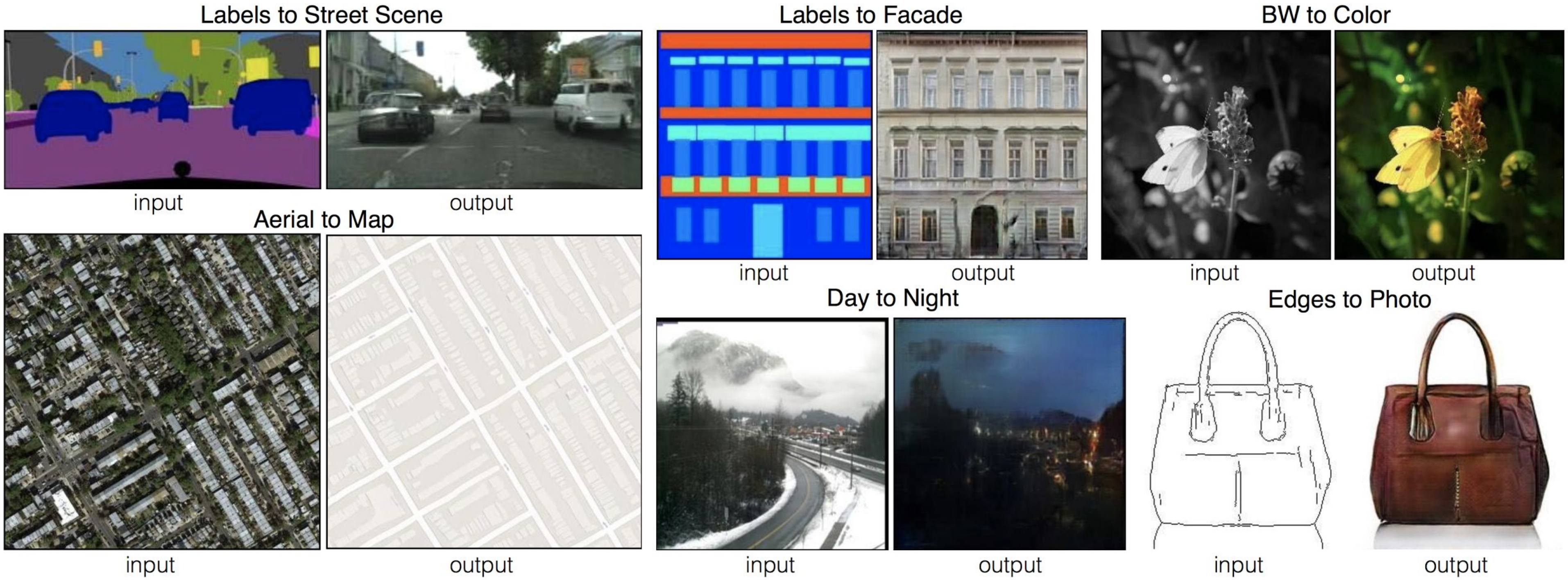
Image-to-Image Translation with Conditional Adversarial Networks

Philip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros
Presented by Alex Tong and Sam Burck

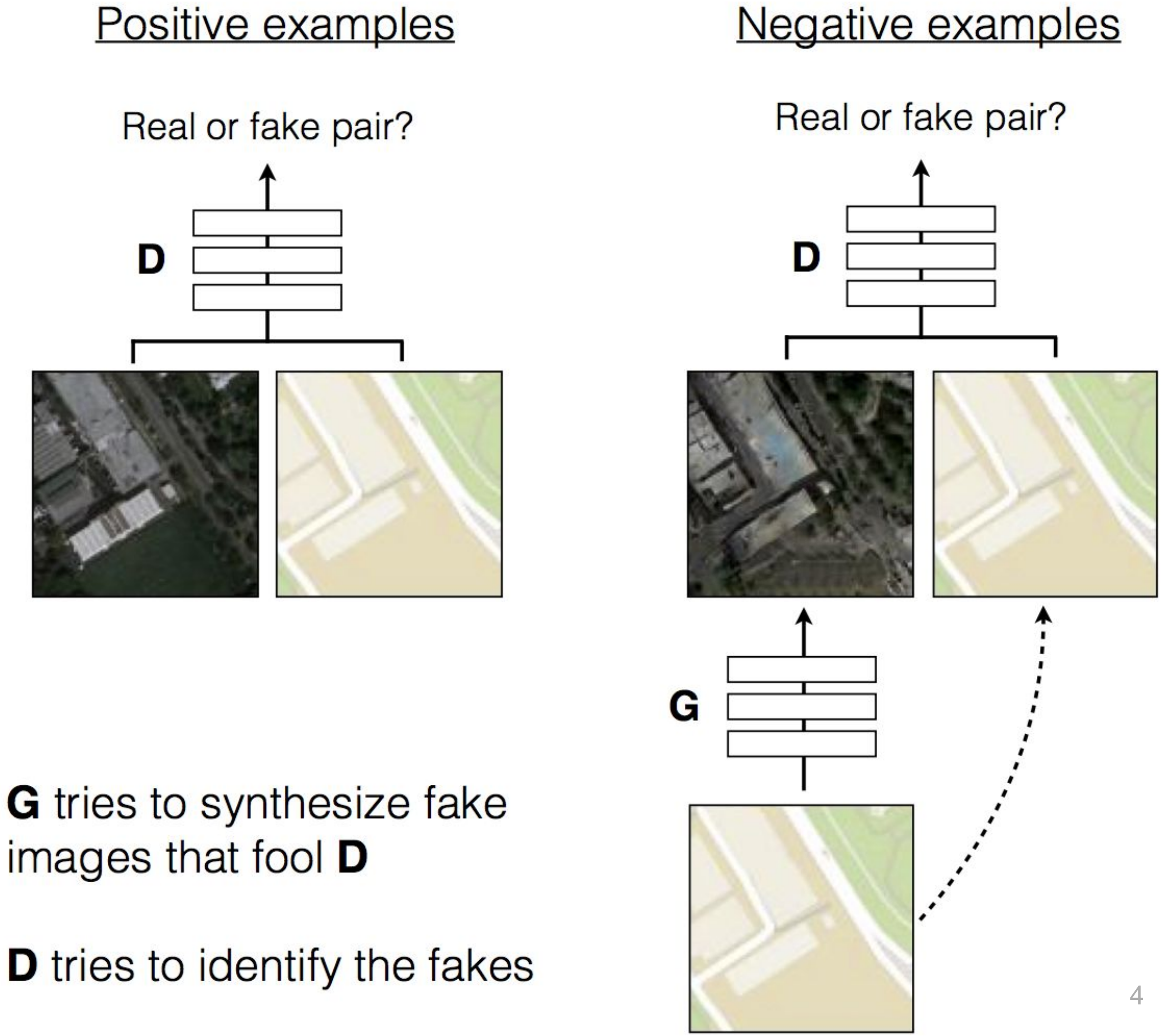
What is the problem?



What is the solution?

Conditional Generative Adversarial Networks

What are cGANs?



What are cGANs?

G : Generator Function

D : Discriminator Function

{x, y} : Image pair

Z : Noise Vector

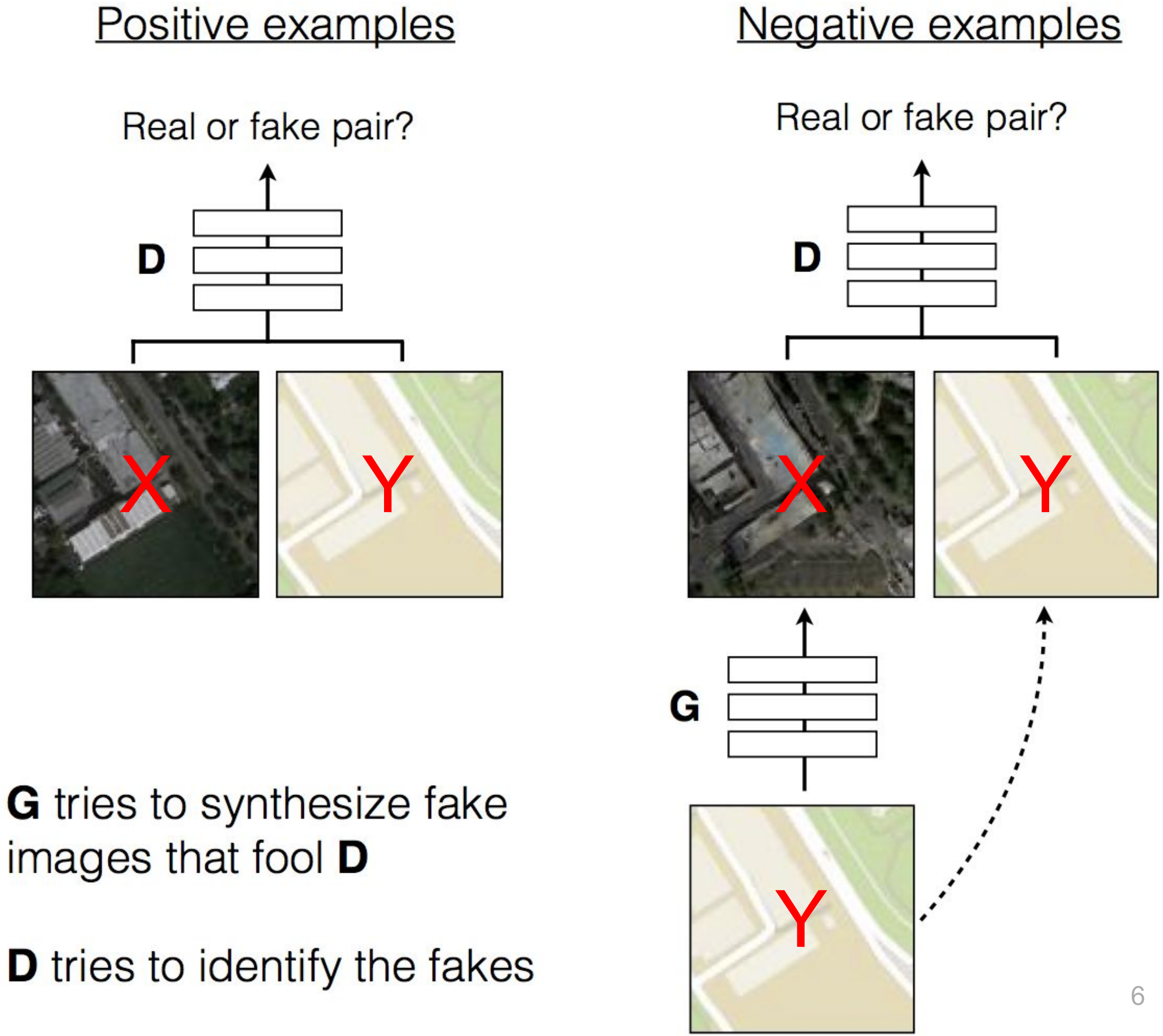
Unconditioned

- $G : z \rightarrow y$
- $D : y \rightarrow [0, 1]$

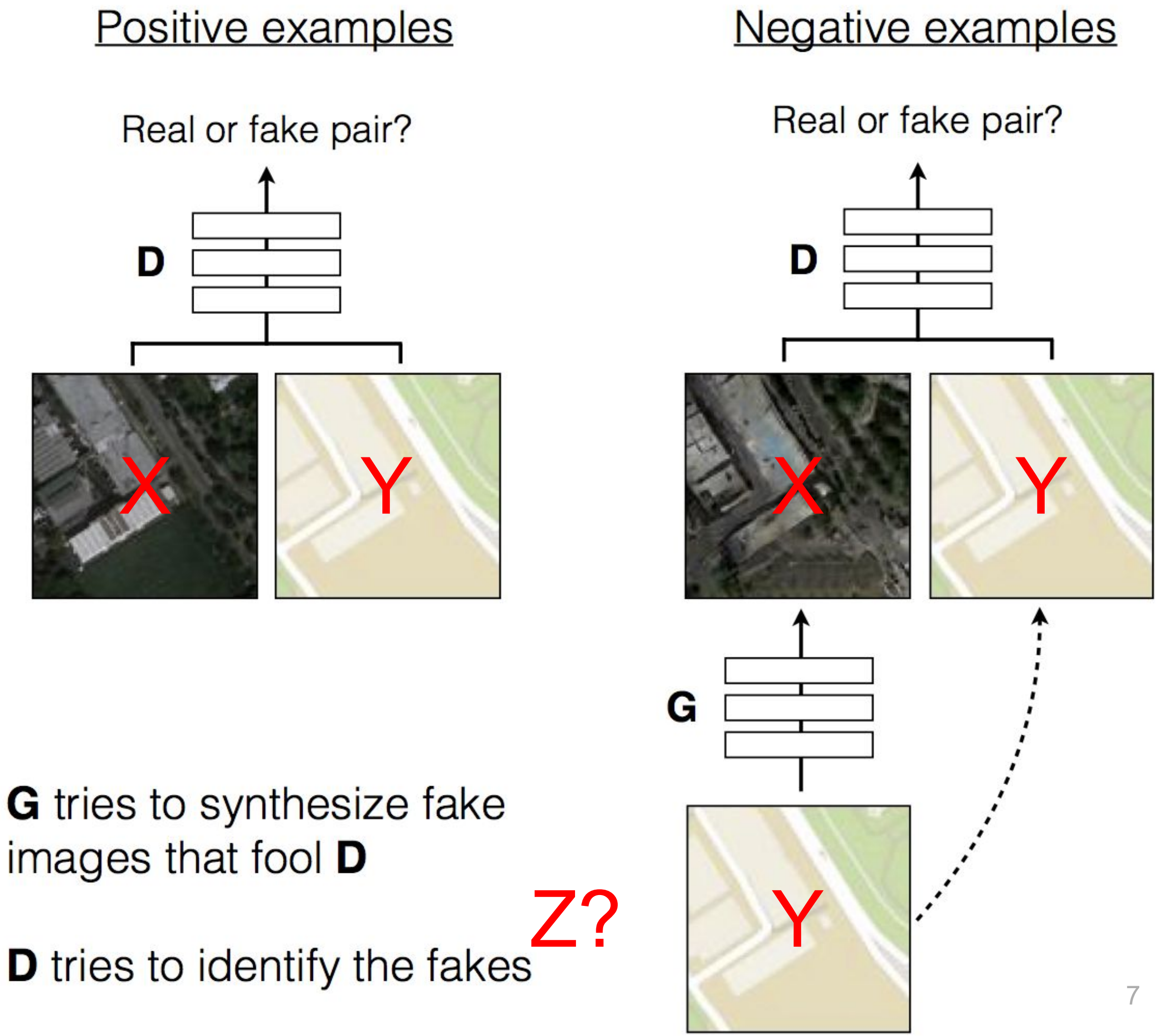
Conditioned

- $G : \{x, z\} \rightarrow y$
- $D : \{x, y\} \rightarrow [0, 1]$

What are cGANs?



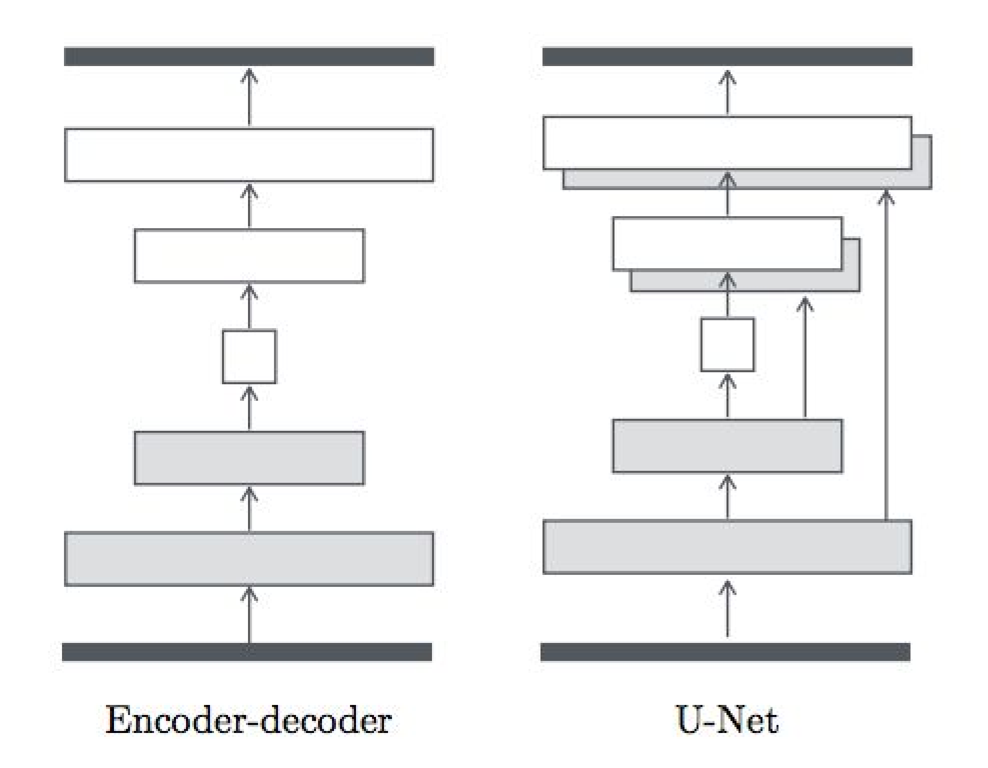
What are cGANs?



Network Architecture

Generator Architecture

- 4x4 convolutions stride 2
- No Z vector
- Dropout on d1, d2, d3

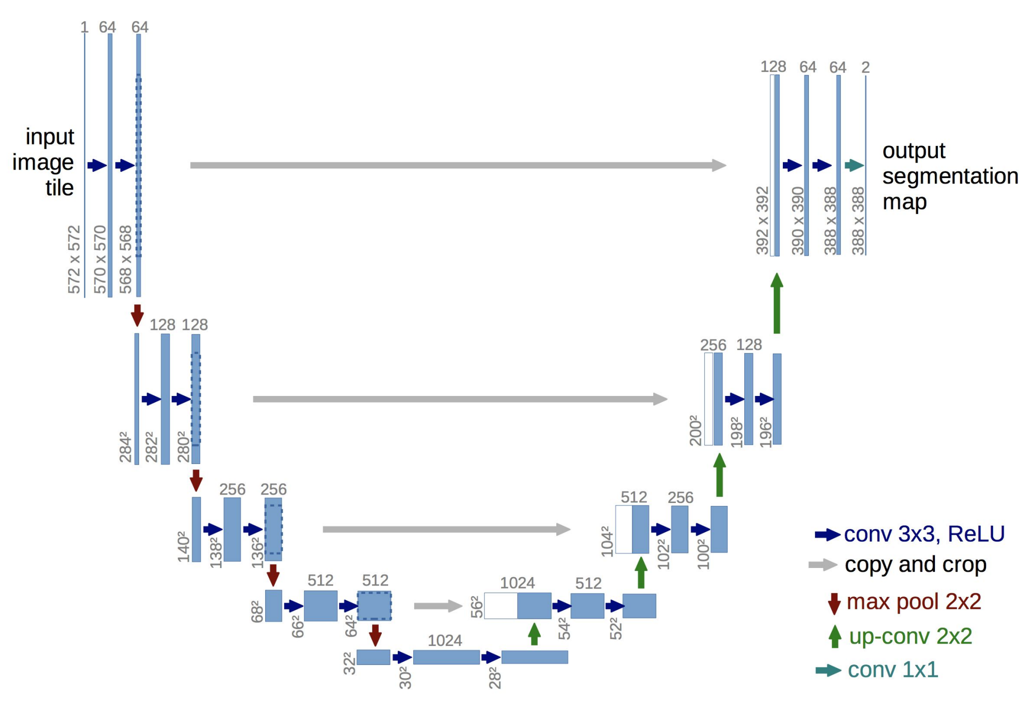


Phillip Isola, Jun-Yan Zhu, Tinghui Zhou: “Image-to-Image Translation with Conditional Adversarial Networks”, 2016; [http://arxiv.org/abs/1611.07004 arXiv:1611.07004].

U-Net

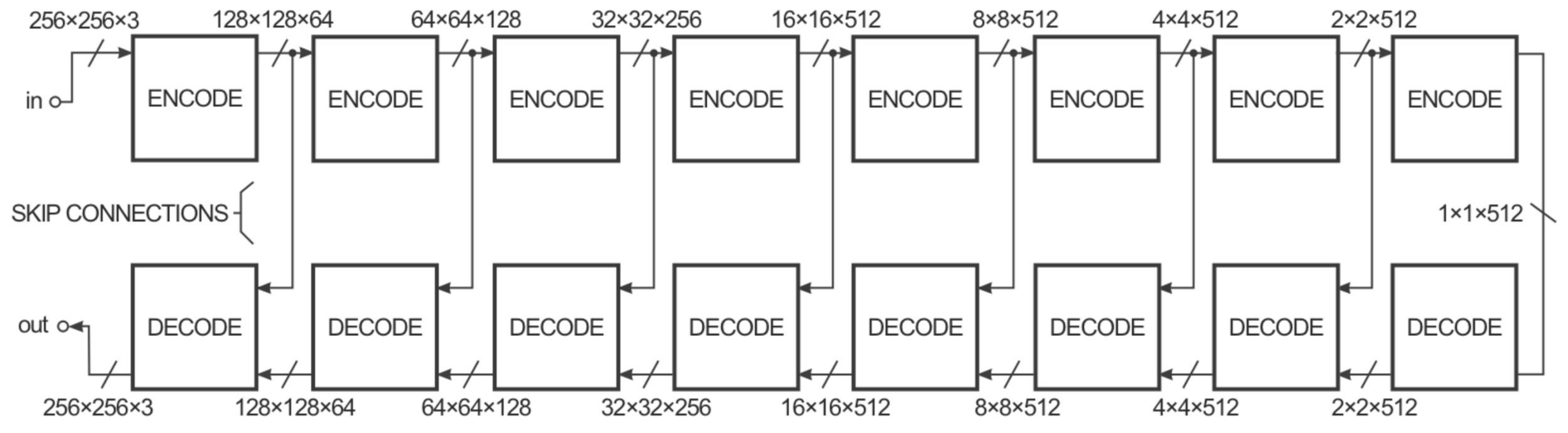
Idea: Low level features are useful

- Tasks using Encoder-Decoder that might need low level features would benefit from U-Net



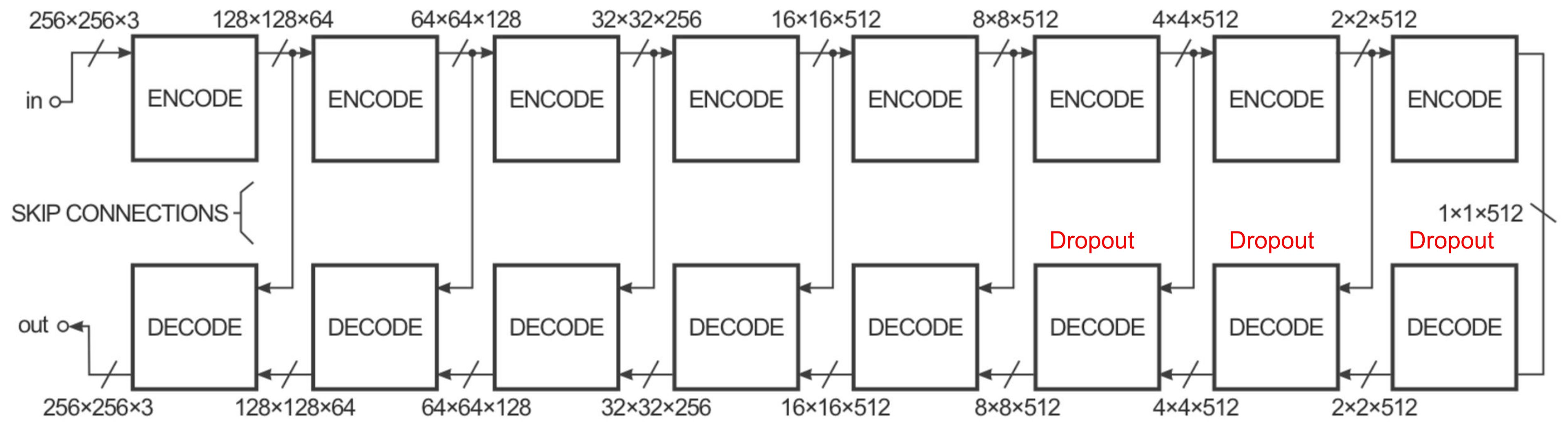
Ronneberger et al. U-Net: Convolutional Networks for Biological Image Segmentation

Generator Architecture



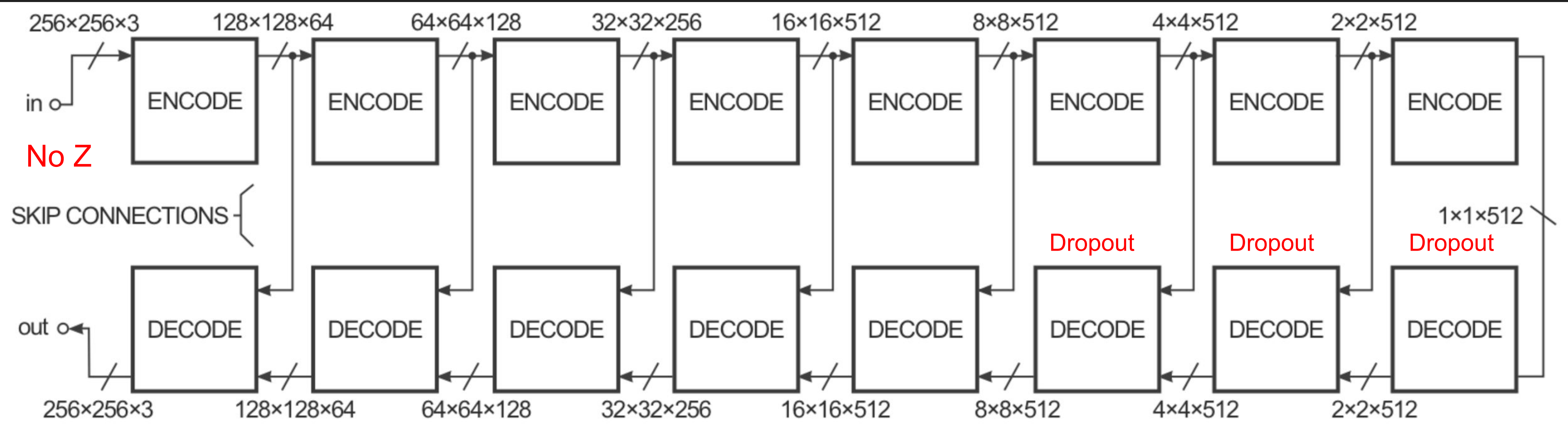
Christopher Hayes. Affine Layer Blog: Image-to-Image Demo

Generator Architecture



Christopher Hayes. Affine Layer Blog: Image-to-Image Demo

Generator Architecture



Christopher Hayes. Affine Layer Blog: Image-to-Image Demo

Objective Function

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

Why use L1 Norm?

- Less blurring than L2
- Models low frequency statistics

L1 vs. L2 Norm

$$L_1 = \sum_{i}^{n} |y_i - f(x_i)|$$

- Least Absolute Error
- Not Robust
- Unstable Solution
- Blurs around uncertain edges

$$L_2 = \sum_{i}^{n} (y_i - f(x_i))^2$$

- Least Squares Error
- Robust
- Stable Solutions
- Blurs Image

Discriminator Architecture

- Model high frequency statistics i.e. texture
    - PixelGAN, PatchGAN, ImageGAN
    - Markov Random Fields

1x1	C64 - C128
16x16	C64 - C128
70x70	C64 - C128 - C256 - C512
256x256	C64 - C128 - C256 - C512 - C512 - C512

Instance Normalization

Ulyanov et al. Texture networks: Feed-forward synthesis of textures and stylized images. (ICML 2016)

Contrast Normalization

- x : Original Image
- y : Normalized Image
- t : Images
- i : color channel
- j : width
- k : height

$$y_{tijk} = \frac{x_{tijk}}{\sum_{l=1}^{W} \sum_{m=1}^{H} x_{tilm}}$$

Ulyanov et al. Instance Normalization: The Missing Ingredient for Fast Stylization

Batch Norm

$$y_{tijk} = \frac{x_{tijk} - \mu_i}{\sqrt{\sigma_i^2 + \epsilon}}, \quad \mu_i = \frac{1}{HWT}\sum_{t=1}^{T}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_i^2 = \frac{1}{HWT}\sum_{t=1}^{T}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - mu_i)^2$$

Contrast/Instance Normalization

$$y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}}, \quad \mu_{ti} = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_{ti}^2 = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - mu_{ti})^2$$

Ulyanov et al. Instance Normalization: The Missing Ingredient for Fast Stylization

Instance Norm = Batch Norm of size 1

Instance Norm = Batch Norm of size 1

(Applied at Test Time)

(a) Content image.

(b) Stylized image.

(c) Low contrast content image.

(d) Stylized low contrast image.

Ulyanov et al. Instance Normalization: The Missing Ingredient for Fast Stylization

Results

Experiments

- Semantic labels↔photo, trained on the Cityscapes dataset
- Architectural labels→photo, trained on the CMP Facades dataset
- Map↔aerial photo, trained on data scraped from Google Maps
- BW→color photos, trained on ImageNet
- Edges→photo, trained on images of shoes from Zappos, handbags from Amazon. Data was processed using Holistically-Nested Edge Detection, developed by Saining Xie and Zhuowen Tu from UC San Diego.
- Sketch→photo: tests edges→photo models on human drawn sketches from
- Day→night, trained on ~17k images from 91 webcams. Data was augmented using jitter and mirroring.

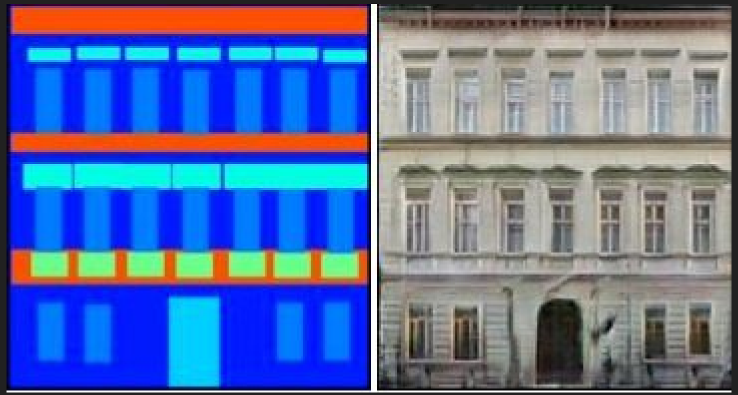
Evaluation: It's hard to evaluate synthesized images!

Amazon Mechanical Turk (Turkers)

- Real vs. Fake image trials
- Images shows for 1 second
- Unlimited time to respond

Tricking Semantic Classifiers

- FCN-8s trained on cityscape set
- Compare labels used for generation to FCN-8s output.

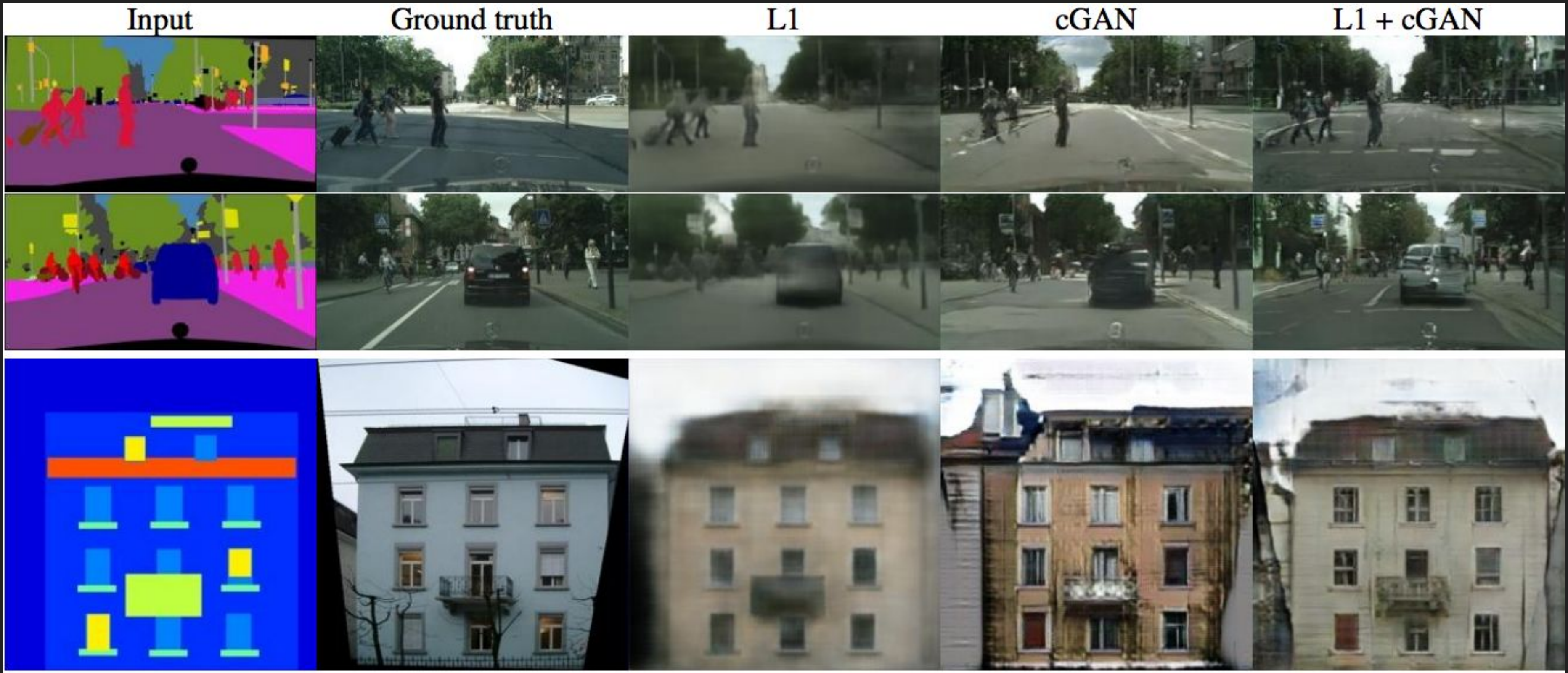


Analysis of Objective Functions

Recall our objective function:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G)$$

- We can visualize the effects of the L1 and GAN terms in objective functions.
- We expect the L1 term to steer our output towards the ground truth
- We expect the L1 term to produce output that’s more blurry than ground truth
- We expect the cGAN term to produce sharp results, however these results may have “artifacts”, and we don’t want to give our generator too much “creative liberty”.
- WHY’S THERE A LINE IN THE STREET?!?!


Input
Ground truth
L1
cGAN
L1 + cGAN

FCN Scores on Cityscapes Label ←→ Photo Dataset

Loss	Per-pixel acc.	Per-class acc.	Class IOU
L1	0.44	0.14	0.10
GAN	0.22	0.05	0.01
cGAN	0.61	**0.21**	**0.16**
L1+GAN	**0.64**	0.19	0.15
L1+cGAN	0.63	**0.21**	**0.16**
Ground Truth	0.80	0.26	0.21

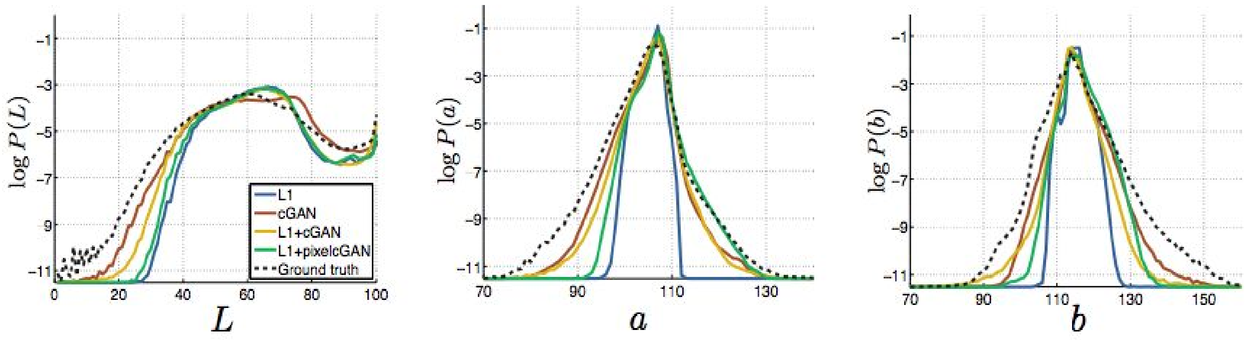
Why is the Non-Conditional GAN so bad at this???

Q: Why is the Non-Conditional GAN so bad at this???

A: When we remove conditioning, from the discriminator, our loss function no longer accounts for mismatch between input and output. In this case, all we need to get a low loss score is a realistic-looking output image. The non-conditional GAN ended up producing near-identical output for all inputs. Adding L1 loss to the GAN objective function gives much better results as the L1 loss accounts for mismatch between input and output.

Chromaticity (color stuff)

- L1 loss converges on the median of the conditional PDF of all possible colors.

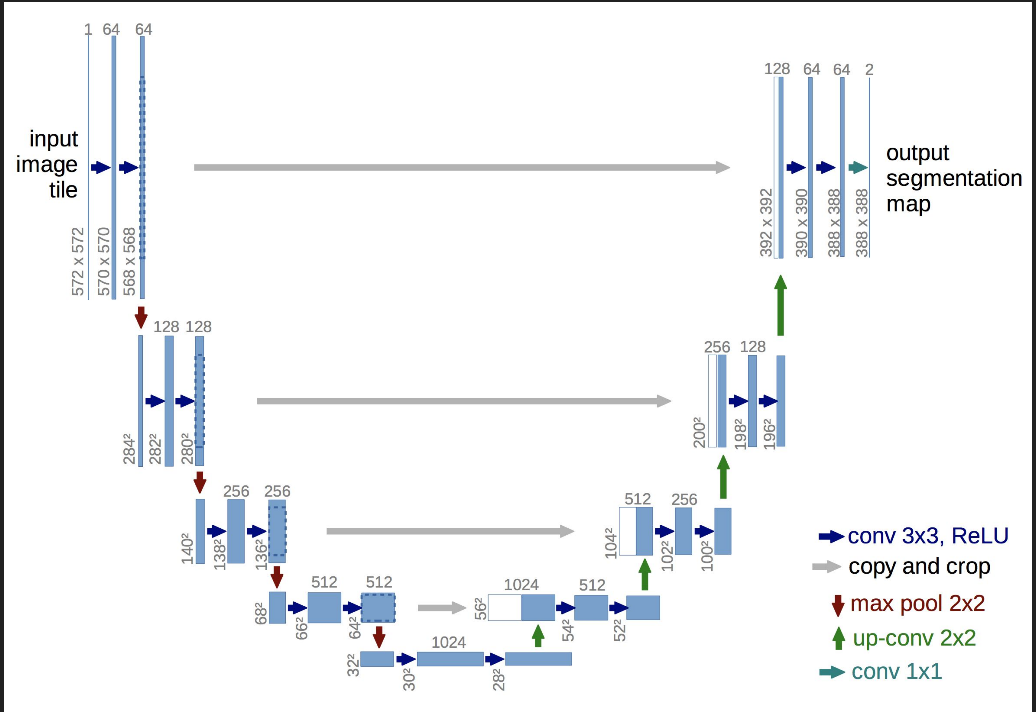


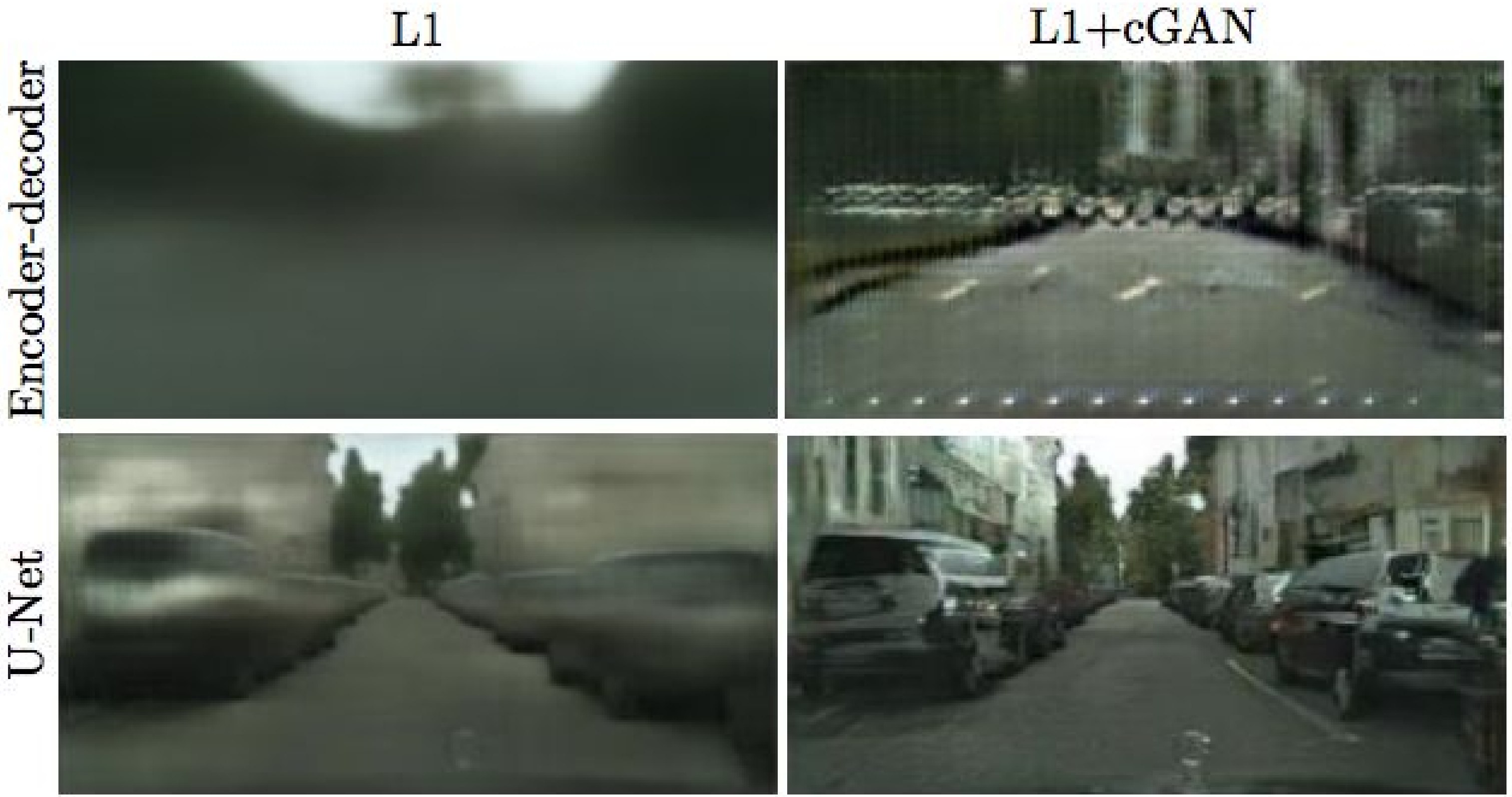
Loss	Histogram intersection against ground truth L	a	b
L1	0.81	0.69	0.70
cGAN	**0.87**	0.74	**0.84**
L1+cGAN	0.86	**0.84**	0.82
PixelGAN	0.83	0.68	0.78

- X axis is output values in lab color space
- L1 squishes the curve!

Remember the U-Net?

- Why do we use it?
- How can we evaluate it's effectiveness?
- Hint: Use scissors...




L1
L1+cGAN
Encoder-decoder
U-Net

Output of Varying Discriminator Patch Sizes

- What are the roles of our L1 loss term vs. our discriminator loss term?
- How can we tune our discriminator to optimally perform its role?
- Hint: Look at the slide title...

Names and Patch Sizes

- 1 x 1 → PixelGAN
- 256 x 256 (or max image width/height) → ImageGAN
- Anything in between → PatchGAN
- Default → 70 x 70 PatchGAN

N x N	Pixel Acc.	Class Acc.	Class IOU
1 x 1	0.44	0.14	0.10
16 x 16	0.62	0.20	**0.16**
70 x 70	**0.63**	**0.21**	**0.16**
256 x 256	0.47	0.18	0.13

IOU = Intersection over Union, metric evaluates accuracy of bounding box placement.

70 x 70 PatchGAN on Larger Images

Can you Fool a Turker?

- Tested map ← → aerial photograph, grayscale → color

RECALL:

- Real vs. Fake image trials
- Images shows for 1 second
- Unlimited time to respond

Map ← → Aerial Photograph

Loss	Photograph → Map	Map → Photograph
L1	2.8% ± 1.0%	0.8% ± 0.3%
L1 + cGAN	6.1% ± 1.3%	18.9% ± 2.5%

Colorization

Method	Synthesized Images Labeled as Real
L2 regression	16.3 ± 2.4%
Zhang et al. 2016 (CNN “classifies” colors)	27.8 ± 2.7%
Image to Image	22.5 ± 1.6%

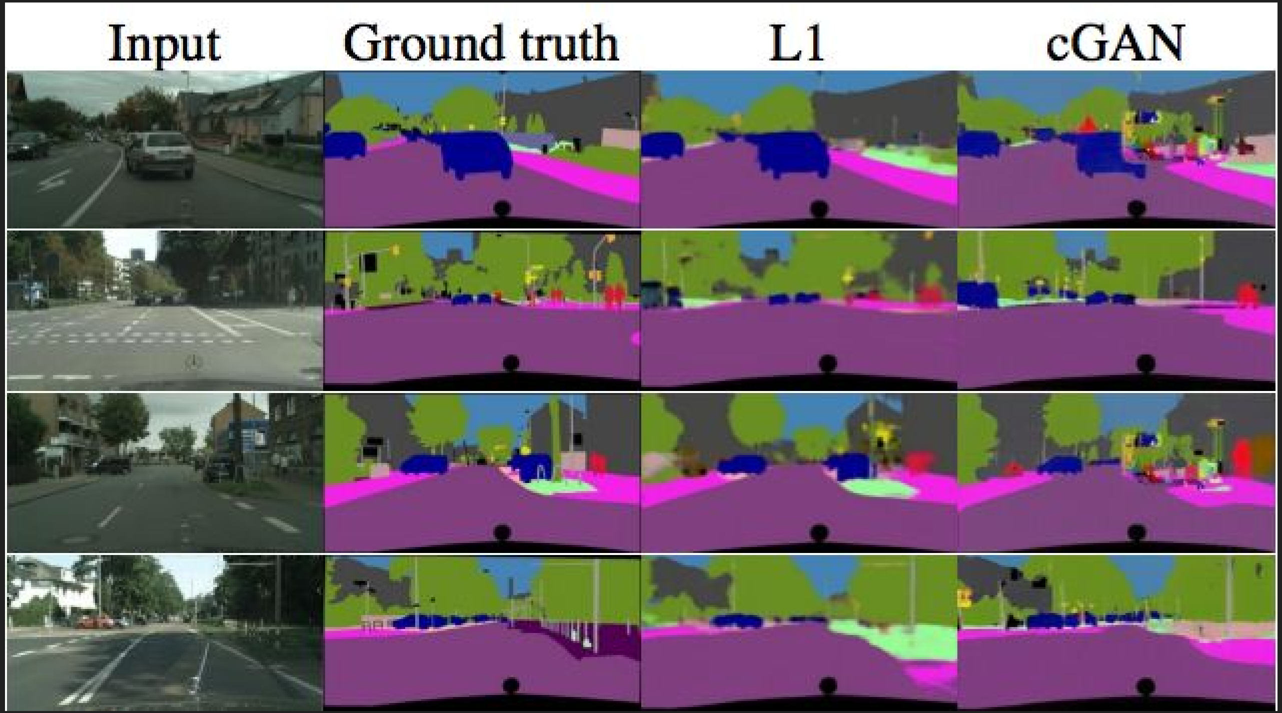
Semantic Segmentation

- cGAN’s are great for generating highly detailed images
- What about problems with simple output, such as Semantic Segmentation?

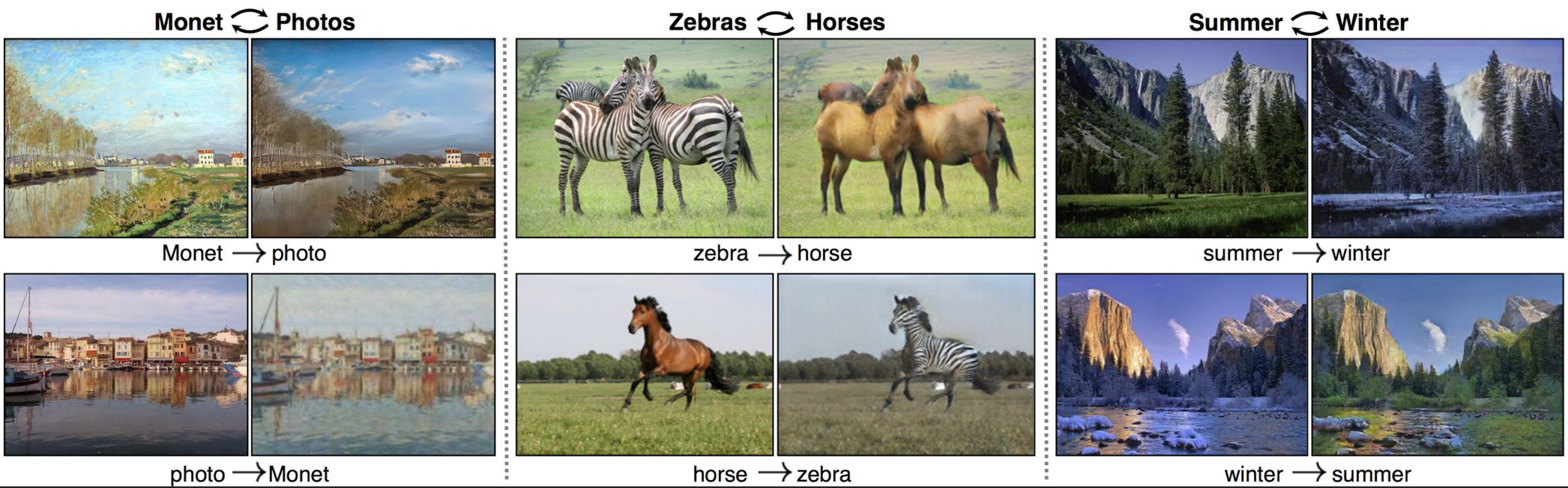
Semantic Segmentation?

- It turns out, cGAN’s are nothing special when it comes to semantic segmentation
- The authors argue that cGAN’s are strong for ambiguous, generative tasks
- L1 loss works better for semantic segmentation

Loss	Per-Pix. Acc.	Per-Class Acc.	Class IOU
L1	0.86	0.42	0.35
cGAN	0.74	0.28	0.22
L1 + cGAN	0.83	0.36	0.29



Future Work

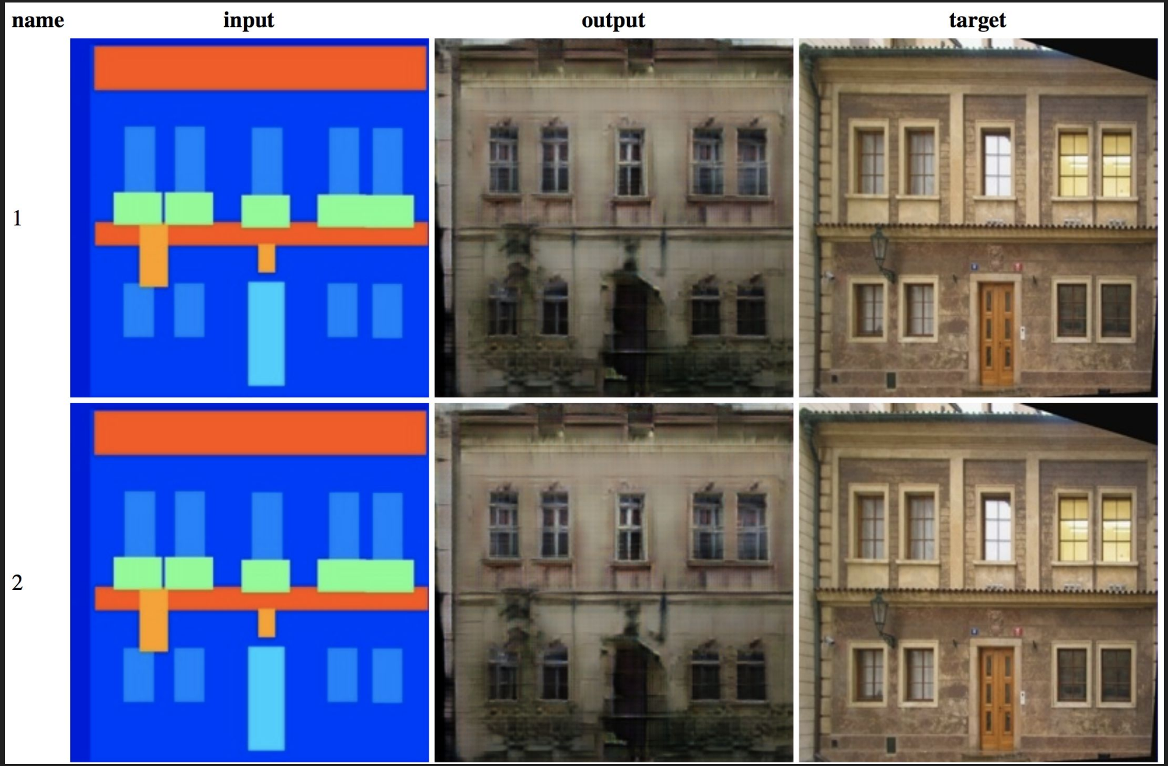


Jun-Yan Zhu et al. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

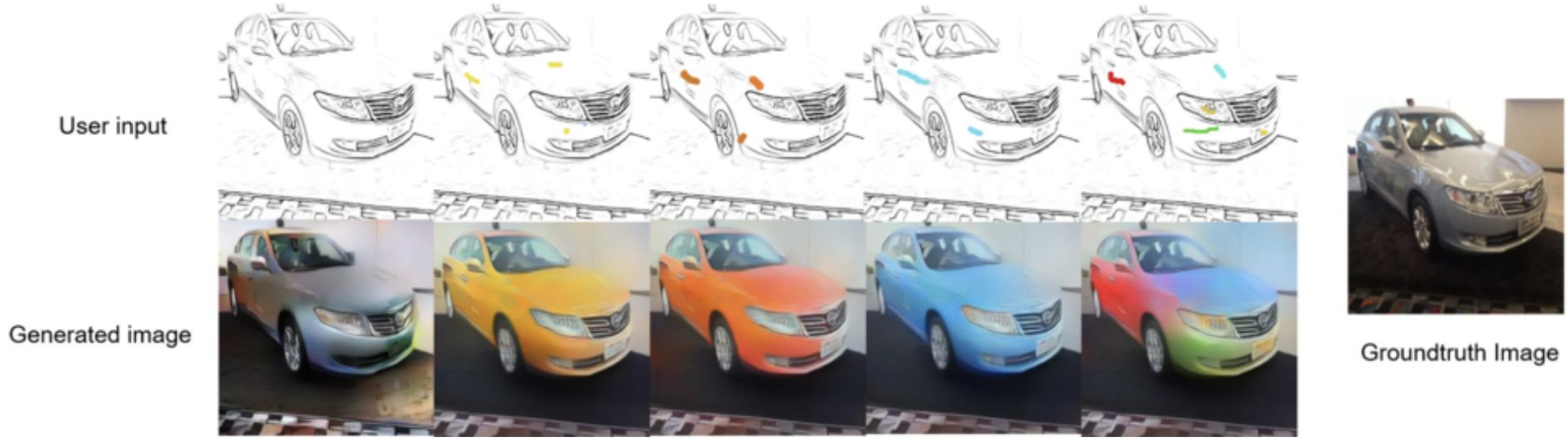
Future Work - Stochasticity

Generate more varied images

- Incorporate z vector
- More Dropout
- Ran 100 times, nearly identical results.



Future Work - Human Input



Patsorn Sangkloy. Scribbler: Controlling Deep Image Synthesis with Sketch and Color

Citations

- Saining Xie: “Holistically-Nested Edge Detection”, 2015; arXiv:1504.06375.
- Richard Zhang, Phillip Isola: “Colorful Image Colorization”, 2016; arXiv:1603.08511.
- Olaf Ronneberger, Philipp Fischer: “U-Net: Convolutional Networks for Biomedical Image Segmentation”, 2015; arXiv:1505.04597.
- Alec Radford, Luke Metz: “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks”, 2015; arXiv:1511.06434.
- Evan Shelhamer, Jonathan Long: “Fully Convolutional Networks for Semantic Segmentation”, 2016; arXiv:1605.06211.
- Chuan Li: “Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis”, 2016; arXiv:1601.04589.
- Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville: “Generative Adversarial Networks”, 2014; arXiv:1406.2661.