SeedNet: Automatic Seed Generation with Deep Reinforcement Learning for Robust Interactive Segmentation

Gwangmo Song* Heesoo Myeong* Kyoung Mu Lee

Department of ECE, ASRI, Seoul National University, 08826, Seoul, Korea
kfsgm@snu.ac.kr, heesoo.myeong@gmail.com, kyoungmu@snu.ac.kr

Abstract

In this paper, we propose an automatic seed generation technique with deep reinforcement learning to solve the interactive segmentation problem. One of the main issues of the interactive segmentation problem is robust and consistent object extraction with less human effort. Most of the existing algorithms highly depend on the distribution of inputs, which differs from one user to another and hence need sequential user interactions to achieve adequate performance. In our system, when a user first specifies a point on the desired object and a point in the background, a sequence of artificial user input is automatically generated for precisely segmenting the desired object. The proposed system allows the user to reduce the number of input significantly. This problem is difficult to cast as a supervised learning problem because it is not possible to define globally optimal user input at some stage of the interactive segmentation task. Hence, we formulate automatic seed generation problem as Markov Decision Process (MDP) and then optimize it by reinforcement learning with Deep Q-Network (DQN). We train our network on the MSRA10K dataset and show that the network achieves notable performance improvement from inaccurate initial segmentation on both seen and unseen datasets.

1. Introduction

Segmenting the object of interest in an image is one of the fundamental problems in computer vision. However, without knowing the user's intention, automatic object selection has inherent limitations because where and what objects should be extracted differs by users. For this reason, an interactive segmentation approach that receives information on the desired object roughly from a human in the form of a scribble or a bounding box and performs segmentation is widely used to extract the object from an image and video.

*First two authors contributed equally

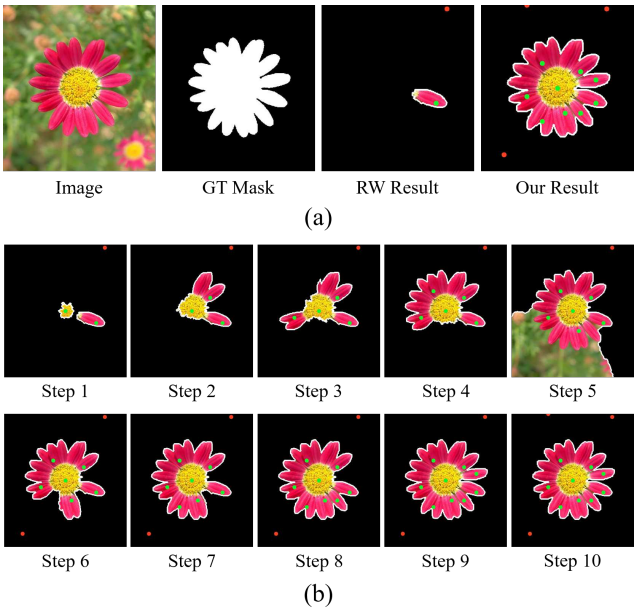

Figure 1. An automatic seed generation example. The green and red dots represent the foreground and the background seeds, respectively. (a) RW result is the output of random walker segmentation [13] algorithm with the initial seeds and our result is the segmentation output with the generated seeds from the SeedNet. (b) Seed generation process through the SeedNet. At each step, the SeedNet creates a new foreground or background seed input.

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 [14], the number of additional efforts by users is used as a measure of system performance. In this study, 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 1. The proposed system is designed to achieve high performance while significantly reducing the number of user input.

In this work, 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 paper include (1) the introduction of a Markov Decision Process (MDP) formulation for the interactive segmentation task where an 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.

2. Related Works

Interactive segmentation: 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 a desired object with various user input such as contour, scribble or bounding box. Numerous methods such as GrabCut [26], random walks [13, 16], geodesics [5], and methods with shape prior [30, 14] have been proposed.

Recently, learning-based interactive segmentation algorithms have attracted considerable attention. Wu *et al*. [33] considered interactive segmentation problem as a weakly supervised learning problem. In [33], sweeping line multiple instance learning (MIL) technique was presented. The MIL-based classifier is trained with foreground and background bags from user-annotated bounding box. Santner *et al*. [27] also treated the interactive segmentation problem in a weakly supervised learning manner. [27] showed that HoG descriptors learned with random forests successfully segment out a textured object. Kuang *et al*. [18] trained optimal parameters for a single image. The weights for color, texture, and smoothing terms are tuned during the iteration process.

Meanwhile, various studies have been carried out on algorithms for extending seed information. These studies are closely related to our work, in that extended seed information is used. Seeded region growing (SRG) techniques [1] is representative work. In each step of SRG, the most similar pixel among adjacent pixels is taken as an additional seed point. This process extends the seed set. GrowCut [31] also uses a similar algorithm concept. It uses cellular automaton as an image model. Automata evolution models the segmentation process. In each step, a labeled cell tries to attack its neighbors. If the defender cell's strength is lower than that of the attacker, the label of defender cell changes to that of the attacker. However, our method differs in that it proposes new points rather than expanding the seed area.

With the recent development of deep learning, Xu *et al*. [34] proposed a neural network architecture for interactive segmentation. Semantic information is considered by using fully convolutional neural networks (FCN) in their framework. By fine-tuning FCN block, the CNN structure can be used efficiently for interactive segmentation problems. Liew *et al*. [15] improved segmentation performance by creating global and local branches based on CNN architecture. However, our goal is not to train binary mask directly, but to train seed generation step that can help the existing segmentation algorithms.

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) [22]. Techniques such as prioritized experience replay [28], double DQN [29], dueling DQN [32], and A3C [21] 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 [9], 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 [9], [7] 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 [4], image captioning [24], video tracking [36], face hallucination [10] and video activity recognition task [35]. Andreas *et al*. [3] applied reinforcement learning to solve the question answering problem. They trained a network structure predictor with rein-

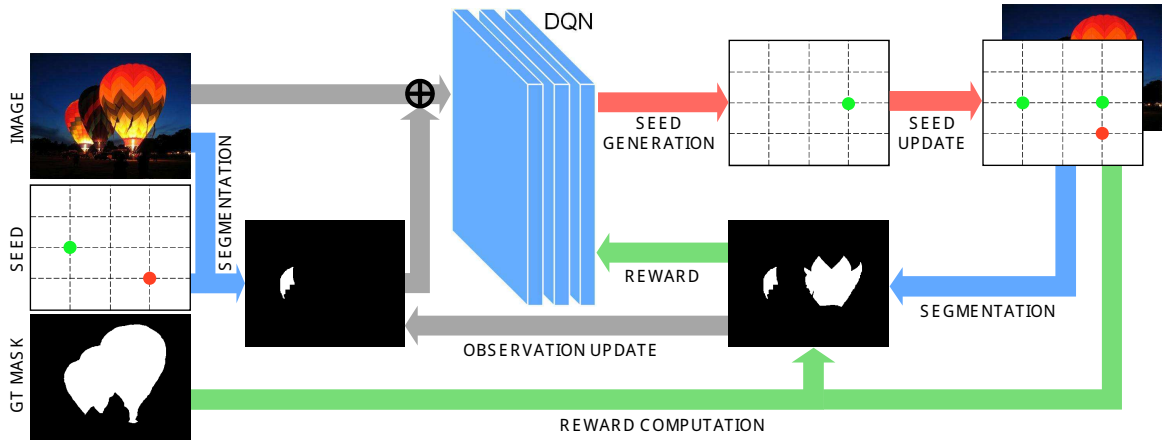

Figure 2. Overview of the proposed SeedNet. The image and the segmentation mask are the input of the DQN. The seed set is updated using the newly created seed from the DQN, and the mask is generated using the revised seed set. The obtained mask is used to calculate the reward value by comparing with 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.

forcement 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.

3. Automatic Seed Generation System

3.1. System Overview

In this work, 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 [22] agent using the segmentation result.

The overall process of SeedNet is shown in Figure 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 [13] 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.

3.2. Markov Decision Process (MDP)

The core part of the proposed SeedNet is to generate a sequence of seeds by the agent. We define the problem as an 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 bi-

nary 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}.

$$R_{\text{IoU}} = IoU(M, G), \tag{1}$$

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}.

$$R_{\text{diff}} = IoU(M, G) - IoU(M_{\text{prev}}, G), \tag{2}$$

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.

$$R_{\text{exp}} = \frac{exp^{k*IoU(M,G)} - 1}{exp^{k} - 1}, \tag{3}$$

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 R_{our} used in this paper is as follows:

$$R_{\text{our}} = \begin{cases} R_{\text{exp}} & \text{if } F_{\text{seed}} \in \text{SF or } B_{\text{seed}} \in \text{SB} \\ R_{\text{exp}} - 1 & \text{if } F_{\text{seed}} \in \text{WF or } B_{\text{seed}} \in \text{WB} \\ -1 & \text{otherwise} \end{cases}, \tag{4}$$

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.

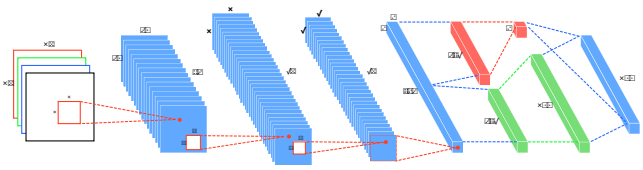

Figure 3. DQN architecture for SeedNet. The red block is the network for the state value function, and the green block is the network for the advantage function.

4. Training an Agent with Deep Reinforcement Learning

4.1. Deep Q-Network (DQN)

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*. [22] 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':

$$r + \gamma max_{a'} Q(s', a'), \tag{5}$$

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:

$$Loss(\theta) = \mathbb{E}[(r + \gamma max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2]. \tag{6}$$

For effective learning, we employ various techniques from Mnih *et al*. [22]. 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. 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.

4.2. Model Architecture

The DQN used in this study is shown in Figure 3. The structure of DQN used is almost similar to that of [22]. To improve the performance of the algorithm, we use the double DQN structure of [29] and dueling DQN structure of [32]. 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.

5. Experiments

We have experimented with several types of datasets. First, we use the MSRA10K saliency dataset [11] 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.

5.1. Network Learning

In this paper, 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, 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 [17] and utilize a learning rate of 1e-4. Also, the update rate to the target network is set to 1e-7.

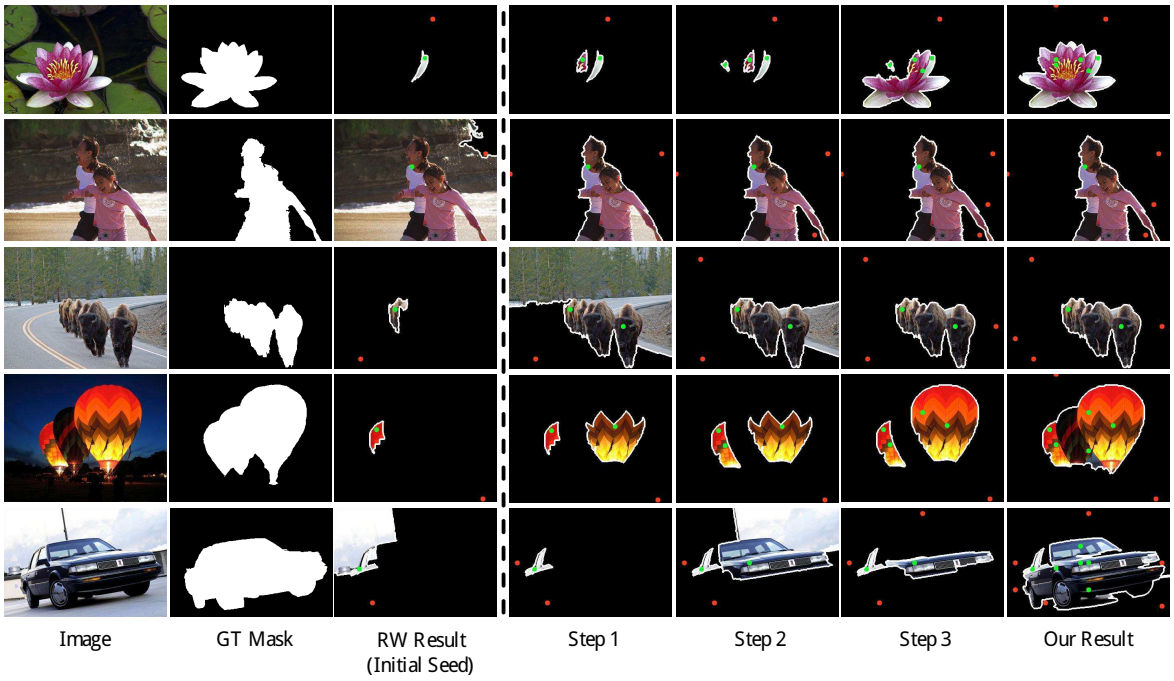

Figure 4. MSRA10K results. The left part shows the input image, GT mask, and initial seed with corresponding RW [13] result. The right part shows the SeedNet result, showing the first three steps and final result.

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.

5.2. Interactive Segmentation Results

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.

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 1. The IoU metric is used for evaluation.

Table 1. MSRA10K Result

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
RW [13]	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 2. Comparison with supervised methods

Method	FCN [19]	iFCN [34]	*SeedNet*
IoU	37.2	44.6	**60.97**

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

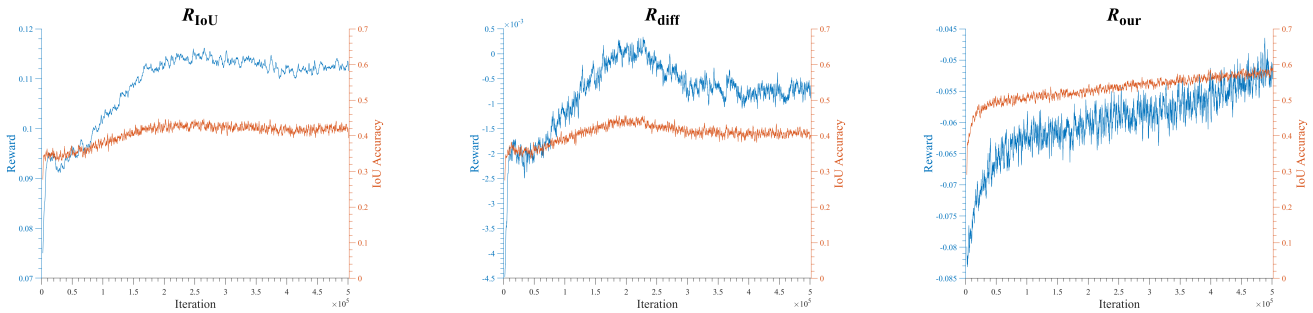

Figure 5. SeedNet learning progress graph using R_{IoU} (left), R_{diff} (center), and R_{our} (right). The reward value is indicated by the blue line and the left axis, and the IoU value is indicated by the orange line and the right axis. A common x-axis represents the progression of the learning iteration. For better visualization, the change is displayed every 100 steps and each point represents the running average value for 1000 steps.

affected by the initial seed distribution, and both RW and SeedNet show similar results. Qualitative results are shown in Figure 4. As shown in the figure, the automatically generated seed information gives better results than the initial seed. The figure 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.

Comparison with supervised methods: Additionally, we implement the FCN [19] and the iFCN [34] 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 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.

5.3. Ablation Experiments

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.

Reward: 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 3.2. 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 5.

Table 3. Ablation Experiments : Reward

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
RW [13]	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**

Table 4. Ablation Experiments : Segmentation

Method	Set 1	Set 2	Set 3	Set 4	Set 5	Mean
GC [26]	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 [14]	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**

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 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. As expected, we can confirm that the proposed reward function has better results than other reward functions.

Segmentation: SeedNet uses RW as an off-the-shelf segmentation algorithm, which can be replaced by other algorithms. SeedNet is trained using GrabCut (GC) [26] and GSCseq (GSC) [14], respectively. The results are shown in

Figure 6. MSRA10K result with SeedNet GC (upper two rows) and GSC versions (bottom two rows).

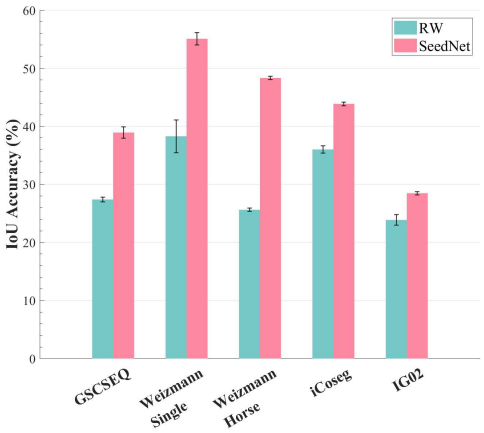

Figure 7. Results for unseen datasets. The horizontal axis represents each dataset, and the vertical axis represents the average IoU accuracy.

Table 4. Both the GC and GSC 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 [34]. The results of using GC and GSC are shown in Figure 6.

5.4. Unseen Dataset Experiments

To verify the scalability of the proposed SeedNet, we conducted experiments on an unseen dataset. As this system is learned 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 [14]: This dataset consists of a total of 151 images, including 49 pieces from the GrabCut dataset [26], 99 pieces from the Pascal VOC dataset [12], and 3 pieces from the Alpha matting dataset [25]. 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 [2]: 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.

Weizmann Horse [8]: A total of 328 images contain a side view of the horse. The dataset contains images and GT binary masks.

iCoseg [6]: 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 [20]: The new annotation of the Graz-02 dataset [23] 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 7. 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.

6. Conclusion

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.

Acknowledgements

This work was partly supported by the National Research Foundation of Korea(NRF) grant funded by the Korea Government(MSIT) (No. NRF-2017R1A2B2011862)

References

[1] R. Adams and L. Bischof. Seeded region growing. In *PAMI*. IEEE, 1994. 2

[2] S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In *CVPR*. IEEE, 2007. 8

[3] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In *NAACL*. Association for Computational Linguistics, 2016. 2

[4] J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In *ICLR*, 2015. 2

[5] X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. In *IJCV*. Springer, 2009. 2

[6] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*. IEEE, 2010. 8

[7] M. Bellver, X. Giro-i Nieto, F. Marques, and J. Torres. Hierarchical object detection with deep reinforcement learning. In *Deep Reinforcement Learning Workshop, NIPS*, 2016. 2

[8] E. Borenstein and S. Ullman. Learning to segment. In *ECCV*. Springer, 2004. 8

[9] J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In *ICCV*. IEEE, 2015. 2

[10] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li. Attention-aware face hallucination via deep reinforcement learning. In *CVPR*. IEEE, 2017. 2

[11] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. In *PAMI*. IEEE, 2015. 5

[12] 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. 8

[13] L. Grady. Random walks for image segmentation. In *PAMI*. IEEE, 2006. 1, 2, 3, 6, 7

[14] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In *CVPR*. IEEE, 2010. 1, 2, 7, 8

[15] J. Hao Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. Regional interactive image segmentation networks. In *ICCV*. IEEE, 2017. 2

[16] T. H. Kim, K. M. Lee, and S. U. Lee. Generative image segmentation using random walks with restart. In *ECCV*. Springer, 2008. 2

[17] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *CoRR*, 2014. 5

[18] Z. Kuang, D. Schnieders, H. Zhou, K.-Y. K. Wong, Y. Yu, and B. Peng. Learning image-specific parameters for interactive segmentation. In *CVPR*. IEEE, 2012. 2

[19] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*. IEEE, 2015. 6, 7

[20] M. Marszalek and C. Schmid. Accurate object localization with shape masks. In *CVPR*. IEEE, 2007. 8

[21] 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 *ICML*, 2016. 2

[22] 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. 2, 3, 5

[23] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. In *PAMI*. IEEE, 2006. 8

[24] Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image captioning with embedding reward. In *CVPR*. IEEE, 2017. 2

[25] C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In *CVPR*. IEEE, 2009. 8

[26] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *ToG*. ACM, 2004. 2, 7, 8

[27] J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof. Interactive texture segmentation using random forests and total variation. In *BMVC*. BMVA, 2009. 2

[28] T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In *ICLR*, 2016. 2

[29] H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In *AAAI*, 2016. 2, 5

[30] O. Veksler. Star shape prior for graph-cut image segmentation. In *ECCV*. Springer, 2008. 2

[31] V. Vezhnevets and V. Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In *Graphicon*, 2005. 2

[32] Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas. Dueling network architectures for deep reinforcement learning. In *ICML*, 2016. 2, 5

[33] 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 *CVPR*. IEEE, 2014. 2

[34] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In *CVPR*. IEEE, 2016. 2, 6, 7, 8

[35] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *CVPR*. IEEE, 2016. 2

[36] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Action-decision networks for visual tracking with deep reinforcement learning. In *CVPR*. IEEE, 2017. 2

CVPR2018 Paper Translation

姓名：陈勖

学号：2015301882

班号：08031503

种子网络：基于深度强化学习解决鲁棒交互式图像分割的自动化种子产生机

摘要

在这篇文章中，我们提出了一个基于深度强化学习的自动化种子产生方法去解决交互式图像分割问题。在交互式图像分割中最主要的话题之一就是在较少人工干预的情况下实现鲁棒和持续的物件提取。大多数现存算法依赖于输入的分布情况，而这一情况随着用户的变化而不同，需要持续的用户交互以获得较好的效果。在我们的系统中，当一个用户制定一个位于目标物体上的一个点和一个背景上的点时，一系列解算出的用户点会被自动产生出来用于精确分割目标物体。提出的系统使得用户能够显著减少输入规模。这个问题很难被转化为一个监督学习问题，因为在交互式分割任务的若干过程中定义一个全局最优的用户输入是不太可能的。因此，我们将这个自动化种子产生问题归结为一个马尔可夫决策过程，然后使用基于深度 Q 网络的强化学习方法进行优化。我们使用 MSRA10K 数据集去训练我们的网络，并且展示出，网络从不精确的可见和不可见的数据集的初始分割结果获得了较好的效果提升。

1 引入

在一副图像中分割出一个感兴趣的物体一直是计算机视觉中一个根本性的问题。但是，如果不知道用户的目的，自动化的物体选择有着内生性的缺陷，因为在哪里和什么样的物体应该被选择随着用户的变化而变化。因此，一个通过用户使用标签或者边界区域对应的输入来获取目标区域的粗略信息然后再进行图像分割的算法广泛使用于从图像或者视频中提取出一个目标。

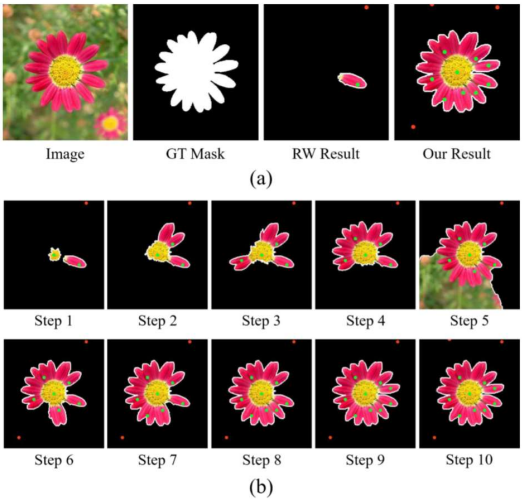

图 1.一个自动化种子产生实例。绿点和红点分别代表前景和后景种子。(a)RW 结果是随机移动算法【13】使用初始化种子进行分割，我们的结果是使用从种子网络中产生的种子产生的输出。(b) 种子从种子网络中产生的过程。在每一步中，种子网络产生一个新的前景和背景种子输入。

在交互式图像分割中一个关键的组成部分是鲁棒的物体提取并且符合用户意愿。对于一个有着很多物体的复杂背景来说，用户通常需要花很大代价去优化通过算法获得的结果。就这一方面来说，如何减少人工干预代价并同时保证交互式分割的效果就十分重要。在【14】，用户的额外干预量被作为一个系统的评价指标。在这个研究中，我们提出了一个新奇的方法去模拟人在指导交互式分割系统去获取需要目标的过程。当用户输入一个目标物体上的店时，我们的系统自动化的产生一系列的预测的用户输入点以精确的对感兴趣的目标物体进行定位，正如图 1 所展示的那样。提出的系统旨在在减少用户输入的情况下获得较好的效果。

在这个工作中，我们将自动化种子产生问题归结为一个连续化决策问题并使用深度强化学习训练种子产生智能体。我们的智能体通过分析图片和使用用户提供的初始化种子产生的前后景分割启动，并确定一个新的前景和后景的种子。在产生一个结合了初始化种子和产生的种子的分割之后，我们的只能提使用这个分割作为下一个输入并重复产生种子的过程。深度强化学习是适合我们这个任务的，因为我们无法在交互式分割的某些过程中定义全局最优的种子。除此以外，为了有效的学习，我们提出了一个新颖的基于 IoU 评分的反馈函数。提出的系统的优势在于，在之前训练过的数据集上的表现和未训练过的数据集上都有一致的较好表现。

这篇文章的贡献包括 (1) 将用户输入图像对应的种子去改进分割结果这一过程归结于马尔可夫决策过程 (2) 新颖的反馈函数设计以使用深度强化学习训练智能体进行自动化的种子产生。

2 相关工作

交互式图像分割：作为计算机视觉中的主要问题之一，交互式分割已经被研究了很长时间。许多交互式分割算法已尝试了使用多种用户输入来分割一个目标物体，例如标签，边框和轮廓。许多方法，例如 GrabCut【26】，随机移动【13，16】，测地线【5】以及形状优先方法【30，14】已经被提出。

最近，基于学习的交互式分割算法也引起了相当多的关注。吴等将交互式分割问题作为一个弱监督学习问题来进行考虑。在【33】中，MIL 方法被提出和展示。这种基于 MIL 的分类器是使用从用户标注中获得的前景和后景包来进行训练的。Santner 等【27】也使用弱监督学习方法来处理交互式图像分割问题。【27】展示了使用随机森林学习 HoG 描述子成功的分割出了一个有纹理的物体。Kuang 等【18】使用单一图像训练优化参数。颜色、纹理和平滑项的权重在迭代过程中被调整。

同时，许多基于延拓种子信息的算法的研究也已经完成。这些研究与我们的工作密切相关，因为在我们的工作中延拓种子对应的信息被使用了。SRF 方法【1】是代表性的工作。在 SRG 中的每一步中，在相邻像素中最相似的像素被作为一个额外的种子点。这个过程延拓了种子点集。GrowCut 也使用了一个相似的算法概念。它使用元胞自动机作为图像模型，将分割过程作为自动化进化过程。在每一步中，一个被标记的细胞尝试攻击它的邻居。如果防护细胞的能量比进攻细胞的能量低，防御细胞的标签就会被改变为进攻细胞的标签。但是，我们的方法与此不同，我们不是扩张种子区域，而是选择新的点。

结合最近的深度学习技术的发展，Xu 等【34】提出了一个为交互式图像分割所需的神经网络结构。在他们的框架中，语义信息通过使用 FCN 被考虑进去。通过精确调整 FCN 模块，卷积层可以被有效的使用来解决交互式分割问题。Llew 等【15】通过创造基于 CNN 结构的全局和局部分支改善了图像分割效果。但是，我们的目的不是直接训练二值化掩模，而是训练能够支持已有分割算法的种子产生步骤。

深度强化学习：深度强化学习一直以来由于其在一个雅塔丽游戏上通过使用深度 Q 网络【22】的精彩表现被大量研究。很多技术，例如优先经验重现【28】，双重 DQN【29】，对抗性 DQN【32】，和 A3C【21】已经被研究以提高强化学习算法的效果。强化学习算法通常被使用于雅塔丽游戏及机器人问题，但是它也有许多潜在使用领域，例如计算机视觉领域。

一个使用强化学习的计算机视觉的典型案例即是物体定位问题。在【9】中，作者将物体定位问题作为一个连续性的动态决策问题。在决策的每一步中，每一个行动被展示为探测区域的改变。通过使用一个图片及之前行动的深度表示，DQN 预测下一步的行动。类似于【9】，【7】使用区域变化行动和 DQN 去预测下一步改变。他们使用一

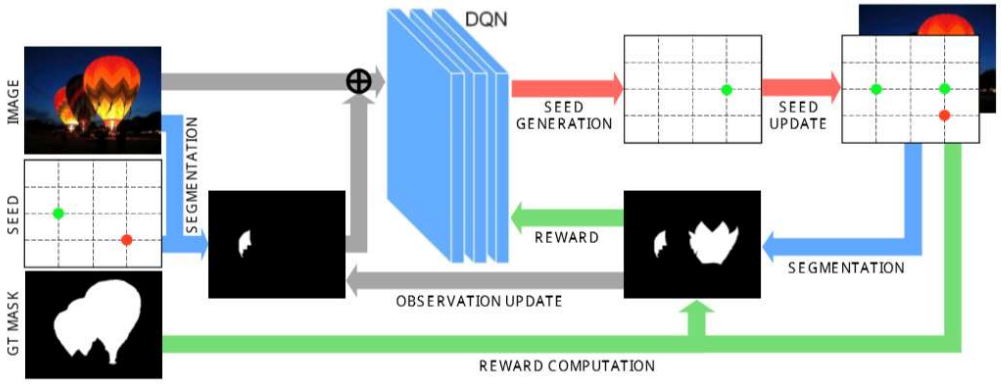

图 2.提出的种子网络的总体结构。图片和分割掩模是 DQN 的输入。使用从 DQN 中新产生的种子，种子集得到了更新，并且掩模从已经改进的种子集中产生。获得的掩模被用作和基本事实掩模进行比较获得反馈函数，并且这一过程重复。灰色的箭头显示了状态相关的行为，红色箭头显示了行动相关的行为，而绿色箭头显示了反馈相关的行为。

个树结构的搜索方法使得在单一过程中定位多个物体成为可能。

强化学习结构也在图像分类问题【4】、图像标注问题【24】、视频追踪问题【36】、面部识别【10】、视频实时识别问题【35】等问题中使用。Andreas 等【3】使用强化学习解决了问题回答这一问题。他们使用强化学习方法训练了一个网络结构预测器。

在大多数机器视觉应用中，研究人员使用的是注意力模型和强化学习的结合模型。但是，我们通过将图像空间视为一个大的行动空间来解决种子点产生问题。

3 自动化种子生产系统

3.1 系统概况

在这个工作中，我们为解决交互式分割问题提出了一个新颖的自动化种子产生系统。我们叫它“种子网络”。当一个图像和稀疏的种子信息被输入时，被提出系统的终极目标是创建额外的种子点并获得准确的分割结果。种子网络的核心模块是一个用于产生人工种子点的深度强化学习智能体。同时，种子网络也有一个根据产生的种子进行分割操作的离线分割模型。整个系统是通过使用分割结果学习 DQN【22】智能体得到的。

种子网络的整体流程由图 2 展示。系统的整个操作由图像和用户给定的初始种子来启动。通过挖掘输入信息，进行交互式分割产生了一个二值化掩模。我们使用随机移动分割【13】作为离线交互式分割算法。获得的二值化掩模和图像被复合并输入到 DQN 中。DQN 模型使用输入产生新的种子信息。新的种子信息包含被产生种子的位置和标签。由此，通过将产生的种子点信息加入已有的种子点信息，种子地图被更新。除此以外，使用新的种子点信息进行图像分割也会产生一个二值化掩模。获得的二值化掩模被用作两个用途：第一是通过比较获得的掩模和基本事实对应的掩模计算反馈函数；第二是获得的二值化掩模被用作下一代迭代中的一个观察结果。

循环操作的序列在整个训练过程中不断被重复。但是，在测试过程中，反馈部分被消除，只有种子产生过程被实施。通过重复产生一个种子的过程，一个有着一些人工产生种子的种子地图被获得。通过这种方式，我们可以在交互式分割中显著减少人工干预成本。

3.2 马尔可夫决策过程

提出的种子网络的核心部分是通过智能体产生一个种子序列。我们将这一问题定

义为一个包含状态、行为、反馈和智能体通过马尔可夫决策过程进行行动的马尔可夫决策过程。智能体将当前状态作为输入，采取一些行动，并获得对应的反馈。这个部分展示了提出的马尔可夫决策过程的定义。

状态：状态需要包含足够的信息去使得智能体采取最好的选择。在我们对问题的归纳过程中，图像整体状态的信息是必要的。除此以外，状态需要包含从每一步操作产生的变化中获得的观察信息。当每一步一个种子产生时，我们可以获得两个方面的信息：一个是新产生的种子地图，另一个是通过使用离线交互式分割算法获得的二值化掩模。考虑到我们希望提出的系统对种子位置具有鲁棒性，我们不考虑种子位置信息，仅将二值化掩模信息加入状态中。另外，过去的观测信息没有被使用，仅仅当前的观测结果在状态中被使用。

由此，在我们的问题归纳中，状态被定义为当前的二值化分割掩模和图像特征。与许多已有工作不同，提出的系统不使用任何深度特征表示作为状态的一部分。

行为：根据一个状态，智能体在行动空间内选择一个行动。在我们的问题归纳中，行动被定义为定位一个新的种子点。智能体通过状态确定一个在 2 维网格中的种子的位置和标签。如果我们设定这个二维网格为图像中的所有像素，行动空间会变得非常大，给训练造成困难。因此，新种子能够被放置的 2 维网格区域被稀疏的设定为 20×20 的大小。由于前景和后景的原因，一共存在有 800 多种行动。如果智能体选择了 800 多种行动中的一种，一个新的种子点就被产生在对应的位置上。同时，过程中不存在明确的终止行动，由于定义终止行动是十分困难的。因此，我们在这个过程产生十个种子点之后终止整个过程。

反馈：反馈信号评价智能体行为产生的结果。总体而言，在一个博弈环境中，输/赢的分数被作为一个反馈函数。在我们的系统中，智能体行为的结果是种子的位置和分割掩模。因此，我们使用分割掩模的精确度作为评分的考量。分割的精确度可以通过比较其与基本事实的掩模作对比来获得。作为评价，IoU 是一个通用的标准。因此，直觉的基本反馈函数使用 IoU 作为一个反馈函数。包含 IoU 反馈函数被描述为 R_{IoU}：

$$R_{IoU} = IoU(M, G), \qquad (1)$$

在这其中 M 指的是获得的分割掩模，G 指的是基本事实对应的掩模。另外一个基本反馈函数是使用 IoU 的变化趋势。它比较了当下的掩模的 IoU 值和之前的掩模的 IoU 值，在这个值上升时给出一个成功信号，在这个值下降时给出一个失败信号。这类似在博弈环境中的赢/输反馈信号。但是，在我们的环境中，我们需要获得变化的数量的同时还需要获得变化的方向。因此，一个更灵活的反馈信号可以被设计为使用IoU的变化作为反馈值，而不是使用二值化反馈。这被描述为 R_{diff}。

$$R_{diff} = IoU(M, G) - IoU(M_{prev}, G) \qquad (2)$$

在这其中 M_{prev} 是前一步的分割掩模。除此以外，通过使用指数 IoU 模型（$R_{\exp}$）而不是线性化的 IoU 模型，我们可以设计一个反馈信号，它可以更多的关注到高的 IoU 值。

$$R_{\exp} = \frac{\exp^{k*IoU(M,G)} - 1}{\exp^{k} - 1} \qquad (3)$$

其中 k 为一个常数。与此同时，如果我们有种子位置的信息，同时也有掩模的信息，我们可以产生一个额外的信号帮助计算IoU反馈。我们通过比较基本事实的掩模和新产生的种子来判断，而不是使用 IoU 变化来判断成功还是失败。那就是，如果心中自的与基本事实中的相对应位置的标签相符的话就是成功，否则判为失败。带着相同的概念，我们将基本事实对应的掩模分为四个区域，将他们与种子标签进行比较。为了将基本事实掩模分为四个区域，我们在物体的内部和外部创建了额外的边界，其与物体的边界有着一定的空间。那就是说，四个区域通过三个边界被创建，包括一个已经存在的物体边

界。这四个区域按照从物体的中心到图像的边缘被命名为强前景（SF），弱前景（WF），弱背景（WB），和强背景（SB）。当一个新的种子点被指定时，不同的反馈函数依照种子点所属的不同点区域被使用。

例如，如果新给定的前景种子点属于掩模的强前景区域，我们使用 IoU 指数反馈。同时，如果前景种子区域属于 WF 区域，它也是一个成功的案例但并不被期待，因此一个减少的反馈信号被使用。否则，一个前景种子点被错误的认为在背景区域，一个固定的-1 反馈值被使用。类似的，当一个背景种子点被给出时，我们可以与前经典类似的获得一个反馈值。在我们论文中使用的 R_{our} 被定义如下：

$$R_{our} = \begin{cases} R_{\exp} & if \quad F_{seed} \in SF \quad or \quad B_{seed} \in SB \\ R_{\exp} - 1 & if \quad F_{seed} \in WF \quad or \quad B_{seed} \in WB \\ -1 & otherwise \end{cases} \tag{4}$$

其中 F_{seed} 指一个前景种子而 B_{seed} 指一个背景种子。我们从掩模信息和离散的从种子信息获得的成功/失败反馈获得一个持续性的评分反馈。最终，我们通过混合两种反馈提出了一个新颖的反馈函数。我们将在实验章节中比较新提出的反馈函数和其他反馈函数的不同。

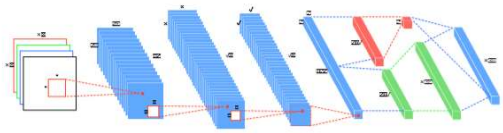

图 3.种子网络的 DQN 结构。红色的区块是状态值函数的网络，绿色的区块是优势函数的网络。

4 使用深度强化学习训练一个智能体

4.1 深度 Q 网络（DQN）

有了前面将问题归结为马尔可夫决策过程的过程，种子产生智能体可以通过深度强化学习进行训练。在这个研究中，我们使用 Mnih 等【22】提出的 DQN 算法训练智能体。DQN 学习行动值函数 $Q(s,a)$，它定义了智能体在状态 s 下采取行动 a 期望得到的反馈。在训练后，智能体根据学习到的 Q 函数来采取行动。Q 学习目标可以根据给定的 s,a,s' 来定义：

$$r + \gamma \max_{a'} Q(s',a') \tag{5}$$

其中 r 是反馈，γ 为不信任因子，并且 s' 和 a' 分别代表了下一步的状态和行动。DQN 是一个使用深度神经网络估计 Q 函数的技术。训练 Q 函数中的损失函数可表述为：

$$Loss(\theta) = \mathbb{E}[(r + \gamma \max_{a'} Q(s', a'; \theta) - Q(s, a; \theta))^2]. \tag{6}$$

为了更有效的训练，我们使用从 Mnih 等【22】获得的许多不同技术。首先，我们使用一个目标网络去解决学习稳定度差的问题。通过引入一个单独于在线网络的目标网络，在在线网络被更新的同时目标网络的参数在几次迭代之后便固定。这一方法已经显著提高了学习的稳定性。其次，我们使用了ε-贪心准则作为行为准测。ε-贪心准则使用一个概率为ε的随机行为和一个概率为 1-ε的最大化 Q 函数的行为。最后是重现已有过程去解决 DQN 学习所使用的数据的自相关问题。我们创建了一个重现已有过程的缓冲区，依ε前进，并将重现记忆存进缓存区 (s,a,r,s') 。在学习过程中，批处理容量的样例是被随机从缓存区选择的，以减少数据的互相关特性。

4.2 模型结构

DQN 被用于研究的方式被展示在图 3 中。DQN 的使用结构与【22】的使用方法几乎类似。为了改进算法的效果，我们使用【29】的双 DQN 结构和【32】的对抗性 DQN 结构。输入图像和上一步分割产生的二值化掩模将被重新放缩为 84×84 然后被输入进网

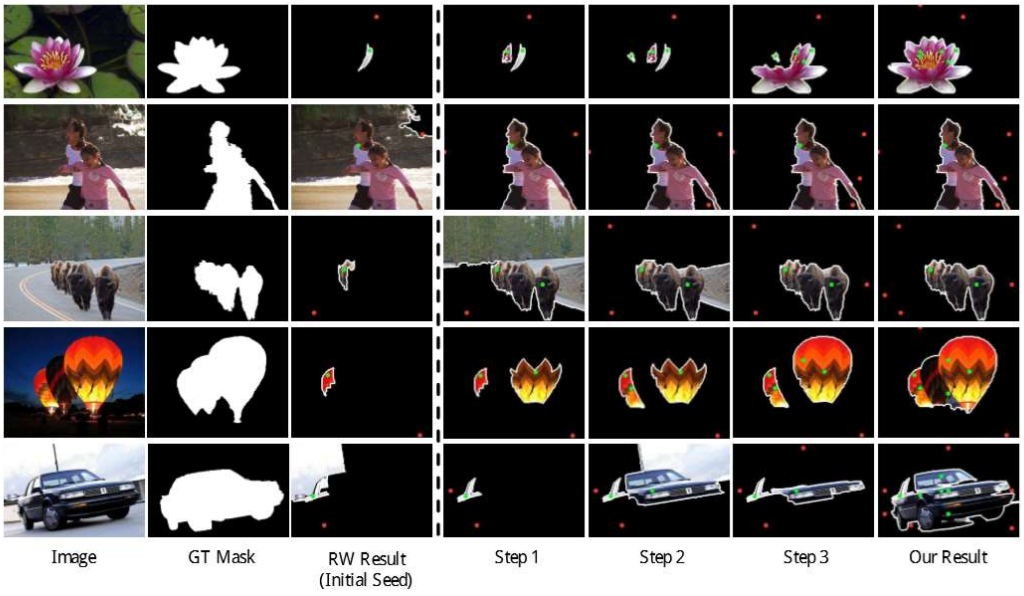

图 4.MSRA10K 结果。左边部分显示的是输入图片，基本事实对应掩模和初始种子带着相应的 RW【13】结果。右边部分展示了种子网络的结果，展示了前三步和最后一步结果。

络。三个卷积操作并接着 Relu 激活函数被使用在输入上。通过发挥对抗性网络的优势，在全连接操作之后的 512 维层被分成两部分去学习优势函数和状态值函数。接着，通过一个全连接操作，优势函数 $A(a,s)$ 输出一个 800 维输出，和行动空间一致。同时，状态值函数 $V(s)$ 是一个标量。最后，优势函数和状态值函数后获得了 Q 函数。根据 Q 函数的最大值对应的行动，当下行动被决定。如果行动标号比 400 小，那么就是前景种子，否则就是后景种子，并将其减 400 去获得相应坐标。最后，将行动标号转化为 20×20 的网格坐标去决定新的种子坐标位置。

5 实验结果

我们使用了多种多种数据集作为实验。首先，我们使用 MSRA10K 显著性数据集【11】去训练并将我们的结果和初始化种子得到的初始化结果进行比较。我们进行了在许多单物体数据集上做了对比试验，而这些数据集并没有在训练时并没有使用这些数据集。

5.1 网络学习

在这篇文章中，种子网络用 MSRA10K 显著性数据集从头训练。在训练过程中，10000 步训练过程被进行以建立一个训练中需要使用的过程重现缓存区。在预训练步骤中，实际的训练没有进行，但是这个过程中的训练经验将存储在缓存区。我们使用了 50000 条经验组成的缓存区并使用 32 作为训练集大小。为了探索，我们使用了ε贪心算法。在训练过程中，ε在 10000 步中由 1 降到 0.3.在序列化训练过程中，ε固定为 0.2。在训练过程，在学习网络中学习出的行为被选择的概率为 1-ε。这个特殊网络的大小参数在图 3 中进行了展示，并且不信任因子γ被固定设置为 0.9.每个历程都包含一个产生 10 个种子的过程。为了训练，我们使用了一个 Adam 优化器【17】并使用一个 1e-4 的学习率。同时，目标网络的更新率被设定为 1e-7。在之前提到，一个 20×20 的网格被使用作为行动空间，而在指数反馈函数中的 k 值固定设为 5。

5.2 交互式分割结果

首先，我们的性能评价是在 MSRA10K 数据集上完成的。MSRA10K 数据集由 10000 幅图像组成，并且我们使用其中的 9000 幅用作训练而剩下的 100 幅作为测试。每幅图像都由一个 RGB 图像和一个代表基本事实的掩模组成，而种子信息并没有被包括进去。图像的大小大约是 300×400 个格子。为了加速学习过程，每张图像和基本事实在学习过程中都被减小到原 1/4 大小。相同的 84×84 图像在学习和训练过程中被输入到 DQN 中。但是，当分割是在一个新的产生的种子的基础上实施的，分割在学习过程中被实施在一个 1/4 大小的图片上以获得更快的结果，原有图片大小是被应用在测试时间。当分割的图片的大小增加时，种子的数量也同时增加了。在训练过程中，一个半径为 3 个像素的圆被用作一个种子，而在测试过程中半径为 13 的像素的圆被用作一个种子。

表 1.MSRA10K 结果

方法	集 1	集 2	集 3	集 4	集 5	平均
RW【13】	39.59	39.65	39.71	39.77	39.89	**39.72**
种子网络	60.70	60.12	61.28	61.87	60.90	**60.97**

表 2.和监督方法的对比

方法	FCN【19】	iFCN【34】	*种子网络*
IoU	37.2	44.6	**60.97**

考虑到种子信息没有包括在 MSRA10K 数据集中，我们使用原有从基本事实掩模信息中随机产生的初始种子点进行实验。我们使用分别对原有的基础事实对应的掩模进行擦除和腐蚀去形成一个与物体边界距离稍远的区域并从区域中随机选择前景和后景种子点。当初始种子点被随机确定时，我们按顺序进行 5 个实验并使用平均值评价效果。我们使用 RW 分割方法作为系统中的一个离线分割算法。结果包含使用了最初点和新产生种子点的结果被比较和展示在表 1 中。IoU 指标被使用作为评价。

结果西安市，当使用新产生的种子网络被使用时相比于用初始种子点进行 RW 分割准确度大大提升。同时，我们改变从集合 1 到集合 5 的初始种子点的分布，但并没有显著影响初始种子点的分布，并且 RW 和种子网络展示出类似的效果。量化结果也被战士在图 4 中。就像在图 4 中展示的那样，自动化产生的种子信息比原有的初始化种子信息能给出更好的结果。图 4 也展示出从第三部到最后的结果。到饱和时种子的被使用的平均数量为 5.39 次点击。因此，提出的算法的阈值是合理的，因为它使得进化算法进行了 10 次。但是，考虑到种子网络是在一个稀疏的网格上产生种子，在第三行的那种情况下很难提出一个更好的位置。与此同时，增加的种子被很好的放置并没有破坏初始种子的意图。

与监督方法的比较：除此以外，我们使用了 FCN【19】和 iFCN【34】基线。我们输入一个 80×80 的图像，和我们的网络输入很相似，将全连接层改变为卷积层，使用填充的方式获得一个 10×10 的输出图像，并将其反卷积回原始大小。同时，网络被精细的训练。我们在 RGB 通道中加入了两个种子输入通道给 iFCN。结果被展示在表 2 中。虽然使用预训练网络和更大的图片输入可以获得更好的结果，我们注意到监督学习获得的效果比当前设置产生的效果更差。

5.3 消融实验

为了分析被提出的系统，我们替换了系统中若干关键部分。当改变了相关元素和保持其他元素不变，实验被进行。

反馈：我们的 DQN 根据比较观察基础事实获得的反馈来进行更新。为了证实提出的反馈函数的有效性，我们训练了在 3.2 节中的叙述的一个使用单一简单补偿函数的系统。为了对比，R_{IoU} 和 R_{diff} 被使用，反馈函数的值随学习时间的变化和训练集的 IoU 精确度随学习实践的变化被展示在图 5 中。被展示在坐标的对应每一个图的反馈值轴是不同的同时，在右边的 IoU 轴在三幅图中含有相

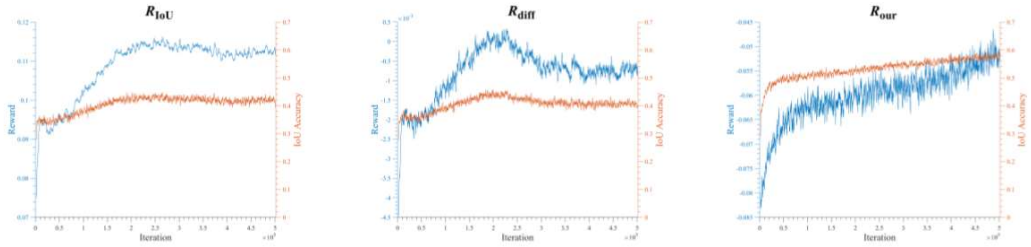

图 5.种子网络学习过程图像使用R_{IoU}（左侧），R_{diff}（中间），和R_{our}（右侧）。反馈函数由蓝线和左侧轴表示，IoU 的值由橘色线及右侧轴表示。一个共同的 x 轴表示学习迭代次数的进程。为了更好的可视化，每一百步的变化被展示出来，每一个点每一千步中的运行平均值。

同的坐标轴。比较三幅图，我们可以看到简单反馈函数一开始的反馈值在增大但保持在一定的书评，因此 IoU 不再提升。同时，在提出的反馈函数中，反馈和 IoU 值都逐渐提升。在每一个测试集上使用种子网络学习使用每一个反馈函数的结果被展示在表 3 中。就像期望的那样，我们可以确定提出的反馈函数可以获得比其他反馈函数更好的结果。

表 3.消融实验：反馈

方法	表 1	表 2	表 3	表 4	表 5	平均
RW【13】	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**

表 4.消融实验：分割

方法	表 1	表 2	表 3	表 4	表 5	平均
GC【26】种子网络（GC 版）	38.15	38.29	38.35	38.70	38.71	**38.44**
	52.43	51.89	51.84	52.10	52.26	**52.10**
GSC【14】种子网络（GSC 版）	57.85	58.10	58.50	58.57	58.70	**58.34**
	63.09	62.70	64.24	63.16	64.19	**63.48**

分割：种子网络使用 RW 作为一个离线分割

图片　　基本事实掩模　　GC/GSC 结果　　我们的结果

图 6.MSRA10K 结果使用 GC 版（上两行）和 GSC 版（下两行）种子网络。

算法，其也可以被其他算法替换。种子网络分别使用使用 GC【26】和 GSCseq【14】进行训练。结果被展示在表 4 中。GC 和 GDC 版本的种子网络都相比原来的结果展示出 IoU 的提升。由于其他分割算法也可以被这样使用，当使用以 CNN 作为基础的算法，例如 iFCN【34】时，可以预期将有更好的结果。GC 和 GCS 的结果被展示在图 6 中。

5.4 未视的数据集结果

为了验证提出的种子网络的可扩展性，我们在未视数据集上进行了实验。因为我们的系统使用了显著性数据集 MSRA10K 进行训练，我们使用许多但物体二值化分割数据集测试我们的智能体，而不是使用 MSRA10K 数据集的验证图像。实验建立过程和 MSRA10K 类似，评价也使用 5 份随机

初始化种子的平均值来进行。

GSCSEQ【14】：这个数据集一共有 151 张图片，包括 GrabCut 数据集【25】中的 49 份图像，和 Pascal VOC 数据集【12】中的 99 张图像，还有三张 Alpha matting 数据集【25】中的三张图像。数据集包括 RGB 图像，基本事实二值化掩模以及其他信息。但是在实验中，种子点是不使用其他信息从二值化掩模中产生的。

Weizamann 单物体【2】：Weizmann 单物体数据集包括 100 张单物体图像，包括每一张图像的三种基本事实掩模。这三种基本事实是随着标记用户的主体不同而由略微不同，但我们仅使用第一个基本事实作为评价。

Weizmann Horse【8】：一共有包含马的侧视的 328 张图。数据集包括图像和二值化掩模的基本事实。

iCoseg【6】：ICoseg 是一个主要用于互分割的数据集，它有 38 类 643 张图像。每张图片都有二值化掩模。

IG02【20】:从 INRIA 中新提出的 Graz-02 数据集【23】包含三类：自行车、汽车和人。每一类一共有 479 张测试图用于测试。有些图片包含多个物体，但只有一个物体用于测试。

测试结果展示在图 7 中。在所有的 5 个数据集中，我们可以看到使用种子网络进行种子产生的结果相比于原有种子对应的结果有了明显的提升。特别的，在 Weizmann Horse 数据集中有了 20%以上精确度的提升。种子网络，在另种角度来看，在 IG02 的表现相对较弱，这是由于我们训练时使用单物体样本而测试样本中多物体存在的缘故。同时，我们可以确定提出的种子网络即使在一个在完全不同并且在训练中没有接触过的环境中的数据集上也有较好的应用。

6 结论

我们提出了一个新颖的交互式分割智能体来帮助用户精确的分割一个物体。智能体能预测用户的目的并介绍用户的干预精力。勇士，这一方法也有潜力在许多计算机视觉的问题上利用用户的意图例如语义分割。不止

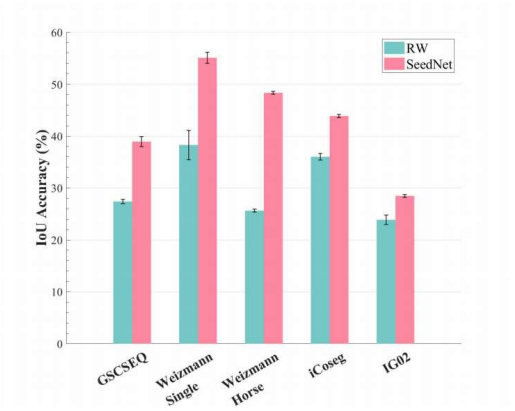

图 7.未视数据集的结果。横轴表示每个数据集，纵轴表示平均 IoU 精确度。

于此，我们的智能体还能够帮助较少像素级标定任务的成本。

致谢

这一工作部分由韩国国家研究中心基金(NRF)提供帮助，其由韩国政府提供资金（MSIT）。(No.NRF-2017R1A2B2011862)。

参考文献

【1】.R. Adams and L. Bischof. Seeded region growing. In PAMI. IEEE, 1994. 2

【2】.S. Alpert, M. Galun, R. Basri, and A. Brandt. Image segmentation by probabilistic bottom-up aggregation and cue integration. In CVPR. IEEE, 2007. 8

【3】.J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Learning to compose neural networks for question answering. In NAACL. Association for Computational Linguistics, 2016. 2

【4】.J. Ba, V. Mnih, and K. Kavukcuoglu. Multiple object recognition with visual attention. In ICLR, 2015. 2

【5】.X. Bai and G. Sapiro. Geodesic matting: A framework for fast interactive image and video

segmentation and matting. In IJCV. Springer, 2009. 2

【6】.D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In CVPR. IEEE, 2010. 8

【7】.M. Bellver, X. Giro-i Nieto, F. Marques, and J. Torres. Hierarchical object detection with deep reinforcement learning. In Deep Reinforcement Learning Workshop, NIPS, 2016. 2

【8】.E. Borenstein and S. Ullman. Learning to segment. In ECCV. Springer, 2004. 8

【9】.J. C. Caicedo and S. Lazebnik. Active object localization with deep reinforcement learning. In ICCV. IEEE, 2015. 2

【10】.Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li. Attention-aware face hallucination via deep reinforcement learning. In CVPR. IEEE, 2017. 2

【11】.M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. In PAMI. IEEE, 2015. 5

【12】.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. 8

【13】.L. Grady. Random walks for image segmentation. In PAMI. IEEE, 2006. 1, 2, 3, 6, 7

【14】.V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman. Geodesic star convexity for interactive image segmentation. In CVPR. IEEE, 2010. 1, 2, 7, 8

【15】.J. Hao Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. Regional interactive image segmentation networks. In ICCV. IEEE, 2017. 2

【16】.T. H. Kim, K. M. Lee, and S. U. Lee. Generative image segmentation using random walks with restart. In ECCV. Springer, 2008. 2

【17】.D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In CoRR, 2014. 5

【18】.Z. Kuang, D. Schnieders, H. Zhou, K.-Y. K. Wong, Y. Yu, and B. Peng. Learning image-specific parameters for interactive segmentation. In CVPR. IEEE, 2012. 2

【19】.J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In CVPR. IEEE, 2015. 6, 7

【20】.M. Marszalek and C. Schmid. Accurate object localization with shape masks. In CVPR. IEEE, 2007. 8

【21】.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 ICML, 2016. 2

【22】.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. 2, 3, 5

【23】. A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. In PAMI. IEEE, 2006. 8

【24】.Z. Ren, X. Wang, N. Zhang, X. Lv, and L.-J. Li. Deep reinforcement learning-based image captioning with embedding reward. In CVPR. IEEE, 2017. 2

【25】.C. Rhemann, C. Rother, J. Wang, M. Gelautz, P. Kohli, and P. Rott. A perceptually motivated online benchmark for image matting. In CVPR. IEEE, 2009. 8

【26】.C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In ToG. ACM, 2004. 2, 7, 8

【27】.J. Santner, M. Unger, T. Pock, C. Leistner, A. Saffari, and H. Bischof. Interactive texture segmentation using random forests and total variation. In BMVC. BMVA, 2009. 2

【28】.T. Schaul, J. Quan, I. Antonoglou, and D. Silver. Prioritized experience replay. In ICLR, 2016. 2

【29】.H. Van Hasselt, A. Guez, and D. Silver. Deep reinforcement learning with double q-learning. In AAAI, 2016. 2, 5

【30】. O. Veksler. Star shape prior for graph-cut image segmentation. In ECCV. Springer, 2008. 2

【31】.V. Vezhnevets and V. Konouchine. Growcut: Interactive multi-label nd image segmentation by cellular automata. In Graphicon, 2005. 2

【32】.Z. Wang, T. Schaul, M. Hessel, H. Van Hasselt, M. Lanctot, and N. De Freitas. Dueling network architectures for deep reinforcement learning. In ICML, 2016. 2, 5

【33】.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 CVPR. IEEE, 2014. 2

【34】.N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. Deep interactive object selection. In CVPR. IEEE, 2016. 2, 6, 7, 8

【35】.S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. Endto-end learning of action detection from frame glimpses in videos. In CVPR. IEEE, 2016. 2

【36】.S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Young Choi. Actiondecision networks for visual tracking with deep reinforcement learning. In CVPR. IEEE, 2017. 2