Realistic Composite Image Creation Using Generative Adversarial Networks

Shivangi Aneja
Technical University of Munich
Germany
shivangi.aneja@tum.de

Soham Mazumder
Technical University of Munich
Germany
soham.mazumder@tum.de

Abstract

Image composting is a challenging problem which requires professional editing skills and considerable amount of time. Our project aims to cater to this problem by making composite images look realistic. To achieve this, we are using GANS [3].

1. Introduction

The goal of our network is to improve the quality of composite images. The previous approaches like Tsai et al. [9] have used autoencoder based architectures for image composting. Inspired by their work, our approach also uses autoencoder based architecture for our generator network. To best of our knowledge, this is the first attempt of using GANs for image composting. We used conditional GANs [6] to achieve this task. The dataset used, network architecture, loss functions and results are described in subsequent sections.

2. Dataset

One of the biggest challenge for image composting is unavailability of any benchmark dataset for evaluation. So we ourselves created the dataset for this task. We used images from iCoSeg Dataset [2]. The original images are ground truth images. Creating composite images is a two step process. First we select a set of "good" style images provided by [5] and apply these styles onto the images of iCoSeg dataset. We call these images as stylized images. Secondly, using the masks of these stylized images we patch the stylized foreground objects from the image to the ground truth images. These images are called composite images. This process is explained in Figure 1. The data splitting is described in table 1.

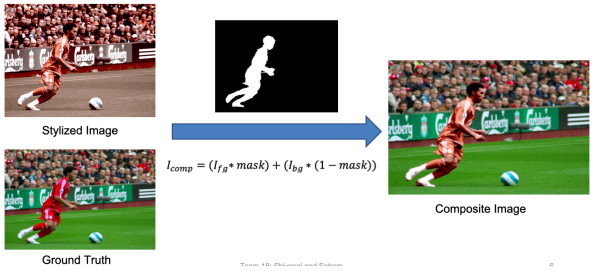


Figure 1: Composite Image Creation Pipeline

Mode	No. of Images
Training	3750
Validation	370

Table 1: Data Splitting

3. Network Architecture

3.1. Generator

The generator network follows a symmetric autoencoder architecture with skip links between corresponding pair of layers. The architecture is described in figure 2. For Convolutions, the filters applied are of size 3×3, followed by Leaky ReLU [1] activations and Batch Normalization [8].

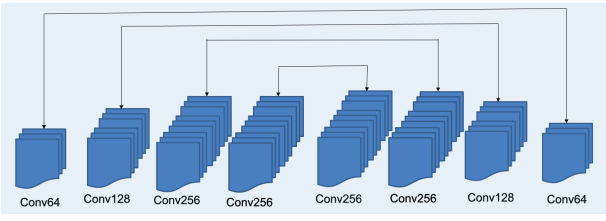


Figure 2: Generator Architecture

3.2. Discriminator

The discriminator architecture is inspired from Patch GAN used in Pix2Pix [7]. In our network, we only used four discriminator convolution layers. Each convolution is followed by Batch Normalization [8] and then Leaky ReLu [1]. The architecture is diagrammatically explained in figure 3.

Figure 3: Discriminator Architecture

4. Loss Functions

We tried out a variety of loss functions to train our generator network. These are explained in this section.

4.1. Reconstruction Loss

For reconstruction we can either use L1-loss or L2-loss on the RGB images.

4.2. Perceptual Loss

This loss function is same as described in [4].

4.3. HSV Loss

This loss is based on Hue, Saturation and Intensity values of images. To compute this loss, we first convert RGB images to HSV colour space and then compute channel wise L2 loss for each of the three channels. For this project, we realized that only Hue and Saturation values gives best results.

$$L_{hue} = L_2(I_{generated}[hue] - I_{real}[hue]) \quad (1)$$

$$L_{sat} = L_2(I_{generated}[sat] - I_{real}[sat]) \quad (2)$$

$$L_{hsv} = L_{hue} + L_{sat} \quad (3)$$

The Generator loss is the combination of above losses and defined in equation 4 , where λ_i are hyperparameters of the network.

$$L_{generator} = L_{adversarial} + \lambda_1 L_{recon} + \lambda_2 L_{perceptual} + \lambda_3 L_{hsv} \quad (4)$$

5. Metrics

To evaluate our results, we used following four metrics.

Mean Squared Error : This is absolute difference between generated image and ground truth image.

Peak Signal To Noise Ratio (PSNR) : This is ratio of maximum possible signal power to power of corrupting noise.

Structural Similarity Index (SSIM) : This metric predicts the perceived quality of image using luminance masking and contrast masking.

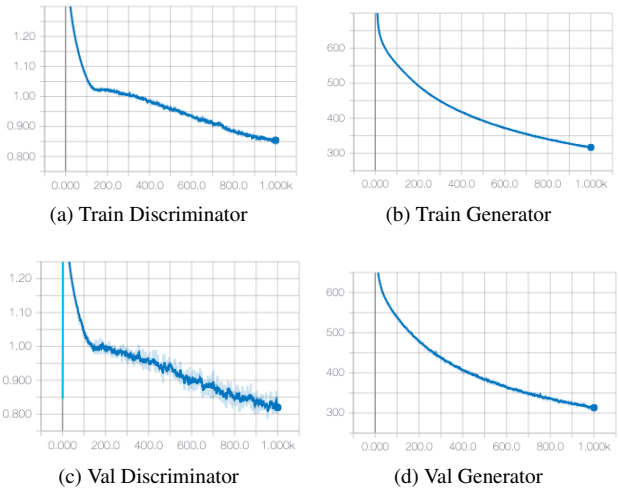
Visual Information Fidelity (VIF) : This metric interprets the image quality as "fidelity" with the reference image.

Parameter	Value
Discriminator Learning Rate	$1e-7$
Generator Learning Rate	$1e-5$
L_{recon}	L1
λ_1	100
λ_2	20
λ_3	50

Table 2: Training Hyperparameters

6. Implementation

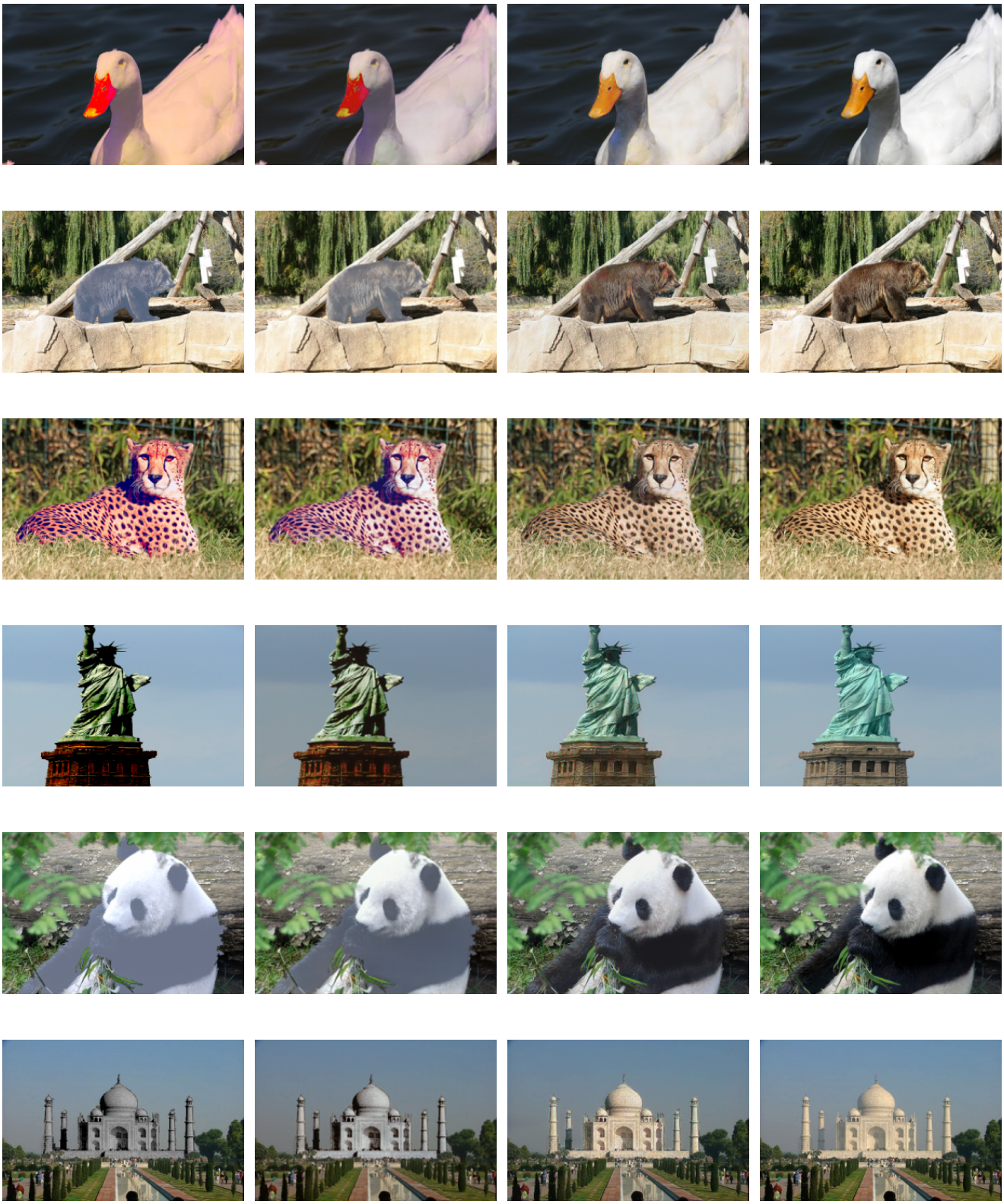
The hyperparameters that gave best results is listed in table 2. Adam optimizer is used for training both the generator and discriminator networks. The training curves for the final model with Patch GAN are shown in figure 4.



(a) Train Discriminator (b) Train Generator

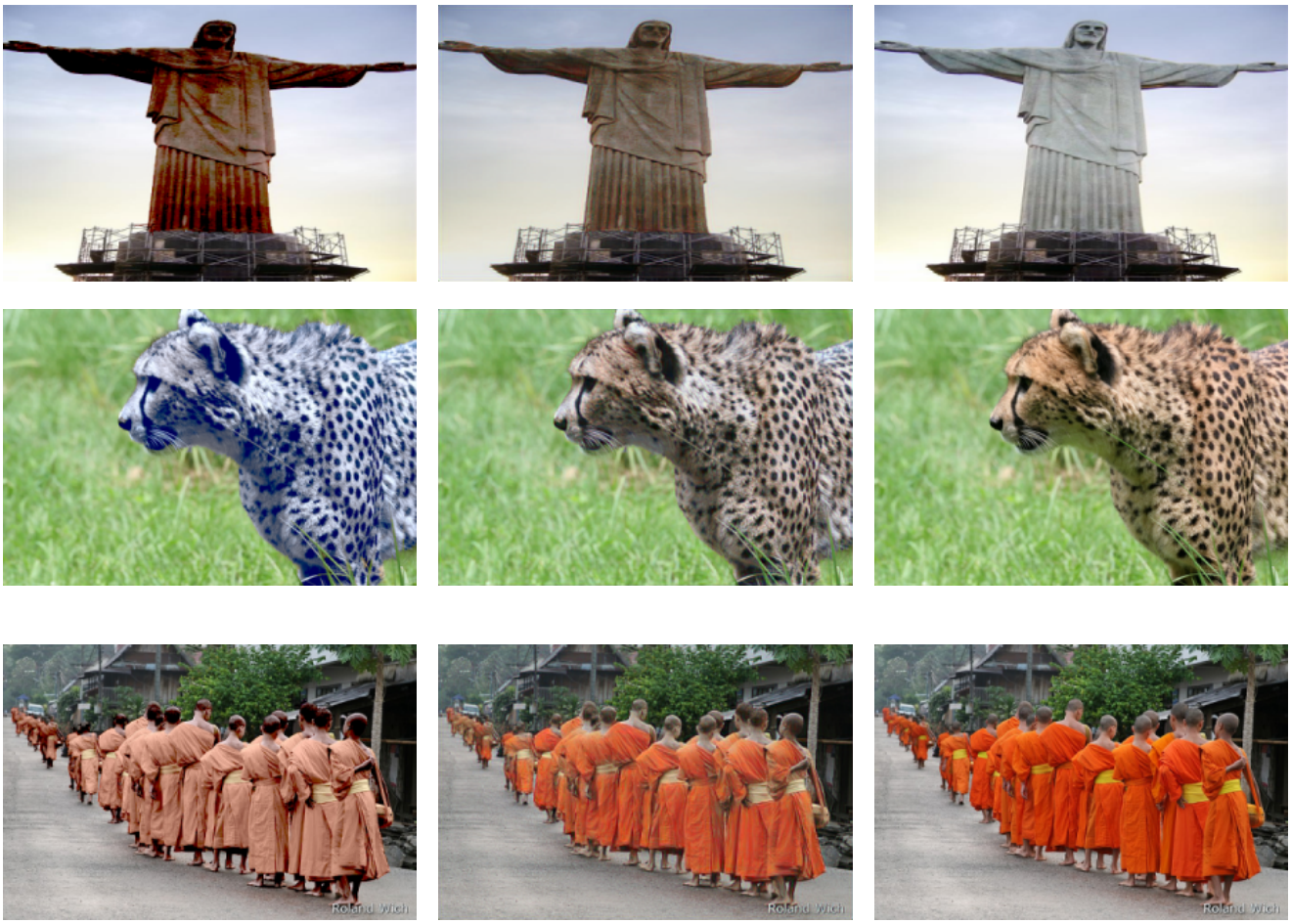
(c) Val Discriminator (d) Val Generator

Figure 4: Loss curves



Composite | Tsai et al. [9] | Ours | Ground Truth

Figure 5: Comparison of results on the synthesized dataset.



Composite Ours Ground Truth

Figure 6: Generated images which are realistic but not similar to ground truth.

Config	MSE	PSNR	SSIM	VIF
Without Patch GAN	1043.65	20.94	0.88	0.61
With Patch GAN	**699.26**	**22.11**	**0.93**	**0.66**

Table 3: Quantitative Results

Composite No Patch GAN With Patch GAN Ground Truth

Figure 7: Comparison of different discriminator networks.

7. Results

In this section, we present our results and compare them with the current state of the art [9]. The quantitative results are shown in table 3. We also see that a patch based discriminator network significantly improve the realism in generated images as shown in figure 7.
The qualitative comparison of our results with current state-of-the-art is shown in figure 5.

8. Conclusions

In this project, we present a network that generates images with a high degree of realism. Experimental results in figure 6 show that our network is able to learn a diverse range of filters and generates images that appear realistic visually even if not close to ground truth images.
An added advantage of our network compared to the current state-of-the-art [9] is that our network does not require foreground masks to generate realistic images.

References

[1] A. Y. N. Andrew L. Maas, Awni Y. Hannun. Rectifier nonlinearities improve neural network acoustic models. 2013.

[2] D. P. J. L. Dhruv Batra, Adarsh Kowdle and T. Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[3] I. Goodfellow. Generative adversarial networks. 2017.

[4] L. F.-F. Justin Johnson, Alexandre Alahi. Perceptual losses for real-time style transfer and super-resolution. 2016.

[5] J.-Y. Lee, K. Sunkavalli, Z. Lin, X. Shen, and I. S. Kweon. Automatic content-aware color and tone stylization. *arXiv preprint arXiv:1511.03748*, 2015.

[6] S. O. Mehdi Mirza. Conditional generative adversarial nets. 2014.

[7] T. Z. A. A. E. Phillip Isola, Jun-Yan Zhu. Image-to-image translation with conditional adversarial networks. 2016.

[8] C. S. Sergey Ioffe. Batch normalization: Accelerating deep network training by reducing internal covariate shift. 2015.

[9] Y.-H. Tsai, X. Shen, Z. Lin, K. Sunkavalli, X. Lu, and M.-H. Yang. Deep image harmonization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.