Image-to-Image Translation with Conditional Adversarial Networks

Deep learning final project

Members:
MISHRA Rudresh
RODRIGUEZ Ricardo

Date:
December 13th, 2019

Lecturer:
Francois ROUSSEAU

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

TABLE OF CONTENTS

I. Introduction

II. Problem Statement

III. Formulation

IV. Architecture

V. Implementation and Results

VI. Conclusions

IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

I. Introduction

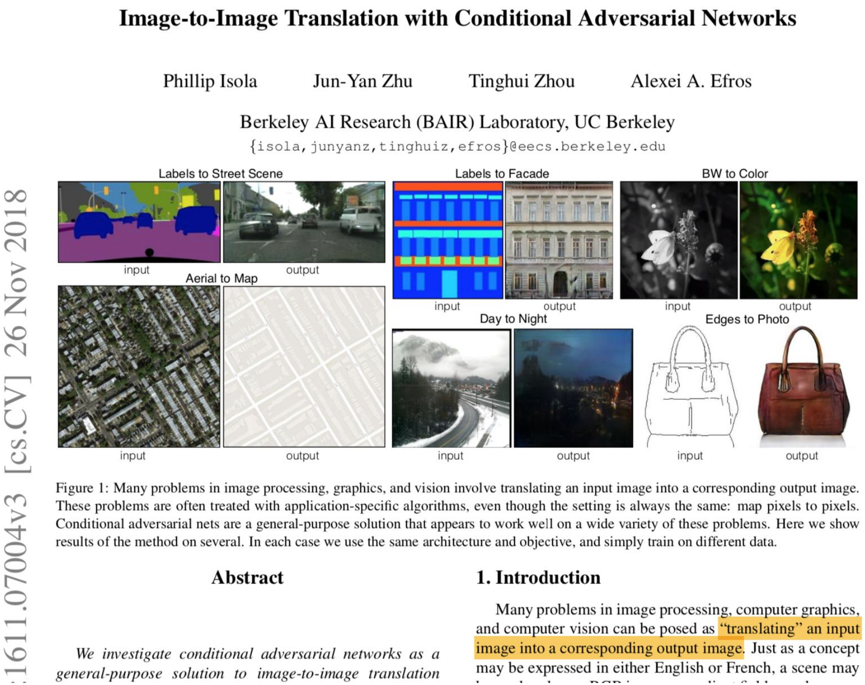


Image-to-Image Translation with Conditional Adversarial Networks

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

Berkeley AI Research (BAIR) Laboratory, UC Berkeley

{isola,junyanz,tinghuiz,efros}@eecs.berkeley.edu

Figure 1: Many problems in image processing, graphics, and vision involve translating an input image into a corresponding output image. These problems are often treated with application-specific algorithms, even though the setting is always the same: map pixels to pixels. Conditional adversarial nets are a general-purpose solution that appears to work well on a wide variety of these problems. Here we show results of the method on several. In each case we use the same architecture and objective, and simply train on different data.

Abstract

We investigate conditional adversarial networks as a general-purpose solution to image-to-image translation

1. Introduction

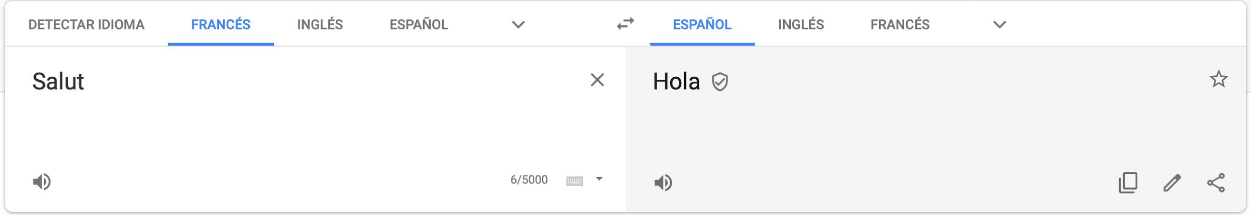
Many problems in image processing, computer graphics, and computer vision can be posed as "translating" an input image into a corresponding output image. Just as a concept may be expressed in either English or French, a scene may

[1] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

II. Problem Statement

The problem of Image to image translation

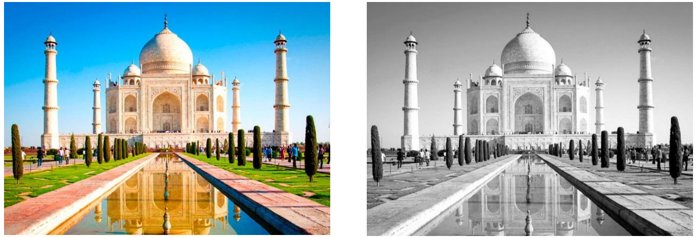
- Similarly to text translation



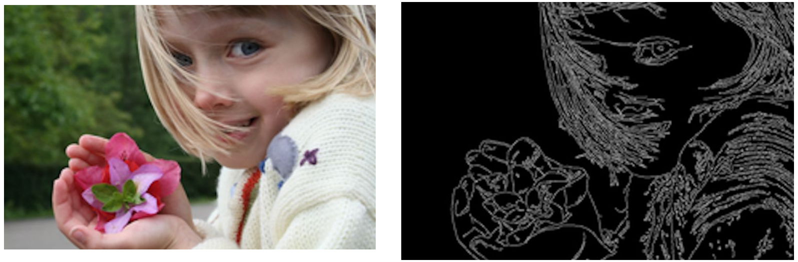
Salut

Hola

- Image to Image translation can be understood as different representations of scenes



RGB Image ↔ GRAY Image

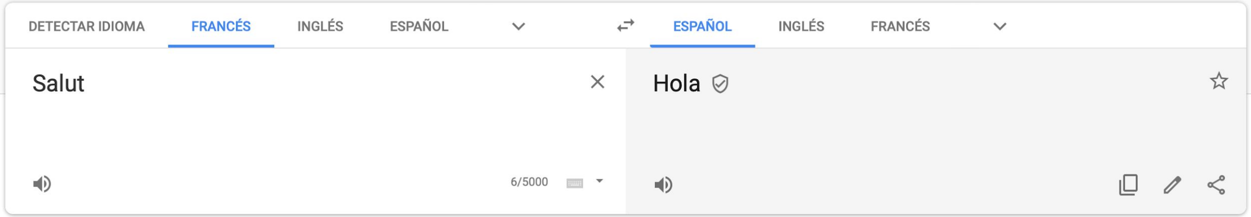


RGB Image ↔ Edges Image

II. Problem Statement

The problem of Image to image translation

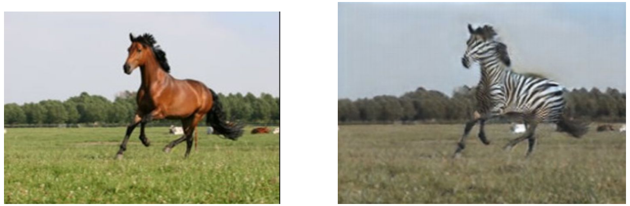
- Similarly to text translation



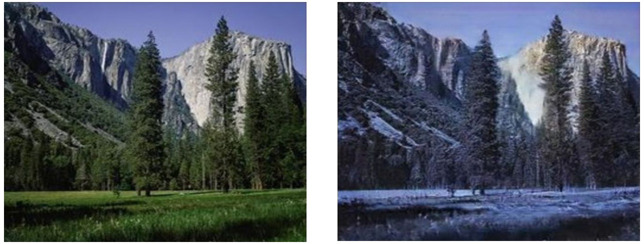
Salut

Hola

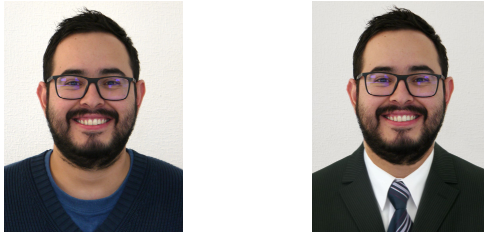
- Image to Image translation can be understood as different representations of scenes



Horse ↔ Zebra



Summer Image ↔ Winter Image



Regular Outfit ↔ Elegant Outfit

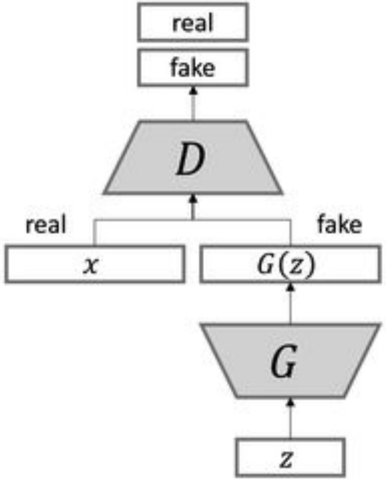
IMT Atlantique
Bretagne-Pays de la Loire
École Mines-Télécom

III. Formulation

3.1 Generative Adversarial Model (GAN)

- Two Neural Networks:
  - **Generator (G)** : Creates Synthetic Images from noise pixels that the discriminator cannot distinguish.

  - **Discriminator (D)**: Identifies whether the image is real or synthetic
- Both networks are trained simultaneously and in a competitive way



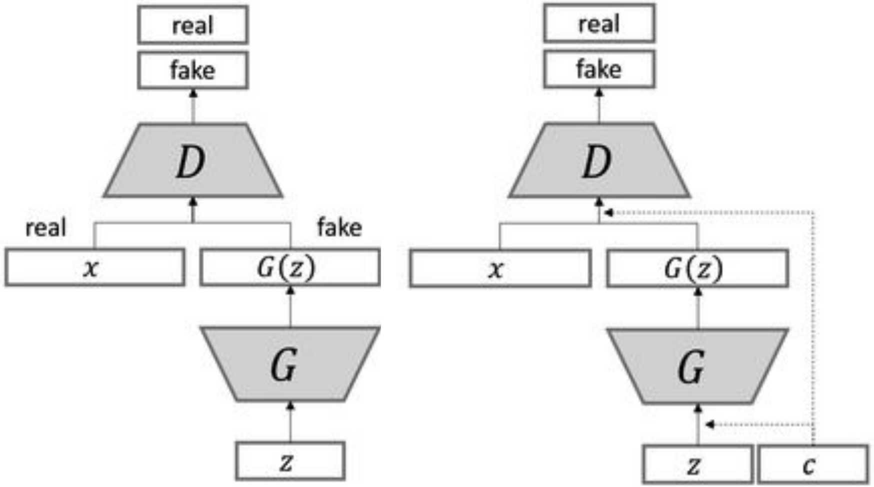
GAN

Fig. 1 GAN Architecture

III. Formulation

3.2 Conditional GAN (CGAN)

- Conditional generation with GANs entails using labeled data to generate images based on a certain class.
- The discriminator and the generator are conditioned on c (target data).
- The input and c are combined in a joint hidden representation and fed as an additional input layer in both network.
- In CGAN (**Conditional GAN**), labels act as an extension to the latent space ***z*** to generate and discriminate images better.



GAN

Conditional GAN

Fig. 2 GAN and **CGAN** Architectures

III. Formulation

3.2 Conditional GAN (CGAN)

- Optimization problem in a conventional GAN:

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

True data

Noise provided for generating data

- Optimization problem in a conditional GAN:

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

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x,z)\|_1]. \qquad \mathcal{L}_{L2}(G) = \mathbb{E}_{x,y,z}\left(y - G(x,z)\right)^2$$

Images taken from [3]

IV. Architecture

4.1 Generator architecture

- It has an Encoder-decoder architecture using a U-Net architecture.
- The model takes a source image and generates a target image
- It does this by first downsampling or encoding the input image down to a bottleneck layer, then upsampling or decoding the bottleneck representation to the size of the output image

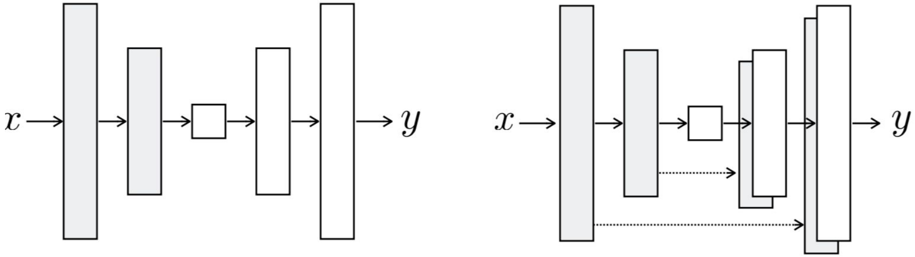


Fig. 3 Encoder-decoder and U-Net

IV. Architecture

4.1 Generator architecture

- It has an Encoder-decoder architecture using a U-Net architecture.
- The model takes a source image and generates a target image
- It does this by first downsampling or encoding the input image down to a bottleneck layer, then upsampling or decoding the bottleneck representation to the size of the output image

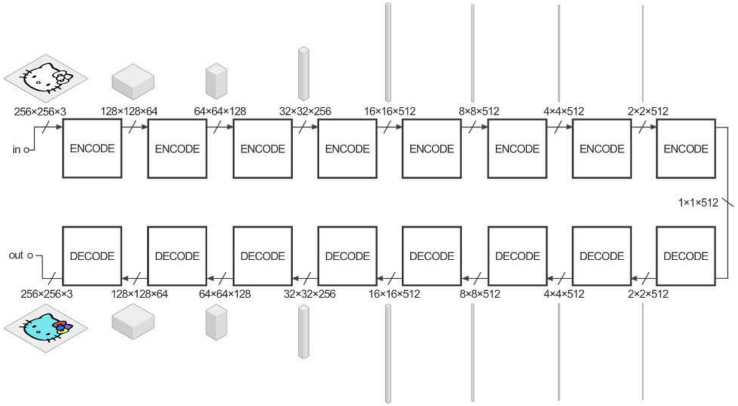


Fig. 4 Representation of the generator model. Taken from [2]

IV. Architecture

4.2 Discriminator architecture

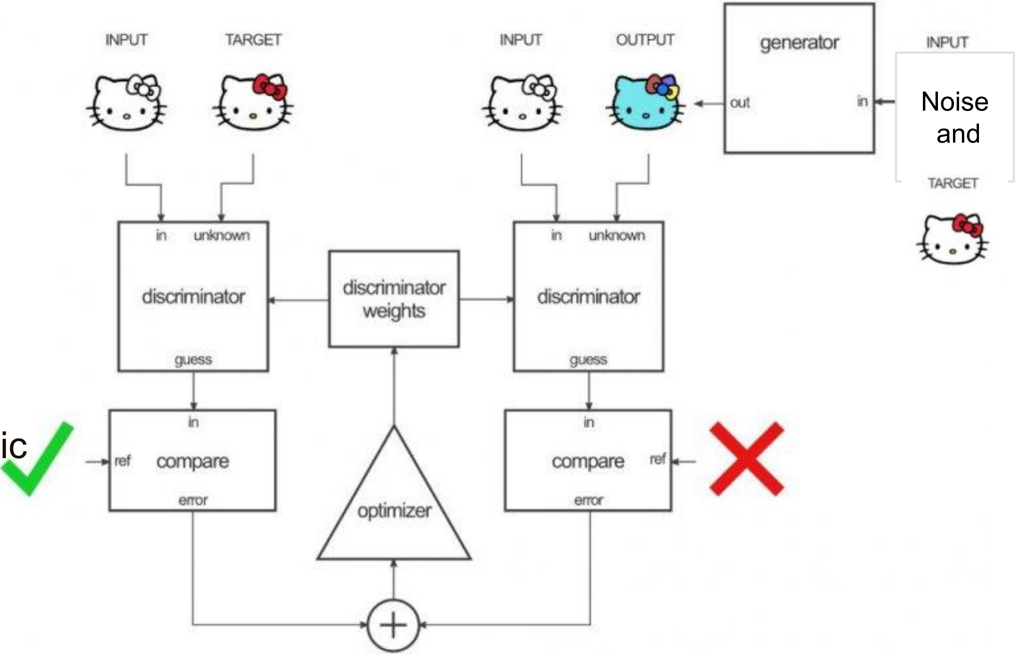
- The model takes two input images that are concatenated together and predicts a patch output of predictions
- The model is optimized using binary cross entropy, and a weighting is used so that updates to the model have half (0.5) the usual effect

Fig. 5 PatchGAN

IV. Architecture

4.3 Training

- Two steps have to be taken: training the discriminator and training the generator.
- All networks were trained from scratch. Weights were initialized from a Gaussian distribution with mean 0 and standard deviation of 0.02.
- First, the generator generates an output image. The discriminator looks at the input/target pair and the input/output pair and produces its guess about how realistic they look.
- The weights vector of the discriminator is then adjusted based on the classification error of the input/output pair and the input/target pair.



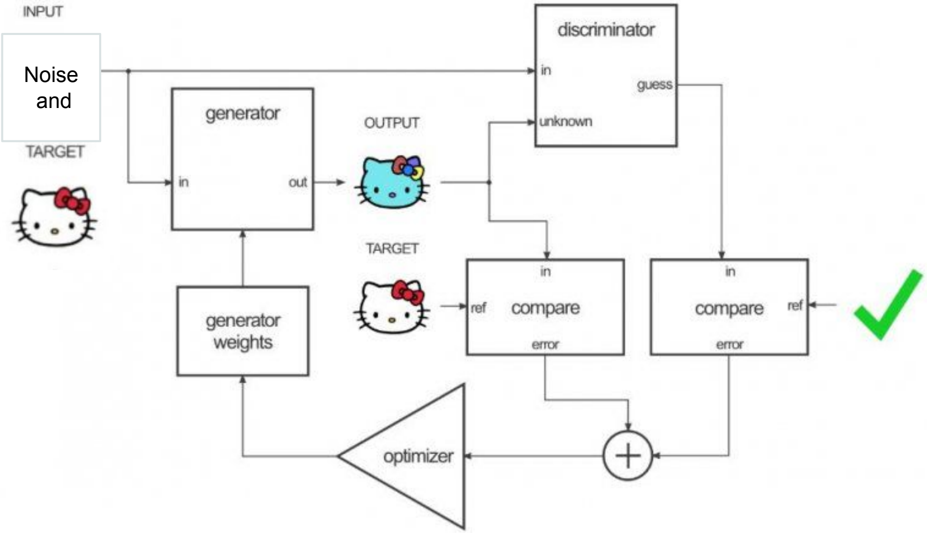
Images taken from [2]

IV. Architecture

4.4 Training

- The generator’s weights are then adjusted based on the output of the discriminator as well as the difference between the output and the target image.

- While the discriminator improves, we’re training the generator to beat the discriminator. If the discriminator is good at its job and the generator is capable of learning the correct mapping function through gradient descent, we should get generated outputs that could fool a human.

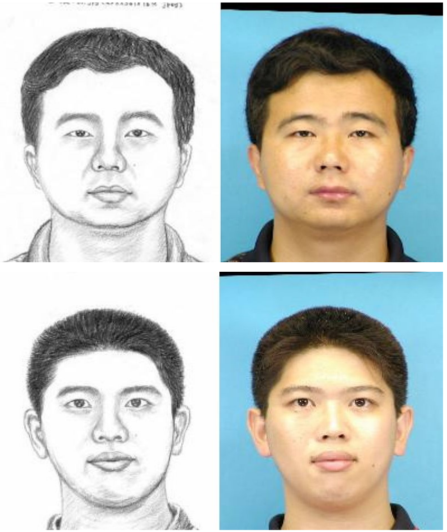


Images taken from [2]

V. Implementation and Results

5.1 Implementation

- Objective: Sketch Image ↔ Real Image
- Dataset [3]:
  - Train: 88 paired Images
  - Test: 100 paired Images (To see the output)
- Epochs: 300
- Loss Optimizer: GAN Loss, Lambda Loss

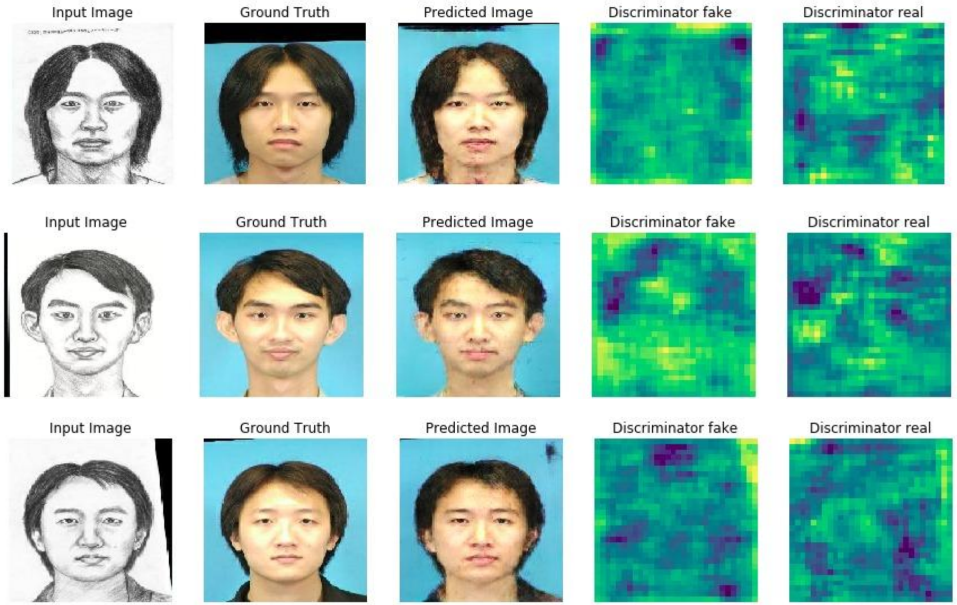


Input: Real Sketch Image and Target Image

V. Implementation and Results

5.2 Results

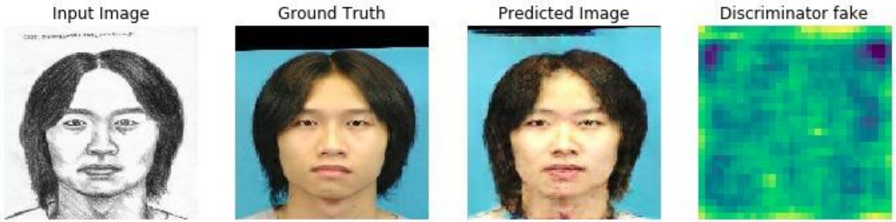
- Visual results of the training:
  - **Sketch -> Real Face**



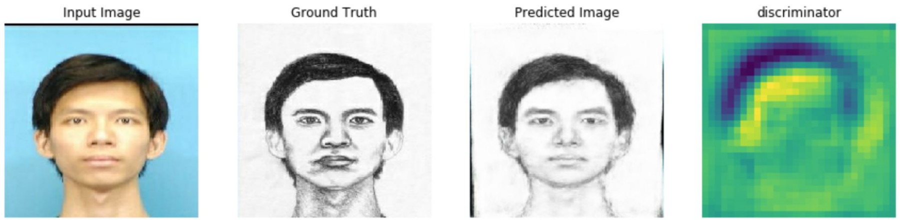
V. Implementation and Results

5.2 Results

- Visual results of the training:
    - **Sketch -> Real Face**



-
    - **Real Face -> Sketch**



- Refer to the notebook for more results...

VI. Conclusions

- This project suggests that 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.
- Some improvements can be done using more epochs during the training and also improving on the quality of the dataset.
- More applications:
  - Image Resolution Improvement
  - Spectrogram translation which allow creation of sounds
  - Background Removal

References

[1] Isola, Phillip, et al. "Image-to-image translation with conditional adversarial networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.
[2] https://neurohive.io/en/popular-networks/pix2pix-image-to-image-translation/
[3] https://www.slideshare.net/Artifacia/generative-adversarial-networks-and-their-applications
[4] http://mmlab.ie.cuhk.edu.hk/archive/sketchdatabase/CUHK/