Eurographics Symposium on Rendering (DL-only Track) (2021)
A. Bousseau and M. McGuire (Editors)

Semantic-Aware Generative Approach for Image Inpainting

Deepankar Chanda and Nima Khademi Kalantari

Texas A & M University, USA

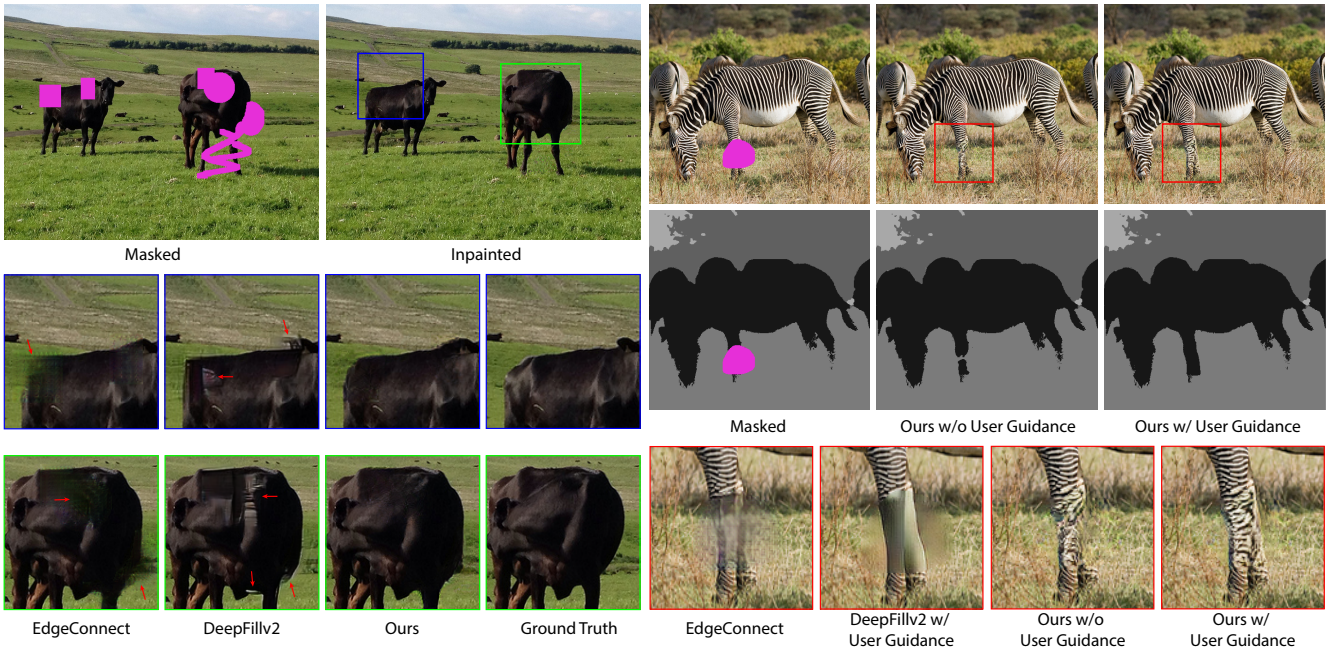

Figure 1: *We propose a novel generative approach for image inpainting by incorporating the semantic information through conditional feature modulation and using dual discriminators to train the network. On the left, we show a comparison against EdgeConnect [NNJ*19] and DeepFillv2 [YLY*19b]. These methods do not incorporate semantic information and, therefore, often produce results with inconsistent color and texture and incorrect object boundaries. Our method produces an overall high-quality result with reasonable object boundaries and visually pleasing textures. On the right, we show an example where we allow the user to manually edit the estimated semantic map to obtain the desired results. EdgeConnect's results are obtained automatically, but we produce the result for DeepFillv2 with their user interactive version. While our automatically generated results are better than both other methods, with the additional user guidance, our method is able to improve the shape of the legs and the stripe patterns.*

Abstract

We propose a semantic-aware generative method for image inpainting. Specifically, we divide the inpainting process into two tasks; estimating the semantic information inside the masked areas and inpainting these regions using the semantic information. To effectively utilize the semantic information, we inject them into the generator through conditional feature modulation. Furthermore, we introduce an adversarial framework with dual discriminators to train our generator. In our system, an input consistency discriminator evaluates the inpainted region to best match the surrounding unmasked areas and a semantic consistency discriminator assesses whether the generated image is consistent with the semantic labels. To obtain the complete input semantic map, we first use a pre-trained network to compute the semantic map in the unmasked areas and inpaint it using a network trained in an adversarial manner. We compare our approach against state-of-the-art methods and show significant improvement in the visual quality of the results. Furthermore, we demonstrate the ability of our system to generate user-desired results by allowing a user to manually edit the estimated semantic map.

CCS Concepts

• ***Computing methodologies*** → *Computational photography;* ***Image processing;***

DOI: 10.2312/sr.20211291

1. Introduction

Image inpainting is the task of reconstructing missing regions in a masked image. It has a variety of applications including image restoration, object removal, image stitching, and novel view synthesis. To effectively complete a masked region, an inpainting approach should understand the scene context, recover the shape of the objects overlapping the mask, and fill them in with plausible textures. In recent years, several approaches have proposed to learn this process using generative adversarial networks [PKD*16; ISI17; YLL*17; NNJ*19; YLY*19b]. However, current methods are generally unable to effectively learn both tasks of contextual scene understanding and plausible texture synthesis together. As a result, they often produce images with inconsistent textures and object boundaries, as shown in Fig. 1 (left).

We propose a guided approach to image inpainting by splitting it into two tasks. First, we predict the scene semantics by estimating a completed semantic segmentation map. Then this semantic information is used to inpaint missing regions in the image. To incorporate the semantic information, we modulate the image features at different layers of the generator using a set of parameters obtained from the semantic segmentation map. Through this conditional feature modulation, the generator is able to synthesize consistent color and texture for each region with clear boundaries between semantically different areas.

Furthermore, we propose to train our network using an adversarial framework with dual discriminators. Our *input consistency* discriminator is conditioned on the mask and ensures that the generated image is consistent with the input masked image. Our *semantic consistency* discriminator, on the other hand, is conditioned on the semantic segmentation map to ensure that the generator synthesizes a high-quality image that adheres to the semantic map. To estimate the segmentation map, we first use a pre-trained model [CPK*18; YWP*18] to generate a map from the input masked image. We then inpaint this estimated segmentation map using a network, which we train in an adversarial manner.

We show that our approach is able to synthesize images with consistent color, texture, and object boundaries that are significantly better than the current state of the art. In summary, we make the following contributions:

- We propose to effectively utilize the semantic information through semantic-aware feature modulation (Section 3.1).
- We propose an adversarial framework with dual discriminators (Section 3.2) and demonstrate that it is necessary for producing high-quality results.
- We demonstrate that we can produce user-desired results by allowing a user to manually edit the semantic map (Fig. 1).

2. Related Work

Image inpainting has been the subject of extensive research. We begin by reviewing the non-learning approaches and follow with a discussion on the more recent learning-based methods. We also discuss the advances in semantic image synthesis.

2.1. Non-learning Approaches

Diffusion-based techniques inpaint missing regions by propagating information from the periphery of missing areas to their center. Ballester et al. [BBC*01] introduce a variational approach to filling in the missing areas. A few methods [BBS01; CS01] propagate isophote information from the mask boundary to the center and utilize gradients at the boundary of the masked region. Levin et al. [LZW03] use a histogram of local features taken from an image to find the solution. Unfortunately, these diffusion-based methods are limited to narrow masks and generate unnatural textures in the areas away from the mask boundaries.

Patch-based approaches use exemplar regions or patch statistics to find candidate solutions from the unmasked regions of the input image to synthesize missing content. Efors and Leung [EL99] propose a model based on Markov Random Field (MRF) which grows texture into the missing regions from a point in the image. Criminisi et al. [CPT04] use exemplar-based methods for propagating color information in missing regions. Kwatra et al. [KEBK05] define an MRF-based similarity metric to perform energy minimization for texture synthesis. To fill in large masked areas, Wexler et al. [WSI07] propose a global patch-based optimization system. Barnes et al. [BSFG09] demonstrate a randomized algorithm which matches plausible image patches as candidate solutions for the missing areas. Kopf [KKDK12] use non-parametric methods to predict output quality from features used for synthesis. Darabi et al. [DSB*12] propose to enrich the patch search with additional geometric and photometric transformations.

However, most patch-based approaches assume that missing content can be found entirely within the masked image which is not always the case. Hays and Efros [HE07] avoid this problem by adopting a dictionary-based approach using a large image dataset, but their quality is heavily dependent on finding good image matches.

2.2. Learning-Based Approaches

Deep learning for inpainting has seen rapid progress due to promising results delivered by new techniques. Convolutional neural networks (CNN) have proven particularly effective at the task. Pathak et al. [PKD*16] use a context encoder to capture visual information surrounding missing regions through a generative adversarial network (GAN). Iizuka et al. [ISI17] use a fully convolutional network with context discriminators which enforce global and local image consistency. Yang et al. [YLL*17] propose a joint optimization framework where local texture synthesis is performed using features extracted from the middle layers of a network. Yan et al. [YLL*18] employ shift-connections to locate plausible features obtained from convolutions. Yu et al. [YLY*18] use contextual attention via region matching to find areas similar to the missing patches to improve the quality of the inpainted content. More recently, techniques have proposed the use of partial convolutions [LRS*18; YLY*19a], that derive information for missing patches from only unmasked areas. Zeng et al. [ZLY*20] propose an iterative method that uses confidence maps to determine valid pixels in a given iteration. Unfortunately, these approaches are not able to properly learn both tasks of semantic scene understanding and texture synthesis, which are required for producing

high-quality results. Therefore, they often produce results with inconsistent textures and incorrect object boundaries.

To avoid synthesizing the missing regions with incorrect boundaries, Nazeri et al. [NNJ*19] and Xiong et al. [XYL*19] propose to estimate the edges of the objects in the masked areas to guide the inpainting process. Other approaches [LJXY19; RYZ*19] try to implicitly model image semantics and structure. However, these approaches often are not able to properly recover the edges in the missing regions and are limited to simpler scenes with a single masked object. Moreover, they can still produce results with inconsistent textures as it is difficult for the network to understand the semantics without guidance.

Song et al. [SYS*18] propose to guide the inpainting network using a semantic segmentation map. However, they concatenate the map as the input to the network. Thus, their approach fails to utilize the semantic information effectively, often producing results with artifacts similar to the previous approaches. Liao et al. [LXW*20] also proposes to utilize semantic information, however, the image and semantic map at each resolution (especially at the coarsest scale) are generated independently. This could potentially create a mismatch between the generated content and the semantic label. Moreover, since they simultaneously perform both image inpainting and semantic map estimation, the semantic map is always estimated from the masked image. This constrains their approach for applications like object removal where the map can be estimated from the full image. Finally, It is not clear how to extend their approach to incorporate user edits to the semantic map.

2.3. Semantic Image Synthesis

A number of approaches propose to perform image-to-image translation using deep neural networks [IZZE17; WLZ*18a; LYS*19]. These methods are able to translate an abstract representation of a scene, such as the semantic map, to an image of the scene. Recent advances have significantly improved the perceptual quality of results. Our work is inspired by the techniques [PLWZ19; LYS*19; ZAQW20; LLWL20a] that perform this task through feature normalization. These approaches adaptively re-normalize the image features based on content acquired through another source. The main difference here is that image to image translation is unconstrained in the sense that any visually pleasing output image is acceptable. In our problem, however, the output of the network should match the input image in the unmasked areas.

3. Semantic-Aware Image Inpainting

Given an input masked image T and a binary mask M identifying the masked areas (1 for masked regions and 0 elsewhere), our goal is to reconstruct an image $\hat{I}$ by filling in the masked areas with visually pleasing content. The final inpainted image $\hat{I}$ is computed using the output of the generator $\hat{T}$ (generated image[†]) as follows:

$$\hat{I} = M \odot \hat{T} + (1 - M) \odot T, \tag{1}$$

[†] We refer to the final result of our system as inpainted image, while generated image is used to refer to the direct output of the network.

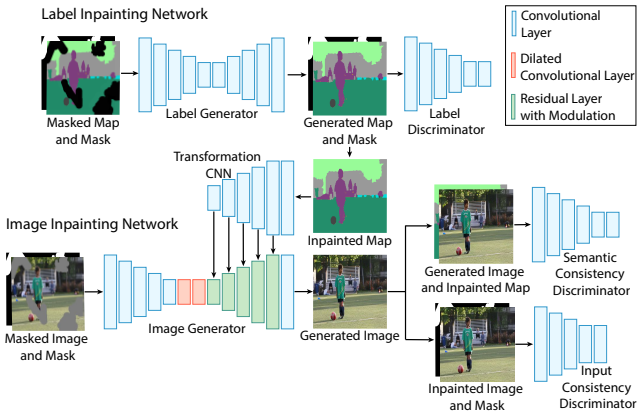

Figure 2: *We show the overview of our approach. The label inpainting network takes the semantic map obtained from a pretrained network as an input and completes the missing regions. This complete estimated segmentation map is then used to modulate the features in the image inpainting network. We use two discriminators to evaluate the quality of the inpainted images, while the label generator is trained using a single discriminator.*

where $\odot$ indicates the Hadamard product. We use an encoder-decoder CNN as our generator and inject semantic information into different layers of the decoder through feature modulation. To train this generator, we propose an adversarial framework with dual discriminators to ensure the generated images are consistent with the semantic labels and fit well with the input image based on Eq. 1. An overview of the approach is given in Fig. 2. The detailed architectures of our networks are provided in the supplementary materials. In the following sections, we first explain our approach for incorporating the semantic information and then discuss our proposed adversarial framework with dual discriminators. Finally, we describe our approach for estimating the semantic map.

3.1. Semantically-Conditioned Feature Modulation

Our goal here is to effectively incorporate the semantic information in the generator. Note that in this section we assume that the complete semantic map $\hat{S}$ is available (i.e, we have access to label values both inside and outside of the masked region). We discuss our approach for estimating the segmentation map in Sec. 3.3. The naïve way to incorporate this map is to concatenate it to the input image and pass that to the generator [SYS*18]. However, the generator in this case is not able to properly use the map because the image feature maps computed by successive convolutional layers in the encoder tend to dilute or wash away the semantic information [PLWZ19]. Therefore this naïve approach produces results with unnatural object boundaries, discoloration, and poor intra-semantic textures, as shown in Fig. 10.

Inspired by the recent success of conditional feature modulation in a variety of applications such as style transfer [WYDL18], image synthesis [PLWZ19; dVSM*17], and Monte Carlo denoising [XZW*19], we propose to integrate semantic information into the generator by modulating the feature maps using the parameters obtained from the semantic segmentation map. Specifically, we modulate the estimated feature maps at each residual

layer [HZRS15] of the decoder, f^i, as follows:

$$f^i_{mod} = \gamma^i \odot f^i + \beta^i. \quad (2)$$

where γ^i and β^i are the modulation parameters, estimated from the semantic map. γ^i and β^i have size $h^i \times w^i \times c^i$, where h^i, w^i, and c^i are the height, width, and channels of the feature map f^i.

Inspired by Wang et al.'s approach [WLT*19], we compute the modulation parameters by first passing the semantic map through an encoder, called transformation CNN, to obtain a set of modulating features at every layer of the encoder, as shown in Fig. 2. We then pass the modulating features at each layer through two separate convolutions to produce the γ and β for that layer. We perform the feature modulation in Eq. 2 before each convolutional layer in the residual layers of the decoder.

As shown in Fig. 2, we propose to only apply the conditional modulation to the layers in the decoder. This is mainly because the decoder is responsible for synthesizing content from the latent representation of the image and, thus, benefits from the conditional modulations. The encoder, on the other hand, does not perform any synthesis and mainly transforms the input image into the latent space. In the next section, we discuss our adversarial framework with dual discriminator for training the generator.

3.2. Adversarial Framework with Dual Discriminator

Due to the success of generative adversarial networks (GAN) [GPM*14] in image inpainting [PKD*16; ISI17; YLY*19b], we train our generator using an adversarial loss function. We introduce two discriminators to distinguish the completed image from the ground truth. The *input consistency* discriminator D_{in} forces the generator to produce results that are consistent with the input masked image, while the *semantic consistency* discriminator D_{sem} is responsible for ensuring that the output of the generator is consistent with the semantic map.

Specifically, the input consistency discriminator distinguishes the final inpainted image $\hat{I}$ (obtained using Eq. 1) from ground truth and is conditioned on the mask M. We provide the final inpainted image so the discriminator can determine if the generated image blends well with the input image. Moreover, by conditioning the discriminator on the mask, we make it easier for the discriminator to focus on the masked areas. The semantic consistency discriminator is responsible for discerning the output of the generator, $\hat{T}$, from ground truth and is conditioned on the semantic map, $\hat{S}$. This discriminator learns to associate feature and texture from the image to the semantic labels. Hence, it ensures the generator is able to synthesize an image that is consistent with the semantic map.

Both discriminators are Markovian [IZZE17] so they evaluate a patch within the input image and indicate whether this region is real or fake. We also use spectral normalization [MKKY18] in both discriminators to improve training stability. We train the generator and both discriminators by optimizing the following objective:

$$\begin{aligned}\mathcal{L} &= \lambda_1 \mathcal{L}_{\text{adv}}(G, D_{\text{in}}) + \lambda_2 \mathcal{L}_{\text{adv}}(G, D_{\text{sem}}) + \lambda_3 (\mathcal{L}_{\text{feat}}(G, D_{\text{in}}) \\ &\quad + \mathcal{L}_{\text{feat}}(G, D_{\text{sem}})) + \lambda_4 \mathcal{L}_{\text{VGG}}(\hat{T}, I) + \lambda_5 \|\hat{T} - I\|_1, \end{aligned} \quad (3)$$

where the first two terms are the adversarial loss for our two discriminators based on hinge loss [LY17]. Furthermore, we use feature matching loss $\mathcal{L}_{\text{feat}}$ [WLZ*18b], to help stabilize the adversarial training. Moreover, we use the VGG-based perceptual loss, $\mathcal{L}_{\text{VGG}}$, between the generated, $\hat{T}$, and the ground truth, I, images, which aids in the generation of high-quality results and in capturing high-level object representations in image. Additionally, the last term ensures that the generated image does not significantly deviate from the ground truth in an L_1 sense. Finally, λ_1 through λ_5 are the weights of each term and we set them to 1.0, 1.0, 10, 5.0, and 20, respectively.

Note that, most existing GAN-based inpainting techniques [XYL*19; NNJ*19; YLY*19b] use a single discriminator. On the other hand, a couple of methods [ISI17; YLY*18] propose to use multiple discriminators (global and local). However, these approaches are fundamentally different from our dual discriminator framework in two major ways. First, unlike these approaches, both our discriminators are global, which allows our system to work on freeform masks. Second, our discriminators evaluate both the generated and inpainted images, which is different from the other approaches, where only either the inpainted or generated images are evaluated.

3.3. Semantic Map Estimation

Our goal here is to estimate a semantic segmentation map $\hat{S}$ for the areas both inside and outside the mask. To do so, we first use a pretrained network to generate the semantic map B for the unmasked areas. We then estimate the semantic map in the masked areas by using this map as the input to an encoder-decoder based generator. We convert the output of the pretrained network into one hot vector before passing it to our network. The output of our network is a probability map, which we convert to a one-hot vector form to obtain the output semantic map, $\hat{B}$. We compute the final estimated semantic map $\hat{S}$ by combining B and $\hat{B}$ using the mask M, i.e., $\hat{S} = M \odot \hat{B} + (1 - M) \odot B$.

To train this network, we optimize the following loss function:

$$\begin{aligned}\mathcal{L} &= \lambda_1 \mathcal{L}_{\text{adv}}(G, D) + \lambda_2 \mathcal{L}_{\text{feat}}(G, D) \\ &\quad + \lambda_3 \mathcal{L}_c(\hat{B}, S) + \lambda_4 \|\nabla \hat{B}\|_1, \end{aligned} \quad (4)$$

where $\mathcal{L}_{\text{adv}}$ and $\mathcal{L}_{\text{feat}}$ are the adversarial and feature matching losses which function similarly to ones described in Eq. 3. The third term $\mathcal{L}_c$ is the cross-entropy loss between the ground truth segmentation map and the probability map estimated by the network. This loss constrains the network to generate results comparable to the ground truth. The last term is the total variation loss which encourages the network to produce smooth segmentation maps. Finally, λ_1 through λ_4 define the weight of each term and we set them to 1.0, 1.0, 1.0, and 10^{-5}, respectively.

4. Experiments

We implement our approach in PyTorch and use Adam [KB15] with $\beta_1 = 0.0$ and $\beta_2 = 0.999$ to perform the optimization. To train the image inpainting network we extract patches of size 256 x 256

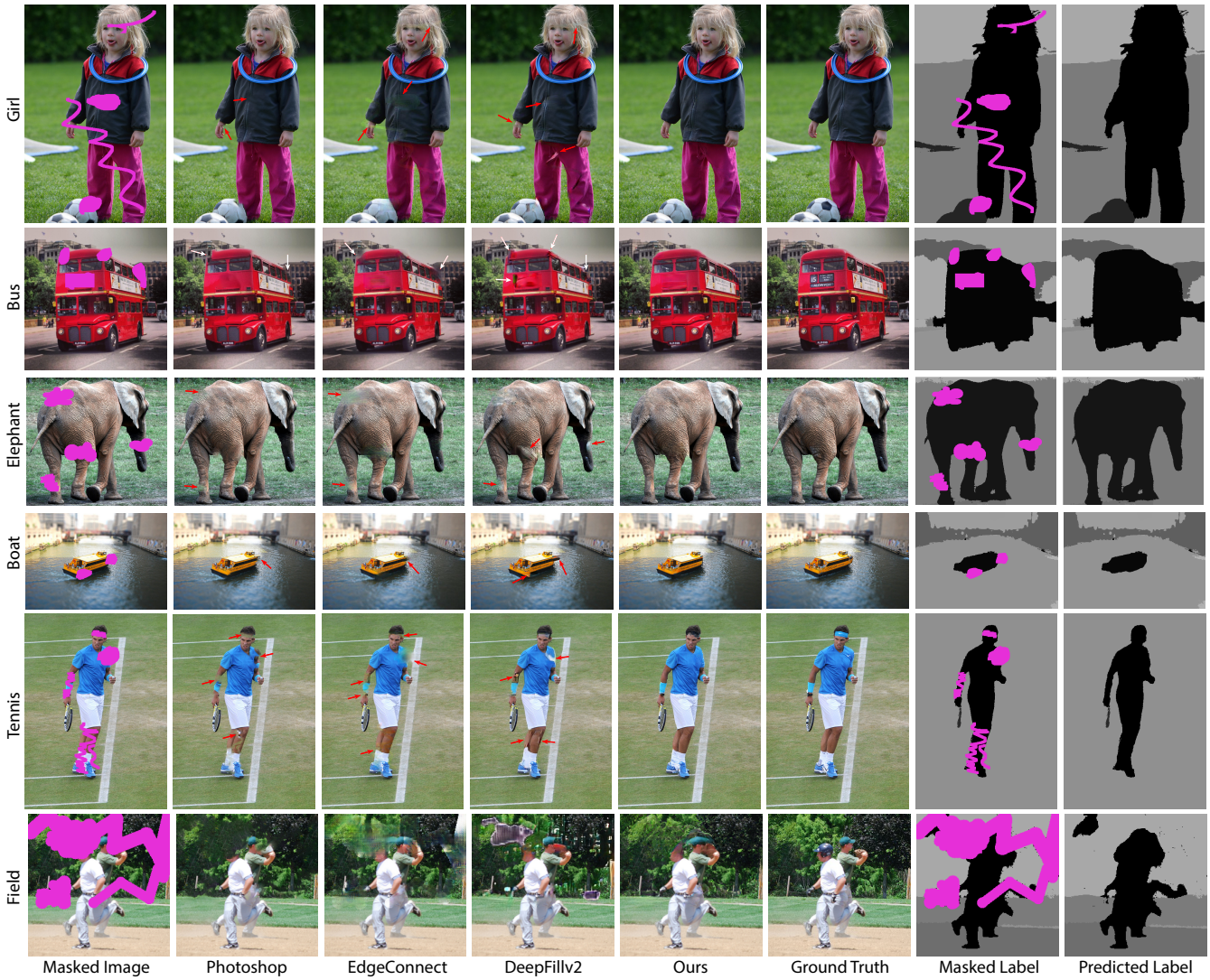

Figure 3: *Comparison with other approaches on image restoration. For each result, we also show the input semantic map generated by DeepLabv2 [CPK*18] and our estimated inpainted map. Guided by the semantic map, our method is able to produce results with better color and texture consistency and object boundaries. To better see the differences, zoom into the electronic version of the paper. The full images are provided in the supplementary materials.*

using random cropping and rescaling. The masks are generated on-the-fly during training. We generate our masks by selecting a mask type randomly from hand-drawn, polygonal, circular, or brush. We then generate a random set of shapes from this choice and place them around a small circular masked region (called a nucleation site). These nucleation sites are centered at a random point in the image. The process of generating random shapes is repeated multiple times to get the final masked image. We also draw masks using a randomized algorithm [YLY*19b] that draws curves given constraints on the angle. This ensures that our masks are fairly diverse and can generalize to hand-drawn examples. We generate random and hand-drawn masks for comparison against other methods. Hand-drawn masks are obtained using Adobe Photoshop's rectangle and brush tools.

To be able to effectively train both the label and image inpainting networks, we perform the training in two phases. In the first stage, we independently train both the image and label inpainting networks using the loss functions in Eqs. 3 and 4, respectively. In the second stage, we jointly optimize the entire system in an end-to-end fashion. Note that, we only use the inpainting loss in Eq. 3 during this stage as we want to fine-tune both networks to maximize the quality of the inpainted results. In phase one, we use a learning rate of 1×10^{-4} for both generator and discriminator of the label inpainting network. On the other hand, the generator and the dual discriminators of the image inpainting network have a learning rate of 1×10^{-4} and 4×10^{-4}, respectively. During joint optimization, we use a learning rate of 1×10^{-4} for the generator of both the label and image inpainting networks and a learning rate of 4×10^{-4} for

Figure 4: *Comparison against the other approaches for object removal. Zoom into the electronic version to better see differences. We provide the full images in the supplementary materials.*

Table 1: *Quantitative comparison against other methods in an image restoration setting on the COCO-Stuff [CUF18] dataset.*

	COCO-Stuff		
Algorithm	FID↓	LPIPS↓	SSIM↑
EdgeConnect	13.315	0.115	0.8544
DeepFillv2	7.449	0.085	0.8801
Profill	18.01	0.0935	0.8385
Baseline	8.2	0.0840	0.8806
Ours	**6.597**	**0.077**	**0.8848**

the dual discriminators [HRU*17]. The complete training process takes about six days on a single GeForce RTX 2080Ti.

We show the results of our model on two publicly available datasets. One is the COCO-Stuff dataset [CUF18] which has over 123,000 images, and a total of 183 classes including an unlabeled class. The other is the CelebAHQ-Mask [LLWL20b] dataset which has over 30,000 images with 19 classes. For the CelebAHQ-Mask dataset, we reduce the number of classes to 15 by combining classes like "left-eye" and "right-eye" to a single label, "eye". In order to obtain the initial semantic map of an image we use the pre-trained DeepLabv2 [CPK*18] network for the COCO-Stuff dataset and BiSeNet [YWP*18] for the CelebAHQ-Mask dataset. In addition to images from these two datasets, we also show results on general images (not from these datasets) to demonstrate the generality of our approach. Note that the full images along with additional examples are included in the supplementary material.

4.1. Comparison Against Other Methods

Throughout this section, we compare our approach against Adobe Photoshop Content-Aware Fill, DeepFillv2 by Yu et al. [YLY*19b], EdgeConnect by Nazeri et al. [NNJ*19], and ProFill by Zeng et al. [ZLY*20]. We use the official implementations of DeepFillv2 and EdgeConnect provided by the authors. To ensure fairness, we retrain both approaches on the COCO-Stuff dataset until convergence. For the CelebAHQ-Mask dataset, we use the official pre-trained models made available by the authors. For ProFill, we use the API provided by the authors to produce results using their pre-trained network. For Adobe Photoshop Content-Aware Fill, we set the area sampling to auto. For the baseline model, we modify our image generator so that it accepts the semantic map as an additional input along with the masked image. Additionally, we replace the residual blocks containing the modulation layers in the decoder of our image generator with convolution layers. This baseline model basically represents our implementation of the approach by Song et al. [SYS*18]. We generate random and hand-drawn masks for comparison against other methods. Hand-drawn masks are obtained using Photoshop rectangle and brush tools.

4.1.1. Image Restoration

In image restoration, the assumption is that the original image is distorted and, thus, we only have access to the masked images. Therefore, we estimate the initial segmentation map on the masked images. We begin by showing quantitative comparisons against EdgeConnect and DeepFillv2 in Table 1. The approaches are compared using three metrics; the Fréchet Inception Distance (FID) [HRU*17], Learned Perceptual Image Patch Similarity (LPIPS) [ZIE*18], and Structural Similarity Index Measure

Table 2: *Quantitative comparison against other methods for different mask size coverage.*

	10%-20%			20%-30%		
Algorithm	FID↓	LPIPS↓	SSIM↑	FID↓	LPIPS↓	SSIM↑
EdgeConnect	8.445	0.171	0.7964	8.485	0.231	0.7308
DeepFillv2	8.115	0.167	0.7995	8.312	0.206	0.7447
Profill	7.109	0.156	0.7483	8.614	0.193	0.7059
Ours	**4.169**	**0.128**	**0.8137**	**4.256**	**0.165**	**0.7583**
	30%-40%			40%-50%		
Algorithm	FID↓	LPIPS↓	SSIM↑	FID↓	LPIPS↓	SSIM↑
EdgeConnect	10.000	0.321	0.6550	15.987	0.421	0.5798
DeepFillv2	8.744	0.251	0.6904	9.136	0.294	0.6389
Profill	10.267	0.234	0.6628	12.065	0.271	0.6226
Ours	**4.377**	**0.205**	**0.7026**	**4.661**	**0.244**	**0.6500**

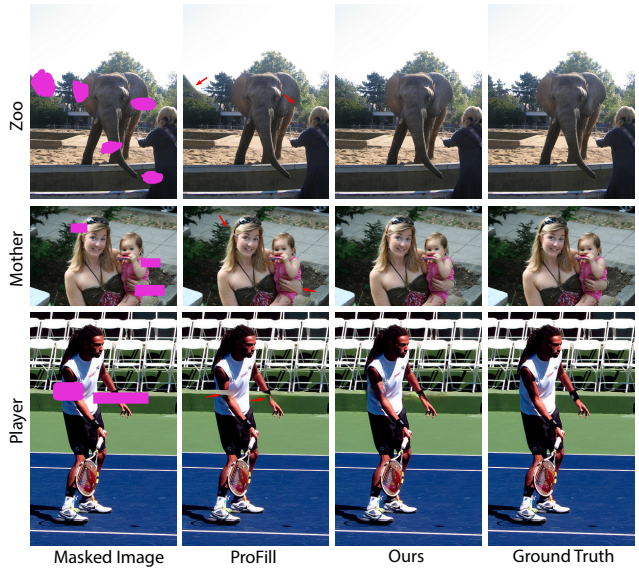

Figure 5: *Comparison against Zeng et al.'s approach [ZLY*20] on three scenes. See supplementary materials for more comparisons.*

(SSIM). We compute these metrics on 4950 test images from the COCO-Stuff dataset. From Table 1, we see that our method outperforms the others across metrics measuring perceptual quality.

We also compare the results numerically for different mask coverage. Table 2 shows a comparison of other methods against ours for masks covering various percentage of the image ranging from 10% to 50%. Our method is robust to distortions occurring as a result of large mask sizes and is significantly better than the other methods across the perceptual metrics.

Next, we visually compare our approach against Photoshop, EdgeConnect, and DeepFillv2 in Fig. 3. Overall, Photoshop is not aware of the semantic information of the scene and often generates textures that are not semantically meaningful. For example, it reconstructs the head and legs of the tennis player with grass textures, or repeats scene texture like in BOAT. EdgeConnect and DeepFillv2 are both deep learning approaches and could potentially learn the semantic information through training. However, they still are not able to fill in the missing regions with consistent texture and color and often reconstruct objects with distorted boundaries. For example, EdgeConnect struggles to reconstruct the textures in the GIRL scene, and is not able to reconstruct the tennis player in TENNIS with visually pleasing boundaries. Similarly, DeepFillv2 reconstructs the boat in the BOAT scene with distorted boundaries and has inconsistent textures in the scene FIELD.

Our approach, on the other hand, produces results with visually pleasing textures and object boundaries. Note that, only our approach is able to properly reconstruct the fine details of the hair in the GIRL scene. Moreover, our system learns to weight the semantic map and image content appropriately especially in regions where the semantic maps are inaccurate or lack details. For example, although the predicted semantic map in the FIELD scene is inaccurate mislabelling parts of the tree as a person, our method is able to properly reconstruct the textures and boundaries of the grass and trees in the background.

In Fig. 5, we compare our approach against the recent method by Zeng et al. [ZLY*20]. Similar to the other existing methods, this approach fails to properly reconstruct the object boundaries. For example, it is not able to properly reconstruct the hand in the MOTHER scene and the arms in the PLAYER scene. Additionally, it often does not reconstruct detailed textures as indicated by the arrows in the ZOO and MOTHER scenes. Our method produces results with overall higher quality and consistent texture and object boundaries.

4.1.2. Object Removal

Here, the goal is to remove certain objects from an image and, thus, we have access to the original image. Therefore, we run the pre-trained segmentation network on the original unmasked images and mask the estimated segmentation map before sending it to our label inpainting network. We show comparison against the other methods on a set of diverse images in Fig. 4. The BUILDING, HORSES, and ZEBRAS scenes are from COCO-Stuff dataset, while the FACE scene is from CelebAHQ-Mask. Moreover, TOURIST is a general image outside these two datasets.

In the BUILDING scene, other approaches struggle to reconstruct a sharp boundary between the street and the buildings. Our method is able to properly inpaint the semantic map which guides our system to reconstruct the missing region with an appropriate boundary

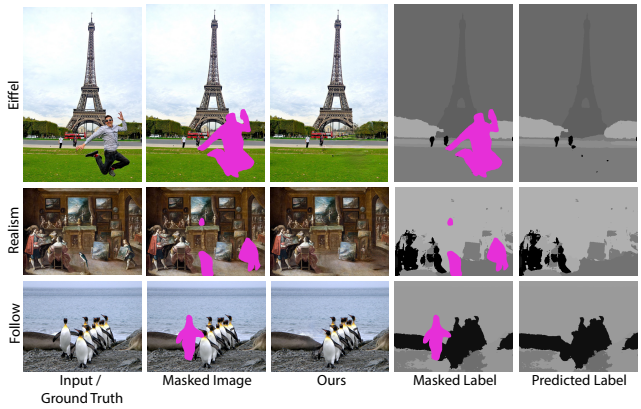

Figure 6: *Our method is able to generalize well on a large variety of images ranging from natural scenes to artistic impressions.*

between the inter-semantic regions. In the HORSES scene, EdgeConnect shows significant blurring in the inpainted region, while Photoshop and DeepFillv2 both generate spurious artifacts. However, our method reconstructs the missing region with plausible textures. In both FACE and TOURIST, Photoshop is unable to generate a realistic result and EdgeConnect produces a result with inter-object boundary artifacts. DeepFillv2 has severe artifacts and has lower perceptual quality compared with our method. Finally, in the ZEBRA scene, both EdgeConnect and DeepFillv2 are unable to properly reconstruct the grass texture. Photoshop has artifacts because it copies the zebra's texture onto the grass. Our method produces consistent textures across the semantic regions.

As evident from the TOURIST scene which is outside both COCO-Stuff and CelebAHQ-Mask datasets, our method is also able to generalize well on novel scenes or scenes with object categories not featured in the training set. In cases where a real image contains segmentable objects, our approach utilizes that information and produces better results. However, our method does not fail in cases where an object cannot be segmented. We show additional results on a few additional general scenes in Fig 6. As shown, even thought some of the subjects (e.g., pinguins in FOLLOW) do not exist as a category in COCO-Stuff dataset, our method is able to generate plausible inpainted images.

4.2. Image Inpainting through User-Interaction

The semantic map predicted using the semantic estimation network can be edited by a user to obtain the desired results. In Fig. 1 the mask covers the hind legs of the zebras. This scene is fairly complex due to the texture patterns of the zebra's stripes and the Savannah grass. Additionally, the two zebras in the front occlude a third zebra. Without any user guidance our method is able to reconstruct the legs of the zebras with the correct texture pattern. With user guidance to refine the semantic map, our method improves the reconstruction and also synthesizes more accurate textures. Figure 7 (top) shows another example where our approach can properly reconstruct the giraffe's neck using user guidance. Figure 7 (bottom) shows another example where creative input from the user can be used to edit content to add facial features like the glasses. Compared to the user interactive version of DeepFillv2, our method produces better results as they provide guidance using only strokes.

Table 3: *Quantitative analysis of ablation studies.*

	COCO-Stuff		
Algorithm	FID$^{\downarrow}$	LPIPS$^{\downarrow}$	SSIM$^{\uparrow}$
Only Input Consistency Dis.	6.5190	0.0781	0.8830
Only Semantic Consistency Dis.	7.1655	0.0806	0.8850
Single Dis.	6.8230	0.0794	0.8874
Naive Generator	9.2227	0.0876	0.8807
No L1 Loss	5.7880	0.0774	0.8813
No VGG Loss	7.2282	0.0806	0.8843
No Feature Loss	5.7528	0.0755	0.8835
Ours (Complete)	**5.2907**	**0.0712**	**0.8883**

4.3. Ablation Studies

4.3.1. Dual Discriminators

We study the impact of the dual discriminators in Fig. 8. Each individual discriminator does not provide sufficient supervision to produce high-quality results. We observe that the input consistency discriminator (D_{in}) tends to meld mask seams to fuse the generated content and masked image. However, it does not learn semantic associations that are responsible for texture generation in the scene. In both the scenes, the results with only input consistency discriminator contains reasonable color information, but the textures are generally inconsistent.

The semantic consistency discriminator (D_{sem}) captures object context and associates texture in the scene to a given semantic label. However, it does not eliminate mask seams due to its inability to capture adequate color information. Combining the two discriminators allows us to both eliminate the mask seams and capture object context through the texture.

We also study the impact of using a single discriminator that takes as input both the mask and the semantic map along with the inpainted image in Fig. 9. In the CASTLE scene, the brick walls have undulating artifacts and pillar shows distortion. Our method captures the textured patterns of the wall as well as the shape of the pillar. Moreover, the single discriminator produces results with noticeable artifacts in the water (BEACH) and sky (PAGODA). Our approach with dual discriminators considerably reduces these artifacts. We also compare these various choices numerically in Table 3 and, as seen, our dual discriminator (Ours (Complete)) performs significantly better than the other discriminators.

4.3.2. Naïvely Integrating the Semantic Map

In the naïve approach, the semantic map is concatenated with the masked image as an input to the generator. Figure 10 shows the result of this experiment. In the CAGE scene, the naïve method is unable to reconstruct finer details like the wire frame of the cage. In the HERD scene, the naïve approach produces results with incorrect skin color and inconsistent stripes. In the GROUP scene, the naïve method does not maintain the door frame's shape. It also does not capture finer details on the man's face such as the eyes. Our method captures the shape of the door frame and also recovers the person's eye to a great extent.

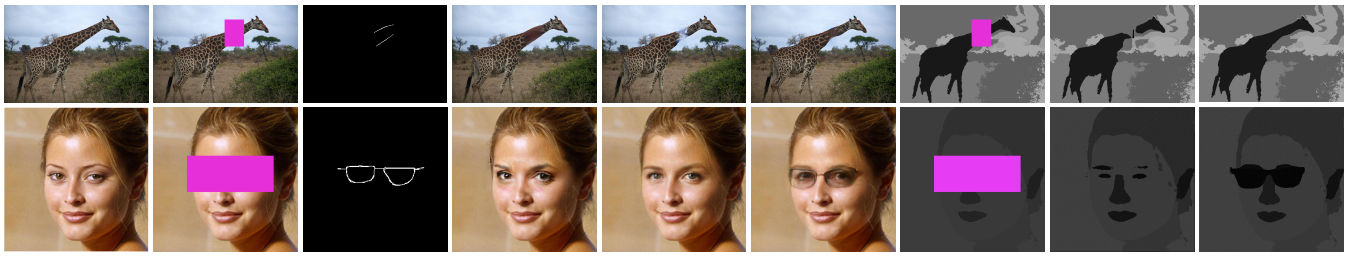

Figure 7: *Comparison against DeepFillv2 [YLY*19b] with user interaction.*

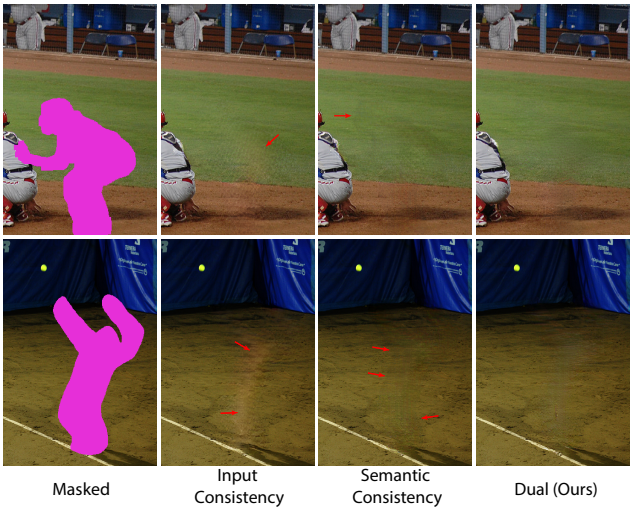

Figure 8: *Evaluating the effect of the two discriminators in the dual discriminator framework.*

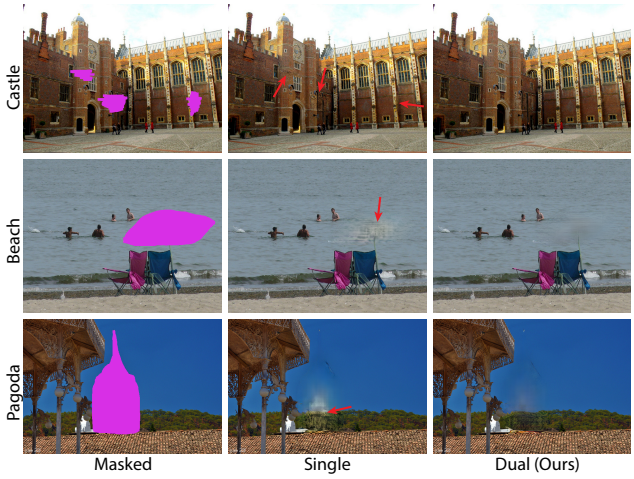

Figure 9: *Comparison between a combined discriminator and the dual discriminators (Ours). The combined discriminator is often unable to capture the color and textures appropriately.*

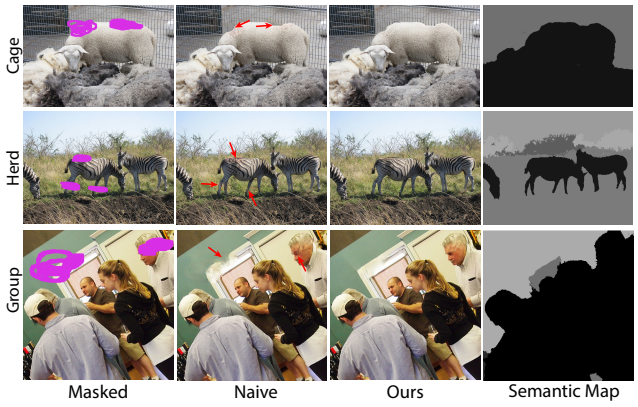

Figure 10: *Comparing our feature modulation approach with the naïve approach where we simply concatenate the semantic map as an input to the encoder.*

4.3.3. Effect of Loss Functions

We analyse the impact of the L_1, VGG-based perceptual loss, and the feature matching loss in Fig. 11. Without the L_1 loss, our method struggles to properly capture the color of the inpainted regions, producing visible seams. Moreover, our system without the VGG and feature matching losses is not able to properly reconstruct the textures and introduces minor artifacts. Our system with the full loss, can properly capture the color and reconstruct the textures.We also show the impact of each loss numerically in Table 3.

4.3.4. Total-Variation Loss

We use this loss in Eq. 4 to train the label inpainting network. As shown in Fig. 12, without TV loss the predicted segmentation maps are porous. The relative edges between semantic regions in the maps are also not preserved without TV loss. With the addition of the TV loss the porosity is eliminated in both cases. Furthermore, the shape of the arm is maintained in the top scene and the leg is fused with the torso in the bottom scene.

4.3.5. Evaluation of Label Generator

We perform a quantitative evaluation to ascertain how well the semantic segmentation inpaining is performed. This is evaluated by comparing the pixel accuracy of Deeplabv2 on the full image and our inpainted maps obtained from the Label generator. Deeplabv2 has a pixel accuracy of 67.48 compared to ours with a pixel accuracy of 66.85. This demonstrates that our inpainting network can produce comparable results to the segmentation map obtained by Deeplabv2 on the full image.

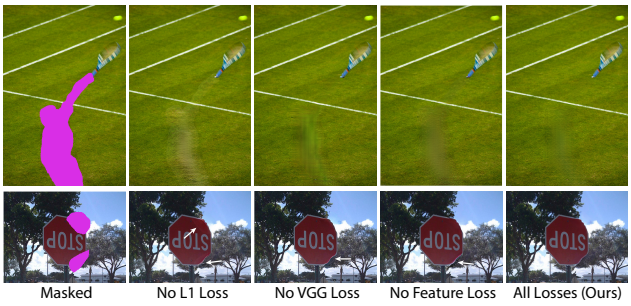

Figure 11: *Impact of each loss component on the inpainted image.*

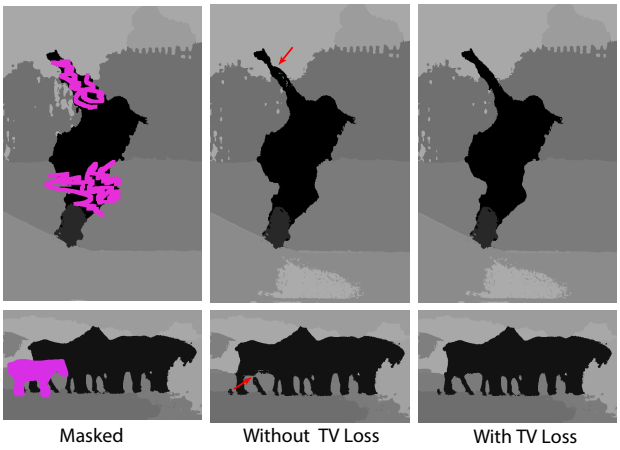

Figure 12: *Evaluating the effect of the total variation loss on the inpainted segmentation map.*

4.4. Limitations

Our approach has some limitations. First, our method relies on the estimated semantic maps to be able to properly inpaint the images. Although, we demonstrated that our system can tolerate inaccuracies in the estimated map, in cases where the off-the-shelf semantic estimation systems drastically fail, our approach would not provide significant benefit compared to existing deep learning methods. Moreover, since our user interaction is done on the semantic maps, the user can only control the shape of the inter-semantic boundaries. Finally, our method fail to produce high-quality results in cases where the class label on the semantic map and the object on the image do not have a correspondence. For example, in Fig. 13 the grass is occluded by a metal fence, but the semantic label for this region is "grass". Therefore, our approach is not able to reconstruct the fence, producing unsatisfactory results.

5. Conclusion

In conclusion, we propose a generative approach for image inpainting by incorporating semantic information through semantic-aware feature modulation. Specifically, we first generate a completed semantic map from the input image and propose to modulate the decoder features of our image inpainting network using parameters estimated from this semantic map. Furthermore, we propose to train our network with two discriminators; an input consistency discriminator which seamlessly fuses the generated and masked images and a semantic consistency discriminator which uses the semantic labels to develop contextual association with object textures and representations. Through extensive experiments we demonstrate significant improvement over state-of-the-art approaches and analyze the impact of each component of our proposed algorithm. We also show a method for user-guided inpainting using user provided completions of the segmentation maps.

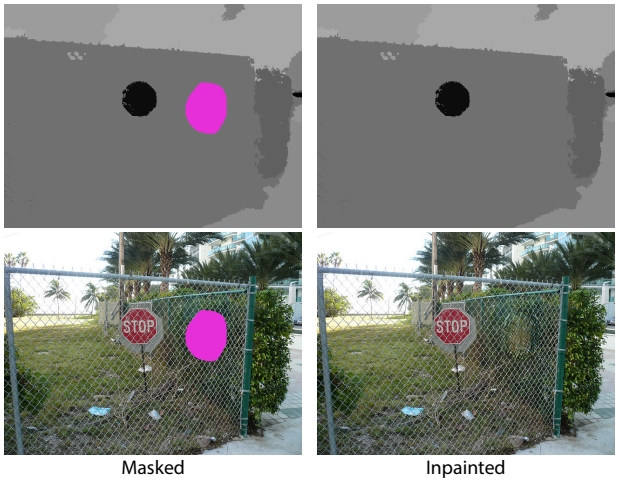

Figure 13: *We show a failure case for our approach. In this case, the fence is in front of the vegetation, but the semantic label for this entire region is grass. Therefore, our system is not able to properly reconstruct the fence, producing a result with artifacts.*

References

[BBC*01] BALLESTER, COLOMA, BERTALMIO, MARCELO, CASELLES, VICENT, et al. "Filling-in by joint interpolation of vector fields and gray levels". *IEEE Transactions on Image Processing* (2001) 2.

[BBS01] BERTALMIO, MARCELO, BERTOZZI, ANDREA L., and SAPIRO, GUILLERMO. "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2001 2.

[BSFG09] BARNES, CONNELLY, SHECHTMAN, ELI, FINKELSTEIN, ADAM, and GOLDMAN, DAN B. "PatchMatch: a randomized correspondence algorithm for structural image editing". *ACM Transactions on Graphics* (2009) 2.

[CPK*18] CHEN, LIANG-CHIEH, PAPANDREOU, GEORGE, KOKKINOS, IASONAS, et al. "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs". *IEEE Transactions Pattern Analysis Machine Intelligence* (2018) 2, 5, 6.

[CPT04] CRIMINISI, ANTONIO, PÉREZ, PATRICK, and TOYAMA, KENTARO. "Region filling and object removal by exemplar-based image inpainting". *IEEE Transactions on Image Processing* (2004) 2.

[CS01] CHAN, TONY F. and SHEN, JIANHONG. "Nontexture Inpainting by Curvature-Driven Diffusions". *Journal of Visual Communication and Image Representation* (2001) 2.

[CUF18] CAESAR, HOLGER, UIJLINGS, JASPER, and FERRARI, VITTORIO. "COCO-Stuff: Thing and Stuff Classes in Context". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 6.

[DSB*12] DARABI, SOHEIL, SHECHTMAN, ELI, BARNES, CONNELLY, et al. "Image melding: combining inconsistent images using patch-based synthesis". *ACM Transactions on Graphics* (2012) 2.

[dVSM*17] De Vries, Harm, Strub, Florian, Mary, Jérémie, et al. "Modulating early visual processing by language". *Advances in Neural Information Processing Systems, NeurIPS*. 2017 3.

[EL99] Efros, Alexei A. and Leung, Thomas K. "Texture Synthesis by Non-parametric Sampling". *Proceedings of the International Conference on Computer Vision (ICCV)*. 1999 2.

[GPM*14] Goodfellow, Ian, Pouget-Abadie, Jean, Mirza, Mehdi, et al. "Generative Adversarial Nets". *Advances in Neural Information Processing Systems, NeurIPS*. 2014 4.

[HE07] Hays, James and Efros, Alexei A. "Scene completion using millions of photographs". *ACM Transactions on Graphics* (2007) 2.

[HRU*17] Heusel, Martin, Ramsauer, Hubert, Unterthiner, Thomas, et al. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". *Advances in Neural Information Processing Systems, NeurIPS*. 2017 6.

[HZRS15] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian. "Deep Residual Learning for Image Recognition". *arXiv preprint arXiv:1512.03385* (2015) 4.

[ISI17] Iizuka, Satoshi, Simo-Serra, Edgar, and Ishikawa, Hiroshi. "Globally and Locally Consistent Image Completion". *ACM Transactions on Graphics* (2017) 2, 4.

[IZZE17] Isola, Phillip, Zhu, Jun-Yan, Zhou, Tinghui, and Efros, Alexei A. "Image-to-Image Translation with Conditional Adversarial Networks". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 3, 4.

[KB15] Kingma, Diederik P. and Ba, Jimmy. "Adam: A Method for Stochastic Optimization". *3rd International Conference on Learning Representations (ICLR)*. 2015 4.

[KEBK05] Kwatra, Vivek, Essa, Irfan, Bobick, Aaron, and Kwatra, Nipun. "Texture Optimization for Example-based Synthesis". *ACM Transactions on Graphics, SIGGRAPH* (Aug. 2005) 2.

[KKDK12] Kopf, Johannes, Kienzle, Wolf, Drucker, Steven Mark, and Kang, Sing Bing. "Quality prediction for image completion". *ACM Transactions on Graphics* (2012) 2.

[LJXY19] Liu, Hongyu, Jiang, Bin, Xiao, Yi, and Yang, Chao. "Coherent Semantic Attention for Image Inpainting". *IEEE International Conference on Computer Vision, ICCV*. 2019 3.

[LLWL20a] Lee, Cheng-Han, Liu, Ziwei, Wu, Lingyun, and Luo, Ping. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2020 3.

[LLWL20b] Lee, Cheng-Han, Liu, Ziwei, Wu, Lingyun, and Luo, Ping. "MaskGAN: Towards Diverse and Interactive Facial Image Manipulation". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020 6.

[LRS*18] Liu, Guilin, Reda, Fitsum A., Shih, Kevin J., et al. "Image Inpainting for Irregular Holes Using Partial Convolutions". *European Conferenece on Computer Vision (ECCV)*. 2018 2.

[LXW*20] Liao, Liang, Xiao, Jing, Wang, Zheng, et al. "Guidance and Evaluation: Semantic-Aware Image Inpainting for Mixed Scenes". *European Conference on Computer Vision - ECCV*. 2020 3.

[LY17] Lim, Jae Hyun and Ye, Jong Chul. "Geometric GAN". *CoRR* (2017) 4.

[LYS*19] Liu, Xihui, Yin, Guojun, Shao, Jing, et al. "Learning to Predict Layout-to-image Conditional Convolutions for Semantic Image Synthesis". *Advances in Neural Information Processing Systems, NeurIPS*. 2019 3.

[LZW03] Levin, Anat, Zomet, Assaf, and Weiss, Yair. "Learning How to Inpaint from Global Image Statistics". *IEEE International Conference on Computer Vision (ICCV*. 2003 2.

[MKKY18] Miyato, Takeru, Kataoka, Toshiki, Koyama, Masanori, and Yoshida, Yuichi. "Spectral Normalization for Generative Adversarial Networks". *6th International Conference on Learning Representations (ICLR)*. 2018 4.

[NNJ*19] Nazeri, Kamyar, Ng, Eric, Joseph, Tony, et al. "EdgeConnect: Generative Image Inpainting with Adversarial Edge Learning". *The IEEE International Conference on Computer Vision (ICCV) Workshops*. 2019 1–4, 6.

[PKD*16] Pathak, Deepak, Krähenbühl, Philipp, Donahue, Jeff, et al. "Context Encoders: Feature Learning by Inpainting". *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*. 2016 2, 4.

[PLWZ19] Park, Taesung, Liu, Ming-Yu, Wang, Ting-Chun, and Zhu, Jun-Yan. "Semantic Image Synthesis With Spatially-Adaptive Normalization". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 3.

[RYZ*19] Ren, Yurui, Yu, Xiaoming, Zhang, Ruonan, et al. "StructureFlow: Image Inpainting via Structure-Aware Appearance Flow". *IEEE International Conference on Computer Vision, ICCV*. 2019 3.

[SYS*18] Song, Yuhang, Yang, Chao, Shen, Yeji, et al. "SPG-Net: Segmentation Prediction and Guidance Network for Image Inpainting". *British Machine Vision Conference (BMVC)*. 2018 3, 6.

[WLT*19] Wang, Ting-Chun, Liu, Ming-Yu, Tao, Andrew, et al. "Few-shot Video-to-Video Synthesis". *Advances in Neural Information Processing Systems NeurIPS*. 2019 4.

[WLZ*18a] Wang, Ting-Chun, Liu, Ming-Yu, Zhu, Jun-Yan, et al. "High-Resolution Image Synthesis and Semantic Manipulation With Conditional GANs". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 3.

[WLZ*18b] Wang, Ting-Chun, Liu, Ming-Yu, Zhu, Jun-Yan, et al. "High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 4.

[WSI07] Wexler, Yonatan, Shechtman, Eli, and Irani, Michal. "Space-Time Completion of Video". *IEEE Transactions on Pattern Analysis Machine Intelligence* (2007) 2.

[WYDL18] Wang, Xintao, Yu, Ke, Dong, Chao, and Loy, Chen Change. "Recovering Realistic Texture in Image Super-Resolution by Deep Spatial Feature Transform". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 3.

[XYL*19] Xiong, Wei, Yu, Jiahui, Lin, Zhe, et al. "Foreground-Aware Image Inpainting". *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 3, 4.

[XZW*19] Xu, Bing, Zhang, Junfei, Wang, Rui, et al. "Adversarial Monte Carlo denoising with conditioned auxiliary feature modulation". *ACM Transactions on Graphics* (2019) 3.

[YLL*17] Yang, Chao, Lu, Xin, Lin, Zhe, et al. "High-Resolution Image Inpainting Using Multi-Scale Neural Patch Synthesis". *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 2.

[YLL*18] Yan, Zhaoyi, Li, Xiaoming, Li, Mu, et al. "Shift-Net: Image Inpainting via Deep Feature Rearrangement". *European Conference on Computer Vision (ECCV)*. 2018 2.

[YLY*18] Yu, Jiahui, Lin, Zhe, Yang, Jimei, et al. "Generative Image Inpainting With Contextual Attention". *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 2, 4.

[YLY*19a] Yu, Jiahui, Lin, Zhe, Yang, Jimei, et al. "Free-Form Image Inpainting With Gated Convolution". *IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019 2.

[YLY*19b] Yu, Jiahui, Lin, Zhe, Yang, Jimei, et al. "Free-form image inpainting with gated convolution". *IEEE International Conference on Computer Vision (CVPR)*. 2019 1, 2, 4–6, 9.

[YWP*18] Yu, Changqian, Wang, Jingbo, Peng, Chao, et al. "BiSeNet: Bilateral Segmentation Network for Real-Time Semantic Segmentation". *European Conference on Computer Vision (ECCV)*. Ed. by Ferrari, Vittorio, Hebert, Martial, Sminchisescu, Cristian, and Weiss, Yair. 2018 2, 6.

[ZAQW20] ZHU, PEIHAO, ABDAL, RAMEEN, QIN, YIPENG, and WONKA, PETER. "SEAN: Image Synthesis With Semantic Region-Adaptive Normalization". *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*. 2020 3.

[ZIE*18] ZHANG, RICHARD, ISOLA, PHILLIP, EFROS, ALEXEI A, et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 6.

[ZLY*20] ZENG, YU, LIN, ZHE, YANG, JIMEI, et al. "High-Resolution Image Inpainting with Iterative Confidence Feedback and Guided Upsampling". *European Conference on Computer Vision - ECCV*. 2020 2, 6, 7.