Reducing Noise Pixels and Metric Bias in Semantic Inpainting on Segmentation Map

Jianfeng He[†], Bei Xiao[‡], Xuchao Zhang[†], Shuo Lei[†], Shuhui Wang[+*], Chang-Tien Lu[†]
[†]Sanghani Center for Artificial Intelligence and Data Analytics, Virginia Tech, Falls Church, VA, USA
[‡]Department of Computer Science, American University, Washington, DC, USA
[+] Key Lab of Intell. Info. Process., Inst. of Comput. Tech., CAS, Beijing, China
[†]{jianfenghe, xuczhang, slei, ctlu}@vt.edu, [‡]bxiao@american.edu, [+]wangshuhui@ict.ac.cn

Abstract

Semantic Inpainting on Segmentation Map (SISM) aims to manipulate segmentation maps by semantics. Providing structural assistance, segmentation maps have been broadly used as intermediate interfaces to achieve better image manipulation. We improve the SISM by considering the unique characteristics of segmentation maps in the both training and testing processes.First, to improve SISM training process, we reduce the noise pixels, which are pixel artifacts from the generation. Because each pixel in the segmentation maps has a much smaller value range in comparison to pixels in natural images, we propose a novel denoise activation (DA) by estimating the possible pixel values for an inpainted area in advance. Second, we improve SISM testing process by reducing the metric bias. The bias is caused by the ignore of latent ground truths in the current metrics in SISM. Based on the analysis of possible latent ground truths, we then propose a novel metric, Semantic Similarity (Sem), to quantify the semantic divergence between the generated and ground-truth target objects. Sem is calculated by a pre-trained semantic classifier using object shapes as training data. Since the classifier is pre-trained on PS-COCO dataset, with a large number of training samples and relatively general classes, Sem is also applicable to other datasets. Our experiments show impressive results of DA and Sem on three datasets.

1. Introduction

Image manipulation, aiming to transform or edit images, is a popular topic [13, 9, 44, 23]. Though its final outputs are usually natural images, many recent works have showed segmentation maps provide auxiliary information for better style [24, 48, 44] or structure manipulation. For example, recent works use manually-edited [24, 33], pre-existing [7, 23], or automatically manipulated segmentation maps [13, 17, 27, 21] as the structures of the objects, followed by style manipulation on the edited structure as their downstream models. Since manual editing on segmentation map is time-consuming, and pre-existing segmentation maps are not always adapted to the context, we focus on Semantic Inpainting on Segmentation Maps (SISM) [13, 10], which automatically manipulates segmentation maps. Concretely, given a bounding box, which provides semantics and location of a target object, SISM outputs an inpainted segmentation map respected to the bounding box (inpainted area), shown as Fig. 1. The SISM results should be consistent with the context and reflect the given semantics, which is different from image inpainting [3, 38].

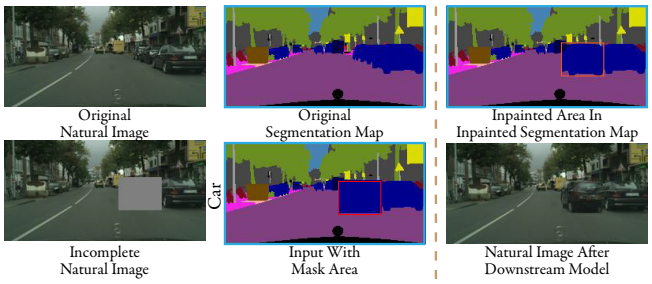

Figure 1. Diagram of SISM and its application in semantic image inpainting [13]. In SISM (figures in blue circumscribed rectangles), given an original segmentation map, users set a mask area (red rectangle) defined by a bounding box with a target label, such as "car". Then, a SISM model generates an inpainted area (orange rectangle) with a "car" in the inpainted segmentation map, which is the SISM result. Then, a downstream model (e.g. a translation model) transfers its result into a manipulated natural image.

Current SISM models [13, 10] are adapted from models of natural image generation, replacing the natural image representations with the segmentation map representations. But they ignore the unique characteristics of segmentation maps, which are helpful to SISM. In this work, we exploit

* Corresponding author.

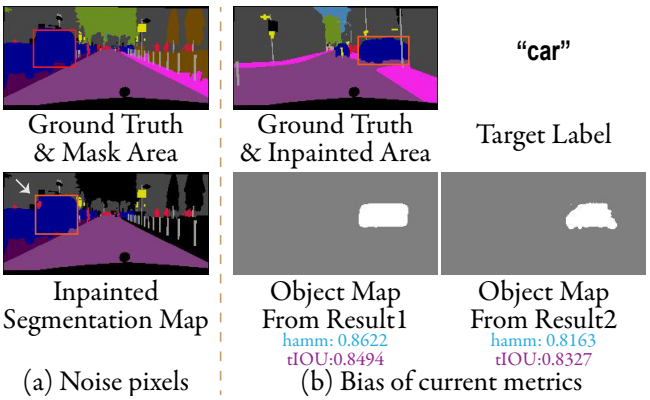

Figure 2. Panel (a) shows the noise pixels in an inpainted segmentation map, where the mask area and inpainted area are represented by red and orange rectangles respectively. The noise pixels are pointed by the white arrow. In (b), we show the binary target object maps extracted from inpainted segmentation maps with their hamm and tIOU scores. The metrics show the object map from result1 is better, but the object map from result2 looks better, which is closer to "car", the required sementic class. This shows that current metrics of SISM are biased.

the characteristics of segmentation maps to improve SISM in the both training and testing processes by reducing the noise pixels and reducing the metric bias respectively.

First, the noise pixels, which are pixel artifacts from the generation, exist in the SISM results. Since SISM is a generation task, its generated pixel values are in a full range; then, the pixels with values out of the range of ground-truth are noises. An example of noise pixels in a SISM result is shown in Fig. 2(a). The noise pixels represent inconsistent semantics. They can impact downstream models, which translate SISM results to natural images by semantics. Besides, one pixel can even introduce an adversarial attack [28, 1]. Thus, the noise pixels should be reduced.

To reduce them, we use a characteristic of segmentation maps, *smaller value range of each segmentation pixel value*. It means that each pixel in a segmentation map has a much smaller number of possible values compared with that of RGB image [25], because each object in a segmentation map has the same pixel value equaling their semantic classes. As a result, the value range equals the number of semantic classes. Recognizing this, we propose a novel Denoise Activation (DA) to reduce noise pixels by estimating a range of pixel values for an inpainted area in advance. DA is effective in both training and testing processes.

Second, we find that current metrics of SISM can bring bias in evaluating SISM, due to ignoring the latent ground truths. Concretely, current metrics of SISM, such as Hamming Distance (hamm) [10, 22] and Target Intersection-Over-Union (tIOU) [13, 10, 18], consider the pixel-level overlapping ratios between the SISM results and the ground truths. Fig. 2(b) shows an example of a biased evaluation. These metrics are biased because they are commonly applied in discriminative tasks with unique ground truths (e.g. image segmentation [35]), while SISM is a generation task with latent ground truths such that different object shapes for the same semantics. Since the metric bias overlooks SISM models with reasonable results which are different from the ground truths, the metric bias should be reduced.

We first analyze the appearances of latent ground truths to reduce the metric bias. The appearances should have target objects with the same semantics as the ground truths, due to the task requirement of SISM. Since having the same semantics is the key factor of latent ground truths, we propose a new metric, Semantic Similarity (Sem), to quantify semantic divergence between the generated and ground-truth target objects by a semantic classifier. The semantic classifier is designed by another characteristic of segmentation maps, *more object semantics from object shapes*. It means that object semantics are mainly represented by the object shapes in the segmentation maps, rather than the textures (colors). Thus, the semantic classifier should be a shape classifier, emphasizing on the object shapes and ignoring the textures. With Sem, the SISM evaluation considers the latent ground truths. Since the semantic classifier is pre-trained on PS-COCO [16], which provides 332,310 training samples with relatively general semantic classes, Sem, like Frechet Inception Distance (FID) [12], is applicable to other datasets. Our main contributions are:
(1) We consider the unique characteristics of segmentation maps to improve SISM, ignored by current SISM models.
(2) To reduce noise pixels in the SISM results, we propose a novel DA using a characteristic of segmentation maps, which allows us to estimate the possible value ranges of pixels in the inpainted areas in advance. To the best of our knowledge, we are the first to address noise pixels in SISM.
(3) The current metrics of SISM are biased, due to ignoring latent ground truths. Thus, we propose Sem to quantify semantic divergence between the generated and ground-truth target objects. Sem is applicable to other datasets. To the best of our knowledge, we are the first to address the bias.
(4) The experiments on Cityscapes [6], ADE20K [46] and PS-COCO [16], achieved impressive results, such as 7.29% improvement in the tIOU on ADE20K by DA. Sem is more robust in different image transformations, such as decreasing 50% slower after flipping the objects.

2. Related Work

Segmentation map manipulation has recently been applied for image manipulation. For examples, [7, 23, 13] all firstly generate the segmentation maps corresponding to the target semantics, which are then translated to natural images for semantic image manipulation. Similar to the framework in above studies, [17, 27] focus on image inpainting. Plus, [7] adds new instances automatically in scenes by firstly cal-

culating their manipulated segmentation maps. Besides the structure manipulation, [21] transfers object shapes in the segmentation maps for attribute manipulation in the natural images. However, these models manipulate segmentation map ignoring the characteristics of segmentation map and treat segmentation maps like natural images.

Noise can be input of generative adversarial net (GAN) [8] for generation [42, 5, 45]. But most noise in the generated results brings degradation [4] or even adversarial attack [28, 1], which can be achieved by one pixel. To address the noise in the outputs of the GAN, several methods are attempted on the natural images. For examples, [20] applies the cycleGAN [47] to translate high noise images to low noise images. A wavelet filtering is added to a generator for high-fidelity generation [41]. And [14] applies hidden Markov model to denoise the natural images. But little denoise has been done in SISM.

Though several metrics, introduced from image segmentation or image inpainting, have been applied in the SISM, most of them do not quantify the SISM results directly and require additional downstream models. For examples, [13], as a SOTA of SISM, evaluates SISM performance by translating the SISM results into natural images, followed by a segmentation model on the translated natural images for new segmentation maps, then calculating the tIOU between the new segmentation maps and ground truths. [17, 10, 23] and [13], apply FID [12] and Structural Similarity Index [34] respectively, for the manipulated images rather than segmentation maps. Though tIOU [18], and hamm [22] are applied for the manipulated segmentation maps directly [13, 10], they ignore the latent ground truths. Besides, though SISM is a generation task, the metrics [37] for GAN, such as 1-NN accuracy [19], Inception Score (IS) [29], and FID [12], are not applicable. This is because that they emphasize on the image quality rather than semantics, and indirect evaluation on natural images from downstream models can bring new biases (shown in Sec. 4.3). In comparison, our Sem considers latent ground truths and is applicable to other datasets.

3. Model

A work flow of SISM is shown in the first row of the Fig. 3 by black arrows [1]. Given a single-channel complete segmentation map $\mathbf{S}^c \in \mathbb{R}^{H\times W\times 1}$, where H and W represent height and width respectively, SISM aims to synthesize the inpainted segmentation map $\hat{\mathbf{S}} \in \mathbb{R}^{H\times W\times 1}$ with semantics defined by a target label l^t. The $\mathbf{S}^c$ and $\hat{\mathbf{S}}$ have each pixel value representing a semantic class.

Concretely, we set an object bounding box $B = \{\mathbf{b}, l^t\}$, as a combination of box corner coordinates $\mathbf{b} \in \mathbb{R}^4$ and a target label l^t, where each $\mathbf{b}$ provides a mask area and each l^t provides a semantic ID. We then construct an incomplete segmentation map $\mathbf{S}^u \in \mathbb{R}^{H\times W\times 1}$ by copying $\mathbf{S}^c$ and masking its pixels in the bounding box B as l^t, which informs the model about the location and the semantics to generate. Then, a learnable structure generator G^0 generates $\widetilde{\mathbf{P}}$ by $\widetilde{\mathbf{P}} = G^0(\mathbf{S}^u, B)$. The $\widetilde{\mathbf{P}} \in \mathbb{R}^{H\times W\times k}$ represents the probabilities that each pixel belongs to each class, where k is the number of semantic classes. Then, we get a SISM initial result $\widetilde{\mathbf{S}} \in \mathbb{R}^{H\times W\times 1}$ by $\widetilde{\mathbf{S}} = \arg\max \widetilde{\mathbf{P}}$. We further fuse the SISM initial result $\widetilde{\mathbf{S}}$ and $\mathbf{S}^u$ to derive an inpainted segmentation map $\hat{\mathbf{S}}$. Finally, $\hat{\mathbf{S}}$ is sent to a downstream model T to perform a subsequent task such as image translation [10] or semantic image inpainting [13].

In the next subsections, we introduce a replaceable structure generator G^0. Then, we introduce our novel DA for SISM based on dilation operation, which can be applied in both training and testing processes. Finally, we propose Sem to quantify the semantic divergence of SISM results.

3.1. Structure Generator

A structure generator G^0 is designed to inpaint the mask area $\mathbf{M}$ in $\mathbf{S}^u$, where $\mathbf{M} \in \mathbb{R}^{H\times W}$ is a binary matrix specified by $B = \{\mathbf{b}, l^t\}$: $M_{ij} = 1$ for all pixels (i, j) inside the bounding box B. The $\widetilde{\mathbf{S}}$, transformed from $\widetilde{\mathbf{P}}$, should include an object reflecting semantics defined by l^t, and should also achieve consistency between the inpainted area and its surrounding context. Since [13] is a state-of-the-art of SISM that satisfies the two requirements, we apply its SISM model as an example of structure generator G^0, which outputs a binary object map $\widetilde{\mathbf{O}}$ and a context map for adding objects in segmentation maps and SISM, respectively. Its two maps are generated by sharing an embedding module. Since our task is SISM, we only use its context maps as our initial SISM results $\widetilde{\mathbf{S}}$ and ignore its binary object maps $\widetilde{\mathbf{O}}$. The loss function of G^0 is as below,

$$\mathcal{L}_{\mathrm{B}} = \lambda_1 \mathcal{L}_{\mathrm{adv}}(\widetilde{\mathbf{O}}, \mathbf{O}^c) + \lambda_2 \mathcal{L}_{\mathrm{rec}}(\widetilde{\mathbf{O}}, \mathbf{O}^c) + \lambda_3 \mathcal{L}_{\mathrm{rec}}(\widetilde{\mathbf{P}}, \mathbf{P}^c) \tag{1}$$

where $\mathbf{O}^c$ is the ground-truth binary object map defined by B. The $\mathbf{P}^c \in \mathbb{R}^{H\times W\times k}$ is a binary tensor transformed from $\mathbf{S}^c$, by setting an element in a channel to 1 if the pixel in $\mathbf{S}^c$ belongs to respective semantic class, otherwise to 0. The $\mathcal{L}_{\mathrm{rec}}(\bullet, \bullet)$ is a reconstruction loss. The $\mathcal{L}_{\mathrm{adv}}(\widetilde{\mathbf{O}}, \mathbf{O}^c)$ is a conditional adversarial loss defined on $\mathbf{S}^u$ and $\mathbf{O}^c$ to ensure the perceptual quality of $\widetilde{\mathbf{O}}$. After deriving $\widetilde{\mathbf{P}}$ from G^0, we get $\widetilde{\mathbf{S}}$ by $\arg\max$. Finally, we have the SISM result, the inpainted segmentation map $\hat{\mathbf{S}}$, by fusing $\widetilde{\mathbf{S}}$ and $\mathbf{S}^u$ as,

$$\hat{\mathbf{S}} = \widetilde{\mathbf{S}} \cdot \mathbf{M} + \mathbf{S}^u \cdot (\mathbf{1} - \mathbf{M}) \tag{2}$$

where $\mathbf{1} \in \mathbb{R}^{H\times W}$ is a matrix with all elements as 1, and $\cdot$ is dot product. From Eq. 2, $\hat{\mathbf{S}}$ only replaces the mask area with the inpainted area in $\widetilde{\mathbf{S}}$, but keeps the rest same as $\mathbf{S}^u$.

[1] We apply bold lowercases and bold capitals to represent vectors and matrices (including tensors) respectively. Examples of the most notations are drawn in the Fig. 3 for better reading.

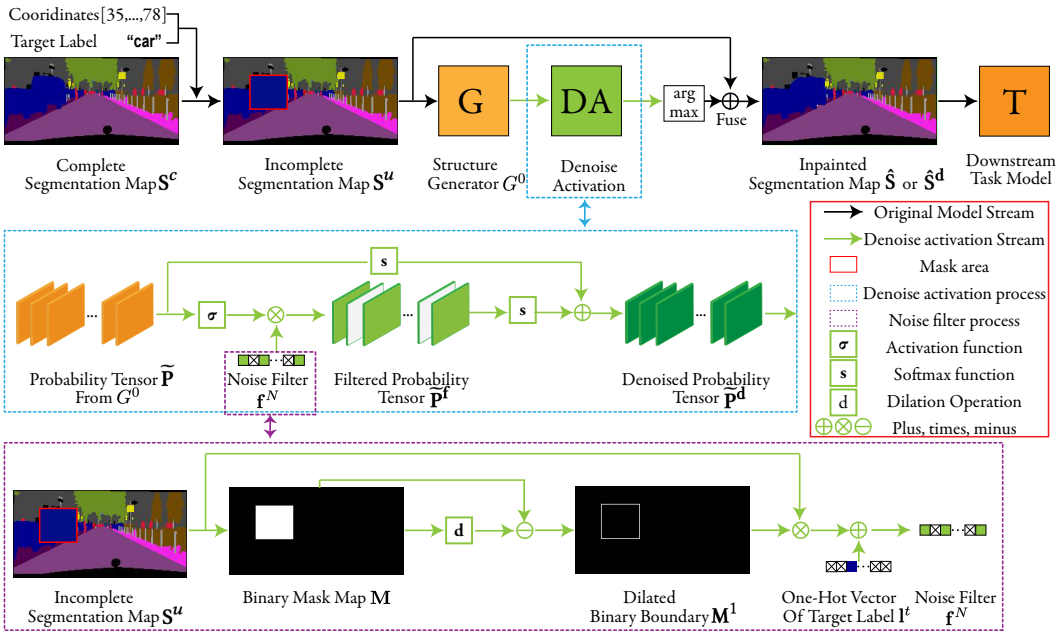

Figure 3. Diagram of work flow of SISM and our DA module, where legends are shown in the red rectangle. The first, second, and third rows respectively show the work flow of SISM with DA module, the work flow of DA module, and work flow of noise filter $\mathbf{f}^N$, which is a key part in DA. Though $\mathbf{S}^c$, $\mathbf{S}^u$ and $\hat{\mathbf{S}}$ are drawn in color for better visualization, they are actually in size of a single-channel, as described in Sec. 3. Each modules in the diagram are labeled by their names and symbols. Some symbols applied in the paper are not shown.

3.2. Denoise Activation (DA)

Because of no limitation on the value ranges of elements in $\widetilde{\mathbf{P}}$, the probabilities for semantic classes that never shown in the ground truths can be the maximum. Since an initial segmentation map $\widetilde{\mathbf{S}}$ is transformed from an probability tensor $\widetilde{\mathbf{P}}$ by $\arg\max$, the SISM initial result $\widetilde{\mathbf{S}}$ can have noise pixels, so as $\hat{\mathbf{S}}$ due to Eq. 2.

To decrease the number of noise pixels, we design a novel DA by a dilation operation [32] without impacting the original performance, illustrated in the second and third rows of Fig. 3. Concretely, we apply a 3×3 kernel with all elements as 1 to dilate the binary mask map $\mathbf{M}$, so that we have the dilated mask map $\mathbf{M}^d$. Due to the kernel size as 3×3, the $\mathbf{M}^d$ is one-pixel dilated compared with $\mathbf{M}$. Then, we get a dilated binary boundary $\mathbf{M}^{\mathbb{1}} \in \mathbb{R}^{H \times W}$ as follow,

$$\mathbf{M}^{\mathbb{1}} = \mathbf{M}^d - \mathbf{M} \tag{3}$$

we calculate $\mathbf{M}^{\mathbb{1}}$, as it is the closest boundary to the mask area. Then, we assume that the pixel values of the context nearby the mask area tend to appear in the inpainted area of $\hat{\mathbf{S}}$ again, while the pixel values never shown in the context nearby the mask area tend to be noises. The assumption is reasonable to segmentation maps, as a bounding box cuts related objects into two parts, and the pixel values among one object in segmentation maps are same as the semantic ID of the object. Then, in the vast cases [2], a set of semantic classes (except the target class l^t) in the inpainted area is a subset of the set of semantic classes in the nearby context. By the assumptions, our noise filter vector $\mathbf{f}^N \in \mathbb{R}^k$ is,

$$\mathbf{f}^N = \min(\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{P}^u \cdot \mathbf{M}^{\mathbb{1}}) + \mathbf{l}^t, 1) \tag{4}$$

where $\mathbf{P}^u \in \mathbb{R}^{H \times W \times k}$ is transformed from $\mathbf{S}^u$ like from $\mathbf{S}^c$ to $\mathbf{P}^c$. And $\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{P}^u \times \mathbf{M}^{\mathbb{1}})$ is a vector in k dimensions, which is the sum of the elements in $\mathbf{P}^u$ with respect to dilated binary boundary $\mathbf{M}^{\mathbb{1}}$ in each semantic channel. It is a possible value range (except target label) of pixels in an inpainted areas. Then, we add it with $\mathbf{l}^t$, which is the one-hot vector of l^t and adds the target label into the possible range. If a vector $\sum_{i=1}^{H}\sum_{j=1}^{W}(\mathbf{P}^u \times \mathbf{M}^{\mathbb{1}}) + \mathbf{l}^t$ has values greater than 0 in some dimensions, it means the respective classes tend to appear in the inpainted area of SISM result again; if the values equal to 0, the respective classes tend to not appear. Thus, we apply a $\min$ operation to indicate whether these classes appear or not by 1 or 0, respectively. For example, if we have a $\mathbf{f}^N = [1, 0, 0, 0, 1, 0]$, it means the first and fifth classes are expected to show in the inpainted area of $\hat{\mathbf{S}}$, while the other four classes are not expected.

Apply DA in the training process. After getting $\mathbf{f}^N$, we implement DA in the training shown in the second row of

[2]The failure cases for our DA are the pixels with unique semantic classes compared to the context and given semantics. Our DA misses them, as the SISM task setting cannot aware these classes.

Fig. 3. It calculates a filtered probability tensor $\widetilde{\mathbf{P}}^{\mathbf{f}}$ as,

$$\widetilde{\mathbf{P}}^{\mathbf{f}} = \sigma(\widetilde{\mathbf{P}}) \cdot \mathbf{f}^N \tag{5}$$

where σ is leakyReLU [36]. Then, motivated by residual learning in resNet [11] and SAGAN [42], we derive our denoised probability tensor $\widetilde{\mathbf{P}}^{\mathbf{d}} \in \mathbb{R}^{H\times W\times k}$ as follows,

$$\widetilde{\mathbf{P}}^{\mathbf{d}} = Softmax(\widetilde{\mathbf{P}}^{\mathbf{f}}) + Softmax(\widetilde{\mathbf{P}}) \tag{6}$$

where we apply the $Softmax$ to boost stability. Then, we substitute $\widetilde{\mathbf{P}}$ by $\widetilde{\mathbf{P}}^{\mathbf{d}}$ in Eq. 1 to have our loss function as,

$$\mathcal{L}_{\text{our}} = \lambda_1\mathcal{L}_{\text{adv}}(\widetilde{\mathbf{O}}, \mathbf{O}^c) + \lambda_2\mathcal{L}_{\text{rec}}(\widetilde{\mathbf{O}}, \mathbf{O}^c) + \lambda_3\mathcal{L}_{\text{rec}}(\widetilde{\mathbf{P}}^{\mathbf{d}}, \mathbf{P}^c) \tag{7}$$

finally, we get the denoised initial SISM results $\widetilde{\mathbf{S}}^{\mathbf{d}} \in \mathbb{R}^{H\times W\times 1}$ by $\widetilde{\mathbf{S}}^{\mathbf{d}} = \arg\max(\widetilde{\mathbf{P}}^{\mathbf{d}})$. Similar to Eq. 2, we can also get a denoised inpainted segmentation map $\hat{\mathbf{S}}^{\mathbf{d}}$ as,

$$\hat{\mathbf{S}}^{\mathbf{d}} = \widetilde{\mathbf{S}}^{\mathbf{d}} \cdot \mathbf{M} + \mathbf{S}^u \cdot (\mathbf{1} - \mathbf{M}) \tag{8}$$

Apply DA in the testing process. We can also apply DA in the testing process to avoid retraining a SISM model. Concretely, instead of conducting Eq. 2 in the testing process, we apply the noise filter $\mathbf{f}^N$ on the probability map $\widetilde{\mathbf{P}}$ from original G^0 as follows,

$$\hat{\mathbf{S}}^{\mathbf{d}} = \arg\max(\widetilde{\mathbf{P}} \cdot \mathbf{f}^N) \cdot \mathbf{M} + \mathbf{S}^u \cdot (\mathbf{1} - \mathbf{M}) \tag{9}$$

where $\arg\max(\widetilde{\mathbf{P}} \cdot \mathbf{f}^N)$ is a single-channel segmentation map. Then, the noise pixels are filtered in the SISM results.

3.3. Semantic Similarity (Sem)

The reason of the metric bias is that current SISM evaluations [13, 10] ignore the latent ground truths. The latent ground truths should have target objects with the same semantics as the ground truths, due to the SISM task requirement. Therefore, we propose a novel metric Sem to quantify the semantic divergence between the generated and ground-truth target objects for SISM.

In the Sem, we provide a semantic classifier F, which is pre-trained on PS-COCO [16]. For the design of F, since the semantics of segmentation maps are mainly represented by object shapes rather than textures, we should emphasize the object shapes and ignore object textures. Thus, the first-channel of the two-channel input to F is a binary target object map $\mathbf{O}^c \in \mathbb{R}^{H\times W}$. It sets certain pixels to 1 if the pixels in the ground-truth inpainted area ($\mathbf{S}^c \cdot \mathbf{M}$) have values as l^t, otherwise to 0. Because the sizes and locations of inpainted areas are various from samples, we design the second-channel of two-channel input as a positional embedding. It equals to $\mathbf{M}$ and informs F where should be classified. Since PS-COCO has 80 classes, the output of F is an 80-dimension vector. Its elements are classification probabilities. We use cross-entropy loss to train F.

The F is revised from EfficientNet [30], on which state-of-the-art image classifier [26] is also built. We revise it by adding a gated convolution [39] at the first layer of EfficientNet. The reasons of adding the gated convolution are two folds. First, there are many all-zero areas in each channel of the input, where gated convolution can alleviate their impact. Second, the object shapes are shown by the object boundaries, which are junctions of all-zero and all-one areas, thus, we expect gated convolution can dynamically learn optimized weights for the object boundaries.

After training F, given a testing sample $\hat{\mathbf{S}}$ (or $\hat{\mathbf{S}}^{\mathbf{d}}$) from any datasets, we extract its target binary object map $\hat{\mathbf{O}} \in \mathbb{R}^{H\times W\times 1}$ from $\hat{\mathbf{S}} \cdot \mathbf{M}$ by setting the pixel values to 1 if the respective pixels in $\hat{\mathbf{S}} \cdot \mathbf{M}$ have values as l^t, or otherwise to 0. Then, Sem calculates semantic divergence as follow,

$$Sem = 0.5\times|Softmax(F_{-1}(\hat{\mathbf{O}})) - Softmax(F_{-1}(\mathbf{O}^c))|_1 \tag{10}$$

where $|\bullet|_1$ is 1-norm and $F_{-1}(\bullet)$ is output of the last layer of F. We use the last layer rather than the penultimate layer, as we expect to normalize Sem into a range from 0 to 1; then, we need $Softmax$, which has better explanation added on the last layer than the penultimate layer. With $F_{-1}(\bullet)$, the Sem is explained as the semantic difference between the ground truths and generated results referred to PS-COCO classes. By Eq. 10, if $\hat{\mathbf{O}}$ shows more similar semantics to $\mathbf{O}^c$, then Sem is smaller. Thus, Sem can quantify the semantic divergence between target objects.

4. Experiments

4.1. Experiment Setup

Datasets. Our experiments are implemented on street scenes in **Cityscapes** [6], indoor scenes in **ADE20K** [46], and panoptic segmentations in COCO (**PS-COCO**) [16].

The Cityscapes and ADE20K are applied for SISM task, follow the setting as [13]. Specifically, we apply 2975 training images and 500 testing images from Cityscapes; 1239 training images and 150 testing images on bedroom images from ADE20K. Based on the rank of object numbers, we choose 8 movable semantic classes from total 35 semantic classes of Cityscapes for SISM. For ADE20K, which has 49 semantic classes, we choose 7 classes for SISM.

The PS-COCO is applied for our shape classifier. Concretely, PS-COCO provides panoptic segmentations, including both semantic-level and instance-level segmentations. Among its 118,287 training panoptic segmentations and 5000 validation panoptic segmentations in 80 object categories, we extract 332,310 binary instance segmentations for training and 14,343 binary instance segmentations for testing. These segmentations are extracted by filtering out the instances with sizes smaller than 1% of the respective image sizes, as we find the instances in too small sizes are too similar to instances in other semantic classes.

Implementation details. For the SISM task, we set $\lambda_1 = \lambda_2 = \lambda_3 = 1$ in Eq. 1 and Eq. 7. Plus, we apply Adam optimizer [15] with learning rate of 0.0002 to train the SISM model for 200 epochs on each dataset. For the downstream task of SISM, we apply the semantic image inpainting task with pre-trained model for ADE20K provided by [13]; we train the model for semantic image inpainting on Cityscapes by pix2pixHD [33] for 300 epochs [3].

For training the shape classifier described in Sec. 3.3, we apply EfficientNet-B7 [30] and initialize its parameters as its pre-trained ones. We then train the EfficientNet-B7 with gated convolution [39] for 50 epochs and choose the parameters resulting in the highest accuracy in the test set, where early stop also happens. The Adam optimizer with learning rate of 0.00001 is applied, where the learning rate decays 30% every 10 epochs.

Evaluation metrics. We apply four metrics to evaluate the overlapping degree, noises and semantics of the SISM results: (1) Target Intersection-Over-Union (**tIOU**) [18, 10], which is the ratio of overlapped area for target objects to their union area; (2) Hamming Distance (**hamm**) [22, 10], which is the ratio of the number of pixels, owning the same pixel values as the ground truths, to the number of total pixels; (3) Noises Number (**NN**), which calculates the average number of the noise pixels of the testing samples. (4) Semantic Similarity (**Sem**), which quantifies the semantic divergence between the generated and ground-truth target objects, by our shape classifier pre-trained on PS-COCO. We do not apply FID [12], as the natural images from downstream models might bring new bias, shown in the Sec. 4.3.

Baseline and ablation settings. For SISM, our baseline is the state-of-the-art SISM model applied in [13], which is also our structure generator introduced in Sec. 3.1. For simplification, we use Two-Stream Model (TwoSM) to represent the SISM model in [13]. Then, our DA is combined with it as TwoSM+DA, which applies the DA in the training by default. To show the effect of DA, we apply the TwoSM+DA(Train), TwoSM+DA(Test) and TwoSM+DA(Both) as our ablation study, to respectively represent applying DA in the training, in the testing, and in both the training and testing processes. To verify the effectiveness of DA, we compare our DA with 4 SOTA denoise methods trained with clean data: **ADCNN** [31] denoises images by feature enhancement and attention mechanisms; **DANet** [40] uses an UNet-based denoiser; **RIDNet** [2] uses a residual structure and feature attention for denoise; **FFDNet** [43] uses sub-images and a noise level map to denoise. we replace our DA with these baselines by revising their # channel of the input and output for segmentation maps, and keeping their input as $\widetilde{P}$, the output of G^0.

[3]More implementation details, metric introduction, data samples, and source code are provided in our supplementary materials.

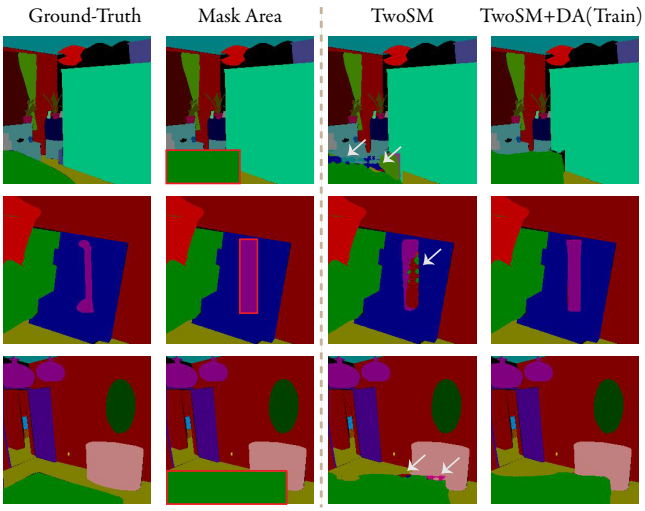

Figure 4. Examples of inpainted segmentation maps of TwoSM and TwoSM+DA(Train) on the ADE20K. The ground-truth segmentation maps and incomplete segmentation maps are shown on the left, where the red rectangles are mask areas. The right two columns show the inpainted segmentation maps of the two methods respectively. The white arrows point to the visible noise pixels.

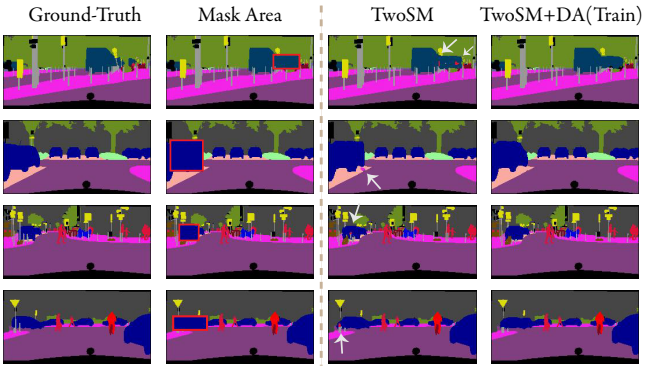

Figure 5. Examples of inpainted segmentation maps of TwoSM and TwoSM+DA(Train) on the Cityscapes. The images are organized in the same way as Fig. 4.

To verify that Sem can quantify semantic divergence for latent ground truths, we compare Sem with tIOU and hamm in various transformations on target objects like [12].

Methods	Cityscapes			
	tIOU↑	hamm↑	NN↓	Sem↓
TwoSM	0.8285	0.8428	5.519	**0.2756**
TwoSM+DANet	0.7683	0.7734	11.61	0.4049
TwoSM+ADCNN	0.8235	0.8332	5.088	0.2920
TwoSM+RIDNet	0.7830	0.7877	6.648	0.3466
TwoSM+FFDNet	0.8328	0.8424	7.352	0.2911
TwoSM+DA(Train)	**0.8330**	**0.8477**	**4.832**	0.2850

Table 1. The tIOU, hamm, NN and Sem on Cityscapes for SISM results in respond to the inpainted areas.

Methods	ADE20K			
	tIOU↑	hamm↑	NN↓	Sem↓
TwoSM	0.6505	0.6750	501.16	0.4762
TwoSM+DANet	0.6338	0.6645	103.64	**0.4640**
TwoSM+ADCNN	0.6861	0.6996	39.04	0.4892
TwoSM+RIDNet	0.6376	0.6612	91.67	0.4677
TwoSM+FFDNet	0.6271	0.6456	262.41	0.5067
TwoSM+DA(Test)	0.6632	0.6843	413.71	0.4767
TwoSM+DA(Train)	**0.6979**	**0.7116**	33.54	0.4891
TwoSM+DA(Both)	0.6967	0.7098	**25.64**	0.4889

Table 2. The tIOU, hamm, NN and Sem on ADE20K for SISM results in respond to the inpainted areas.

4.2. Quantitative Results Of SISM

Tab. 1 and Tab. 2 report the quantitative results of SISM by tIOU, hamm, NN and Sem. We can conclude as below,

(1) Our DA model achieves the best results on tIOU, hamm and NN in two datasets compared to the baselines. The reasons are twofold. Firstly, DA is an activation function, requiring much less parameters compared to the four baselines, then, the overfitting has less impact. Secondly, the four baselines are designed for images, where it is hard to guess the noise pixels, so the baselines learn the distribution of noise pixels only by the DNN. But DA guesses the noise pixel values besides the distribution of noise pixels.

(2) The DA module is effective in either training or testing process, For example, TwoSM+DA(Train) improves 7.29% in tIOU and decreases 0.687 noise pixels per SISM result on the Cityscapes. Plus, TwoSM+DA(Test) improves tIOU by 1.95% and hamm by 1.38% on ADE20K.

(3) The Sem is effective to quantify the semantics. In each dataset, the decreases of noise pixels do not obviously improve the semantics of target objects, shown in Fig. 4 and 5. And the Sem in each dataset also negligibly changes, confirming the ability of Sem in quantifying the semantics.

4.3. Qualitative Results Of SISM

Fig. 4 and Fig. 5 show the inpainted segmentation maps by TwoSM+DA on the ADE20K and Cityscapes, respectively. To show the application of SISM and the bias of evaluating SISM by its downstream natural images, the SISM results and their downstream natural images from semantic image inpainting [13] are both shown in the Fig. 7 and Fig. 8. From these figures, we conclude as below,

(1) The TwoSM+DA(Train) can effectively remove the noise pixels without disrupting the semantics. For examples, the noise pixels generated by TwoSM shown in the first row of Fig. 4 and Fig. 5 are removed in TwoSM+DA(Train).

(2) TwoSM+DA(Train) performs better than TwoSM+DA(Test). The Fig. 7 shows that more noises are removed by TwoSM+DA(Train).

(3) Some images from the downstream models cannot show the improvements of SISM results. From the respective fourth rows of Fig. 7 and Fig. 8, we cannot clearly see the effects of noise pixels in the natural images. The reason is that, the current downstream models do not perform optimally on the image inpainting, even when the noise pixels are removed. Thus, evaluating the SISM results by downstream results brings new bias caused by their limitation.

(4) Some noise pixels lead to obvious disturbances to downstream model performance, such as the noise pixels in the second row of Fig. 7, which represent "wall" semantics, and are translated into a wall attached to the corner of the bed in TwoSM and TwoSM+DA(Test).

4.4. Metrics Comparison

To further show the performance of Sem, we compare Sem with tIOU and hamm by the transformed target objects on the PS-COCO testing set. The comparison is similar to FID [12]. Here, we design 6 various image transformations with 4 different levels (levels 0-3). The results are shown in Fig. 6. Then, we conclude as below,

(1) From (a) and (b), Gaussian noise in the binary input obviously impacts Sem, as the noisy binary inputs have large semantic divergence to the ground truths. In the practical testing process, the inputs to Sem are binary with no noise due to the extraction method described in Sec. 3.3.

(2) From (c) and (d), all three metrics are sensitive to both the erode and dilate. But Sem is more sensitive than the hamm and tIOU, because the erode and dilate can mask details of the shape and alter the semantics.

(3) From (e), the flip transformation should keep the most original semantics of ground truths among the six transformations. Though all three metrics are sensitive to the flip, which changes the orientation of target objects, 1-Sem decreases around 50% slower than hamm and tIOU. It shows Sem quantifies semantic divergence for the latent ground truths better than hamm and tIOU.

(4) From (f), the semantic divergence to the ground truths should increase when the level enlarges, as the shapes, sizes, and orientations of target objects change more in higher levels. Though the tIOU and hamm do not consistently decrease at level-2, 1-Sem decreases consistently. This again verifies the superiority of Sem.

5. Conclusion

We improve the current SISM models by considering the unique characteristics of segmentation maps in both testing and training processes. First, we improve SISM training process by reducing the noise pixels in the SISM results. We propose DA to estimate a reasonable range of the pixel values in an inpanted area. The estimation is conducted based on the characteristic of segmentation maps, *smaller value*

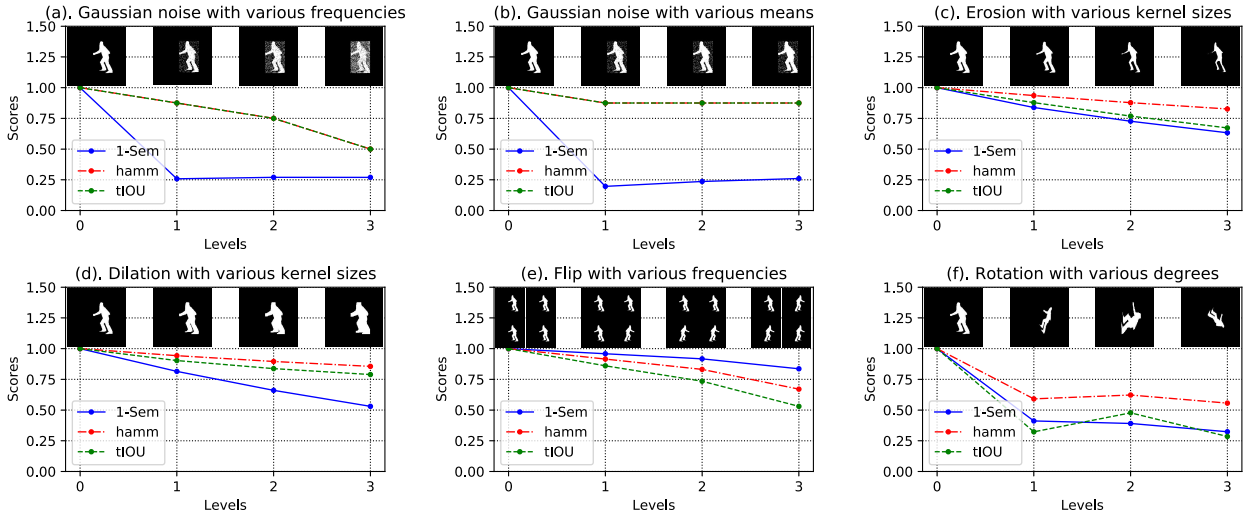

Figure 6. Results of metrics comparisons on the PS-COCO testing set with various transformations on target objects, where the transformations are only conducted on the bounding-box areas of objects. The titles of the sub-figures are descriptions of the transformations, and the inserted images are examples of each transformation at different levels. Concretely, we set all level-0 as no transformation. For (a), we fix the means and standard deviations of Gaussian noise both as 0.5, and set its appearing frequencies to 0.125, 0.25 and 0.5 for the levels 1-3. For (b), we fix standard deviations of Gaussian noise as 0.25, fix its appearing frequencies as 0.125, and set its means as 0.25, 0.5, and 0.75 for the levels 1-3. For (c) and (d), we set the kernel sizes of erosion or dilation as 3, 5, and 7 for the levels 1-3. For (e), we set the flip frequencies as 0.25, 0.5, and 0.75 for levels 1-3. For (f), we set the rotate degrees as 45, 90 and 135 for levels 1-3. To better compare the curves, we use 1-Sem, rather than Sem. In this case, higher 1-Sem refers to higher semantic similarity of the generated target objects compared with the respective ground truths.

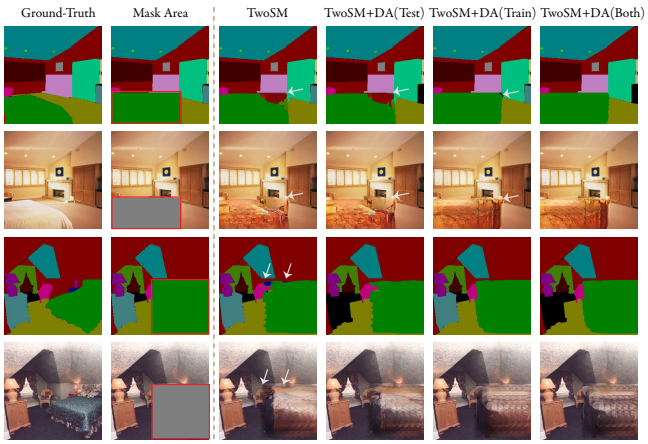

Figure 7. Examples of SISM and downstream model results on the ADE20K. The ground-truth segmentation maps, incomplete segmentation maps and ones for natural images are shown on the left, where the red rectangles are mask areas. The white arrows indicate the visible noise pixels or their respective regions in the natural images. Please zoom in for better visualization.

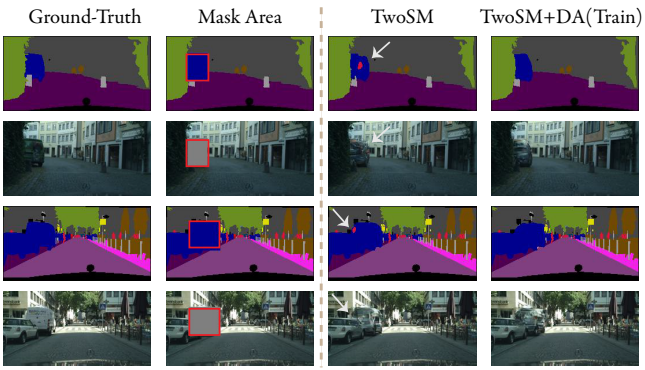

Figure 8. Examples of SISM and downstream model results on the Cityscapes. The images are organized in a similar way as Fig. 7. Please zoom in for better visualization.

range of each segmentation pixel value. Our DA is effective on reducing noise pixels in the both training and testing processes. Second, we improve SISM testing process by reducing the metric bias, because current metrics ignore the latent ground truths. We propose Sem to quantify semantic divergence between the generated and ground-truth target objects. The framework and training data of the pre-trained semantic classifier in Sem is designed by the characteristics of segmentation maps, *more object semantics from object shapes*. Sem is applicable to other datasets. Experiments on three datasets show impressive results of DA and Sem.

6. Acknowledgement

Shuhui Wang was supported in part by National Natural Science Foundation of China: 62022083.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access*, 6:14410–14430, 2018.

[2] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3155–3164, 2019.

[3] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. Semantic photo manipulation with a generative image prior. *ACM Transactions on Graphics (TOG)*, 38(4):1–11, 2019.

[4] Anja Borsdorf, Rainer Raupach, Thomas Flohr, and Joachim Hornegger. Wavelet based noise reduction in ct-images using correlation analysis. *IEEE transactions on medical imaging*, 27(12):1685–1703, 2008.

[5] Wengling Chen and James Hays. Sketchygan: Towards diverse and realistic sketch to image synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9416–9425, 2018.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[7] Yanhong Ding, Guowei Teng, Yuwei Yao, Ping An, Kai Li, and Xiang Li. Context-aware natural integration of advertisement object. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 4689–4693. IEEE, 2019.

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[9] Xintong Han, Zuxuan Wu, Weilin Huang, Matthew R Scott, and Larry S Davis. Finet: Compatible and diverse fashion image inpainting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4481–4491, 2019.

[10] Jianfeng He, Xuchao Zhang, Shuo Lei, Shuhui Wang, Qingming Huang, Chang-Tien Lu, and Bei Xiao. Semantic editing on segmentation map via multi-expansion loss. *arXiv preprint arXiv:2010.08128*, 2020.

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[12] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[13] Seunghoon Hong, Xinchen Yan, Thomas S Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Advances in Neural Information Processing Systems*, pages 2708–2718, 2018.

[14] Asem Khmag, Abd Rahman Ramli, Shaiful Jahari bin Hashim, and Syed Abdul Rahman Al-Haddad. Additive noise reduction in natural images using second-generation wavelet transform hidden markov models. *IEEJ Transactions on Electrical and Electronic Engineering*, 11(3):339–347, 2016.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019.

[17] Liang Liao, Jing Xiao, Zheng Wang, Chia-wen Lin, and Shin'ichi Satoh. Guidance and evaluation: Semantic-aware image inpainting for mixed scenes. *arXiv preprint arXiv:2003.06877*, 2020.

[18] Xihui Liu, Guojun Yin, Jing Shao, Xiaogang Wang, et al. Learning to predict layout-to-image conditional convolutions for semantic image synthesis. In *Advances in Neural Information Processing Systems*, pages 568–578, 2019.

[19] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

[20] Ilja Manakov, Markus Rohm, Christoph Kern, Benedikt Schworm, Karsten Kortuem, and Volker Tresp. Noise as domain shift: Denoising medical images by unpaired image translation. In *Domain Adaptation and Representation Transfer and Medical Image Learning with Less Labels and Imperfect Data*, pages 3–10. Springer, 2019.

[21] Sangwoo Mo, Minsu Cho, and Jinwoo Shin. Instagan: Instance-aware image-to-image translation. *arXiv preprint arXiv:1812.10889*, 2018.

[22] Juhyeok Mun, Won-Dong Jang, Deuk Jae Sung, and Chang-Su Kim. Comparison of objective functions in cnn-based prostate magnetic resonance image segmentation. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 3859–3863. IEEE, 2017.

[23] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. Sesame: Semantic editing of scenes by adding, manipulating or erasing objects. *arXiv preprint arXiv:2004.04977*, 2020.

[24] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[25] Mohammad Tanvir Parvez and Adnan Abdul-Aziz Gutub. Rgb intensity based variable-bits image steganography. In *2008 IEEE Asia-Pacific Services Computing Conference*, pages 1322–1327. IEEE, 2008.

[26] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.

[27] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction

and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018.

[28] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[29] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[30] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[31] Chunwei Tian, Yong Xu, Zuoyong Li, Wangmeng Zuo, Lunke Fei, and Hong Liu. Attention-guided cnn for image denoising. *Neural Networks*, 124:117–129, 2020.

[32] Pei Wang and Albert CS Chung. Focal dice loss and image dilation for brain tumor segmentation. In *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, pages 119–127. Springer, 2018.

[33] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[35] Jun Wei, Shuhui Wang, Zhe Wu, Chi Su, Qingming Huang, and Qi Tian. Label decoupling framework for salient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13025–13034, 2020.

[36] Bing Xu, Naiyan Wang, Tianqi Chen, and Mu Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[37] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*, 2018.

[38] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020.

[39] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[40] Zongsheng Yue, Qian Zhao, Lei Zhang, and Deyu Meng. Dual adversarial network: Toward real-world noise removal and noise generation. In *European Conference on Computer Vision*, pages 41–58. Springer, 2020.

[41] Shaoning Zeng and Bob Zhang. Noise homogenization via multi-channel wavelet filtering for high-fidelity sample generation in gans. *arXiv preprint arXiv:2005.06707*, 2020.

[42] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International Conference on Machine Learning*, pages 7354–7363. PMLR, 2019.

[43] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[44] Pan Zhang, Bo Zhang, Dong Chen, Lu Yuan, and Fang Wen. Cross-domain correspondence learning for exemplar-based image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5143–5153, 2020.

[45] Lei Zhao, Qihang Mo, Sihuan Lin, Zhizhong Wang, Zhiwen Zuo, Haibo Chen, Wei Xing, and Dongming Lu. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5741–5750, 2020.

[46] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.

[47] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[48] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5104–5113, 2020.