UNIVERSITÀ DI TRENTO

Semantic Image Synthesis with Spatially-Adaptive Normalization

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, Jun-Yan Zhu

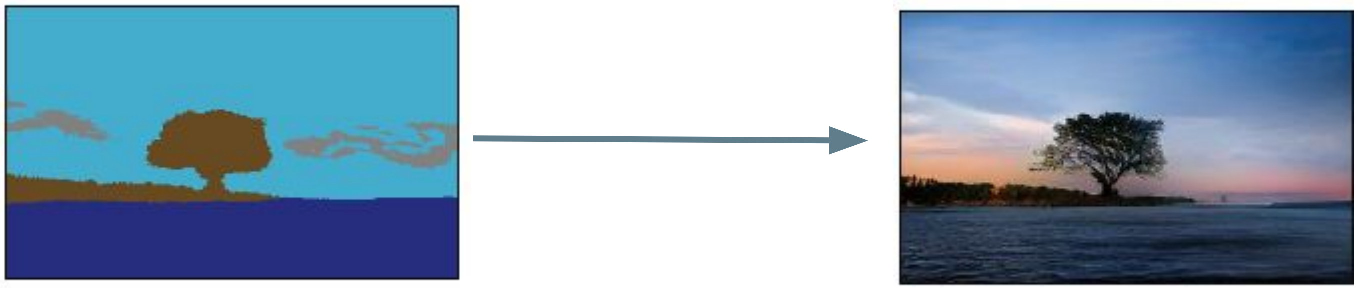
UC Berkeley, NVIDIA, MIT CSAIL

Presented by:

Matteo Destro, Zihadul Azam

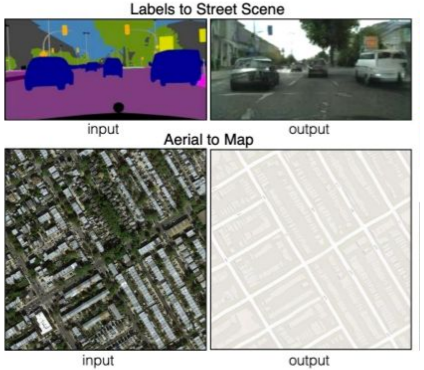
1.
Introduction

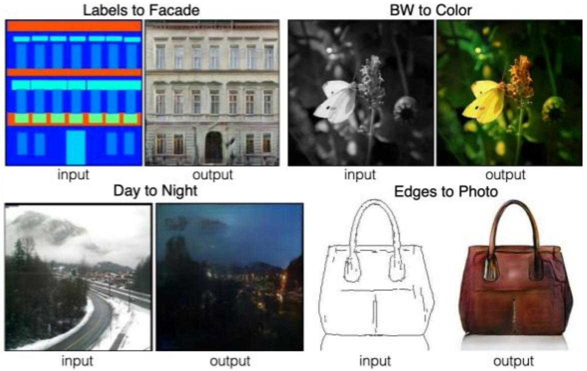
What is this paper about?

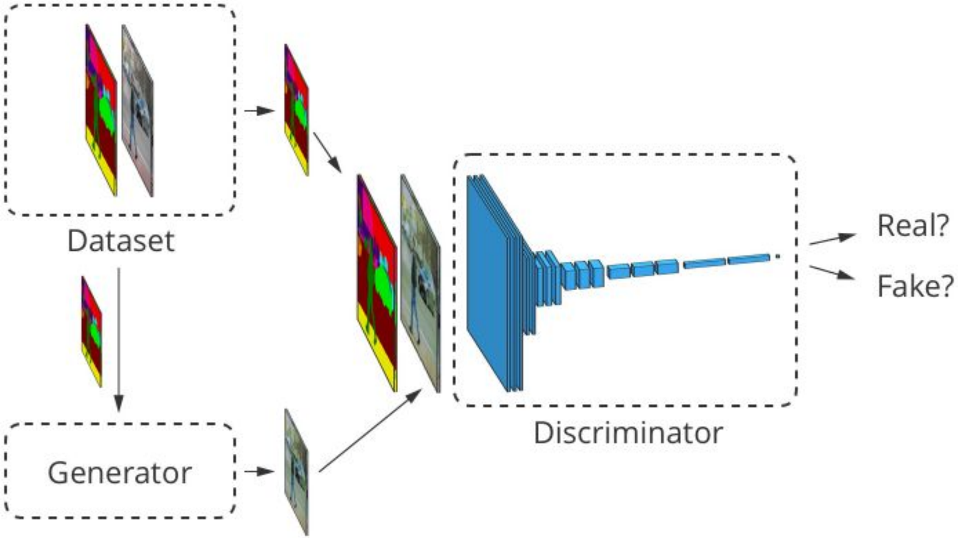


- Converting a **semantic segmentation** mask to a **photorealistic image**
- Preserve semantic information
- Produce images with multiple styles

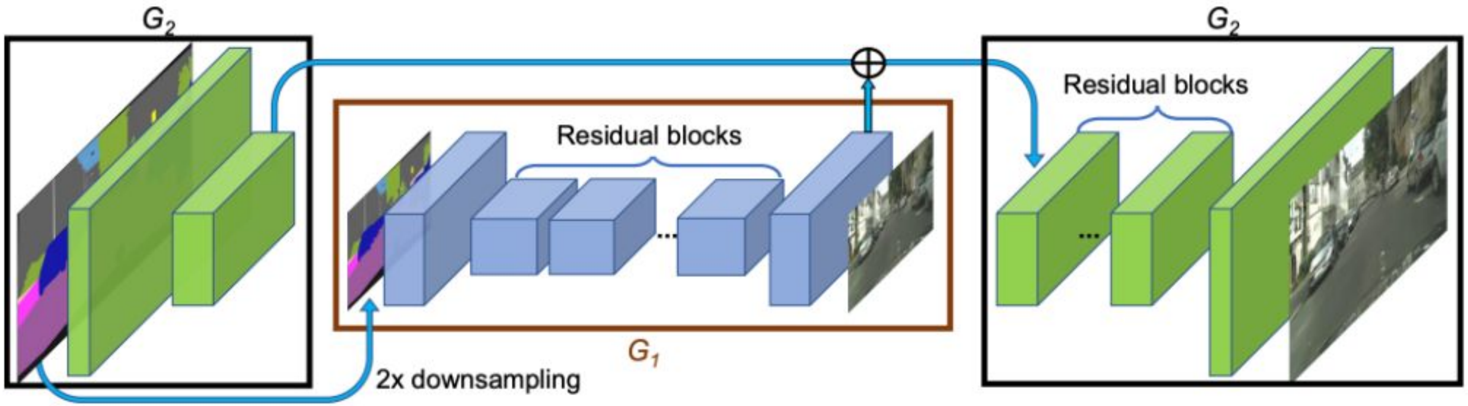
Image-to-image translation







Pix2pixHD: Generator



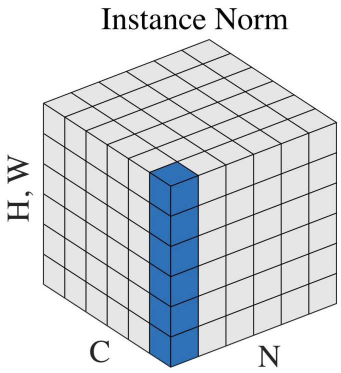
- Encoder for downscale
- Decoder for upscale

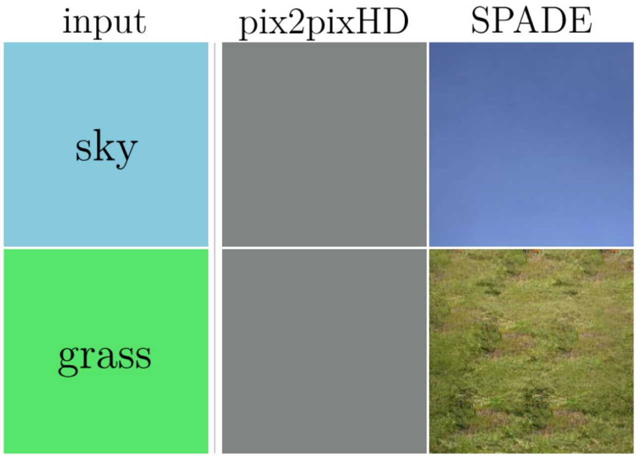
Image-to-Image Translation with Conditional Adversarial Networks | Phillip Isola, Jun-Yan Zhu | [arvix]

High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs | Ting-Chun Wang, Ming-Yu Liu | [arvix]

What is wrong with pix2pixHD ?

- Use **Instance Normalization**
- **“Wash away information”** problem: in pix2pixHD, instance normalization tends to throw away information from the segmentation map. For single-class images, it produces the same image regardless of the class.
- Instance normalization **produce 0** as output of the normalization with uniform semantic mask.





Solution:

SPADE

SPatially-Adaptive (DE)normalization

SPADE: normalization

$$\gamma^i_{c,y,x}(\mathbf{m})\frac{h^i_{n,c,y,x}-\mu^i_c}{\sigma^i_c}+\beta^i_{c,y,x}(\mathbf{m})$$

Where:

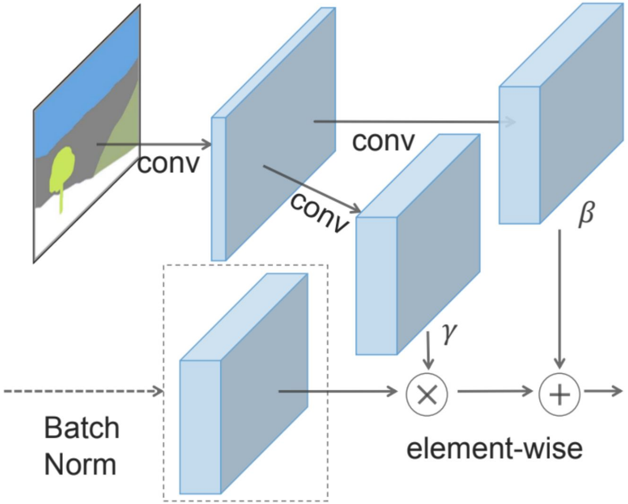
- $\mathbf{h}^{\mathbf{i}}_{\mathbf{n;c;y;x}}$ is the activation at site before normalization
- $\boldsymbol{\mu}_c$ and $\boldsymbol{\sigma}_c$ are the mean and standard deviation of the activations in channel **c**
- **m** is the segmentation mask

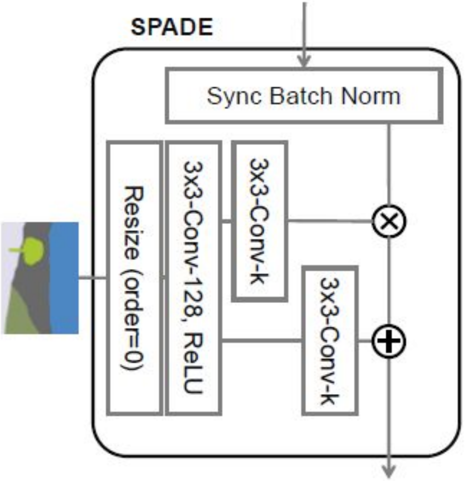
$$\mu^i_c=\frac{1}{NH^iW^i}\sum_{n,y,x}h^i_{n,c,y,x}$$

$$\sigma^i_c=\sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}\left((h^i_{n,c,y,x})^2-(\mu^i_c)^2\right)}$$

SPADE

- The mask is first projected onto an **embedding space**
- Then it's convolved to produce the modulation parameters γ and β
- The produced γ and β are multiplied and added to the normalized activation element-wise

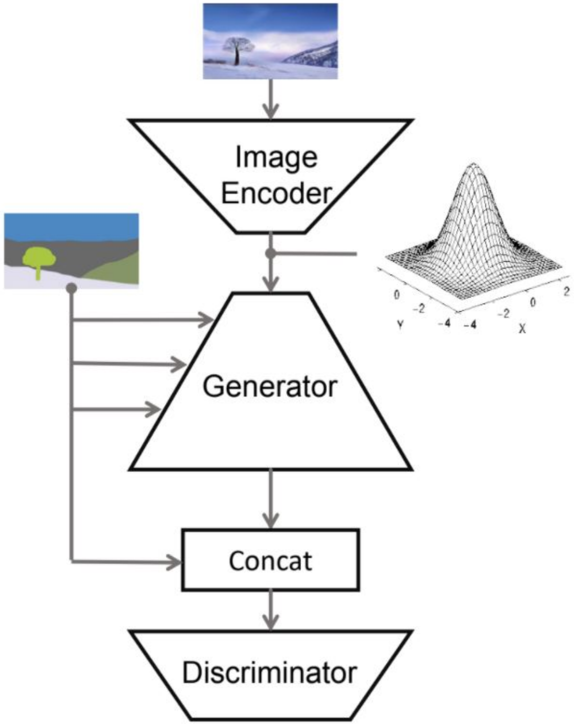




2.

Architecture

GauGAN architecture



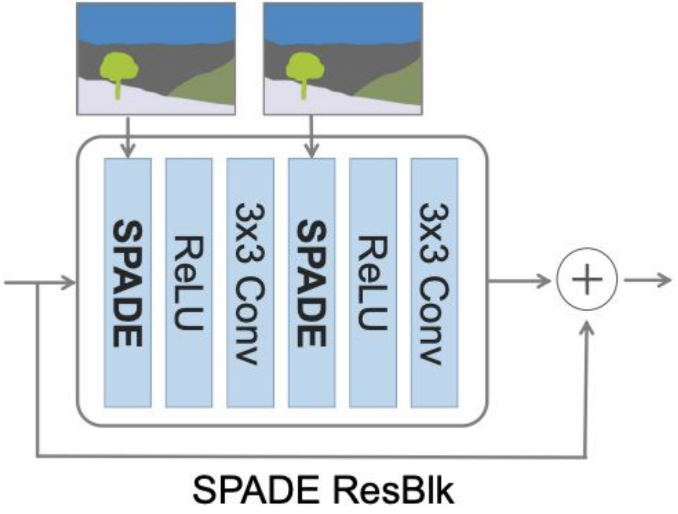
Three main components:

- **Generator:** generate the final synthetic image.
- **Discriminator:** discriminate between real and synthetic images. Used only during training.
- **Encoder:** convert an input image to its latent space representation z. Can be used to transfer the style of an image to the generated one.

Generator: SPADE Residual Block

The generator is composed by a sequence of **SPADE Residual Blocks:**

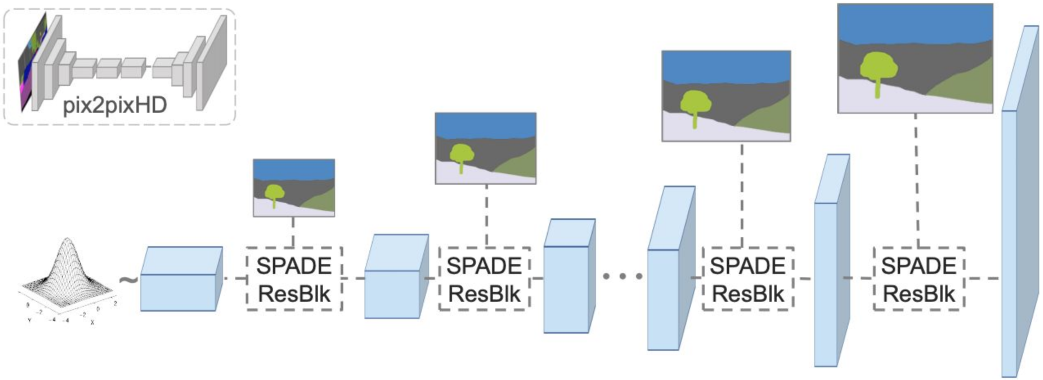
- **SPADE** normalizations:
  - the semantic mask is downsampled to match the spatial resolution
- **ReLU** activations
- **Convolutions:**
  - 3x3 kernel with padding 1
  - *k* channels in output
  - input and output spatial dimensions are equal
- **Residual connection**

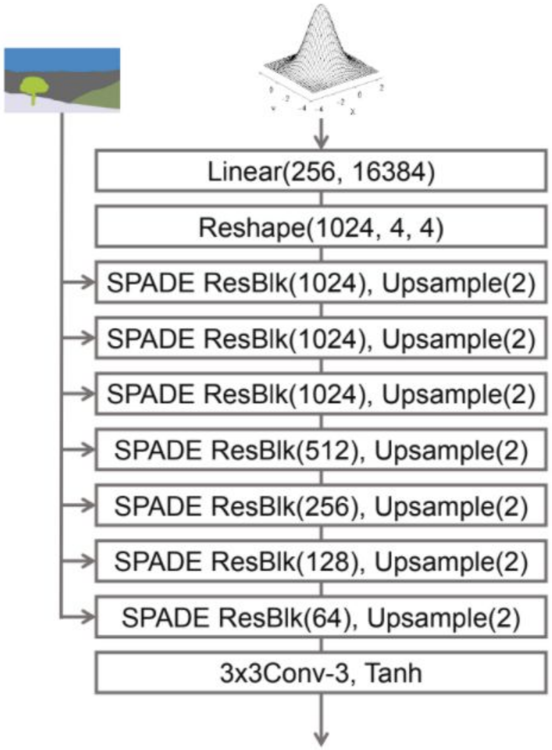


Generator

The generator is composed by a sequence of **SPADE Residual Blocks** interleaved by **upsampling layers.**

Less parameters: no need for downsampling layers like Pix2pixHD to feed the semantic map to the model (SPADE modulations parameters are enough).



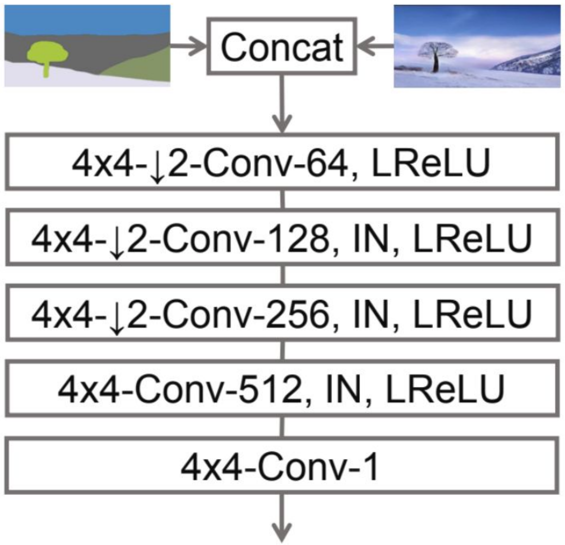


Discriminator: PatchGAN

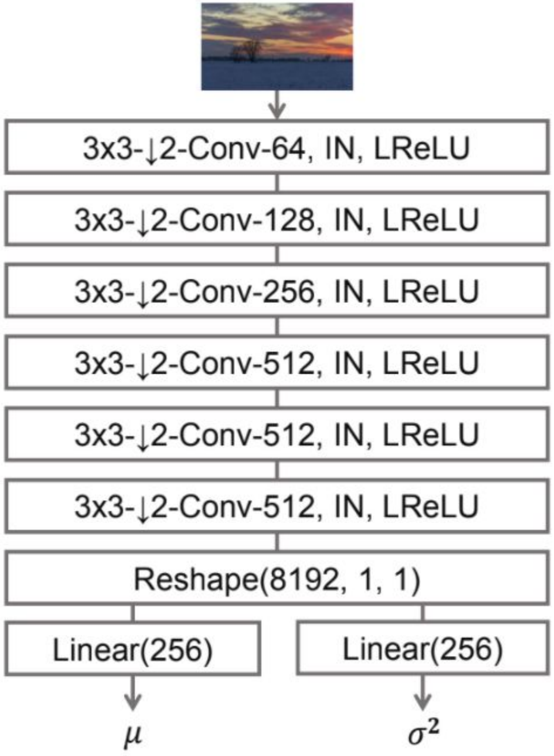
The discriminator:

- Takes in input the concatenation of the semantic map (one-hot encoded) and the image
- Is composed by a series of **convolutional layers:**
  - 4x4 kernel with stride 2
  - Leaky ReLU activations
  - Instance normalization
- A final **convolutional layer** computes the final prediction

Two discriminators are used, working at different scales.



Encoder: VAE



The encoder is composed by:

- Series of **convolutional layers:**
  - 3x3 kernel with stride 2
  - Leaky ReLU activations
  - Instance normalization
- Two **linear layers** to compute:
  - mean vector (**μ**)
  - variance vector ($\boldsymbol{\sigma}^2$)

The final latent space representation **z** is computed via a reparameterization trick:

$$z = \mu + \mathcal{N}(0,1) \cdot e^{\frac{1}{2}\sigma^2}$$

Loss functions

- **Adversarial loss:** uses the **Hinge loss** instead of the Least Squares loss of Pix2pixHD:

$$L_D = -\mathbb{E}\left[\min(0, -1 + D(x, y))\right] - \mathbb{E}\left[\min(0, -1 - D(G(z, \mathbf{m}), y))\right]$$

$$L_G = -\mathbb{E}\left[D(G(z, \mathbf{m}), y)\right]$$

- **Feature matching loss:**

$$L_{FM} = \mathbb{E}\left[\sum_{i=1}^{L} \left\| D_k^{(i)}(x) - D_k^{(i)}(G(z, \mathbf{m})) \right\|_1\right]$$

- **Perceptual loss:** similar concept to Feature loss, but uses activations of a VGG-19 pretrained model

$$L_{VGG} = \mathbb{E}\left[\sum_{i=1}^{L} \left\| VGG^{(i)}(x) - VGG^{(i)}(G(z, \mathbf{m})) \right\|_1\right]$$

- **Encoder loss** (if used):

$$L_{KLD} = \mathcal{D}_{KL}(q(z|x) \,||\, p(z))$$

3.
Results

Datasets

- **COCO-Stuff:**
  - 118’000 training and 5’000 validation images
  - 182 semantic classes
- **ADE20K (-outdoor):**
  - 20’120 training and 2’000 validation images
  - 150 semantic classes
- **Cityscapes:**
  - 3’000 training and 500 validation images
  - 19 semantic classes
- **Flickr Landscapes:**
  - 41’000 training and 1’000 validation images
  - Semantic maps computed using a pre-trained DeepLabV2 model

Evaluation metrics

Compare:

- **Human evaluation**
- **Fréchet Inception Distance** (FID)
- **Mean Intersection Over Union** (mIoU)
- **Pixel accuracy** (accu)

Idea: run a **semantic segmentation** model on the generated images and compare the resulting segmentation map with the original input.

Use state-of-the-art models for each dataset: **DeepLabV2** for COCO-Stuff, **UperNet101** for ADE20K, **DRN-D-105** for Cityscapes

Experiment results

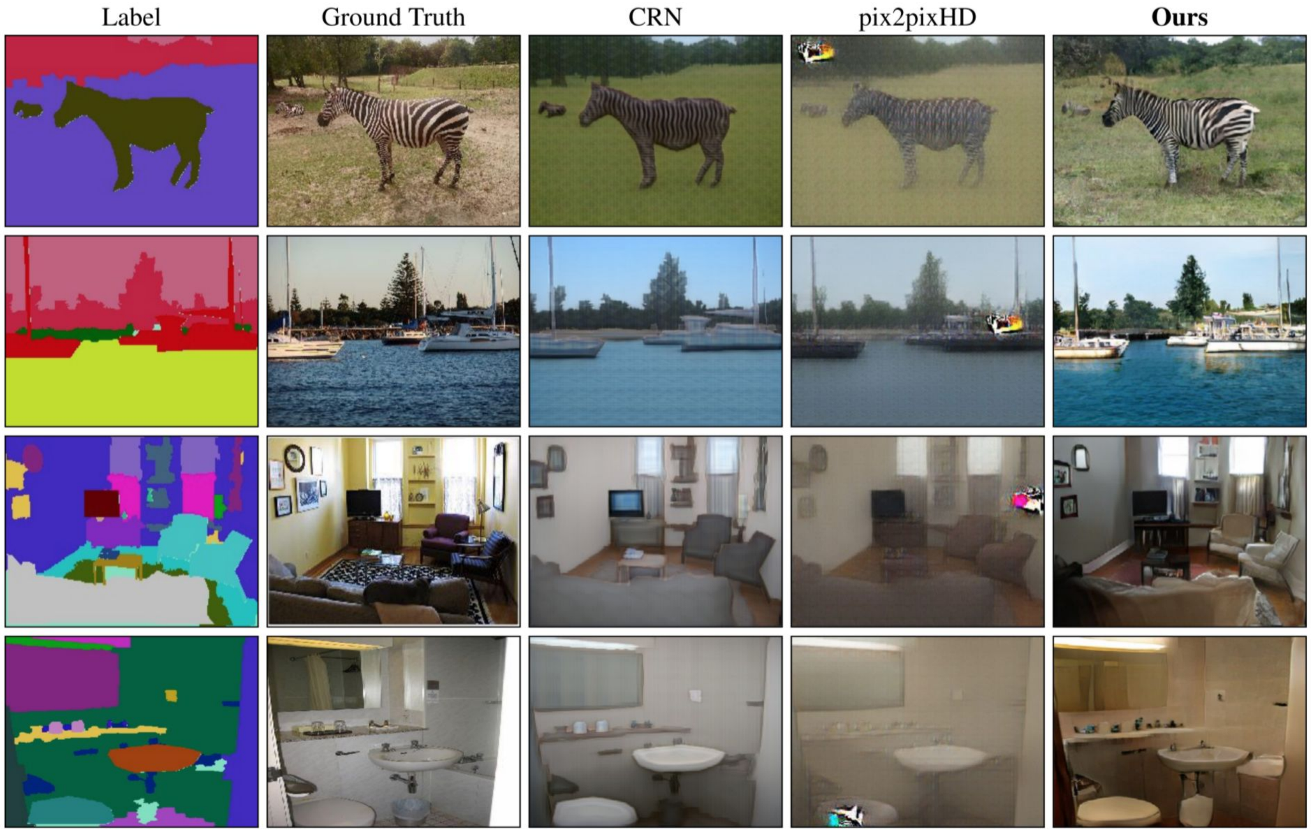
Results compared with state-of-the-art semantic image synthesis models:

- **Cascaded Refinement Network**[1] (CRN)
  Uses a deep network that repeatedly refines the output from low to high resolution
- **Semi-parametric Image Synthesis Method**[2] (SIMS)
  Semi-parametric approach that composites real segments from a training set and refines the boundaries.
- **Pix2pixHD**
  State-of-the-art GAN-based conditional image synthesis framework

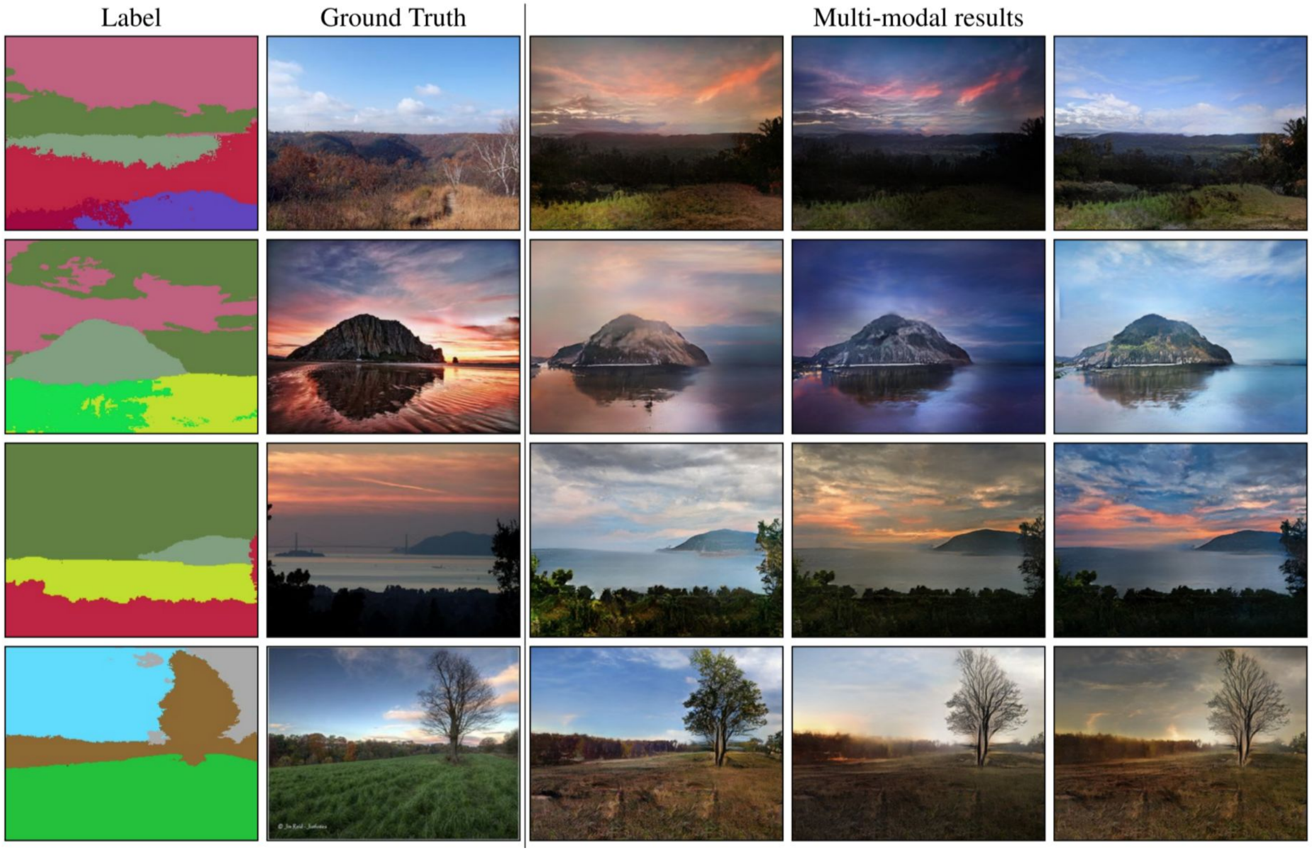
Method	COCO-Stuff			ADE20K			ADE20K-outdoor			Cityscapes		
	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID
CRN [6]	23.7	40.4	70.4	22.4	68.8	73.3	16.5	68.6	99.0	52.4	77.1	104.7
SIMS [43]	N/A	N/A	N/A	N/A	N/A	N/A	13.1	74.7	67.7	47.2	75.5	**49.7**
pix2pixHD [48]	14.6	45.8	111.5	20.3	69.2	81.8	17.4	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

[1] Q. Chen and V. Koltun, “Photographic image synthesis with cascaded refinement networks”, ICCV 2017
[2] X. Qi, Q. Chen, J. Jia, and V. Koltun, “Semi-parametric image synthesis”, CVPR 2018

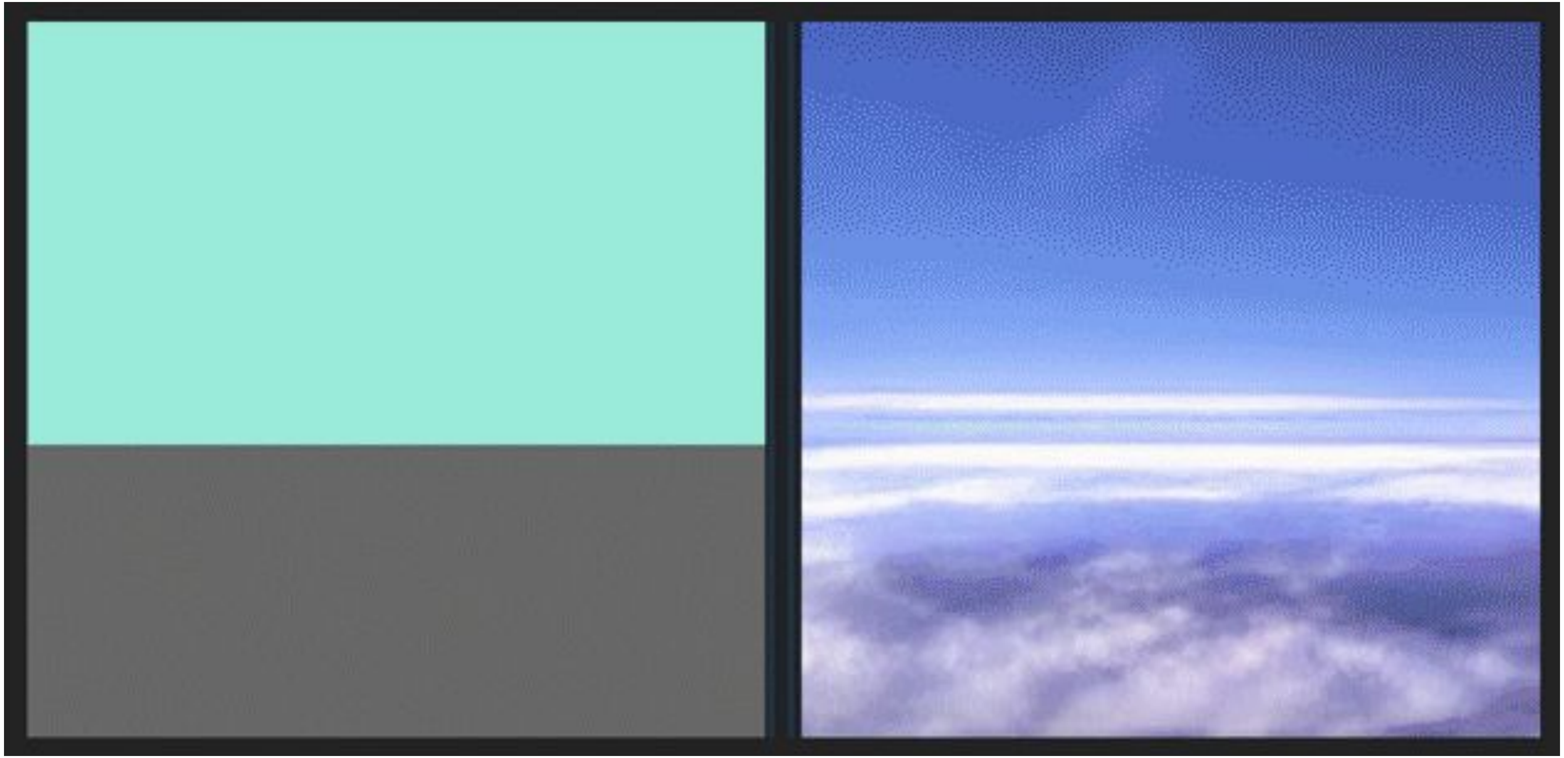
Experiment results: Image Synthesis



Experiment results: Multimodal Synthesis



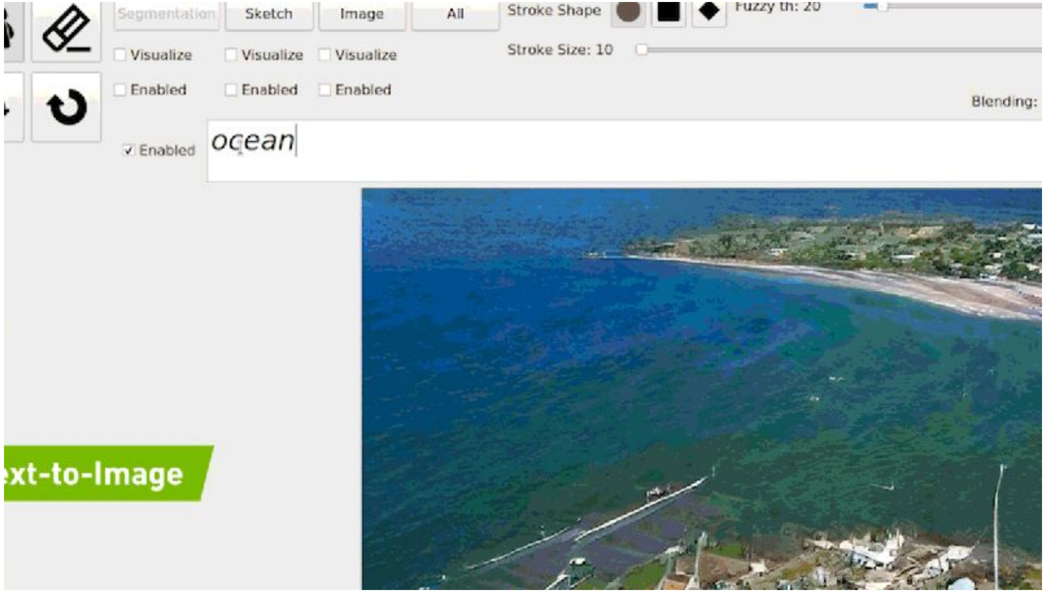
Demo: Flickr Landscapes dataset



http://gaugan.org/gaugan2/

Improvements from other works

- **CLADE**: Class-Adaptive (DE)normalization
- **SEAN**: Image Synthesis with Semantic Region-Adaptive Normalization
- **GauGAN2**



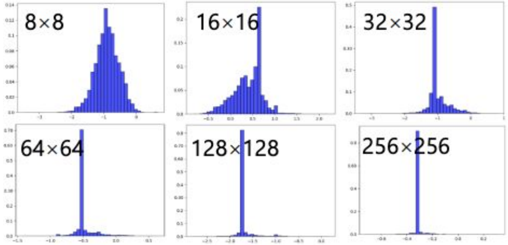
Efficient Semantic Image Synthesis via Class-Adaptive Normalization | Z. Tan, D. Chen, Q. Chu| 2021 | [arvix]
SEAN: Image Synthesis with Semantic Region-Adaptive Normalization | P. Zhu, R. Abdal, Y. Qin | 2020 | [arvix]

4.

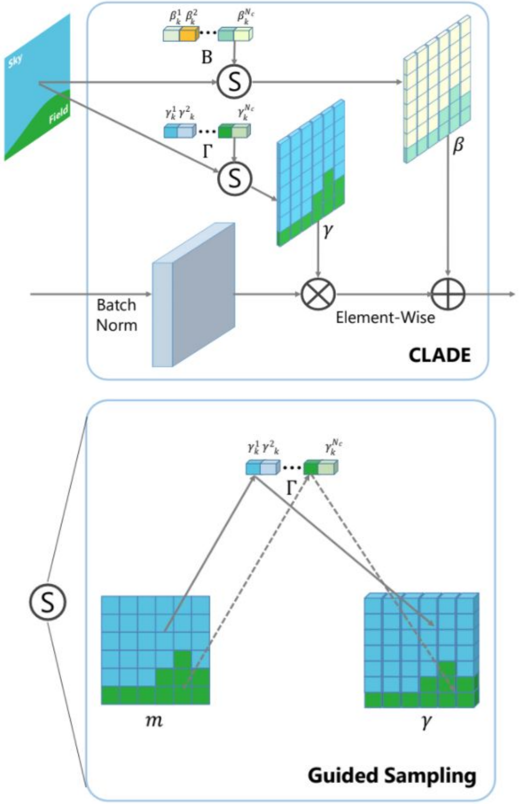
Improvements From Other Works

CLADE: Class-Adaptive (DE)normalization

They showed that the spatial information is not very beneficial in SPADE, its effectiveness is given by its semantic awareness.



γ of "building" class



Idea:

- Learn a single γ and β for each semantic class, instead of modulating them through two a shallow CNN
- Create spatial-aware modulation parameters through **guided sampling**

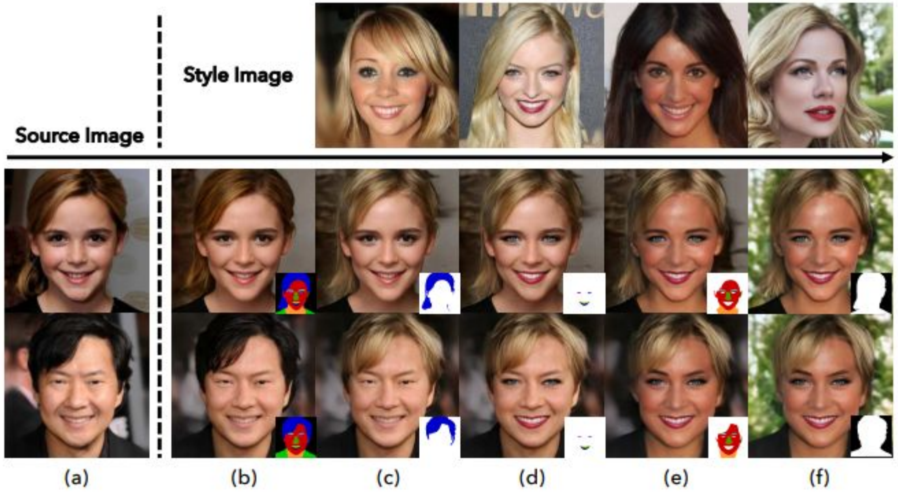
The modulation convolutional network of SPADE is removed

Z. Tan, D. Chen, Q. Chu, M. Chai, J. Liao, M. He and G. Hua, "Efficient Semantic Image Synthesis via Class-Adaptive Normalization", TPAMI 2021

SEAN: Image Synthesis with Semantic Region-Adaptive Normalization

Face image editing controlled via style images and segmentation masks:

- This network architecture can control the **style** of each semantic region individually
- Here we can specify one style reference image per region

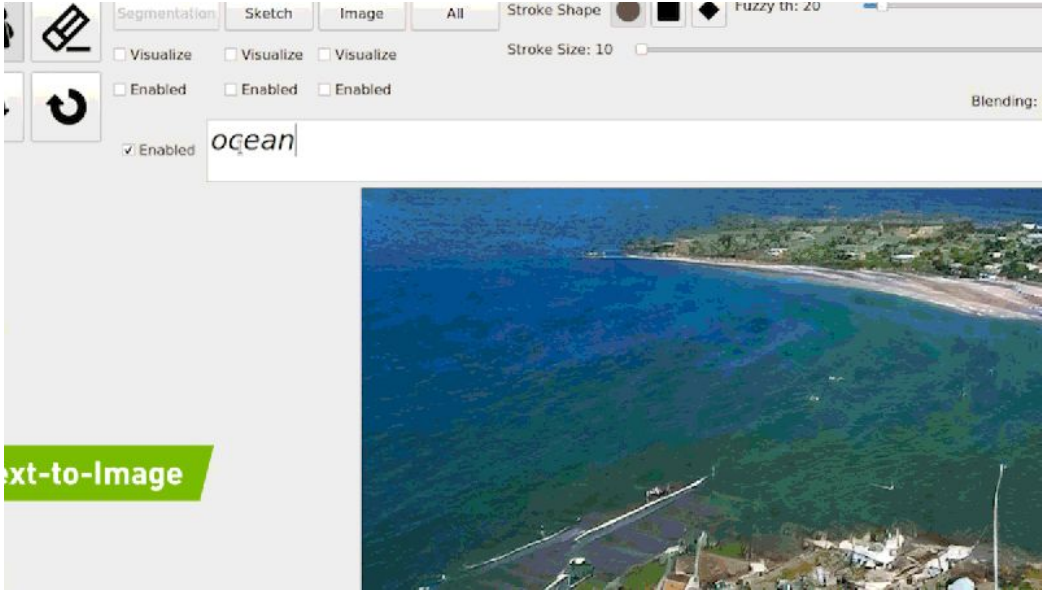


Z. Peihao, A. Rameen, Q. Yipeng and W. Peter, “SEAN: Image Synthesis With Semantic Region-Adaptive Normalization”, CVPR 2020

GauGAN2

GauGAN2 is designed to create photorealistic art with a **mix of words** and **drawings**:

- Trained on 10 million images
- Can translate natural language descriptions into landscape images



5.

Unpaired SPADE

Unpaired Semantic Image Synthesis

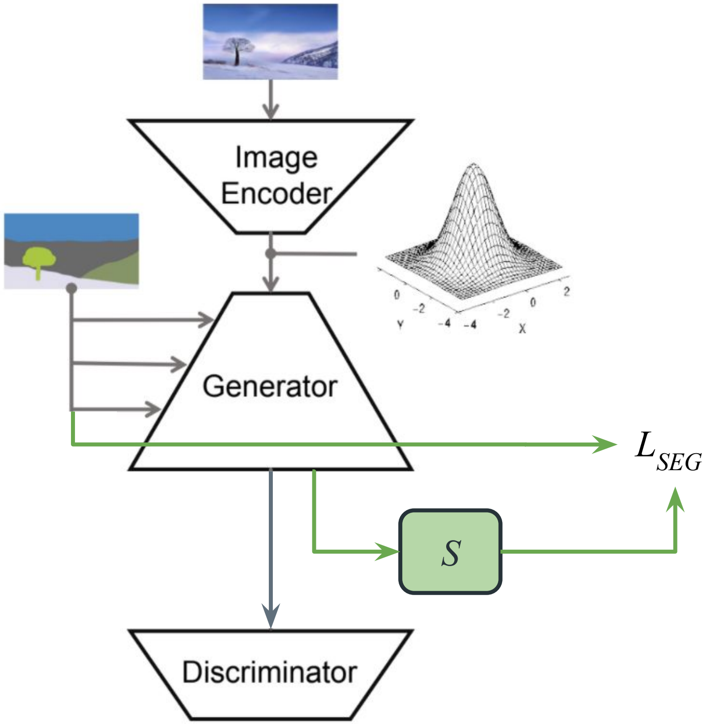
What if we don’t have paired samples?

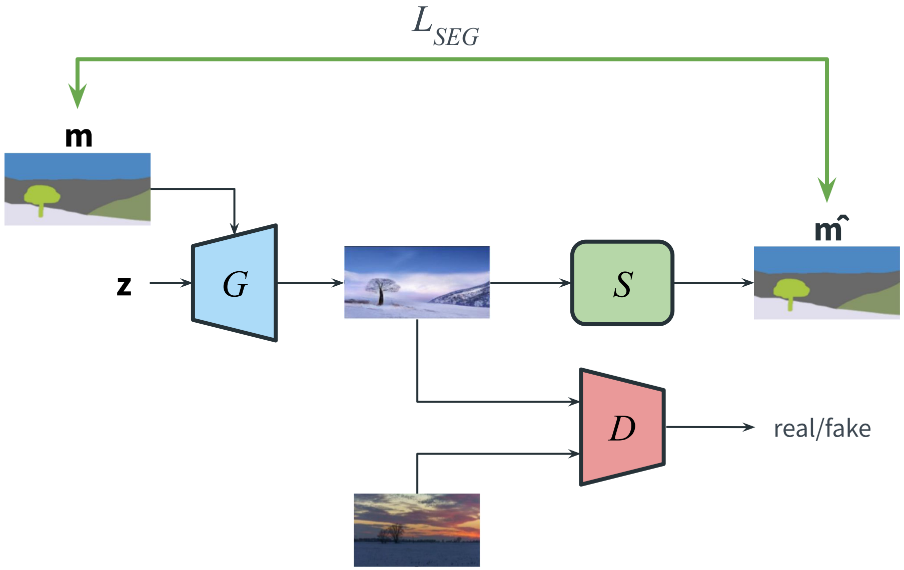
Idea: exploit a **semantic segmentation model** to train the model in an unpaired setting, introducing a reconstruction loss w.r.t the semantic map input.

We can use an approach similar to CycleGAN (unpaired image-to-image translation):

- Add a **segmentation model** in the loop (e.g. U-Net, DeepLabV3, ...)
- Add a **segmentation loss** based on the reconstructed segmentation

Unpaired Semantic Image Synthesis





Unpaired Semantic Image Synthesis

Introduce a class-balanced semantic segmentation loss in the generator:

- **Class-balanced** loss based on **effective number of samples**[1]: $\alpha_c = \dfrac{1-\beta}{1-\beta^{n_c}}$

- **Focal loss**[2]: $L_{SEG} = -\sum_{c,i,j}^{C\times H\times W} \alpha_c \cdot \mathbf{m}_{c,i,j} \cdot (1-\hat{\mathbf{m}}_{c,i,j})^{\gamma} \log(\hat{\mathbf{m}}_{c,i,j})$

Final generator loss: $L_G = -\mathbb{E}\left[D(G(z,\mathbf{m}))\right] + \lambda\mathbb{E}\left[L_{SEG}\right]$

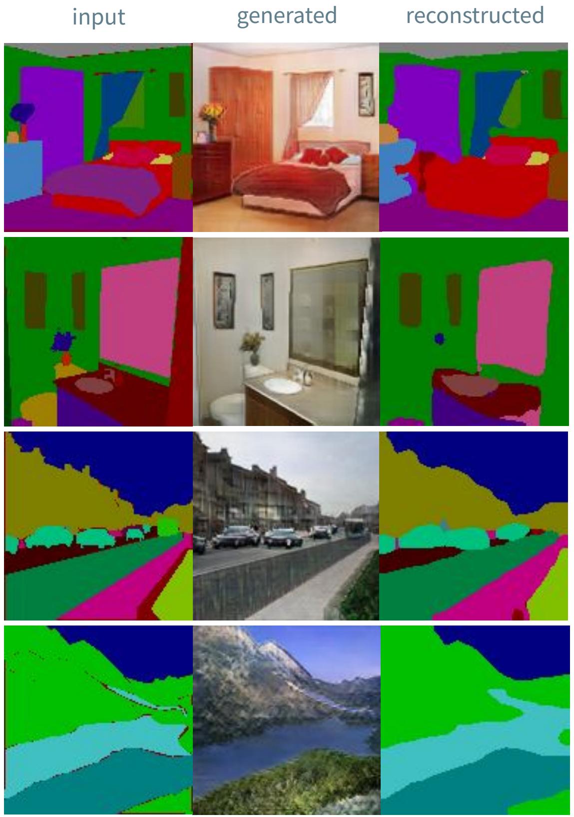
[1] Y. Cui, M. Jia, T. Lin, Y. Song and S. Belongie, “Class-Balanced Loss Based on Effective Number of Samples”, CVPR 2019
[2] T. Lin, P. Goyal, R.Girshick, K. He, and P. Dollar, “Focal loss for dense object detection”, PAMI 2018

Unpaired Semantic Image Synthesis: Results

Performances obtained over the ADE20K dataset. To reduce training time, we used:

- **Smaller model**: 14.1M parameters instead of 96.5M
- **Smaller input images**: 64x64 instead of 256x256
- **Less epochs**: 50 instead of 200
- UPerNet18[1] as **segmenter** network

Method	mIoU	Accuracy	FID
SPADE (baseline)	27.03	58.35	87.13
SPADE + pretrained UPerNet	29.74	59.96	80.10
SPADE + untrained UPerNet	29.54	59.73	83.67



[1] T. Xiao, Y. Liu, B. Zhou, Y. Jiang and J. Sun, "Unified Perceptual Parsing for Scene Understanding", ECCV 2018

Unpaired Semantic Image Synthesis: Results

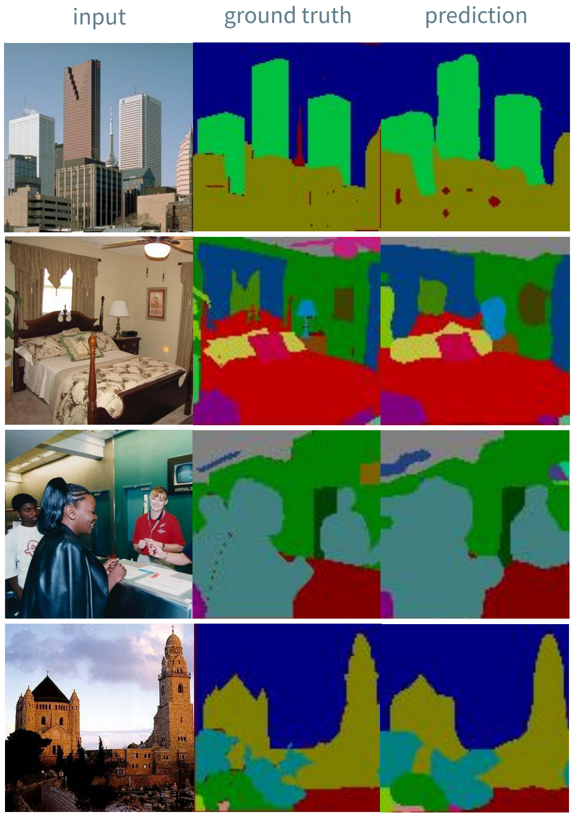
input | generated | ground truth | reconstructed

input | generated | ground truth | reconstructed

Unpaired Semantic Image Synthesis: Results

As side-effect, we train a **semantic segmentation** network in an **unsupervised** way.

Method	mIoU	Accuracy
Pretrained UPerNet	38.18	78.64
UPerNet trained with SPADE	26.76	57.27

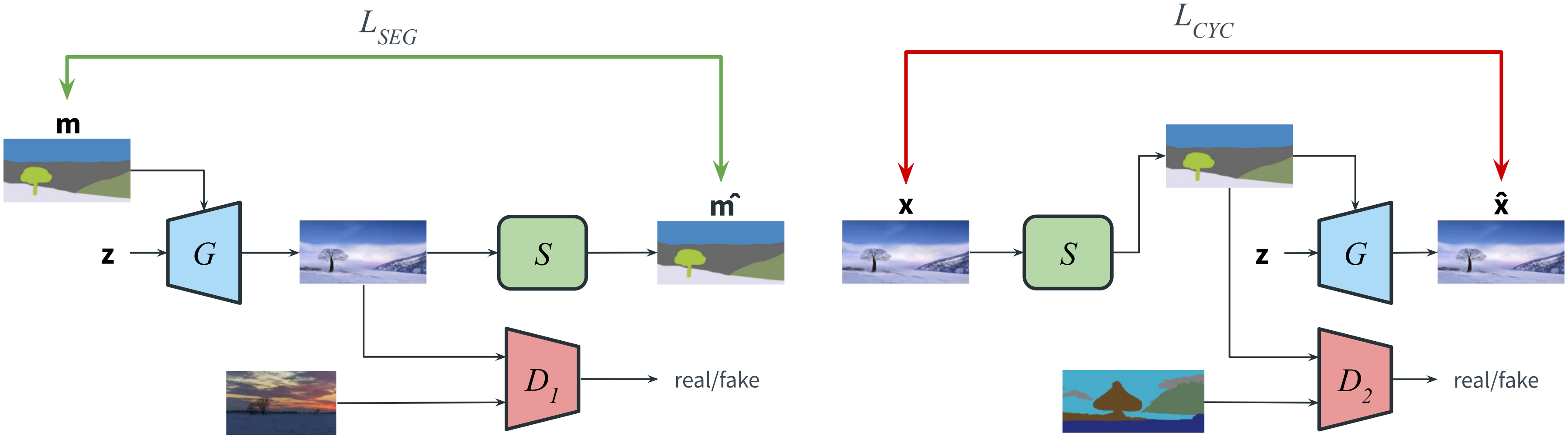


Unpaired Semantic Image Synthesis: Future works

We may also add a second discriminator that detect real/fake semantic maps in order to introduce a **cycle consistency loss:**

$$L_G = -\mathbb{E}\left[D_1(G(z, \mathbf{m}))\right] - \mathbb{E}\left[D_2(S(x))\right] + \lambda_1 \mathbb{E}\left[L_{SEG}\right] + \lambda_2 \mathbb{E}\left[L_{CYC}\right]$$

$$L_{CYC} = ||G(z, S(x)) - x||_1$$



Unpaired Semantic Image Synthesis: Recap

- No need for **paired** label-image samples (e.g. we could create **synthetic input labels**)
- Secondary result: **unsupervised semantic segmentation** model
- The **rescaled loss** give more importance to classes with small objects
- We can exploit pretrained semantic segmentation models, if available
- If a set of paired samples is available (even small), it can be exploited to further train the semantic segmentation network

Thank you for the attention