pix2pix: A general purpose solution for image-to-image learning using cGANs

Thomas Jacob [1] Arif Ahmad [2] Ishan Kapnadak [3] Sanidhya Anand [4]

[1]Roll no- 190070068

[2]Roll no- 190110010

[3]Roll no- 190070028

[4]Roll no- 19d170027

April 2022

Note:

The ideas and concepts discussed in this presentation heavily draw insights, from the paper cited below:

Isola, Phillip, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. "Image-to-image translation with conditional adversarial networks." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125-1134. 2017.

Outline

Introduction

Some common image-to-image translation problems . . .

Figure: Semantic labels $\Longleftrightarrow$ photo, trained on the Cityscapes dataset.

Introduction

Some common image-to-image translation problems ...

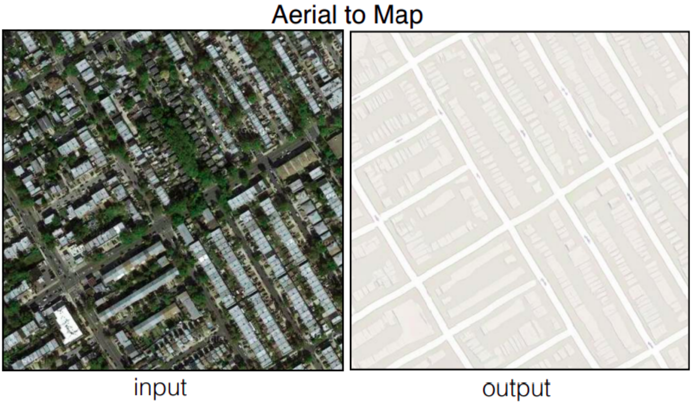

Figure: Map $\Longleftrightarrow$ aerial photo, trained on data scraped from Google Maps.

Introduction

Some common image-to-image translation problems . . .

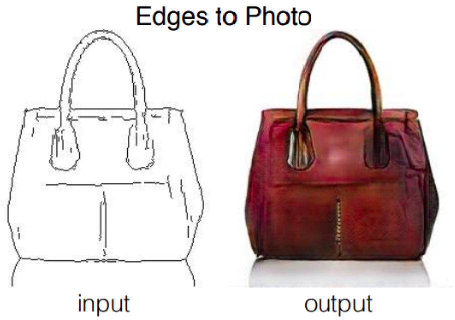

Figure: Sketch $\longrightarrow$ photo: tests edges to photo models on human-drawn sketches

Introduction

Some common image-to-image translation problems ...

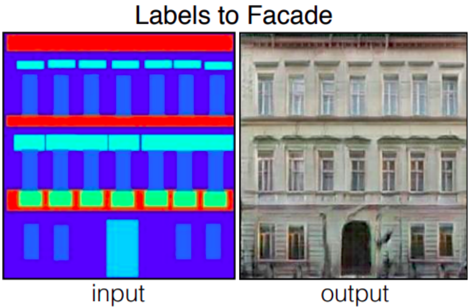

Figure: Architectural labels $\longrightarrow$ photo, trained on CMP Facades

Introduction

Some common image-to-image translation problems . . .

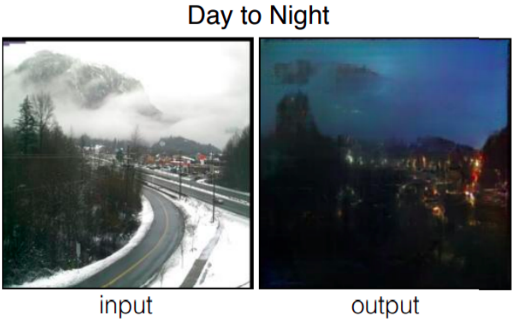

Figure: Day $\longrightarrow$ night image mappings

Introduction

Some common image-to-image translation problems . . .

While all these tasks appeared very different. There is something fundamentally same in all these tasks.

Introduction

Some common image-to-image translation problems . . .

While all these tasks appeared very different. There is something fundamentally same in all these tasks.

So we ask the question:

"What is common in all such image-to-image translation problems?"

Introduction

Some common image-to-image translation problems . . .

While all these tasks appeared very different. There is something fundamentally same in all these tasks.

So we ask the question:

"What is common in all such image-to-image translation problems?"

Introduction

Some common image-to-image translation problems ...

Answer:

All automatic image-to-image translation tasks can be seen as translating one possible representation of a scene into another, given sufficient training data.

Introduction
Problem Statement

Problem

Train a model given enough data to generate images (**Y**) given an input image (**X**) such that the image **Y** is another representation of **X** following a certain rule or constraint. The model should be as general as possible and applicable to any image-to-image translation task without change of architecture or loss function.

Introduction
Problem Statement

Problem

Train a model given enough data to generate images (**Y**) given an input image (**X**) such that the image **Y** is another representation of **X** following a certain rule or constraint. The model should be as general as possible and applicable to any image-to-image translation task without change of architecture or loss function.

Presented Solution to the problem:
Using a conditional Generative Adversarial Network (**cGAN**):

`pix2pix` model

Previous work

- CNNs have become a staple in machine computer vision and image processing over the past decade.

Previous work

- CNNs have become a staple in machine computer vision and image processing over the past decade.
- Problems are still treated as an optimization problem where our goal is to reduce a loss function

Previous work

- CNNs have become a staple in machine computer vision and image processing over the past decade.
- Problems are still treated as an optimization problem where our goal is to reduce a loss function

Given a specific image processing or computer vision problem, a lot of time is spent in hand-crafting an appropriate loss function minimizing which will give us the desired output.

Previous work

Structured Loss

- Many previous works treat the output space as unstructured in the sense that the output pixel is considered conditionally independent from all others given the input image.

Previous work

Structured Loss

- Many previous works treat the output space as unstructured in the sense that the output pixel is considered conditionally independent from all others given the input image.
- Conditional GANs however learn a structured loss, i.e. the losses penalize the joint configuration of the output.

Previous work

Structured Loss

- Many previous works treat the output space as unstructured in the sense that the output pixel is considered conditionally independent from all others given the input image.
- Conditional GANs however learn a structured loss, i.e. the losses penalize the joint configuration of the output.

Using structured losses has been explored earlier commonly in methods using conditional random fields (CRFs).

Previous work

Conditional GANs

- Prior works on GANs have well explored the conditional GAN setting in a variety of problems using texts and images.

Previous work

Conditional GANs

- Prior works on GANs have well explored the conditional GAN setting in a variety of problems using texts and images.
- Other papers have also used GANs for image-to-image mappings, but only applied the GAN unconditionally.

Previous work

Conditional GANs

- Prior works on GANs have well explored the conditional GAN setting in a variety of problems using texts and images.
- Other papers have also used GANs for image-to-image mappings, but only applied the GAN unconditionally.

Conditional GANs train on a labeled data set and let you specify the label for each generated instance. For example, an unconditional MNIST GAN would produce random digits, while a conditional MNIST GAN would let you specify which digit the GAN should generate.

pix2pix architecture

Math and Notations

GANs:

Learn a mapping from random noise vector z to output image y, $G : z \mapsto y$.

cGANs:

Learn a mapping from observed image x and random noise vector z, to y, $G : \{x, z\} \mapsto y$.

pix2pix architecture

Math and Notations

The generator G is trained to produce outputs that cannot be distinguished from "real" images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator's "fakes".

Objective Function

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

Objective Function

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

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1] \tag{2}$$

Objective Function

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

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[||y - G(x, z)||_1] \tag{2}$$

our objective function is:

$$G^* = \arg\min_G \max_D \mathcal{L}_{cGAN}(G, D) + \lambda \mathcal{L}_{L1}(G) \tag{3}$$

Here λ decides relative weights between cGAN loss and L1 loss. In our code we have used $\lambda = 100$.

Generator Network Architecture: U-Net

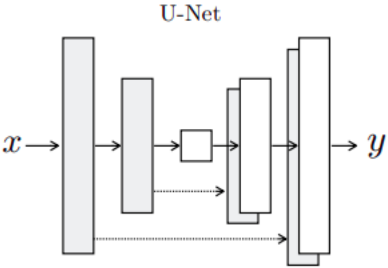

Figure: U-Net encoder-decoder architecture with skip connections [1].

Discriminator Network Architecture: PatchGAN

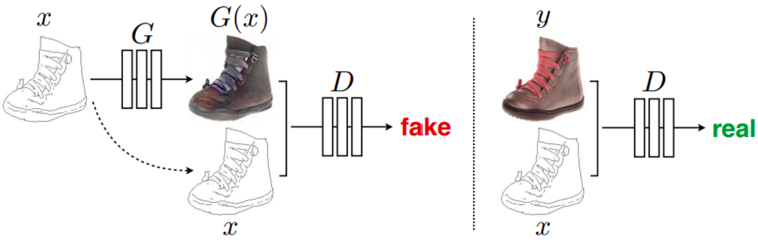

Figure: The discriminator, D, learns to classify between fake (synthesized by the generator) and real edge, photo tuples

Discriminator Network Architecture: PatchGAN

We restrict the GAN discriminator to capture high frequency information. Hence, the design of *Patch*GAN that only penalizes structure at the scale of patches. *Patch*GAN tries to classify if each $N \times N$ patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D.

Results
Training

Method Alternating between one gradient descent step on D, then one step on G

Results

Training

Method Alternating between one gradient descent step on D, then one step on G

Dataset facades dataset

Results

Training

Method Alternating between one gradient descent step on D, then one step on G

Dataset facades dataset

Training time 2 hours for appreciable results
4 hours for visually appealing results

Results

Snapshot in time which shows how our model prediction improves with increasing training epochs on one particular image from facades dataset:

Results

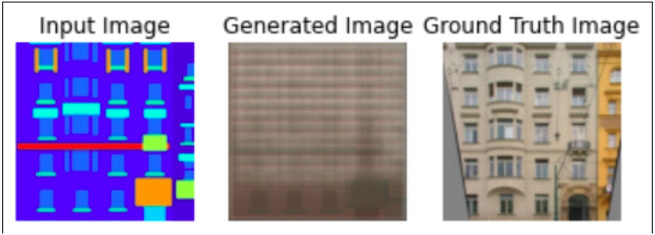

Figure: Generated image by network before training

Results

Figure: Generated image by network at epoch 30

Results

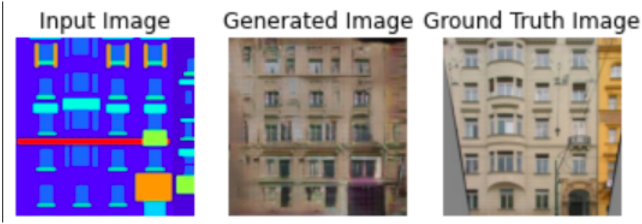

Figure: Generated image by network at epoch 60

Results

Figure: Generated image by network at epoch 90

Results

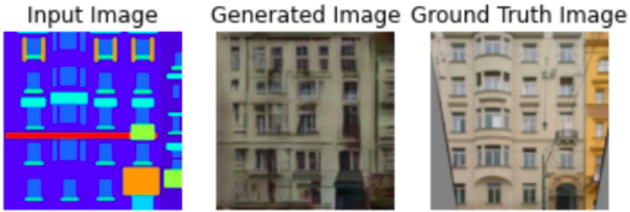

Figure: Generated image by network at epoch 120

Results

Figure: Generated image by network at epoch 150

Results

Next we see how our model makes predictions during test time on unseen images from facades dataset:

Results

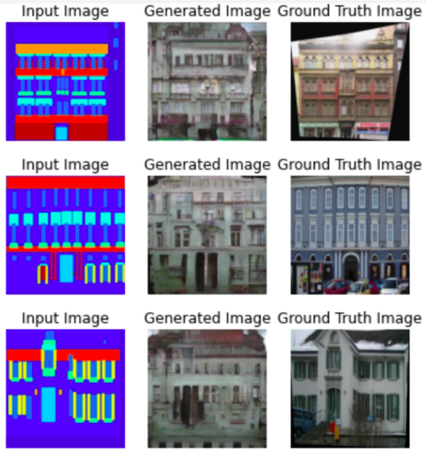

Figure: Generated image by network during testing

Results

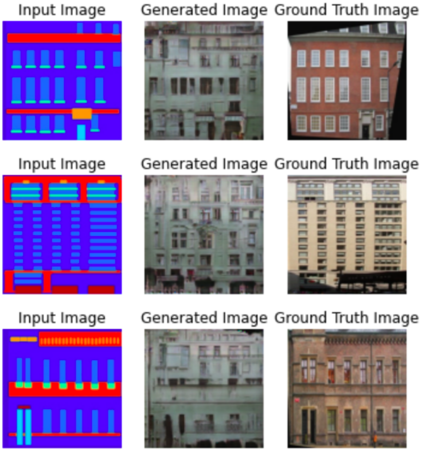

Figure: Generated image by network during testing

Conclusions

1. Conditional adversarial networks are a promising approach for many image-to-image translation tasks, especially those involving highly structured graphical outputs.
2. With pix2pix we can learn a loss adapted to the task and data at hand, which makes them applicable in a wide variety of settings.
3. Producing stochastic outputs can capture the full entropy of the conditional distributions they model, however this could not be achieved by pix2pix and is left as an open problem