

# SECI-GAN: Semantic and Edge Completion for dynamic objects removal

Francesco Pinto

University of Oxford, UK  
francesco1.pinto@mail.polimi.it

Andrea Romanoni

Politecnico di Milano, Italy\*  
andrea.romanoni@polimi.it

Matteo Matteucci

Politecnico di Milano, Italy  
matteo.matteucci@polimi.it

Philip H.S. Torr

University of Oxford, UK  
philip.torr@eng.ox.ac.uk

**Abstract**—Image inpainting aims at synthesizing the missing content of damaged or corrupted images to produce visually realistic restorations; typical applications are in image restoration, automatic scene editing, super-resolution, and dynamic object removal. In this paper, we propose Semantic and Edge Conditioned Inpainting Generative Adversarial Network (SECI-GAN), an architecture that jointly exploits the high-level cues extracted by semantic segmentation and the fine-grained details captured by edge extraction to condition the image inpainting process. SECI-GAN is designed with a particular focus on recovering big regions belonging to the same object (e.g. cars or pedestrians) in the context of dynamic object removal from complex street views. To demonstrate the effectiveness of SECI-GAN, we evaluate our results on the Cityscapes dataset, showing that SECI-GAN is better than competing state-of-the-art models at recovering the structure and the content of the missing parts while producing consistent predictions.

## I. INTRODUCTION

Given an image with missing regions, image inpainting aims at synthesizing the missing content by producing a visually realistic and semantically coherent image. For this reason, it is also known in literature as image completion or image hole-filling [34], [22], [21], [8]. The missing pixels are assigned the value 0 in a binary mask, the non-missing pixels are assigned the value 1. The incomplete image is assumed to be generated by multiplying the image and the mask pixel-wise.

Early approaches apply diffusion-based methods to fill the missing regions by propagating image appearance around the holes [4], [1]. However, these methods only handle relatively small holes in regions with simple textures. Others, like [2], [14], [15], [36], [33], progressively fill the holes by copying patches from other parts of the image or from other images by using heuristic algorithms based on low-level feature-based similarity. These methods do not rely on a high-level understanding of the scene, thus they are not able to synthesize novel content based on the context surrounding the hole.

With the advent of Generative Adversarial Networks (GANs), Pathak *et al.* [20] showed that deep learning approaches could outperform the previous methods by learning how to encode a latent representation of the damaged image, and decode it into the corresponding inpainted image. Since then, other researchers built upon this basic architecture [12], [35], [34], [29], [25], [31], [9], [13] or exploited additional



Fig. 1. SECI-GAN is a novel inpainting neural network that can be effectively applied for dynamic object removal.

data to condition the inpainting process [19], [28], [30], [23]. Many efforts were dedicated to the solution of human faces inpainting [16] also exploiting geometry and semantics [32], [27], nevertheless, faces datasets present, in the training split, a large number of visually aligned images with similar structure and contents. When dealing with complex scenes, like those in urban city datasets, inpainting has to deal with relatively few examples in the training split and with a high variability. In this paper, we will focus on the latter kind of scenes.

To facilitate the inpainting process, methods like EdgeConnect [19] and SPGNet [28], perform a first step to inpaint, respectively, the edge image or the semantic segmentation of the damaged image and then they use the result to condition the actual image inpainting process. On the one hand, image edges capture fine details of the scene, on the other hand, semantics is illumination, scale, and view invariant and captures high-level cues. In this paper, we propose a novel architecture, named Semantic and Edge Conditioned Inpainting via Generative Adversarial Network (SECI-GAN) to exploit the benefits of both edges and semantics. Using both provides relevant information to improve the dynamic object removal performance from complex street view images as shown in Figure 1.

Dynamic object removal is crucial for vehicle Simultaneous Localization and Mapping (SLAM) or Urban 3D Reconstruction systems. For this reason, some of them [3], [24] explicitly neglect dynamic objects from the images to improve the reconstructed map of the environment. Instead, Dynaslam [5]

\*Work done prior to Amazon involvement of the author and does not reflect views of the Amazon company

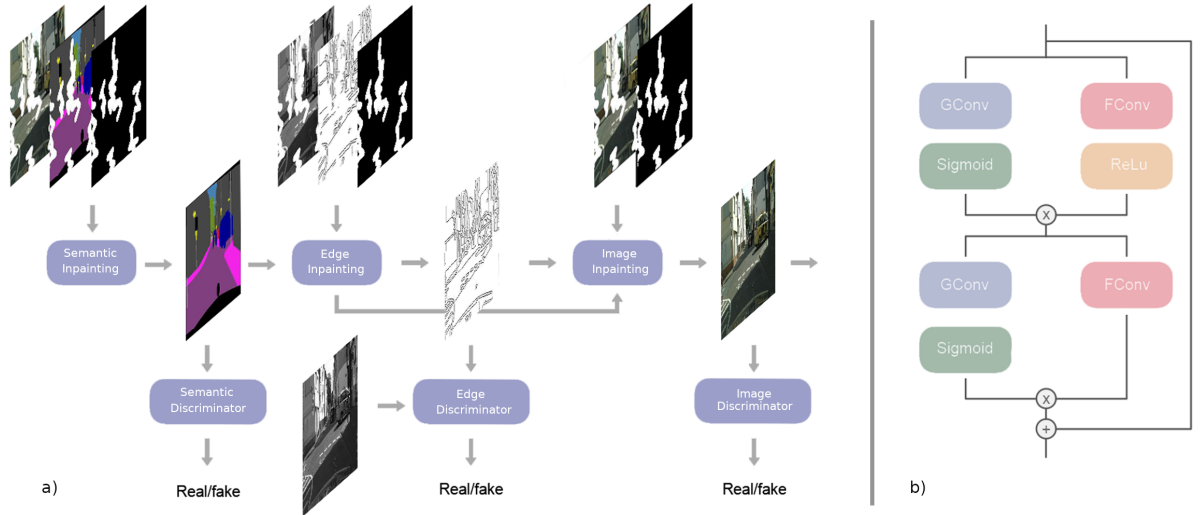


Fig. 2. a) The three networks of SECI-GAN are trained adversarially with their own generator. The edge discriminator network is also fed the gray-scale ground-truth image. b) The Gated ResNet blocks schema.

integrates a simple inpainting step into a SLAM pipeline and this is shown to lead to more robust mapping and tracking. However, the proposed method is not able to recover large missing regions as, in this instance of the inpainting problem, masks are usually big, and semantically homogeneous, since they belong to a certain class of objects, e.g., cars, pedestrian, motorcycles.

In this paper we show how SECI-GAN can effectively recover the content of large areas of images corresponding to dynamic objects. This could be beneficial for both the SLAM and 3D reconstruction communities, as well as for the image inpainting field as a whole. To summarize, the contributions proposed in this paper are the following:

- SECI-GAN is the first approach that jointly exploits the benefits of semantic and edge data to condition the inpainting process
- We show extensive ablation studies to highlight the contribution of each component of SECI-GAN, to better understand their impact on the overall performance
- We demonstrate that image inpainting can be effectively used to perform dynamic object removal from single images representing complex scenes.

## II. RELATED WORKS

The first approach exploiting GANs for image inpainting was proposed by Pathak et al. [20]. Their Context Encoders, originally intended to unsupervisedly learn semantic visual features, are Convolutional Neural Networks (CNNs) trained to generate the content of damaged image regions, conditioned on their surroundings. To learn a feature representation capturing both appearance and semantics they adversarially train a generator and discriminator using a linear combination of a reconstruction loss and an adversarial loss.

The damaged regions of an image are generally represented with a binary mask that is used to zero out some parts of the

image. Since replacing such parts with zeros could negatively affect the inpainting, especially when handling large holes, Liu et al. [17] explicitly prevent the network from estimating pixel values based on these placeholder values through partial convolutions with a mask update step. Yu et al. [34] improved this technique by replacing the partial convolution with a gated convolution: the mask and the damaged image are the input of two parallel convolutional layers, one to learn the features and one to learn their weights, whose outputs are multiplied per-pixel. This mechanism allows the network to give a different importance to the features extracted from the input. The authors of DeepFillv2 [35], [34] opted for a coarse-to-fine approach; a first coarse inpainting network produces a blurry estimate, then, the refinement inpainting network synthesizes high frequency details starting from the coarse estimate.

Other authors tried to expand receptive fields to learn spatial correlations across large regions. For instance, Iizuka et al. [12] proposed to increase the receptive field of inner layers while keeping constant the number of trainable parameters via dilated convolutions. One possible limitation is that dilated convolutions are restricted to spreading the convolution sampling grid on a sparse, fixed-shape grid. Yu et al. [35] implemented a contextual attention mechanism that synthesises features drawing information from patches coming from the whole image. While it draws fine patterns from distant parts of the input image, it has the drawback of high memory requirements, and hence it is inappropriate to directly handle high-resolution images.

More recent methods exploit additional data to condition the inpainting process. For instance, Yu et al. [34] show how hand-drawn sketches could be used to guide the inpainting process. Subsequently, SPG-Net [28] repairs the semantic map extracted from the damaged image via a semantic inpainting network, which is exploited to condition the image inpainting

network. The experiments they run were limited to just a subset of the Cityscapes dataset, exploiting small and rectangular-shaped masks. As our experiments point out, the semantic information alone is not enough to properly reproduce fine details on large holes. Indeed, the semantic map can only capture coarse geometrical information which is not enough to reconstruct the spatial arrangement of tiny details. Specularly, EdgeConnect [19] repairs the edge map of the damaged image, then it exploits edges during the image inpainting process, achieving current state-of-the-art performance. Edges represent a better source of data to reproduce fine details, nevertheless, as shown in our experiments, the edge completion network is not suited for handling large masks on complex scenes.

### III. ARCHITECTURE

SECI-GAN leverages high-level information carried by semantics, and fine cues coming from image edges to condition the inpainting procedure. To this end, the architecture is split into three components: (i) semantic inpainting, (ii) edge inpainting and (iii) image inpainting. We employ one inpainting network for each task and combine them into a single end-to-end model (see Figure 2a).

The architecture of the three inpainting networks draws inspiration from the auto-encoder proposed in [12]. We replace each convolutional layer with the gated convolutional layer proposed in [34]. Moreover, the middle layers are replaced with Gated ResNet blocks. The Gated ResNet block is a ResNet block variant made by a sequence of two gated convolutional layers where the output of the second convolutional layer is summed to the input of the first layer (see Figure 2b). Intuitively, the ResNet blocks reduce the degradation of the information about input patterns across deep layers [10], [9]. Combining them with a gating mechanism allows us to better preserve this information while weighing it differently based on the conditioning inputs. Finally, we use Instance Normalization instead of classic Batch Normalization as in [34], and ReLU activation functions.

The semantic inpainting network is fed the RGB damaged image, the damaged semantic map and the mask. Its last layer is a softmax layer, so as to predict the pixel-wise probability for each class. The edge inpainting network is fed the output of the semantic inpainting network, the gray-scale damaged image, the damaged edge map, and the mask. Each convolutional layer employs spectral normalization [18], as it accelerates and stabilizes the training. The last layer of the the edge inpainting network is followed by a sigmoid layer, to estimate whether a pixel is an edge or not.

In addition to the predicted edge map, we also add (via summation) the *semantic edges* extracted through the Canny Edge detector [6] from the reconstructed semantic map (i.e. the object outlines in the reconstructed semantic map). The result of this operation is the so called *enhanced edge map*, or *enhanced edges*. Adding the semantic edges to the edge map extracted from the gray-scale image generally improves the inpainting performance. Indeed, especially in the dynamic object removal scenario, semantic maps are more likely to

TABLE I  
STRUCTURE OF NETWORK COMPONENTS. O REPRESENTS THE OUTPUT CHANNEL SIZE, K THE KERNEL SIZE, S THE SCALE FACTOR, P THE PADDING

Generators					Discriminators					ResNet				
Layer Number	O	K	S	P	Layer Number	O	K	S	P	Layer Number	O	K	S	P
1	64	7	1	0	1	64	3	2	1	1	256	3	1	0
2	128	4	2	1	2	64	3	2	1	2	256	3	2	1
3	256	4	2	1	3	128	3	2	1					
8 × Gated ResNet					4	256	3	2	1					
4	128	3	2	1	5	512	3	1	1					
5	64	3	2	1	6	1	3	1	1					
6	C	7	1	0										

be properly repaired than edges. Finally, the image inpainting network is fed both the reconstructed semantic map, the enhanced edges, the RGB damaged image and the mask. The activation function of the final layer of the decoder is a hyperbolic tangent, followed by a normalization step to bring its output in the range  $[0, 1]$ .

The discriminators have all the same architecture. They are made by a sequence of six spectrally normalized [18] convolutional layers with a stride equal to 2.

### IV. TRAINING

Generators are pre-trained individually with their own discriminator. If a generator is fed some data that should be produced by a previous generator in the pipeline, this data is replaced with the ground-truth as far as the individual training is concerned. When the networks reach the convergence, they are connected in the complete architecture of Figure 2a, and the training is completed in an end-to-end fashion. All the networks have been trained using the ADAM optimizer. The learning rate has been set to  $1e-4$  for the generator and to  $1e-6$  for the discriminator, the parameters  $\beta_1$  and  $\beta_2$  have been set to 0.0 and 0.9, respectively. Drawing inspiration from EdgeConnect, the weights in the loss functions of Equations (1), (2) and (3) have been set to  $\alpha_s = \alpha_e = 10$ ,  $\alpha_i = 1$ ,  $\delta_i = \beta_s = 0.1$  and  $\beta_e = 1$ ,  $\gamma_i = 0.1$  and  $\beta_i = 0.1$ .

*Semantic Inpainting:* Let  $I_s = G_s(I_{in}, S_{in}, M_{in})$  be the output of the semantic generator given the damaged input image  $I_{in}$ , the damaged semantic map<sup>1</sup>  $S_{in}$  and the mask  $M_{in}$ . Let us define  $D_s(S)$  the output of the discriminator  $D_s$  given a semantic map  $S$ . We define the loss:

$$\mathcal{L}_{sem} = \alpha_s \mathcal{L}_{FM_s} + \beta_s \max_{D_s} \mathcal{L}_{adv_s}, \quad (1)$$

where  $\mathcal{L}_{FM_s} = \mathbb{E} \left[ \sum_{i=1}^L \|D_s^{(i)}(I_s) - D_s^{(i)}(S_{gt})\|_1 \right]$ , and  $\mathcal{L}_{adv_s} = \mathbb{E}[\log D_s(S_{gt})] + \mathbb{E}[\log(1 - D_s(I_s))]$ .  $S_{gt}$  represents the undamaged semantic map corresponding to  $S_{in}$ , while  $\alpha_s$  and  $\beta_s$  are importance weights to be empirically determined.  $D_s^{(i)}$  represents the output of the  $i$ -th layer among the all the  $L$  layers of the discriminator. Thus  $\mathcal{L}_{FM_s}$  measures the distance between the features produced by the  $i$ -th layer when processing the generated semantic map and the ground-truth

<sup>1</sup> $S_{in}$  has been obtained by applying the mask to the ground truth annotations of the Cityscapes dataset

semantic map. This kind of loss is called feature matching loss in literature [19].

*Edge Inpainting:* Let  $I_{edge} = G_e(Gr_{in}, E_{in}, M_{in})$  be the output of the edge generator given the damaged gray-scale input image  $Gr_{in}$ , the damaged edge map<sup>2</sup>  $E_{in}$  and the mask  $M_{in}$ . Let us define  $D_e(Gr, E)$  the output of the discriminator given an edge map  $E$  and its corresponding gray-scale image  $Gr$ . The loss is:

$$\mathcal{L}_{edge} = \alpha_e \mathcal{L}_{FM_e} + \beta_e \max_{D_e} \mathcal{L}_{adv_e}, \quad (2)$$

where  $\mathcal{L}_{FM_e} = \mathbb{E}[\sum_{i=1}^L ||D_e^{(i)}(Gr_{gt}, I_{edge}) - D_e^{(i)}(Gr_{gt}, E_{gt})||_1]$ , and  $\mathcal{L}_{adv_e} = \mathbb{E}[\log D_e(Gr_{gt}, E_{gt})] + \mathbb{E}[\log(1 - D_e(Gr_{gt}, I_{edge}))]$ . where  $E_{gt}$  represents the undamaged edge corresponding to  $E_{in}$ ,  $Gr_{gt}$  represents the undamaged gray-scale image corresponding to  $Gr_{in}$ ,  $\alpha_e$  and  $\beta_e$  are importance weights to be empirically determined.  $D_e^{(i)}$  represents the output of the  $i$ -th layer of the discriminator.

*Image Inpainting:* Let  $I_{gen} = G_i(I_{in}, S_{in}, E_{in}, M_{in})$  be the output of the image inpainting generator given the damaged input image  $I_{in}$ , the inpainted semantic map  $S_{in}$ , the inpainted edge map  $E_{in}$  and the mask  $M_{in}$ . Let us define  $D_i(I)$  the output of the discriminator given an image  $I$ . The image inpainting loss is:

$$\mathcal{L}_{inp} = \alpha_i \mathcal{L}_{L1} + \beta_i \mathcal{L}_{perc} + \gamma_i \mathcal{L}_{style} + \max_{D_i} \delta_i \mathcal{L}_{adv_i}, \quad (3)$$

where

$$\mathcal{L}_{L1} = \mathbb{E}[||I_{gen} - I_{gt}||_1], \quad (4)$$

$$\mathcal{L}_{perc} = \mathbb{E}\left[\sum_{p=0}^{P-1} \frac{||\Psi_p(I_{gen}) - \Psi_p(I_{gt})||_1}{N_{\Psi_p(I_{gt})}}\right], \quad (5)$$

$$\mathcal{L}_{style} = \mathbb{E}\left[\sum_{p=0}^{P-1} ||(\Psi_p(I_{gen}))^T (\Psi_p(I_{gen})) - (\Psi_p(I_{gt}))^T (\Psi_p(I_{gt}))||_1\right], \quad (6)$$

$$\mathcal{L}_{adv_i} = \mathbb{E}[\log D_i(I_{gt})] + \mathbb{E}[\log(1 - D_i(I_{gen}))]. \quad (7)$$

$I_{gt}$  represents the ground-truth undamaged image corresponding to  $I_{in}$ , while,  $\alpha_i$ ,  $\beta_i$ ,  $\gamma_i$  and  $\delta_i$  are importance weights to be empirically determined.  $\Psi_p(I)$  represents the activation maps from the layers of a pre-trained model (in our case, we took the `relu1_1`, `relu2_1`, `relu3_1`, `relu4_1` and `relu5_1` of VGG-19 [26]) and  $N_{\Psi_p(I)}$  represents their size.

#### A. Mask Generation

To achieve better generalization to unknown mask shapes, at training time, we randomly generate free-form masks employing a variation of the algorithm of Yu *et al.* [34]. Our algorithm draws a random number of strokes, whose shape is, like in Yu *et al.* [34], made by a continuous sequence of segments whose vertices are smoothed. The length and width of these segments are random too. The algorithm depends on

<sup>2</sup> $E_{in}$  has been obtained by applying the Canny Edge detector on  $Gr_{in}$

TABLE II  
PARAMETERS FOR MASK GENERATION

Parameters	Test 256 × 256 images		
	$M_{small}$	$M_{medium}$	$M_{large}$
maxVertex	30	30	50
maxLength	30	30	30
maxBrushWidth	15	25	30
minBrushWidth	5	10	10
maxAngle	2	2	4
maxNumStrokes	10	10	10
minNumStrokes	4	4	4

some parameters. At test time, we employed three masks sets  $M_{small}$ ,  $M_{medium}$  and  $M_{large}$ , whose parameters are reported in Table II. For training we choose the  $M_{large}$  parameters.

#### V. EXPERIMENTS

We evaluate SECI-GAN on Cityscapes [7], a commonly used computer vision dataset containing street view images. It contains 5000 densely annotated images, with a rich set of semantics labels. The training and the evaluation splits consist, respectively, of 2975 and 500 RGB images with the corresponding semantic maps. Since the ground-truth semantic annotations of the additional test split is not publicly available, we used the evaluation split as test set for final evaluation. Before being fed to the networks, we rescaled the images to  $256 \times 256$  resolution. Flipping along the horizontal direction has been applied randomly as a form of data augmentation.

When considering the task of dynamic object removal, we remove the objects according to their semantic label. We split the labels into two disjoint subsets, so that each class can be either dynamic or static. Given a semantic map, a mask for dynamic object removal (we will call it dynamic mask) can be obtained by setting to 1 all the pixels classified as belonging to a dynamic class. We only consider the semantic label to classify an object as dynamic or not dynamic, since it is the most trustworthy information that can be leveraged to this purpose in single images. We leave the extension of the method to videos, and the exploitation of movement cues for this purpose to future research.

##### A. Inpainting Results

We performed an objective evaluation by means of the PSNR, SSIM [38], FID [11], LPIPS [37] metrics. The first three are commonly used in the inpainting literature, even though the PSNR is not representative of the actual perceptual quality of the inpainting [38]. We also use LPIPS, a recent metric which has been observed to be more consistent with the way human observers perceive the similarity between images. The results are reported in Table III.

They show that SECI-GAN has slightly better performance than EdgeConnect [19], and both EdgeConnect and SECI-GAN remarkably outperform DeepFillv1 and DeepFillv2.

##### B. Dynamic Object Removal Results

Assessing the quality of the dynamic object removal results is extremely difficult. Indeed, to the best of our knowledge, no dataset is available that presents an explicit and exact mapping

	$M_{small}$				$M_{medium}$				$M_{large}$			
	EC	Ours	DF2	DF1	EC	Ours	DF2	DF1	EC	Ours	DF2	DF1
PSNR $\uparrow$	32.05	<b>32.20</b>	29.75	28.30	28.81	<b>28.95</b>	26.85	25.58	24.59	<b>24.62</b>	23.24	21.21
SSIM $\uparrow$	96.97	<b>97.09</b>	94.51	92.47	93.83	<b>94.00</b>	89.51	86.63	<b>88.95</b>	88.89	83.23	78.80
FID $\downarrow$	12.63	<b>12.32</b>	16.34	24.45	<b>20.13</b>	20.14	26.43	30.85	31.91	<b>31.66</b>	35.20	46.08
LPIPS $\downarrow$	3.60	<b>3.58</b>	5.40	7.70	6.85	<b>6.68</b>	9.24	11.66	<b>11.12</b>	11.37	13.67	16.87

TABLE III

COMPARISON BETWEEN EDGECONNECT (EC) [19], DEEPFILLV1 (DF1) [35], DEEPFILLV2 (DF2) [34] AND SECI-GAN (OURS) FOR IMAGE INPAINTING WITH MASKS SETS OF DIFFERENT SIZE.

between dynamic and static images. Hence, no available objective metric is capable of reliably measuring the quality of dynamic object removal.

Nevertheless, it is possible to observe that, concerning the task of dynamic object removal, the addition of the semantic conditioning input results fundamental to achieve better performance. The reason is that, generally speaking, the repaired semantic masks tend to produce plausible results and provide a guidance both in terms of plausible enhanced edges and plausible semantic content. This allows to better capture the structure and the color of the missing parts. On the other hand, EdgeConnect tends to produce more blurred, noisy and unstructured predictions. Indeed, trying to reconstruct the edges of large parts is hard, and, without relying on semantics, the edge inpainting network produces wrong results that cause relevant deformations and artifacts in the output image. This phenomenon is alleviated in SECI-GAN.

It should be noticed that, since the semantic inpainting network is not trained explicitly for dynamic object removal, it might hallucinate the presence of dynamic patches. This issue can be alleviated by iteratively applying the semantic inpainting process, each time extracting the dynamic mask from the resulting inpainted semantic map. Without dynamic classes in the resulting semantically inpainted map, we can prevent the image inpainting network from hallucinating them. This control of the content, based on the semantic map, is an advantage of conditioning the inpainting process on semantics.

A drawback of inpainting for dynamic object removal performed using semantic masks is that it does not remove the shadows of the removed object. A simple solution of this issue could be to annotate the shadows and add the pixels classified as shadows to the mask. Then our method could be applied without modifications. Preliminary experiments with mask dilation show that removing shadows can improve the ability of SECI-GAN to recover the background structure; the same does not seem to hold for EdgeConnect (as shown in Figure 4). Nevertheless, dilation is a naive, aggressive approach to shadow removal, and its use can destroy relevant information for the inpainting process.

### C. Dynamic Object Removal User Test

Since there is no objective way to assess the improvements of SECI-GAN over EdgeConnect, DeepFillv1 and DeepFillv2 concerning the task of dynamic object removal, we performed a user study (with 100 participants) showing 60 ground-truth images, the corresponding dynamic masks, and the output of

TABLE IV

SUMMARY OF THE RESULTS OF OUR USER TEST FOR DYNAMIC OBJECT REMOVAL FOR THE COMPARISON OF SECI-GAN WITH EDGECONNECT (EC), DEEPFILLV1 (DFv1) AND DEEPFILLV2 (DFv2). THE FIRST ROW OF EACH TABLE REPORTS THE NUMBER OF QUESTIONS, OUT OF 60, FOR WHICH SECI-GAN WAS JUDGED, RESPECTIVELY, TO BE BETTER THAN THE OTHER METHOD, WORSE, OR SIMILAR BY COUNTING THE NUMBER OF VOTES. ON THE SECOND ROW OF EACH TABLE, WE REPORT THE RESULTS OF THE BERNOULLI PROPORTION TEST, REPRESENTED BY THE NUMBER OF TIMES THE BERNOULLI STATISTICAL TEST SUPPORTS THE JUDGMENT WITH STRONG EVIDENCE, I.E., WITH A P-VALUE  $< 0.05$ .

SECI-GANvsEC		>	<	=
Maj. Votes	<b>53</b>	7	0	
Stat. Test	<b>51</b>	5	4	
SECI-GANvsDFv1		>	<	=
Maj. Votes	<b>60</b>	0	0	
Stat. Test	<b>59</b>	0	1	
SECI-GANvsDFv2		>	<	=
Maj. Votes	<b>60</b>	0	0	
Stat. Test	<b>59</b>	0	1	

EdgeConnect, SECI-GAN, DeepFillv1, and DeepFillv2. The test participants were asked to express which of the output they believed to be the most plausible. The test results are summarized in Table IV. The majority of votes has been assigned to SECI-GAN in all the comparisons against EdgeConnect (EC), DeepFillv1 (DFv1) and DeepFillv2 (DFv2). To assess the statistical relevance of the results, we performed a hypothesis test. Denoting with  $p$  the probability of voting SECI-GAN, we checked whether SECI-GAN was significantly better than the other method by setting, for each question,  $H_0 : p \leq 0.5$  and  $H_1 : p > 0.5$  and performing the Bernoulli proportion hypothesis test. Similarly, we checked whether the other method was significantly better than SECI-GAN by setting, for each question,  $H_0 : p \geq 0.5$  and  $H_1 : p < 0.5$ . In Table IV we count the number of questions for which these hypotheses were rejected, determining the superiority, inferiority and similarity of SECI-GAN to the other methods. In all the comparisons, SECI-GAN resulted to perform significantly better than the competitors<sup>3</sup>.

## VI. ABLATION STUDIES

In this section we analyze both the impact of the gating mechanism and the effectiveness of different combinations of

<sup>3</sup>Datasets used for user study validation are publicly available at <http://airlab.deib.polimi.it/datasets-and-tools/>.



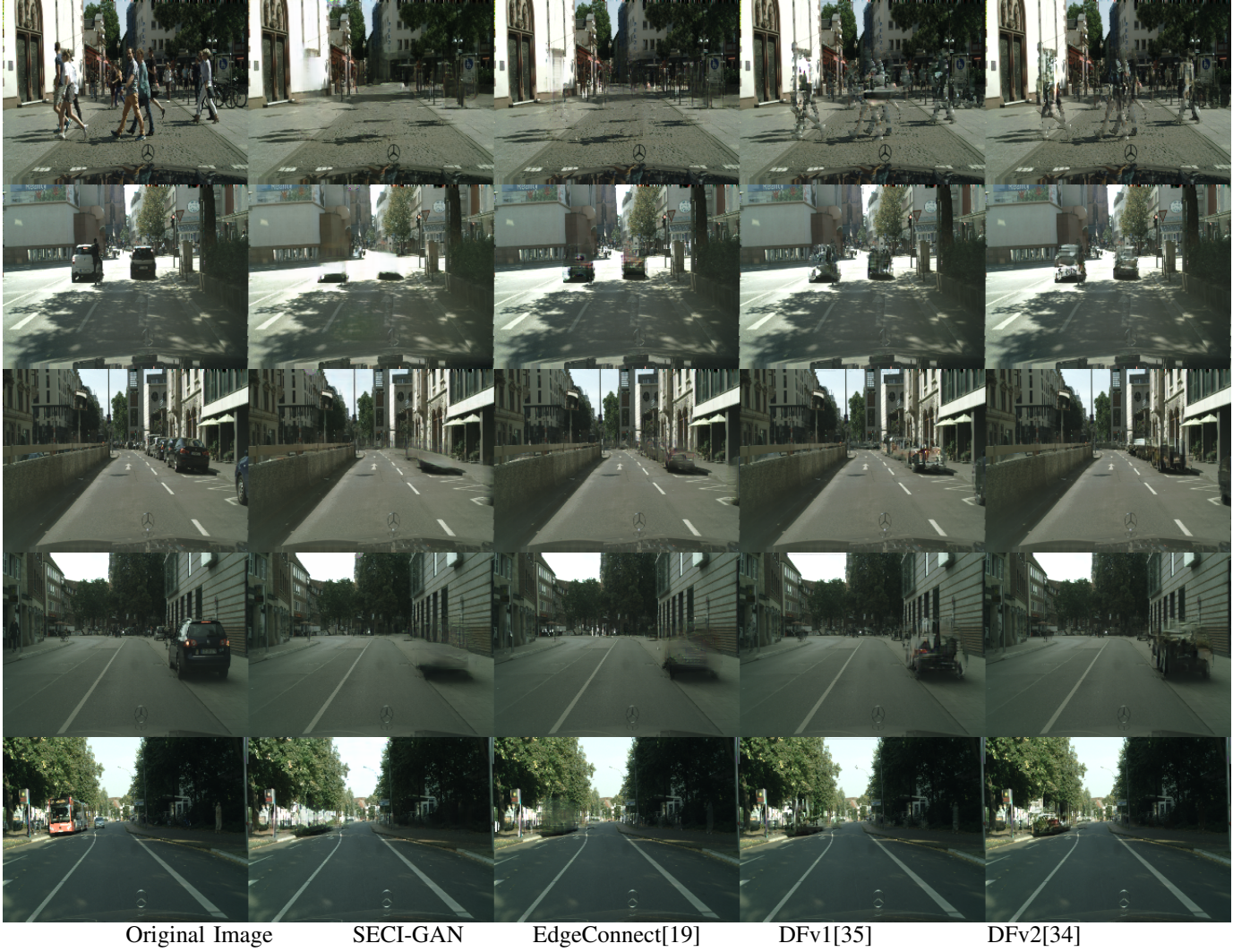


Fig. 3. Comparison for dynamic object removal (zoom in for details).

TABLE V

ACCURACY AND MEAN INTERSECTION OVER UNION (mIoU) FOR THE SEMANTIC INPAINTING NETWORK WITH GATING (G), WITHOUT GATING (noG).

	$M_{small}$		$M_{medium}$		$M_{large}$	
	Acc	mIoU	Acc	mIoU	Acc	mIoU
noG	99.20	95.28	98.15	91.12	95.95	84.30
G	<b>99.28</b>	<b>95.62</b>	<b>98.34</b>	<b>91.57</b>	<b>96.38</b>	<b>84.60</b>

TABLE VI

COMPARISON BETWEEN SECI-GAN WITHOUT GATING (noG), WITH GATING (G) AND EDGECONNECT (EC).

	$M_{small}$			$M_{medium}$			$M_{large}$		
	noG	G	EC	noG	G	EC	noG	G	EC
PSNR↑	34.19	<b>34.36</b>	34.09	31.58	<b>31.76</b>	31.38	27.70	<b>27.76</b>	27.43
SSIM↑	98.22	<b>98.33</b>	98.17	96.68	<b>96.92</b>	96.56	94.12	<b>94.51</b>	93.92
FID↓	6.74	<b>6.00</b>	6.68	10.61	<b>9.57</b>	11.11	15.03	<b>13.37</b>	16.06
LPIPS↓	2.23	<b>2.07</b>	2.38	3.84	<b>3.45</b>	4.13	5.99	<b>5.45</b>	6.69

conditioning data (i.e, edges and semantics).

a) *Semantic Inpainting Network*: Table V shows that the semantic inpainting subnetwork with gating attains better performance than the one without gating. In this context we cannot perform a direct quantitative comparison with the semantic inpainting network of SPGNet [28] since the source code is not available. However, our network achieves qualitatively good performance on more complex masks after only 100 training epochs, against the 200 epochs required by the semantic inpainting generator of SPGNet.

b) *Edge Inpainting Network*: We trained our edge inpainting network both with and without gating, with and without semantic conditioning of the input, with and without spectral normalization in the generator layers. In all the cases we considered, the edge inpainting networks struggle to produce realistic results. We can conjecture this is due to the imbalance between the proportion of pixels classified as being edge or non-edge, the complexity of the scene and of the



Fig. 4. Comparison between SECI-GAN and EdgeConnect using (wDil) or not using dilated masks.

TABLE VII  
COMPARISON BETWEEN THE IMAGE INPAINTING NETWORK WITHOUT EDGE AND SEMANTIC CONDITIONING (NOC), WITH ONLY SEMANTIC CONDITIONING (CS), WITH ONLY EDGE CONDITIONING (CE), AND WITH BOTH (CSE).

	$M_{small}$				$M_{medium}$				$M_{large}$			
	noc	cS	cE	cSE	noc	cS	cE	cSE	noc	cS	cE	cSE
PSNR $\uparrow$	31.64	32.37	34.17	<b>34.36</b>	28.86	29.60	31.51	<b>31.76</b>	24.61	25.76	27.64	<b>27.76</b>
SSIM $\uparrow$	96.91	97.28	98.22	<b>98.33</b>	93.94	94.76	96.66	<b>96.92</b>	88.95	90.79	94.11	<b>94.51</b>
FID $\downarrow$	12.85	10.83	6.62	<b>6.00</b>	20.20	16.25	9.96	<b>9.57</b>	28.48	23.19	14.45	<b>13.37</b>
LPIPS $\downarrow$	3.93	3.32	2.20	<b>2.07</b>	7.05	5.80	3.67	<b>3.45</b>	11.03	8.93	5.72	<b>5.45</b>

considered masks, the dependence of the edges from lighting conditions, viewpoint, scale and resolution. The results produced by the edge inpainting network of EdgeConnect (similar to ours without semantic edges) for the task of dynamic object removal are generally poor. On the contrary, when semantic edges enhancement is introduced, the quality of the reconstructed edges improves. This way, our network is capable of handling larger and more complex masks. Nevertheless, it is not possible to make a real quantitative comparison between the two methods.

*c) Image Inpainting Network:* In Table VII we show the impact of conditioning on different data sources (none, semantics only, edges only, and both). The tests have been performed with ground truth conditioning inputs, so as to disentangle their impact from the quality achieved by the other networks. The figures clearly support the idea that exploiting both semantics and edge data helps improving the inpainting performance. They also evidence that edge data consistently contribute to a remarkable improvement with respect to the absence of conditioning inputs.

In Table VI we show that gating improves the performance with respect to the case without gating. Comparing the results reported in Tables III and VII, we can also notice that when the true edges are used, the results get better with respect to the case with inpainted edges. This is caused by sometimes implausible outputs from the edge inpainting network.

## VII. CONCLUSIONS

In this paper we presented SECI-GAN, a novel inpainting neural network, the first exploiting both semantic and edge data to condition the inpainting process. The problem of inpainting is split into three subproblems (semantic, edge and image inpainting), each addressed by a subnetwork. We showed that the joint contribution of the inpainted semantic maps and edge maps yields state-of-the-art performance for dynamic object removal on a complex street view dataset like Cityscapes [7]. Our experiments pointed out that the convergence of an edge inpainting network on a complex street view dataset is problematic. While the network is capable of filling small holes, it produces wrong edges when dealing with large holes. We alleviated this issue using edge maps enhanced with the outlines from the semantic map.

Finally, we showed that SECI-GAN yields a significant improvement over previously existing techniques for dynamic object removal; nevertheless, our method is still incapable of removing the shadows casted by the dynamic objects. Beside this, in view of combining dynamic object removal with Stereo 3D reconstruction and Visual SLAM systems, we will extend our method to work on stereo video sequences and we will also investigate how to exploit information from projective geometry to improve the inpainting performance.

## REFERENCES

- [1] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels,"

- Trans. Img. Proc.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001. [Online]. Available: <http://dx.doi.org/10.1109/83.935036>
- [2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “Patch-Match: A randomized correspondence algorithm for structural image editing,” *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.
  - [3] I. A. Barsan, P. Liu, M. Pollefeys, and A. Geiger, “Robust dense mapping for large-scale dynamic environments,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) 2018*. IEEE, May 2018.
  - [4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’00. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 2000, pp. 417–424. [Online]. Available: <http://dx.doi.org/10.1145/344779.344972>
  - [5] F. J. C. J. Bescos, Berta and J. Neira, “DynaSLAM: Tracking, mapping and inpainting in dynamic environments,” *IEEE RA-L*, 2018.
  - [6] J. Canny, “A computational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Jun. 1986. [Online]. Available: <https://doi.org/10.1109/TPAMI.1986.4767851>
  - [7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
  - [8] C. Guillemot and O. Le Meur, “Image inpainting : Overview and recent advances,” *Signal Processing Magazine, IEEE*, vol. 31, pp. 127–144, 01 2014.
  - [9] Z. Guo, Z. Chen, T. Yu, J. Chen, and S. Liu, “Progressive image inpainting with full-resolution residual network,” in *Proceedings of the 27th ACM International Conference on Multimedia*, ser. MM ’19. New York, NY, USA: ACM, 2019, pp. 2496–2504. [Online]. Available: <http://doi.acm.org/10.1145/3343031.3351022>
  - [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.
  - [11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 6626–6637.
  - [12] S. Iizuka, E. Simo-Serra, and H. Ishikawa, “Globally and Locally Consistent Image Completion,” *ACM Transactions on Graphics (Proc. of SIGGRAPH)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.
  - [13] L. Jiao, H. Wu, H. Wang, and R. Bie, “Multi-scale semantic image inpainting with residual learning and gan,” *Neurocomputing*, vol. 331, 11 2018.
  - [14] J. Kopf, C.-W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, and T.-T. Wong, “Solid texture synthesis from 2d exemplars,” *ACM Trans. Graph.*, vol. 26, no. 3, Jul. 2007. [Online]. Available: <http://doi.acm.org/10.1145/1276377.1276380>
  - [15] S. Lefebvre and H. Hoppe, “Parallel controllable texture synthesis,” *ACM TRANSACTIONS ON GRAPHICS*, pp. 777–786, 2005.
  - [16] Y. Li, S. Liu, J. Yang, and M.-H. Yang, “Generative face completion,” in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
  - [17] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, “Image inpainting for irregular holes using partial convolutions,” in *The European Conference on Computer Vision (ECCV)*, 2018.
  - [18] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, “Spectral normalization for generative adversarial networks,” *ArXiv*, vol. abs/1802.05957, 2018.
  - [19] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, “Edgeconnect: Structure guided image inpainting using edge prediction,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
  - [20] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. Efros, “Context encoders: Feature learning by inpainting,” in *CVPR*, 2016.
  - [21] R. T. Pushpalwar and S. H. Bhandari, “Image inpainting approaches - a review,” in *2016 IEEE 6th International Conference on Advanced Computing (IACC)*, Feb 2016, pp. 340–345.
  - [22] Y. Qiao and C. Jung, “Dictionary based hole filling with assistance of depth,” in *2014 IEEE International Conference on Multimedia and Expo (ICME)*, July 2014, pp. 1–6.
  - [23] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, “Structureflow: Image inpainting via structure-aware appearance flow,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 181–190.
  - [24] A. Romanoni, D. Fiorenti, and M. Matteucci, “Mesh-based 3d textured urban mapping,” *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3460–3466, 2017.
  - [25] R. Shetty, M. Fritz, and B. Schiele, “Adversarial scene editing: Automatic object removal from weak supervision,” in *Advances in Neural Information Processing Systems 31*. Montréal, Canada: Curran Associates, 2018, pp. 7716–7726.
  - [26] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in *International Conference on Learning Representations*, 2015.
  - [27] L. Song, J. Cao, L. Song, Y. Hu, and R. He, “Geometry-aware face completion and editing,” in *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*. AAAI Press, 2019, pp. 2506–2513. [Online]. Available: <https://doi.org/10.1609/aaai.v33i01.33012506>
  - [28] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C. C. J. Kuo, “Spg-net: Segmentation prediction and guidance network for image inpainting,” 2018, cite arxiv:1805.03356Comment: BMVC 2018 camera ready. [Online]. Available: <http://arxiv.org/abs/1805.03356>
  - [29] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, “Image inpainting via generative multi-column convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2018, pp. 331–340.
  - [30] W. Xiong, J. Yu, Z. L. Lin, J. Yang, X. Lu, C. Barnes, and J. Luo, “Foreground-aware image inpainting,” in *CVPR*, 2019.
  - [31] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, “Shift-net: Image inpainting via deep feature rearrangement,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 1–17.
  - [32] R. A. Yeh\*, C. Chen\*, T. Y. Lim, S. A. G., M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, \* equal contribution.
  - [33] H. Ying, L. Kai, and Y. Ming, “An improved image inpainting algorithm based on image segmentation,” *Procedia Comput. Sci.*, vol. 107, no. C, pp. 796–801, Apr. 2017. [Online]. Available: <https://doi.org/10.1016/j.procs.2017.03.175>
  - [34] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, “Free-form image inpainting with gated convolution,” *arXiv preprint arXiv:1806.03589*, 2018.
  - [35] —, “Generative image inpainting with contextual attention,” *arXiv preprint arXiv:1801.07892*, 2018.
  - [36] N. Zhang, H. Ji, L. Liu, and G. Wang, “Exemplar-based image inpainting using angle-aware patch matching,” *EURASIP Journal on Image and Video Processing*, vol. 2019, pp. 1–13, 2019.
  - [37] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, “The unreasonable effectiveness of deep features as a perceptual metric,” in *CVPR*, 2018.
  - [38] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, April 2004.