Semantic Image Synthesis with Spatially-Adaptive Normalization

Taesung Park[1,2], Ming-Yu Liu[2], Ting-Chun Wang[2], Jun-Yan Zhu[2,3]
[1]UC Berkeley, [2]NVIDIA ,[3]MIT CSAIL

Alexander Koenig, Li Nguyen

Workshop in Machine Learning Applications for Computer Graphics
Blavatnik School of Computer Science, Tel Aviv University

April 1, 2020

Outline

Outline

Conditional Image Synthesis

- **Conditional Image Synthesis**: method to generate photorealistic images based on certain input (e.g. labels, text, ...)
- Semantic Image Synthesis: method to generate photorealistic images based on **semantic segmentation mask**

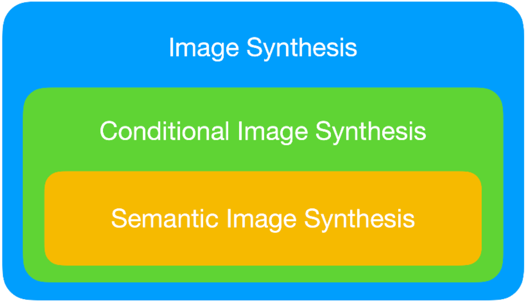


Figure: Euler diagram of image synthesis methods

What is a Semantic Segmentation Mask?

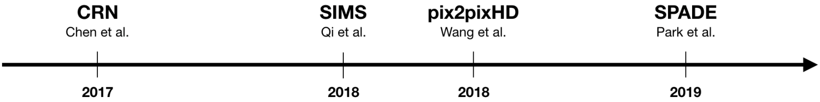
Figure: Ground truth [4]

Figure: Segmentation mask [4]

- Semantic segmentation: clustering image pixels together which belong to the same object class [6]
- Goal: turn segmentation mask into a photorealistic image
- Application of Semantic Image Synthesis: content generation and image editing

Outline

Related Work



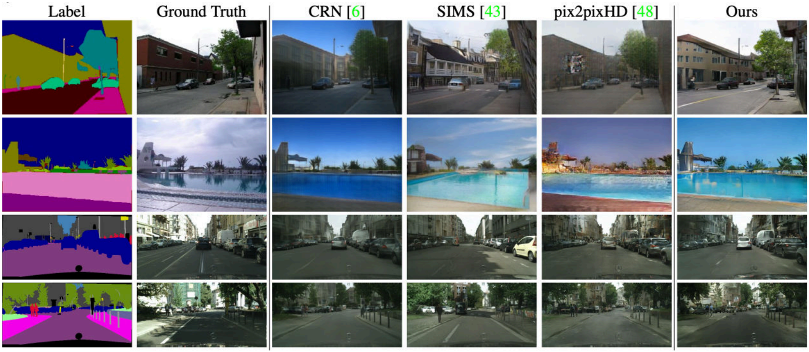


Figure: Visual Comparison of Park et al. to Related Works

Related Work: **C**ascaded **R**efinement **N**etwork (CRN)

- The architecture consists of a **cascade** of refinement modules which operate at **different resolutions** each
- Each layer is followed by convolutions, normalization and a non-linearity [1]

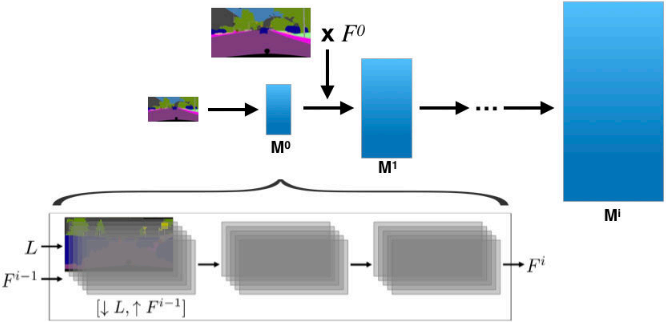


Figure: Network Architecture of CRN

Related Work: SIMS

- SIMS = **S**emi-parametric **IM**age **S**ynthesis
- Image synthesis is performed by **"stitching"** parts of images together. The parts of the images stem from a memory bank of **image segments** which is created from a training set of images beforehand [5]

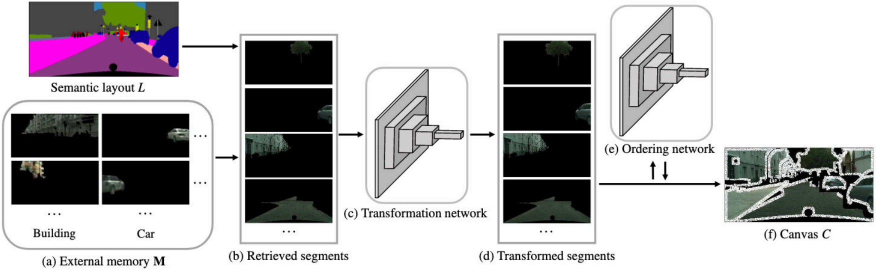


Figure: Canvas Generator for SIMS

Related Work: pix2pixHD

- Focus images with **high resolution** and **photorealism**
- **Approach:** using a coarse-to-fine generator and multi-scale discriminator architectures
- Decompose the generator into two **sub-networks** *G1* and *G2* to combine the **global** and **local** information [7]

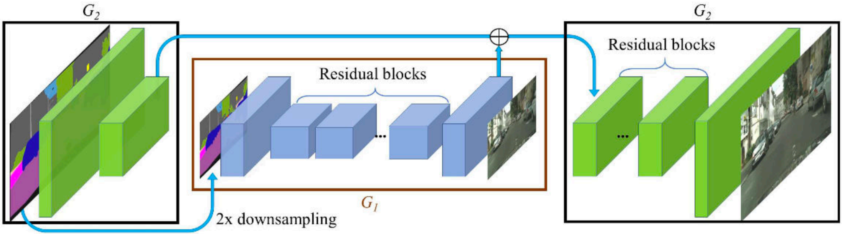


Figure: Network Architecture of pix2pixHD's Course-to-fine Generator

Related Work: pix2pixHD

Multi-scale Discriminator

- **Problem:** Discriminator needs **large receptive field** to differentiate between **high resolution** images. However, constructing a deeper network could lead to **overfitting** and a larger **memory** footprint
- **Solution:** Multi-scale discriminators: decompose into 3 identical discriminators (*D1*, *D2*, *D3*) with **different image scales**

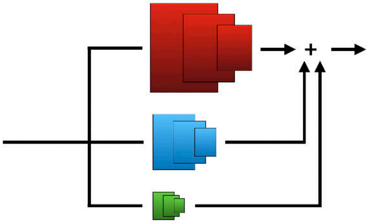


Figure: Architecture of the **pix2pixHD's** Multi-Scale Discriminator

Issues with Related Works

- **Current Approach**: Semantic information is direct input to neural network and processed through stacks of convolution, normalization, and non-linearity layers

Issues with Related Works

- **Current Approach**: Semantic information is direct input to neural network and processed through stacks of convolution, normalization, and non-linearity layers
- **Problem**: Semantic information is not well preserved (normalization layers tend to "wash away" semantic information)

Issues with Related Works

- **Current Approach**: Semantic information is direct input to neural network and processed through stacks of convolution, normalization, and non-linearity layers
- **Problem**: Semantic information is not well preserved (normalization layers tend to "wash away" semantic information)
- **Solution**: A novel conditional normalization method (SPADE) that modulates the activations using semantic layouts

Outline

Spatially-Adaptive Denormalization (SPADE) Layer

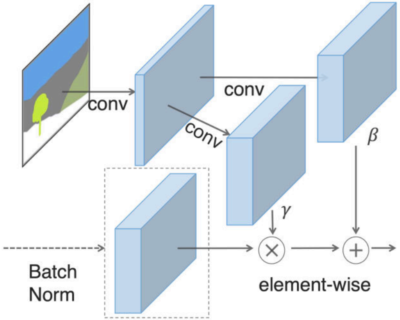


Figure: The novel SPADE Layer

1. Unconditional normalization of activations of previous layer with BatchNorm
2. Denormalization with modulation parameters (scale γ and bias β)

Spatially-Adaptive Denormalization (SPADE) Layer

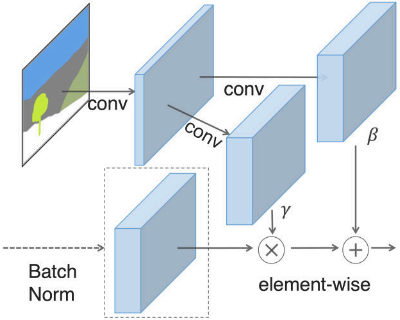


Figure: The novel SPADE Layer

1. Unconditional normalization of activations of previous layer with BatchNorm
2. Denormalization with modulation parameters (scale γ and bias β)

Novelty

- γ and β are learned and depend on location in segmentation mask!
- Modulation parameters encode semantic layout

SPADE Generator

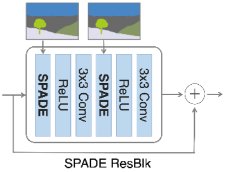


Figure: ResBlk

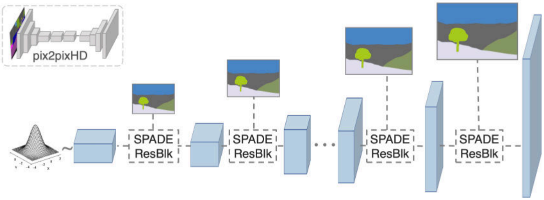


Figure: SPADE Generator

- ResBlk: residual block with skip connection

SPADE Generator

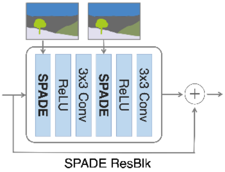


Figure: ResBlk

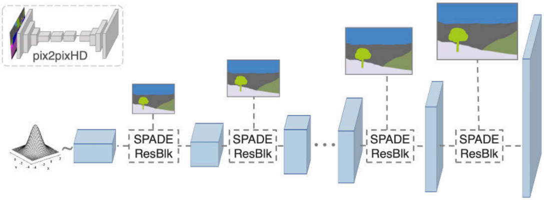


Figure: SPADE Generator

- ResBlk: residual block with skip connection
- Resized seg. masks influence generation through SPADE ResBlks

SPADE Generator

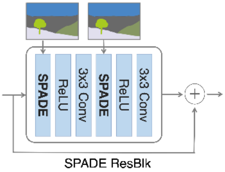


Figure: ResBlk

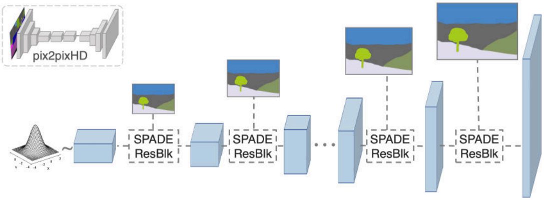


Figure: SPADE Generator

- ResBlk: residual block with skip connection
- Resized seg. masks influence generation through SPADE ResBlks
- Nearest neighbor upsampling

SPADE Generator

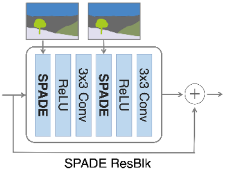


Figure: ResBlk

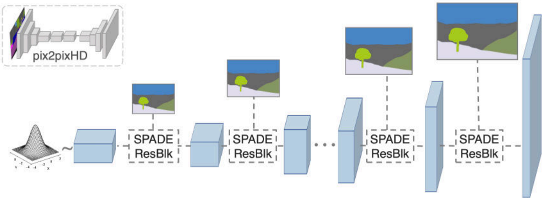


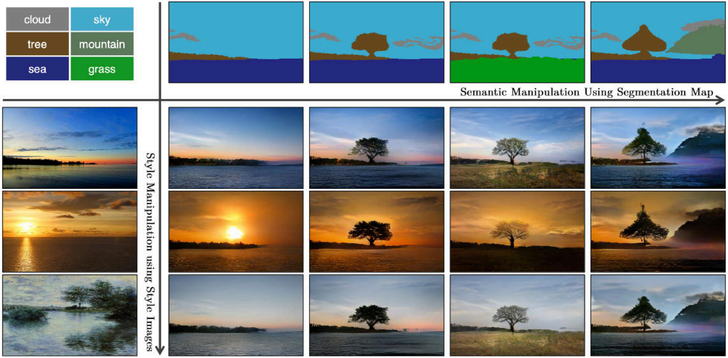
Figure: SPADE Generator

- ResBlk: residual block with skip connection
- Resized seg. masks influence generation through SPADE ResBlks
- Nearest neighbor upsampling
- Random noise fed to first layer instead of segmentation mask

Multi-Modal Synthesis

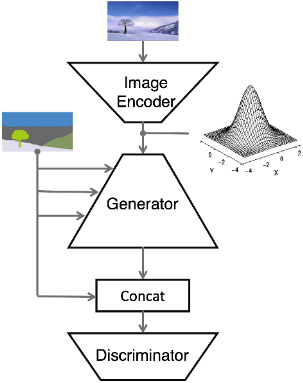
- Different random inputs with the same segmentation mask lead to different appearances but same semantic layout

Guided Image Synthesis



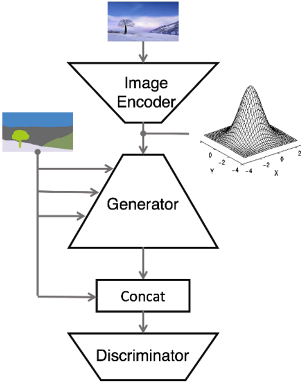
- Control semantics with segmentation mask and appearance with style image (style and semantics disentanglement)
- Interactive web application GauGAN

Network Architecture



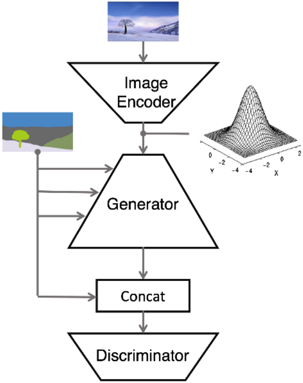
- **Image encoder** captures style of a real image in a latent representation. Outputs a mean vector μ and a variance vector σ^2

Network Architecture



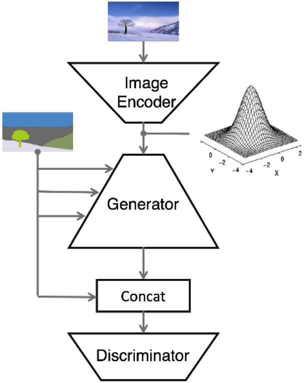
- **Image encoder** captures style of a real image in a latent representation. Outputs a mean vector μ and a variance vector σ^2
- **Generator** combines encoded style and seg. mask to reconstruct original image (Encoder + Generator = VAE)

Network Architecture



- **Image encoder** captures style of a real image in a latent representation. Outputs a mean vector μ and a variance vector σ^2
- **Generator** combines encoded style and seg. mask to reconstruct original image (Encoder + Generator = VAE)
- **Concat** concatenates segmentation mask and generated image for comparison

Network Architecture



- **Image encoder** captures style of a real image in a latent representation. Outputs a mean vector μ and a variance vector σ^2
- **Generator** combines encoded style and seg. mask to reconstruct original image (Encoder + Generator = VAE)
- **Concat** concatenates segmentation mask and generated image for comparison
- **Discriminator** same architecture and learning objective as pix2pixHD, but replace LS-GAN loss with Hinge loss.

Outline

Main Datasets

Figure: COCO-Stuff

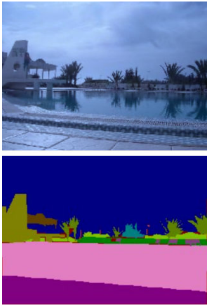


Figure: ADE20K

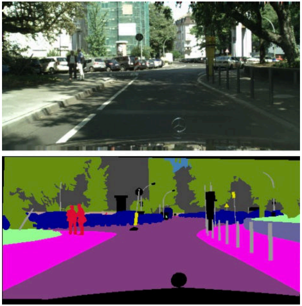


Figure: Cityscapes

Name	Train	Val	Classes	Description
COCO-Stuff	118k	5k	182	Challenging due to diversity
ADE20K	≈20k	2k	150	Similar to COCO, very diverse
Cityscapes	3k	0.5k	30	Street scene images

Comparison of Qualitative Results

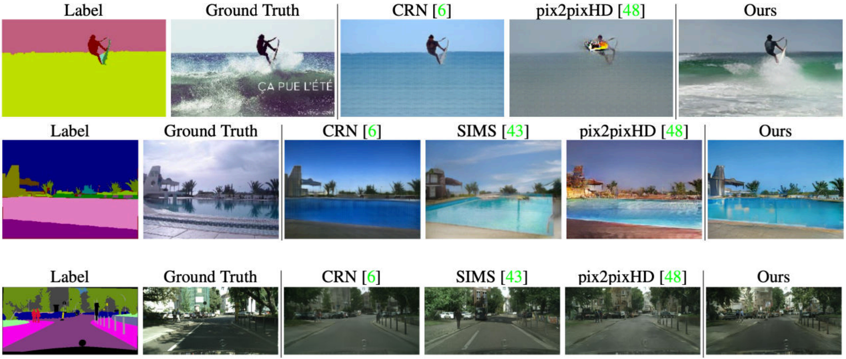


Figure: Top: COCO-Stuff, Middle: ADE20K, Bottom: Cityscapes

Comparison of Quantitative Results

Method	COCO-Stuff			ADE20K			ADE20K-outdoor			Cityscapes		
	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID
CRN [6]	23.7	40.4	70.4	22.4	68.8	73.3	16.5	68.6	99.0	52.4	77.1	104.7
SIMS [43]	N/A	N/A	N/A	N/A	N/A	N/A	13.1	74.7	67.7	47.2	75.5	**49.7**
pix2pixHD [48]	14.6	45.8	111.5	20.3	69.2	81.8	17.4	71.6	97.8	58.3	81.4	95.0
Ours	**37.4**	**67.9**	**22.6**	**38.5**	**79.9**	**33.9**	**30.8**	**82.9**	**63.3**	**62.3**	**81.9**	71.8

- Synthesized images are segmented with well-trained models and evaluated with performance metrics

1. **Mean Intersection over Union**: What is the percentage overlap between predicted and ground truth mask?
2. **Pixel accuracy**: What is the percentage of correctly classified pixels?
3. **Fréchet Inception Distance**: What is the distance between distributions of feature vectors?

Why does the SPADE work better?

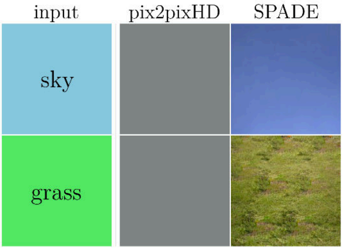


Figure: Semantic information loss after normalization layer

- Unconditional normalization layers (e.g. InstanceNorm) loose semantic info of uniform masks as **normalized activations are zero**

Why does the SPADE work better?

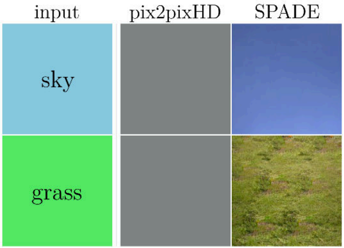


Figure: Semantic information loss after normalization layer

- Unconditional normalization layers (e.g. InstanceNorm) loose semantic info of uniform masks as **normalized activations are zero**
- SPADE better **preserves semantic information** because segmentation mask is not normalized but only modulated

Why does the SPADE work better?

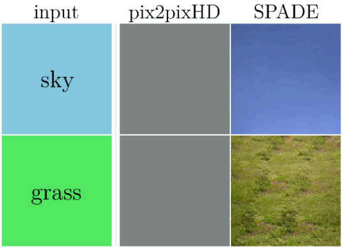


Figure: Semantic information loss after normalization layer

- Unconditional normalization layers (e.g. InstanceNorm) loose semantic info of uniform masks as **normalized activations are zero**
- SPADE better **preserves semantic information** because segmentation mask is not normalized but only modulated
- SPADE also improves performance of traditional architectures!

Outline

Conclusion

TASK	DATASET	MODEL	METRIC NAME	METRIC VALUE	GLOBAL RANK	COMPARE
Image-to-Image Translation	ADE20K Labels-to-Photos	SPADE	mIoU	38.5	# 2	See all
Image-to-Image Translation	ADE20K Labels-to-Photos	SPADE	Accuracy	79.9%	# 2	See all
Image-to-Image Translation	ADE20K Labels-to-Photos	SPADE	FID	33.9	# 2	See all
Image-to-Image Translation	ADE20K-Outdoor Labels-to-Photos	SPADE	mIoU	30.8	# 1	See all
Image-to-Image Translation	ADE20K-Outdoor Labels-to-Photos	SPADE	Accuracy	82.9%	# 1	See all
Image-to-Image Translation	ADE20K-Outdoor Labels-to-Photos	SPADE	FID	63.3	# 1	See all
Image-to-Image Translation	Cityscapes Labels-to-Photo	SPADE	Per-pixel Accuracy	81.9%	# 2	See all
Image-to-Image Translation	Cityscapes Labels-to-Photo	SPADE	mIoU	62.3	# 2	See all

Figure: SPADE ranking as of March 2020 [3]

- Introduced spatially-adaptive normalization (SPADE) layer
- SPADE network outperforms the 2019 state-of-the-art methods by a large margin and is still top-performing (#1 is [2])

Outline

References I

[1] Q. Chen and V. Koltun.
Photographic image synthesis with cascaded refinement networks.
In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017.

[2] X. Liu, G. Yin, J. Shao, X. Wang, and H. Li.
Learning to predict layout-to-image conditional convolutions for semantic image synthesis, 2019.

[3] Papers With Code.
Evaluation Results SPADE.
`https://paperswithcode.com/paper/semantic-image-synthesis-with-spatially`, 2020.

[4] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu.
Semantic image synthesis with spatially-adaptive normalization.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019.

[5] X. Qi, Q. Chen, J. Jia, and V. Koltun.
Semi-parametric image synthesis.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8808–8816, 2018.

[6] M. Thoma.
A survey of semantic segmentation, 2016.

References III

[7] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro.
High-resolution image synthesis and semantic manipulation with conditional gans.
In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.