Evaluating the Capability of Pix2Pix in Producing Synthetic Images

Arsyi Syarief Aziz
Muh. Yusuf Syam
Aso Ahmad Amin Rais
Computer Science Study Program
Universitas Hasanuddin

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Evaluating the Capability of Pix2Pix in Producing Synthetic Images

***Abstract*—Pix2Pix is a general purpose framework that is designed for image-to-image translation. This framework implements a the conditional type of a generative adversarial network, known a cGAN. The use of cGANs in this framework allows it to be used in various context, such as image reconstruction, and image coloring. To explore this flexibility, we have conducted an evaluation on the framework to test its ability in imitating images from three different datasets: *maps*, *edges2shoes*, and *facades*. The assessment of the framework's performance was done by manual inspection though the use of a quick response program. The result of this evaluation shows that Pix2Pix was able to successfully synthesize realistic looking images for two of the three datasets.**

I. Introduction

In this technical report, we will discuss about the capability of the Pix2Pix framework in synthesizing realistic images.

Pix2Pix in this report is defined as a general purpose framework that is designed for image-to-image translation. This framework is built upon the idea of a generative adversarial network (GAN), specifically the conditional type. The use of conditional GANs in this context allows the network to generate image samples that correspond to a given label image.

Many people have demonstrated the use of this framework to generate images in many context, such as image reconstruction, image coloring. To further explore the flexibility of this framework, we will evaluate its capability in imitating images from three different datasets: *maps*, *edges2shoes*, and *facades*.

More details about the framework and our evaluation methodology are explained in the following sections.

II. Problem Defintion

This technical report focuses on understanding the capability of the Pix2Pix framework in synthesizing realistic looking images. To understand this capability, we have completed a few experiments on the framework, testing it on three different datasets from [7]. Ultimately, through the results of these experiments we will answer one important question regarding the Pix2Pix framework: Is it able to produce realistic looking images for different types of images?

III. Related Works

Before we explain about the Pix2Pix framework, we will first explain about the underlying concepts that construct it. Specifically speaking, this section will provide the definition of the generative adversarial network and the conditional generative adversarial network.

A. Generative Adversarial Network

Generative adversarial network (GANs) were created by [2] to provide a framework to estimate generative models. It works though an adversarial process where two model compete with each other to be better than their counterpart. These models are a generator G, used to capture a data distribution, and a discriminator D, used to predict whether a sample is synthetic or not.

The training process of this framework consists of three steps. First, the generator generates a synthetic image from a random distribution $p_z(z)$. Second, the generated synthetic image is inputted into the discriminator along with a real image, where the discriminator is tasked to correctly identify the synthetic image. Third, the parameters of the two models are updated based on the results of the prediction. This process is repeated continuously until the generator is able to consistently fool the discriminator.

The above explanation might seem intuitive as it is analogous to how a criminal would learn to create counterfeit money and avoid law enforcement, however it is just a simplification of what actually occurs during training. To formally define the training process, we state the following goals of the models. For the generator model, its goal is to learn a function $G(z;\theta_g)$ such that it maps a random noise $p_z(z)$ to the data space. Meanwhile, for the discriminator, its goal is to learn a function $D(x;\theta_d)$ that outputs a scalar value that represents the probability that an input x originates from training data and not from the generator [2].

In addition to generating images by the use of a generator and identifying images through a discriminator, we have also previously stated that the parameters of the models have to be updated based on the results of the prediction. These parameters follow a formula. For the generator, its parameters are updated to minimize $\log\left(1 - D(G(z))\right)$, and for the discriminator, parameters are updated to maximize $\log\left(D(x)\right)$.

As one might be able to see, the described training process of a GAN follows the minimax game [2]. Thus, by defining V(D, G) as its value function, we can define the objective function as equation 1.

$$\begin{aligned}\min_G \max_D V(D,G) =& \mathbb{E}_{x\sim p_{data}(x)}\left[\log D(x)\right] + \\ & \mathbb{E}_{z\sim p_z z}\left[\log\left(1 - D(G(z))\right)\right]\end{aligned} \tag{1}$$

B. Conditional Generative Adversarial Networks

Conditional generative adversarial networks (cGANs) are a variant of GANs that condition the generator and disciminator

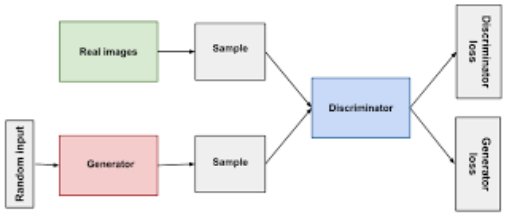

Fig. 1. Structure of the GAN framework [1]

.

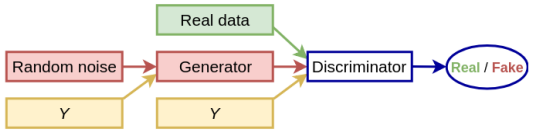

Fig. 2. Structure of the cGAN framework [5]

.

based on some information y [4]. This conditioning allows cGANs to be able to produce samples that correspond to y, such as samples that belong to a specific class.

Implementation wise, cGANs are similar to normal GANs, however, as we have stated, its models are conditioned on y. To condition these models, we add y to both of their inputs. For the generator, we use y and some random noise vector z and map them to the data space, and for the discriminator we use y and input data x to identify whether x originates from training data or from the generator.

From this description, we can define the objective function of cGANs as equation 2.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x|y)] + \mathbb{E}_{z\sim p_z z}[\log(1 - D(G(z|y)))] \quad (2)$$

IV. Pix2Pix

Pix2Pix was created by [3] to provide a common framework to conduct picture-to-picture translation. More specifically speaking, it is used to translate a representation image of an object, called a label image, to a synthetic image that resembles an object.

The following subsections will explain about the Pix2Pix framework though the description of its objective function, network architecture, and how it is optimized.

A. Objective Function

Pix2Pix is built upon the cGAN framework. In this implementation, a label image is used to condition both the generator and discriminator to generate images corresponding to that label image. In addition to using cGANs, Pix2Pix also implements the L1 distance in its objective function. This addition helps helps Pix2Pix capture low frequencies [3].

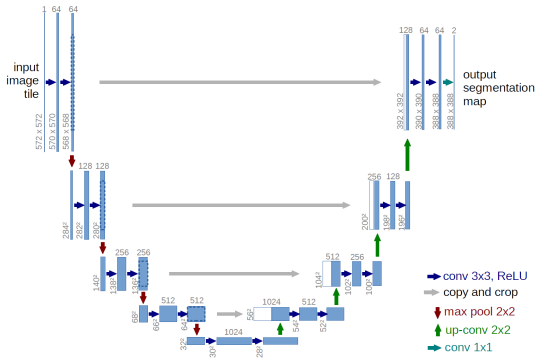

Fig. 3. U-Net architecture [6]

From this description, we can define the objective function of the Pix2Pix framework

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

B. Model Architecture

Pix2Pix implements two different architectures for its generator and discriminator. For its generator, Pix2Pix implements the "U-Net" architecture. Meanwhile, for its discriminator, Pix2Pix implements the "PatchGAN" architecture.

1) U-Net: U-Net is an implementation of the encoder-decoder architecture that contains skip connections in between its encoder and decoder blocks. The addition of these skip connections allow the U-Net architecture to avoid data bottlenecks which are caused by the downsampling and upsampling processes. Additionally, it also allows low level information from the input image to be easily propagated to the resulting output image [3].

The U-Net architecture consists of two main parts, which are:

- *The contracting path* (left of Fig.3) which is built by a general convolutional process that downsamples sample data during information extraction.
- *The expansive path* (left of Fig.3) which is built by transposed 2D convolutional layers that upsamples information.

2) PatchGAN: PatchGAN, also known as the Markovian discriminator is a type of discriminator that classifies an image by dividing the image into blocks of size NxN, known as patches.

To classify the images, the discriminator does a convolution on all the patches in an image and tries to classify whether or not the patches represent real or fake images. The result of each of these classifications is not a scaler, but an array of size NxN, where each element of the array has a value in the range of [0-1]. The result of the overall classification of the image is obtained by finding the average value of each patch.

Dividing an image into several patches is advantageous to the models performance. By reducing the image into small patches, then the number of parameters required by the model is thus reduced (see [3]).

C. Optimization

Optimization of the networks in this framework follows the standard approach from [2]. In this approach, we alternate between one gradient descent step on D and one gradient decent step on G. However, rather than training G to minimize $\log(1 - D(x, G(x, z))$, this framework instead trains to maximize $\log D(x, G(x, z))$ [3]. Additionally, we also divide the objective function by 2 when optimizing D, which helps slow down the learning rate of D relative to G.

V. Criteria for Assessing the Framework

The criteria we used to assess the Pix2Pix framework is based on whether or not it is capable of producing realistic looking images. As it can be difficult to define a quantitative measure for this criteria, we conducted this assessment based on manual inspection of the images through the use of a quick response program.

This program evaluates 50 random images relating to a specific data set, which include 25 real images (the control group) and 25 fake images (the experimental group). Each of these images are displayed one-by-one and at random to an human assessor, who is given a few seconds to identify whether or not the corresponding image is real or fake. After assessing all the images, the program will display the number of correct identifications for both the control group and the experimental group, as well as the label image, ground truth image, and predicted image corresponding to the each of the random images.

VI. Methodology

To provide an understanding of how our experiments were conducted, this section will explain about the methodology of our experiments. This includes an explanation about the datasets and the hyperparameters used.

A. Dataset

The datasets we used in our experiments are three of the datasets used in the original Pix2Pix paper, which include the *facades*, *maps*, and *edges2shoes* data sets. These datasets can be retrieved from [7].

Each of these data sets consists of ground truth image and label image pairs. Firstly, in the facades dataset, the ground truth images are pictures of building facades and its label images are architectural elements the buildings. Secondly, in the maps data set, its ground truth images are satellite photos and its label images are map images. Finally, in the edges2shoes data set, the ground truth images are pictures of shoes and its label images are sketches of shoes.

B. Hyperparameters

For each of the data sets, we trained a separate Pix2Pix model which use the same hyper-parameters. These hyper-parameters include the use of mini-batch SGD and the Adam optimizer with a learning rate of 0.0002 and a β_1 value of 0.5, as well as a total of 100 epochs.

Fig. 4. A real maps image which was incorrectly identified

.

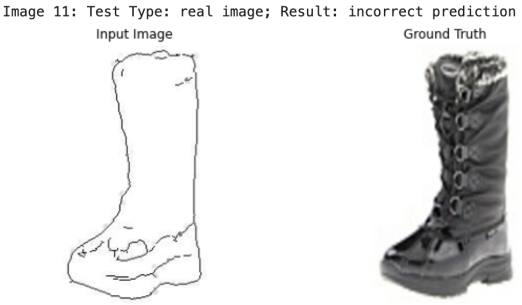

Fig. 5. A real edges2shoes image which was incorrectly identified

.

VII. Analysis and Interpretation

Table I shows the results of our evaluation. In this table, we can see that for both the *maps* and *edges2shoes* datasets, our human assessor was only able to correctly identify 60% of the synthetic images and 72% of the real images. This indicates that the models were able to synthesize a few realistic images for these data sets. However, it also indicates that our human assessor had some difficulty in identifying the real images, which can be attributed to the low resolution of the images (see fig 4 and fig 5). For the *facades* dataset, our human assessor was able to identify almost all of the images correctly (96% of real images and 96% of fake images). This indicates that our Pix2Pix model could not produce realistic looking images for this dataset.

Upon close inspection of the resulting images (see Appendix), we made a couple of discoveries related to Pix2Pix's ability in producing realistic images. The first discovery was that Pix2Pix favors simple label images, such as the edges used in the edges2shoes dataset. The second discovery was that Pix2Pix favors label images that preserves a lot of information from its ground truth image. This idea is supported by the fact that the edges2shoes and the maps datasets (which contain label images that are just a simplification of the ground truth images) were able to be used to synthesize a few realistic images. Meanwhile, the facades dataset (which contain label images that are only architectural elements of a building) had difficulty producing elements not shown by the label image, such as the roof of buildings.

Dataset	Group	
	Control (Real Image)	Experiment (Synthetic Image)
maps	72%	60%
edges2shoes	72%	60%
facades	96%	96%

TABLE I. PERCENTAGE OF CORRECTLY IDENTIFIED IMAGES FOR EACH DATASET

VIII. CONCLUSION AND RECOMMENDATIONS

The results of this technical report suggest that the Pix2Pix framework is able to synthesize realistic looking images for certain types of datasets. Out of the three datasets that we used, only two of them were able to successfully synthesize a reasonable amount of realistic images, these were *maps* and *edges2shoes* datasets.

Following this conclusion, we recommend that further evaluation should be conducted on this framework with different datasets to better understand the capabilities of this framework.

REFERENCES

[1] G. Developers, *Overview of gan structure*, image retrieved from https://developers.google.com/machine-learning/gan/gan_structure, 2019.

[2] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[3] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[4] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[5] M. Nayak, *High-level cgan's architecture diagram*, image retrieved from https://medium.datadriveninvestor.com/an-introduction-to-conditional-gans-cgans-727d1f5bb011, 2019.

[6] O. Ronneberger, P. Fischer, and T. Brox, *Overview of gan structure*, image retrieved from https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/, 2015.

[7] University of California, Berkeley, *Pix2pix datasets [data]*, data retrieved from http://efrosgans.eecs.berkeley.edu/pix2pix/datasets, 2017.

APPENDIX A
IMAGE SAMPLES

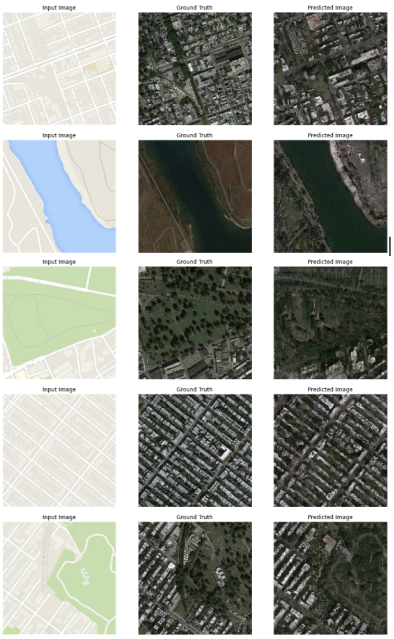

Fig. 6. Five samples taken from the maps dataset.

Fig. 7. Five samples taken from the edges2shoes dataset.

Fig. 8. Five samples taken from the facades dataset.

.