Image-to-Image Translation with Conditional Adversarial Networks

조재춘 소아람 이찬희 김규경

KOREA UNIVERSITY

Natural Language Processing & Artificial Intelligence

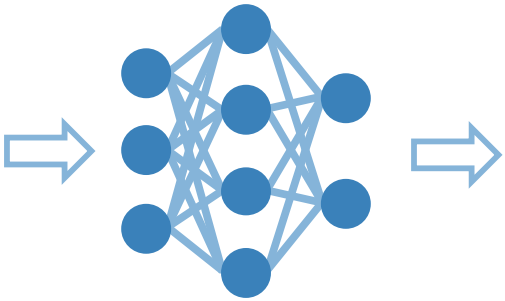
INDEX

Background

Brief Introduction to Generative Adversarial Nets

Neural Networks

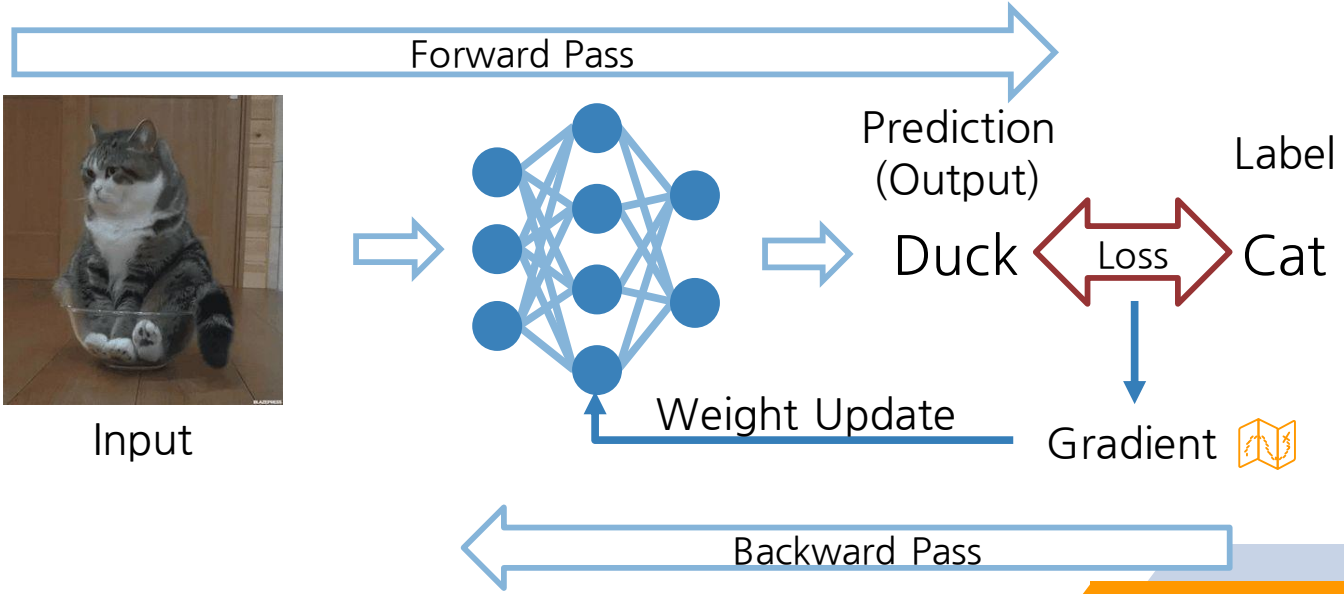
- Transforms input to output
- NN is a “set of weights(parameters)”
- Need to change the weights(train) to make it do what we want



Cat

How to Train a NN

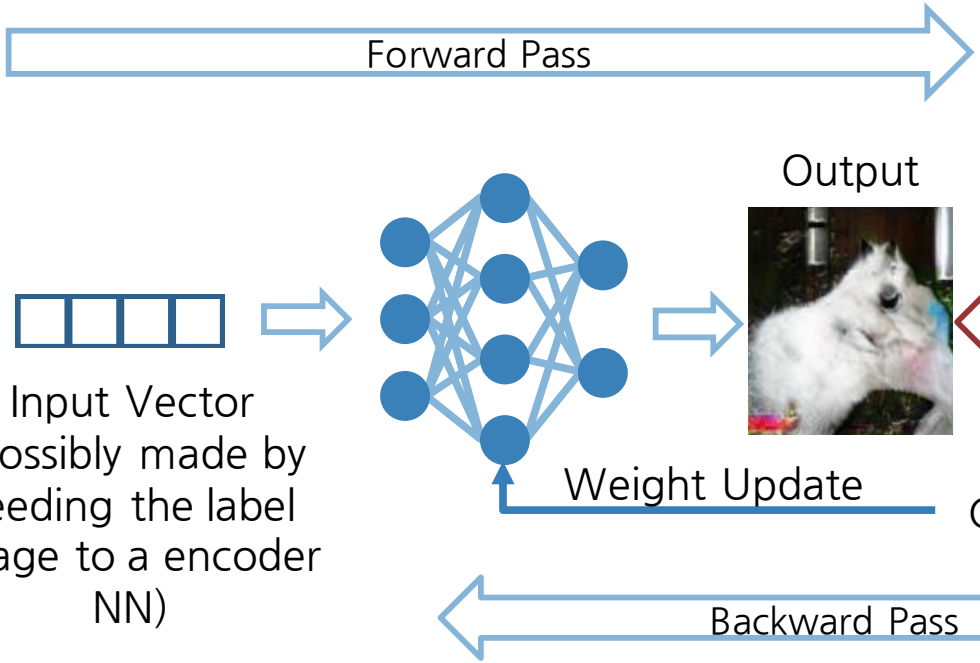
- With loss and gradient



Generating Images

- The previous slide shows an example of a classification model - from image to vector
- Can we do it the opposite way? - from vector to image
- We can try to feed a vector to the model, and give an image that we want to generate as the label
- Minimizing the L1 or L2 distance between output and label images will (hopefully) train out model

Generating Images



Results

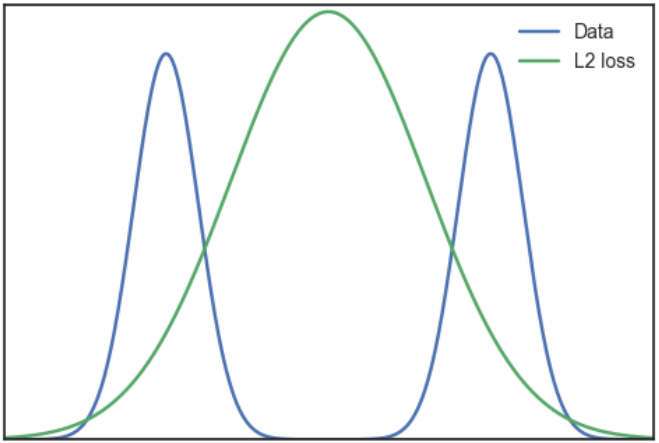
- Tends to generate blurry outputs

Generated

Original

Why Blurry?

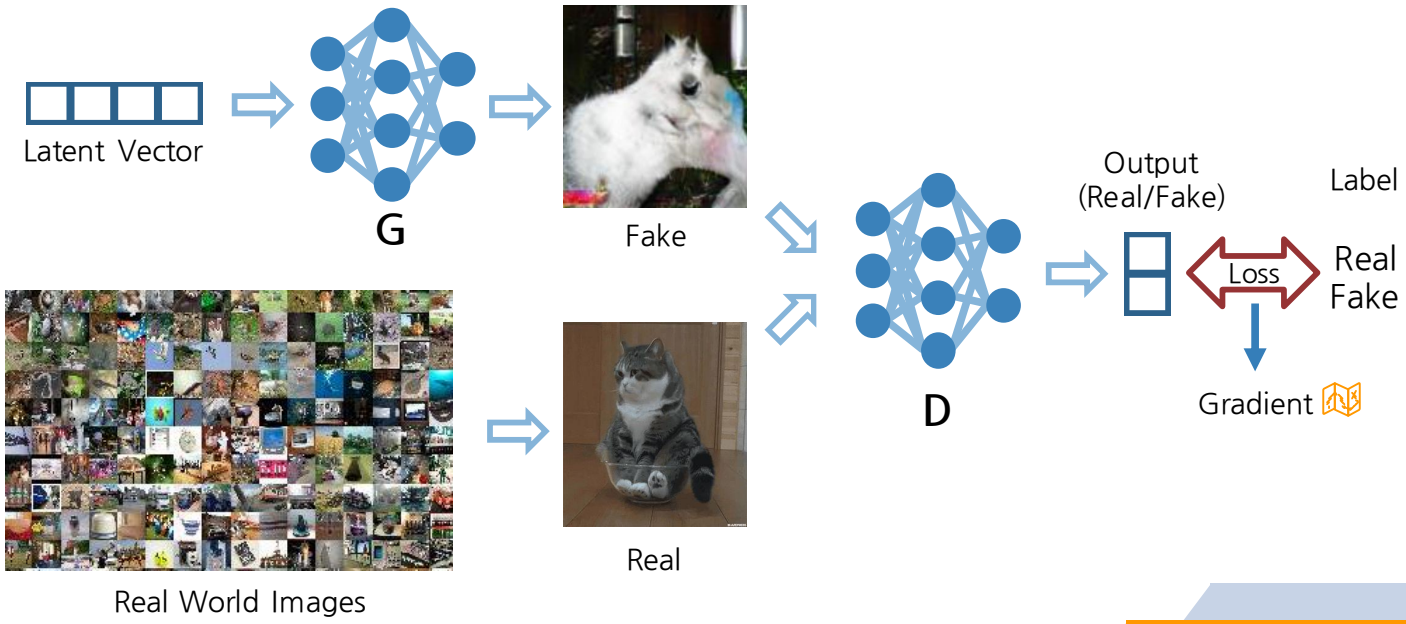
- L2 loss trains the model in a way that it generates an “average” of possible outputs



A Better Loss Function

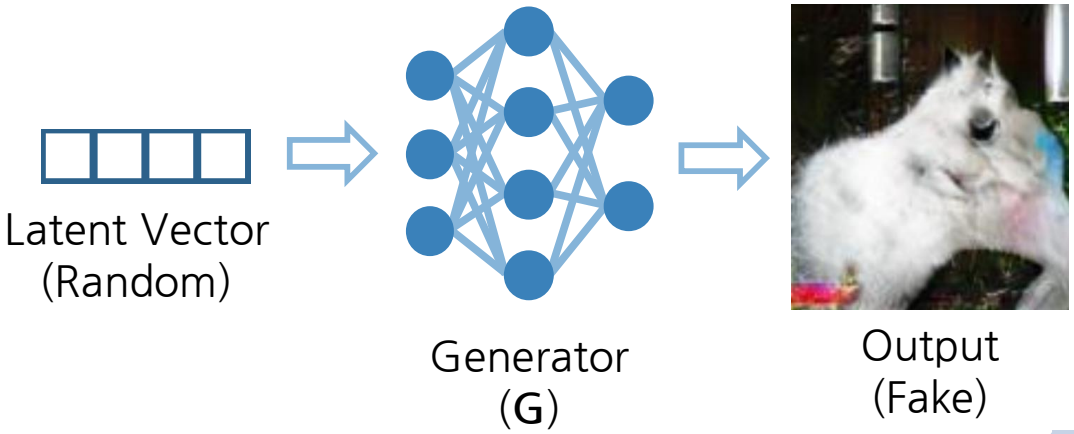
- Can we have a loss function that says “generate images that look like real images”?
- Adversarial loss in Generative Adversarial Network(GAN) does this
- GAN consists of two NNs, Generator and Discriminator
  - Generator: generate fake samples, tries to fool the Discriminator
  - Discriminator: tries to distinguish between real and fake samples

GAN's Architecture



Generator Network

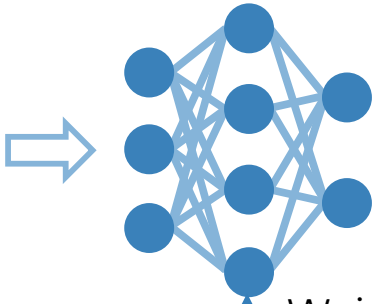
- Generator network generates an output (fake) image from a latent random vector z



Discriminator Network

- Discriminator network takes a real/fake image as input and discriminates it as real/fake

Input
(Real/Fake)



Discriminator
(**D**)

Output
(Real/Fake)

Label

Real
Fake

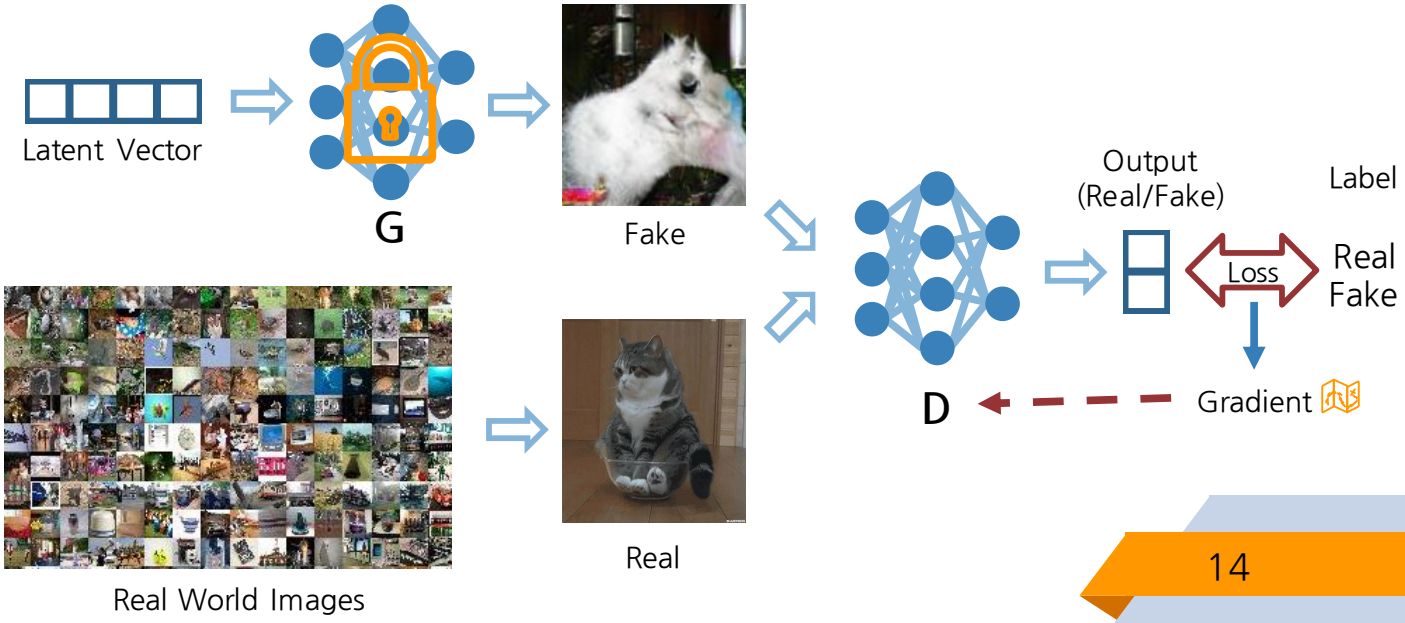
Loss

Weight Update

Gradient

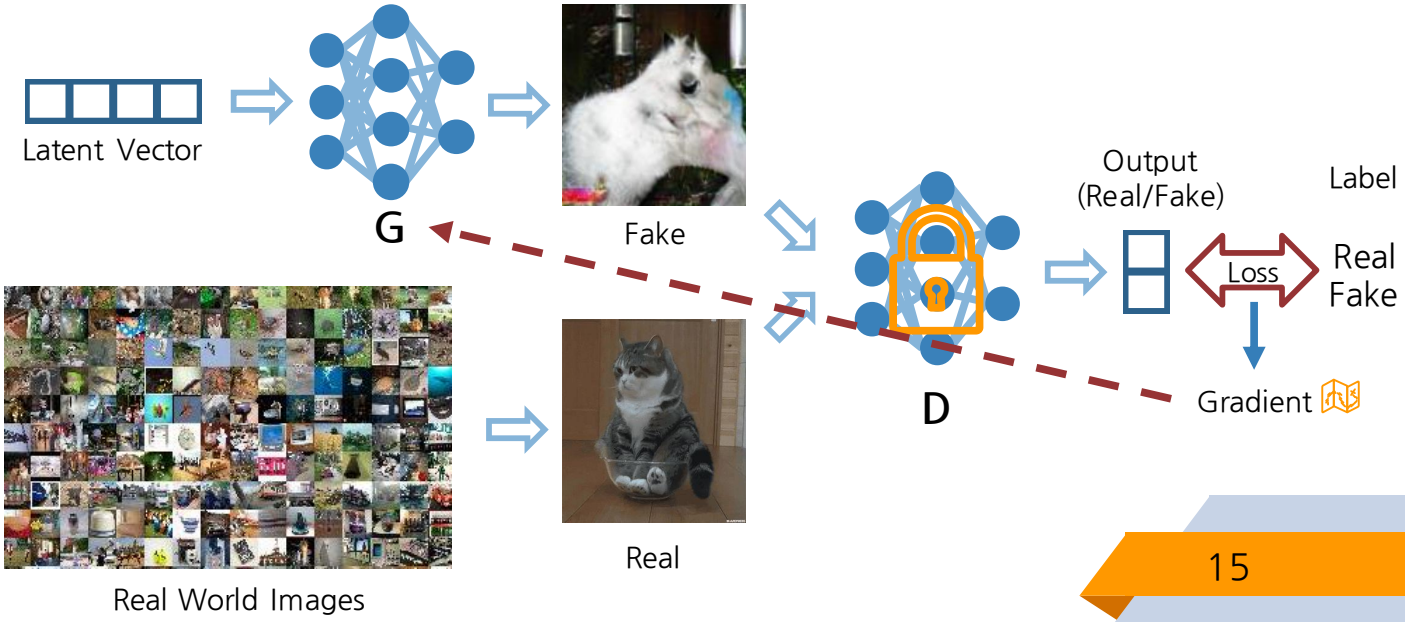
Training D Network

- Update the weights of D to “minimize” the real/fake classification loss



Training G Network

- Update the weights of G to “maximize” the real/fake classification loss



Loss Function and Training Algorithm

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

---

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

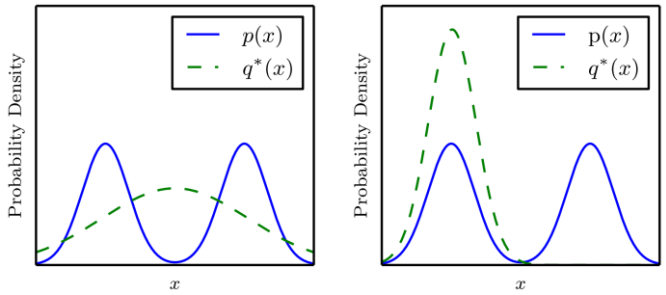
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Advantages of GAN

- Doesn't require labeled data
- Tends to generate sharper outputs
  - Instead of minimizing Kullback-Leibler divergence, it minimizes Jensen-Shannon divergence / Wasserstein distance / …



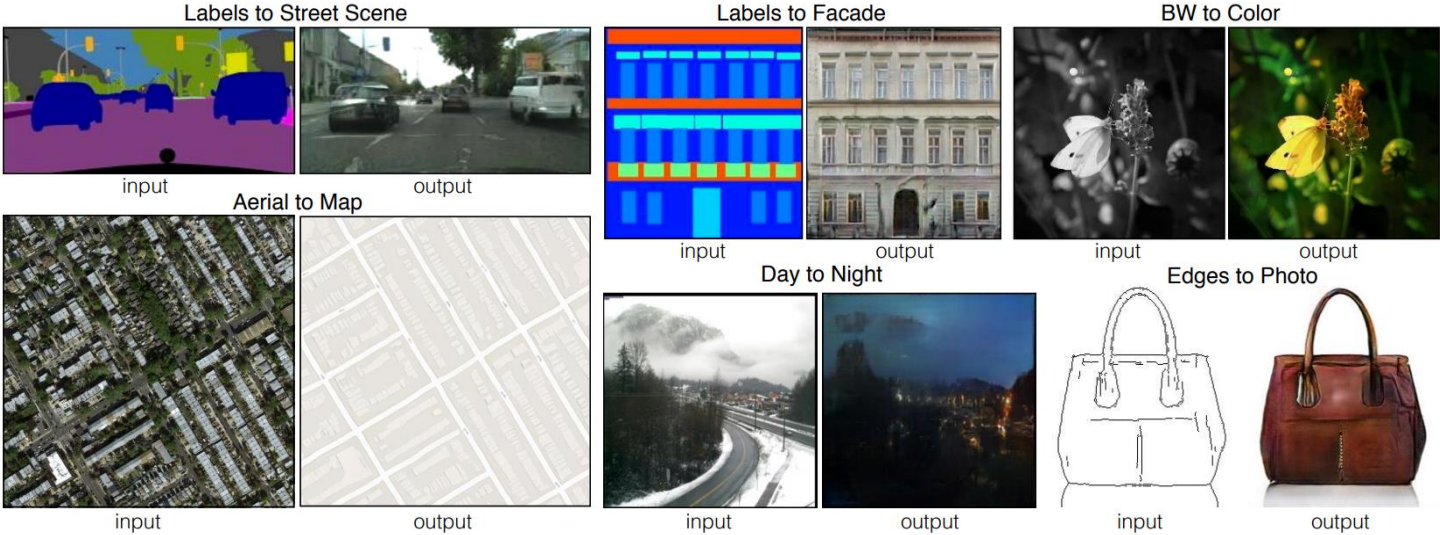
Introduction

Image-to-Image Translation with Conditional Adversarial Networks

Introduction

- Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image
- Conditional GANs are a general-purpose solution that appears to work well on a wide variety of these problems
- Earlier papers have focused on specific applications

Introduction



Contribution

- Demonstrate that on a wide variety of problems, conditional GANs produce reasonable results
- Present a simple framework sufficient to achieve good results
- Analyze the effects of several important architectural choices

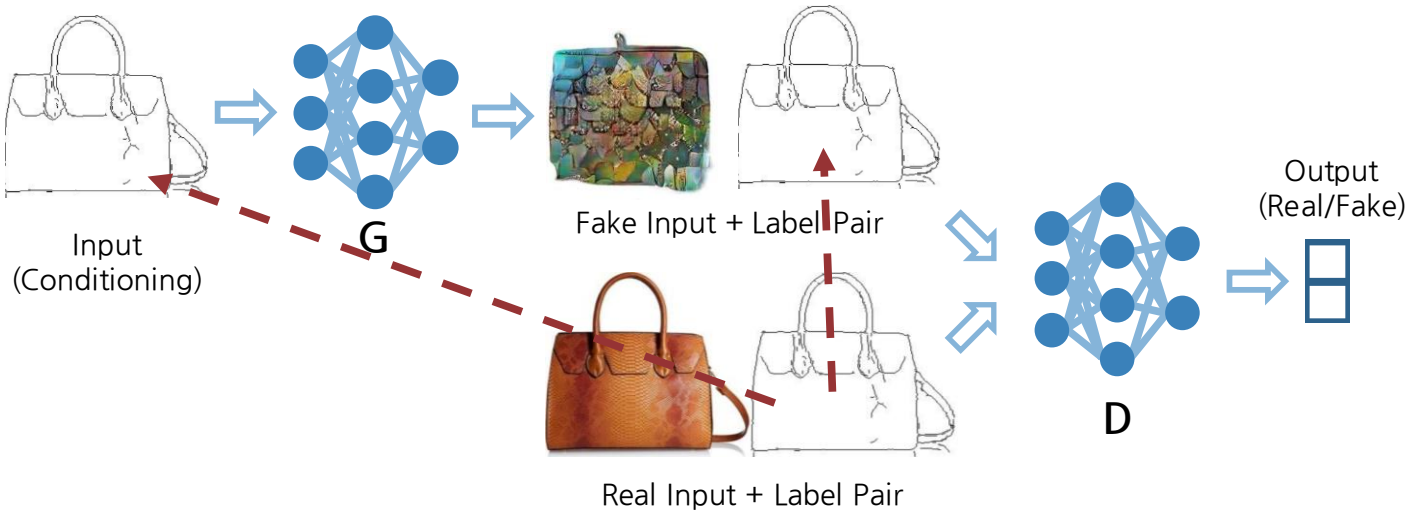
Related Work & Method

Image-to-Image Translation with Conditional Adversarial Networks

Conditional GAN

- With vanilla GAN, we cannot control what to generate
- If we have labeled data, we can condition on this label to control the output ⇨ Conditional GAN
  - Make a pair of input and label, and D network tries to discriminate between real and fake “pair”
  - G network is given a label as input, instead of a random vector

Conditional GAN



Objective Function

- Conditional GANs learn a mapping from observed image x and random noise vector z, to y
  $G: \{x, z\} \rightarrow y$

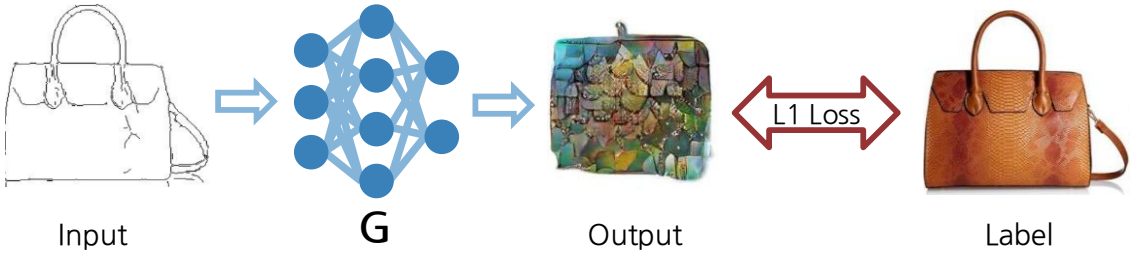
$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[\log(1 - D(x, G(x, z))]$$

- Random noise vector z didn't have any impact on the result - the generator learned to ignore the noise. Therefore, no noise vector is fed to the G network
- Final objective function

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

Objective Function

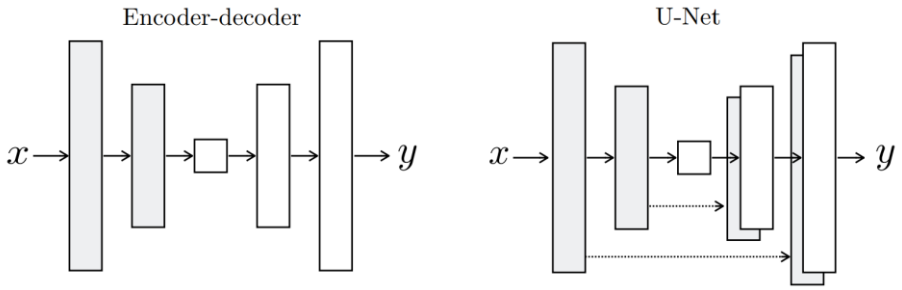
- Previous approaches have found it beneficial to mix the GAN objective with L1/L2 loss
- Either L1 or L2 can be used, but previous studies shows that L1 loss results in less blurry result



$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]$$

Generator with Skips

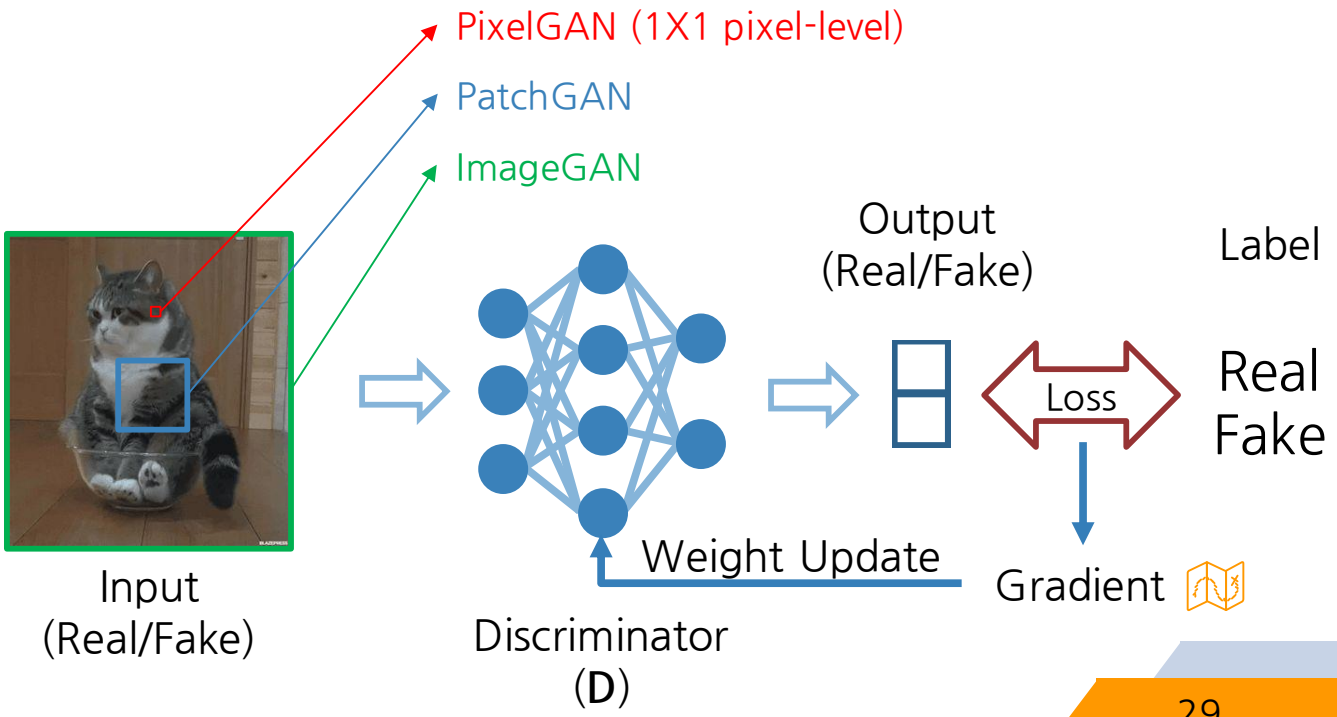
- For many image translation problems, there is a great deal of low-level information shared between the input and output
- To help this information flow, they use a encoder-decoder network with skip connections(depth-wise concatenation) as the G network



PatchGAN

- L1 loss in the objective is good enough to capture low frequencies, so GAN objective doesn't need to be applied at image-level
- PatchGAN discriminator tries to classify if each N × N patch in an image is real or fake. Run this convolutionally across the image, and average all responses
- Advantages: fewer parameters, runs faster, can be applied on arbitrarily large images

PatchGAN


PixelGAN (1X1 pixel-level)
PatchGAN
ImageGAN
Output
(Real/Fake)
Label
Real
Fake
Loss
Weight Update
Gradient
Input
(Real/Fake)
Discriminator
(D)

Optimization and Inference

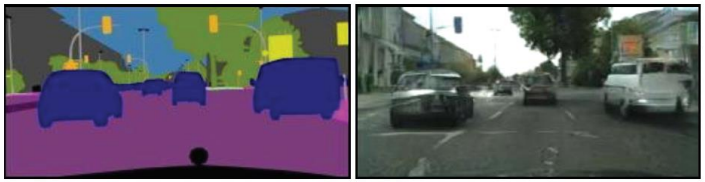
- One G step + one D step
- Divide objective by 2 for D to slow down D training
- Minibatched SGD with Adam optimizer
  - Batch size: 1~10
- Apply dropout on both train and test phase (as a noise)
- Apply batch normalization

Experiments

Image-to-Image Translation with Conditional Adversarial Networks

Generality of Conditional GANs

- Semantic labels ↔ photo

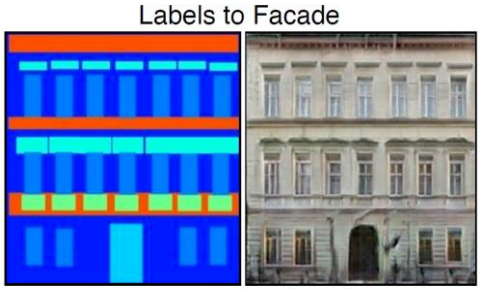


input output

- Map ↔ aerial photo

input output

- Architectural labels → photo



input output

- BW → color photos

input output

- Edges → photo

input output

- Sketch → photo : tests edges → photo

- Thermal → color photos

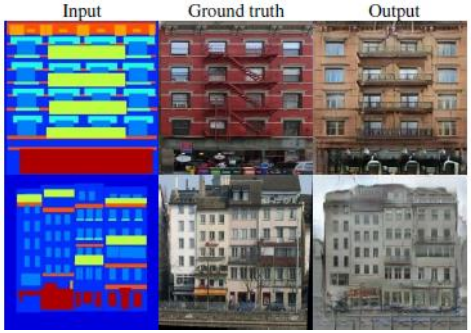
- Day → night

input output

- Photo with missing pixels → inpainted photo

Data Requirements and Speed

- Façade training set
  - 400 images
  - Training : less than 2 hours

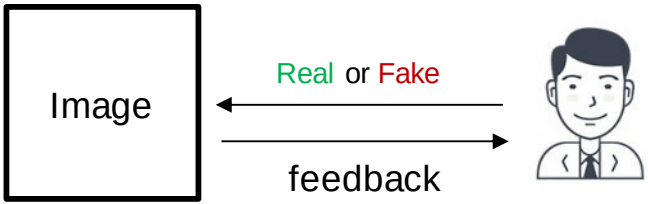


- Day to Night training set
  - 91 unique webcams

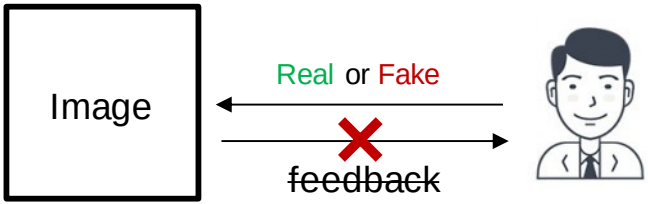
Evaluation Metrics

- Real vs Fake
  - Amazon Mechanical Turk(AMT)
  - A crowdsourcing Internet marketplace enabling individuals and businesses (known as Requesters) to coordinate the use of human intelligence to perform tasks that computers are currently unable to do.

- TEST(10 images)

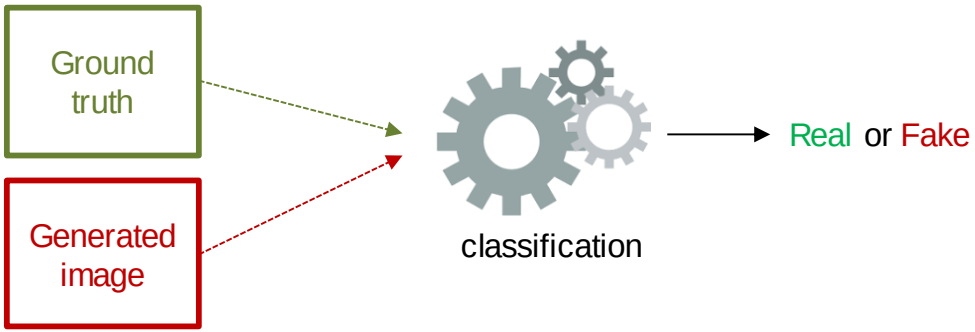


- TEST(40 images)



FCN-score

- If the generated images are realistic, classifiers trained on real images will be able to classify the synthesized image correctly as well

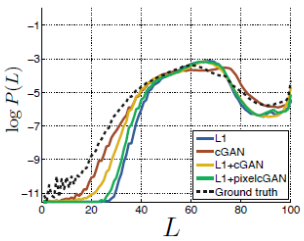


Analysis of the Objective Function

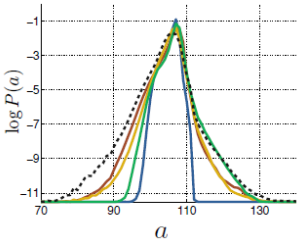
Loss	Per-pixel acc.	Per-class acc.	Class IOU
L1	0.42	0.15	0.11
GAN	0.22	0.05	0.01
cGAN	0.57	0.22	0.16
L1+GAN	0.64	0.20	0.15
L1+cGAN	**0.66**	**0.23**	**0.17**
Ground truth	0.80	0.26	0.21

Table 1: FCN-scores for different losses, evaluated on Cityscapes labels↔photos.

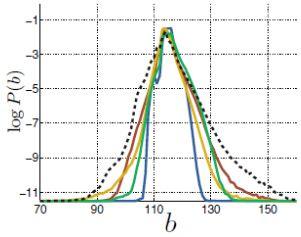
Colorfulness



(a)



(b)

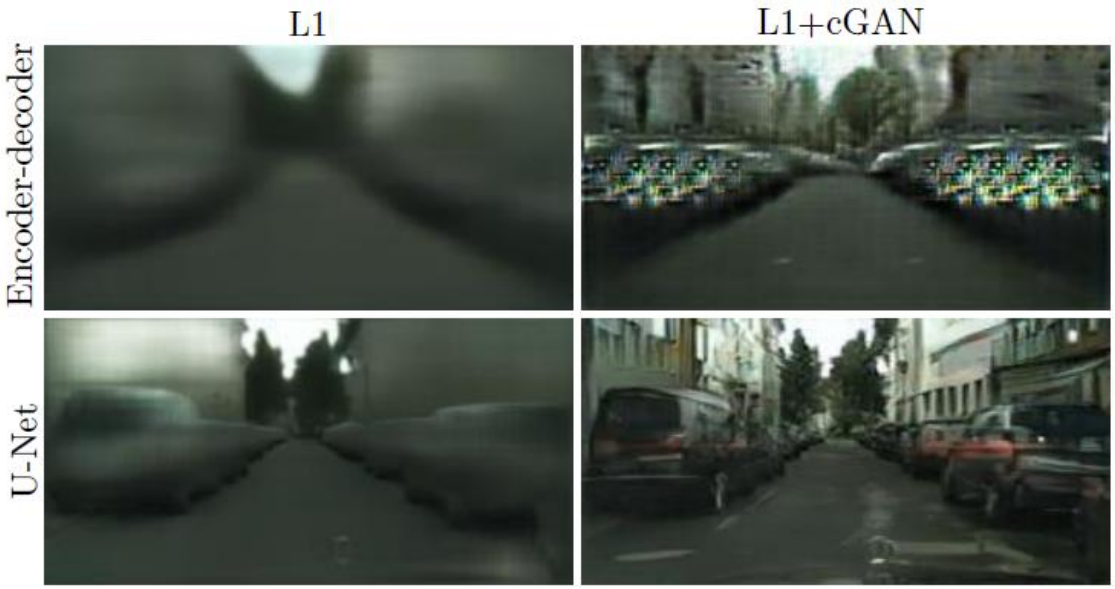


(c)

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

(d)

Analysis of the Generator Architecture



Loss	Per-pixel acc.	Per-class acc.	Class IOU
Encoder-decoder (L1)	0.35	0.12	0.08
Encoder-decoder (L1+cGAN)	0.29	0.09	0.05
U-net (L1)	0.48	0.18	0.13
U-net (L1+cGAN)	**0.55**	**0.20**	**0.14**

From PixelGANs to PatchGANs to Image GANs

- Test the effect of varying patch size N on proposed Discriminator receptive field.
  - Comparison & Analysis of 1x1 PixelGAN to 286x286 ImageGAN.
  - Uncertainty in the output manifests itself differently for different loss functions

Discriminator receptive field	Per-pixel acc.	Per-class acc.	Class IOU
1×1	0.39	0.15	0.10
16×16	0.65	0.21	**0.17**
70×70	**0.66**	**0.23**	**0.17**
286×286	0.42	0.16	0.11

From PixelGANs to PatchGANs to Image GANs

From PixelGANs to PatchGANs to Image GANs

- Test the effect of varying patch size N on proposed Discriminator receptive field.
  - The PixelGAN has no effect on spatial sharpness, but does increase the colorfulness of the results.
  - Using a 16×16 PatchGAN is sufficient to promote sharp outputs, and achieves good FCN-scores, but also leads to tiling artifacts.
  - The 70 × 70 PatchGAN alleviates these artifacts and achieves slightly better similar scores
  - the full 286 × 286 ImageGAN, does not appear to improve the visual quality of the results, and in fact gets a considerably lower FCN-score

Perceptual Validation

- Validate the perceptual realism of our results on the tasks of map ↔ aerial photograph and grayscale ↔ color.

Perceptual Validation

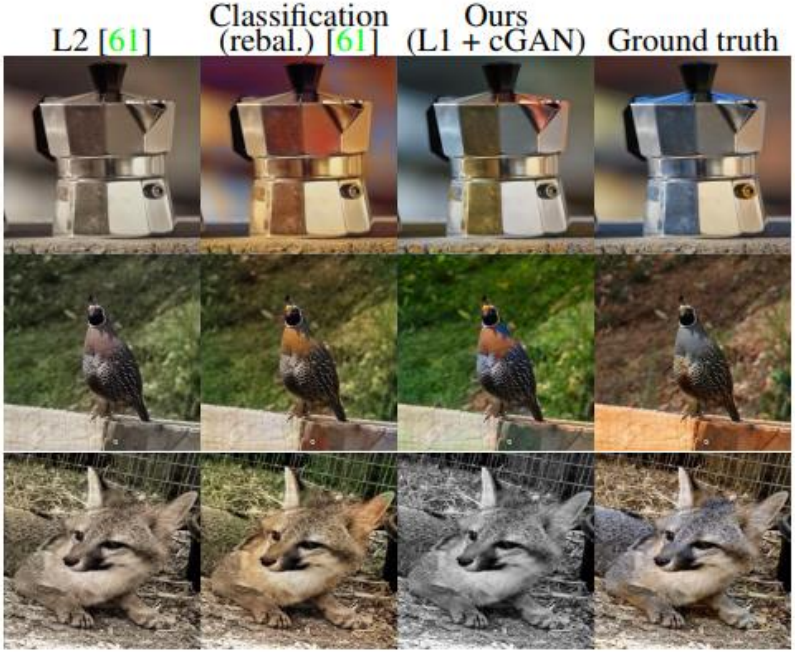
Validate the perceptual realism of our results on the tasks of map ↔ aerial photograph and grayscale ↔ color.

- Map to Aerial
  - The aerial photos generated by the proposed method fooled 18.9% of the participants on trial.
  - The aerial photos generated by the L1 method barely fooled anyone.
- Aerial to Map
  - The maps generated by either methods showed similar results.

	Photo → Map	Map → Photo
Loss	% Turkers labeled *real*	% Turkers labeled *real*
L1	2.8% ± 1.0%	0.8% ± 0.3%
L1+cGAN	6.1% ± 1.3%	**18.9% ± 2.5%**

Perceptual Validation

Validate the perceptual realism of our results on the tasks of map ↔ aerial photograph and grayscale ↔ color.



Perceptual Validation

Validate the perceptual realism of our results on the tasks of map ↔ aerial photograph and grayscale ↔ color.

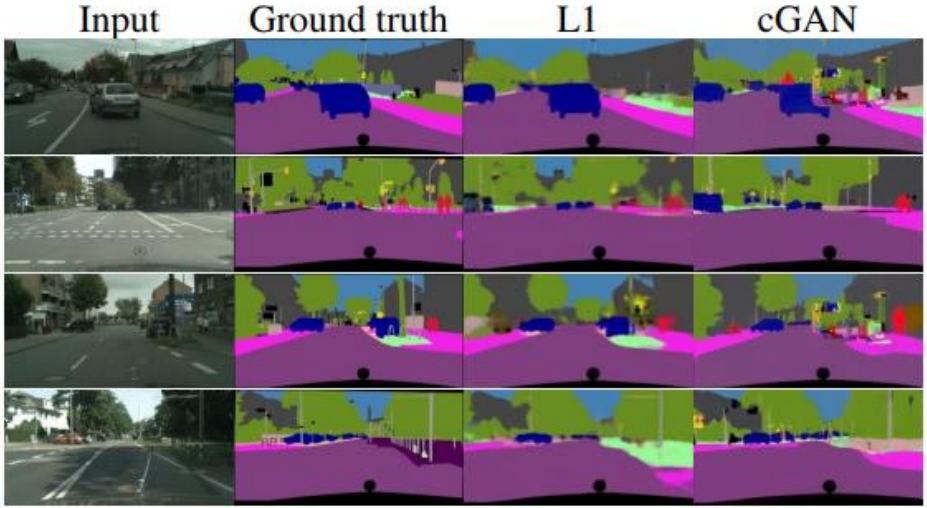
- L2 regression deceived 16.3% of the participants.
- Colorization method deceived 27.8% of the participants.
- Proposed model deceived 22.5% of the participants.

Method	% Turkers labeled *real*
L2 regression from [61]	16.3% ± 2.4%
Zhang et al. 2016 [61]	**27.8% ± 2.7%**
Ours	22.5% ± 1.6%

Semantic Segmentation

Validate the perceptual accuracy of our results on the tasks of photo → labels on cityscapes.

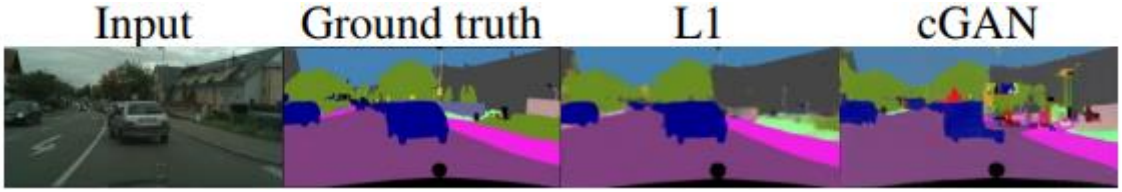
- To test this, we train a cGAN (with/without L1 loss) on cityscape photo → labels.



Semantic Segmentation

Validate the perceptual accuracy of our results on the tasks of photo → labels on cityscapes.

- The cGAN produces sharp images that look at glance like the ground truth, but in fact include many small, hallucinated objects.
- cGANs, trained without the L1 loss, performs with a reasonable degree of accuracy



Loss	Per-pixel 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

Semantic Segmentation

Validate the perceptual accuracy of our results on the tasks of photo → labels on cityscapes.

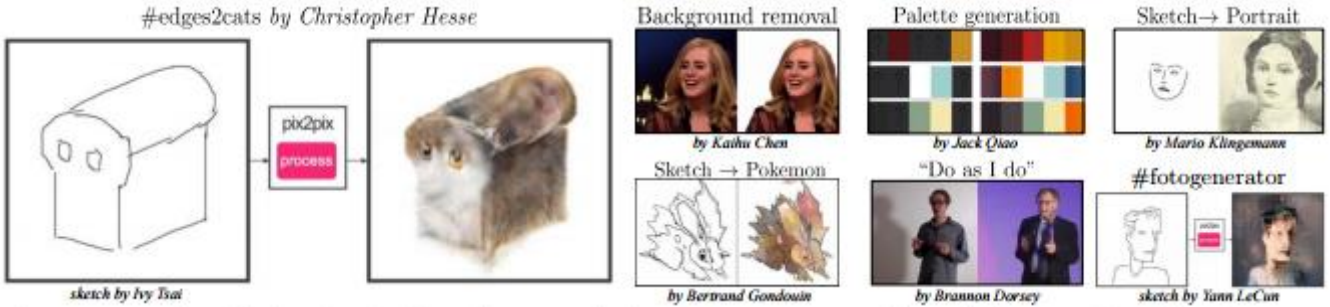
- The first demonstration of GANs successfully generating “labels”.
- Yet, still far from the best available method for solving this problem.
  → Simply using L1 regression gets better scores than using a cGAN.

Loss	Per-pixel 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

Community Driven Research

Since the initial release of the paper's model, the Twitter community have successfully applied our framework to a variety of novel image-to-image translation tasks.

- This includes Background removal, Palette generation, Sketch → Portrait, Sketch → Pokemon, "Do as I Do" pose transfer and etc.



Conclusion

- Conditional adversarial networks are a promising approach for many image-to-image translation tasks, especially those involving highly structured graphical outputs
- These networks learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.

THANKS!

Any questions?