Image-to-Image Translation with Conditional Adversarial Networks

Phillip Isola Jun-Yan Zhu Tinghui Zhou Alexei A. Efros

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

CVPR July 21-26 HONOLULU 2017

Image-to-image translation: Overview

➔ transforms an input image into a corresponding output image while **preserving its key structure.**

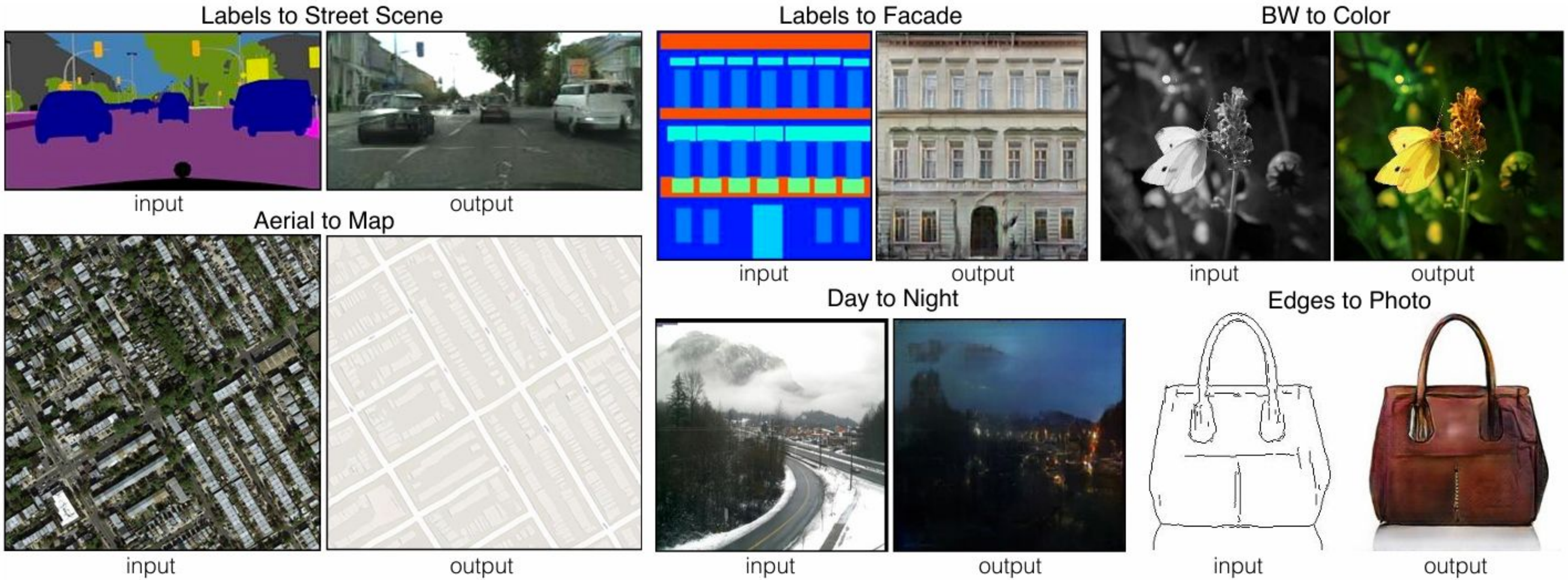


Image source: [Image-to-image translation use cases]

GANs: Review

generative models that learn to **map random noise vector to required image** which is **similar to the sample from the trained distribution**.

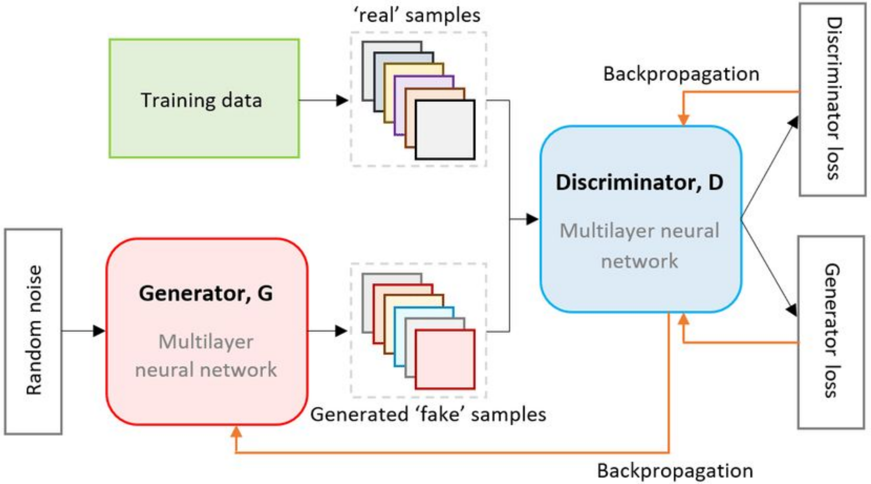


Image source: [GAN architecture]

7 Conclusions and future work

This framework admits many straightforward extensions:

1. A *conditional generative* model $p(\boldsymbol{x} \mid \boldsymbol{c})$ can be obtained by adding $\boldsymbol{c}$ as input to both G and D.
2. *Learned approximate inference* can be performed by training an auxiliary network to predict $\boldsymbol{z}$ given $\boldsymbol{x}$. This is similar to the inference net trained by the wake-sleep algorithm [15] but with the advantage that the inference net may be trained for a fixed generator net after the generator net has finished training.

Image source: [Future work of GANs]

Research gap:

- ➔ Existing architectures have been **developed for each task separately,** although they have **same setting: predict pixels from pixels**.
- ➔ The effectiveness of image-conditional GANs as a **unified solution** for image-to-image translation **remains unclear**.

Solution: Image-to-Image Translation with Conditional GAN

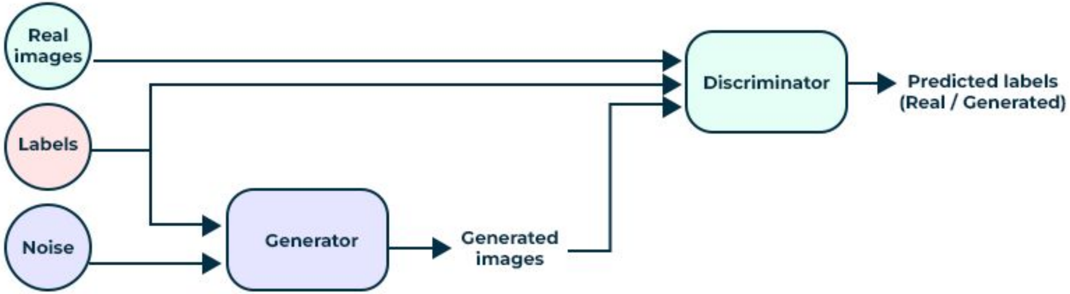


Image source: [Architecture of cGANs]

High level architecture:

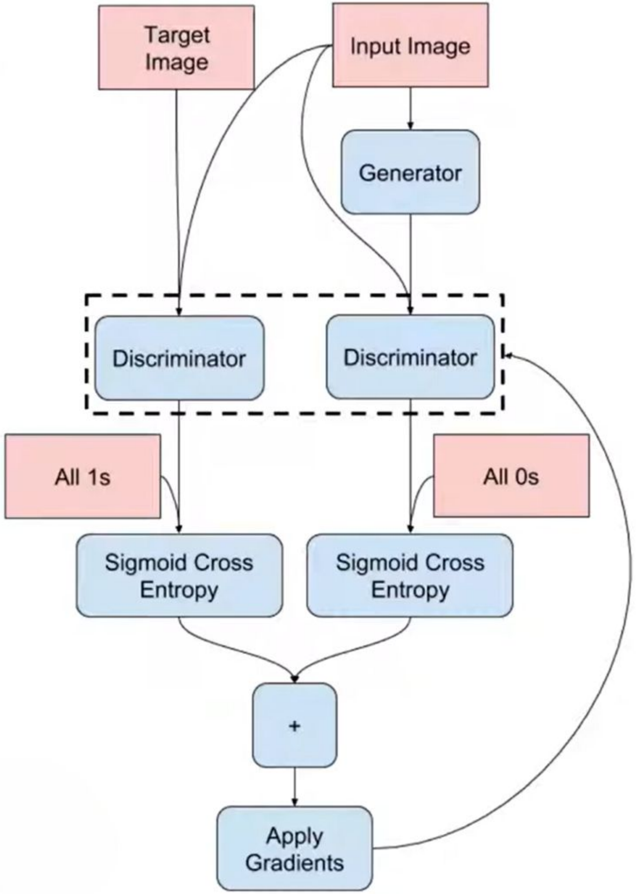


Image source: [Architecture of proposed cGAN]

Conditional GAN architecture: Generator

- ➔ Previous encoder-decoder architecture is replaced with encoder-decoder architecture with skip connection: **"U-Net"**

- ➔ Why do we need **skip connections** for image-to-image translation ?
  - To address this image-to-image translation problem, **low level informations** (features from shallow layers) should be shared from input output for generate the required output. `

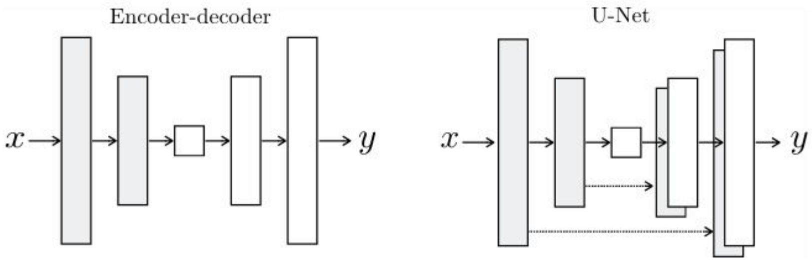


Image source: [Encoder-decoder vs U-Net]

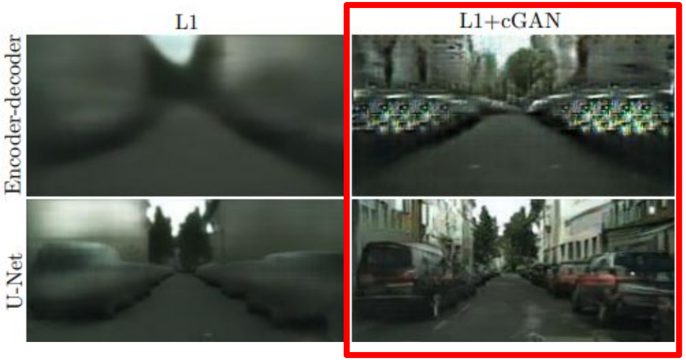


Image source: [Results comparison between encoder-decoder and U-Net]

Conditional GAN architecture: Discriminator

➔ Introduced **Markovian discriminator (PatchGAN)** as learnable loss function.

➔ Penalize the loss to **preserve high frequency features** and able to produce **high resolution outputs**.

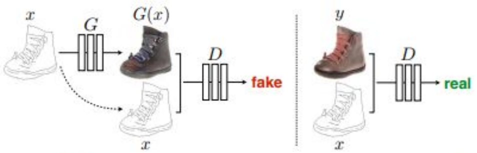


Image source: Process of discriminator

Image source: Results of patchGAN discriminator with different patch sizes.

Discriminator: PatchGAN discriminator

- ➔ Classify NxN patch (receptive area) into real or fake classes.
- ➔ Convolve across the image and then average the all response to obtain final output.
- ➔ Computationally effective for high resolution images and able to capture **low level features (high frequency features)**.

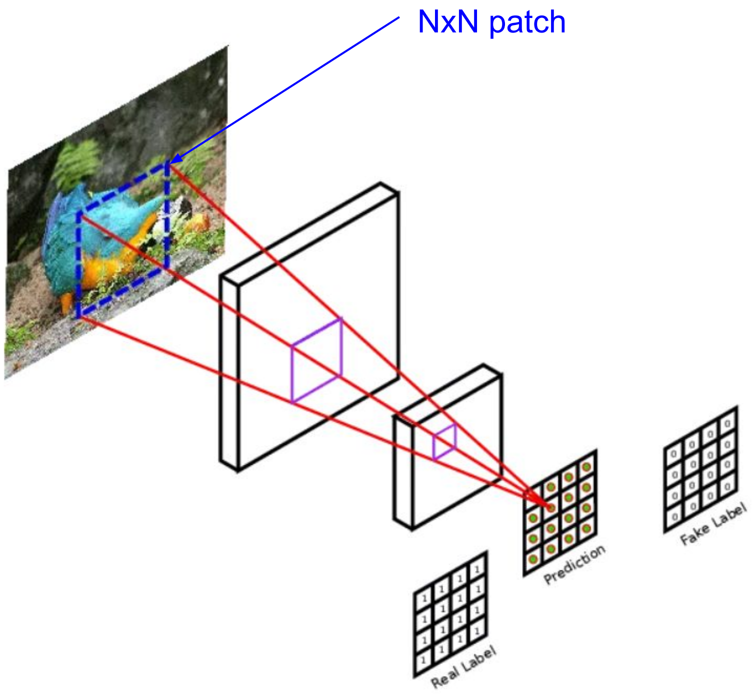


Image source: PatchGAN mechanism

Conditional GAN architecture: Loss functions

Objective function of a conditional GAN; L1 loss for capturing low frequency features;

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

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

Final objective function;

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

➔ Optimizer : **ADAM** with a **learning rate** of **0.0002**, and **momentum** parameters **β1= 0.5, β2 = 0.999**

Effect of different loss functions:

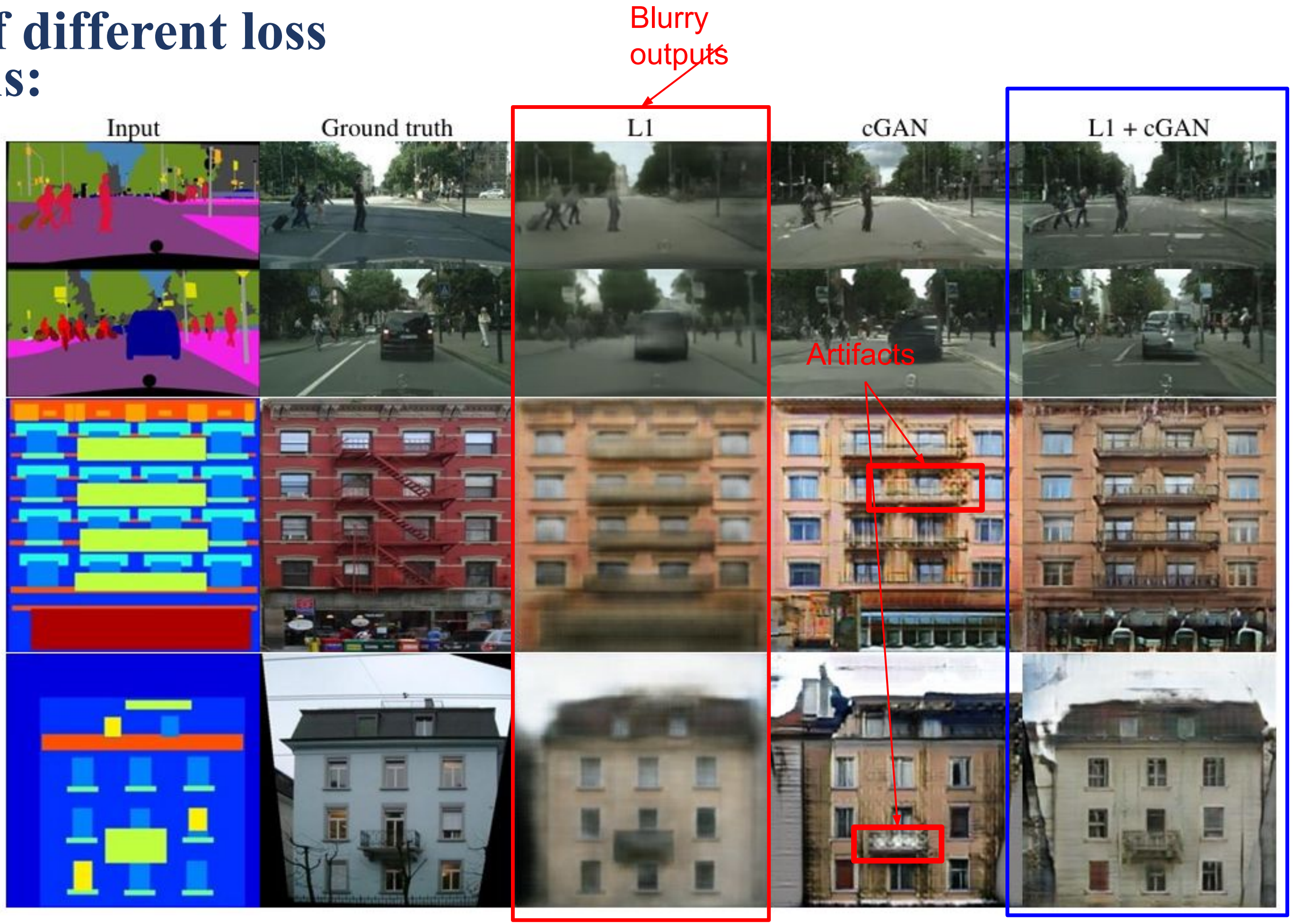


Image source: [Effect of each loss function for cGANs]()

Applications:

- ❑ Semantic labels ⟺ photo
- ❑ Architectural labels ⟹ photo
- ❑ Map ⟺ aerial photo
- ❑ Edges ⟹ photo
- ❑ Sketch ⟹ photo
- ❑ Day ⟹ night
- ❑ Image inpainting

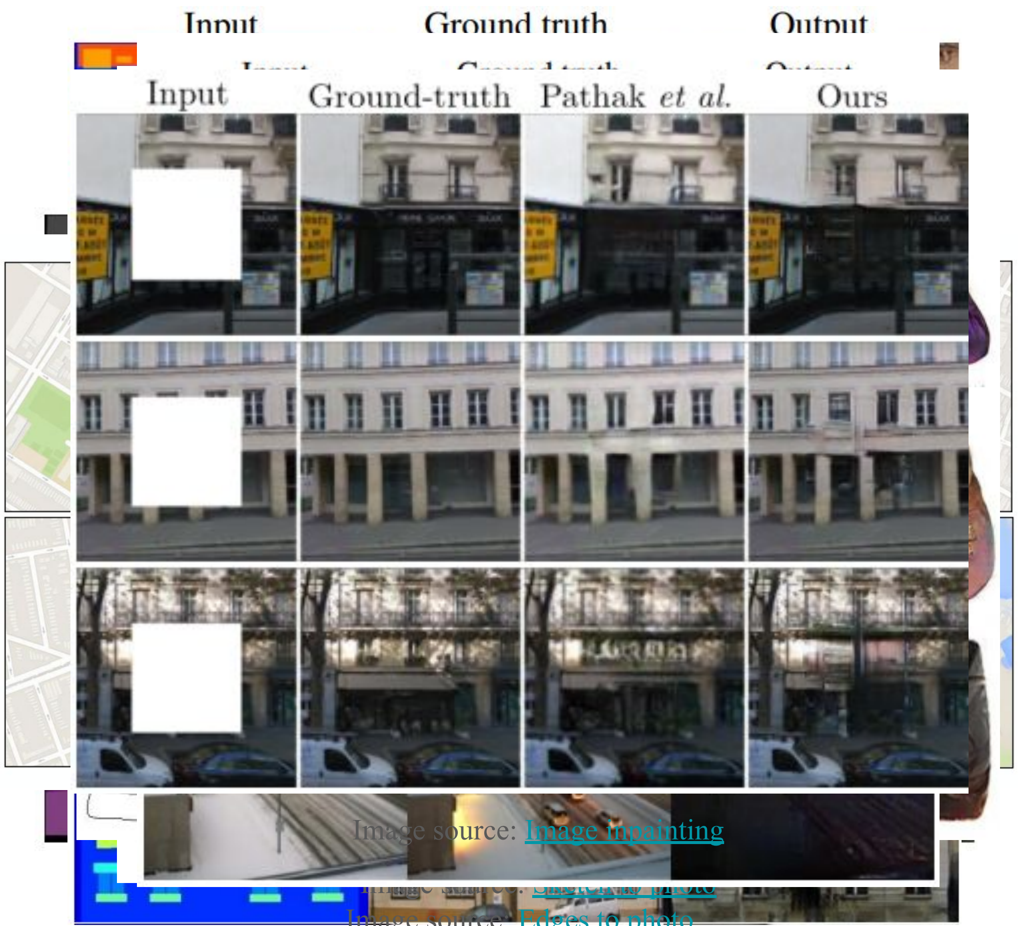


Image source: Image inpainting

Image source: Edges to photo

Image source: Day to night

Image source: Architecture label to photo

Evaluation metrics: Perceptual evaluation on Amazon Mechanical Turk (AMT)

- ➔ Evaluate outputs by using **human perception** for image **colorization and photo generation.**

- ➔ Turkers try to predict whether the given image is **fake or real** for each trial.

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

Table 4: AMT "real vs fake" test on maps↔aerial photos.

Method	% Turkers labeled *real*
L2 regression from [62]	16.3% ± 2.4%
Zhang et al. 2016 [62]	**27.8% ± 2.7%**
Ours	22.5% ± 1.6%

Table 5: AMT "real vs fake" test on colorization.

Image source: [Results of AMT based method]

Evaluation metrics: FCN-score

➔ Quantitative analysis of the cGAN by using **FCN-8s** model trained on **"cityscapes"** dataset.

➔ The idea is that if the generated images appear realistic, classifiers trained on real images should also be **able to accurately classify the synthesized images**.

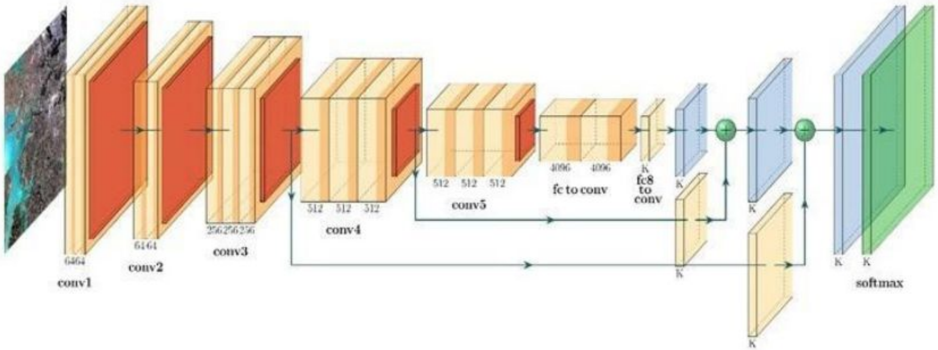


Image source: [Architecture of FCN-8 model]

Evaluation metrics: FCN-score ctd.

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.

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**

Table 2: FCN-scores for different generator architectures (and objectives), evaluated on Cityscapes labels↔photos. (U-net (L1-cGAN) scores differ from those reported in other tables since batch size was 10 for this experiment and 1 for other tables, and random variation between training runs.)

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

Table 3: FCN-scores for different receptive field sizes of the discriminator, evaluated on Cityscapes labels→photos. Note that input images are 256×256 pixels and larger receptive fields are padded with zeros.

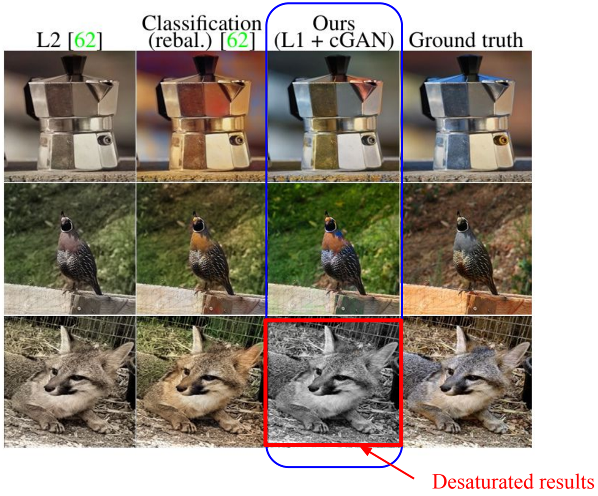
Image source: [Results of FCN-score based method]

Limitations:

- ➔ In image colorization, sometimes, the model produces **grayscale or desaturated results.**

- ➔ **Hallucinated objects** might appear in generated images.

- ➔ Inconsistent Output for Complex Images.



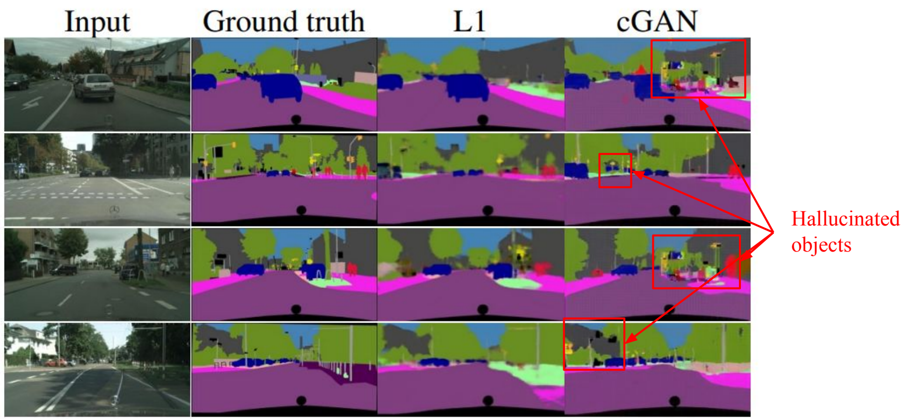


Image source: [Limitations of cGAN]()

Community contributions:

➔ Other contribution from research community;
- Background removal
- Palette generation
- Sketch → Portrait
- Sketch→Pokemon
- "Do as I Do" pose transfer

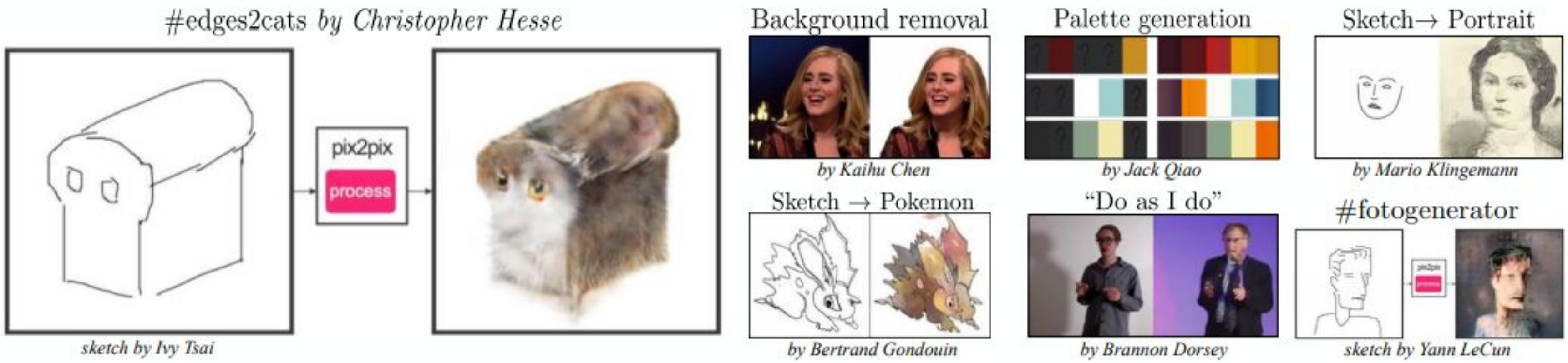


Image source: Other research contribution based on this paper


dont worry about it if you don't
understand

Let’s dive into the code!

Thank you