Bachelor Project

Czech Technical University in Prague

F3

Faculty of Electrical Engineering
Department of Cybernetics

Data Augmentation by Image-to-Image Translation for Image Retrieval

Albert Möhwald

Supervisor: Ing. Tomáš Jeníček
Field of study: Open Informatics
Subfield: Computer Science
May 2020

BACHELOR'S THESIS ASSIGNMENT

I. Personal and study details

Student's name: **Möhwald Albert** Personal ID number: **474592**

Faculty / Institute: **Faculty of Electrical Engineering**

Department / Institute: **Department of Cybernetics**

Study program: **Open Informatics**

Branch of study: **Computer and Information Science**

II. Bachelor's thesis details

Bachelor's thesis title in English:

Data Augmentation by Image-to-Image Translation for Image Retrieval

Bachelor's thesis title in Czech:

Generování trénovacích dat pro vizuální vyhledávání pomocí neuronových sítí

Guidelines:

1. Familiarize yourself with conditional Generative Adversarial Networks (cGAN) for image-to-image translation.
2. Identify suitable cGAN methods and datasets for the task of translation between different lighting conditions, such as day and night. Choose one method that requires pairs of training images and one that works with domain-labeled images alone.
3. Implement the chosen cGAN methods in an existing image retrieval codebase for both, training and testing.
4. Train both methods on the chosen datasets.
5. Define an evaluation protocol suited for the consequent goal of improving the image-retrieval performance.
6. Use the trained methods to augment training data for image retrieval in order to make an image retrieval method illumination-invariant.
7. Evaluate the proposed approach on standard image retrieval benchmarks and compare the cGAN methods based on paired and unpaired training data.
8. Provide self-contained codes and their documentation.

Bibliography / sources:

[1] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros; Image-to-Image Translation with Conditional Adversarial Networks; The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1125-1134; https://arxiv.org/pdf/1611.07004.pdf
[2] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros; Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks; The IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2223-2232; https://arxiv.org/pdf/1703.10593.pdf
[3] Filip Radenović, Giorgos Tolias, Ondřej Chum. Fine-tuning CNN image retrieval with no human annotation; IEEE transactions on pattern analysis and machine intelligence, 2018, 41.7: 1655-1668. http://cmp.felk.cvut.cz/~radenfil/publications/Radenovic-TPAMI18.pdf
[4] Arruda et al.; Cross-Domain Car Detection Using Unsupervised Image-to-Image Translation: From Day to Night; International Joint Conference on Neural Networks (IJCNN), 2019; https://ieeexplore.ieee.org/document/8852008

Name and workplace of bachelor's thesis supervisor:

Ing. Tomáš Jeníček, Visual Recognition Group, FEE

Name and workplace of second bachelor's thesis supervisor or consultant:

Date of bachelor's thesis assignment: **10.01.2020** Deadline for bachelor thesis submission: **22.05.2020**

Assignment valid until: **30.09.2021**

Ing. Tomáš Jeníček	doc. Ing. Tomáš Svoboda, Ph.D.	prof. Mgr. Petr Páta, Ph.D.
Supervisor's signature	Head of department's signature	Dean's signature

III. Assignment receipt

The student acknowledges that the bachelor's thesis is an individual work. The student must produce his thesis without the assistance of others, with the exception of provided consultations. Within the bachelor's thesis, the author must state the names of consultants and include a list of references.

. _____
Date of assignment receipt

_____
Student's signature

Acknowledgements

Chtěl bych poděkovat vedoucímu práce panu Ing. Tomáši Jeníčkovi za jeho trpělivost, velké množství času, mnoho konstruktivních konzultací a vedení, jež mi poskytnul s touto prací.

I would like to express my sincere thanks to my supervizor Ing. Tomáš Jeníček for his patience, high amount of time, numerous of constructive consultation and guidance he provided me with this thesis.

Declaration

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze dne 22. února 2020

I declare that the presented work was developed independently and that I have listed all sources of information used within it in accordance with the methodical instructions for observing the ethical principles in the preparation of university theses.

Prague, 22. May 2020

Abstract

Daytime and nighttime visual appearance changes are addressed with artificially learned data augmentation. Convolutional neural networks (CNNs) are one of the state-of-the-art techniques for image retrieval. However, powerful deep neural networks are data-driven resulting in poor performance, when an irregular query, different from training data, is inputted. Augmentation is addressed with pix2pix a CycleGAN, used to provide image-to-image translation from regular daytime images into irregular nighttime images and are trained over four image datasets. To measure image translation quality, Generative Adversarial Network (GAN) evaluation scores are explored and compared with data augmentation. The final data augmentation effect is tested on the image retrieval benchmarks, where results show improvement on the 24/7 Tokyo dataset with minor performance loss on daytime Revisited Oxford and Paris datasets.

Keywords: Image Retrieval, Data Augmentation, Generative Adversarial Network, Image-to-image Translation

Supervisor: Ing. Tomáš Jeníček

Abstrakt

Denní a noční změny vzhledu obrázků jsou řešeny uměle naučenou augmentací dat. Konvoluční neuronové sítě (CNN) jsou jednou z nejmodernějších technik pro vizuální vyhledávání. Nicméně, výkon hlubokých neuronových sítí je závislý na počtu dat. pokud dojde k zadání nepravidelného vyhledávání, které se liší od učících dat, projeví se to na nízké úspěšnosti vyhledávání. Augmentace je provedena pomocí pix2pix a CycleGAN, jež poskytují překlad z obrázku do obrázku, kde z běžných denních obrázků jsou generovány nepravidelné noční obrázky, a tento překlad je trénován na čtyřech datasetech. Pro změření kvality překladu obrázků jsou využita evaluační skóre pro generující adversariální sítě (GAN), která jsou v této práci zkoumána a porovnána s datovou augmentací. Výsledný efekt augmentace je testován prostřednictvím meřítek pro visuální vyhledávání, kde výsledky ukazují zlepšení na datasetu 24/7 Tokyo za menší ztráty výkonu na znovuvytvořených datasetech Oxford a Paris.

Klíčová slova: Vizuální vyhledávání, Augmentace dat, Generující adversariální sítě, Překládání obrázku do obrázku

Překlad názvu: Generování trénovacích dat pro vizuální vyhledávání pomocí neuronových sítí

Contents

Figures

Tables

Chapter 1

Introduction

Image retrieval is the task of finding image entries in a large image database given a query image. For instance, imagine you took a photo containing Prague Astronomical Clock (Orloj), and now, you are interested in other images similar to Prague Orloj. Images related to the inputted image can be found with image search engines, where you input the photo, the engine performs an image retrieval search over an image database, and finds images most related to the input; see top images in Figure 1.1.

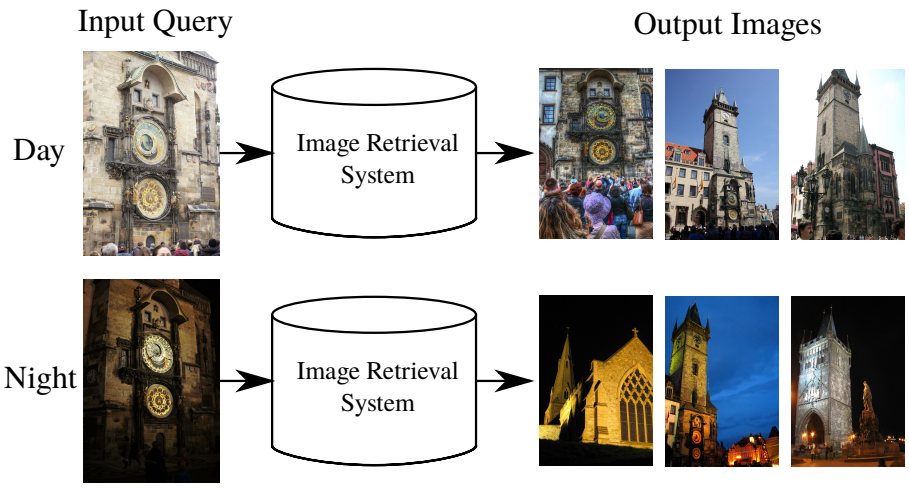


Figure 1.1: An example of an easy and challenging *search-by-example* query. The input query is a image of Prague Orloj in day or night visual domains. Input images have a similar viewpoint, but different lighting conditions. Pridat denni jenThe output is expected to return images related to input images. However, in the night visual domain (bottom), the output contains images of unrelated buildings having similar visual appeareance to the input.

Recently, the increasing performance of artificial neural networks enabled image retrieval to improve. Neural networks learn to perform tasks from examples without task-specific programming of manual rules. Such neural networks perform embedding with "simpler" data as *embedding networks*, which transform images into vectors while maintaining image retrieval information. The dimensionality is reduced by this operation, e.g. the dimension

of $1024 \times 1024 \times 3$ pixel image (tensor) is reduced into 512-sized vector. A main advantage of embedding is that search becomes extremely efficient in the lower dimension. Specifically, to find images similar to Prague Orloj in a retrieval database, a vector of Orloj from the input Orloj image is obtained by embedding network, and then database images having the same or most similar vectors to the Orloj vector are outputted as the search result.

Even today image retrieval has its limits. The inputted photo could be taken under conditions infrequent in the training dataset, such as nighttime, and when the night photo of Prague Orloj is inputted, the most related image found with the image retrieval system can be unexcepted and faulty, because the photo of Orloj taken at daytime is visually very different to the photo of Orloj taken at nighttime, although the image content is similar. See bottom images in Figure 1.1. The cause of this problem lies in a large imbalance of day/night training data. Therefore, to learn equivalent retrieval in day and night, the success lies mainly in lots of diverse training data examples provided to the embedding network. However, such a training dataset containing all images taken under all adversarial weather conditions can be expensive or even impossible to obtain.

In this work, training data insufficiency is solved by artificially generating fake images and adding them to training data, which is known as *data augmentation*. This thesis aims to increase image retrieval performance through data augmentation by generating fake night images dealing with the most usual daily base varying lighting conditions - daytime and nighttime. The rise of generative adversarial networks (GANs) [1] allowed GAN-based image-to-image translation methods to develop [2]. In this work, two image-to-image translation methods – pix2pix [3] and CycleGAN [4] – are used to transform daytime into nighttime images, and they are trained and validated on four different image datasets. Also, for the consequent image retrieval performance training, GAN evaluation methods are explored and used to measure image translation model scores under different hyperparameter settings. According to GAN evaluation, the best performing settings are chosen to train and test image retrieval embedding networks with training data augmentation.

In Chapter 2, terms used in this work are described as well as necessary background. Also, work, related to domain adaptation is analyzed. In Chapter 3, training datasets, GAN together with image retrieval embedding network architecture and implementation are described. In Chapter 4, evaluation scores are defined and applied to measure image-to-image translation quality. In Chapter 5, data augmentation results are shown. Finally, in Chapter 6, the thesis is concluded.

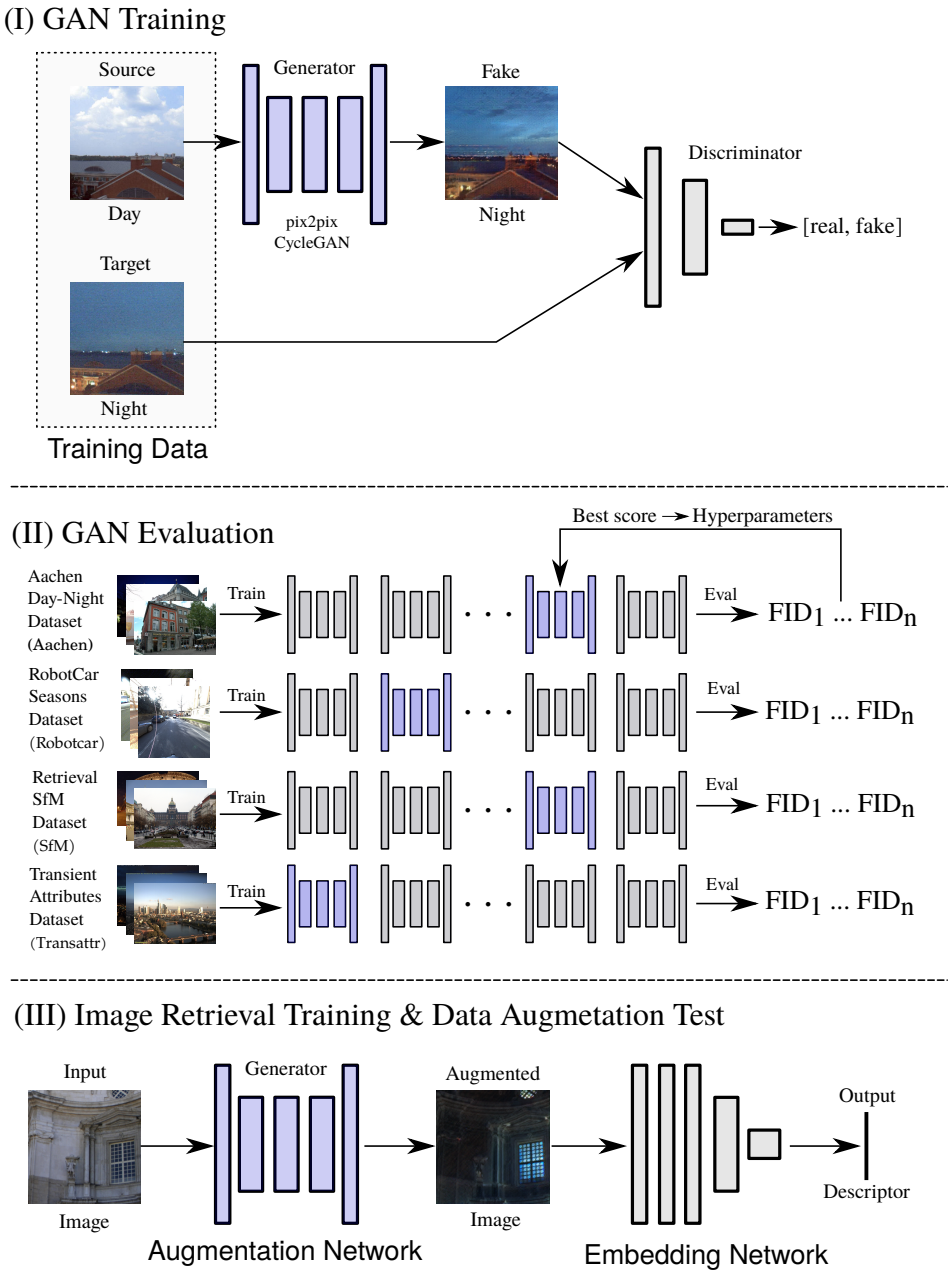


Figure 1.2: The overview of this work approach. (I) Training for pix2pix and CycleGAN was implemented. Networks are trained to perform day to night image-to-image translation. (II) Evaluation measures were explored and chosen metrics (e.g. FID) are used to measure image translation quality per network with given hyperparameters. For each dataset, best-performing hyperparameters are searched. (III) Chosen parameters are used to train data augmentation network per dataset, and then, to train and test embedding networks for image retrieval.

Chapter 2
Background

In Section 2.1, the image retrieval is introduced together with its challenge of searching images given an input image with unexcepted photometric conditions caused by adversarial weather. In this work, only the most common varying condition (the change of day and night visual conditions) is solved. The solution to the generalization inability is visual domain adaptation, where common day images are transformed into night images, intending to augment data to the image retrieval embedding network training with image-to-image translation, described in Section 2.2. Such technology, able to provide proper image-to-image translation, is a generative adversarial network (GAN), with optimized network architectures, described in Section 2.3. In the end, in Section 2.4 related work associated with image retrieval, image-to-image translation, and GANs is analyzed.

2.1 Image Retrieval

Image retrieval is the computer vision problem where the task is to search for images with similar content in a large collection of digital images. The input of the search query can be given in different formats such as **keyword**, e.g. "car", **image**, for example find images similar to input image. When the search analyses text-based metadata of images, e.g. keywords, headings, etc. this problem is called as **context-based image retrieval**. Vice versa, when the content of images is analyzed, e.g. pixels, colors, textures, etc. this task is referred to **content-based image retrieval**. In this work, only content-based image retrieval is solved.

Image retrieval is effectively done through an trained CNN **embedding network**, which provides the mapping from images into image descriptors [5] Then, image retrieval is performed in two main steps:

1. An image descriptor is extracted from the input image.
2. Image retrieval is executed by Eucledian search [5], or nearest neighbor methods [6] to find the most similar image descriptors corresponding to its similar images in an image database.

According to Jenicek & Chum [7], there are two types of challenges in image retrieval. The first main challenge is in increasing retrieval efficiency for increasing the size of image collections. The second challenge lies in varying

geometric and photometric conditions in image collections, examples are as follows:

- *Scale and/or viewpoint change.* The input query image can be zoomed-in or zoomed-out with respect to most of the images in the image database.
- *Occlusion.* There is an object in the input query image that blocks the view on the retrieved object.
- **Visual appearance change**. The input query image has different lighting, weather, or time conditions, for example, spring/summer/autumn/winter, day/night, etc.
- *Different objects that are visually similar.* For instance, you input an image query with the Arch of Titus in Rome. However, you receive an image of the Arc de Triomphe in Paris.

2.1.1 Adversarial Weather

Different weather and time conditions can change a visual appearance greatly. In general, images taken at different times, weather or seasons have overall diverse characteristics, causing different viewing conditions, such as lighting and colors, which, for example, affect vehicular navigation systems to match its position incorrectly from place images [8].

A more challenging situation arises when a content-based image retrieval system is tasked with uncommon input, different from most of the training data. Usually, images are taken under normal conditions, but there could be a lack of images taken under more extreme conditions causing retrieval systems to output unexcepted images showing poor generalization [7]. For CNN based image retrieval, different illumination changes cause retrieval systems to fail, notwithstanding, the structure of images is preserved [9]. As a leading example, in the *query-by-example* i.e. a task of searching images similar to the input image, when the input of the search query is Prague Orloj at night, the output could be night images of anything other than Prague Orloj. More specifically, see Figure 1.1.

2.1.2 Visual Domain Adaptation

The input data could be difficult to represent for many deep learning systems [10]. State-of-the-art CNN based methods often fail when illumination conditions change [11]. The output in the leading example (Figure 1.1) indicates that the deep network of a retrieval system is unable to generalize new inputs in the night visual domain, which did not learn previously. Therefore, there is a desirable task of processing images, that are in arbitrary (day or night) visual domain.

In the computer vision area, Goodfellow et al. describe (this whole paragraph cite [10]), the most usual way dealing with difficult inputs for deep networks is data preprocessing. The best way to improve machine learning model is to provide more training data. However, in practice, the amount of available data is limited. A straightforward solution is a **data augmentation** i.e. to generate fake data and add it to the training set, for example

adding rotated, zoomed, sliced images, adding images with Gaussian noise, etc. Data augmentation can be seen as data preprocessing limited only on a training dataset. In the problem with the lack of night images, the domain adaptation solution is to augment training images the fake images of night domain generated from available day images.

2.2 Image-to-Image Translation

The data augmentation task requires a function, which can generate fake night images from any real day image, i.e. a mapping, that transforms an image in the day domain into an image in the night domain.

Image-to-image translation is the task of taking images from source visual domain and translating them into target visual domain, so they have the same characteristics (style, or representation) as the target domain [3].

2.2.1 Image Translation Notation

Let $h \in \mathbb{N}$ be height of images, $w \in \mathbb{N}$ be width of images. Let $X \subseteq \mathbb{R}^{h \times w \times 3}$ be the day image visual domain. Image x is from the day image domain X if $x \in X$. Let $x_{i,j,1}, x_{i,j,2}, x_{i,j,3}$ denote red, green, blue, intensity at row i and column j, respectively. Let p_X denote data distribution of the domain X. When image x follows distribution p_X, it is denoted as $x \sim p_X$. Night domain Y, night image y and night data distribution p_Y are defined similarly.

2.2.2 Supervised and Unsupervised Learning

The terminology of supervised and unsupervised machine learning is different from the terminology of image-to-image translation, although the image-to-image translation is a machine learning problem.

Supervised machine learning is the task, where each training data sample is associated with target or label. For example, training samples can be in form $\{(x_i, k_i)\}_{i=1}^{N}$, where x_i is the data sample and k_i is its corresponding class (domain, or label). In the learning process, the task is to learn to predict k from x, where feedback can be provided to the trained system in form of target k, with an analogy to a teacher, who shows to trained system what to do providing labels k [10].

Unsupervised machine learning is the task, when there is no label with the training observations, nor information about the output. An example of a possible training data form is $\{x_i\}_{i=0}^{N}$, where x_i is the data sample. The tasks can be to learn the probability distribution $p(x)$, clustering, dimensionality reduction, or interesting properties observation [10].

Paired image-to-image translation (also known as *supervised image-to-image translation*) is trained on images provided in pairs, such that the first image is in the source domain and the second image is the same as the first image but in the target domain [3]. Image pairs have the same structure, but different visual attributes usually.

Unpaired image-to-image translation (also known as *unsupervised image-to-image translation*) receives training images in both source and target domain, but without the ground truth target image paired with its respective source image [4]. Although unpaired image-to-image translation is also called *unsupervised*, it is weakly supervised machine learning, since there are available expected attributes of the output in the target domain.

For the difference between paired and unpaired image-to-image translation, see Figure 2.1.

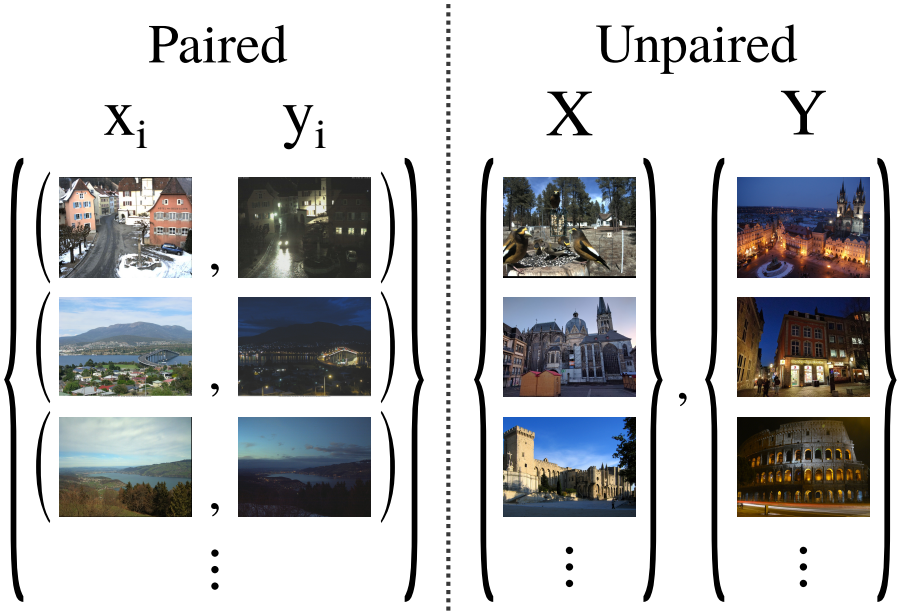


Figure 2.1: The difference between paired (left) and unpaired (right) image-to-image translation training examples. For paired translation, training samples are available in form of corresponding image pairs. In unpaired translation, only source and target domains are available with no information about matching samples. X denotes image set in the day visual domain, Y denotes image set in the night visual domain; $x_i \in X$ and $y_i \in Y$ are corresponding image pairs from day and night visual domains, respectively.

2.3 Generative Adversarial Networks (GANs)

Image retrieval systems generalize poorly when lighting conditions of the input query suddenly change. Such kind of input can be viewed as an adversarial sample, available to be used for other adversarial samples recreation. Training data used for image retrieval systems can be augmented with adversarial samples, resulting in more robust image retrieval. However, providing just random adversarial weather samples is useless, since those examples can be in a different class, for example, embedding network is trained primarily on day images of cities, but from the night, only night images of countryside domain are provided. Therefore, I describe a generator network able to provide the

mapping from the training sample into its corresponding adversarial training sample.

At first, original GAN and its intermediate stage conditional GAN (cGAN) are described, however not used in this work. Then, the pix2pix and the CycleGAN are described as they are derived models used for day to night image-to-image translation. Also, convergence problems related to image-to-image transformation are described as an introduction to evaluation.

Pix2pix and CycleGAN are both derived GAN models from the original GAN. Derived GAN models can be split into two main groups: **Architecture optimization-based GANs** (e.g. DCGAN) and **Objective function optimization-based GANs** (e.g. WGAN-GP) [2]. In this work, I only focus on architecture optimization-based GANs.

2.3.1 Original Game-theoretic GANs

In 2014, Goodfellow et al. proposed a deep generative model estimating a full probabilistic model trained with the adversarial learning process [1]. Before 2014, deep generative models (maximum likelihood estimation and similar methods) had smaller success than deep discriminative models especially due to CNN classifiers [1]. In the GAN training, two networks simultaneously compete against each other. A **generator** network G generates fake data fitting training data distribution as much as possible, whereas a **discriminator** network D distinguishes if generated data are real or fake.

GAN

GAN learning can be expressed in a game-theoretic manner. Let p_x be training data distribution and p_z be a random noise variable distribution. Training data sample x and random noise z can be represented as a single value, vector, or tensor.[1] The objective of the generator is to generate data $G(z)$, so it mimics x, the objective of the discriminator is to correctly classify real data x and fake data $G(z)$, so that the probability $D(x)$ approaches 1 and the probability $D(G(z))$ approaches 0, see Fig 2.2 [1]. G and D play the following minimax two-player zero-sum game,

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_x}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))], \qquad (2.1)$$

where V is the value function [1]. From 2.1 we can see that objectives of D and G are conflicting each other. When $D(G(z)) = \frac{1}{2}$, the discriminator cannot determine, if $G(z)$ comes from training data distribution or from fake data distribution; this state is the global optimum for optimization task 2.1 [1].

[1] There, the notation of x is overloaded. From the image notation Section 2.2.1 x is image in the day domain. In GAN and cGAN formulation, x is more abstract training data sample taken from the distribution p_x, which can be also an image.

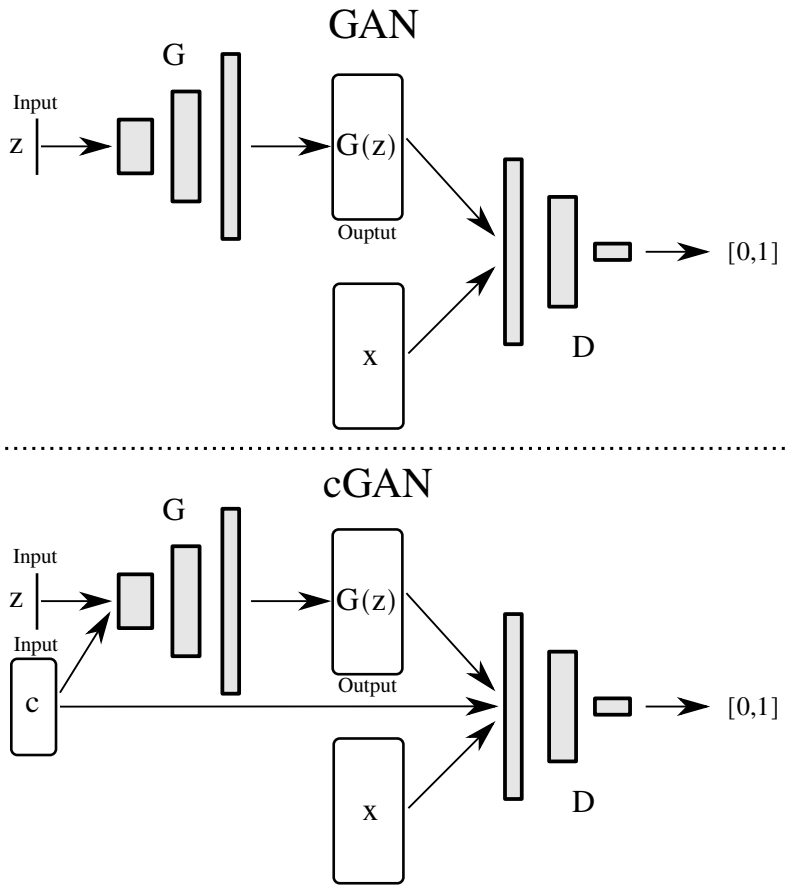


Figure 2.2: Architectures of GAN and cGAN. The generator G tries to transform a random noise vector z into fake sample $G(z)$. The discriminator D tries to classify if data sample x or $G(z)$ is real or fake, respectively. In the cGAN architecture, the generator and the discriminator also takes c as the conditional input.

cGAN

In contrast with basic GANs, the generator and the discriminator of conditional generative adversarial networks (cGANs) receive extra information c, see Fig 2.2. Then, G and D play a minimax game:

$$\begin{aligned}\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_x}[\log D(x,c)] \\ + \mathbb{E}_{z\sim p_z}[\log(1-D(G(z,c)))],\end{aligned} \tag{2.2}$$

which is similar to 2.1, [12].

2.3.2 Pix2pix

Pix2pix, proposed by Isola et al. [3], is cGAN derived model with optimized architecture suited for image-to-image translation from one visual domain into another. However, learning image-to-image translation with pix2pix is

supervised, and thus, pix2pix requires strictly pixel-aligned training example pairs (Figure 2.1, paired case).

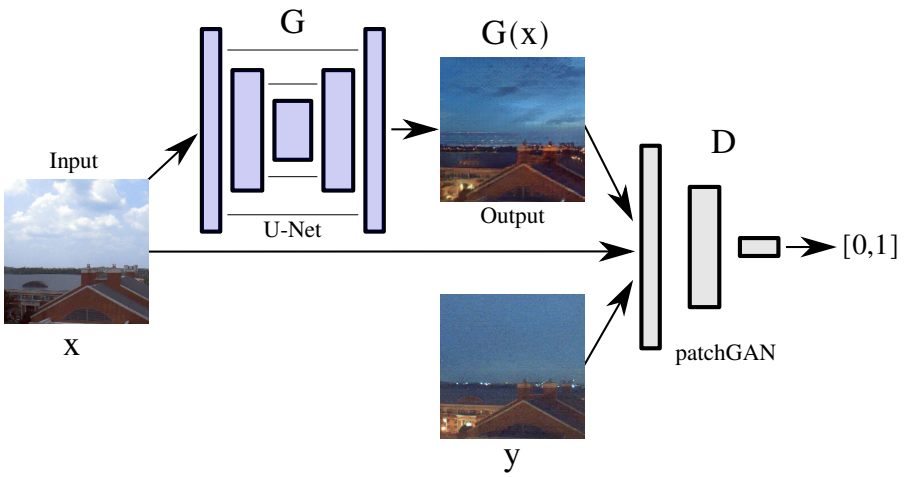


Figure 2.3: The architecture of pix2pix. The U-Net generator G (blue) transforms a day image x into fake image $G(x)$. The patchGAN discriminator D tries to classify if image y or $G(x)$ is real or fake, respectively. Since the model is conditional, the discriminator also takes x as the input.

Formulation

The goal of pix2pix is to learn the mapping from domain X to domain Y given training sample pairs $\{(x_i, y_i)\}_{i=0}^{N}$, where $x_i \in X$ and $y_i \in Y$ are corresponding image samples[2]. Pix2pix consists of a generator $G : X \to Y$ and a discriminator $D : X \times Y \to [0, 1]$ [3].

As pix2pix is derived cGAN, it needs additional information common for the generator and the discriminator, which is the input domain image x, and therefore, discriminator also receives image x, see Figure 2.3 [3].

Loss

Considering the generator defined by Goodfellow et al. [1], it is important to notice, that G does not take a noise vector. Still, G can learn the mapping from X to Y, resulting in G to produce deterministic outputs [3]. To avoid this, a Gaussian noise z could be added as additional input with x. Unfortunately, this strategy is ineffective since G learns to ignore noise z, so there is no difference between learning $G : X \to Y$ and $G : X \times Z \to Y$, where Z is the set of all latent noise vectors [3, 13]. In order to have noise in G, I experiment with dropout in generator architectures the same way as it is implemented by Isola, Zhu, et al. [3, 4].

When the equation 2.2 is adapted for the image-to-image translation from

[2]For simplicity, i is omitted.

domain X into Y, G and D play the following game:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_X, y\sim p_Y}[\log D(y,x)] + \mathbb{E}_{x\sim p_X}[\log(1 - D(G(x),x))]. \tag{2.3}$$

Then, the loss of pix2pix is expressed as

$$\mathcal{L}_{cGAN}(D,G) = \mathbb{E}_{x\sim p_X, y\sim p_Y}[\log D(y,x)] + \mathbb{E}_{x\sim p_X}[\log(1 - D(G(x),x))]. \tag{2.4}$$

When G tries to trick D, G can produce fake images $G(x)$ that confuse D, without visual similarity between $G(x)$ and y [13]. Previous works solve this using a loss with regularization. The most common regularization approach is to mix the $\mathcal{L}_{cGAN}(D,G)$ with L2 distance. Note that regularized loss makes a difference only for the G, the optimization step for the D remains unchanged. However, regularized loss with the task of minimizing L2 distance between fake and ground truth images results in blurred images, because L2 distance is minimized by averaging all feasible outputs [3]. This problem is solved with L1 distance regularization instead of L2 distance regularization. The regularizing loss for G is

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x\sim p_X, y\sim p_Y}[||y - G(x)||_1]. \tag{2.5}$$

Note that there is no need to use the discriminator with regularization.

The resulting optimization task can be expressed as:

$$\min_G \max_D \mathcal{L}_{cGAN}(D,G) + \lambda \mathcal{L}_{L1}(G), \tag{2.6}$$

where $\lambda \in \mathbb{R}$ is regularization multiplier hyperparameter.

Architecture

The generator is required to render images with the same resolution as the input image and to preserve the structure of the input image. A generator network meeting these requirements is an encoder-decoder U-Net shaped network [14], where the input information is downsampled through series of layers until it reaches the bottleneck layer from which it is upsampled in a reverse way from downsampling [3]. Also, to preserve input image structure, which could be lost in the bottleneck layer, skipping connections between downsampling and its respective upsampling layer is added [3].

PatchGAN discriminator is designed to only penalize structures at the scale of patches. In practice, for each $N \times N$ image patch, where $N \in \{1, \ldots, \min(h,w)\}$, the patchGAN tries to determine whether that patch is real or fake [3]. This discriminator can classify images of arbitrary sizes [3].

2.3.3 CycleGAN

Sometimes, training data sample pairs of both visual domains are not available, are expensive to obtain, or could be impossible to obtain e.g. obtaining a

sufficient amount of image pairs for Photo & Vincent-Van-Gogh painting visual domains is impossible (Figure 2.1, unpaired case). For the desired image-to-image translation generator, when enough data from both source and target domains are available, supervision can be applied on set base, however, in contrast with pix2pix, no conditional information is provided, input image structure cannot be bounded to the target image, and therefore, such a network is likely to produce any output of the target domain or fall into mode collapse [4]. CycleGAN bridges this gap with **cycle consistency**, requiring recreated "double-fake" image, reconstructed back with the second generator from the target domain into the source domain, to be identical with the real input image.

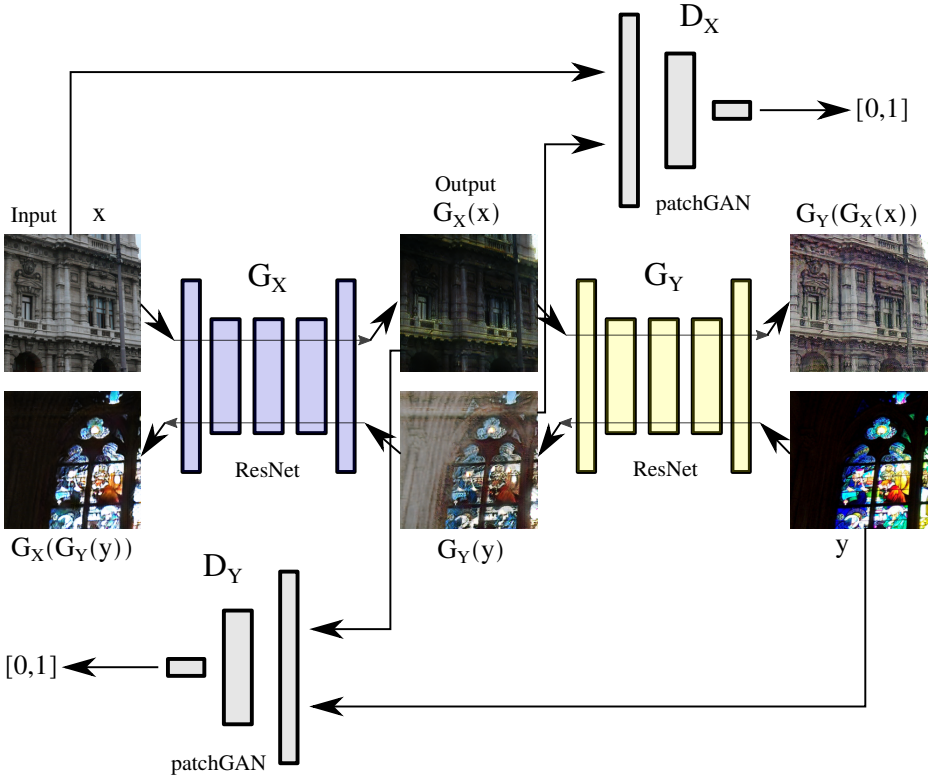


Figure 2.4: The architecture of CycleGAN. The first ResNet generator G_X tries to transform a real day image x into fake night image $G_X(x)$ (blue) and the second ResNet generator G_Y tries to transform a real night image y into fake day image $G_Y(y)$ (yellow). The first patchGAN discriminator D_X tries to classify if night data samples x or $G_Y(y)$ is real or fake, respectively, and the second patchGAN discriminator D_Y tries to classify if day data samples y or $G_X(x)$ is real or fake, respectively.

Formulation

I form the CycleGAN learning very similarly to Zhu et al. [4]. Having two domains X and Y, the task of CycleGAN is to find a double-sided mapping between X and Y, given training data images $\{x_i\}_{i=1}^{N}$ and $\{y_j\}_{j=1}^{M}$ where

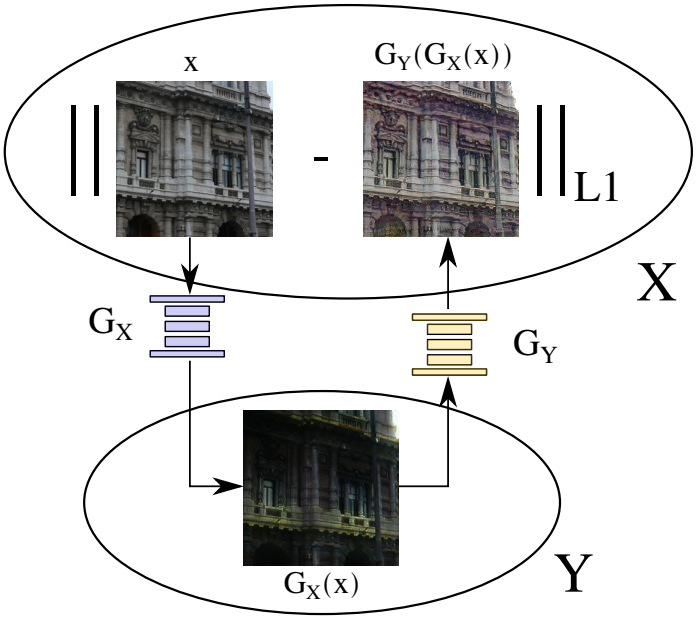


Figure 2.5: Cycle consistency loss. Day image x from day visual domain X pass through the generator G_X translating into fake night image $G_X(x)$ in night domain Y, and this image pass again through the generator G_Y translating into fake reconstructed day image $G_Y(G_X(x))$. Cycle consistecy loss measures, how much are original image x and reconstructed image $G_Y(G_X(x))$ different. CycleGAN training optimization minimizes this loss as a regularization to the adversarial loss.

$x_i \in X$, $y_j \in Y$ are not corresponding data pairs[3], so training data do not provide pixel-aligned pairs as in the pix2pix formulation 2.3.2.

The CycleGAN model has 4 networks in total. Two generators, where $G_X : X \to Y$ and $G_Y : Y \to X$, provide the double-sided mapping between X and Y, and two discriminators, where $D_X : X \to [0, 1]$ discriminates between x and $G_Y(y)$ and $D_Y : Y \to [0, 1]$ discriminates between y and $G_X(x)$, trying to distinguish between real and fake images, see Figure 2.4 [4].

Loss

The loss function of CycleGAN consists of adversarial loss and cycle consistency loss.

The adversarial loss is the same adversarial loss defined by Goodfellow et al. for GANs [1]. The adversarial loss for generator G_X and its discriminator D_Y is expressed as

$$\begin{aligned}\mathcal{L}_{GAN}(D_Y, G_X, X, Y) = \mathbb{E}_{y \sim p_Y}[\log D_Y(y)] \\ + \mathbb{E}_{x \sim p_X}[\log(1 - D_Y(G_X(x)))].\end{aligned} \tag{2.7}$$

From 2.7, we can see $G_X : X \to Y$ generate images $G_X(x)$ similar to images from domain Y, while $D_Y : Y \to [0, 1]$ tries to classify real image y and fake

[3] For simplicity, i and j are ommited

image $G_X(x)$. G_X aims to minimize this loss, D_Y aims to maximize this loss. For the generator G_Y and the discriminator D_X, the adversarial loss $\mathcal{L}_{GAN}(D_X, G_Y, Y, X)$ is formed similarly.

The cycle consistency is motivated by the possibility that generators can learn a mapping in which one single inputted image translates into any image of the target domain, resulting in learning to generate random permutations of the target domain [4]. This issue is fixed with **cycle consistency loss**, which is defined as the L1 distance between the image of the input domain and the reconstructed image from the target domain [4]. For example, for the pass $X \to Y$, the cycle consitency constraint ensures that for i-th image in X, $x_i \xrightarrow{G_X} y_i' \xrightarrow{G_Y} x_i' \sim x_i$, informally, see Figure 2.5. The cycle consistency loss is defined as

$$\begin{aligned}\mathcal{L}_{cyc}(G_X, G_Y) = \mathbb{E}_{x\sim p_X}[||x - G_Y(G_X(x))||_1] \\ + \mathbb{E}_{y\sim p_Y}[||y - G_X(G_Y(y))||_1].\end{aligned} \tag{2.8}$$

The resulting optimization task is

$$\begin{aligned}\min_{G_X,G_Y}\max_{D_X,D_Y} \mathcal{L}_{GAN}(D_Y, G_X, X, Y) + \mathcal{L}_{GAN}(D_X, G_Y, Y, X) \\ + \lambda\mathcal{L}_{cyc}(G_X, G_Y).\end{aligned} \tag{2.9}$$

2.3.4 Convergence

Finding equilibrium between G and D is a very difficult problem. Gradient descent numerical methods for GANs are likely to fail. Specifically, from the optimization task 2.1, assuming D is not linear or affine function, we can see that the value function is non-convex, because $D(x)$ and $-D(G(z))$ are opposite functions that cannot be both convex. Therefore, finding the minimum of the general non-convex function with continuous high-dimensional parameters is a difficult problem.

Mode collapse

The generator can fall into **mode collapse** (network falls into parameter setting when it returns the same output for arbitrary input), where the gradient of the generator loss function approaches zero w.r.t. input noise z resulting in empty backpropagation [15].

In pix2pix training, since the network is trained with paired source and target domain images, it is less likely the pix2pix generator falls into mode collapse.

In CycleGAN training, when cycle consistency loss is removed or it is removed in one direction, CycleGAN training can fall into mode collapse [4].

Model oscillation

Both the generator and the discriminator loss functions can oscillate simultaneously, so one network finds parameters that highly drop its loss and

increase the loss of the second network [16]. To prevent this, Salimans et al. [15] proposed a heuristics called *historical averaging*, where L2 cost $||\theta - \frac{1}{t}\sum_{i=1}^{t}\theta_i||_2^2$ between the current network parameters θ and average $t \in \mathbb{N}$ network parameters θ_i is added to each network loss function.

In CycleGAN training, an alternative technique of Shrivastava et al. [17] is utilized. In this technique, the discriminator is not updated from the last output of the generator, but it is updated with one of the last 50 generated images by probability 0.5 and with the last generator output by 0.5 probability [4].

2.4 Related Work

Similarly to this work, Arruda et al. [18] explores domain adaptation from day to night visual domains using unsupervised (unpaired) image-to-image translation. They solve cross-domain (day-night) car detection problem having annotated day training image samples and unannotated night images with data augmentation, first by learning image-to-image translator to transform day images into night images. Then they transform day annotated samples into fake night keeping annotation, resulting in car detector learned from both day and night annotated samples.

An alternative solution to the day visual domain adaptation was introduced by Annosleh et al. [8]. The proposed ToDayGAN is a modified unpaired image-to-image translation model of ComboGAN [19] having the same generator architecture as CycleGAN [4], but modified discriminator architectures, to improve localization. First, the image-to-image translator is learned to translate night images into day images. Then both reference and translated images are used for featurization to obtain feature vector per image, where the query is estimated by the nearest neighbor of the day image. Notice, the solution to the night data insufficiency is proposed in the reverse way to this work, ToDayGAN is not trained to augment training data with night samples, but to adapt queries from night into day visual domain.

Related to CNN image retrieval performance under illumination-invariant conditions, Jenicek & Chum [7] proposed a photometric normalization U-Net network, which translates any image into domain less sensitive to illumination changes. This work builds on its multi-domain image retrieval codebase and uses the same image retrieval evaluation protocol to test data augmentation performance.

Another models, close this work, are the group of derived GAN models designed for unsupervised image-to-image translation. ComboGAN [19] provide unpaired translation between multiple visual domains having N generators and N discriminators for N visual domains, where the generators are encoder-decoder networks able to encode an image in one domain into a feature vector, which is decoded with any other decoder into its respective domain. Also, StarGAN [20] performs unpaired multi-domain image-to-image translation using only one generator and one discriminator networks, where the discriminator, although it learns to distinguish between real and fake

images, also classifies domain of the real image, and the generator is learned to output fake images given input image and target domain label, however, StarGAN was only applied to CelebA [21] and RaFD [22] datasets for face attributes modification having only slight shifts between visual domains (eg. happy, blonde hair, aged, etc.).

Chapter 3

Implementation

In Section 3.1, all used datasets with their preprocessing are described. In Section 3.2, hyperparameter settings and used architectures in pix2pix and CycleGAN are briefly described. In Section 3.3, image retrieval network setting, used for visual domain adaptation test, is described.

3.1 Datasets

There are 4 image datasets I found suitable for image-to-image translation in day-night training.

Transient Attributes Database (abbreviated as *Transattr*) is an image dataset containing 8571 images from 101 webcams with 40 attribute annotations [23]. This is the only dataset that can be processed into paired training data for paired image-to-image translation with pix2pix. In order to obtain sets of day and night images from 40 attributes, I first selected image groups of those webcams, which have at least one image with the *night* attribute greater than 0.9 (69 webcams satisfies this condition), and made day-night pairs as the cartesian product of images with *night* attribute greater than 0.9 and the remaining images from the webcam group, resulting in pairs composed of 4774 day and 310 night unique images.

Aachen Day-Night Dataset (abbreviated as *Aachen*) contains images of the old inner city of Aachen in Germany [24, 25]. Dataset has 5313 day images and only 113 night images taken with mobile phones with HDR setting at nighttime.

RobotCar Seasons Dataset (abbreviated as *Robotcar*) have images captured from 3 cameras, mounted on a vehicle, taken under different conditions at the 49 city locations [24, 26]. For the CycleGAN training, I only use rear camera images. Images taken under conditions of *sun*, *snow*, *rain*, *dawn* and *dusk* are used for day, and images under *night* and *night-rain* conditions are used for night domain. I chose all these domains, because one single domain has 400 images in average, which is little for GAN training in practice. In total, this results in 2247 day and 878 night domain images.

Retrieval-SfM [5] contains 146714 day and 16957 night images recognized from day-night annotations [27]. Some images have height or width less than 256 px, therefore, I removed images having any dimension less than 512 px

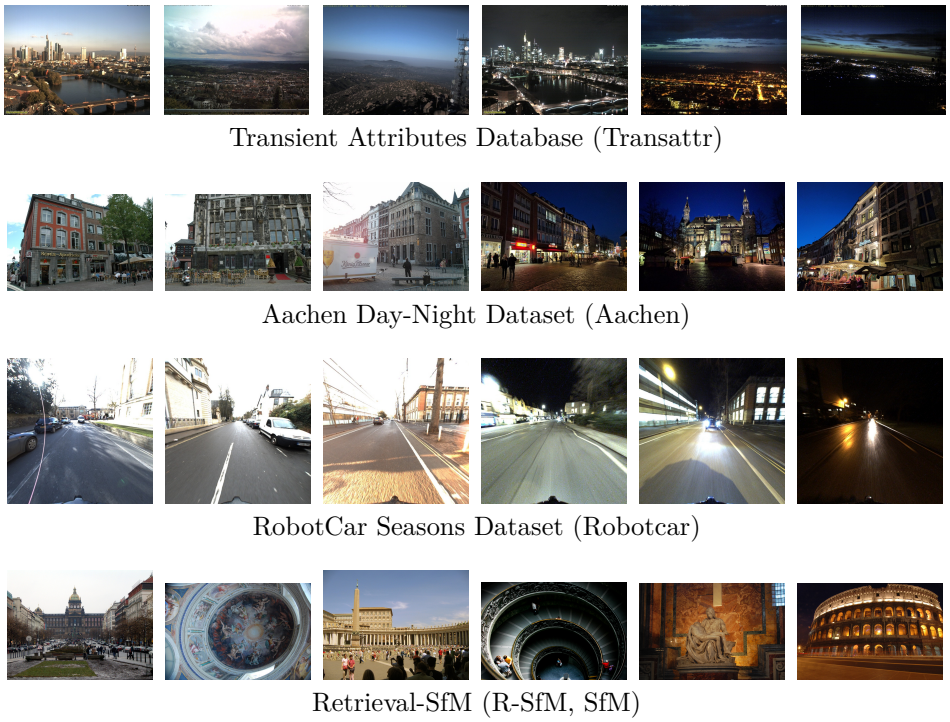


Figure 3.1: Training data sample images for image-to-image translation. Images are ordered by corresponding dataset and visual domain. Row order from top to bottom corresponds to Transattr, Aachen, Robotcar, Retrieval-SfM datasets. In each row first three left images correspond to the day visual domain and last three right images correspond to the night visual domain. Details are best viewed on a computer screen.

resulting in 118910 day, 13710 night images. Moreover, I manually observed ambiguities, where some of the images are difficult to visually sort out into day or night domain, and few images have label mistakes.

3.2 GAN Training

Training details and network hyperparameter settings, used in this work, are briefly described.

All three network architectures, U-net generator [3], Resnet generator [4], and PatchGAN discriminator [3, 4], are used in the same architecture as described in [3, 4].

3.2.1 Training Details

As Goodfellow suggests [1], early in the learning expressed in 2.1, the generator makes poor fake samples easy to be distinguished from training samples. I train the generator to maximize while the discriminator minimizes adversarial loss. The resulting adversarial task for fake image recognition

becomes as minimization of $\mathbb{E}_{x\sim p_X}[\log(D(G(x)))]$ instead of maximization of $\mathbb{E}_{x\sim p_X}[\log(1 - D(G(x)))]$ from the discriminator, which provides stronger gradients early in the learning [1]. The total loss for the generator is the sum of all its losses, while the total loss for the discriminator is the average of real and fake image losses. In the loss functions, I use weights (multipliers) $\lambda = 100$ for L1 regularization loss in the pix2pix Equation 2.6 and $\lambda = 10$ for cycle consistency loss in the CycleGAN Equation 2.9.

For both network optimization, I use Adam solver [28] with learning rate 0.0002, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and zero learning decay.

In data preprocessing, each batch of size 5 is augmented such that it is randomly scaled from 0.6 to 1 from the original size[1] and cropped to have the size of 256x256 px, and finally, it is normalized to mean and deviation of 0.5 in all 3 channels.

I implemented pix2pix and CycleGAN training in Multi-Domain Image Retrieval[2] codebase. Specifically, I added the learning procedures in epoch iteration, and wrote training and output scenarios.

3.2.2 Pix2pix Architecture

I borrow discriminator and generator architectures from Isola et al. [3]. Their network architectures are mostly adapted from architectures of Deep Convolutional GAN (DCGAN) [29].

The generator is the U-Net based generator implemented by [3]. I trained generators both with and without dropout with 0.5 probability.

As a discriminator, patchGAN discriminator, implemented by [3], is used.

3.2.3 CycleGAN Architecture

Again, I borrow discriminator and generator architectures from Zhu et al. [4].

The generator is ResNet-based network adopted from the neural style transfer framework of Johnson et al. [30]. For image-to-image translation between day and night domains, I use 9 resnet blocks. I trained generators both with and without dropout with 0.5 probability.

The discriminator network is the same patchGAN discriminator, used in pix2pix architecture (Section 3.2.2).

3.2.4 Loss Weights Normalization

Regularization weights in total loss settings for pix2pix in Equation 2.6 and CycleGAN in Equation 2.9 are high, e.g. $\lambda = 100$ for L1 regularization in pix2pix, which causes generator and discriminator learning rate imbalance, since weighted regularized loss derivative is λ-times higher, than without its

[1] Except for the robotcar dataset, which is always scaled to size 256x256 px with no cropping.

[2] Image retrieval codebase and implementation progress is available online at CTU Gitlab at https://gitlab.fel.cvut.cz/jenicto2/mdir

weight. The solution is the weight normalization, which divides each weight loss by the sum of all weights, so the resulting pix2pix loss function would be

$$\mathcal{L}(D,G) = \lambda_1 \mathcal{L}_{cGAN}(D,G) + \lambda_2 \mathcal{L}_{L1}(G), \text{ where } \lambda_1 = \frac{1}{1+\lambda}, \lambda_2 = \frac{\lambda}{1+\lambda}, \tag{3.1}$$

the normalized loss for the CycleGAN is similar to 3.1. I train pix2pix and CycleGAN with both options – with and without weight normalization.

During GAN evaluation, weight loss normalization does not change GAN performance significantly. I consider this more as an implementation option than optimization hyperparameter.

3.3 Image Retrieval Training

The task is to train an embedding network in an unsupervised manner, that provides the mapping from inputted images into image descriptors.

3.3.1 CNN Image Retrieval Architecture

The embedding network is the CNN image retrieval network, following the procedure of Radenovic et al. [31, 5], see embedding network in Figure 3.2.

Convolutional layers are taken from the pretrained VGG16 network [32], where the last fully connected layers are removed. Training the embedding network is specifically the task of **fine-tunning** i.e. taking a network pretrained for one task and then training it for a different task.

Instead of removed layers of the truncated VGG16, GeM pooling is appended. GeM layer takes K feature maps of the tensor $\mathcal{X}$ with $W \times H \times K$ dimensions, outputted by VGG16, and produces a single vector $\mathbf{f}$:

$$\mathbf{f} = [\mathsf{f}_1 \dots \mathsf{f}_k \dots \mathsf{f}_K]^T, \text{where } \mathsf{f}_k = \left(\frac{1}{|\mathcal{X}_k|} \sum_{x \in \mathcal{X}_k} x^{p_k} \right)^{\frac{1}{p_k}}, \tag{3.2}$$

where p_k is the pooling parameter, which is learned with backpropagation [5]. GeM output is normalized by the $L2$ normalization layer [5].

For training the embedding network, contrastive loss [33] is used to provide feedback. Training aims to minimize the contrastive loss of inputted image pair, resulting in an embedding network outputting similar image descriptors for *matching*[3] pairs and different image descriptors for *non-matching* pairs [5].

For each trained network, whitening of image descriptors is learned as a post-processing step [5].

[3] Image pair is *matching* (*positive*), if both images capture the same object or scene. Image pair is *non-matching* (*negative*), of both images are taken far from each other [5].

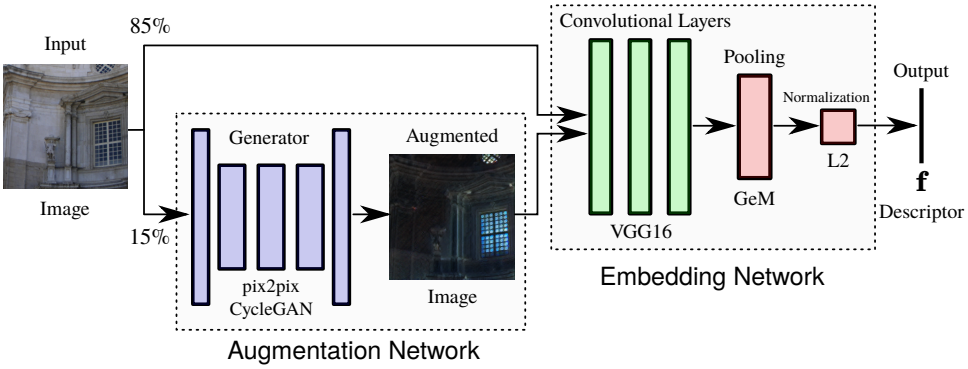


Figure 3.2: The setting of augmented and embedding networks. An input image is translated with the augmentations network into night image by 0.15 probability or directly skips the augmentation network otherwise. Then, the image descriptor **f**, representing the input image, is extracted from the input image by the embedding network. In fine-tunning the embedding network, the truncated part is denoted with green color, added parts are denoted with red color, compared to original VGG16.

3.4 Data Augmentation

Once the image translation network is prepared, it is used to augment the inputted data during the training of the embedding network. Data augmentation is implemented as network sequence of two networks – augmenting and embedding – where first, the augmenting network translates input image into night image by 0.15 probability or sends the original image forward otherwise, and second, the embedding network takes the image from augmenting network and performs regular embedding training step. Specifically, each iteration of the embedding network takes 7 specific images[4], and each image is transformed by the probability of 0.15, which is close to $\frac{1}{7}$. During backpropagation, the augmentation network is frozen. See Figure 3.2.

During image retrieval fine-tunning, input images have the size of 362x362 px, in order to use the same size as Radenovic et al. [5]. However, the pix2pix generator architecture is U-Net based, and therefore it can only process images, having dimensions divisible by 256. I deal with this issue the same way as Jenicek & Chum [7]; before each fine-tunning iteration, images are padded up to the first possible dimension divisible by 256 maintaining the contextual information with reflection padding. After the image translation, images are sliced down resulting in the augmented images having the same dimensions as the images before reflection pad preprocessing.

[4]One image is the query, one image is *positive*, and the remaining 5 images are *negative*.

Chapter 4

Evaluation

GAN evaluation is hard because the generator objective is changing each iteration and image quality is subjective to define. Generators do not use an objective function, but a discriminator, instead of being trained directly, making them difficult to compare them with each other [15]. Human visual examination of generated samples is the simplest, but slow and expensive way to evaluate GANs [34] since generated image quality definition is subjective and not defined mathematically. Yet, none of the proposed metrics is agreed to be the one common GAN benchmark, used for generator models comparison, capturing all their strengths and weaknesses [34].

In Section 4.1, appropriateness of the most common GAN evaluation metrics is discussed.

4.1 GAN Evaluation Scores

Given generated image samples from pix2pix or CycleGAN, from [34] a high-quality evaluation score for image retrieval should:

- correlate with image retrieval performance,
- measure sample diversity, be sensitive to mode collapse and overfitting,
- be transformation invariant (score should not change if the semantic meaning of the image do not change),
- have well-defined lower and upper bounds,
- have low computational complexity.

Note, there is no need to correlate with human judgment or to be discriminative since the generated data are used for image retrieval data augmentation.

4.1.1 Structural Similarity (SSIM)

Traditional metrics do not match well with visual image quality. For example, L2 distance measure average differences between pixel intensities. However, humans perceive images more structurally e.g. when there is a soft pixel Gaussian noise between the two same images, L2 distance can be still low, but a human can see those images as different.

SSIM was defined to measure the similarity between two images more precisely than traditional simple metrics [35]. Let x and y be images of

our similarity interest. The SSIM index measure is a combination of three components:

$$\text{SSIM}(x, y) = [l(x, y)]^{\alpha} \cdot [c(x, y)]^{\beta} \cdot [s(x, y)]^{\gamma}, \tag{4.1}$$

where between images x and y: l denotes luminance, c denotes contrast and s denotes structure[1] and $\alpha > 0$, $\beta > 0$, $\gamma > 0$ are parameter weights used to adjust relative importance of these components. By default, the weights are $\alpha = \beta = \gamma = 1$. When this applies, SSIM can be expressed as

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{4.2}$$

where $\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$ and μ_y are means of x and y, $\sigma_x^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)$ and σ_y^2 are variances of x and y, $\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y)$ is the covariance of x and y, $C_1 = (k_1 R)^2$, $C_2 = (k_2 R)^2$, where R is the difference between minimum and maximum possible pixel value and $k_1 \ll 1$, $k_2 \ll 1$ are parameters which are $k_1 = 0.01$, $k_2 = 0.03$ by default.[2]

SSIM ranges between 0 and 1, a high value of SSIM corresponds to perceptually more similar images. SSIM is a symmetric function: $\text{SSIM}(x, y) = \text{SSIM}(y, x)$. SSIM does not hold the identity of indiscernibles, but holds the unique maximum property: $\text{SSIM}(x, y) = 1 \iff x = y$. SSIM does not hold the triangular inequality. However, under certain conditions, SSIM can be converted into a normalized metric [36]. In addition, SSIM is boundedness, satisfying: $\text{SSIM}(x, y) \leq 1$.

Multi-scale Structural Similarity (MS-SSIM)

SSIM index evaluates two images on a single scale which could be inaccurate because the correct scale depends on viewing conditions such as view distance or image resolution [37].

Having the general SSIM definition 4.1, MS-SSIM, proposed in 2003 by Wang et al. [37], calculates with $M \in \mathbb{N}$ multiple scales. Images x and y are iteratively processed with two steps. At the first step, contrast and structure are calculated for $1, \ldots, M-1$ iterations, and complete SSIM is calculated in the M-th iteration. At the second step, x and y are downsampled by 2D average-pooling with kernel size 2, preparing the downsampled x and y are used the next iteration. The overall MS-SSIM evaluation can be expressed as

$$\text{MS-SSIM}(x, y) = [l_M(x, y)]^{\alpha_M} \cdot \prod_{i=1}^{M} [c_i(x, y)]^{\beta_i} [s_i(x, y)]^{\gamma_i}, \tag{4.3}$$

where, similarly to 4.1, α_M, β_i and γ_i are parameter weights used for luminance, constrast and structure relative importance adjustment. In practice,

[1] For simplicity, I do not define these functions. Proper definitions and more detailed explanation of SSIM can be found in [35].

[2] In this variant, SSIM is implemented in scikit-image python library: https://scikit-image.org/docs/dev/api/skimage.metrics.html#skimage.metrics.structural_similarity

weights are simplified as in SSIM 4.1.1 for all $i = 1, \ldots, M : \alpha_i = \beta_i = \gamma_i = 1$. Also, the default setting for scales are $[0.0448, 0.2856, 0.3001, 0.2363, 0.1333]$ meaning, at i-th iteration luminance, contrast and structure are powered by the i-th scale.

MS-SSIM was also abused by Odena et al. [38] to evaluate the diversity of generated images within one class, where the mean MS-SSIM close to 1 indicates that randomly chosen image pairs have low diversity, possibly pointing to the mode collapse, and mean MS-SSIM close to 0 indicates generated images have high diversity [34].

For the pix2pix evaluation, MS-SSIM has very promising properties. Calculating MS-SSIM between fake images and target images estimates information about how precisely did pix2pix learn to generate fake images. However, MS-SSIM measures human judgment correlation rather than image retrieval performance correlation.

In CycleGAN evaluation, a single generator alone cannot be evaluated without the second one. Therefore, when one generator fails, FID increases, but it is not known which generator failed. Target images are not available, and therefore MS-SSIM can be only used to evaluate cycle-consistency, which evaluates both generators at once without possibility to evaluate the single one only, which makes it inappropriate measure to the CycleGAN ability to generate night images.

4.1.2 Inception Score (IS)

IS, proposed by Salimans et al. [15], is the first, often used GAN metric highly correlating with human annotators (Amazon Mechanical Turk), judging the visual quality of images, aiming to measure image quality and diversity of images.

Obtaining IS for evaluation of a generator G which produces images x following distribution p_G is defined as:

$$\mathrm{IS}(G) = \exp(\mathbb{E}_{x \sim p_G} KL(p(k|x)||p(k))), \tag{4.4}$$

where $p(k|x)$ is the conditional class distribution indicating the probability an inception network assigns to each class label $k \in [0, 1]^{1000}$ given image x, $p(k) = \int_x p(k|x)p_G(x)$ is the marginal class distribution and KL is the Kullback–Leibler divergence, which comes out as

$$KL(p(k|x)||p(k)) = \sum_{i=1}^{1000} p(k_i|x)(\log(p(k_i|x)) - \log(p(k_i))), \tag{4.5}$$

after substitution with conditional and marginal distributions [15]. When inputted images have high IS, a generator outputs high quality and diverse images.

The inception network, which calculates the $p(k|x)$, is the Inception v3 Network [39], designed to classify images from the ImageNet, from an image dataset having 1000 classes containing 1.2 million images [40].

Properties of IS are that its possible values are bound to $1 \leq \mathrm{IS}(G) \leq 1000$; IS is high if the conditional label distribution has low entropy, as images classified strongly as one class over other 1000 indicate high image quality, also IS is high if the marginal label distribution is high, indicating high diversity among generated images [41].

However, when the generator falls into mode collapse, IS could be still high, resulting in an inappropriate evaluation metric for CycleGAN [41]. Moreover, datasets for image retrieval contains similar images very often, and since IS is dependent on marginal probability estimation containing classification over 1000 classes for $p(k)$ estimation, $p(k)$ can result as imbalanced distribution having only a few high probability classes among 1000, which raises doubts about correct $p(k)$ estimation.

Although mode collapse detection inability can be solved with modified *mode score* [42], concerning the problem with estimated distribution imbalance for image retrieval data, IS is not appropriate to be used for pix2pix or CycleGAN evaluation.

4.1.3 Fréchet Inception Distance (FID)

In 2017, Heusel et al. [43] proposed another score metrics dealing with mode collapse with comparing statistics between real and generated samples as an improvement to the IS. Similarly to IS, the Inception v3 module is used to measure image features [39], specifically the last pooling layer before the output classification layer outputs 2048 activations [43].

Assuming, feature vectors follow a multi-dimensional Gaussian distribution with parameters mean $\mu \in \mathbb{R}^{2048}$ and covariance $\Sigma \in \mathbb{R}^{2048 \times 2048}$, FID is calculated as the Fréchet distance [44] between real sample features Gaussian x and fake sample features y Gaussian

$$\mathrm{FID}(x, y) = ||\mu_x - \mu_y||_2^2 + \mathrm{tr}(\Sigma_x + \Sigma_y - 2(\Sigma_x \Sigma_y)^{\frac{1}{2}}), \tag{4.6}$$

where μ_x and μ_y are the means of real features and fake features, respectivelly, Σ_x and Σ_y are covariance matrixes of real features and fake features, respectivelly, tr is the matrix trace [43]. FID is also known as Wasserstein-2 distance [45]. FID indicates well-generated images with low distance.

For pix2pix and CycleGAN evaluation, FID offers a wide variety of desired properties among other evaluation metrics proposed until 2018, such as high discriminability, invariacy to image transformations, low computational complexity [34]. Apart from IS, FID does not use distributions made from 1000 classifications, but feature vectors making it similar to image retrieval embedding networks.

However, FID is not the best sensitive metric to overfitting and mode collapse [34]. To address sample diversity, a manual examination of the best performing GANs having the highest FID score using their PRD curves results in decent pix2pix and CycleGAN evaluation.

4.1.4 Precision and Recall for Distributions (PRD)

For state-of-the-art networks, a one-dimensional score is not providing information about how much is a generator failing in different cases. In 2018, during an internship at Google, Sajjadi et al. [46] proposed a novel definition for precision and recall function which returns a two-dimensional score for two distributions.

PRD indicates how much are two distributions intersected and disjointed. More formally, let $\alpha \in (0, 1]$ be denoted as the precision, $\beta \in (0, 1]$ be denoted as the recall and let P and Q be probability distributions. If there exists intersection μ of P and Q, relative complement ν_P indicating what part of P is missed by Q and relative complement ν_Q indicating what part of Q is missed by P, then

$$\mathrm{PRD}(Q, P) = \{(\alpha, \beta); P = \beta\mu + (1-\beta)\nu_P, Q = \alpha\mu + (1-\alpha)\nu_Q\} \cup \{(0, 0)\}, \tag{4.7}$$

is PRD of the distribution Q w.r.t. distribution P [46].

In practice, the image range is unknown, computing PRD from definition 4.7 is cumbersome, since the existence of suitable μ, ν_P and ν_Q must be searched for each α and β. Sajjadi et al. introduced another task to compute PRD, equivalent to task 4.7:

$$\begin{aligned} \mathrm{PRD}(Q, P) &= \{(\alpha(\lambda), \beta(\lambda)); \lambda \in \Lambda\}, \text{ where} \\ \alpha(\lambda) &= \sum_{\omega \in \Omega} \min(\lambda P(\omega), Q(\omega)) \\ \beta(\lambda) &= \sum_{\omega \in \Omega} \min\left(P(\omega), \frac{Q(\omega)}{\lambda}\right) \\ \Lambda &= \left\{\tan\left(\frac{i}{m+1}\frac{\pi}{2}\right); i = 1, 2, \ldots, m\right\}, \end{aligned} \tag{4.8}$$

where m is the number of angles, $m = 101$ by default, Ω is the finite state space which is implemented as cluster distribution calculated using minibatch k-means [47] of $P \cup Q$ [46]. PRD is calculated from real images and fake images by measuring feature vectors of images with Inception v3 Network [39] using the last pooling layer the same way as in FID measurement [43], to obtain distributions P and Q [46]. Then the union of P and Q is clustered, with $k = 20$ by default, using mini-batch k-means to obtain two histograms indicating, how many features from distributions P and Q fall into the corresponding cluster, and because k-means have randomized initialization, histograms are calculated 10 times and PRD curves are averaged [46].

In the CycleGAN evaluation, PRD can provide more specific insights on generated images scoring on FID, since FID is expressed in a single number, especially capturing if the generator is overfitting or fall into mode collapse, making PRD and FID together a strong evaluation score.

However, PRD scores are difficult and clumsy to automatically compare, because comparing them results in a single values comparison, e.g. with their integral computation, it turns back PRD score from 2-dimensions into 1-dimension score with cumbersome, imprecise way.

Chapter 5

Results

To test image retrieval performance, three datasets – Revisitied Oxford [48], Revisited Paris [48] and 24/7 Tokyo [49] – are used to evaluate embedding networks. 24/7 Tokyo contains images of day, night, and sunset lighting conditions, while Revisited Oxford and Paris contain common daytime images. The evaluation protocol with accordance to Jenicek & Chum [7] is used.

The Mean Average Precision (mAP) is used to measure and compare image retrieval quality. For better clarity, the score is multiplied with 100, so the possible mAP ranges are in $[0, 100]$, where high values indicate high retrieval quality.

The baseline VGG16 GeM [9] is compared against the augmented VGG16 GeM networks, trained with different data augmentation GANs each across four GAN training datasets. Each GAN training dataset is one of the testing categories, with Transattr dataset providing paired images, so augmentation can be tested both with CycleGAN and pix2pix resulting in a total of 5 testing categories. Before the image retrieval test, based on the GAN evaluation, settings with well-performing FID and PRD were chosen, to train augmentation networks, each category three times with different seeds under the same settings. For each dataset category, loss weight normalization, and their most combinations with dropout were examined. Also, for few image datasets, different learning rates and regularization hyperparameters were tried, but default values used by Isola, Zhu, et al. [3, 4] perform the best, and are used as described in 3.2.1. Depending on the best validation FID and PRD scores, the following data augmentation GANs were chosen:

- CycleGAN, Aachen, unnormalized weights, without dropout,
- CycleGAN, Robotcar, normalized weights, without dropout,
- CycleGAN, R-SfM, normalized weights, without dropout,
- CycleGAN, Transattr, normalized weights, without dropout
- Pix2pix, Transattr, normalized weights, with dropout.

To train GANs on datasets equally, the number of epochs is approximately set to correspond the number of iterations during the training. For training on Aachen, Robotcar, R-SfM and Transattr datasets, 150, 200, 150 and 100 epochs, respectively, are trained with. In the Transattr dataset, one training epoch has approximately 4500 iterations.

The resulting mAP is reported as the average of 3 embedding networks each trained with one augmentation network with the different seed.

Embedding Network / Augmentation Network	FID SfM	Tokyo [mAP]	ROxf [mAP]	RParis [mAP]	Avg [mAP]
Baseline VGG16 GeM	–	79.31	**60.73**	69.10	52.29
CycleGAN, Aachen,	89.67	**88.49**	58.26	68.46	**53.80**
CycleGAN, Robotcar	189.70	81.50	57.00	67.60	51.53
CycleGAN, R-SfM	**81.63**	**86.61**	59.77	**69.17**	**53.89**
CycleGAN, Transattr	**56.64**	82.21	**59.86**	**69.24**	52.83
Pix2pix, Transattr	144.53	82.31	56.34	68.07	51.93

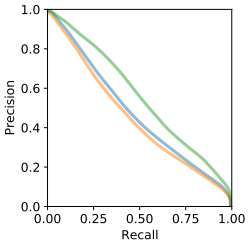
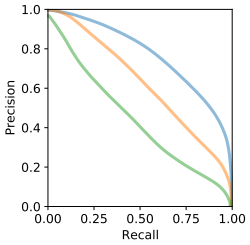
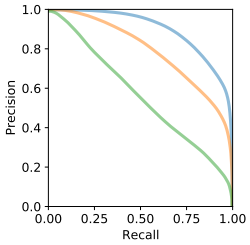
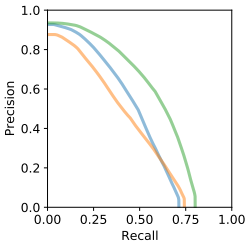
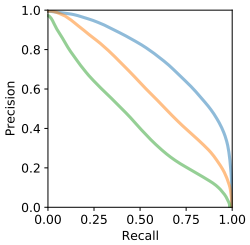
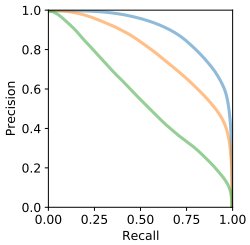
Table 5.1: Performance comparison with and without data augmentation. Results are composed of two parts: image-to-image translation evaluation with FID on Retrieval-SfM and image retrieval evaluation on 24/7 Tokyo [49], Revisited Oxford [48] and Revisited Paris [48]. Image retrieval performance is expressed in mean average precision (mAP) and multiplied by 100 for better readability. Training of CycleGAN SfM is reported with weight loss normalization, which have higher Tokyo and average mAP than unnormalized case. The best results are highlighted **bold red**, the second best results are **bold** and the worst results are blue.

5.1 Discussion

Embedding network results show data augmentation trained with CycleGAN on R-SfM has the best retrieval results. Surprisingly, CycleGAN trained on Aachen has high performance on the 24/7 Tokyo dataset, although it often generates fake images classified as obviously fake with humans. Also, Transattr dataset used to train CycleGAN providing decent retrieval improvement in night visual domain with minimal performance loss (Oxford) and improvement (Paris) in the day domain, however, pix2pix performs suddenly worse. Robotcar is insufficient for general day to night image translation.

Image dataset diversity and quantity affect GAN performance in image-to-image translation mostly. Transattr dataset provides promising training images, where networks trained on it have very favorable results (see Fig 5.3, right column) compared to R-SfM used to train both GAN generator and embedding network. R-SfM is the only one dataset among other the four which has enough night images, however, it contains ambiguous images for day-night classification resulting in fake night images with yellow shade. Aachen dataset has very few night images, and therefore, the generator can sometimes learn to make fake night images with buildings unnaturally lighted. Robotcar dataset contains a lot of very similar images among each other resulting worse image translation performance.

Lastly, image retrieval tests show different results than GAN score results. FID and PRD scores do not correlate with resulting image retrieval mAP measures. Also, not always the volume under PRD curve visually correlates with its corresponding FID (see Figure 5.1, CycleGAN Aachen, R-SfM versus CycleGAN, Transattr, *data*).

CycleGAN, Aachen	CycleGAN, R-SfM, normalized	CycleGAN, R-SfM, unnormalized
		
$\text{FID}_{\text{S-RfM}} = 89.67$	$\text{FID}_{\text{R-SfM}} = 81.63$	$\text{FID}_{\text{R-SfM}} = \mathbf{67.55}$
		
$\text{FID}_{data} = 184.42$	$\text{FID}_{data} = 81.63$	$\text{FID}_{data} = 67.55$

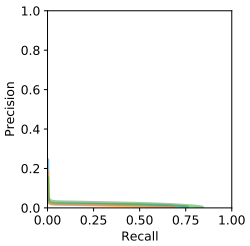
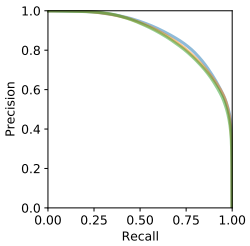
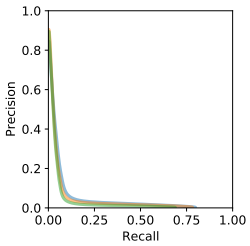
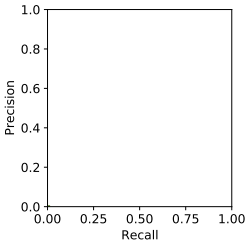
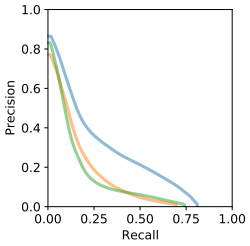
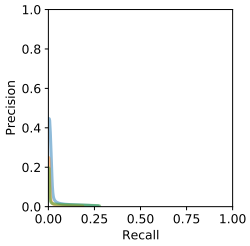
CycleGAN, Robotcar	CycleGAN, Transattr	Pix2pix, Transattr
		
$\text{FID}_{\text{R-SfM}} = 189.70$	$\text{FID}_{\text{R-SfM}} = \mathbf{56.64}$	$\text{FID}_{\text{R-SfM}} = 144.53$
		
$\text{FID}_{data} = 284.52$	$\text{FID}_{data} = 77.16$	$\text{FID}_{data} = 134.73$

Figure 5.1: The comparison of GAN evaluation results. For each network, the first plot shows PRD of original and fake images (generated with corresponding GAN) from R-SfM dataset, followed by corresponding FID, the second PRD plot and its corresponding FID show results from dataset, on which is the corresponding network trained. Each PRD plot shows precision (vertical axis) and recall (horizontal axis) of all three tested networks trained under different seeds.

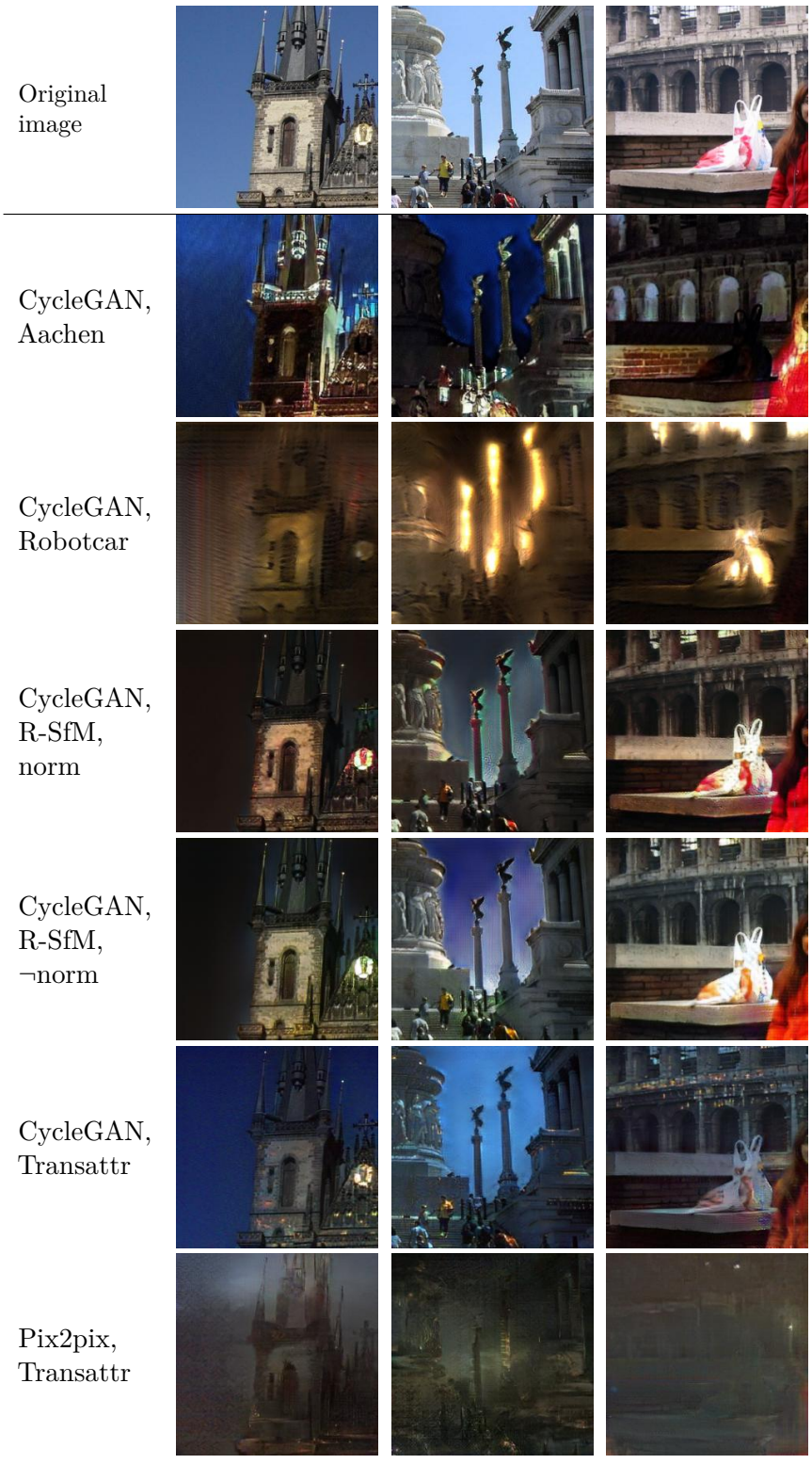


Figure 5.2: Translation comparison I. Day image from R-SfM dataset (top) is translated into night image with corresponding network trained on corresponding dataset (left). ¬norm denotes unnormalized weights.

Figure 5.3: Translation comparison II. Day image from R-SfM dataset (top) is translated into night image with corresponding network trained on corresponding dataset (left). ¬norm denotes unnormalized weights.

Chapter 6

Conclusions and Future Work

Difficult search queries from the night visual domain were addressed with training data augmentation using pix2pix and CycleGAN to adapt day into night visual domain. I implemented pix2pix and CycleGAN methods in Multi-Domain Image Retrieval codebase and trained these methods on 4 datasets to augment training data for image retrieval. I explored various GAN evaluation metrics, chose FID and PRD, and explained why other metrics were not used. I compared data augmentation performance of embedding networks augmented with GAN networks. Image dataset diversity and quantity affect GAN performance in image-to-image translation mostly.

For the following future work, training image database composition can greatly increase augmentation power. To improve GAN evaluation metrics for retrieval augmentation, the inception embedding network can be replaced with a network commonly used for image retrieval e.g. fine-tunned VGG16. To further improve the training of the embedding network with GAN data augmentation, targetted augmentation of query image or *positive* image can be tested.

Bibliography

[1] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[2] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent progress on generative adversarial networks (gans): A survey. *IEEE Access*, 7:36322–36333, 2019.

[3] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[4] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[5] F. Radenović, G. Tolias, and O. Chum. Fine-tuning CNN image retrieval with no human annotation. *TPAMI*, 2018.

[6] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, 124(2):237–254, 2017.

[7] Tomas Jenicek and Ondrej Chum. No fear of the dark: Image retrieval under varying illumination conditions. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[8] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. Night-to-day image translation for retrieval-based localization. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 5958–5964. IEEE, 2019.

[9] Filip Radenovic, Johannes L. Schonberger, Dinghuang Ji, Jan-Michael Frahm, Ondrej Chum, and Jiri Matas. From dusk till dawn: Modeling in the dark. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[11] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Deep shape matching. In *Proceedings of the european conference on computer vision (eccv)*, pages 751–767, 2018.

[12] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[13] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[15] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in neural information processing systems*, pages 2234–2242, 2016.

[16] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[17] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[18] Vinicius F Arruda, Thiago M Paixão, Rodrigo F Berriel, Alberto F De Souza, Claudine Badue, Nicu Sebe, and Thiago Oliveira-Santos. Cross-domain car detection using unsupervised image-to-image translation: From day to night. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[19] Asha Anoosheh, Eirikur Agustsson, Radu Timofte, and Luc Van Gool. Combogan: Unrestrained scalability for image domain translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 783–790, 2018.

[20] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for

multi-domain image-to-image translation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8789–8797, 2018.

[21] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[22] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. *Cognition and emotion*, 24(8):1377–1388, 2010.

[23] Pierre-Yves Laffont, Zhile Ren, Xiaofeng Tao, Chao Qian, and James Hays. Transient attributes for high-level understanding and editing of outdoor scenes. *ACM Transactions on Graphics (proceedings of SIGGRAPH)*, 33(4), 2014.

[24] Torsten Sattler, Will Maddern, Carl Toft, Akihiko Torii, Lars Hammarstrand, Erik Stenborg, Daniel Safari, Masatoshi Okutomi, Marc Pollefeys, Josef Sivic, Fredrik Kahl, and Tomas Pajdla. Benchmarking 6DOF Outdoor Visual Localization in Changing Conditions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[25] Torsten Sattler, Tobias Weyand, Bastian Leibe, and Leif Kobbelt. Image Retrieval for Image-Based Localization Revisited. In *British Machine Vision Conference (BMCV)*, 2012.

[26] Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36(1):3–15, 2017.

[27] Filip Radenovic, Johannes L Schonberger, Dinghuang Ji, Jan-Michael Frahm, Ondrej Chum, and Jiri Matas. From dusk till dawn: Modeling in the dark. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5488–5496, 2016.

[28] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[29] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[30] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[31] Filip Radenović, Giorgos Tolias, and Ondřej Chum. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*, 2016.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[34] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41–65, 2019.

[35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[36] Dominique Brunet, Edward R Vrscay, and Zhou Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, 2011.

[37] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[38] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org, 2017.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[40] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[41] Shane Barratt and Rishi Sharma. A note on the inception score. *arXiv preprint arXiv:1801.01973*, 2018.

[42] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode regularized generative adversarial networks. *arXiv preprint arXiv:1612.02136*, 2016.

[43] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in neural information processing systems*, pages 6626–6637, 2017.

[44] Maurice Fréchet. Sur la distance de deux lois de probabilité. *COMPTES RENDUS HEBDOMADAIRES DES SEANCES DE L ACADEMIE DES SCIENCES*, 244(6):689–692, 1957.

[45] Leonid Nisonovich Vaserstein. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, 5(3):64–72, 1969.

[46] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lučić, Olivier Bousquet, and Sylvain Gelly. Assessing Generative Models via Precision and Recall. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[47] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010.

[48] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondřej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5706–5715, 2018.

[49] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015.