Semantic Image synthesis with Spatially-Adaptive Normalization

Sanghyuck Na

August, 8, 2019

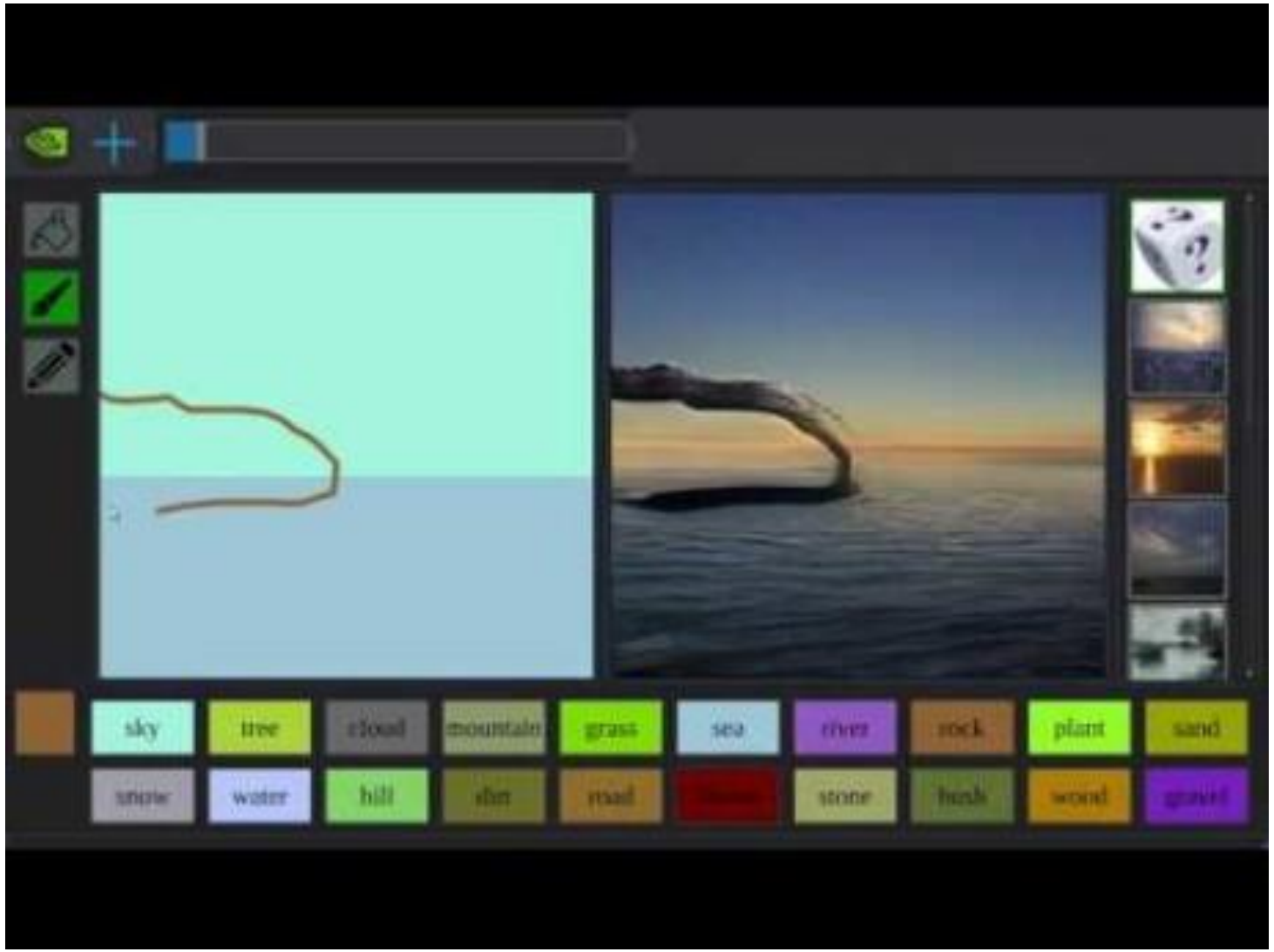
Dongguk University

Artificial Intelligence Laboratory

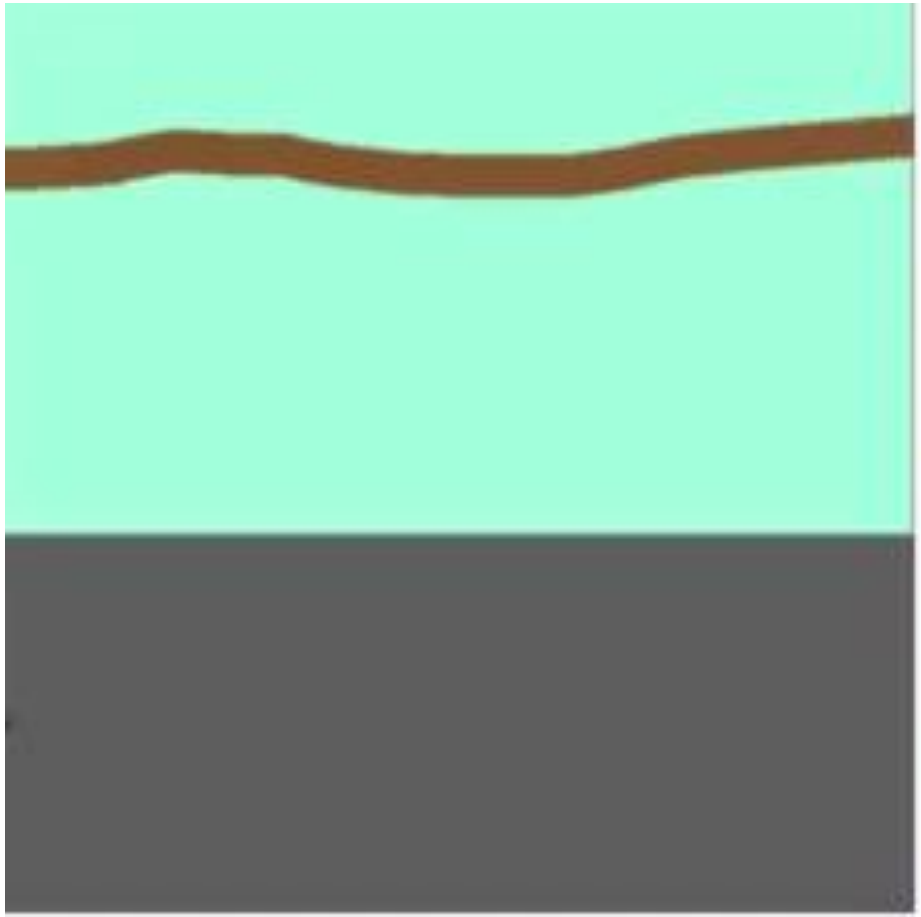
shna@Dongguk.edu

0 Contents

1 Introduction



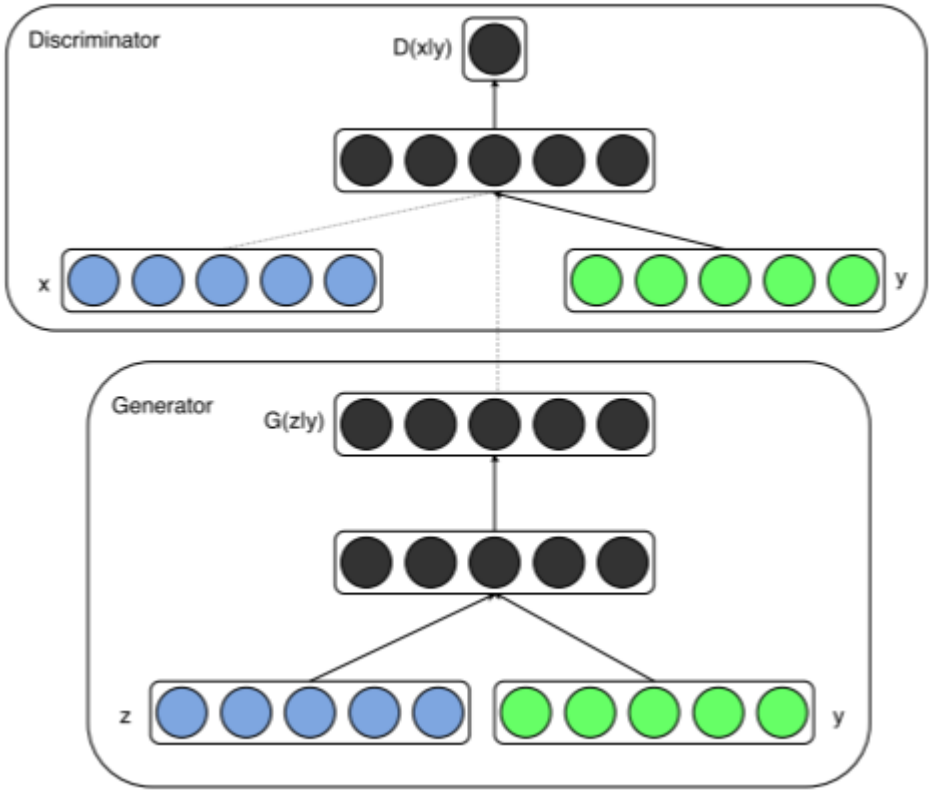
1 Introduction



1 Introduction

Conditional Generative adversarial networks

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[log(D(x|y)] + \mathbb{E}_{z\sim p_z(z)}[log\big(1 - D\big(G(z|y)\big)\big]$$



2 Pix2pixHD

Input image

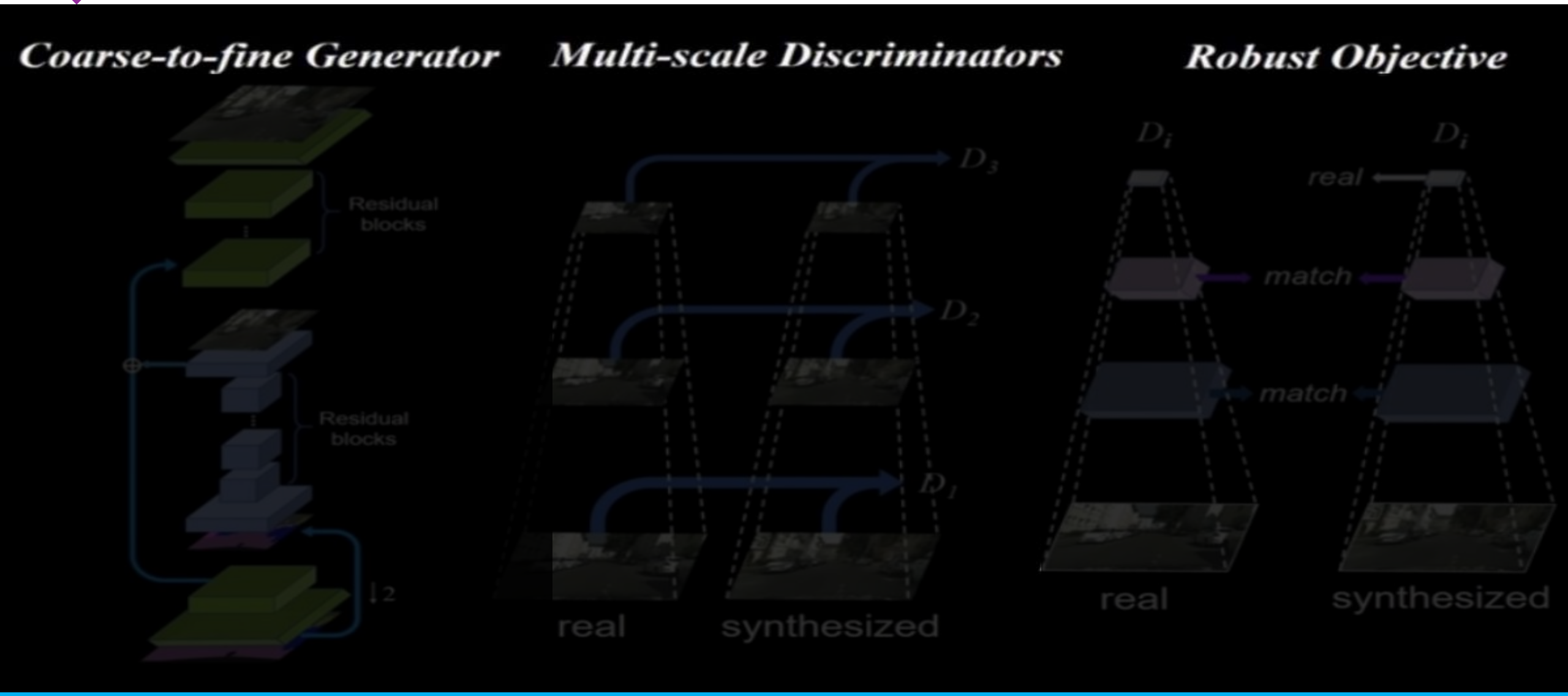
Semantic label map

Instance label map

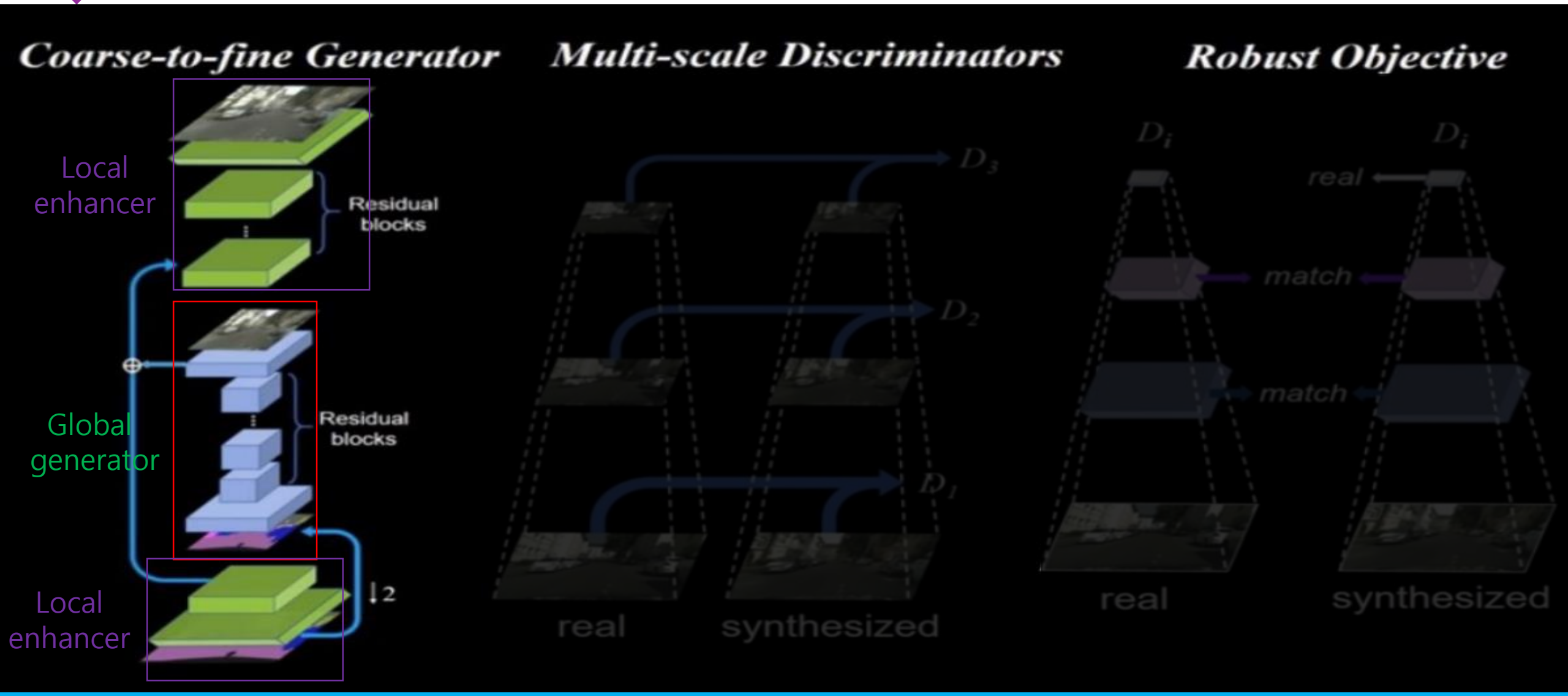
Semantic label maps don’t distinguish object of the same class

Instance label maps has a unique ID for each individual object

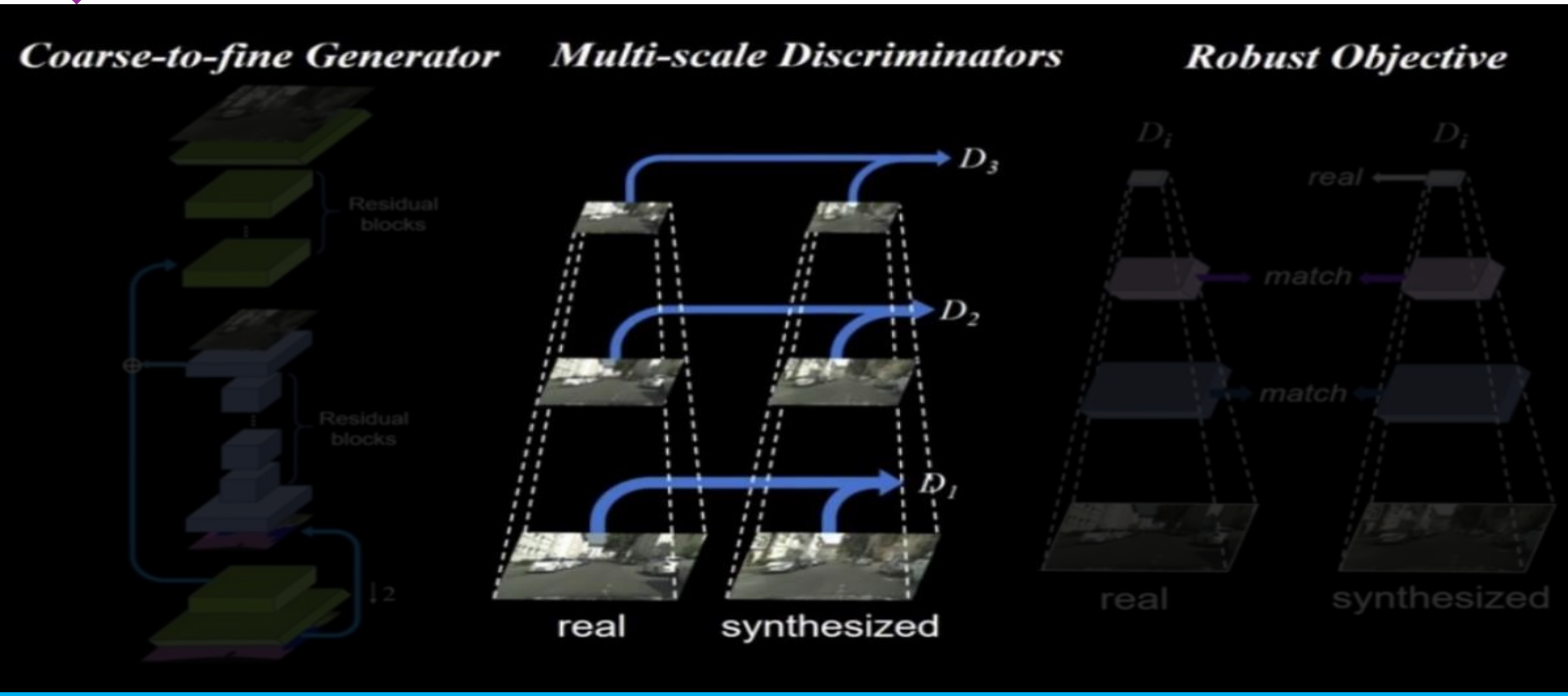
2 Pix2pixHD


Coarse-to-fine Generator
Multi-scale Discriminators
Robust Objective
Residual blocks
Residual blocks
D_3
D_2
D_1
real
synthesized
D_i
D_i
real
match
match
real
synthesized

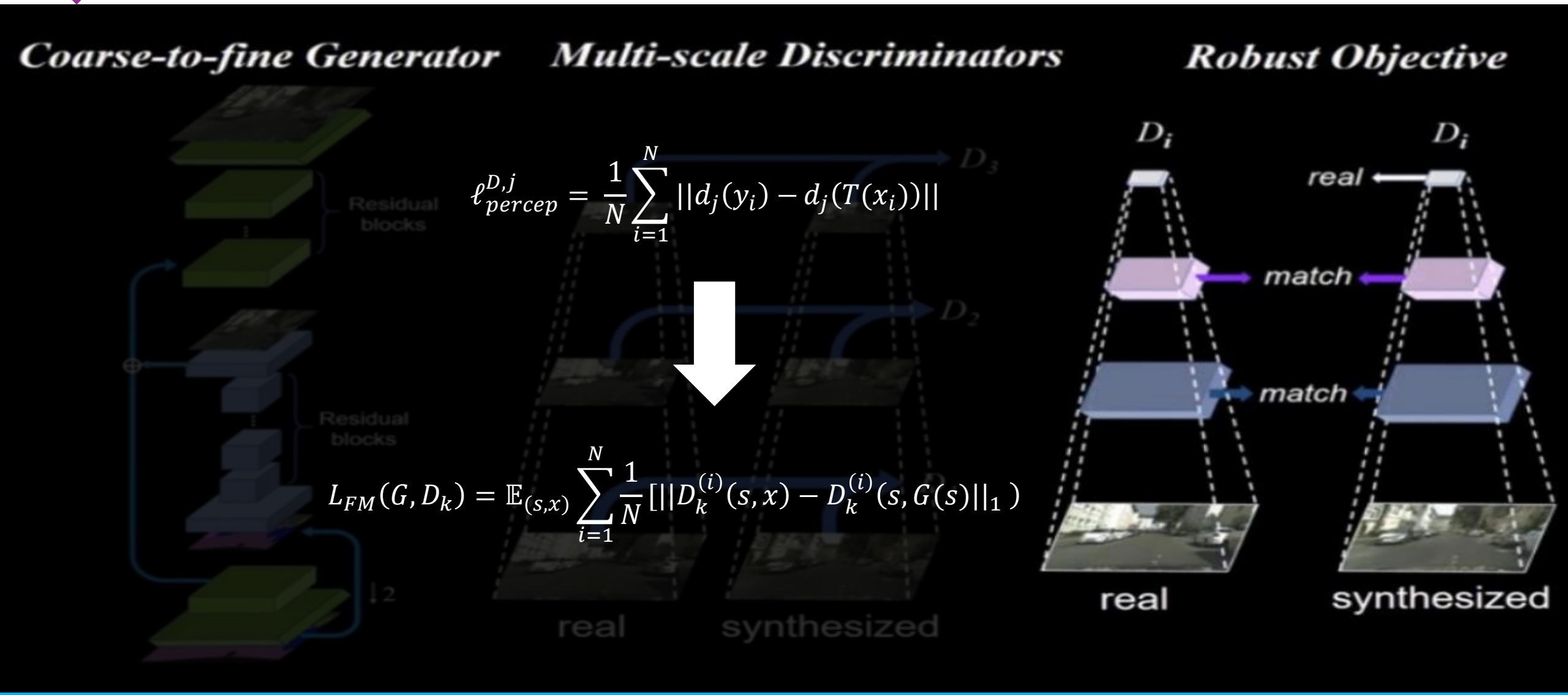
2 Pix2pixHD


Coarse-to-fine Generator
Multi-scale Discriminators
Robust Objective
Local enhancer
Residual blocks
Global generator
Residual blocks
Local enhancer
↓2
D_3
D_2
D_1
real
synthesized
D_i
D_i
real
match
match
real
synthesized

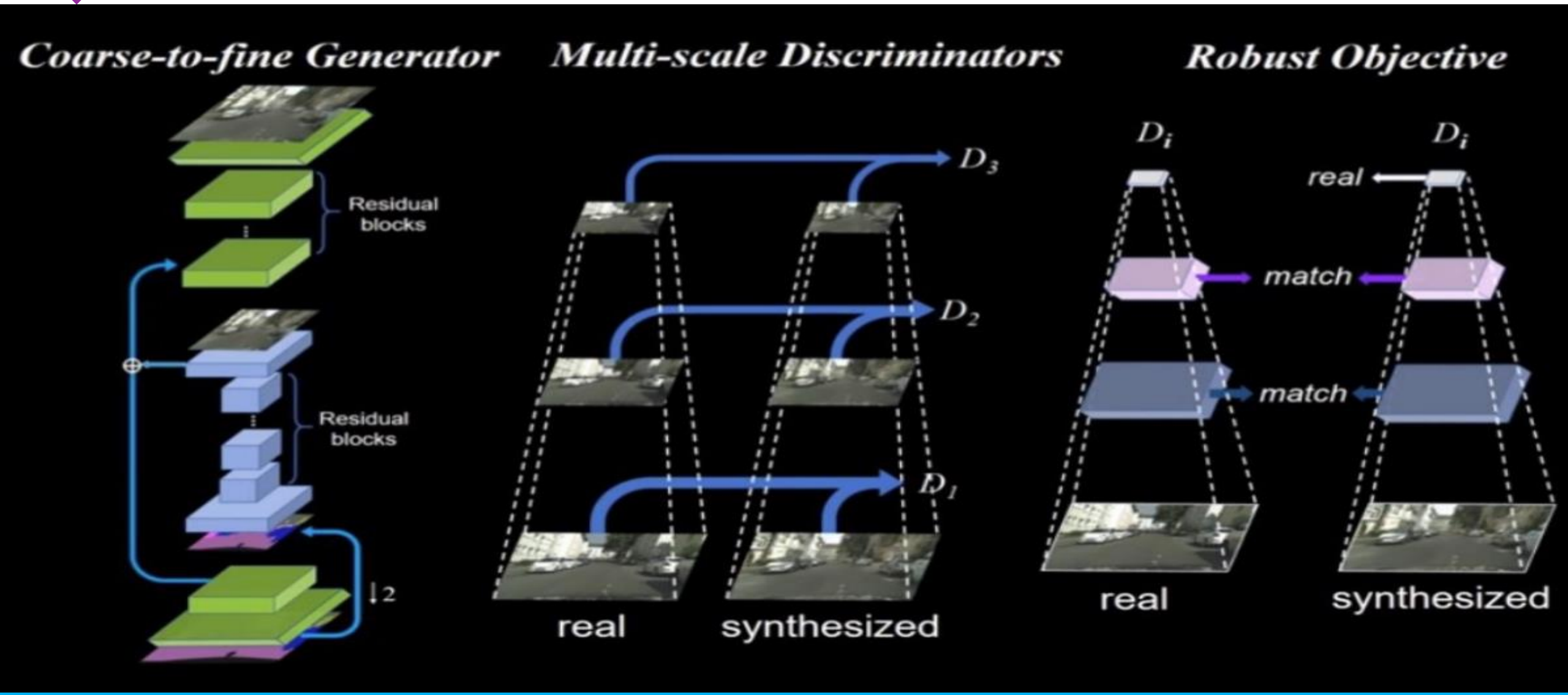
2 Pix2pixHD


Coarse-to-fine Generator
Multi-scale Discriminators
Robust Objective
Residual blocks
Residual blocks
2
D_3
D_2
D_1
real
synthesized
D_i
D_i
real
match
match
real
synthesized

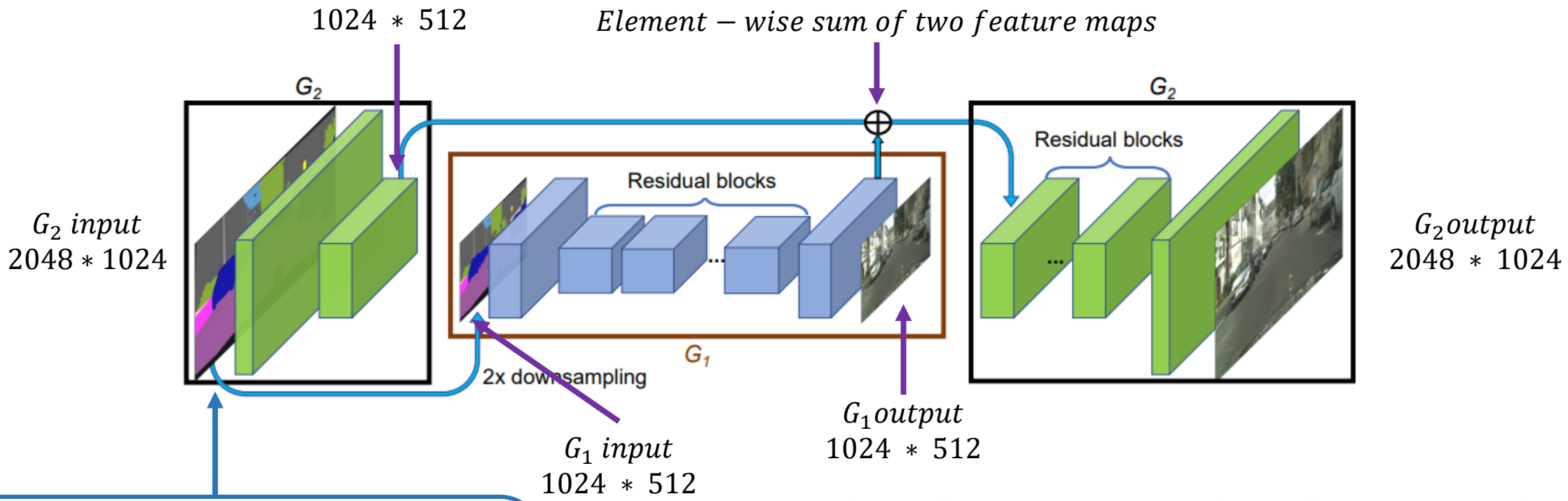
2 Pix2pixHD

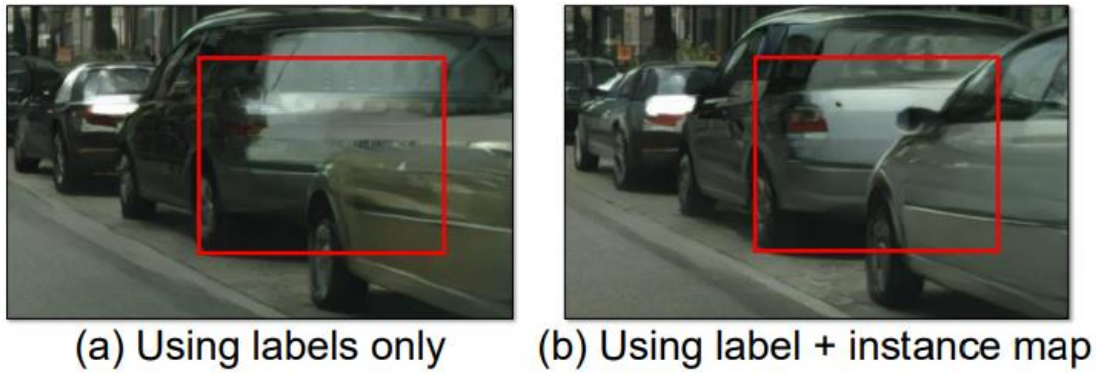

Coarse-to-fine Generator
Multi-scale Discriminators
Robust Objective
Residual blocks
Residual blocks
D_3
D_2
real
synthesized
$\ell^{D,j}_{percep} = \frac{1}{N}\sum_{i=1}^{N} ||d_j(y_i) - d_j(T(x_i))||$
$L_{FM}(G, D_k) = \mathbb{E}_{(s,x)} \sum_{i=1}^{N} \frac{1}{N} [||D_k^{(i)}(s, x) - D_k^{(i)}(s, G(s)||_1)$
D_i
D_i
real
match
match
real
synthesized

2 Pix2pixHD

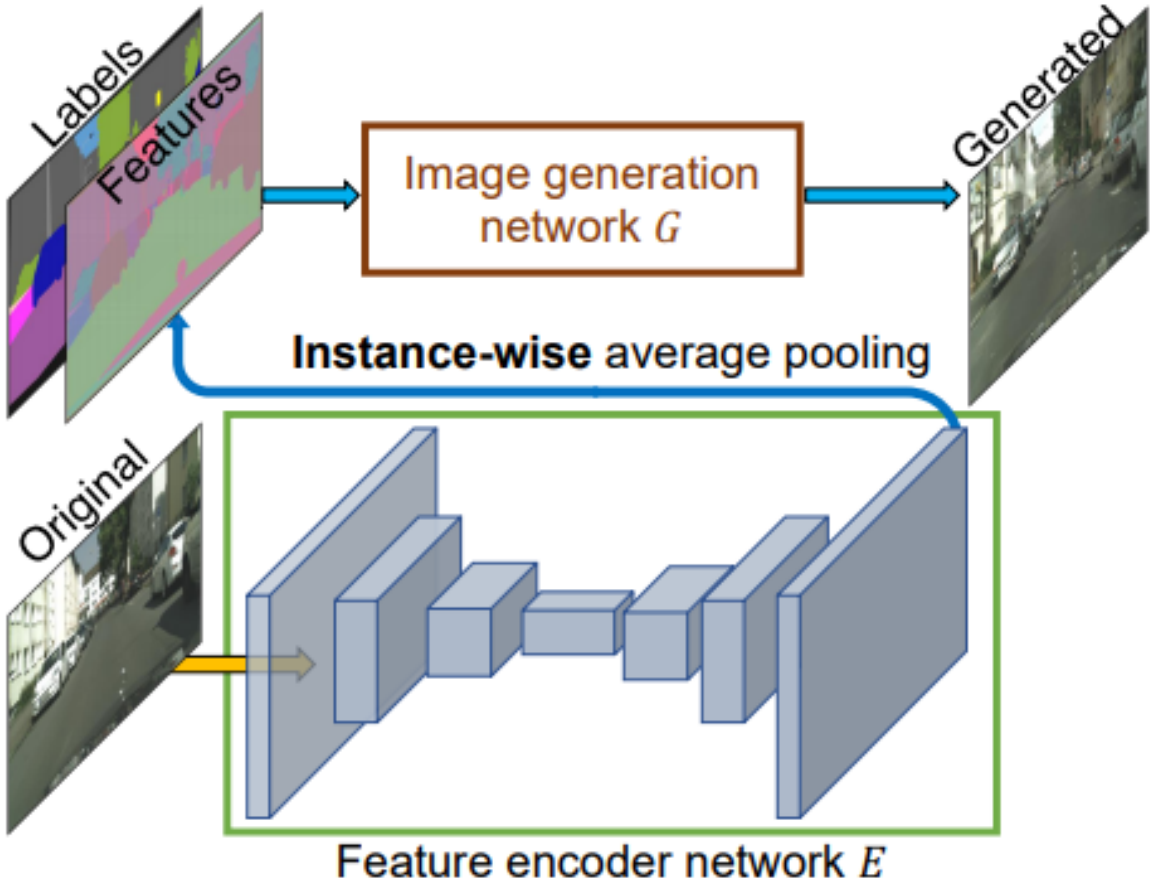

Coarse-to-fine Generator
Multi-scale Discriminators
Robust Objective
Residual blocks
Residual blocks
2
D_3
D_2
D_1
real
synthesized
D_i
D_i
real
match
match
real
synthesized

2 Pix2pixHD


1024 * 512
Element − wise sum of two feature maps
G_2
G_2
Residual blocks
Residual blocks
G_2 input
2048 * 1024
G_2output
2048 * 1024
G_1
2x downsampling
G_1 input
1024 * 512
G_1output
1024 * 512
concat
(a) Semantic labels
(b) Boundary map


(a) Using labels only
(b) Using label + instance map

2 Pix2pixHD


Labels
Features
Image generation network G
Generated
Instance-wise average pooling
Original
Feature encoder network E

Least Squares generative adversarial networks

$$\min_D V_{LSGAN}(D) = \frac{1}{2}\mathbb{E}_{x\sim p_{data}(x)}[(D(x)-b)^2] + \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-a)^2]$$

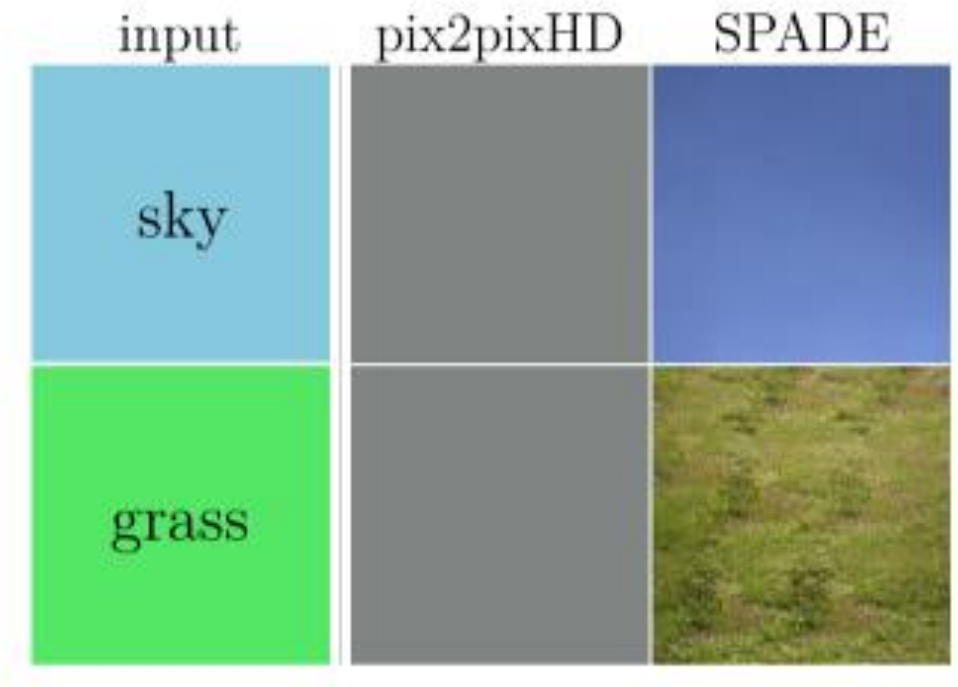
$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z)}[(D(G(z))-c)^2]$$

a = fake label 0
b = real label 1
c = the value that G wants to do for fake data 1

GAN loss used by pix2pixHD

$$\min_{G}((\max_{D_1,D_2,D_3}\sum_{k=1,2,3}L_{GAN}(G,D_k))+\lambda\sum_{k=1,2,3}L_{FM}(G,D_k))$$

Pix2pixHD



What's wrong with pix2pixHD?

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

3 Normalization

- Batch normalization

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i$$ mini-batch mean

$$\sigma_{\mathcal{B}}^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_{\mathcal{B}})^2$$ mini-batch variance

$$\hat{x_i} \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma_{\mathcal{B}}^2 + \epsilon}}$$ normalize

$$y_i \leftarrow \gamma \hat{x_i} + \beta \equiv BN_{\gamma,\beta}(x_i)$$ scale and shift

Reduces the internal covariate shift

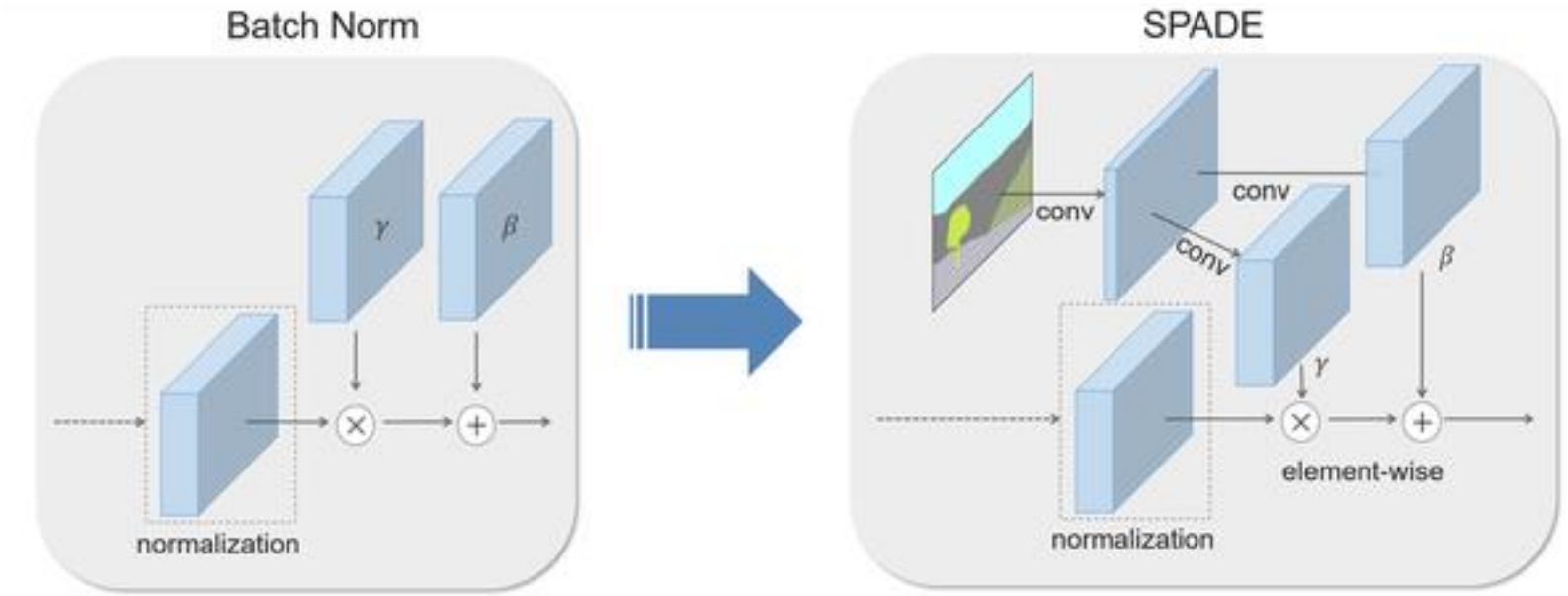
Faster training

Higher accuracy

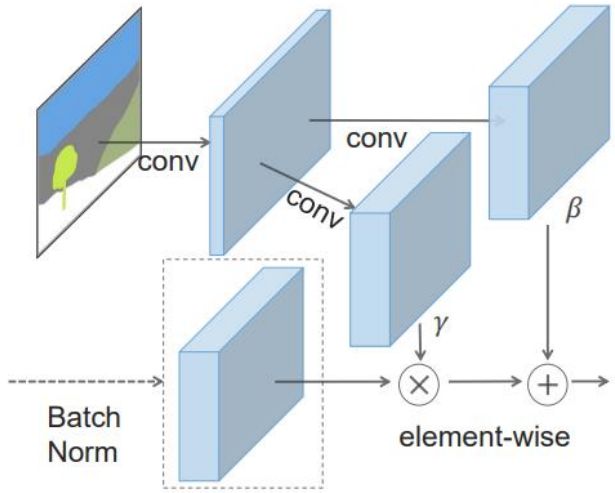
Higher learning rate

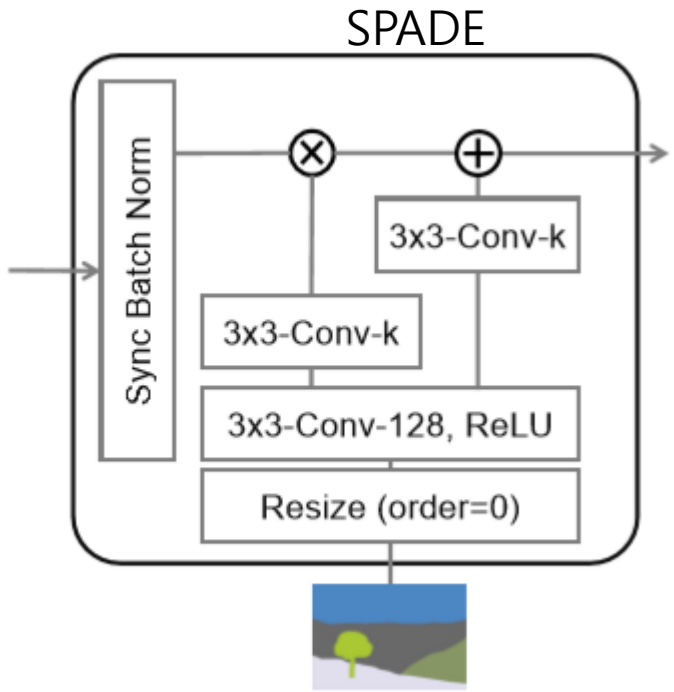
Reduces the need for dropout

4 spatially-adaptive denormalization



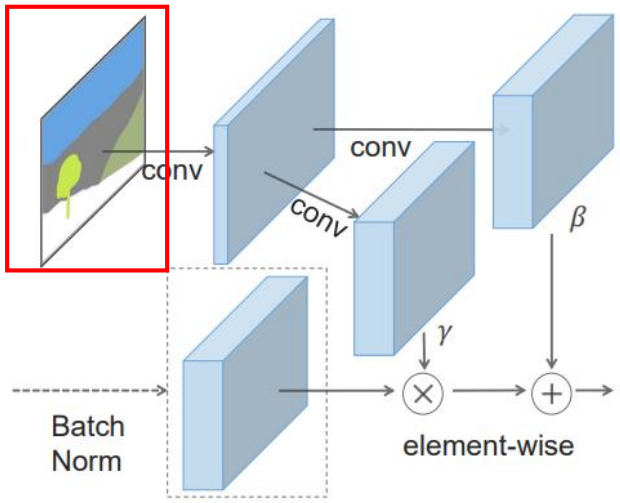
spatially-adaptive denormalization





4 spatially-adaptive denormalization

Segmentation mask



$$\mu_c^i = \frac{1}{NH^iW^i}\sum_{n,y,x} h_{n,c,y,x}^i$$

$$\sigma_c^i = \sqrt{\frac{1}{NH^iW^i}\sum_{n,y,x}(h_{n,c,y,x}^i)^2 - (\mu_c^i)^2}$$

$$\gamma_{c,y,x}^i(m)\frac{h_{n,c,y,x}^i - \mu_c^i}{\sigma_c^i} + \beta_{c,y,x}^i(m)$$

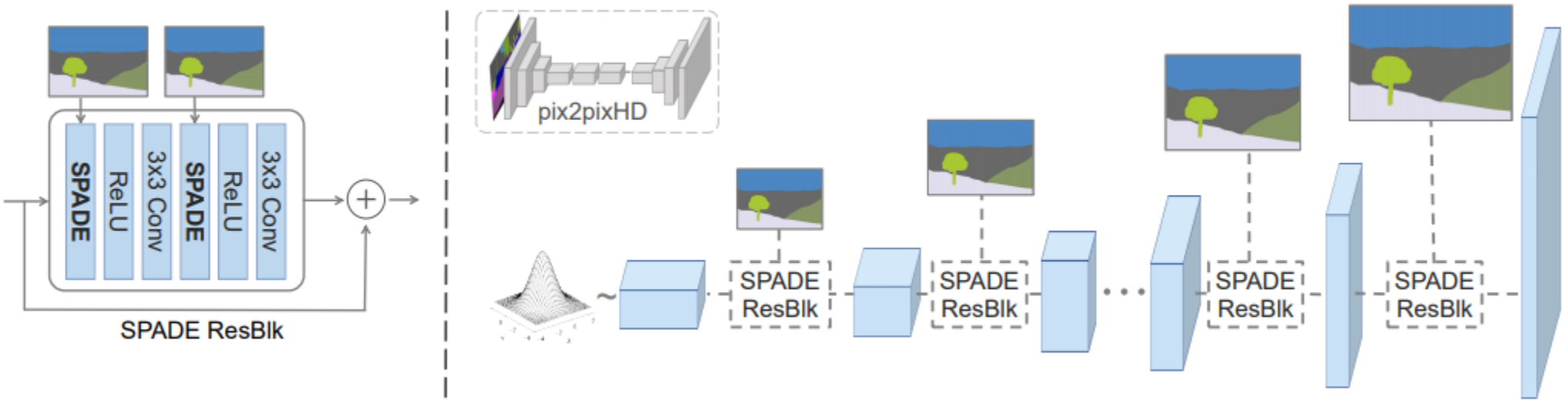
$m \in \mathbb{L}^{H*W}$
m : semantic segmentation mask
$\mathbb{L}$: semantic labels
h^i : activations of the i-th layer
C^i : channels in the layer
H^i : activation map height
W^i : activation map width
N : batch of N samples

$n \in N, c \in C^i, y \in H^i, x \in W^i$

μ_c^i = mean
σ_c^i = standard deviation

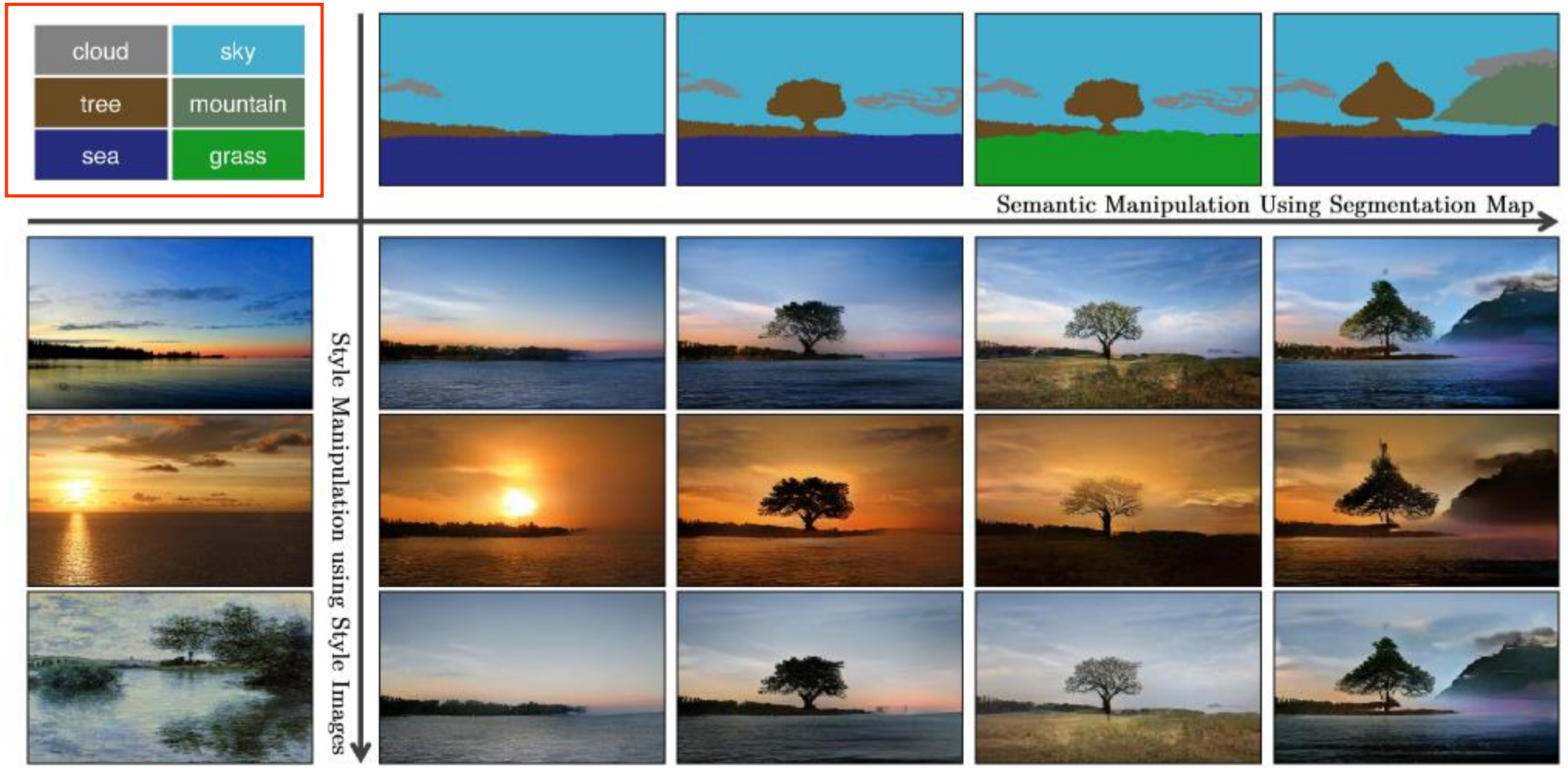
$\beta_{c,y,x}^i(m)$: modulation parameters
$\gamma_{c,y,x}^i(m)$ :: modulation parameters

spatially-adaptive denormalization



5 Result & Discussion

Label maps



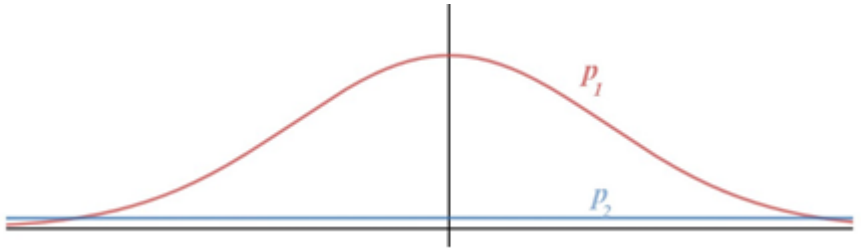
5 Result & Discussion

5 Result & Discussion

5 Result & Discussion

5 Result & Discussion

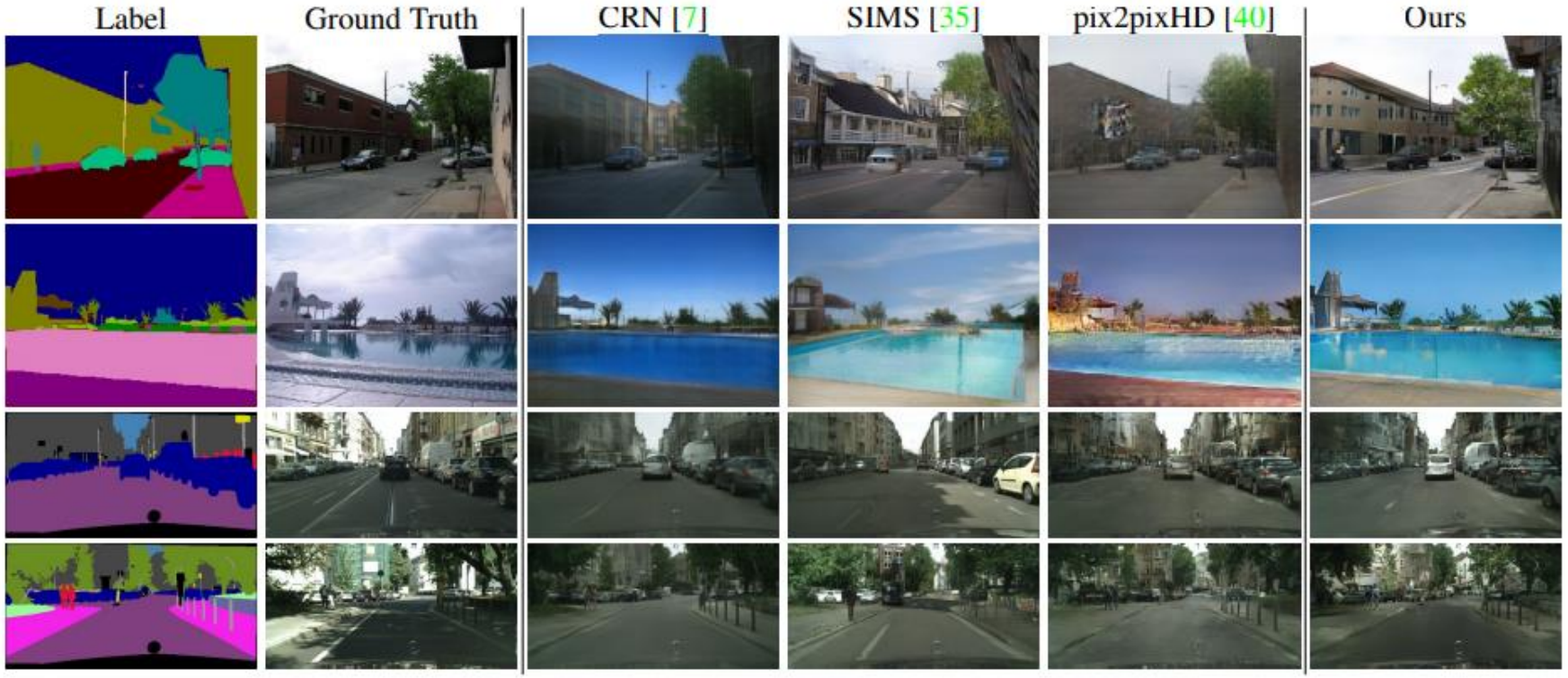
- (Fréchet Inception Distance)FID
- Pixel accuracy
- (mean Intersection-over-Union)mIoU



$$Accuracy = \frac{match\ pixel}{All\ pixel}$$

$$mIou = \frac{1}{n}\frac{Area\ of\ Overlap}{Area\ of\ Union}$$

5 Result & Discussion



Method	COCO-Stuff			ADE20K			ADE20K-outdoor			Cityscapes		
	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID	mIoU	accu	FID
CRN [7]	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 [35]	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 [40]	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

mIou Higher is better

pixel accuracy Higher is better

FID lower is better

6 Reference

- https://adamdking.com/blog/gaugan/
- https://arxiv.org/pdf/1903.07291.pdf
- https://arxiv.org/pdf/1711.11585.pdf
- http://www.vision.ee.ethz.ch/ntire18/talks/Ming-YuLiu_pix2pixHD_NTIRE2018talk.pdf
- https://www.quora.com/How-does-Conditional-Batch-normalization-work-and-how-is-it-different-from-regular-Batch-normalization
- https://arxiv.org/pdf/1502.03167.pdf
- https://dade-ai.github.io/paperclip/style/adain/README.html

Thank you