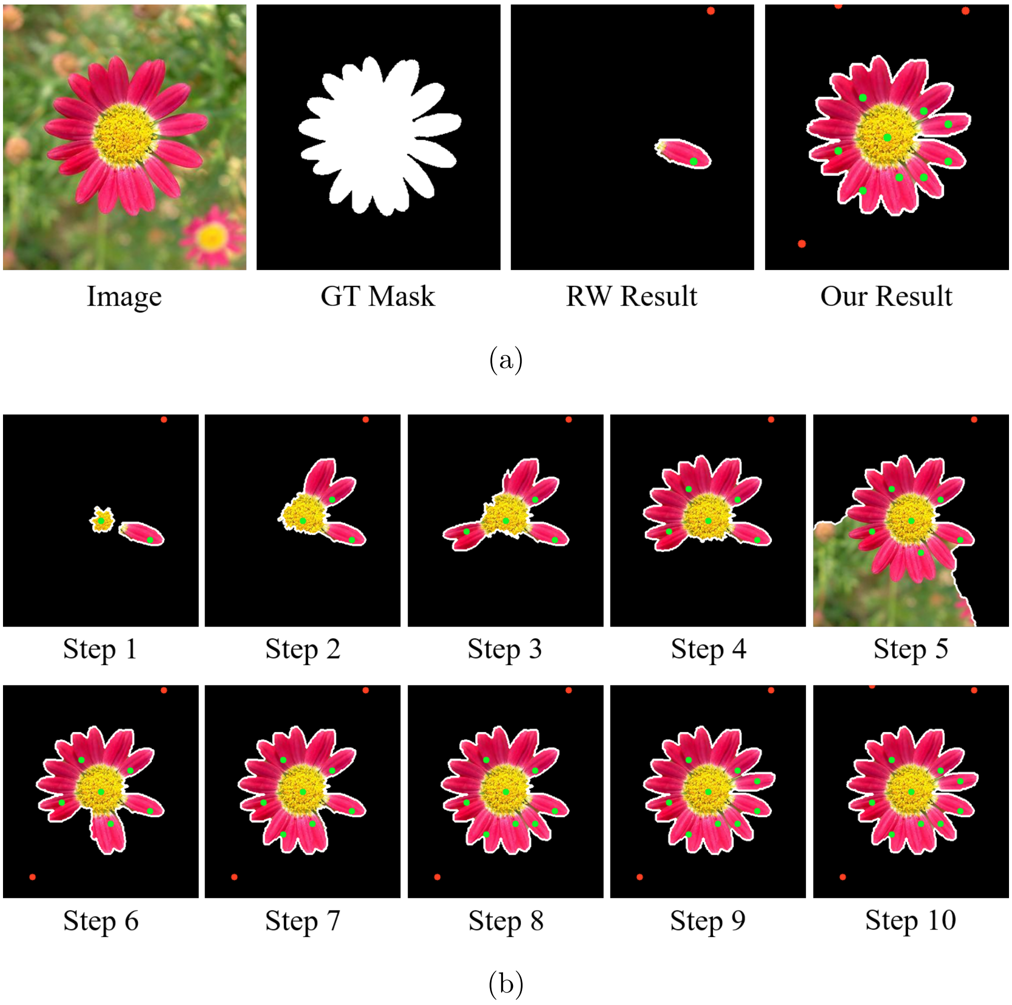


Image GT Mask RW Result Our Result

(a)

Step 1 Step 2 Step 3 Step 4 Step 5

Step 6 Step 7 Step 8 Step 9 Step 10

(b)

Figure 3.1: An automatic seed generation example. The green and red dots represent the foreground and the background seeds, respectively. (a) RW result is the output of random walker segmentation [2] algorithm with the initial seeds and our result is the segmentation output with the generated seeds from the SeedNet. (b) Seed generation process through the SeedNet. At each step, the SeedNet creates a new foreground or background seed input.

agent puts seeds on the image to improve segmentation and (2) the novel reward function design to train the agent for automatic seed generation with deep reinforcement learning.

3.2 Related Works

Deep reinforcement learning: Research on deep reinforcement learning has been actively carried out due to its excellent performance in an Atari game via Deep Q-Network (DQN) [46]. Techniques such as prioritized experience replay [47], double DQN [48], dueling DQN [49], and A3C [50] have been studied to improve the performance of the reinforcement learning algorithm. The reinforcement learning algorithm is often applied in Atari games or robotics problems, but it also has many potential applications in computer vision fields.

A typical application to computer vision using reinforcement learning is the object localization problem. In [51], the authors interpreted the object localization problem as a sequential dynamic decision-making problem. In each decision step, an action is represented by the transformation of a detection box. With a deep representation of an image and previous actions, DQN predicts the action of next step. Similar to [51], [52] used box transformation actions and DQN to predict the next action. They employed a tree-structured search to enable the localization of multiple objects in a single run.

Reinforcement learning framework is also used for image classification problems [53], image captioning [54], video tracking [55], face hallucination [56] and video activity recognition task [57]. Andreas *et al.* [58] applied reinforcement learning to solve the question answering problem. They trained a network structure predictor with reinforcement learning technique.

In most computer vision applications, researchers used a combination of attention models and reinforcement learning. However, we solved the problem of generating seed points by directly using the image space as a large action space.

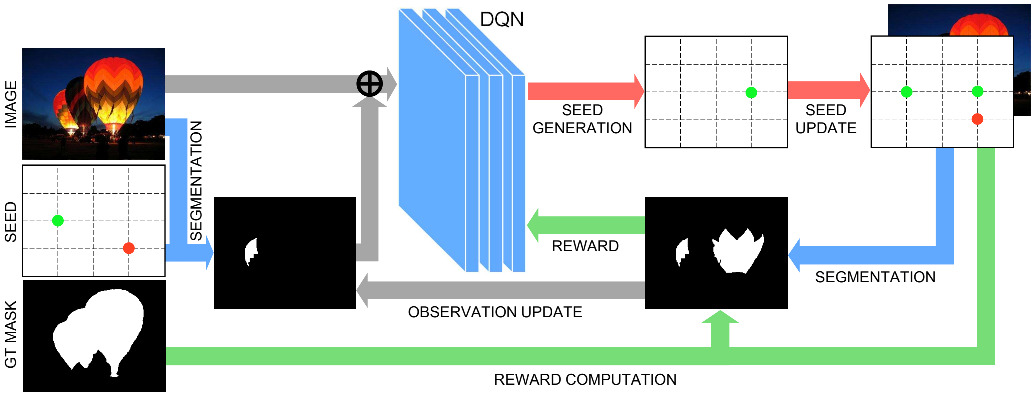


Figure 3.2: Overview of the proposed SeedNet. The image and the segmentation mask are the input of the DQN. The seed set is updated using the newly created seed from the DQN, and the mask is generated using the revised seed set. The obtained mask is used to calculate the reward value by comparing with the GT mask, and this process is repeated. The gray arrows indicate state-related behavior, red arrows indicate action-related behavior, and green arrows indicate reward-related behavior.

3.3 Proposed Method

3.3.1 System Overview

In this chapter, we propose a novel automatic seed generation system for the task of interactive segmentation. We call it **SeedNet**. When an image and sparse seed information are entered, the ultimate goal of the proposed system is to create additional seed points and obtain accurate segmentation result. The core module of SeedNet is a deep reinforcement learning agent for generating artificial seed points. Also, SeedNet includes an off-the-shelf segmentation model that performs the segmentation operation with the generated seed. The entire system is constructed by learning the DQN [46] agent using the segmentation result.

The overall process of SeedNet is shown in Figure 3.2. The operation of the system proceeds with the image and the initial seed map given by the user. By utilizing this input information, performing interactive segmentation yields a binary mask. We use Random Walk (RW) segmentation [2] as an off-the-shelf interactive segmentation algorithm. The obtained binary mask and image are concatenated and then input to the DQN. The DQN model proposes new seed information by using the input. The new seed information contains the position and label of the proposed seed. As a result, the seed map is updated by adding the proposed seed point to the existing seed information. In addition, segmentation of the image using the new seed information results in a new binary mask. The obtained binary mask is used for two purposes: the first is to compute the reward signal by comparing the obtained mask with the ground truth (GT) mask. The reward is a value that evaluates the operation of the DQN and is used to update the network. Second, the acquired binary mask is used as an observation of the next iteration.

The sequence of cyclic operations is repeated throughout the training process. However, during the test time, the reward part is omitted, and only the seed generation process is performed. By repeating the steps of generating a seed, a seed map containing several artificial seeds is obtained. In this way, we significantly reduce the human effort on interactive segmentation task.

3.3.2 Markov Decision Process

The core part of the proposed SeedNet is to generate a sequence of seeds by the agent. We define the problem as an Markov Decision Process (MDP) consisting of state, action, and reward and the agent operates through the MDP. The agent takes the current state as an input, performs some action, and receives a corresponding reward. This section presents the definition of the proposed MDP.

State: The state should contain enough information to allow the agent to make the best choice. For our problem formulation, information on the whole image is essential. Additionally, the state should include information on observation that changes at each step. We can obtain two kinds of information when a seed is generated at every step: one is the newly created seed map, and the other is a binary mask using off-the-shelf interactive segmentation algorithm. Given that we want the proposed system to be robust to the seed position, we exclude the seed position information and add only the binary mask information to the state. In addition, past observations are not used, and only the current observation is utilized as the state.

As a result, in our formulation, the state is defined as the current binary segmentation mask and image features. Unlike many existing works, the proposed system does not use any deep feature representation as the state.

Action: Given a state, the agent selects an action within the action space. In our formulation, the action is defined as a positioning new seed point. The agent decides the label (foreground/background) and position of the seed in the 2D grid given the states. If we set the 2D grid to correspond to all the pixels in the image, the action space becomes too large, causing problems in training. Therefore, the 2D grid where the new seed can be placed is sparsely set to 20×20 size. There are a total of 800 kinds of actions because of the foreground and background grids. If an agent selects one of 800 actions, a new seed point is created at the corresponding location. Meanwhile, there is no explicit terminal action because it is hard to define the termination station. Thus, we terminate the process after proposing 10 seed points.

Reward: The reward signal evaluates the result for the action of the agent. Generally, in a game environment, a score or win/loss is used as a reward function. In our system, the results of agent action are seed position and segmentation mask. Thus, we can use the accuracy of the segmentation mask as a score concept. The accuracy of the mask can be determined by comparison with the ground truth (GT) mask. For evaluation, IoU is the common metric. Therefore, the intuitive basic reward function is to use IoU as a reward function. The reward function with IoU is described as R_{IoU}.

$$R_{\text{IoU}} = IoU(M, G), \tag{3.1}$$

where M denotes the obtained segmentation mask and G denotes the GT mask. Another basic reward function is to use the change trend of IoU. It compares the IoU value of the current mask with the IoU of the previous step mask and gives a success signal if the value is increased and a failure signal if it is decreased. It is like win/loss reward signal in the game environment. In our environment, however, we can obtain the amount of change as well as the direction of change. Therefore, a more flexible reward signal can be designed by using the variation of IoU as the value of reward instead of the binary type reward. It is described as R_{diff}.

$$R_{\text{diff}} = IoU(M, G) - IoU(M_{\text{prev}}, G), \tag{3.2}$$

where M_{prev} is the segmentation mask of the previous step. In addition, by using an exponential IoU model(R_{exp}) instead of a linear IoU model, we can design a reward signal that gives more attention to changes in high IoU values.

$$R_{\text{exp}} = \frac{exp^{k*IoU(M,G)} - 1}{exp^{k} - 1}, \tag{3.3}$$

where k is a constant value. Meanwhile, given that we have information on the seed position as well as information about the mask, we can generate an additional signal to assist the IoU reward. Instead of judging success/failure by using the change in IoU, we can judge by comparing GT mask with the newly generated seed. That is, if the label of the new seed matches the GT label of the corresponding location, it is a success; otherwise, it is a failure. With a similar concept, we divide the GT mask into four regions and compare them with the seed label. To divide GT mask into four regions, we create additional boundaries inside and outside the object that give some margin from the object boundary. That is, four regions are generated from three boundaries, including an existing object boundary. These four regions are named strong foreground (SF), weak foreground (WF), weak background (WB), and strong background (SB), in the order from the center of the object to the edge of the image. When a new seed point is assigned, different reward functions are applied to the divided areas according to seed type.

For example, if the newly given foreground seed belongs to the SF area of the mask, we apply exponential IoU reward. Also, if foreground seed belongs to the WF domain, it is also a success case but is not recommended, so a reduced reward signal is applied. Otherwise, if foreground seed is wrongly suggested on the background area, a fixed reward value of -1 is returned. Likewise, when a new background label seed is given, we can obtain a reward similar to the foreground case. The proposed reward function R_{our} is as follows:

$$R_{\text{our}} = \begin{cases} R_{\text{exp}} & \text{if } F_{\text{seed}} \in \text{SF or } B_{\text{seed}} \in \text{SB} \\ R_{\text{exp}} - 1 & \text{if } F_{\text{seed}} \in \text{WF or } B_{\text{seed}} \in \text{WB} \\ -1 & \text{otherwise} \end{cases}, \tag{3.4}$$

where F_{seed} means foreground seed and B_{seed} means background seed. We obtain a continuous score reward from the mask information and a discrete success/failure reward from the seed information. Finally, we propose a novel reward function by mixing the two types of reward. We compare the differences between the newly proposed reward function and other reward functions in the experimental section.

3.3.3 Deep Q-Network

With the proposed MDP formulation, the seed generation agent can be trained through the deep reinforcement learning. In this study, we use the DQN algorithm by Mnih *et al.* [46] to train the agent. DQN learns the action-value function $Q(s, a)$, the expected reward that the agent receives when taking action a in a state s. After training, the agent selects the action with the learned Q-function. The Q-learning target can be defined with the given s, a, s':

$$r + \gamma max_{a'} Q(s', a'), \tag{3.5}$$

where r is the reward, γ is a discount factor, and s' and a' represent the state and action of the next step, respectively. DQN is a technique that approximates the Q-function with a deep neural network. The loss function for training the Q-function can be expressed:

$$Loss(\theta) = \mathbb{E}[(r + \gamma max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2]. \tag{3.6}$$

For effective learning, we employ various techniques from Mnih *et al.* [46]. First, we use a target network to solve the problem of poor learning stability. By introducing a target network separately from the online network, the parameters of the target network during a few iterations are fixed while the online network is updated.

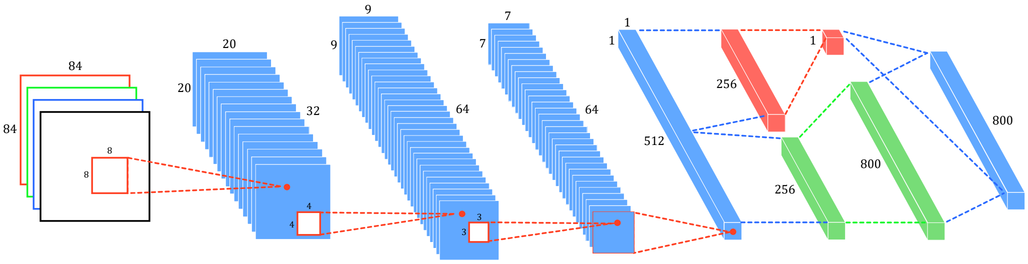


Figure 3.3: DQN architecture for SeedNet. The red block is the network for the state value function, and the green block is the network for the advantage function. Numbers denote dimension sizes (width, height, channel).

This method has significantly improved the stability of learning. Next, we use an ϵ-greedy policy as a behavior policy. The ϵ-greedy policy uses a random action with a probability of ϵ and an action that maximizes the Q-function with a probability of 1-ϵ. The last is experience replay to solve the correlation problem of data used for DQN learning. We created an experience replay buffer, proceeded with the episode, and stored the replay memory in the buffer (s, a, r, s'). During the learning process, samples of the batch size are randomly selected from the buffer to reduce the correlation between the data.

3.3.4 Model Architecture

The DQN used in this chapter is shown in Figure 3.3. The structure of DQN used is almost similar to that of [46]. To improve the performance of the algorithm, we use the double DQN structure of [48] and dueling DQN structure of [49]. The input image and the binary mask resulting from the segmentation at the previous stage are resized to 84 × 84 and input to the network. Three convolution operations followed by ReLU activation are performed on the input. By taking advantage of the dueling structure, the 512-D layer after the fully-connected operation is split into two parts to learn the advantage function and state value function. Then, through a fully-connected operation, the advantage function $A(a, s)$ comes out as an 800-D output corresponding to the action space size. Meanwhile, the state value function $V(s)$ is a scalar value. Finally, the advantage function is added to the state value function to obtain the Q-function. The action is determined according to the Q-function having the maximum value. If the action label is less than 400, it will be the foreground seed. Otherwise, it will be the background seed and reduces the action label by 400 for conversion to grid coordinates. Finally, converting the action label to 20×20 grid coordinates will determine where the new seed will be located.

3.4 Experiments

We have experimented with several types of datasets. First, we use the MSRA10K saliency dataset [59] to train and compare our results against the initial results from the initial seed. We also conduct a comparative experiment on various single object datasets that were not included in the training dataset.

3.4.1 Implement Details

SeedNet is trained for MSRA10K saliency dataset from scratch. In the training process, 10,000 pre-training steps are preceded to build an experience replay buffer to be used for learning. During the pre-training step, the actual learning does not proceed, but the experience that goes through the episode is stored in the buffer. We used 50,000 experience as a buffer and 32 as a batch size. For exploration, we use ϵ-greedy policy. During training, ϵ decreases from 1 to 0.2 over 10,000 steps. In the subsequent training process, ϵ is fixed to 0.2. As the learning progresses, the action is randomly selected as the probability of ϵ, and the action according to the learned network is selected by the probability of 1-ϵ. The parameters for the specific network size are shown in Figure 3.3, and the discount factor γ is set to 0.9. Each episode contains a total of 10 seed point generation processes. For training, we use an Adam optimizer [60] and utilize a learning rate of 1e-4. Also, the update rate to the target network is set to 1e-7. As previously mentioned, a 20×20 size grid is used as the action space, and the k value of the exponential reward function is set to 5.

3.4.2 Performance

First, our performance evaluation is done on the MSRA10K dataset. The MSRA10K dataset consists of 10,000 images, and we use 9,000 of them as training and the remaining 1,000 as test. Each image consists of an RGB image and a mask representing the GT, and seed information is not included. The size of the image is approximately 400×300 pixels. To accelerate the learning process, each image and GT are reduced to 1/4 size in the learning stage. The same image size of 84×84 is input to the DQN during training and testing. However, when segmentation is performed with a newly generated seed, segmentation is applied to a 1/4 size image in the learning process to obtain a fast result, and the original image size is used in the test time. As the size of segmented images increases, the size of the seeded points increases simultaneously. In training, a circle with a diameter of 3 pixels is used as a seed, and a circle with a diameter of 13 pixels is used as a seed in the test.

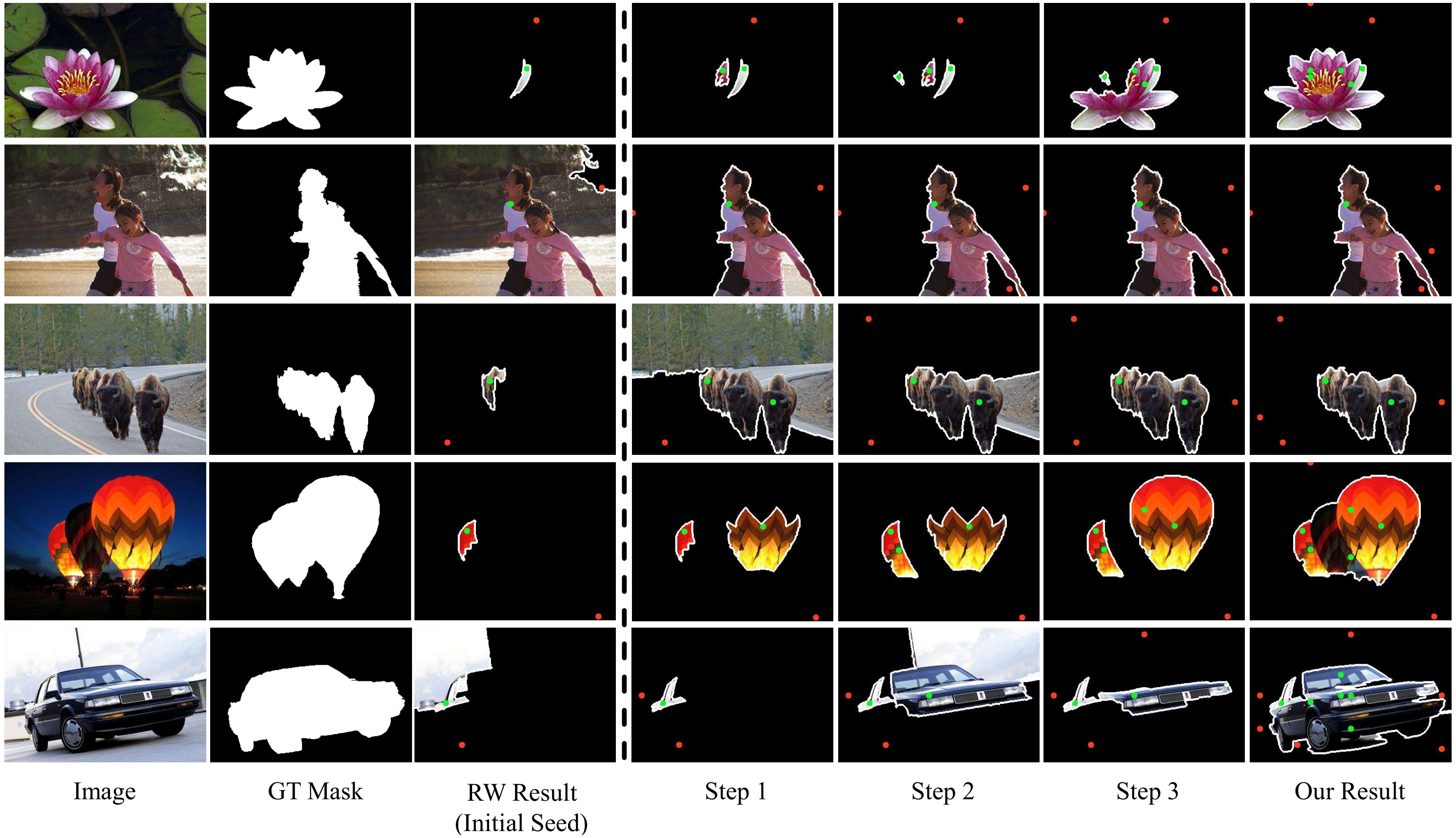


Figure 3.4: MSRA10K results. The left part shows the input image, GT mask, and initial seed with corresponding RW [2] result. The right part shows the SeedNet result, showing the first three steps and final result.

Table 3.1: MSRA10K Result. IoU results for 5 randomly generated initial seed sets.

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
RW [2]	39.59	39.65	39.71	39.77	39.89	**39.72**
SeedNet	60.70	60.12	61.28	61.87	60.90	**60.97**

Table 3.2: Comparison with supervised methods. IoU score for each methods.

Method	FCN [13]	iFCN [12]	*SeedNet*
IoU	37.2	44.6	**60.97**

Given that seed information is not included in the MSRA10K dataset, we experiment with initial seed point randomly generated using the GT mask information. We apply dilation and erosion separately to the GT mask to form a region slightly distant from the object boundary and randomly select foreground and background seed points from each region. As the initial seed point is determined randomly, we perform five experiments sequentially and evaluate the performance using the average value. We use the RW segmentation method as an off-the-shelf segmentation algorithm in our system. The results obtained using only the initial seed point and the newly proposed seeds of this system are compared and shown in Table 3.1. The IoU metric is used for evaluation.

The results show that the accuracy is significantly increased when seed information generated by the proposed SeedNet is used compared with RW segmentation using only the initial seed. Meanwhile, we change the initial seed distribution from Set 1 to Set 5, but it is not significantly affected by the initial seed distribution, and both RW and SeedNet show similar results. Qualitative results are shown in Figure 3.4. As shown in the figure, the automatically generated seed information gives better results than the initial seed. Figure 3.4 also shows the results up to step 3 and the final result. The average number of seeds used until saturation is **5.39** clicks. Therefore, the threshold of the proposed algorithm, which proposes generation up to 10 times, is reasonable. However, given that SeedNet generates a seed on a sparse grid, it is difficult to propose a seed in a finer position as in the case of the third row. Nevertheless, the additional seed is well presented without losing the intention of the initial seed. More experimental results are on Figure 3.9.

Comparison with supervised methods: Additionally, we implement the FCN [13] and the iFCN [12] baseline. We input 80×80 image similar to our network input size, change the fully-connected layer to convolution layer in our network, give padding to make 10×10 output map, and perform deconvolution to the original size. Also, networks are trained from scratch. We add two seed input channels to the RGB channel for iFCN. The results are shown in Table 3.2. Although it is possible to obtain better performance by using the pretrained network and larger images, it is observed that the supervised segmentation has lower performance in the current configuration.

Failure cases: Figure 3.5 shows the failure cases of the proposed algorithm. First, as shown in the left-most figure, there are cases where the algorithm fails because the learned user's intention and the actual user's intention are different. Also, as in the middle column case, the seed point was accurately generated, but it is insufficient to give satisfactory results. Finally, as in the right-most figure, there are cases of failure by simultaneously generating a foreground seed and a background seed on the same object. Failure cases occur when sufficient space search is not performed during training or the user's intention is ambiguous. Improvements in the RL algorithm or the usage of other semantic cues can be introduced later as improvements.

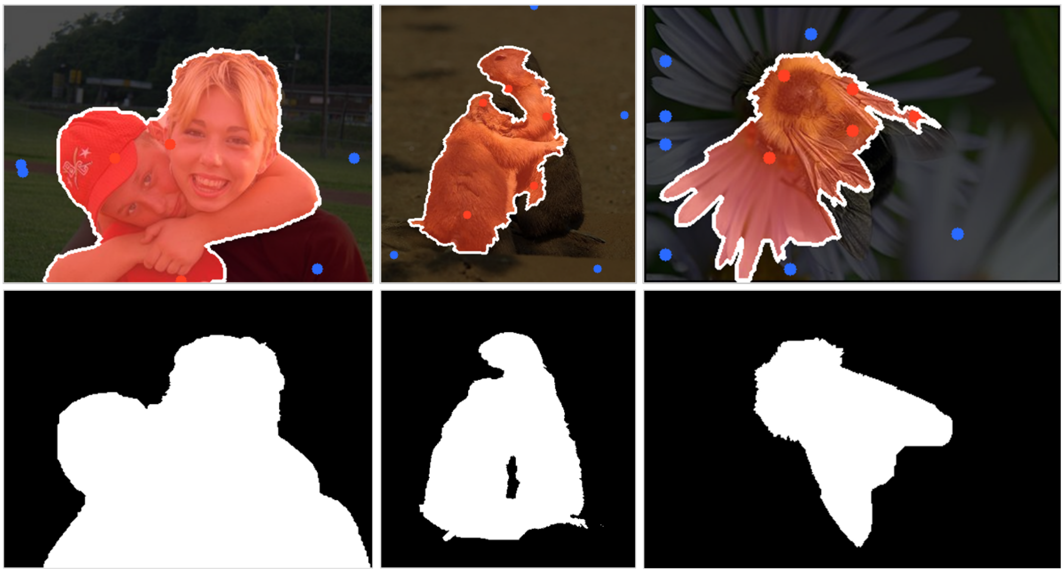


Figure 3.5: Failure cases. *Upper row* : the result of the proposed algorithm and proposed seed points. Red dots denote foreground, and blue dots denote background. *Bottom row* : GT mask.

3.4.3 Ablation Study

To analyze the proposed system, we replace several key components of the system. Experiments are carried out while changing only the corresponding elements and keeping other parts intact.

3.4.3.1 Reward function

Our DQN is updated with a reward comparing the GT with the observation. To verify the effectiveness of the proposed reward function, we train the system using a simple reward described in 3.3.2. For comparison, R_{IoU} and R_{diff} are used, and the change in reward value according to the learning time and the change in IoU accuracy of the training set according to the learning time are shown in Figure 3.6. The reward axis shown on the left has different axes for each graph because the scales are different for each reward function. Meanwhile, the IoU axis on the right

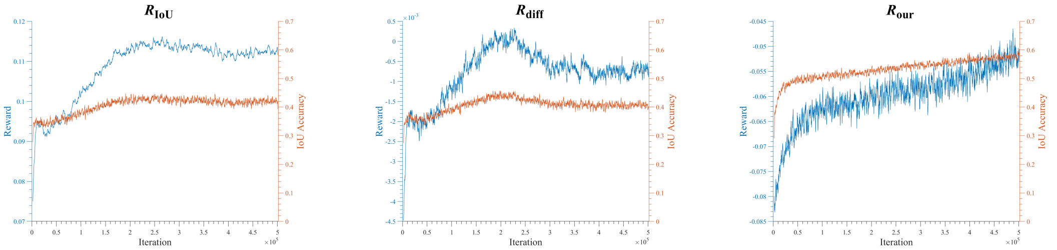


Figure 3.6: SeedNet learning progress graph using R_{IoU} (left), R_{diff} (center), and R_{our} (right). The reward value is indicated by the blue line and the left axis, and the IoU value is indicated by the orange line and the right axis. A common x-axis represents the progression of the learning iteration. For better visualization, the change is displayed every 100 steps and each point represents the running average value for 1000 steps.

Table 3.3: Ablation Experiments : Reward. IoU results for 5 randomly generated initial seed sets.

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
RW [2]	39.59	39.65	39.71	39.77	39.89	**39.72**
R_{IoU}	42.00	42.77	43.69	42.96	41.33	**42.55**
R_{diff}	44.33	44.80	45.09	44.19	43.82	**44.45**
R_{our}	60.70	60.12	61.28	61.87	60.90	**60.97**

has the same axis for all three graphs. Comparing the three graphs, we can see that simple reward functions initially increase in reward value but stay at a certain level, so that IoU no longer improves. Meanwhile, in the proposed reward function, both the reward and IoU values are steadily increased. The result of applying SeedNet learned by each reward function to the test set is shown in Table 3.3. As expected, we can confirm that the proposed reward function has better results than other reward functions.

3.4.3.2 Segmentation Method

SeedNet uses RW as an off-the-shelf segmentation algorithm, which can be replaced by other algorithms. SeedNet is trained using GrabCut (GC) [3], GSCseq (GSC) [4]

Table 3.4: Ablation Experiments : Segmentation. IoU results for 5 randomly generated initial seed sets. Each version of *SeedNet* is according to an off-the-shelf segmentation module.

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
GC [3]	38.15	38.29	38.35	38.70	38.71	**38.44**
SeedNet (GC*ver.*)	52.43	51.89	51.84	52.10	52.26	**52.10**
GSC [4]	57.85	58.10	58.50	58.57	58.70	**58.34**
SeedNet (GSC*ver.*)	63.09	62.70	64.24	63.16	64.19	**63.48**
RWR [1]	35.35	36.8	35.96	35.17	35.28	**35.71**
SeedNet (RWR*ver.*)	52.56	55.15	52.61	52.38	52.49	**53.04**

and RWR [1], respectively. The results are shown in Table 3.4. all versions of SeedNet show an increase in IoU compared with the initial results. As other segmentation algorithms can be applied in this way, better results can be expected using CNN based algorithms, such as iFCN [12]. The results of using diverse segmentation methods are shown in Figure 3.7.

3.4.4 Other Datasets

To verify the scalability of the proposed SeedNet, we conducted experiments on other datasets. We show that our algorithm can be applied universally by experimenting on a dataset that has not been used for training. As this system is trained using the saliency dataset, MSRA10K, we test our agent on various single-object binary segmentation datasets instead of the validation images of the MSRA10K datasets. The experimental setup is the same as that of MSRA10K, and the evaluation is also performed with an average IoU for five random initial seeds.

GSCSEQ [4]: This dataset consists of a total of 151 images, including 49 pieces

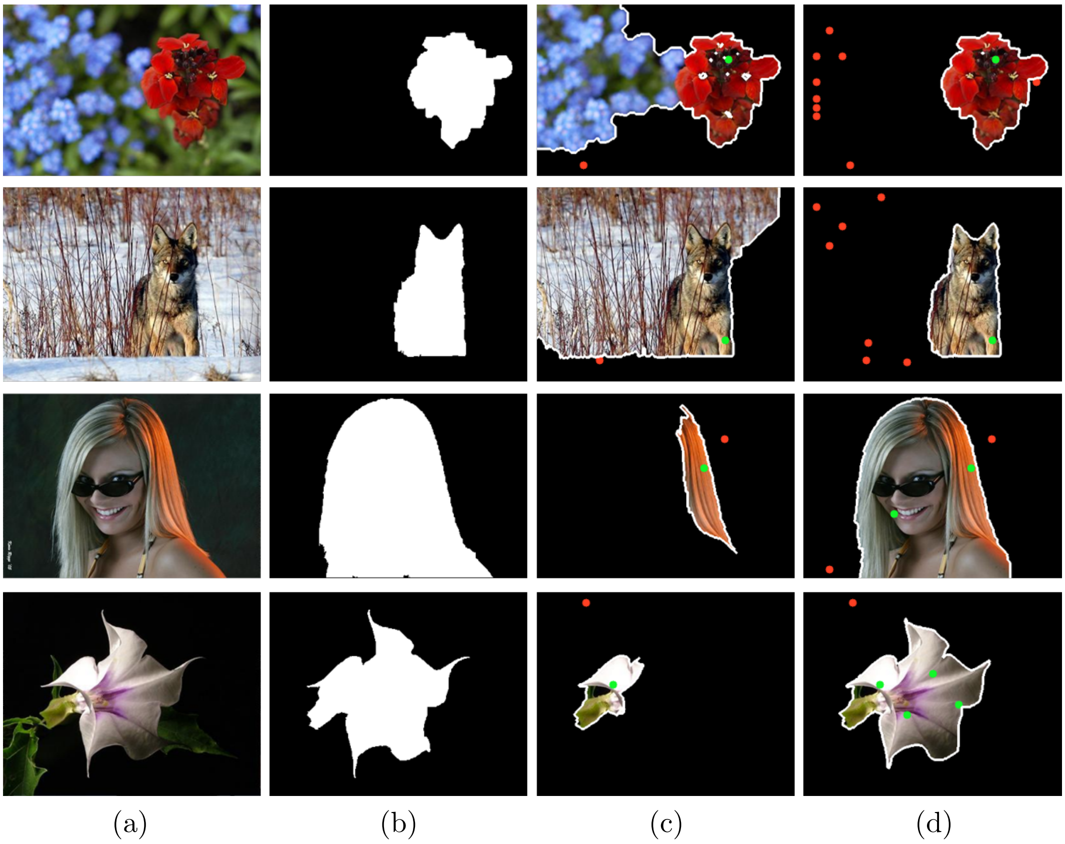


Figure 3.7: MSRA10K result with SeedNet GC version (upper two rows) and GSC version (bottom two rows). (a) Input image (b) GT mask (c) Baseline result (d) Our result. Green dots denote foreground, and red dots denote background.

from the GrabCut dataset [3], 99 pieces from the Pascal VOC dataset [43], and 3 pieces from the Alpha matting dataset [44]. The dataset includes RGB images, GT binary masks, and scribble information. However, in this experiment, seed points are generated from the mask without using scribble information.

Weizmann Single Object [45]: The Weizmann single object dataset consists of 100 single object images, including three types of GT binary masks for each image. The three types of GT are slightly different depending on the subject of the labeling user, and we only use the first GT for evaluation.

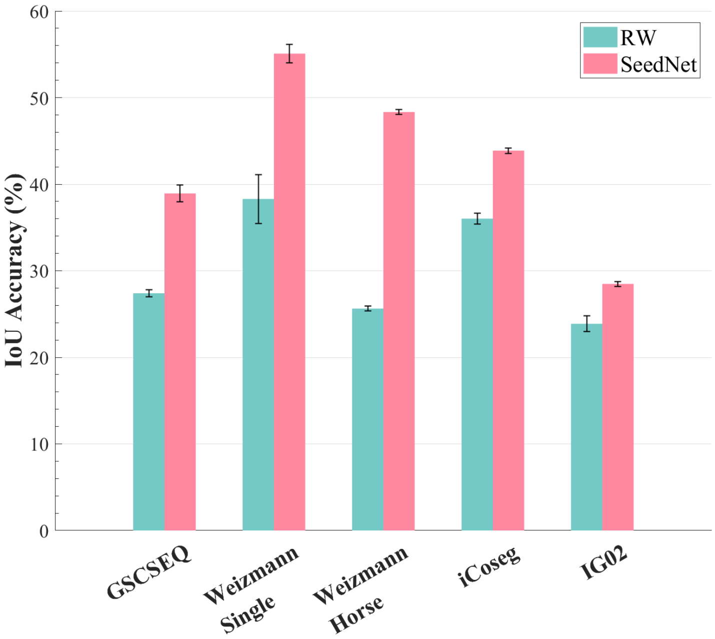


Figure 3.8: Results for other datasets. The horizontal axis represents each dataset, and the vertical axis represents the average IoU accuracy.

Weizmann Horse [61]: A total of 328 images contain a side view of the horse. The dataset contains images and GT binary masks.

iCoseg [62]: iCoseg is a dataset mainly used for cosegmentation, and it has 38 categories and consists of 643 images in total. There are GT binary masks for each image.

IG02 [63]: The new annotation of the Graz-02 dataset [64] from INRIA consists of three categories: bikes, cars, and people. A total of 479 test images from each category are used for this experiment. Some images contain several objects, but only one object is tested in this experiment.

The experimental results are shown in Figure 3.10. In all five datasets, we can see that the result of using seed generated through SeedNet is significantly improved compared with the initial seed. In particular, the Weizmann Horse dataset shows an increase in accuracy of more than 20%. SeedNet, on the other hand, is relatively weak for the IG02 dataset, where multiple objects exist because we only train from a single object case. Nevertheless, we can confirm that the proposed SeedNet is applied well even though it is a dataset of different nature that has never been seen during training. Qualitative results are on Figure 3.10.

3.5 Summary

We have proposed a novel interactive segmentation agent for assisting a user to segment an object accurately. The agent can predict the user's intention and reduce the user's effort. Also, this approach has the potential to leverage the user's intent in various computer vision problems such as semantic segmentation. Furthermore, our agent can help to reduce the cost of pixelwise labeling task.

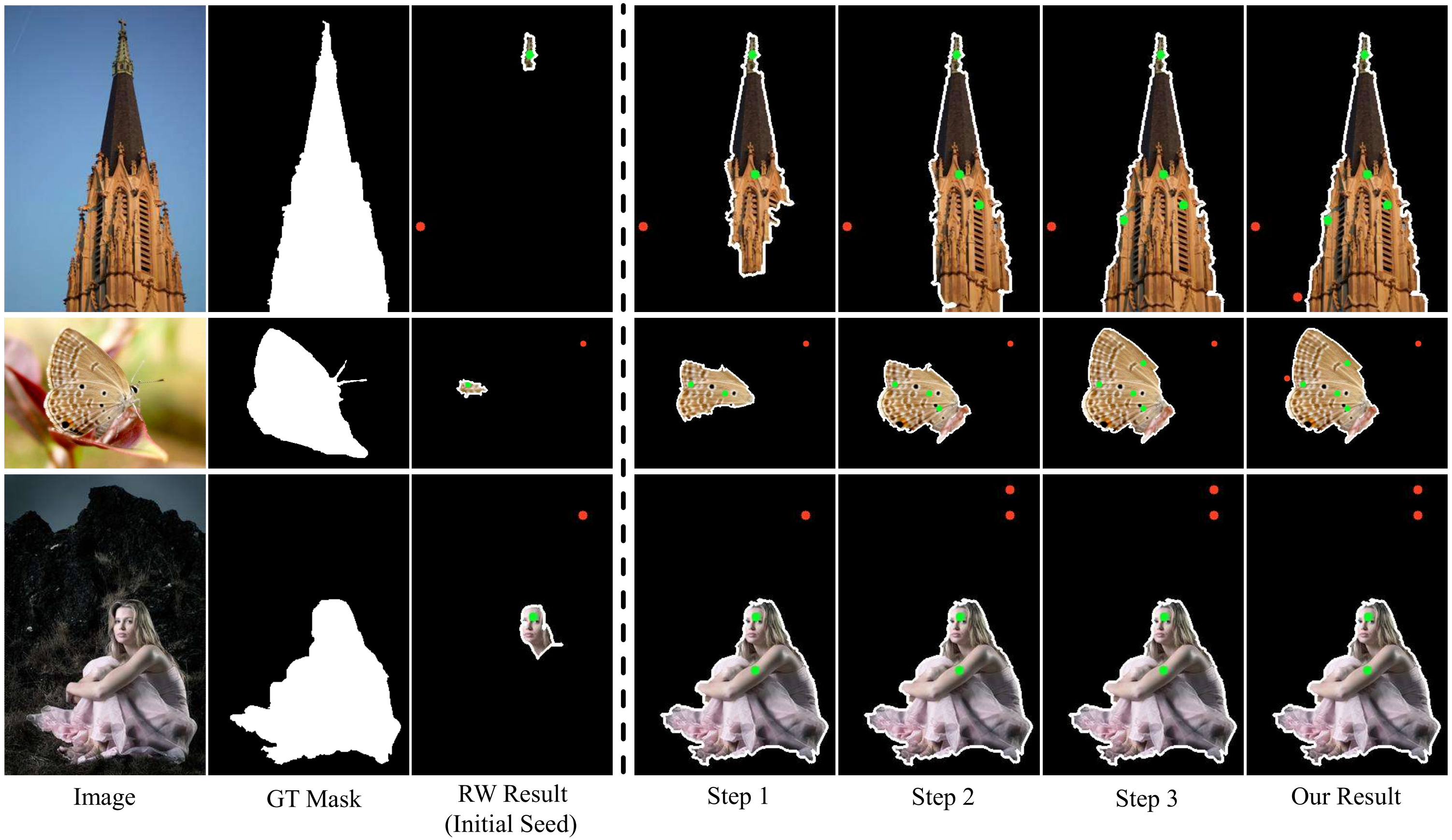
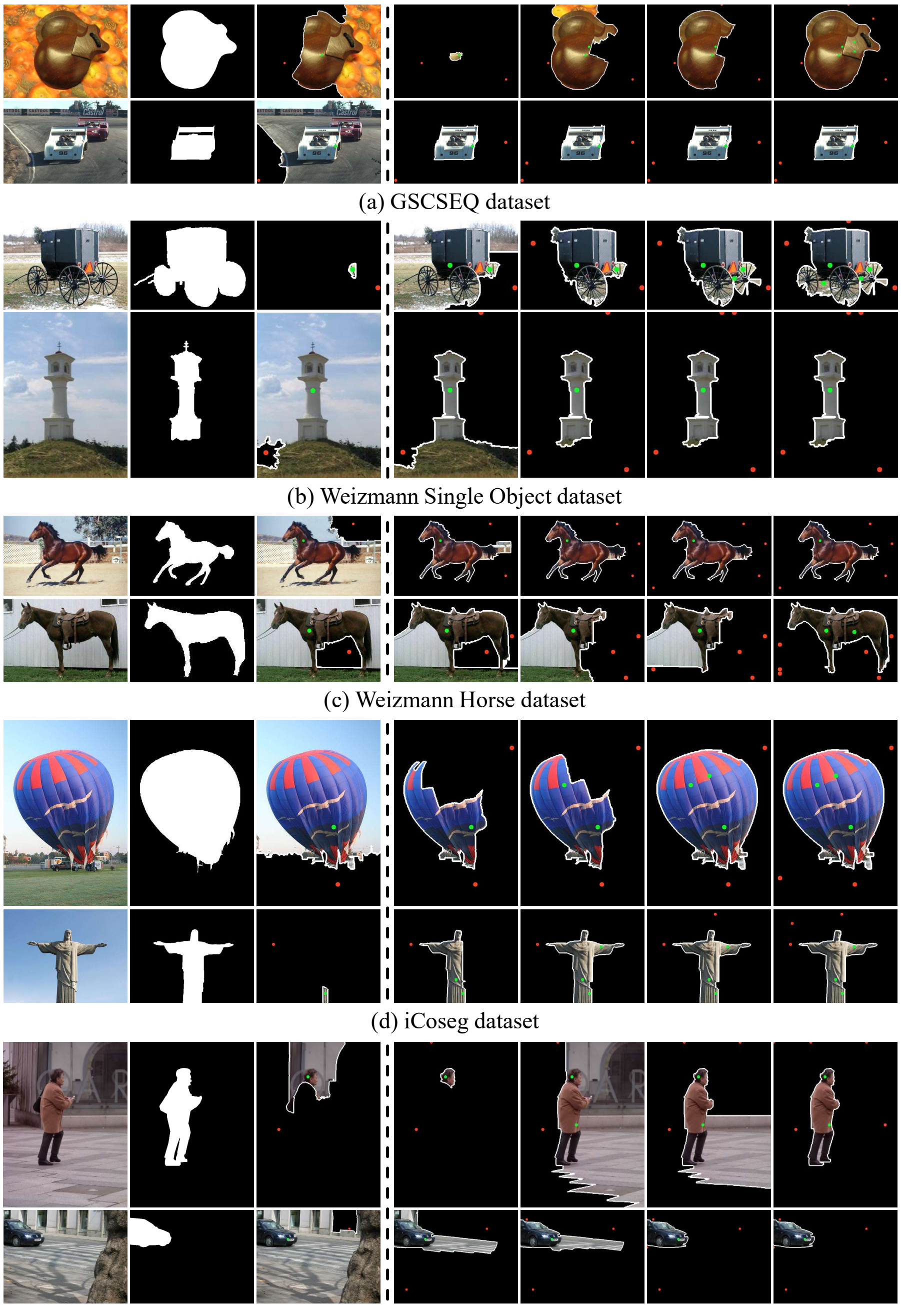


Figure 3.9: MSRA10K results. The left part (first to third columns) contains the input image, GT mask, and initial seed with corresponding RWR [1] result. The right part is the SeedNet result, showing the first three steps (fourth to sixth columns) and the final result (seventh column).



(a) GSCSEQ dataset

(b) Weizmann Single Object dataset

(c) Weizmann Horse dataset

(d) iCoseg dataset

(e) IG02 dataset

Figure 3.10: Unseen dataset results. The left part (first to third columns) contains the input image, GT mask, and initial seed with corresponding RWR [1] result. The right part is the SeedNet result, showing the first three steps (fourth to sixth columns) and the final result (seventh column).

Chapter 4

Interactive Segmentation with Seed Attention

4.1 Introduction

In the previous chapters, we dealt with the spatial expansion of seeds. However, when a real user provides a seed, the user acts based on the semantic relationship of the object in the image. Therefore, to reflect the user's intention, semantic information of the image must be considered. With the recent development of deep learning, it is possible to analyze the semantic information of an image more accurately. We reinforce seed information by using semantic information obtained through a deep network. By adding semantic information to the seed, the seed has richer information. Also, in the previous chapters, we dealt with the seed enrichment step and segmentation module separately. In this chapter, we propose a system that handles the segmentation process and the seed enrichment process simultaneously. The model we propose performs seed enhancement and seed attention through mutual exchange between seed branch and segmentation branch. Furthermore, seed attention solves the problem of forgetting seed information through an emphasis on the seed area.

Recent deep learning-based interactive segmentation algorithms [12, 15, 14, 17] show remarkable performance improvement. Most of these algorithms operate based on seed points given in click form. In most cases, the seed point is converted into a map form and used as an additional channel of the input image. However, in this case, the seed information may be weakened while passing through the deep layer, which leads to inaccurate segmentation results. Recently, BRS algorithms [65, 66] have noted the problem of forgetting seed information. BRS [65] applied a backpropagating refinement scheme to solve the difference between the seed label information and the corresponding point label of the predicted mask. They gave perturbation to the input seed map so that the object mask result matches the input seed information. However, BRS takes a long time because it has to repeat the network several times until the condition is satisfied. To solve this, fBRS [66] significantly reduced the time by giving perturbation at the feature level without giving perturbation to the input. However, they have the problem of repeating forward and backward steps of the network several times. In contrast, we propose a method of deriving an accurate segmentation mask by repeatedly using seed information in a single forward step.

Seed indicates label information of the object located at the point. That is, the seed contains spatial information about a part of GT. Meanwhile, since the segmentation network is composed of a fully convolutional structure, spatial information is preserved through the network. Therefore, the label information of the input seed should be delivered to the label information of the final output mask without losing its spatial information. In order to preserve the seed information and transmit it to the output, the seed information must be emphasized not only at the input but also at an intermediate stage of the network. Maintaining the seed information in the network helps to create a better segmentation mask by strengthening the semantic information of objects around the seed.

We used the attention module to utilize the seed information. Recently, attention mechanisms, including self-attention, have been successfully introduced in the NLP field and are widely used in various fields. It also has been applied to the computer vision field. The attention mechanism works to strengthen the context information of the feature by making the network focus on the critical part. In the case of interactive segmentation, a seed can provide information on where the critical part is. In this chapter, we proposed a bi-directional attention module and newly applied it to the interactive segmentation problem. We call our module Bi-directional Seed Attention (BSA). Through bi-directional attention, the feature map of the network pays attention to the seed map and accepts its spatial information. At the same time, the seed map focuses on the semantic information of the feature and expands to contain more relevant information. In other words, the feature map with strong semantic information and the seed map with strong spatial information exchange information with each other to improve the segmentation results.

We compared the results with existing interactive segmentation models. We trained using SBD dataset [67], which is widely used in segmentation tasks, and compared the number of clicks required to reach a target accuracy. In addition to the SBD dataset, we experiment on the GrabCut [3], Berkeley [68], and DAVIS [69] datasets, and the proposed algorithm recorded state-of-the-art results in comparison with various algorithms. Also, we verified the superiority of the proposed module through comparison with the existing attention mechanism.

4.2 Related Works

Attention Mechanism: Recently, the attention model has been applied to the vision field, resulting in high-performance improvement [70, 71, 72, 73, 74]. In particular, it was successfully applied to semantic segmentation, similar to the problem we are dealing with. Chen *et al.* [75] applied attention models to features obtained from networks using different scale inputs. DANet [76] utilized a dual attention module composed of a position attention module and a channel attention module. CCNet [77] used the Criss-Cross attention module to collect context information more efficiently. AUNet [78] solved the problem of panoptic segmentation. These algorithms have in common that they apply the attention module to the feature map obtained through the backbone network. On the other hand, like BAM [79] and CBAM [80], we applied the attention module in the encoder network.

Li *et al.* [81] have introduced a dual branch, as in our model. They applied the attention model to the video salient object detection problem. They used an optical flow map to give attention information to the leading network. Unlike the [81], where additional parameters are significantly increased by using a separate network, we constructed the network efficiently using the newly proposed bi-directional attention module. Bi-directional attention strengthens the information of each element by paying attention to each other like [82, 83]. We improved performance by applying the attention model to the interactive segmentation problem for the first time.

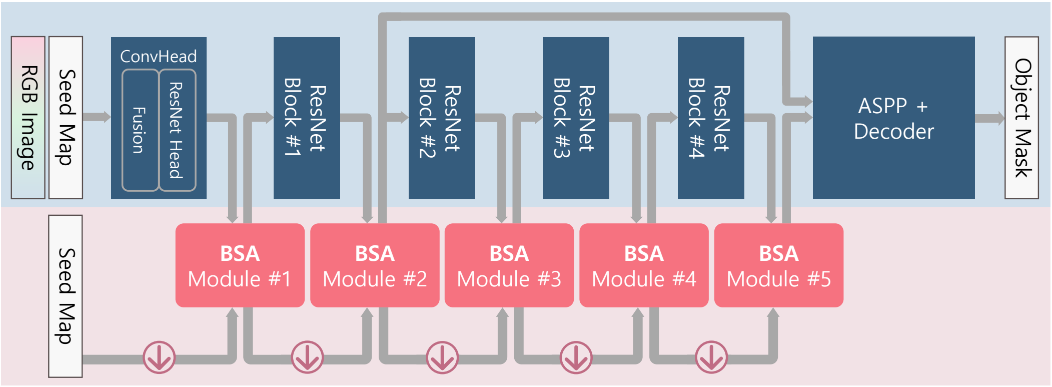


Figure 4.1: Our interactive segmentation network architecture. The blue shaded part is the baseline network responsible for segmentation, and the red shaded part is the seed branch containing the newly proposed attention module. The downward purple arrow indicates the downscale operation.

4.3 Proposed Method

In this chapter, we propose a segmentation network that improves performance by expanding and highlighting the seed given from the user. Our system receives information from the user in the form of a point click and outputs a binary mask. We added the bi-directional seed attention module to the existing interactive segmentation network to enhance the spatial and semantic information of the feature based on the given point seed information.

4.3.1 Interactive Segmentation Network

Figure 4.1 shows the overall structure of our segmentation network. It consists of two parts; the backbone segmentation module and the BSA (Bi-directional Seed Attention) module. As in [12], the input of the network is the RGB image and seed map, and the output becomes the corresponding binary segmentation mask. Any segmentation network can be used as the backbone module. In our work, we used the structure in [66] that is based on DeeplabV3+ [84] as our backbone since it shows excellent performance. It is composed of the encoder, Atrous Spatial Pyramid Pooling (ASPP) module, and decoder. We used a structure that transmits low level features to decoder through skip connection to preserve local information, as in DeeplabV3+. The backbone encoder network is pretrained, and the seed map is obtained through distance transform as commonly done in other works.

ConvHead block located at the beginning of the network consists of two parts. One is the distance maps fusion module used in [66] to combine the RGB image and the seed map, and the other is the head module of ResNet [85]. The feature map created from the ConvHead block goes through the ResBlocks and Decoder and produces the final result. However, as the seed information of the input passes deep through layers, it becomes difficult to maintain the seed information stable. Therefore, we employ attention modules to bring the seed information stably to the end of the network. As in Figure 4.1, the output feature map of each ResBlock is not fed into the next block directly but after being updated through the BSA module. In the BSA module, the feature map is strengthened by the seed information to emphasize the semantic information of objects around the seed.

Meanwhile, the seed map also undergoes an update process. After going through the downscale process to fit the size with the feature map, it is updated based on the semantic information of the feature map. At this time, seed information is a kind of auxiliary variable. Therefore, instead of applying a separate loss function to the seed map, the seed map is converted to have the appropriate information to update the feature map. As shown in Figure 4.2, the feature map of the baseline (backbone) network does not catch semantic information around the seed due to weakened information. On the other hand, the proposed BSA network preserves the seed information well enough through the layers, and in turn, makes the feature maps

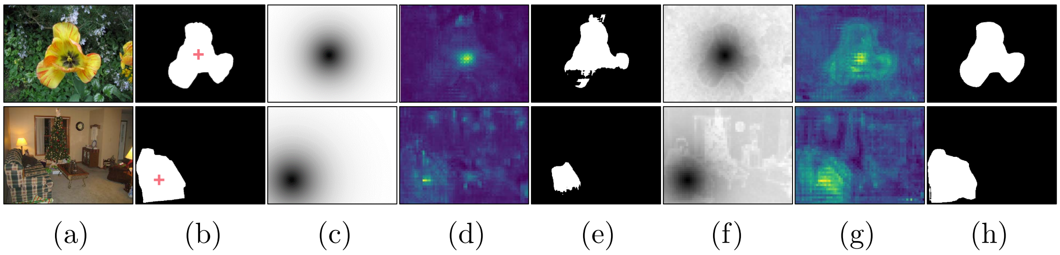


Figure 4.2: Segmentation examples. (a) RGB input image (b) GT object mask and seed location (c) foreground seed distance map (d) feature map of baseline (e) segmentation mask of the baseline (backbone) network (f) updated foreground seed map (g) feature map of our BSA network (h) segmentation mask of our BSA network

emphasize the semantic information about objects around the seed. The feature map shown in the figure is the output of the encoder and shows the average value of the channel dimension. The updated seed map shows the seed used in the last BSA module.

4.3.2 Bi-directional Seed Attention Module

The structure of the proposed BSA module is shown in Figure 4.3. Since it is a bi-directional structure, it has two inputs and two outputs. The feature map $F_{in} \in \mathbb{R}^{C \times H \times W}$ obtained from the ResBlock of the previous layer and the seed map $S_{in} \in \mathbb{R}^{2 \times H \times W}$ of the seed branch are the input of our module, and the feature map $F_{out} \in \mathbb{R}^{C \times H \times W}$ to be used as input to the next layer and the updated seed map $S_{out} \in \mathbb{R}^{2 \times H \times W}$ are output. The two channels of S_{in} and S_{out} represent foreground and background, respectively. We can mainly divide the operation of the module into three parts. Those are the part that updates the feature map based on the seed information, the part that updates the seed from the feature information, and the part that performs self-attention on the feature map.

In the bi-directional attention module, the feature map is updated first. For this,

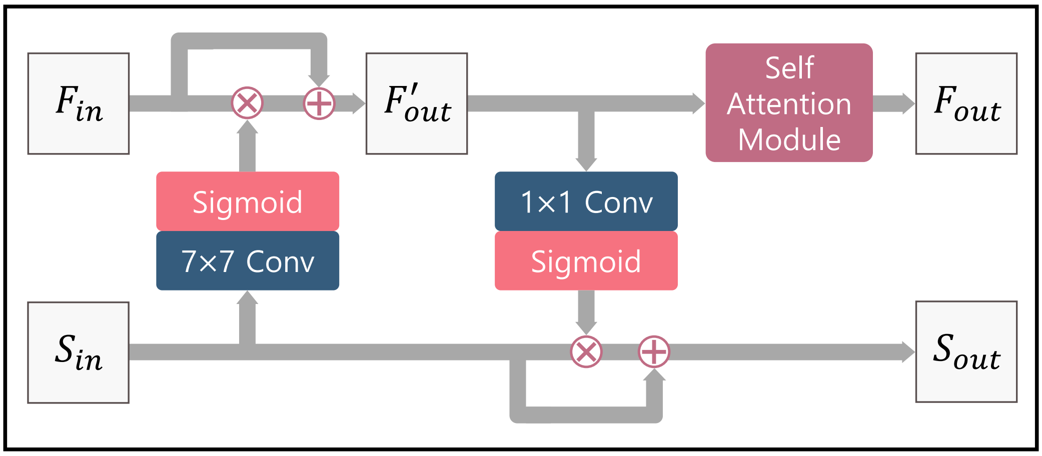


Figure 4.3: Our BSA module. Both the multiplication and addition marks are element-wise operations. $F^{'}_{out}$ is used for module #1 and #5, and F_{out} is used for the remaining modules.

the seed map is converted into an attention map $A_S \in \mathbb{R}^{1 \times H \times W}$ through a convolution operation, and then represented in the form of probability by a Sigmoid transformation. The attention map exhibits information about where to pay attention based on the input seed information. The attention map is then applied to the feature map through element-wise multiplication for each channel. Finally, we complete the feature map update through the residual operation. The seed update process goes through a similar fashion, as shown in Figure 4.3. Both feature and seed map update processes can be described as follows.

$$F^{'}_{out} = F_{in} + F_{in} \otimes \sigma(h^{7\times7}(S_{in})), \tag{4.1}$$

$$S_{out} = S_{in} + S_{in} \otimes \sigma(h^{1\times1}(F^{'}_{out})), \tag{4.2}$$

where σ is the Sigmoid function and $\otimes$ means element-wise multiplication. The convolution operation is represented by $h^{k \times k}$, which has a kernel size of $k \times k$, and if necessary, preserves the size of the input through padding. The output channel size of $h^{k \times k}$ is 1, which serves as a channel reduction. For the feature update, we use a larger size kernel since the spatial information is crucial for the feature. In the case of the feature map, channel reduction is sufficient because the information to be transmitted to the seed is semantic information. However, in the case of a seed map, it is necessary to have a receptive field so that spatial information of the foreground seed and the background seed can be synthesized and converted into information suitable for the update.

Unlike the seed update, which directly outputs S_{out}, feature update takes one more step. We further concentrate the semantic information by using the self-attention module that uses its feature information rather than seed information. At this time, any module can be used for self-attention, and we employed BAM [79]. We obtain the final $F_{out} \in \mathbb{R}^{C \times H \times W}$ through BAM composed of spatial attention and channel attention. However, we did not adjust self-attention for all BSA modules, but only for modules #2, #3, and #4. It is according to the configuration in the original implementation of BAM. In module #1 and #5, $F^{'}_{out}$ was used instead of F_{out}. The following is a summary of our modules.

$$F_{out} = BAM(F_{in} + F_{in} \otimes \sigma(h^{7 \times 7}(S_{in}))), \tag{4.3}$$

$$S_{out} = S_{in} + S_{in} \otimes \sigma(h^{1 \times 1}(F_{in} + F_{in} \otimes \sigma(h^{7 \times 7}(S_{in})))). \tag{4.4}$$

4.4 Experiments

4.4.1 Datasets

For evaluation, we used four standard segmentation benchmark datasets: SBD [67], GrabCut [3], Berkeley [68], and DAVIS [69] datasets. We experimented with the validation set of the SBD dataset. It consists of a total of $2,820$ images with a total number of $6,671$ instance object masks. The GrabCut dataset consists of a total of 50 images, and each image has segmentation information for one object. The test image of the Berkeley dataset is 96 and has 100 object masks, which means some images contain information about multiple segments. Finally, the DAVIS dataset is a dataset for video object segmentation and consists of 50 types of video. We sampled 10% of the frames as in [65] and used them for evaluation.

4.4.2 Metrics

The most widely used metric for segmentation is the intersection over union (IoU). In addition to IoU, we additionally evaluate the performance of interactive segmentation algorithms using the Number of Clicks (NoC) like in [12, 66]. NoC is a method of counting the number of inputs required to reach the target IoU when a robot user [4] is applied to an interactive segmentation system. As a method to simulate an actual annotation environment, it is a metric suitable for evaluating performance that minimizes user effort. Also, we used the value of the area under curve (AuC) in the Click-IoU graph as another metric. For the AuC value, we used a normalized value for easy identification.

4.4.3 Implement Details

We used ResNet-34 model and ResNet-101 model as our backbone (baseline) networks for the experiment. ResNet-101 model was used for performance comparison, while ResNet-34 was used for the ablation study. We trained the networks using the SBD dataset [67] consisting of a total of $8,498$ images. All images were cropped to be the same size of 320×480. For augmentation, we adopted random flipping and resizing. We used binary cross entropy loss for network optimization. During the first 100 epoch, a learning rate of 5×10^{-4} was used, and during the remaining epoch, a learning rate was 10 times lower. In the case of the backbone network, a learning rate of 5×10^{-5}, which is 10 times lower, was applied. We trained our BSA model for a total of 150 epochs with a batch size of 16.

In the case of the SBD dataset, since there is no seed data, seed information is generated through sampling from the GT mask. We sampled training seed data through 3 strategies as in [12]. Furthermore, we adopted the refinement and Zoom-In skill of f-BRS-B [66] in our inference step to improve performance. The f-BRS-B algorithm, like us, has the purpose of preserving seed information, but unlike us, it is applied to the decoder stage and is also a refinement technique so that it can be orthogonal to our algorithm.

4.4.4 Performance

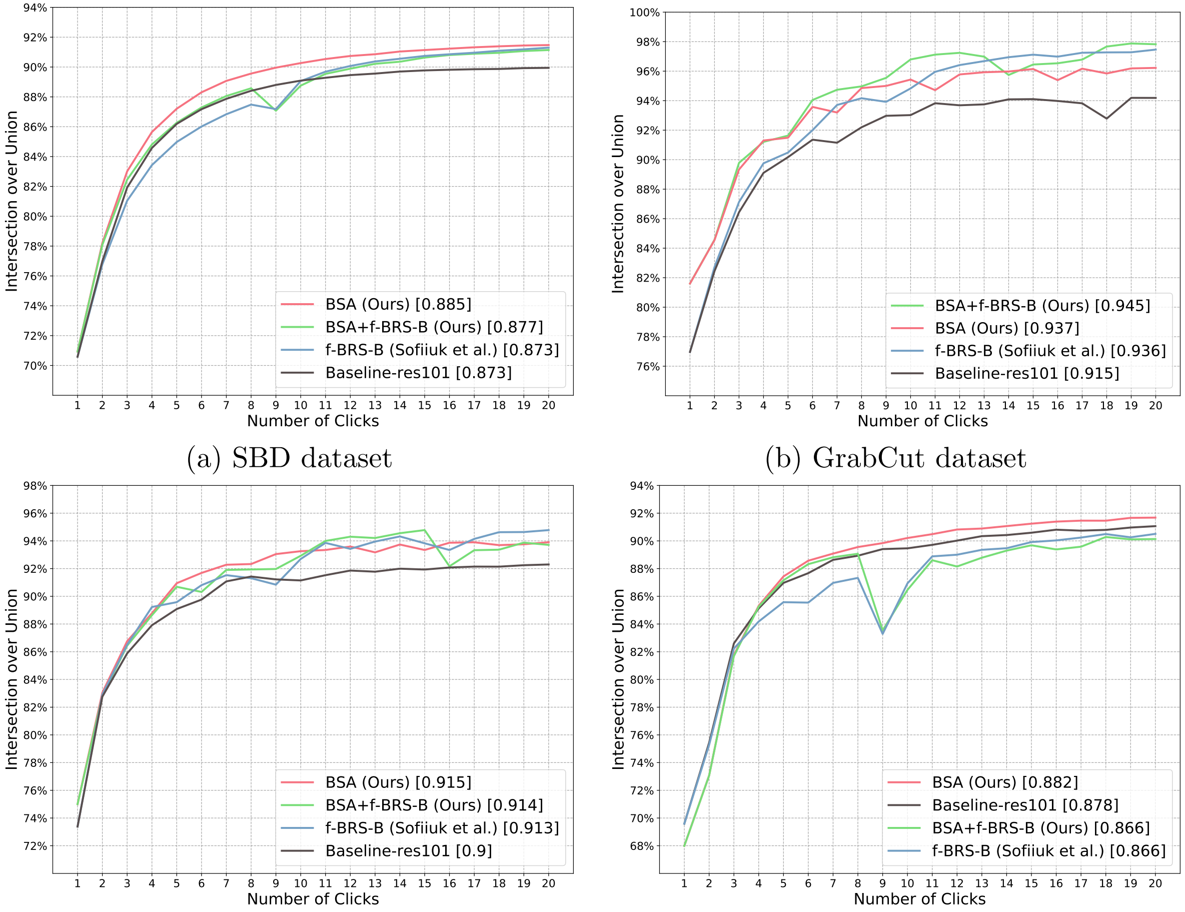
We first verified the effectiveness of the proposed method by comparing the proposed algorithm with the baseline. The results of comparing IoU and NoC of each technique are shown in Table 4.1. In the case of IoU, it is the IoU value for the result of using only the seed given initially.

Table 4.1: Comparison with baseline. In the case of IoU, a higher value indicates a better result, and in the case of NoC, a lower value indicates a better result. Better values are shown in bold.

	SBD	GrabCut	Berkeley	DAVIS
IoU (%)				
Baseline	70.59	76.97	73.38	**69.59**
BSA	**70.93**	**81.60**	**75.00**	67.99
NoC (@90)				
Baseline	8.31	3.86	5.27	7.59
BSA	**7.30**	**2.42**	**4.23**	**7.06**

For the IoU results, the BSA recorded poor results in the DAVIS dataset, but the proposed BSA algorithm showed better results in other datasets, including SBD, the primary dataset. And at NoC, which evaluates the effort of additional user input, the BSA recorded better results in all datasets. *@90* represents the number of user inputs required to reach 90 percent of the IoU. A good result in the NoC metric can be interpreted as a quick correction of the erroneous region. In other words, because the proposed BSA emphasizes semantic information, it converges to an accurate result faster.

Figure 4.4 shows a graph of changes in IoU according to click. We compared the baseline with the proposed BSA, and compared with the f-BRS-B [66], which added a refinement step to the baseline. We also showed that the performance of the algorithm can be improved by adding the same f-BRS-B refinement step to the proposed BSA. In all datasets, including SBD, our BSA or BSA+f-BRS-B algorithm records the best AuC. In particular, f-BRS-B has low performance in DAVIS dataset, which has many difficult objects to segment.



(a) SBD dataset

(b) GrabCut dataset

(c) Berkeley dataset

(d) DAVIS dataset

Figure 4.4: Click-IoU curve graph. The horizontal axis represents the number of clicks by the robot user, and the vertical axis represents the IoU value. The number in the legend indicates the AuC score.

We showed the results of segmentation experiments for each dataset. We compared ResNet-101 based baseline with our BSA results, and f-BRS-B refinement was not applied. In the case of the GrabCut and Berkeley datasets (Fig 4.7), we compared the results of the first click. In the case of SBD and DAVIS (Fig 4.8), the results of the robot user click were compared. At this time, we generated the robot user click based on the result of the baseline. Therefore, in the case of BSA, we applied the seeds generated by the baseline. In the figure, the red cross represents the foreground seed, and the blue cross represents the background seed. The feature map shows the channel-wise mean of the output feature of the encoder.

Additionally, we have compared our algorithm with existing state-of-the-art methods including the classic approaches like GC [23], RW [2], GSC [4], and recent deep learning based techniques such as DIOS [12], RIS-Net [14], LD [15], ITIS [86], FCTSFN [16], CMG [17], BRS [65], DIOS [18], and f-BRS-B [66]. They can also be divided into three categories: Methods without training, methods trained with the PASCAL VOC[87] dataset, and methods trained with the SBD dataset. Therefore, it is not a completely fair comparison, but we can compare using a common dataset. Table 4.2 shows the experimental results. The proposed algorithm, BSA, recorded the least number of clicks in most results. For GrabCut datasets, DIOS [18] gives the best results, but BSA shows the best among algorithms that are trained on the SBD dataset. Meanwhile, the baseline (backbone) network we used is the same as the baseline of f-BRS-B [66]. Even when we apply only the BSA module to the baseline, it shows better performance than f-BRS-B. Furthermore, when f-BRS-B refinement step is additionally applied, the performance gain becomes much larger.

Table 4.2: Comparison with other interactive segmentation methods (NoC 85% and NoC 90%). The best results are shown in bold, and the second-best results are underlined

Method	Train Set	SBD		GrabCut		Berkeley	DAVIS	
		@85	*@90*	*@85*	*@90*	*@90*	*@85*	*@90*
GC [23]	-	13.60	15.96	7.98	10.00	14.22	15.13	17.41
RW [2]	-	12.22	15.04	11.36	13.77	14.02	16.71	18.31
GSC [4]	-	12.69	15.31	7.10	9.12	12.57	15.35	17.52
DIOS [12]	*SBD*	9.22	12.80	5.08	6.08	-	9.03	12.58
RISNet [14]	*VOC*	-	-	-	5.00	6.03	-	-
LD [15]	*SBD*	7.41	10.78	3.20	4.79	-	5.05	9.57
ITIS [86]	*VOC*	-	-	-	5.60	-	-	-
FCTSFN [16]	*VOC*	-	-	-	3.76	6.49	-	-
CMG [17]	*VOC*	-	-	-	3.58	5.60	-	-
BRS [65]	*SBD*	6.59	9.78	2.60	3.60	5.08	5.58	8.24
DIOS [18]	*VOC*	-	-	-	**1.96**	4.31	-	-
f-BRS-B [66]	*SBD*	4.81	7.73	2.30	2.72	4.57	5.04	7.41
Baseline-*res*101	*SBD*	5.25	8.31	3.12	3.86	5.27	5.15	7.59
BSA	*SBD*	4.63	7.44	2.22	2.58	4.97	**5.01**	7.17
***BSA* + f-BRS-B**	*SBD*	**4.44**	**7.30**	**2.00**	2.42	**4.23**	5.20	**7.06**

4.4.5 Ablation Study

4.4.5.1 Attention Modules

We conducted an ablation study to analyze the proposed BSA module further. We compared various types of attention modules by applying them to the baseline (backbone) network. Also, we modified our BSA to analyze the effect of each component. The results are shown in Table 4.3. All the results are f-BRS-B refinement applied. BAM [79] is a self-attention module that is used in our BSA module. Even with BAM alone, the performance is improved compared to the baseline, but the improvement is not significant. MGA-tmc [81] is a uni-directional attention module, and they apply attention to both the decoder and encoder. Additional information is also updated through a separate ResNet branch. In our test, instead of the optical flow originally used as additional information, we used a seed map. MGA-tmc also has a performance improvement but has the disadvantage of requiring many training parameters due to separate ResNet. The attention modules of SAGAN [74] and DANet [76] have similar structures. We used these two modules by attaching them to the next part of the backbone, as in DANet. However, unlike DANet, our decoder structure is applied to compare it fairly with other attention modules. SAGAN and DANet performed well on the GrabCut dataset, but not on SBD.

Next, we have examined the contribution of each component of our BSA module. Our proposed BSA is a bi-directional module. We tried to change this to *uni-directional* format without seed update and output only F'_{out} of (4.1). That is, a network is a similar form to MGA-tmc, but the seed information is not updated, so it records slightly lower performance than MGA-tmc. However, by changing this to our bi-directional format and applying a *seed update* process in (4.4), we get better

Table 4.3: Ablation study on attention type. It shows the number of parameters in the network and the NoC values (85% and 90%) for the SBD and GrabCut datasets. The best results are shown in bold, and the second-best results are underlined.

Method	$\#Params$	SBD		GrabCut	
		@85	*@90*	*@85*	*@90*
Baseline-$res34$	23.34M	5.17	8.33	2.48	3.00
BAM [79]	23.36M	5.11	8.20	2.40	2.82
MGA-tmc [81]	47.00M	4.90	7.91	**1.98**	2.62
SAGAN [74]	23.67M	5.17	8.30	2.12	2.82
DANet [76]	33.11M	5.36	8.61	2.16	2.64
BSA-*res*34					
Uni-directional	23.34M	4.93	7.97	2.22	2.94
+ Seed Update	23.34M	4.87	7.87	2.06	2.60
+ Self Attention	23.37M	**4.80**	**7.74**	2.06	**2.56**

results than MGA-tmc, which updates information via ResNet. Finally, applying *self-attention* in (4.3) increases the parameter usage slightly, but gives improved results. Overall, we can see that the proposed BSA module boosts the performance significantly while requiring fewer parameters than other attention modules. More ablation experiment results and examples are shown in the supplementary material.

4.4.5.2 Model Configurations

We experimented with changing the elements of the proposed module. Based on our BSA module, we only changed one element for each experiment. Tested on four main subjects, and the experimental results are shown in the Table 4.4. First, we changed the location of the self-attention module. In the BSA module, we place the self-attention module after bi-directional attention. Instead, self-attention was placed *before* or *parallel* to bi-directional attention. The following configuration is for the sequence of bi-directional attention. Currently, BSA performs feature update

Table 4.4: Ablation study on model configuration. NoC values (85% and 90%) for the SBD and GrabCut. The best results are shown in bold, and the second-best results are underlined.

Method	SBD		GrabCut	
	@85	*@90*	*@85*	*@90*
Self Attention Location				
$before$	5.00	8.05	2.10	2.84
$parallel$	4.91	7.87	**1.96**	**2.38**
Attention Order				
$reverse$	4.97	7.93	2.14	2.72
$parallel$	5.12	8.30	2.08	2.66
Module Usage				
all self-att.	4.90	7.93	2.24	2.82
w.o. #1, #5	4.97	8.01	2.08	2.60
Seed type				
$Gaussian$	5.01	8.05	2.30	2.92
$convolution$	<u>4.88</u>	<u>7.81</u>	<u>1.98</u>	2.58
BSA-*res*34	**4.80**	**7.74**	2.06	<u>2.56</u>

first. By *reversing* this, we tried to perform the seed update first. Also, we conducted two attentions independently in *parallel.*

In the BSA system, we use two types of modules depending on the presence or absence of self-attention. Self-attention applies only to modules #2, #3, and #4. We experimented with applying self-attention to *all module.* Besides, if it is effective not to use self-attention in modules #1 and #5, remaining bi-directional attention was also removed. That is, we tested the configuration *without modules* #1 and #5 at all. Finally, we changed the type of seed used for the seed branch. In our seed map, we apply the euclidean distance transform, so the pixel at the seed position has the lowest value. Intuitively, it seems to be easy to apply attention and update when the seed area is highlighted at the seed map. Therefore, we use a *Gaussian*-transformed seed map that emphasizes the seed region. Also, in order to transform the input seed suitable for the seed branch, we apply an additional three-layer *convolution* filter to the input seed map.

Although some module configurations show a partial advantage, BSA showed the best results overall. In the case of the seed type test, contrary to our expectations, the seed in the form of a basic distance map showed the best results.

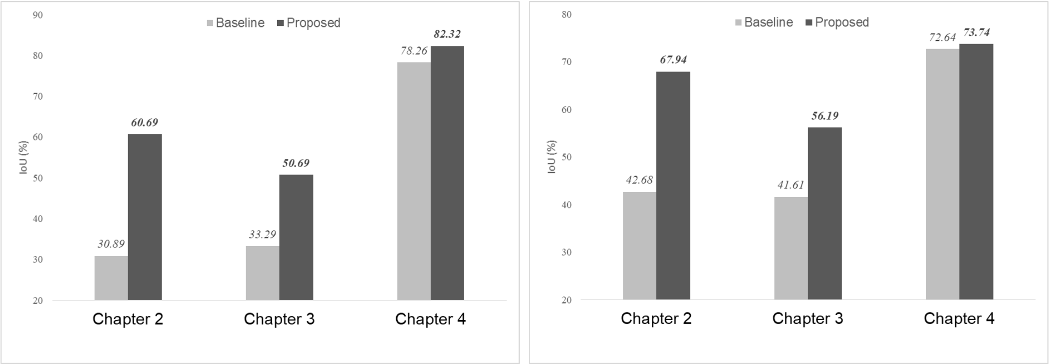
4.4.6 Seed enrichment methods

We performed comparative experiments on the proposed algorithms in this dissertation. The algorithms proposed in each chapter showed superior performance compared to the baseline techniques. However, the comparison results between the proposed algorithms are also significant. We compared the IoU and NoC of each algorithm using a common dataset. Also, the practical usability of the algorithm was verified by comparing the operation speed.

4.4.6.1 IoU comparison

We compared each algorithm by calculating the IoU score for the initially given seed. To assume the minimum user seed case, which is the purpose of this dissertation, one foreground seed point and one background seed point were provided as initial seed. The initial seed is located in the center of each area according to the robot user's principle. We compared the IoU values using two datasets: GrabCut dataset [3] and Weizmann single object dataset [45]. The results are shown in Figure 4.5.

As shown earlier, the proposed algorithms recorded higher results than baseline in all chapters. Compared to Chapters 2 and 3 using the MRF-based segmentation module, it can be seen that the CNN-structured Chapter 4 algorithm shows the



(a) (b)

Figure 4.5: IoU comparison for (a) GrabCut dataset (b) Weizmann dataset. In each chapter, the performance of the baseline (light gray bar) and the proposed algorithm (dark gray bar) was compared.

best results. Also, in the case of Chapter 2, it shows remarkable performance gain despite its simple configuration. In the case of Chapter 3, it has strengths compared to Chapter 2 for divided objects. However, the effects did not appear well in the experiment.

4.4.6.2 NoC comparison

Another factor that puts a burden on the user in interactive segmentation is the additional input. If the result is not satisfactory, the user has to input additional inputs, but the more this number is, the greater the burden. We applied the robot user to evaluate the degree of additional input burden. Applying the robot user, the number of clicks (NoC) required until reaching a specific IoU was calculated. We used the GrabCut dataset, and the result is shown in the Table 4.5.

We evaluated the performance by adding a seed while one foreground seed and one background seed were provided as initial seed. In the case of NoC_{10}@80, it represents the number of clicks required to reach 80% of IoU within a maximum of

Table 4.5: NoC comparison for GrabCut dataset. NoC values for 85% and 90%. The lower the better.

	Chapter 2		Chapter 3		Chapter 4	
	Baseline	*Proposed*	*Baseline*	*Proposed*	*Baseline*	*Proposed*
NoC_{10}@80	7.82	4.86	7.02	7.00	3.12	2.78
NoC_{20}@85	12.39	7.27	9.76	10.65	3.33	2.86

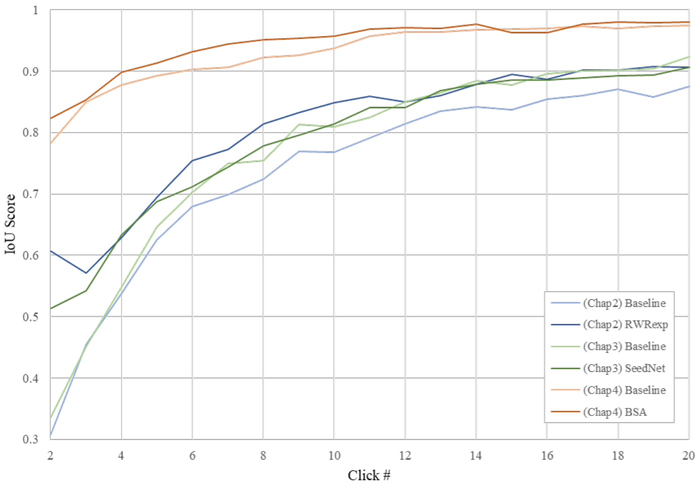


Figure 4.6: Click-IoU curve for GrabCut dataset. The horizontal axis represents the number of robot user inputs, and the vertical axis represents the IoU score.

10 clicks. Likewise, NoC_{20}@85 has 20 maximum inputs and a target of 85%. In most cases, the proposed algorithm shows better results than the baseline. In other words, it reduces the burden on the user by requiring fewer additional inputs. However, in the case of Chapter 3, the proposed algorithm shows poor results. We can check this in more detail by looking at Figure 4.6. In the Click-IoU graph, the proposed algorithm of Chapter 3 shows higher performance than baseline at the beginning but gradually shows similar performance. This means that when the SeedNet proposes a new seed point, it does not effectively reflect the corrected seed by a robot user. It needs to be improved through more diverse training cases.

Table 4.6: Running time of each algorithm.

	Chapter 2		Chapter 3		Chapter 4	
	Baseline	*Proposed*	*Baseline*	*Proposed*	*Baseline*	*Proposed*
Params	-	-	-	3,777k	58,431k	58,794k
Time	2.312s	0.661s	2.426s	26.835s	0.264s	0.270s

4.4.6.3 Time comparison

In order to use interactive segmentation in real life, operation time is an essential factor. In order to reduce the operation time, we proposed a pyramid structure in Chapter 2. We experimented with the whole algorithm in the same environment. The operating environment consists of an Intel i7-6700K processor, an NVIDIA Geforce GTX 1080ti graphics card, and 32GB RAM. The running time and number of trainable parameters are shown in the Table 4.6.

The algorithm of Chapter 4, which operates based on CUDA, calculates the result the fastest. However, because it requires a graphics card and many parameters, the algorithm of Chapter 2 can be an alternative in light environments. The proposed algorithm of Chapter 3 has a slow operation time due to proposing seeds several times. However, when analyzing this, the time required by the segmentation module is $26.784s$, and the time to propose a seed is only $0.025s$. Therefore, the operation time can be greatly reduced by using a segmentation module like a CNN-based model.

4.5 Summary

In this chapter, we proposed novel bi-directional seed attention (BSA) network for interactive segmentation. By adding a simple BSA module to the backbone segmentation network, we simultaneously enhanced the seed information and the semantic information around the seed to obtain a better segmentation mask. We demonstrated the superiority of the proposed network over existing state-of-the-art methods on various benchmark datasets. Also, we have justified the validity of the proposed module structure through comparison with other attention modules.

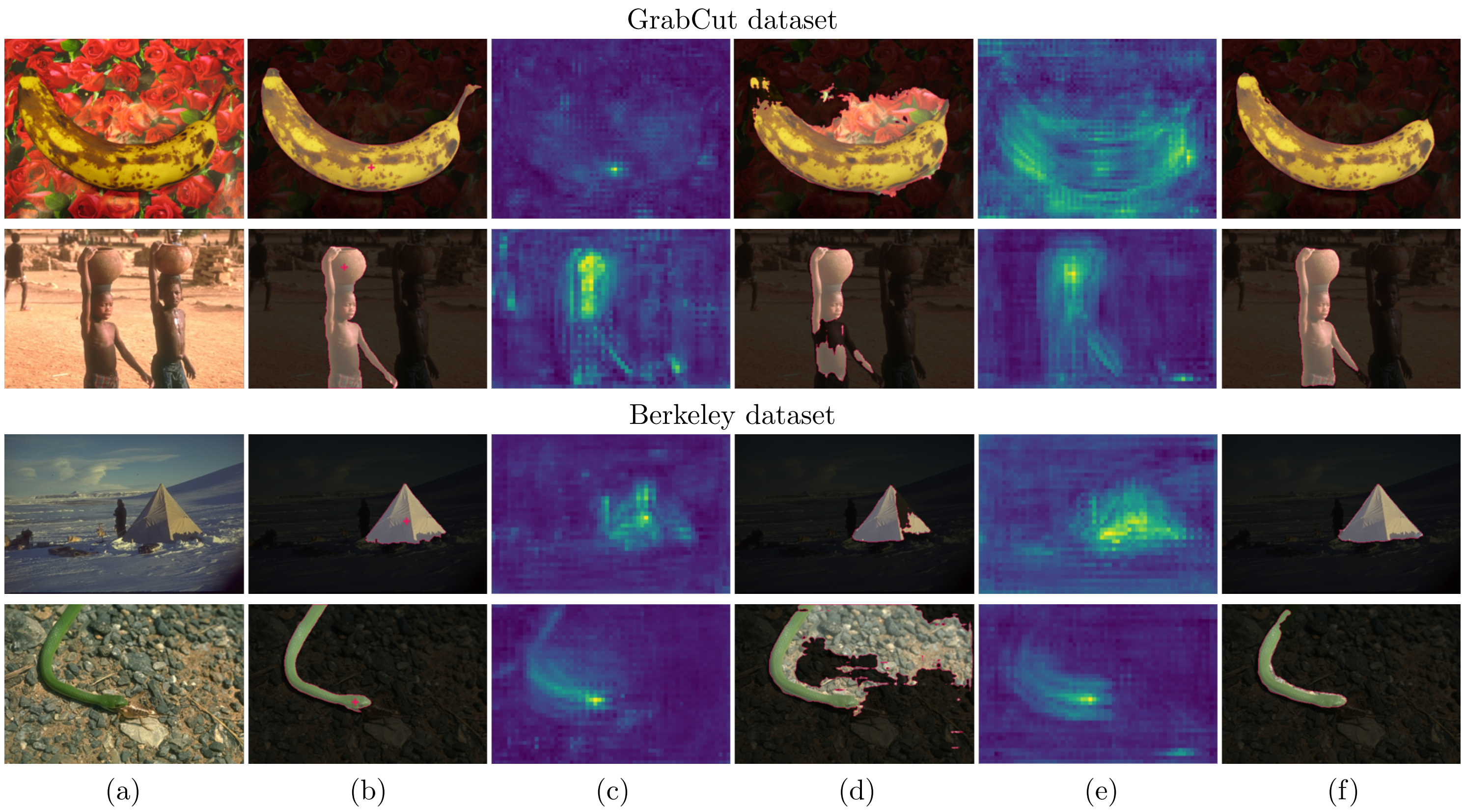


Figure 4.7: GrabCut, Berkeley dataset results. (a) RGB Image (b) GT mask and seed point (c) Feature map of baseline network (d) Segmentation result of baseline (e) Feature map of BSA network (f) Segmentation result of BSA

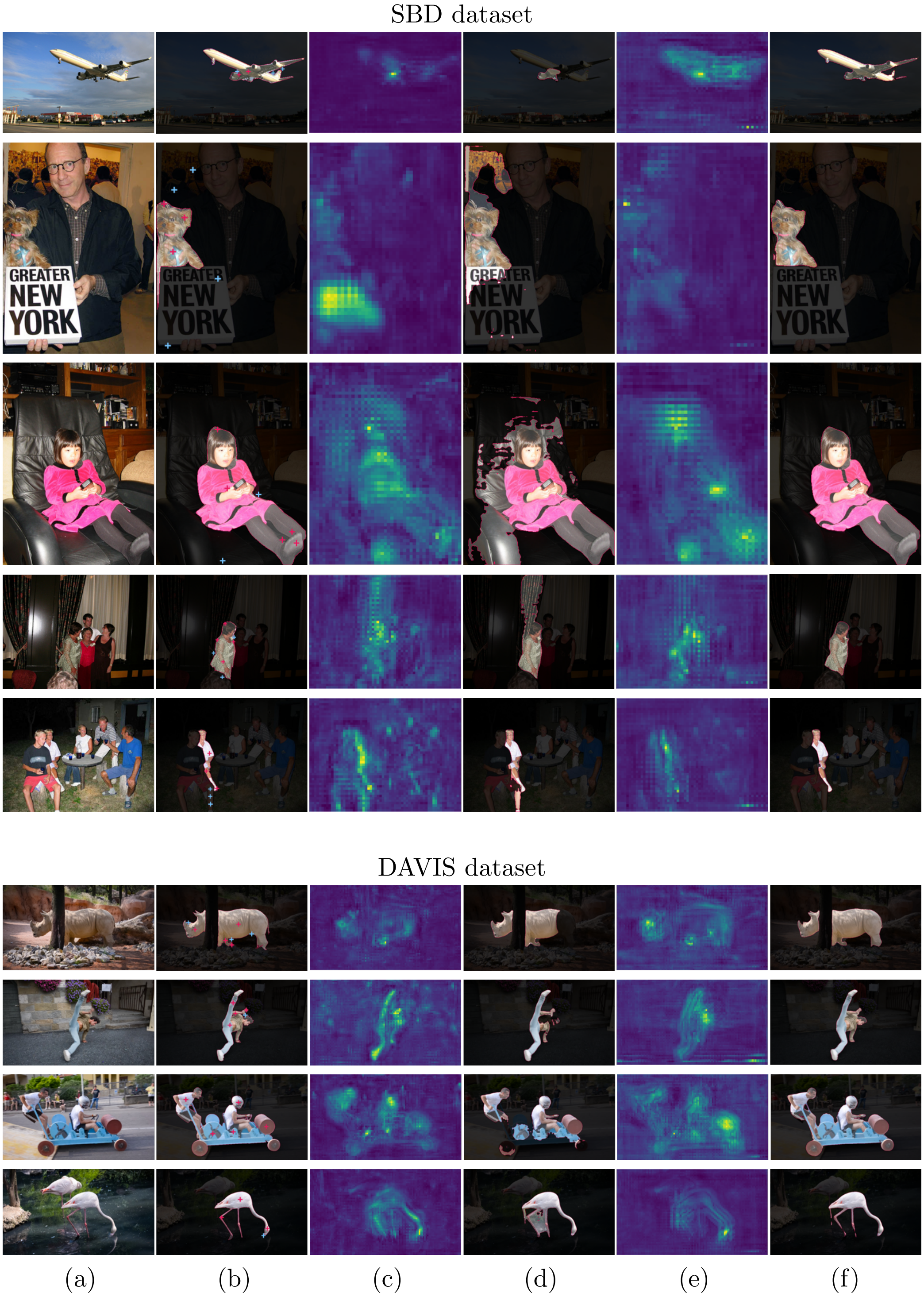


Figure 4.8: SBD, DAVIS dataset results. (a) RGB Image (b) GT mask and seed point (c) Feature map of baseline network (d) Segmentation result of baseline (e) Feature map of BSA network (f) Segmentation result of BSA

Chapter 5

Conclusions

To conclude, the proposed techniques in the current dissertation can effectively solve the interactive segmentation problem particularly when seed information is insufficient. A method was used to enrich the given seed information and transform it into a seed containing rich information. Such seed enrichment was performed spatially or semantically, and the proposed technique was based on seed expansion, seed generation, and seed attention.

Seed Expansion: To solve the insufficient seed information problem, a method to expand the seed area was proposed. The proposed seed expansion step consists of two stages and follows the label propagation format. In the first stage, a seed was expanded from pixel level to superpixel level. In the second step, RWR segmentation was used to extend the seed near to the object which produced a more accurate segmentation mask. The final result was obtained through refinement using global information from the saliency map. A comparison with the existing techniques, our approach was found to solve the sparse seed problem and the unbalanced seed problem more effectively.

Seed Generation: The proposed seed expansion expanded the seed spatially, but its area was limited to the area around the seed. In this case, if the seed distribution is uneven, problem occurs in enriching the seed information. To address this issue, a new algorithm was proposed which would generate a seed at a new point. The process of selecting a new point imitates the user's behavior. If the segmentation result is not satisfactory, the user provides a new seed to the erroneous area. We proposed a system that would generate a sequence of seeds by training this user-like process through reinforcement learning. The trained system recorded better results than the baseline segmentation algorithm.

Seed Attention: The algorithms proposed above dealt with the spatial expansion of a seed. Spatial expansion of a seed is important, but in order to contain more information, the seed was further extended to the semantic domain. For this purpose, we used the feature of the segmentation network that contains semantic information. By adding semantic information to the seed information, it was possible to analyze the user's intention better. Also, at the same time, we tried to improve the performance of segmentation by adding seed information to semantic information. In other words, we proposed a system in which segmentation and seed enrichment processes can interact. Our system was constructed using a novel bi-directional attention module. Through this, we focused on the semantic information of the image and recorded better performance than the existing state-of-the-art techniques.

5.1 Summary

Our proposed techniques successfully enriched the seed information. All proposed algorithms showed better results than the baseline with insufficient seed information. Therefore, we developed an interactive segmentation system which can reduce the burden on users and produce satisfactory results. However, there are also limitations and improvements. First, there is no guarantee that the extended seed will not violate the intention of the user. In the case of seed generation, improvement is necessary by using a segmentation module based on deep learning. Finally, since both spatial expansion and semantic expansion can be applied simultaneously, an integrated model study is needed as future work.

Bibliography

[1] T. H. Kim, K. M. Lee, and S. U. Lee, "Generative image segmentation using random walks with restart," in *European Conference on Computer Vision.* Springer, 2008.

[2] L. Grady, "Random walks for image segmentation," in *Transactions on Pattern Analysis and Machine Intelligence.* IEEE, 2006.

[3] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive foreground extraction using iterated graph cuts," in *Transactions on Graphics.* ACM, 2004.

[4] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, "Geodesic star convexity for interactive image segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2010.

[5] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watersheds: A new image segmentation framework extending graph cuts, random walker and optimal spanning forest," in *International Conference on Computer Vision.* IEEE, 2009.

[6] X. Bai and G. Sapiro, "Geodesic matting: A framework for fast interactive image and video segmentation and matting," in *International Journal of Computer Vision.* Springer, 2009.

[7] O. Veksler, "Star shape prior for graph-cut image segmentation," in *European Conference on Computer Vision.* Springer, 2008.

[8] W. Casaca, L. G. Nonato, and G. Taubin, "Laplacian coordinates for seeded image segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2014.

[9] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu, "Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2014.

[10] J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof, "Interactive texture segmentation using random forests and total variation." in *British Machine Vision Conference.* BMVA, 2009.

[11] Z. Kuang, D. Schnieders, H. Zhou, K.-Y. K. Wong, Y. Yu, and B. Peng, "Learning image-specific parameters for interactive segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2012.

[12] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang, "Deep interactive object selection," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2016.

[13] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2015.

[14] J. Hao Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng, "Regional interactive image segmentation networks," in *International Conference on Computer Vision.* IEEE, 2017.

[15] Z. Li, Q. Chen, and V. Koltun, "Interactive image segmentation with latent diversity," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2018.

[16] Y. Hu, A. Soltoggio, R. Lock, and S. Carter, "A fully convolutional two-stream fusion network for interactive image segmentation," *Neural Networks*, 2019.

[17] S. Majumder and A. Yao, "Content-aware multi-level guidance for interactive instance segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[18] J. H. Liew, S. Cohen, B. Price, L. Mai, S.-H. Ong, and J. Feng, "Multiseg: Semantically meaningful, scale-diverse segmentations from minimal user input," in *International Conference on Computer Vision.* IEEE, 2019.

[19] S. Dutt Jain and K. Grauman, "Predicting sufficient annotation strength for interactive foreground segmentation," in *International Conference on Computer Vision.* IEEE, 2013.

[20] G. Song, H. Myeong, and K. M. Lee, "Interactive segmentation with seed expansion," in *Asia-Pacific Signal and Information Processing Association Annual Summit and Conference.* IEEE, 2014.

[21] ——, "Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2018.

[22] G. Song and K. M. Lee, "Bi-directional seed attention network for interactive image segmentation," *Signal Processing Letters*, 2020.

[23] Y. Y. Boykov and M.-P. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *International Conference on Computer Vision*. IEEE, 2001.

[24] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp, "Image segmentation with a bounding box prior," in *International Conference on Computer Vision*. IEEE, 2009.

[25] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *International Conference on Machine Learning*, 2003.

[26] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," *Advances in Neural Information Processing Systems*, 2003.

[27] R. Adams and L. Bischof, "Seeded region growing," in *Transactions on Pattern Analysis and Machine Intelligence*. IEEE, 1994.

[28] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recognition Letters*, 1997.

[29] V. Vezhnevets and V. Konouchine, "Growcut: Interactive multi-label nd image segmentation by cellular automata," in *Graphicon*, 2005.

[30] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014.

[31] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2014.

[32] J. Zhang and S. Sclaroff, "Saliency detection: A boolean map approach," in *International Conference on Computer Vision.* IEEE, 2013.

[33] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: A discriminative regional feature integration approach," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2013.

[34] R. S. Srivatsa and R. V. Babu, "Salient object detection via objectness measure," in *International Conference on Image Processing.* IEEE, 2015.

[35] M.-Y. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2011.

[36] A. Jain, S. Chatterjee, and R. Vidal, "Coarse-to-fine semantic video segmentation using supervoxel trees," in *International Conference on Computer Vision.* IEEE, 2013.

[37] T. Cour, F. Benezit, and J. Shi, "Spectral segmentation with multiscale graph decomposition," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2005.

[38] G. Sundaramoorthi, A. Yezzi, and A. C. Mennucci, "Coarse-to-fine segmentation and tracking using sobolev active contours," *Pattern Analysis and Machine Intelligence*, 2008.

[39] J. Chang, D. Wei, and J. W. Fisher, “A video representation using temporal superpixels,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2013.

[40] A. Schick, M. Bäuml, and R. Stiefelhagen, “Improving foreground segmentations with probabilistic superpixel markov random fields,” in *Conference on Computer Vision and Pattern Recognition Workshops.* IEEE, 2012.

[41] P. Mehrani and O. Veksler, “Saliency segmentation based on learning and graph cut refinement.” in *British Machine Vision Conference.* BMVA, 2010.

[42] Y. Fu, J. Cheng, Z. Li, and H. Lu, “Saliency cuts: An automatic approach to object segmentation,” in *International Conference on Pattern Recognition.* IEEE, 2008.

[43] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge 2009,” in *2th PASCAL Challenge Workshop*, 2009.

[44] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott, “A perceptually motivated online benchmark for image matting,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2009.

[45] S. Alpert, M. Galun, R. Basri, and A. Brandt, “Image segmentation by probabilistic bottom-up aggregation and cue integration,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2007.

[46] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, “Human-level

control through deep reinforcement learning," in *Nature*. Nature Research, 2015.

[47] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations*, 2016.

[48] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning." in *Conference on Artificial Intelligence*. AAAI, 2016.

[49] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas, "Dueling network architectures for deep reinforcement learning," in *International Conference on Machine Learning*, 2016.

[50] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016.

[51] J. C. Caicedo and S. Lazebnik, "Active object localization with deep reinforcement learning," in *International Conference on Computer Vision*. IEEE, 2015.

[52] M. Bellver, X. Giro-i Nieto, F. Marques, and J. Torres, "Hierarchical object detection with deep reinforcement learning," in *Conference on Neural Information Processing Systems Workshop*, 2016.

[53] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," in *International Conference on Learning Representations*, 2015.

[54] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li, "Deep reinforcement learning-based image captioning with embedding reward," in *Conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[55] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi, “Action-decision networks for visual tracking with deep reinforcement learning,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2017.

[56] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, “Attention-aware face hallucination via deep reinforcement learning,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2017.

[57] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei, “End-to-end learning of action detection from frame glimpses in videos,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2016.

[58] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Learning to compose neural networks for question answering,” in *Conference of the North American Chapter of the Association for Computational Linguistics.* ACL, 2016.

[59] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu, “Global contrast based salient region detection,” in *Transactions on Pattern Analysis and Machine Intelligence.* IEEE, 2015.

[60] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *International Conference on Learning Representations*, 2015.

[61] E. Borenstein and S. Ullman, “Learning to segment,” in *European Conference on Computer Vision.* Springer, 2004.

[62] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, “icoseg: Interactive co-segmentation with intelligent scribble guidance,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2010.

[63] M. Marszalek and C. Schmid, “Accurate object localization with shape masks,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2007.

[64] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer, “Generic object recognition with boosting,” in *Transactions on Pattern Analysis and Machine Intelligence.* IEEE, 2006.

[65] W.-D. Jang and C.-S. Kim, “Interactive image segmentation via backpropagating refinement scheme,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[66] K. Sofiiuk, I. Petrov, O. Barinova, and A. Konushin, “f-brs: Rethinking backpropagating refinement for interactive segmentation,” *Conference on Computer Vision and Pattern Recognition*, 2020.

[67] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors,” in *International Conference on Computer Vision.* IEEE, 2011.

[68] D. Martin, C. Fowlkes, D. Tal, and J. Malik, “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics,” in *International Conference on Computer Vision.* IEEE, 2001.

[69] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2016.

[70] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2017.

[71] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2018.

[72] M. Choi, H. Kim, B. Han, N. Xu, and K. M. Lee, "Channel attention is all you need for video frame interpolation," in *Conference on Artificial Intelligence.* AAAI, 2020.

[73] Z. Zhu, W. Wu, W. Zou, and J. Yan, "End-to-end flow correlation tracking with spatial-temporal attention," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2018.

[74] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *International Conference on Machine Learning*, 2019.

[75] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2016.

[76] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[77] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, "Ccnet: Criss-cross attention for semantic segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[78] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[79] J. Park, S. Woo, J.-Y. Lee, and I. S. Kweon, "Bam: Bottleneck attention module," *British Machine Vision Conference*, 2018.

[80] S. Woo, J. Park, J.-Y. Lee, and I. So Kweon, "Cbam: Convolutional block attention module," in *European Conference on Computer Vision.* Springer, 2018.

[81] H. Li, G. Chen, G. Li, and Y. Yu, "Motion guided attention for video salient object detection," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2019.

[82] Q. Cui, H. Sun, Y. Li, and Y. Kong, "A deep bi-directional attention network for human motion recovery," in *International Joint Conferences on Artificial Intelligence.* AAAI Press, 2019.

[83] M. Seo, A. Kembhavi, A. Farhadi, and H. Hajishirzi, "Bidirectional attention flow for machine comprehension," *International Conference on Learning Representations*, 2017.

[84] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *European Conference on Computer Vision.* Springer, 2018.

[85] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Conference on Computer Vision and Pattern Recognition.* IEEE, 2016.

[86] S. Mahadevan, P. Voigtlaender, and B. Leibe, “Iteratively trained interactive segmentation,” in *British Machine Vision Conference.* BMVA, 2018.

[87] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International Journal of Computer Vision*, 2010.

국문초록

영상에서 원하는 물체 영역을 잘라내는 것은 컴퓨터 비전 문제에서 필수적인 요소이다. 영상을 해석하거나 분석할 때, 대부분의 알고리즘들이 의미론적인 단위 기반으로 동작하기 때문이다. 그러나 영상에서 물체 영역을 분할하는 것은 모호한 문제이다. 사용자와 목적에 따라 원하는 물체 영역이 달라지기 때문이다. 이를 해결하기 위해 사용자와의 교류를 통해 원하는 방향으로 영상 분할을 진행하는 대화형 영상 분할 기법이 사용된다. 여기서 사용자가 제공하는 시드 정보가 중요한 역할을 한다. 사용자의 의도를 담고 있는 시드 정보가 정확할수록 영상 분할의 정확도도 증가하게 된다. 그러나 풍부한 시드 정보를 제공하는 것은 사용자에게 많은 부담을 주게 된다. 그러므로 간단한 시드 정보를 사용하여 만족할만한 분할 결과를 얻는 것이 주요 목적이 된다.

우리는 제공된 희소한 시드 정보를 변환하는 작업에 초점을 두었다. 만약 시드 정보가 풍부하게 변환된다면 정확한 영상 분할 결과를 얻을 수 있기 때문이다. 그러므로 본 학위 논문에서는 시드 정보를 풍부하게 하는 기법들을 제안한다. 최소한의 사용자 입력을 가정하고 이를 다양한 시드 확장 기법을 통해 변환한다. 우리는 시드 확대, 시드 생성, 시드 주의 집중에 기반한 총 세 가지의 대화형 영상 분할 기법을 제안한다. 각각 시드 주변으로의 영역 확대, 새로운 지점에 시드 생성, 의미론적 정보에 주목하는 형태의 시드 확장 기법을 사용한다.

먼저 시드 확대에 기반한 기법에서 우리는 시드의 영역 확장을 목표로 한다. 두 단계로 구성된 확대 과정을 통해 처음 시드 주변의 비슷한 픽셀들을 시드 영역으로 편입한다. 이렇게 확장된 시드를 사용함으로써 시드의 희소함과 불균형으로 인한 문제를

해결할 수 있다. 다음으로 시드 생성에 기반한 기법에서 우리는 시드 주변이 아닌 새로운 지점에 시드를 생성한다. 우리는 오차가 발생한 영역에 사용자가 새로운 시드를 제공하는 동작을 모방하여 시스템을 학습하였다. 사용자의 의도를 학습함으로써 효과적으로 시드를 생성할 수 있다. 생성된 시드는 영상 분할의 정확도를 높일 뿐만 아니라 약지도학습을 위한 데이터로써 활용될 수 있다. 마지막으로 시드 주의 집중을 활용한 기법에서 우리는 의미론적 정보를 시드에 담는다. 기존에 제안한 기법들과 달리 영상 분할 동작과 시드 확장 동작이 통합된 모델을 제안한다. 시드 정보는 영상 분할 네트워크의 특징맵과 상호 교류하며 그 정보가 풍부해진다.

제안한 모델들은 다양한 실험을 통해 기존 기법 대비 우수한 성능을 기록하였다. 특히 시드가 부족한 상황에서 시드 확장 기법들은 훌륭한 대화형 영상 분할 성능을 보였다.

주요어: 대화형 영상 분할, 시드 확장, 강화 학습, 주의 집중 모듈, 심층 신경망

학번: 2012-20795