Unsupervised Meta-Learning of
Figure-Ground Segmentation via Imitating Visual Effects

Ding-Jie Chen,[†] Jui-Ting Chien,[‡] Hwann-Tzong Chen,[‡] Tyng-Luh Liu[†]

[†]Institute of Information Science, Academia Sinica, Taiwan
[‡]Department of Computer Science, National Tsing Hua University, Taiwan
{djchen.tw, ydnaandy123}@gmail.com, htchen@cs.nthu.edu.tw, liutyng@iis.sinica.edu.tw

Abstract

This paper presents a "learning to learn" approach to figure-ground image segmentation. By exploring webly-abundant images of specific visual effects, our method can effectively learn the visual-effect internal representations in an unsupervised manner and uses this knowledge to differentiate the figure from the ground in an image. Specifically, we formulate the meta-learning process as a compositional image editing task that learns to imitate a certain visual effect and derive the corresponding internal representation. Such a generative process can help instantiate the underlying figure-ground notion and enables the system to accomplish the intended image segmentation. Whereas existing generative methods are mostly tailored to image synthesis or style transfer, our approach offers a flexible learning mechanism to model a general concept of figure-ground segmentation from unorganized images that have no explicit pixel-level annotations. We validate our approach via extensive experiments on six datasets to demonstrate that the proposed model can be end-to-end trained without ground-truth pixel labeling yet outperforms the existing methods of unsupervised segmentation tasks.

Introduction

In figure-ground segmentation, the regions of interest are conventionally defined by the provided ground truth, which is usually in the form of pixel-level annotations. Without such supervised information from intensive labeling efforts, it is challenging to teach a system to learn what the figure and the ground should be in each image. To address this issue, we propose an unsupervised meta-learning approach that can simultaneously learn both the *figure-ground concept* and the corresponding *image segmentation*.

The proposed formulation explores the inherent but often unnoticeable relatedness between performing image segmentation and creating visual effects. In particular, to visually enrich a given image with a special effect often first needs to specify the regions to be emphasized. The procedure corresponds to constructing an internal representation that guides the image editing to operate on the target image regions. For this reason, we name such an internal guidance as the *Visual-Effect Representation* (VER) of the image. We observe that for a majority of visual effects, their
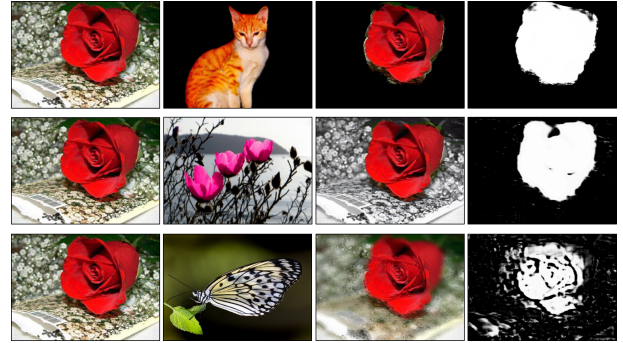
Figure 1: Given the same image (1st column), imitating different visual effects (2nd column) can yield distinct interpretations of figure-ground segmentation (3rd column), which are derived by our method via referencing the following visual effects (from top to bottom): `black background`, `color selectivo`, and `defocus`/Bokeh. The learned VERs are shown in the last column, respectively.

resulting VER is closely related to image segmentation. Another advantage of focusing on visual-effect images is that such data are abundant from the Internet, while pixel-wise annotating large datasets for image segmentation is time-consuming. However, in practice, we only have access to the visual-effect images, but not the VERs as well as the original images. Taking all these factors into account, we reduce the meta-problem of figure-ground segmentation to predicting the proper VER of a given image for the underlying visual effect. Owing to its data richness from the Internet, the latter task is more suitable for our intention to cast the problem within the unsupervised generative framework.

Many compositional image editing tasks have the aforementioned properties. For example, to create the `color selectivo` effect on an image, as shown in Fig. 2, we can *i*) identify the target and partition the image into foreground and background layers, *ii*) convert the color of background layer into grayscale, and *iii*) combine the converted background layer with the original foreground layer to get the final result. The operation of color conversion is local—it simply "equalizes" the RGB values of pixels in certain areas. The quality of the result depends on how properly the layers

are decomposed. If a part of the target region is partitioned into the background, the result might look less plausible. Unlike the local operations, to localize the proper regions for editing would require certain understanding and analysis of the global or contextual information in the whole image. In this paper, we design a GAN-based model, called Visual-Effect GAN (VEGAN), that can learn to predict the internal representation (*i.e.*, VER) and incorporate such information into facilitating the resulting figure-ground segmentation.

We are thus motivated to formulate the following problem: Given an unaltered RGB image as the input and an image editing task with known compositional process and local operation, we aim to predict the proper VER that guides the editing process to generate the expected visual effect and accomplishes the underlying figure-ground segmentation. We adopt a data-driven setting in which the image editing task is exemplified by a collection of image samples with the expected visual effect. The task, therefore, is to transform the original RGB input image into an output image that exhibits the same effect of the exemplified samples. To make our approach general, we assume that *no* corresponding pairs of input and output images are available in training, and therefore supervised learning is not applicable. That is, the training data does not include pairs of the original color images and the corresponding edited images with visual effects. The flexibility is in line with the fact that although we could fetch a lot of images with certain visual effects over the Internet, we indeed do not know what their original counterpart should look like. Under this problem formulation, several issues are of our interest and need to be addressed.

First, *how do we solve the problem without paired input and output images?* We build on the idea of generative adversarial network and develop a new unsupervised learning mechanism (shown in Figs. 2 & 3) to learn the internal representation for creating the visual effect. The *generator* aims to predict the internal VER and the *editor* is to convert the input image into the one that has the expected visual effect. The compositional procedure and local operation are generic and can be implemented as parts of the architecture of a ConvNet. The *discriminator* has to judge the quality of the edited images with respect to a set of sample images that exhibit the same visual effect. The experimental results show that our model works surprisingly well to learn meaningful representation and segmentation without supervision.

Second, *where do we acquire the collection of sample images for illustrating the expected visual effect?* Indeed, it would not make sense if we have to manually generate the labor-intensive sample images for demonstrating the expected visual effects. We show that the required sample images can be conveniently collected from the Internet. We provide a couple of scripts to explore the effectiveness of using Internet images for training our model. Notice again that, although the required sample images with visual effects are available on the Internet, their original versions are unknown. Thus supervised learning of pairwise image-to-image translation cannot be applied here.

Third, *what can the VER be useful for, in addition to creating visual effects?* We show that, if we are able to choose a suitable visual effect, the learned VER can be used to not only establish the intended figure-ground notion but also derive the image segmentation. More precisely, as in our formulation the visual-effect representation is characterized by a real-valued response map, the result of figure-ground separation can be obtained via binarizing the VER. Therefore, it is legitimate to take the proposed problem of VER prediction as a surrogate for unsupervised image segmentation.

We have tested the following visual effects: *i*) `black background`, which is often caused by using flashlight; *ii*) `color selectivo`, which imposes color highlight on the subject and keeps the background in grayscale; *iii*) `defocus/Bokeh`, which is due to depth of field of camera lens. The second column in Fig. 1 shows the three types of visual effects. For these tasks our model can be end-to-end trained from scratch in an unsupervised manner using training data that do not have either the ground-truth pixel labeling or the paired images with/without visual effects. While labor-intensive pixel-level segmentations for images are hard to acquire directly via Internet search, images with those three effects are easy to collect from photo-sharing websites, such as Flickr, using related tags.

Related Work

We discuss below some related work on the topics of generative adversarial networks and image segmentation.

Generative Adversarial Networks

The idea of GAN (Goodfellow et al. 2014) is to generate realistic samples through the adversarial game between generator G and discriminator D. GAN becomes popular owing to its ability to achieve unsupervised learning. However, GAN also encounters many problems such as instability and model collapsing. Hence later methods (Radford, Metz, and Chintala 2016; Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) try to improve GAN in both the aspects of implementation and theory. DCGAN (Radford, Metz, and Chintala 2016) provides a new framework that is more stable and easier to train. WGAN (Arjovsky, Chintala, and Bottou 2017) suggests to use Wasserstein distance to measure the loss. WGAN-GP (Gulrajani et al. 2017) further improves the way of the Lipschitz constraint being enforced, by replacing weight clipping with gradient penalty.

To reduce the burden of G, Denton *et al.* (Denton et al. 2015) use a pyramid structure and Karras *et al.* (Karras et al. 2018) consider a progressive training methodology. Both of them divide the task into smaller sequential steps. In our case, we alleviate the burden of G by incorporating some well-defined image processing operations into the network model, *e.g.*, converting background color into grayscale to simulate the visual effect of `color selectivo`, or blurring the background to create the `Bokeh` effect.

Computer vision problems may benefit from GAN by including an adversarial loss into, say, a typical CNN model. Many intricate tasks have been shown to gain further improvements after adding adversarial loss, such as shadow detection (Nguyen et al. 2017), saliency detection (Pan et al. 2017), and semantic segmentation (Luc et al. 2016). However, those training methodologies require paired images (with ground-truth) and hence lack the advantage of

Figure 2: Learning and applying our model for the case of "color selectivo" visual effect. The image collection for learning is downloaded using Flickr API. Without explicit ground-truth pixel-level annotations being provided, our method can learn to estimate the visual-effect representations (VERs) from *unpaired* sets of natural RGB images and sample images with the expected visual effect. Our generative model is called Visual-Effect GAN (VEGAN), which has an additional component *editor* between the generator and the discriminator. After the unsupervised learning, the generator is able to predict the VER of an input color image for creating the expected visual effect. The VER can be further transformed into figure-ground segmentation.

unsupervised learning. For the applications of modifying photo styles, some methods (Liu, Breuel, and Kautz 2017; Yi et al. 2017; Zhu et al. 2017) can successfully achieve image-to-image style transfer using unpaired data, but their results are limited to subjective evaluation. Moreover, those style-transfer methods cannot be directly applied to the task of unsupervised segmentation.

Since our model has to identify the category-independent subjects for applying the visual effect without using image-pair relations and ground-truth pixel-level annotations, the problem we aim to address is more general and challenging than those of the aforementioned methods.

Image Segmentation

Most of the existing segmentation methods that are based on deep neural networks (DNNs) to treat the segmentation problem as a pixel-level classification problem (Simonyan and Zisserman 2015; Long, Shelhamer, and Darrell 2015; He et al. 2016). The impressive performance relies on a large number of high-quality annotations. Unfortunately, collecting high-quality annotations at a large scale is another challenging task since it is exceedingly labor-intensive. As a result, existing datasets just provide limited-class and limited-annotation data for training DNNs. DNN-based segmentation methods thus can only be applied to a limited subset of category-dependent segmentation tasks.

To reduce the dependency of detailed annotations and to simplify the way of acquiring a sufficient number of training data, a possible solution is to train DNNs in a semi-supervised manner (Hong, Noh, and Han 2015; Souly, Spampinato, and Shah 2017) or a weakly-supervised manner (Dai, He, and Sun 2015; Kwak, Hong, and Han 2017; Pinheiro and Collobert 2015) with a small number of pixel-level annotations. In contrast, our model is trained without explicit ground-truth annotations.

Existing GAN-based segmentation methods (Nguyen et al. 2017; Luc et al. 2016) improve their segmentation performance using mainly the adversarial mechanism of GANs. The ground-truth annotations are needed in their training process for constructing the adversarial loss, and therefore they are GAN-based but not "unsupervised" from the perspective of application and problem definition.

We instead adopt a meta-learning viewpoint to address figure-ground segmentation. Depending on the visual effect

to be imitated, the proposed approach interprets the task of image segmentation according to the learned VER. As a result, our model indeed establishes a general setting of figure-ground segmentation, with the additional advantage of generating visual effects or photo-style manipulations.

Our Method

Given a natural RGB image I and an expected visual effect with known compositional process and local operation, the proposed VEGAN model learns to predict the visual-effect representation (VER) of I and to generate an edited image I_{edit} with the expected effect. Fig. 2 illustrates the core idea. The training data are from two *unpaired* sets: the set $\{I\}$ of original RGB images and the set $\{I_{\text{sample}}\}$ of images with the expected visual effect. The learning process is carried out as follows: i) *Generator* predicts the VER ν of the image I. ii) *Editor* uses the known local operation to create an edited image I_{edit} possessing the expected visual effect. iii) *Discriminator* judges the quality of the edited images I_{edit} with respect to a set $\{I_{\text{sample}}\}$ of sample images that exhibit the same visual effect. iv) *Loss* is computed for updating the whole model. Fig. 3 illustrates the components of VEGAN. Finally, we perform *Binarization* on VER for quantitatively assess the outcome of figure-ground segmentation.

Generator: The task of the generator is to predict the VER ν that can be used to partition the input image I into foreground and background layers. Our network architecture is adapted from the state-of-the-art methods (Johnson, Alahi, and Fei-Fei 2016; Zhu et al. 2017) which show impressive results on image style transfer. The architecture follows the rules suggested by DCGAN (Radford, Metz, and Chintala 2016) such as replacing pooling layer with strided convolution. Our base architecture also uses the 9-residual-blocks version of (Johnson, Alahi, and Fei-Fei 2016). We have also tried a few slightly modified versions of the generator. The differences and details are described in the experiments.

Discriminator: The discriminator is trained to judge the quality of the edited images I_{edit} with respect to a set $\{I_{\text{sample}}\}$ of sample images that exhibit the same effect. We adopt a 70×70 patchGAN (Isola et al. 2017; Ledig et al.
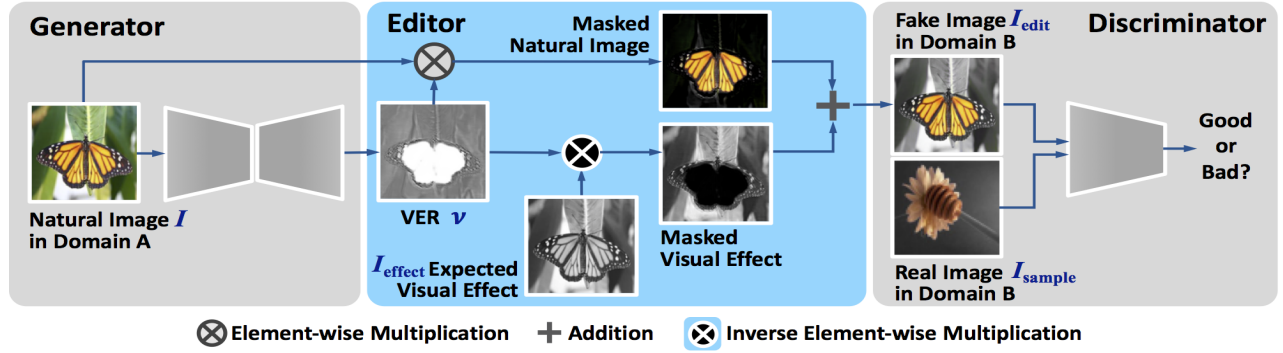
Figure 3: The proposed Visual-Effect GAN (VEGAN) model. Here we take `color selectivo` as the expected visual effect. The visual-effect representation (VER) produced by the generator indicates the strength of the visual effect at each location. The *editor* uses a well-defined trainable procedure (converting RGB to grayscale in this case) to create the expected visual effect. The discriminator receives the edited image I_{edit} and evaluates how good it is. To train VEGAN, we need unpaired images from two domains. Domain A comprises real RGB images and Domain B comprises images with the expected visual effect.

2017; Li and Wand 2016; Zhu et al. 2017) as our base discriminator network. PatcahGAN brings some benefits with multiple overlapping image patches. Namely, the scores change more smoothly and the training process is more stable. Compared with a full-image discriminator, the receptive field of the 70×70 patchGAN might not capture the global context. In our work, the foreground objects are sensitive to their position in the whole image and are center-biased. If there are several objects in the image, our method would favor to pick out the object closest to the center. In our experiment, 70×70 patchGAN does produce better segments along the edges, but sometimes the segments tend to be tattered. A full-image discriminator (Goodfellow et al. 2014; Radford, Metz, and Chintala 2016; Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017), on the other hand, could give coarser but more compact and structural segments.

Editor: The editor is the core of the proposed model. Given an input image I and its VER ν predicted by the generator, the editor is responsible for creating a composed image I_{edit} containing the expected visual effect. The first step is based on the well-defined procedure to perform local operations on the image and generate the expected visual effect I_{effect}. More specifically, in our experiments we define three basic local operations: *black-background*, *color-selectivo*, and *defocus/Bokeh*, which involve *clamping-to-zero*, *grayscale conversion*, and 11×11 *average pooling*, respectively. The next step is to combine the edited background layer with the foreground layer to get the final editing result I_{edit}. An intuitive way is to use the VER ν as an alpha map $\boldsymbol{\alpha}$ for image matting, *i.e.*, $I_{\text{edit}} = \boldsymbol{\alpha} \otimes I + (1 - \boldsymbol{\alpha}) \otimes I_{\text{effect}}$, where $\boldsymbol{\alpha} = \{\alpha_{ij}\}$, $\alpha_{ij} \in (0, 1)$ and $\otimes$ denotes the element-wise multiplication. However, in our experiments, we find that it is better to have $\boldsymbol{\nu} = \{\nu_{ij}\}$, $\nu_{ij} \in (-1, 1)$ with hyperbolic-tangent as the output. Hence we combine the two layers as follows:

$$I_{\text{edit}} = \tau(\boldsymbol{\nu} \otimes (I - I_{\text{effect}}) + I_{\text{effect}}), \quad \nu_{ij} \in (-1, 1),$$
$$(1)$$

where $\tau(\cdot)$ truncates the values to be within $(0, 255)$, which guarantees the I_{edit} can be properly rendered. Under this formulation, our model turns to learning the residual.

Loss: We refer to SOTA algorithms (Arjovsky, Chintala, and Bottou 2017; Gulrajani et al. 2017) to design loss functions $\mathcal{L}_G$ and $\mathcal{L}_D$ for generator (G) and discriminator (D):

$$\mathcal{L}_G = -\mathbb{E}_{x \sim \mathbb{P}_g}[D(x)], \qquad (2)$$
$$\mathcal{L}_D = \mathbb{E}_{x \sim \mathbb{P}_g}[D(x)] - \mathbb{E}_{y \sim \mathbb{P}_r}[D(y)]$$
$$+ \lambda_{gp} \, \mathbb{E}_{\hat{x} \sim \mathbb{P}_{\hat{x}}}[(\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1)^2]. \qquad (3)$$

We alternately update the generator by Eq. 2 and the discriminator by Eq. 3. In our formulation, x is the edited image I_{edit}, y is an image I_{sample} which exhibits the expected visual effect, $\mathbb{P}_g$ is the edited image distribution, $\mathbb{P}_r$ is the sample image distribution, and $\mathbb{P}_{\hat{x}}$ is for sampling uniformly along straight lines between image pairs from $\mathbb{P}_g$ and $\mathbb{P}_r$. We set the learning rate, λ_{gp}, and other hyper-parameters the same as the configuration of WGAN-GP (Gulrajani et al. 2017). We keep the history of previously generated images and update the discriminator according to the history. We use the same way as (Zhu et al. 2017) to store 50 previously generated images $\{I_{\text{edit}}\}$ in a buffer. The training images are of size 224×224, and the batch size is 1.

Binarization: The VEGAN model can be treated as aiming to predict the strength of the visual effect throughout the whole image. Although the VER provides effective intermediate representation for generating plausible edited images toward some expected visual effects, we observe that sometimes the VER might not be consistent with an object region, particularly with the `Bokeh` effect. Directly thresholding VER to make a binary mask for segmentation evaluation will cause some degree of false positives and degrade the segmentation quality. In general, we expect that the segmentation derived from the visual-effect representation to be

smooth within an object and distinct across object boundaries. To respect this observation, we describe, in what follows, an optional procedure to obtain a smoothed VER and enable simple thresholding to yield a good binary mask for quantitative evaluation. Notice that all the VER maps visualized in this paper are obtained without binarization.

To begin with, we over-segment (Achanta et al. 2012) an input image I into a superpixel set $\mathcal{S}$ and construct the corresponding superpixel-level graph $\mathcal{G} = (\mathcal{S}, \mathcal{E}, \omega)$ with the edge set $\mathcal{E}$ and weights ω. Each edge $e_{ij} \in \mathcal{E}$ denotes the spatial adjacency between superpixels s_i and s_j. The weighting function $\omega : \mathcal{E} \rightarrow [0, 1]$ is defined as $\omega_{ij} = e^{-\theta_1 \|c_i - c_j\|}$, where c_i and c_j respectively denote the CIE Lab mean colors of two adjacent superpixels. Then the weight matrix of the graph is $\mathbf{W} = [\omega_{ij}]_{|\mathcal{S}| \times |\mathcal{S}|}$.

We then smooth the VER via propagating the averaged value of each superpixel to all other superpixels. To this end, we use r_i to denote the mean VER value of superpixel s_i where $r_i = \frac{1}{|s_i|} \sum_{(i,j) \in s_i} \nu_{ij}$ and $|s_i|$ is the number of pixels within s_i. The propagation is carried out according to the feature similarity between every superpixel pair. Given the weight matrix $\mathbf{W}$, the pairwise similarity matrix $\mathbf{A}$ can be constructed as $\mathbf{A} = (\mathbf{D} - \theta_2 \mathbf{W})^{-1} \mathbf{I}$, where $\mathbf{D}$ is a diagonal matrix with each diagonal entry equal to the row sum of $\mathbf{W}$, θ_2 is a parameter in $(0, 1]$, and $\mathbf{I}$ is the $|\mathcal{S}|$-by-$|\mathcal{S}|$ identity matrix (Zhou et al. 2003). Finally, the smoothed VER value of each superpixel can be obtained by

$$[\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_{|\mathcal{S}|}]^T = \mathbf{D}_{\mathbf{A}}^{-1} \mathbf{A} \cdot [r_1, r_2, \ldots, r_{|\mathcal{S}|}]^T, \quad (4)$$

where $\mathbf{D}_{\mathbf{A}}$ is a diagonal matrix with each diagonal entry equal to the corresponding row sum of $\mathbf{A}$, and $\mathbf{D}_{\mathbf{A}}^{-1} \mathbf{A}$ yields the row normalized version of $\mathbf{A}$. From Eq. 4, we see that the smoothed VER value $\hat{r}_i$ is determined by not only neighboring superpixels of s_i but also all other superpixels.

To obtain the binary mask, we set the average value of $\{\hat{r}_1, \hat{r}_2, \ldots, \hat{r}_{|\mathcal{S}|}\}$ as the threshold for obtaining the corresponding figure-ground segmentation for the input I. We set parameters $\theta_1 = 10$ and $\theta_2 = 0.99$ in all the experiments.

Experiments

We first describe the evaluation metric, the testing datasets, the training data, and the algorithms in comparison. Then, we show the comparison results of the relevant algorithms and our approach. Finally, we present the image segmentation and editing results of our approach. More experimental results can be found in the supplementary material.

Evaluation Metric. We adopt the intersection-over-union (IoU) to evaluate the binary mask derived from the VER. The IoU score, which is defined as $\frac{|P \bigcap Q|}{|P \bigcup Q|}$, where P denotes the machine segmentation and Q denotes the ground-truth segmentation. All algorithms are tested on Intel i7-4770 3.40 GHz CPU, 8GB RAM, and NVIDIA Titan X GPU.

Datasets. The six datasets are GC50 (Rother, Kolmogorov, and Blake 2004), MSRA500, ECSSD (Shi et al. 2016), Flower17 (Nilsback and Zisserman 2006), Flower102 (Nilsback and Zisserman 2008), and CUB200 (Wah et al. 2011). MSRA500 is a subset of the MSRA10K dataset (Cheng et al. 2015), which contains 10,000 natural images. We randomly

Table 1: Testing datasets and number of images.

GC50	MSRA500	ECSSD	Flower17	Flower102	CUB200
50	500	1,000	1,360	8,189	11,788

partition MSRA10K into two non-overlapping subsets of 500 and 9,500 images to create MSRA500 and MSRA9500 for testing and training, respectively. Their statistics are summarized in Table 1. Since these datasets provide pixel-level ground truths, we can compare the consistency between the ground-truth labeling and the derived segmentation of each image for VER-quality assessment.

Training Data. In training the VEGAN model, we consider using the images from two different sources for comparison. The first image source is MSRA9500 derived from the MSRA10K dataset (Cheng et al. 2015). The second image source is Flickr, and we acquire unorganized images for each task as the training data. We examine our model on three kinds of visual effects, namely, `black background`, `color selectivo`, and `defocus/Bokeh`.

- For MSRA9500 images, we randomly select 4,750 images and then apply the three visual effects to yield three groups of *images with visual effects*, *i.e.*, $\{I_{\mathrm{sample}}\}$. The other 4,750 images are hence the input images $\{I\}$ for the generator to produce the *edited images* $\{I_{\mathrm{edit}}\}$ later.

- For Flickr images, we use "black background," "color selectivo," and "defocus/Bokeh" as the three query tags, and then collect 4,000 images for each query-tag as the *real images with visual effects*. We randomly download additional 4,000 images from Flickr as the images to be edited.

Algorithms in Comparison. We quantitatively evaluate the learned VER using the standard segmentation assessment metric (IoU). Our approach is compared with several well-known algorithms, including two semantic segmentation algorithms, three saliency based algorithms, and two bounding-box based algorithms, listed as follows: ResNet (He et al. 2016), VGG16 (Simonyan and Zisserman 2015), CA (Qin et al. 2015), MST (Tu et al. 2016), GBMR (Yang et al. 2013), MilCutS and MilCutG (Wu et al. 2014), GrabCut (Rother, Kolmogorov, and Blake 2004). The two *supervised* semantic segmentation algorithms, ResNet and VGG16, are pre-trained on ILSVRC-2012-CLS (Russakovsky et al. 2015) and then fine-tuned on MSRA9500 with ground-truth annotations. The bounding boxes of the two bounding-box based algorithms are initialized around the image borders.

Quantitative Evaluation

The first part of experiment aims to evaluate the segmentation quality of different methods. We first compare several variants of the VEGAN model to choose the best model configuration. Then, we analyze the results of the VEGAN model versus the other state-of-the-art algorithms.

VEGAN Variants. In the legend blocks of Fig. 4, we use a compound notation "*TrainingData - Version*" to account for the variant versions of our model. Specifically, *TrainingData* indicates the image source of the training data. The notation for *Version* contains two characters. The first

Table 2: Comparison of VEGAN variants. All variants are trained with MSRA9500. Each entry shows the *version* and the *mean IoU score* (in parentheses) of a VEGAN variant.

Visual Effect	Testing Dataset MSRA500 mean IoU			
Black Background	B1 (0.67)	B2 (0.73)	B3 (0.70)	**B4 (0.76)**
Color Selectivo	C1 (0.73)	C2 (0.73)	C3 (0.74)	C4 (0.75)
Defocus/Bokeh	D1 (0.70)	D2 (0.66)	D3 (0.70)	D4 (0.73)

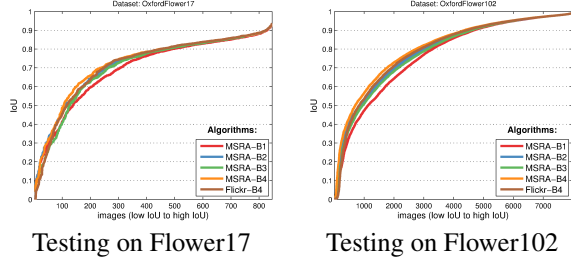Testing on Flower17 · Testing on Flower102

Figure 4: Comparison of two training sources: MSRA9500 and Flickr. Each sub-figure depicts the sorted IoU scores for the variants. Note that Flickr-B4 is trained from the queried Flickr images and performs as well as other variants.
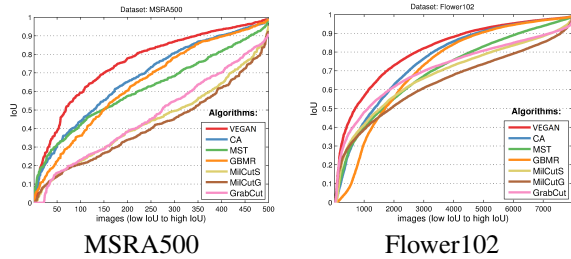

MSRA500 · Flower102

Figure 5: Comparisons among algorithms. Each sub-figure depicts the sorted IoU scores as the segmentation accuracy.

character denotes the type of visual effect: "B" for `black background`, "C" for `color selectivo`, and "D" for `defocus`/Bokeh. The second character is the model configuration: "1" refers to the combination of base-generator and base-discriminator described in **Our Method**; "2" refers to using ResNet as the generator; "3" is the model "1" with additional skip-layers and replacing transpose convolution with bilinear interpolation; "4" is the model "3" yet replacing patch-based discriminator with full-image discriminator.

We report the results of VEGAN variants in Table 2, and depict the sorted IoU scores for the test images in Flower17 and Flower102 datasets in Fig. 4. It can be seen that all models have similar segmentation qualities no matter what image source is used for training. In Table 2 and Fig. 4, the training configuration "B4" shows relatively better performance under `black background`. Hence, our VEGAN model adopts the version of MSRA-B4 as a representative variant for comparing with other state-of-the-art algorithms.

Unseen Images. We further analyze the differences of the learned models on dealing with *unseen* and *seen* images. We test the variants B4, C4, and D4 on MSRA500 (un-

seen) and the subset $\{I\}$ of MSRA9500 (seen). We find that the performance of VEGAN is quite stable. The IoU score for MSRA500 is only 0.01 lower than the score for MSRA9500 $\{I\}$. Note that, even for the seen images, the ground-truth pixel annotations are unknown to the VEGAN model during training. This result indicates that VEGAN has a good generalization ability to predict segmentation for either seen or unseen images. For comparison, we do the same experiment with the two supervised algorithms, ResNet and VGG16. They are fine-tuned with MSRA9500. The mean IoU scores of ResNet are 0.86 and 0.94 for MSRA500 and MSRA9500, respectively. The mean IoU scores of VGG16 are 0.72 and 0.88 for MSRA500 and MSRA9500, respectively. The performance of both supervised techniques significantly degrades while dealing with unseen images.

From the results just described, the final VEGAN model is implemented with the following setting: *i*) *Generator* uses the 9-residual-blocks version of (Johnson, Alahi, and Fei-Fei 2016). *ii*) *Discriminator* uses the full-image discriminator as WGAN-GP (Gulrajani et al. 2017).

Results. The top portion of Table 3 summarizes the mean IoU score of each algorithm evaluated with the six testing datasets. We first compare our method with five well-known segmentation/saliency-detection techniques, including CA (Qin et al. 2015), MST (Tu et al. 2016), GBMR (Yang et al. 2013), MilCutS/MilCutG (Wu et al. 2014), and GrabCut (Rother, Kolmogorov, and Blake 2004). The proposed VEGAN model outperforms all others on MSRA500, ECSSD, Flower17, and Flower102 datasets, and is only slightly behind the best on GC50 and CUB200 datasets.

The bottom portion of Table 3 shows the results of two SOTA supervised learning algorithms on the six testing datasets. Owing to training with the paired images and ground-truths in a "supervised" manner, the two models of ResNet and VGG16 undoubtedly achieve good performance so that we treat them as the oracle models. Surprisingly, our unsupervised learning model is comparable with or even slightly better than the supervised learning algorithms on the MSRA500, Flower17, and Flower102 datasets.

Fig. 5 depicts the sorted IoU scores, where a larger area under curve means better segmentation quality. VEGAN achieves better segmentation accuracy on the two datasets.

Qualitative Evaluation

Fig. 6 shows the results generated by our VEGAN model under different configurations. Each triplet of images contains the input image, the visual effect representation (VER), and the edited image. The results in Fig. 6 demonstrate that VEGAN can generate reasonable figure-ground segmentations and plausible edited images with expected visual effects.

Visual-Effect Imitation as Style Transfer. Although existing GAN models cannot be directly applied to learning figure-ground segmentation, some of them are applicable to learning visual-effect transfer, *e.g.*, CycleGAN (Zhu et al. 2017). We use the two sets $\{I\}$ and $\{I_{sample}\}$ of MSRA9500 to train CycleGAN, and show some comparison results in Fig. 7. We find that the task of imitating `black background` turns out to be challenging for CycleGAN since the information in $\{I_{sample}\}$ is too limited to derive the

Table 3: The mean IoU scores of each algorithm on the six datasets. The VEGAN model uses configuration "MSRA-B4" for comparison. Notice that the "supervised learning algorithms" ResNet (He et al. 2016) and VGG16 (Simonyan and Zisserman 2015) are pre-trained with ILSVRC-2012-CLS and then fine-tuned with MSRA9500.

		Testing Dataset—Mean IoU Score					
		GC50	MSRA500	ECSSD	Flower17	Flower102	CUB200
Algorithm	VEGAN	0.58	**0.76**	**0.58**	**0.72**	**0.81**	0.52
	CA	0.59	0.67	0.50	**0.72**	0.75	0.51
	MST	**0.60**	0.61	0.53	0.68	0.70	**0.54**
	GBMR	0.52	0.64	0.48	0.68	0.71	0.49
	MilCutS	0.54	0.43	0.48	0.67	0.67	0.43
	MilCutG	0.50	0.41	0.46	0.64	0.63	0.41
	GrabCut	0.51	0.46	0.45	0.68	0.71	0.39
Oracle	ResNet	0.70	0.86	0.67	0.72	0.79	0.60
	VGG16	0.68	0.72	0.63	0.73	0.80	0.57

Table 4: Percentage of preferring the VEGAN results.

26 participants	mean	median
Black Background	90.70%	95.00%
Color Selectivo	80.30%	80.00%
Defocus/Bokeh	75.75%	80.00%

inverse mapping back to $\{I\}$. Moreover, CycleGAN focuses more on learning the mapping between local properties such as color or texture rather than learning how to create a globally consistent visual effect. VEGAN instead follows a systematic learning procedure to imitate the visual effect. The generator must produce a meaningful VER so that the editor can compose a plausible visual-effect image that does not contain noticeable artifacts for the discriminator to identify.
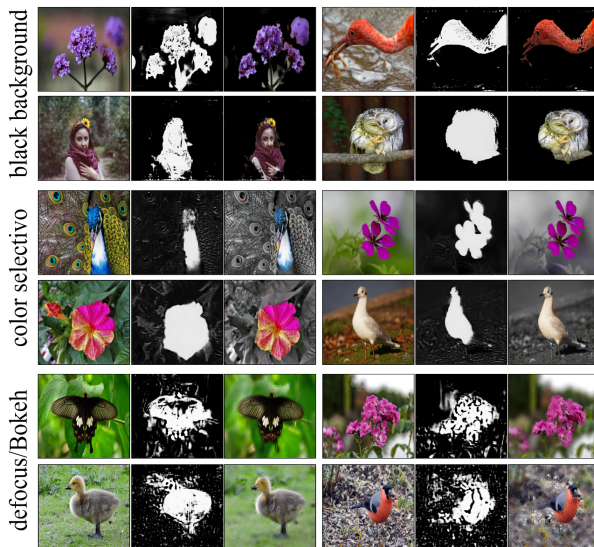

Figure 6: The edited images generated by VEGAN with respect to specific visual effects. Each image triplet from left to right: the input image, the VER, and the edited image.

Black Background   Color Selectivo

Figure 7: The edited images by VEGAN and CycleGAN. CycleGAN is trained with MSRA9500. The task of imitating `black background` is challenging for CycleGAN.
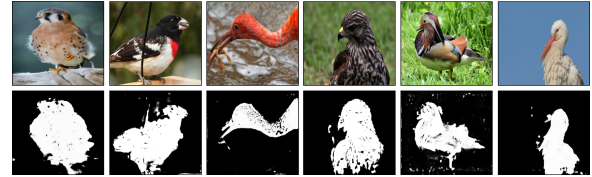

Figure 8: Testing on Flickr "bird" images using VEGAN model trained with Flickr "flower" images. The first row shows the input images. The second rows shows the VERs.

User Study. Table 4 lists the results from 26 participants. The survey is presented in Google forms, comprising 40 edited image pairs (*e.g*., as in Fig. 7) for each visual effect. Each user is asked to select the preferred one from an image pair of random order by CycleGAN and VEGAN.

Unseen Figures. To demonstrate that the VEGAN model can learn the general concept of figure-ground segmentation and thus handle unseen foreground "figures." Fig. 8 shows VERs that testing on Flickr "bird" images using VEGAN models trained merely with Flick "flower" images. The results suggest that the meta-learning mechanism enables VEGAN to identify unseen foreground figures based on the learned knowledge embodied in the generated VERs.

Conclusion

We characterize the two main contributions of our method as follows. First, we establish a meta-learning framework to learn a general concept of figure-ground application and an effective approach to the segmentation task. Second, we propose to cast the meta-learning as imitating relevant visual effects and develop a novel VEGAN model with following advantages: *i*) Our model offers a new way to predict meaningful figure-ground segmentation from unorganized images that have no explicit pixel-level annotations. *ii*) The training images are easy to collect from photo-sharing websites using related tags. *iii*) The *editor* between the generator and the discriminator enables VEGAN to decouple the compositional process of imitating visual effects and hence allows VEGAN to effectively learn the underlying representation (VER) for deriving figure-ground segmentation. We have tested three visual effects, including "black background," "color selectivo," and "defocus/Bokeh" with extensive experiments on six datasets. For these visual effects, VEGAN can be end-to-end trained from scratch using unpaired training images that have no ground-truth labeling.

References

Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; and Süsstrunk, S. 2012. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.* 34(11):2274–2282.

Arjovsky, M.; Chintala, S.; and Bottou, L. 2017. Wasserstein generative adversarial networks. In *ICML*, 214–223.

Cheng, M.; Mitra, N. J.; Huang, X.; Torr, P. H. S.; and Hu, S. 2015. Global contrast based salient region detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 37(3):569–582.

Dai, J.; He, K.; and Sun, J. 2015. Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 1635–1643.

Denton, E. L.; Chintala, S.; Szlam, A.; and Fergus, R. 2015. Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, 1486–1494.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial nets. In *NIPS*, 2672–2680.

Gulrajani, I.; Ahmed, F.; Arjovsky, M.; Dumoulin, V.; and Courville, A. C. 2017. Improved training of wasserstein gans. In *NIPS*, 5769–5779.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

Hong, S.; Noh, H.; and Han, B. 2015. Decoupled deep neural network for semi-supervised semantic segmentation. In *NIPS*, 1495–1503.

Isola, P.; Zhu, J.; Zhou, T.; and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*, 5967–5976.

Johnson, J.; Alahi, A.; and Fei-Fei, L. 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 694–711.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2018. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*.

Kwak, S.; Hong, S.; and Han, B. 2017. Weakly supervised semantic segmentation using superpixel pooling network. In *AAAI*, 4111–4117.

Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A. P.; Tejani, A.; Totz, J.; Wang, Z.; and Shi, W. 2017. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*.

Li, C., and Wand, M. 2016. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, 702–716.

Liu, M.; Breuel, T.; and Kautz, J. 2017. Unsupervised image-to-image translation networks. In *NIPS*, 700–708.

Long, J.; Shelhamer, E.; and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In *CVPR*.

Luc, P.; Couprie, C.; Chintala, S.; and Verbeek, J. 2016. Semantic segmentation using adversarial networks. In *NIPSW*.

Nguyen, V.; Vicente, T. F. Y.; Zhao, M.; Hoai, M.; and Samaras, D. 2017. Shadow detection with conditional generative adversarial networks. In *ICCV*, 4520–4528.

Nilsback, M., and Zisserman, A. 2006. A visual vocabulary for flower classification. In *CVPR*, 1447–1454.

Nilsback, M., and Zisserman, A. 2008. Automated flower classification over a large number of classes. In *ICVGIP*.

Pan, J.; Canton-Ferrer, C.; McGuinness, K.; O'Connor, N. E.; Torres, J.; Sayrol, E.; and Giró i Nieto, X. 2017. Salgan: Visual saliency prediction with generative adversarial networks. In *CVPRW*.

Pinheiro, P. H. O., and Collobert, R. 2015. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 1713–1721.

Qin, Y.; Lu, H.; Xu, Y.; and Wang, H. 2015. Saliency detection via cellular automata. In *CVPR*, 110–119.

Radford, A.; Metz, L.; and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*.

Rother, C.; Kolmogorov, V.; and Blake, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.* 23(3):309–314.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. S.; Berg, A. C.; and Li, F. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115(3):211–252.

Shi, J.; Yan, Q.; Xu, L.; and Jia, J. 2016. Hierarchical image saliency detection on extended CSSD. *IEEE Trans. Pattern Anal. Mach. Intell.* 38(4):717–729.

Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

Souly, N.; Spampinato, C.; and Shah, M. 2017. Semi supervised semantic segmentation using generative adversarial network. In *ICCV*, 5689–5697.

Tu, W.; He, S.; Yang, Q.; and Chien, S. 2016. Real-time salient object detection with a minimum spanning tree. In *CVPR*, 2334–2342.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, Caltech.

Wu, J.; Zhao, Y.; Zhu, J.; Luo, S.; and Tu, Z. 2014. Milcut: A sweeping line multiple instance learning paradigm for interactive image segmentation. In *CVPR*, 256–263.

Yang, C.; Zhang, L.; Lu, H.; Ruan, X.; and Yang, M. 2013. Saliency detection via graph-based manifold ranking. In *CVPR*, 3166–3173.

Yi, Z.; Zhang, H. R.; Tan, P.; and Gong, M. 2017. Dualgan: Unsupervised dual learning for image-to-image translation. In *ICCV*, 2868–2876.

Zhou, D.; Bousquet, O.; Lal, T. N.; Weston, J.; and Schölkopf, B. 2003. Learning with local and global consistency. In *NIPS*, 321–328.

Zhu, J.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2242–2251.