3D scenario generation using generative models

Unsupervised virtual-to-real domain adaptation

Master's thesis in Complex Adaptive Systems

ANDREA RAMAZZINA

Department of Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

MASTER'S THESIS 2019:NN

3D scenario generation using generative models

Unsupervised virtual-to-real domain adaptation

ANDREA RAMAZZINA

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Computer Vision Group
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

3D scenario generation using generative models
Unsupervised virtual-to-real domain adaptation
ANDREA RAMAZZINA

Supervisors: Saudin Botonjic and Saurabh Gawande, Volvo Cars
Examiner: Fredrik Kahl, Department of Electrical Engineering

Master's Thesis 2019:NN
Department of Electrical Engineering
Computer Vision Group
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Sample synthetic image and corresponding refined output

Typeset in LaTeX
Gothenburg, Sweden 2019

3D scenario generation using generative models
Unsupervised virtual-to-real domain adaptation
ANDREA RAMAZZINA
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Virtual scenario is a promising approach to test the software in a fast and repetitive way with a much broader parameter space compared to using expensive and time consuming real data. However, there is a significant risk of the model not being able to generalize well enough on real images, since there is still a lot of gap between synthetic and real images, leading the model to learn only the features present in the synthetic data.

In this work we focus on the task of designing and experimenting with generative models able to transform a computer-generated image in a photo-realistic one, indistinguishable from a given dataset of real images, while preserving the semantic information of the underlying scene. The models used are based on the Generative Adversarial Network (GAN) framework, a particular sub-category of Neural Networks (NN).

We show that through the addition of novel objective functions and the modification of the model architecture it is possible to achieve state-of-the-art photo-realistic images which retain the semantic information of the synthetic input.
Furthermore, in order to evaluate the results in a quantitative and coherent manner, we also propose different evaluation metrics to assess the quality of the resulting generated images and examine how different metrics give different insights on the output quality.

Keywords: Machine Learning, Computer Vision, Artificial Intelligence, Neural Network, Unsupervised Learning, Generative Network, Image Transformation, Domain Translation

Acknowledgements

I would like to thank my supervisors Saudin Botonjic and Saurabh Gawande for giving me this opportunity and helping me throughout the thesis. Furthermore, I would like to thank professor Fredrik Kahl for his continuous availability and help, as well as Anders Ödblom for the interesting discussions and observations. I truly believe such interactions have played an important role in the shaping of this work and I have learned much from them.
Finally, I would like to thank my friends and family for the everlasting support.

Andrea Ramazzina, Gothenburg, June 2019

Contents

List of Figures

List of Tables

1
Introduction

Thanks to the recent advancements in both hardware and software technologies, machine learning (ML) solutions are starting to revolutionize and disrupt the automotive sector. Namely, autonomous driving and driver assistance capabilities are being developed and incorporated in a growing number of vehicles.
As one of the major vehicle company in Europe, Volvo Cars has been in the recent years deeply committed in the development of autonomous technologies, with particular regards to safety and reliability.

1.1 Background

In the active safety department at Volvo Cars, there is a lot of driving data being generated on regular basis via expeditions conducted by different teams in diverse environmental conditions. This data is a key point in the way software applications are developed and verified.
Furthermore, one of the main problem involved in the development of these technology is the ever-growing amount of data needed to develop and test ML algorithms. Gathering real-world data is usually time consuming and expensive.

Generating virtual scenarios based on parameterized real-life data might be a solution to this problem. By doing so, it is possible to test the software in a fast and repetitive way with a much broader parameter space.In fact,with the progress in computer graphics simulation platforms are increasingly improving in generating synthetic data for modelling purposes.

Nevertheless, using these synthetic images for training models poses a significant risk of the model not being able to generalize well enough on real images, since there is still a lot of gap between synthetic and real images, leading the model to learn only the features present in the synthetic data. This makes it paramount to reduce this 'gap' and generate synthetic images as realistic as possible.

One way of closing this gap is using content modelling and state of the art rendering algorithms, however these approaches can be extremely time consuming and computationally demanding. Moreover, even the latest rendering algorithms fail to account for the characteristics of real images bringing back the problem of generalization on real world data.
A relatively inexpensive and promising approach is Generative adversarial networks which have in practice demonstrated generating high quality realistic images[2] .

1.2 Problem description

The aim of this thesis is to study the applicability of the Generative Adversarial Network (GAN) framework to generate realistic images, with main focus on urban scenarios starting from samples generated by simulator environment such as the Unity 3D platform. Ideally, the generated synthetic data should be generated on demand/ quantity simulating various scenarios, especially related to Autonomous driving and safety and keeping the annotation-information of the synthetic images.

Figure 1.1: An example of virtual image extracted from a simulated environment (top image) and a corresponding desired outcome of the GAN (bottom image). Examples from the Virtual KITTI Dataset [4]

The main objective of this work is to refine the quality of synthetic images in an effort to bring them closer to real world data without changing the semantic content. This is achieved through the design of effective generative models,namely GANs, and novel extensions (architecture, loss functions etc.) to state of the art approaches.

Figure 1.1 shows an example of synthetic and real images couple which represent the same underlying scene . In this case, the semantic information of the real images has been extracted and the synthetic image is the product of a simulation environment using such information.

As a second step of this thesis, it is also of interest to propose and study different quantitative metrics to assess the quality of the refined images. In particular, the scope is to design quan-

titative measures to characterize both the the similarity to real data and its coherence with the starting synthetic scene.

1.2.1 Limitations

There are issues which will not be dealt during this work.
First, as the intended scope of this project is to study the feasibility and potential of these new approaches and technologies, achieving a fast and efficient solution-implementation is not going to be particularly taken in account. Hence, the proposed solutions will be mainly written in higher level programming languages (and tested on high-performance platform).

Furthermore, the focus of this work is on image-to-image transformations. This implies that, due to the chaotic nature of GANs architectures, the resulting transformation of a video sequence (obtained through single transformations of each frame) is likely not to be smooth and coherent. Since taking in account this fact would arise the complexity of the problem, this has not been taken in consideration and is left to further works.

Finally, only the single-to-single domain will be treated as generalizing to multi-domain approach would require a more complex framework. Additionally, such method would require additional datasets and tuning of different variables, which is time consuming and source of further issues.

1.2.2 Main contributions

The essence of this work can be summarized in three main contributions:

- Adaptation of current state-of-the-art models for image-to-image translation across multiple input modalities to suit the case in exam
- Proposal of a different objective function formulation able to yield a model capable of a more rigorous semantics preservation
- Introduction and comparison of different metrics to assess the quality of the refined image keeping in account the semantics preservation requirement

1.3 Thesis outline

This report is organized in five chapter. First, in the **Theory** chapter an holistic overview of the problem in exam is given, together with the main approaches introduced to solve such task.
In the following chapter **Methods** it is described how the task has been approached and the proposed solution is explained.
The **Discussion** chapter contains the outcomes and results of the implemented model and finally the **Conclusions** chapter sums up the overall work along with presenting thoughts and conclusions.

2

Theory

2.1 Generative models

Broadly speaking, image-to-image translation belongs to the family of the generative models approaches. Given two (observable) variables X and Y, the scope of a generative model is to learn the (usually non deterministic) mapping between the two domain $f : X \to Y$, or in other words the model of the conditional probability $P(Y|X = x)$. The main application is hence to get for any sample $x \in X$ its (most-likely) counterpart in the second domain.
Moreover, if the (marginal) probability distribution $P(X)$ is known, it is thus possible to get the joint probability distribution $P(X, Y) = P(Y|X) \cdot P(X)$ and have full control on the generative process.
A multitude of different methods and approaches have been proposed for different uses. Some of the most well-understood and used approaches are for example Gaussian mixture model, Hidden Markov model or Bayesian networks, but all these traditional algorithms have in good part failed to generate complex and realistic images.
The advent of deep neural networks allowed to greatly push forward the boundaries of generative algorithms, allowing approaches with models able to capture and reproduce more complex distributions.

2.2 Image-to-image translation

At the intersection of image processing, computer vision and generative approaches areas, the image-to-image translation problem has recently caught a wide interest both in industry and academia.
This area mainly deals with the task of transforming one image from a certain domain to a different one, preserving the intrinsic information.
Many steps forward have been made in this relatively-new area and, while different classic algorithms have been proposed for several tasks[5] [6], with the recent advancements in the fields of Statistics and Machine Learning (in particular with the introduction of Convolutional neural networks [7]), neural network-based approaches have become the standard solution to approach the problem in analysis.

2.2.1 Supervised case

In the supervised learning setting, the aim is to learn the mapping $f : X \rightarrow Y$ and it is given a set of images samples (x, y) from the target joint distribution $P(X, Y)$.
A traditional approach is to consider the output-space as mostly unstructured, that is each pixel in the output image y is not dependent from the other pixels, hence such relationship can be neglected and the image transformation problem can be reformulated as a per-pixel regression problem. This approach has shown to work well in different tasks, such as image colorization problems [36] [37] or realistic image synthesis [38]. Although different architecture are used, the main idea of these works is to use a deep convolutional neural network to learn the target mapping in an end-to-end way through direct supervision.

2.2.2 Unsupervised case

However, the previously mentioned approaches can not be used in the case in which the given dataset contains samples of unpaired images for the distinct domains X and Y. In other words, only samples from the marginal distributions $P(X)$, $P(Y)$ rather than from the joint distribution. It is hence not possible to directly train a network end-to-end and other methodologies have to be adopted. For example, in [39] the style and content of the input image are extracted using a deep convolutional neural network pre-trained for unrelated object classification tasks and such distinct sets of information are then recombined in order to reproduce the same image content in another style,i.e. the counterpart image in the target output domain.
Other approaches include using Markov random fields with pairwise potentials to model the output image distribution and a Bayesian network for the conditioned likelihood [40]

2.3 Generative adversarial network

The types of generative models used in this work are mainly Generative Adversarial Networks (GANs)[25]. Introduced in 2014 by Ian Goodfellow, this framework quickly become one of the most used generative approach thanks to its flexibility and potential.
Currently,there are more three hundreds papers regarding GANs published[3] and they have been used to tackle problems in several different areas like Medicine, Finance or Animation.
The GAN framework is now presented and some of the different variations employed or used for inspiration during this work are explained. Next, some of the current state-of-the-art solution for the problem in exam are presented as well as their limitations and how to overcome them.

Generative adversarial network is a particular class of adversarial machine learning algorithms, formed by two artificial neural networks,called generator and discriminator network, competing against each-other. As shown in Figure 2.1, in the original version the scope of the generator is to learn a mapping between a given latent space and a target distribution (from which we have some samples), while the discriminator has to effectively distinguish a sample coming from the original distribution from one created by the generator.
In other words, the generator has to 'deceive' the discriminator in believing that the sample it got as input comes from the original distribution.

Such framework can be expressed as a min-max optimization problem:

$$\min_G \max_D L = \min_G \max_D \mathbb{E}_{x\sim\mathbb{P}_r}\left[logD(x)\right] + \mathbb{E}_{z\sim\mathbb{P}_z}\left[log(1-D(G(z)))\right], \tag{2.1}$$

where G is the generator, D the discriminator, $\mathbb{P}_r$ is the distribution we want the Generator to approximate, x is a sample extracted from such distribution, $\mathbb{P}_z$ is our latent distribution and z a sample from it.

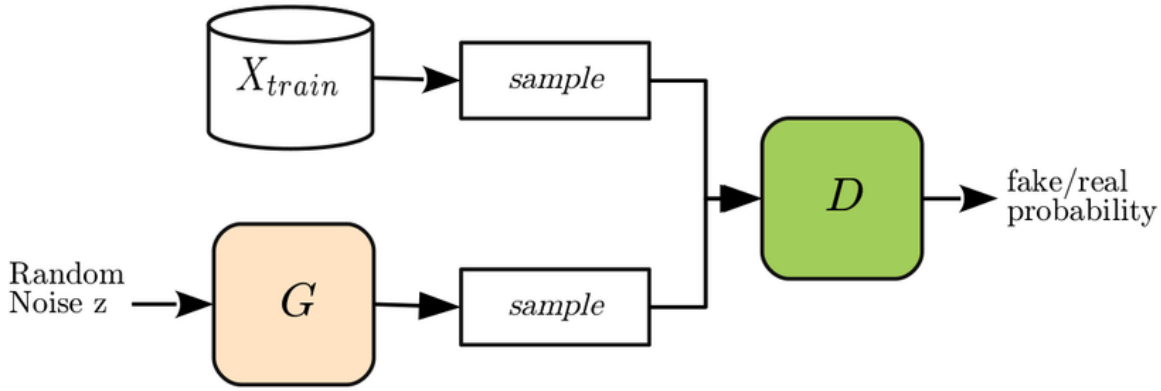


Figure 2.1: Illustration of the basic functioning of the Generative Adversarial Networks. X_{train} is the real samples dataset,G and D are the two neural networks.

As both the generator and discriminator are neural networks, such optimization problem equals to finding the parameters (i.e. the weights) of the two networks, usually through concurrent backpropagation of both the networks.

There are different possible architectures for both the Generator and Discriminator, and it is difficult to assess which configuration might be better suited for the problem in exam.
For example, recurrent neural networks (RNN) have been shown useful for the generation of times series[8] [24], while skip connections and transposed convolutions have been at the basis of high quality image generation [41]

2.3.1 GANs for image-to-image transformation

In the image-to-image transformation problem, we can interpret $\mathbb{P}_r$ of eq. 2.1 as the space in which the synthetic images lie (thus x is a sampled synthetic image), while $\mathbb{P}_z$ is the space of the real images. The generator transforms a synthetic image into a real one, which is right the goal that we want to achieve.
However, from this simple formulation different problems arise.
In fact, there is not any further requirement on the generator's output and input , allowing the network to produce results which do not preserve important information present in the simulated image. Furthermore, the training is often unstable, not always converging, and the model is prone to fall in sub-optimal states (called "mode collapse").
It hence of critical importance to further expand the model adding constrains or changing objective function formulation. Analogously to the previous overview, also in this sub-area there is a distinct difference between supervised and unsupervised approaches.

2.3.1.1 Supervised GAN approaches

A popular and effective approach to try to overcome the above-mentioned problems is to frame the problem in a supervised learning framework, such as in the popular Pix2Pix model[33]. During training, for each input image z the corresponding target image r is known. Thus, it is possible to get the generator to produce a result $G(z)$ as close as possible (usually in terms of L1 distance) to r, getting the following total loss:

$$L = \mathbb{E}_{x\sim\mathbb{P}_r}\left[logD(x)\right] + \mathbb{E}_{z\sim\mathbb{P}_z}\left[log(1 - D(G(z)))\right] + \lambda\mathbb{E}_{z,r}\left[||r - G(z)||_1\right], \qquad (2.2)$$

where λ is a weighting constant parameter used to set the importance of the new term. The latter loss term is also indicative of the paradigm shift from a mere unstructured to a inter-pixel-dependent output space view.

Another interesting feature of the proposed model is the architecture of the discriminator and its loss. In fact, as exemplified in Figure 2.2, instead of having the discriminator produce a single number indicative of the whole image a better approach is to have its output as a probability map [34] [35], in which each element focuses only on a sub-part of the image (which can be identified back-tracing its total field of view). For the loss calculation, the resulting map can then be averaged to get a single number. The benefit of this approach is that the discriminator weights each part of the image equally and it is forced not to simply focus on small details which might only be present in a small portion of the image.

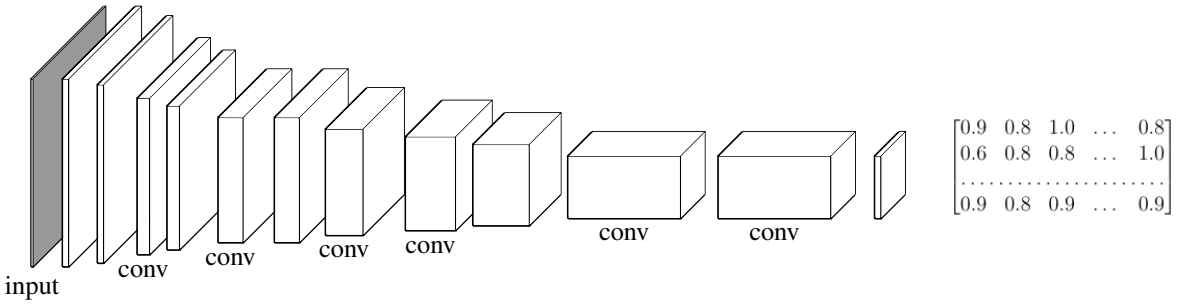


Figure 2.2: Example of the architecture of a discriminator. In most of the cases, its principal component is the convolutional layer, usually alternating it with an activation function such as ReLu for the inner layer and a Sigmoid for the final one.

The main problem with this category of approaches is that it is fundamental to have a dataset of paired images, which is pretty difficult to obtain in our specific case.
In fact, the only reasonably possible way to get it is to start from real images, understand the whole image scenario, then recreate it with high fidelity in a simulation. Along with perception and classification challenges, this task would have to be carried with high precision since the L1 loss term is a per-pixel comparison hence even a small spatial displacement of reproduction could bring to an high error.
This is why the focus of this thesis is on the unsupervised learning case, in which we have two independent datasets, one of real images and one of synthetic ones, which are not linked in any meaningful way.

2.3.1.2 Cycle GAN for unsupervised learning

To overcome the unpaired dataset problem, several recent papers have introduced the idea of using two separated GANs, which allows the addition of new loss components.
In [43] the generator of the first GAN (formed by the generator G and discriminator D_Y) translates a synthetic image in a real one while the generator of the second GAN (formed by the generator F and discriminator D_X) does the opposite, that is translating from the real to the synthetic domain, as illustrated in Figure 2.3. In this setting it is thus possible to impose a cycle-consistency condition, asking $G(F(y))$ and $F(G(x))$ to be equal to respectively y and x:

$$\begin{aligned} L &= L_{GAN_{X\to Y}} + L_{GAN_{X\to Y}} + L_{Cycle_X} + L_{Cycle_Y} \\ &= L_{GAN_1} + L_{GAN_2} + \lambda_1 \mathbb{E}_x \left[||x - F(G(x))||_1 \right] + \lambda_2 \mathbb{E}_y \left[||y - G(F(y))||_1 \right], \end{aligned} \tag{2.3}$$

where L_{GAN_1} and L_{GAN_2} are the previously-described standard losses of the two GANs, while the latter two terms are the cycle-consistency losses introduced in this new formulation.

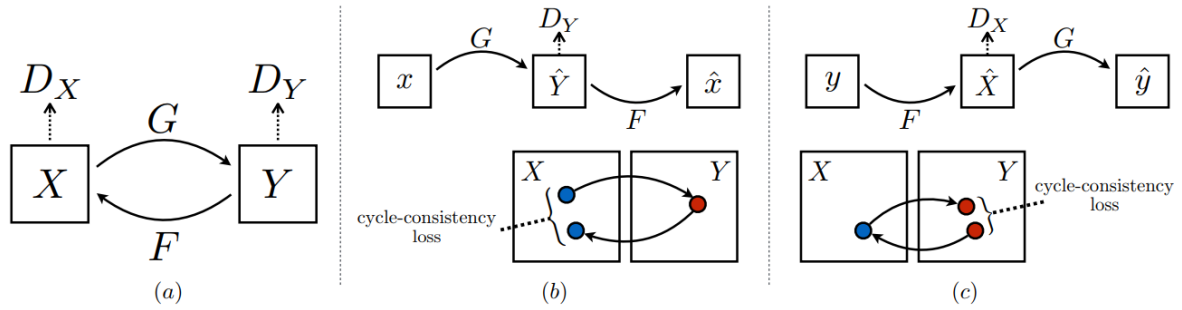


Figure 2.3: Illustration of the basic functioning of the CycleGAN. G and F are the two Generators, while D_x,D_y are the two discriminators of respectively the first and second GANs.

Through the addition of this new term in the loss function the aim is to force a generator not to lose any embedded information of the scene during domain translation, as the original image has to be then restored to the original one by the other generator.
However, in different situations the cycle-consistency loss is not enough and the image translation process is still sub-optimal. This might happen for several reasons, such as the wrong hyper-parameters selection or the malfunctioning of either one of the two GANs, but the most critical ones are related to the high complexity involved in the domain translation and the ill-posed characteristic of the problem.

2.3.1.3 Latent space assumption and weight sharing

An approach to address the ill-posed nature of this problem is to impose further assumptions on the model.
An example is the latent space assumption introduced in the paper "Unsupervised Image-to-Image Translation Networks" [14]. In the specific, such assumption states that a pair of samples coming from two distinct distributions (in our case the real and virtual domains) can be mapped in a shared latent space to the same representation.

As illustrated in Figure 2.4, given two domains X_1 and X_2 instead of obtaining a direct transformation between these two spaces (i.e. the generator network) the goal is to learn the mappings between the domains and the shared latent space Z. In order to do so, such functions are framed as the parts of two variational autoencoders (VAE).

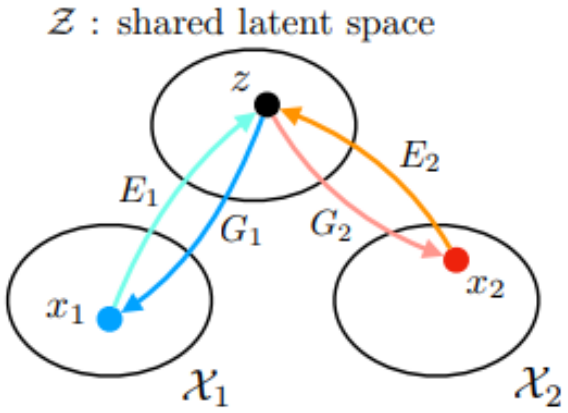


Figure 2.4: Illustration of the shared latent space assumption. Instead of a direct translation from/to the two domains (that is X_1,X_2), the interest is to learn the mapping to the latent space Z shared by the two domains.

Furthermore,as first introduced in the paper "Coupled Generative Adversarial Networks"[15], a further constrain is imposed to the model through the weight sharing between the last layers of the two networks mapping to the latent space. These are the layers extracting the high level features present in the scene.
Analogously, also the first layers weights of the networks mapping from the shared space to the two domains are shared.This further constrain imposes the same encoding and decoding of the high level representations for both the image domains.

2.3.1.4 Multiple inupt modalities

Another different approach to address the previously-mentioned problems is through the addition of additional input modalities. Thus, the task now is to translate the input in a target domain starting from multiple ones. For example, we might want to obtain a RGB image starting from the Near Infrared and Grayscale versions.
While such problem can be approached using different methods [9] [10] [11], the focus in this work is on how such additional information can be used in a GAN framework.

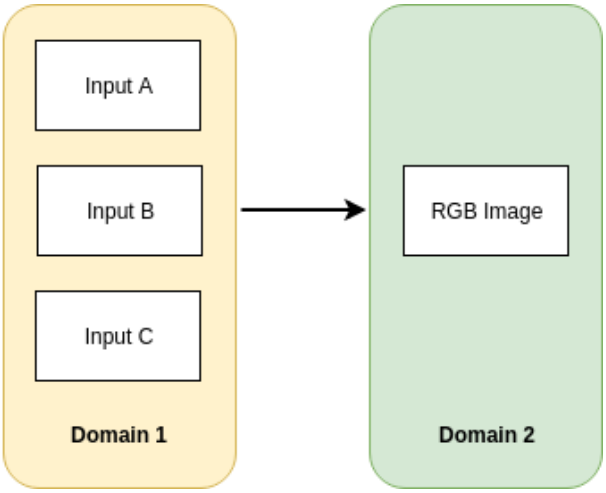


Figure 2.5: Illustration of multiple-inputs-to-image translation model. The main focus is how to successfully make use of the different input modalities.

2.3.1.4.1 Semantic-aware grad-GANs

An advantage of using different input domains is the possibility to introduce further constrains to the transformation that the network has to learn, in this case usually expressed as additional loss terms. Such approach has been adopted in the paper "Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption"[23], trying to perform virtual-to-real domain adaption. In particular, in the proposed approach it has made use of the semantic labels maps as additional domain, which allows two distinct additions to the standard Cycle-GAN paradigms.
First, the discriminator makes use of the semantic labels maps thanks to which it is able to better discriminate images parts based on their semantic classes. Such network is similar to the PatchGAN[33] previously introduced, however instead of producing a single activation map it outputs several channels. Each element of the multi-channel matrix does not directly give the probability of the corresponding area to come from the real dataset, rather it contains such probability conditioned to belonging to the different classes. To extract the probability maps is hence enough to multiply (element-wise) such tensor with the ground truth semantic labels map, as show in Figure 2.6.

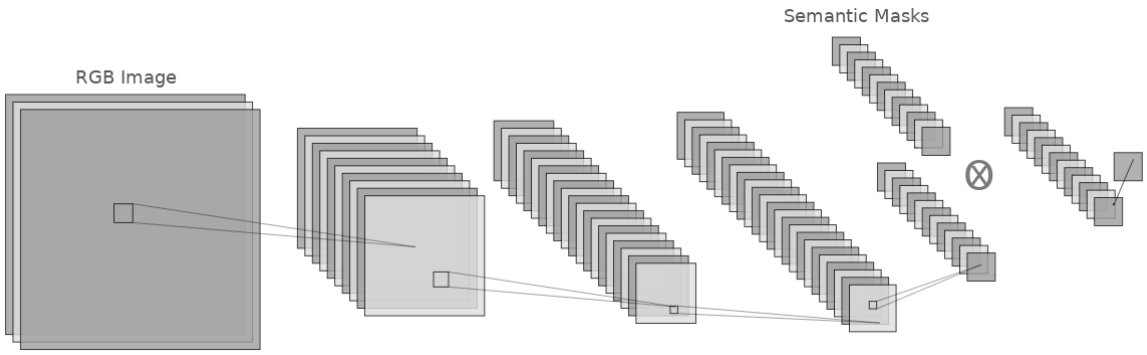


Figure 2.6: Illustration example of the functioning of the semantic-aware discriminator. Instead of giving as output a probability matrix, the CNN outputs a tensor which is then multiplied with the Semantic labels map to get the final probabilities matrix.

Furthermore, a gradient-sensitive loss term is added to enforce the preservation of semantic boundaries during the translation process. In order to get such measure, first the edges of both the synthetic and generated image are computed. It is hence reasonably to expect such edges in the two domains to be similar on the semantic boundaries. In practice, this requirement can be expressed as:

$$\begin{aligned} L_{grad} = L_{grad_1} + L_{grad_2} = \\ \lambda_1 \mathbb{E}_x \left[|| \left(| \left(|C_i * x| - |C_i * G(x)| \right) | \right) \odot sign(C_s * s_x) ||_1 \right] + \\ \lambda_2 \mathbb{E}_y \left[|| \left(| \left(|C_i * y| - |C_i * F(y)| \right) | \right) \odot sign(C_s * s_y) ||_1 \right] \end{aligned} \tag{2.4}$$

Where C is a gradient filter used to extract the edges through convolution. In the paper in exam such filter is the Sobel filter, thus $C = \{C_x, C_y\}$ defined as follows:

$$C_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, C_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{2.5}$$

However, it is important to note that both this additions relies on the ability to provide ground truth semantic labels for both the input domains. If the semantics are given only for one domain, it is not possible to either train the semantic-aware discriminator or to compute L_{grad_2}. In fact, the problem in exam is better represented in Figure 2.5, where the synthetic domain might be formed by multiple modalities, but the real domain is only a set of RGB images.

2.3.1.4.2 Multi-image-to-image GAN

A different (and more) flexible approach has been used in the paper "In2I : Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks" [12] which makes use of the different input domains (regardless their type) through a multimodal generator structure allowing for the addition of further loss terms.
In particular, the overall structure is analogous to the CycleGAN but the two domains can be composed by different modalities. The generator of each GAN is thus not performing a one-to-one image translation, but possibly a one-to-many or many-to-many.
In the proposed approach, an independent discriminator is set for each single modality and such model will be further discussed in the following chapter.

2.4 Image quality assessment

In the context of image-to-image translation, devising a quantitative metric for the quality of the output image is not an easy task. First of all, such simplistic terminology leaves much ambiguity for what "good" means and which aspects of the image such measure is suppose to quantify.
In fact, different measures have been proposed to assess different attribute of an image. The two main categories are Subjective and Objective measures.[17] The first group mainly makes use of scores given by humans on different questions[13] [16] , whereas the second group avoid the human feedback and devices measure based on computational methods. Although in many cases the results of the subjective approaches are considered as more plausible, such family of methods have different drawbacks, namely the lack of impartiality and explainabily or the fact that it is highly time consuming and hardly automatable.
Hence, devising a more quantitative metrics is of great interest and importance.
Furthermore, as the scope of the model in exam is to translate an input image preserving the underlying scene, it is also important to consider whether the model is able to consistently preserve such information.

2.4.1 Subjective image quality

As previously mentioned, in many cases humans' feedback is considered as the best judgment of the quality of an image. The basic approach is the Single-stimulus method, in which each participant expresses a feedback on a given processed image. In the Double-stimulus variant the human is given with both the starting and modified images [16]. Usually, such feedback is given as a score and these (independent) ratings are then averaged out to form the Mean Opinion Score (MOS) measure [13].

2.4.2 Objective image quality

The Objective Image quality measures can be divided in full-reference based measures, in which the assessment is made through a comparison of the image in analysis with the source image which is supposed to be perfect in quality, and no-reference based measures which do not have the original image as term of comparison.

2.4.2.1 Reference based metrics

One of the most common metrics used to compare the resulting image x with its original version y is the Mean Squared Error (MSE):

$$MSE(x, y) = \frac{1}{n} \sum_{i=1}^{n} (x_i - y_i)^2 \ , \tag{2.6}$$

where n is the total amount of pixels in the image. Connected to the MSE metrics is the Peak signal-to-noise ratio (PSNR), which quantify the relationship between the power of corrupting noise and maximum power of a signal:

$$PSNR = 10 \cdot log_{10} \left(\frac{255^2}{MSE} \right) \ . \tag{2.7}$$

The main problem with these metrics formulation is that its outcome highly differs from the feedback given by an human judgment. A possible reason for this mismatch is that the human judgment is (in part) based on the structural information of the image naturally extracted by human eye, rather than a mere per-pixel comparison.
To address this problem, the Structural Similarity Measure (SSIM) [18] focuses at the pixels dependencies and tries to quantify the image quality comparing the structures of the two images:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.8}$$

, where C_1 and C_2 are constants put to avoid instabilities occurring when the numerator of denominator goes towards zero, defined as:

$$C_1 = (K_1 L)^2 \,, \;\; C_2 = (K_2 L)^2 \,,$$

in which K_1 and K_2 are small constants close to zero and L is the range of the pixel values (commonly 255).
Finally, μ_x and μ_y are the mean intensities of the source and modified image:

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i \,, \;\; \mu_y = \frac{1}{N}\sum_{i=1}^{N} y_i$$

and σ_x and σ_y are the standard deviation of the two images and σ_{xy} can be expressed as the covariance:

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)^2\right)^{1/2} \,, \;\; \sigma_y = \left(\frac{1}{N-1}\sum_{i=1}^{N}(y_i - \mu_y)^2\right)^{1/2}$$

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \,.$$

2.4.2.2 No-reference metrics

Regardless some differences of approach, all above-mentioned measures rely on the (strong) assumption that the source image is provided and the measure can thus be expressed as the output of some sort of comparison. However, this family of approaches can not be used if the reference image is not provided (or does not exists).
In such case No-Reference Image Quality Assessment (NR-IQA) metrics are used. There are two categories of NR-IQA: general-purpose and application-specific [19]. Application-specific NR-IQA measures assume that the image in analysis is the product of a specific distortion being applied on a source image. Thus, the quality metrics is attributed as the level of such distortion. For example, in the case of a blurried image different methods have been proposed to quantify the level of such distortion. One of the most popular is the Maximum Local Variation (MLV), which assess the sharpness of the image using the pixel intensities' high variations. [21]

General-purpose IQA metrics do not assume any specific underlying deformation, rather relying on a more holistic approach usually divided in two parts: feature extraction, in which

different features linked to the quality of the image are extracted, and model mapping, that computes the quality measure from the extracted features [20]. While it is possible to hand-craft these feature through domain expertise, more recent solutions rely on a end-to-end neural network approach to learn both the feature-extraction and mapping tasks[22].

2.4.2.3 Application-based metrics

Instead of trying to construct a metric that matches the human vision system's feedback, an alternative is to employ other methods of comparisons, depending on which application such model is designed for.
For example, in a data augmentation setting an informative metrics can be the increase of the model accuracy using the generated data [42] [35].
In specific case in exam, since the aim is to transform an image preserving the underlying information of the scene, a good result is expected to preserve the semantics of the original image. Hence, a quality estimation can be interpreted as the agreement of the semantics map extracted from the refined image (namely using a segmentation algorithm) and the semantics of the synthetic image used for the transformation[23].
Analogously, another possible approach is to use part of the GAN model's as source of feedback on which it is possible do construct a quality metric. For example, the output of the discriminator on a refined image can be a good indicator of how well such image can fool the discriminator, or in other words how similar it is to the real images[26].

3
Methods

The work of this thesis can mainly be divided in two parts. First, the design and implementation of a suitable model for the problem in exam, consequentially the set up of different quantitative metrics to assess the quality of the refined image obtained from the designed model.

3.1 Datasets choice

Two open-source datasets have been using to train and validade the model. For the synthetic domain, the dataset used is a part of the SYNTHIA VIDEO SEQUENCES Datasets[27]. In particular, the fall sub-sequences 02 and 05, featuring a New York-like city. Such choice is motivated by the goal to have a coherent dataset representing an urban environment.
Although there are datasets with an higher quality synthesized images [4] [28], the peculiarity of the SYNTHIA dataset is to provide the semantics and the depth map (pixel-level) for each image sample.
While the dataset provides images of the front,rear,left and right car view, only the first two classes are considered for consistency with the real dataset. The resulting dataset is composed by a a total of of 2870 images.

Figure 3.1: Example set from the SYNTHIA Dataset. In order from left to right, it is shown the RGB image, depth map and semantic labels map of a scene

Regarding the real domain, it has been opted for employing the Cityscape Dataset [30], one of the most commonly used image dataset that provides ground truth pixel-level semantics. Furthermore, the images are taken from an angulation similar to the SYNTHIA's one, easing the

translation process.

Figure 3.2: Sample image from the Cityscape Dataset.

At training, the resulting dataset for the real domain is composed by 1879 pictures (taken in different German cities). The images from both the dataset are then resized to 256x256 pixels dimension to be fed in the model. Figures 3.1 and 3.2 show examples of respectively synthetic and real datasets samples.

3.2 Model strucutre

3.2.1 Base model

The overall model structure is analogous to the one introduced in Multi-image-to-image GAN [12]. As an expansion of the CycleGAN framework, such model is composed by two distinct GANs. In the first one, shown in the left part of Figure 3.3 , the generator translates a set of input (RGB image, depth and semantics maps) in an image that looks as realistic as possible (i.e. is labeled as true by the corresponding discriminator) and satisfies certain conditions, namely the conservation of the underlying information of the scene, while the discriminator's aim is to differentiate between real and generated images.

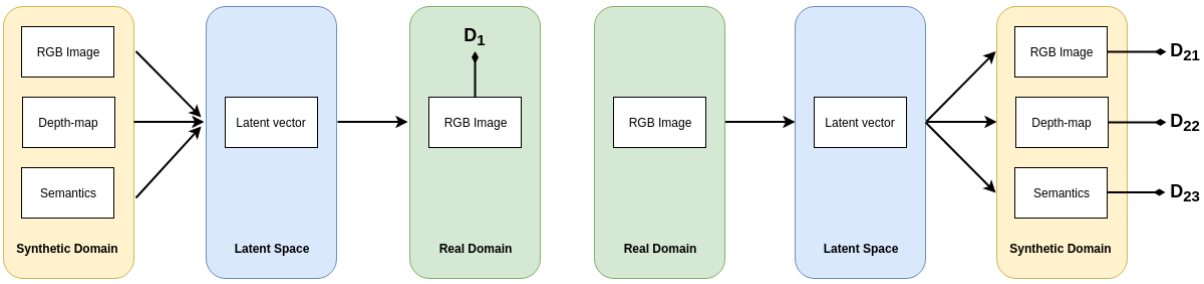


Figure 3.3: Illustration of the overall structure of the basic model. On the left is shown the first GAN (from synthetic to real), while on the right the second one (from real to synthetic)

The second GAN functioning is the opposite: its generator produces the synthetic domain's output from a real domain's image while the discriminator classifies a set of input modalities

as belonging to the given synthetic dataset or as a result of the corresponding generator.
Let R be the real dataset, S the synthetic one, G_1,G_2 and D_1,D_2 respectively the generators and discriminators of the two GANs. In this formulation, the maximization objective function of the first discriminator is:

$$L_{D_1} = \mathbb{E}_{r\sim\mathbb{P}_R}\left[logD_1(r)\right] + \mathbb{E}_{s\sim\mathbb{P}_S}\left[log(1 - D_1(G_1(s)))\right] \ . \tag{3.1}$$

In the original formulation there is an indipendent discriminator for each of the modality of the synthetic domain, thus the loss is their output sum (or average):

$$L_{D_2} = \sum_{i=1}^{3} D_{2i} = \sum_{i=1}^{3} \mathbb{E}_{s\sim\mathbb{P}_S}\left[logD_{2i}(s_i)\right] + \mathbb{E}_{r\sim\mathbb{P}_R}\left[log(1 - D_{2i}(G_2(r)_i))\right] \ . \tag{3.2}$$

Following the eq. 2.3 presented in the CycleGAN model, the minimization objective function for the two genereators is:

$$\begin{aligned} L_{CycleGAN} &= L_D + L_{cycle} = L_{D_1} + L_{D_2} + L_{cycle_1} + L_{cycle_2} \\ &= L_{D_1} + L_{D_2} + \lambda_1\mathbb{E}_r\left[||r - G_1(G_2(r))||_1\right] + \lambda_2\mathbb{E}_s\left[||s - G_2(G_1(s))||_1\right]. \end{aligned} \tag{3.3}$$

Furthermore, following the assumption that the real and synthetic domains have a common latent representation, it is possible to further constrain the problem requiring such latent representation to be preserved during a cycle as discussed in section 2.3.1.3. The two generators transformations can be decomposed as :

$$\begin{aligned} G_1(\cdot) &= E_1(F_1(\cdot)) \\ G_2(\cdot) &= E_2(F_2(\cdot)) \ . \end{aligned} \tag{3.4}$$

Thus, the latent consistency loss can be written as:

$$\begin{aligned} L_{latent} &= L_{l_1} + L_{l_2} \\ &= \lambda_3\mathbb{E}_r\left[||F_2(r) - F_1(E_2(F_2(r)))||_1\right] + \lambda_4\mathbb{E}_s\left[||F_1(s) - F_2(E_1(F_1(s)))||_1\right] \ . \end{aligned} \tag{3.5}$$

The overall architecture of the generator network of the first GAN can be divided in two distinguished parts,namely E_1 and F_1. The structure of F_1 can be seen in Figure 3.4. First, for each input modality an extractor network formed by convolutional and residual blocks, is used to extract the meaningful features. The resulting features are then stacked and the encoder network will give the latent-space representation.

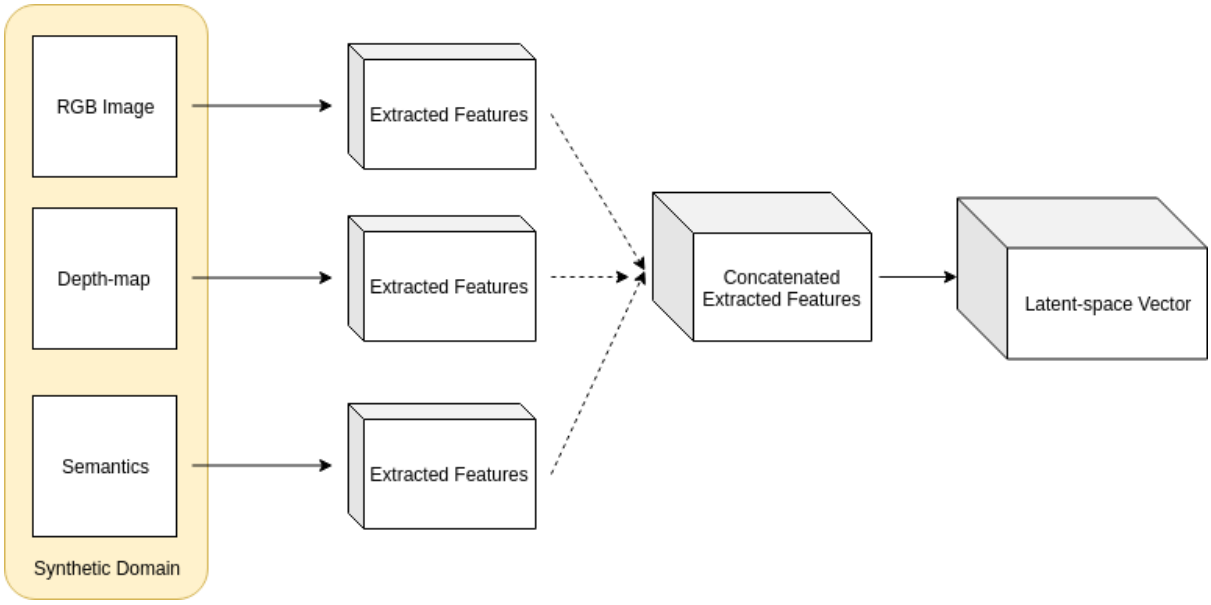


Figure 3.4: Illustration of the first half F_1 of the Generator for the Synthetic-to-Real translation. Its aim is to combine different input modalities to extract the latent representation of the input scene

Finally, the decoder network E_1 (formed by residual and deconvolutional blocks) translates the latent-space representation to the real domain.
The overall architecture and functioning of the second GAN (taking as input the RGB image from the real domain and generating the different synthetic modalities) is specular to the one of the just described.

3.2.2 Grad-sensitive loss

As long as the cycle consistencies losses are kept low, the semantics of the generated image might vary considerably. It might be easier for the generator to produce considerable changes and distortions (which can then be removed by the second generator) to the image so to lower its discriminator loss rather than correctly preserving the semantics of the original scene. In fact, the cycle-consistency loss enforces the preservation of the starting image information (through a cycle) but does not put a constrain on how such information is encoded in the generated image. For this reason, a more explicitly loss term which (only) makes use of the generator's input and output is needed.

Hence, in attempt to specifically stronger enforce the preservation of the semantics, a gradient sensitive loss similar to the one described in Section 2.3.1.4.1 is introduced.
As exemplified in Figure 3.5, the gradients similarity is enforced only at the borders between different classes because the changes within a class should not be punished since it might be given by a change of texture or lighting.

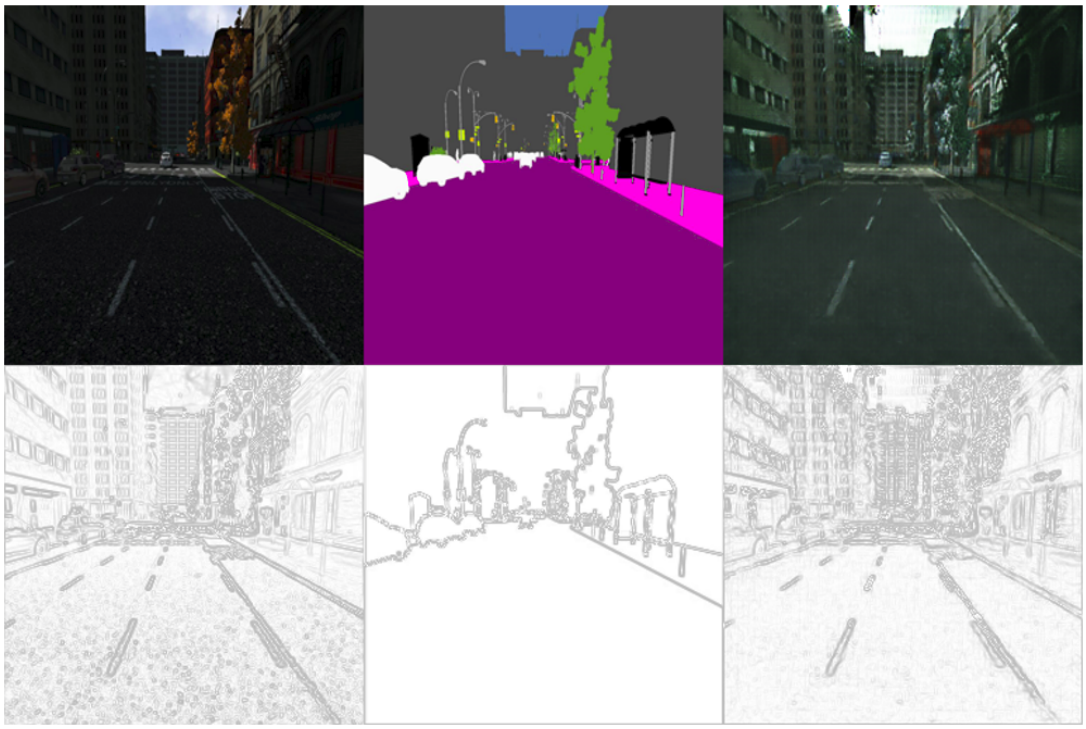


Figure 3.5: Top row, from left to right: the starting synthetic image s_1, its labels map s_3 and the refined image $G_1(s)$. Bottom row, from left to right: the extracted edges for the syntethic image $C * s_1$, the class boundaries $C * s_3$ and the edges for the refined image $C * G_1(s)$. Note how for example the texture of the road changes

However, differently from the reference case, the semantics label map is given only in the synthetic domain thus the gradient-sensitive loss term is reduced to:

$$L_{grad} = \lambda_5 \, \mathbb{E}_s \left[|| \left(| \left(|C * s_1| - |C * G_1(s)| \right) | \right) \odot sign(C * s_3) ||_1 \right] \tag{3.6}$$

Where $s = \{s_1, s_2, s_3\}$ is the triplet formed by respectively RGB image, depth map, semantics label map (of the synthetic dataset) and C is the filter used to extract the edges.
It would be also possible to express such loss term for the second GAN, using the generated semantics instead of the ground truth (as it is not given for the real domain). However, since the resulting semantic label map is usually inaccurate (especially at the beginning), such loss would hinder the correct training process penalizing the wrong gradients differences.

3.2.3 Semantic-aware discriminator

Another problem that might arise concerns the structure of the discriminator for the second GAN. In fact in the current form, that is an independent network for each modality, it models the three marginal distributions $\mathbb{P}_{s1}, \mathbb{P}_{s2}, \mathbb{P}_{s3}$ rather than their joint distribution $\mathbb{P}_s = \{s_1, s_2, s_3\}$ impacting consequently the generator's learning. In other words, the generator is only pushed to generate outputs single-modality-wise realistic but it is not punished if the three modalities do not match.
On the other hand, simply devising a discriminator that takes as input the three input modalities

together might be sub-optimal, as such network could simply learn to focus on a single modality (the easiest to exploit) neglecting the other two.
To solve this problem, an additional discriminator has been added. Its framework is analogous to the Semantic-aware discriminator introduced in Section 2.3.1.4.1 with both RGB image and semantic mask given by the generator.

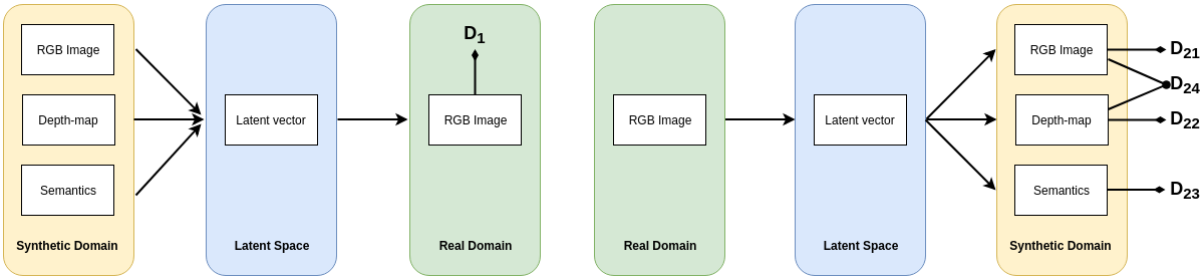


Figure 3.6: Illustration of the overall structure of the modified model. On the left is shown the first GAN(from synthetic to real), while on the right the second one (from real to synthetic

In case of 'disagreement' between the two output modalities, for example if a building is labeled as car, it would be relatively easy for such discriminator to correctly classify the sample as a generator's output, forcing the latter to produce more cross-modality coherent results.
The total loss for the second GAN discriminators introduced as Eq. 3.2 thus becomes:

$$L_{D_2} = \mathbb{E}_{s\sim\mathbb{P}_S}\left[logD_{24}(s_1|s_3)\right] + \mathbb{E}_{r\sim\mathbb{P}_R}\left[log(1 - D_{24}(G_2(r)_1|G_2(r)_3))\right] + \sum_{i=1}^{3}\mathbb{E}_{s\sim\mathbb{P}_S}\left[logD_{2i}(s_i)\right] + \mathbb{E}_{r\sim\mathbb{P}_R}\left[log(1 - D_{2i}(G_2(r)_i))\right] \ . \tag{3.7}$$

3.2.4 Weigthed cycle-consistency

In its basic form, the Cycle-Consistency loss for the first GAN is the L_1 distance between the initial sample s and the resulting output of the two generators $G_2(G_1(s))$:

$$L_{cycle_2} = \lambda_2 \mathbb{E}_s\left[||s - G_2(G_1(s))||_1\right] \ . \tag{3.8}$$

This loss is the principal enforcing mechanism for the content preservation [43], however by definition is prone to give more importance to the classes which have bigger presence in the image. Smaller objects, such as pedestrian or poles, might not be accurately preserved during the translations as they do not weight much in the current loss formulation.
It might hence be useful to modify the cycle-consistency loss to attribute to each of the n class the same weight, using the ground truth labels s_3:

$$L_{cycle_2} = \lambda_2 \ \mathbb{E}_s\left[\frac{1}{n}\sum_{i=1}^{n}\frac{||\ |s - G_2(G_1(s))| \odot s_3(i)||_1}{||s_3(i)||_1}\right] \ , \tag{3.9}$$

where s_3 is the semantic labels map in a one hot encoding (OHE) version. Thus, $s_3(i)_{x,y}$ is 1 if the pixel (x, y) belongs to the class i, 0 otherwise (there are not cases in which a pixel belongs to two different classes simultaneously).

Such loss term could be also added to the second GAN using the generated semantic map $G_2(r)_3$ however, similarly to the gradient sensitive loss case, it has been opted not to include it since it might encourage the generator to minimize such loss term in undesired ways, such as having the most number of classes empty (so that multiple term of the class average are zero).

3.2.5 Self regularization

A different approach to enforce the preservation of the information is to introduce a self-regularization term that penalizes a big per-pixel difference between the starting and modified image[35]:

$$L_{sreg} = \mathbb{E}_s \left[||s - G_1(s)||_1 \right] + \mathbb{E}_r \left[||r - G_2(r)||_1 \right] \ . \tag{3.10}$$

Such loss term poses a stronger constrain on the generation and, coupled with the cycle loss, might encourages the identity mapping. However it might nonetheless be a beneficial term as it discourages abrupt changes in the modified image which might then be countered during the second GAN transformation still yielding to a low cycle loss.

Summing up, the total (minimization for G_1, G_2, maximization for D_1, D_2) objective loss of the model is:

$$L = \lambda_D L_D + \lambda_{cycle} L_{cycle} + \lambda_{latent} L_{latent} + \lambda_{grad} L_{grad} + \lambda_{sreg} L_{sreg} \ . \tag{3.11}$$

3.3 Training procedure and settings

The overall training procedure is analogous to the CycleGAN one. At each step, both the discriminators and generators get update once as shown in Algorithm 1 (with learning rate respectively of $1 \cdot 10^{-4}$ and $2 \cdot 10^{-4}$, linearly decreased after the first 100 epochs). Such loop is repeated for a total number of 200 epochs and batch size of one, using the Adam algorithm[31] as optimization algorithm.
The weights of the networks have been initialized using the Xavier method [32].

Algorithm 1: Model training procedure

Initialize the networks D_1,D_{21},D_{22},D_{23},D_{24} and G_1,G_2
for *epochs_number* **do**
    Update learning parameters and $\lambda_{i=1,..,5}$
    for *samples **in** synthetic dataset* **do**
        take random image from real dataset
        train step D_1
        train step D_{21}
        train step D_{22}
        train step D_{23}
        train step D_{24}
        train step G_1 and G_2
    end
end

The model has been implemented using Python 3.6 and the machine learning library Pytorch. Training and inference has been run on a Debian GNU 9 operating system equipped with Nvidia GeForce GTX 1080 Ti.

3.4 Evaluation metrics

As already explained in the previous chapter, there are different metrics to evaluate the quality of the output. Overall, the resulting image has to look as realistic as possible (with respect to the real images dataset), while at the same time preserving the semantics of the synthetic input. In this work it has been decided to employ four different approaches and compare their different output.

3.4.1 Mean opinion score

The mean opinion score is probably one of the most robust and used metrics. In particular for this work a survey composed by a variable number of questions has been completed by a total of 84 participants.
In each question a pair of synthetic and refined image is shown and the user is asked to give a score from one to five regarding how well he or she thinks that the refined image could be a realistic version of the synthetic counterpart.
The reason why also the synthetic image is shown is the need to keep in consideration how well the semantics are preserved, avoiding the case in which a heavily-altered image receives an high score even if the content of the refined image is different.

3.4.2 Model-based metrics

Another possible approach relies on using part of the model itself as a source of feedback on the quality of the image.
Specifically, the main idea is to use the output of the discriminator of the first GAN as an indicator of the image quality. In fact, a not-realistically-looking refined image would not be similar to a real one thus it is reasonable to expect the discriminator to be confident in labelling it as fake. Given a sample from the syntehtic dataset $s = s_1, s_2, s_3 \in S$, (respectively RGB image, Depth map, semantics map) such measure m_1 is defined as :

$$m_1(s) = mean\left(D_1(G_1(s))\right) \ . \tag{3.12}$$

Along with the output of the discriminator, it might be also of interest to visualize the other fundamental part of the CycleGAN, namely the cycle results. However, instead of extracting the (per pixel) difference between the original synthetic image and the reconstructed one, what has been computed is the difference between the refined image and the result of a consequent cycle:

$$m_2(s) = ||G_1(s) - G_1(G_2(G_1(s)))||_1 \ . \tag{3.13}$$

The aim is to have such measure highlight the discrepancies between the synthetic and refined images that the mere cycle loss would not be able to show as the generators have expressly trained to reduce it.

3.4.3 Segmentation metrics

The preservation of the scene information embedded in the synthetic image can be well represented by its semantic labels. Thus, it is reasonable to expect the refined image to preserve such attribute.
In order to be able to extract the semantic labels from the refined image, a semantic segmentation algorithm is trained on the real-images dataset (with the corresponding ground truth labels). Such trained algorithm is then used on the refined image and its output is compared on the semantic labels of the original synthetic image. Let $A(\cdot)$ be the segmentation algorithm trained on the real dataset R, such quality metrics can be defined as:

$$m_3(s, A) = ||s_3 - A(G_1(s))||_1 \ . \tag{3.14}$$

The choice or model and training of the Segmentation Algorithm plays a crucial role in such metrics definition, as a poorly-performing algorithm would make such metrics meaningless.
For this case it has been adopted a Densely Connected Convolutional Networks (DensNet) tailored for the specific purpose as described in [44] and shown in Figure 3.7.

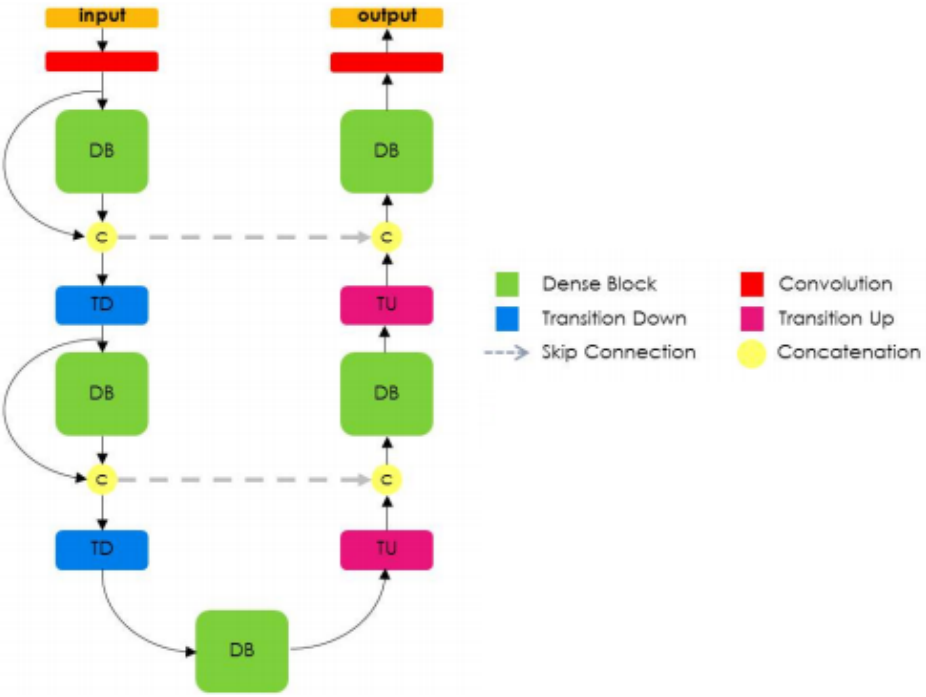


Figure 3.7: Illustration of the overall structure of the Fully Convolutional DenseNet, showing the building blocks used. The Dense Block is formed by the concatenation of different resolutions feature maps, Transition Down is composed by a 1x1 convolution followed by pooling operation and Transition Up is an upsampling operation.

This model has been chosen since it has been shown to perform well on the segmentation task of the CamVid dataset [29]. The training has been conducted for 300 epochs with batch size of one.

3.4.4 Classifier based metrics

Expanding on what introduced in the section 3.4.2, the discriminator-based metrics can be extended to the use of a different network. In fact, a different architecture and training procedure might yield to a classifier better able to discriminate the real from the fake images, as its training is independent from the generator's one.
With this goal, a model introduced in [45] has been trained using the real dataset and a set of refined images.
Such trained binary classification algorithm $C(\cdot)$, whose structure is illustrated in Figure 3.8, is then used (instead of the discriminator) to try to assess the image quality:

$$m_3(s) = C(G_1(s)) \ . \tag{3.15}$$

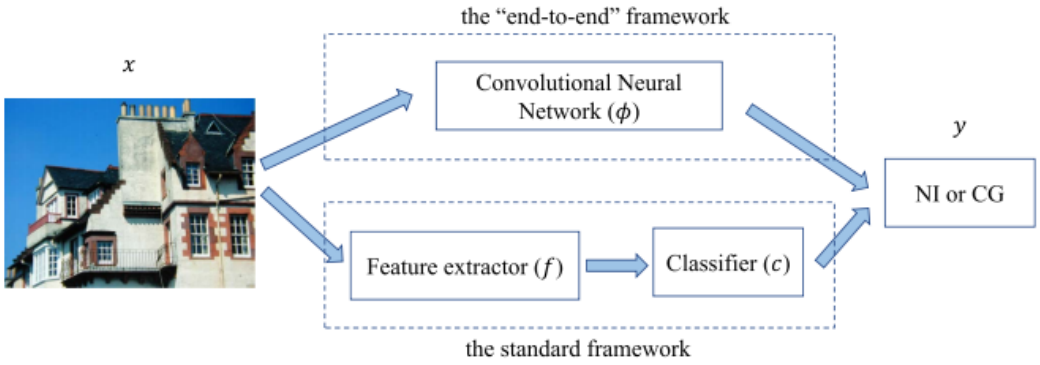


Figure 3.8: Illustration of the overall structure of the Classifier algorithm. The model makes use of both automatically extracted features (through the Convolutional neural network) and handcrafted features, combining then their result

The model is trained for 40 epochs (which are enough to achieve a considerably high accuracy), learning rate of 0.1 with batch size 1.

4

Results

In this chapter the results of the trained models are presented.
The results are shown and compared both visually and using quantitative metrics.

4.1 Experimental results

As shown in equation 3.11, the total loss of the model is given by the weighted sum of different terms previously explained. Hence, the choice of the corresponding weight parameters is critical. The higher one of this parameter, the more effect its corresponding loss term has on the total loss and consequently the generators will focus on it, rather than the other terms.
For this reason, different configuration have been tried with the aim of study how such changes might affect the model behaviour. In total, four different configuration have been tested and the outcome is now shown.

4.1.1 First model

As first trial, the model has been set to be closer with the original approach. The first three parameters have been kept unchanged, that is $\lambda_D = 1$ and $\lambda_{cycle} = 10$, $\lambda_{latent} = 1$, and the cycle consistency loss is equal to the original, hence without the weighted method explained in section 3.2.4.
The weighting parameter for the gradient-sensitive loss term has been put as $\lambda_{grad} = \frac{1}{6 \cdot 10^4}$ and the self-regularization term λ_{sreg} is put to zero to analyze the results without such component.

Examples for the current set are shown below in Figure 4.1:

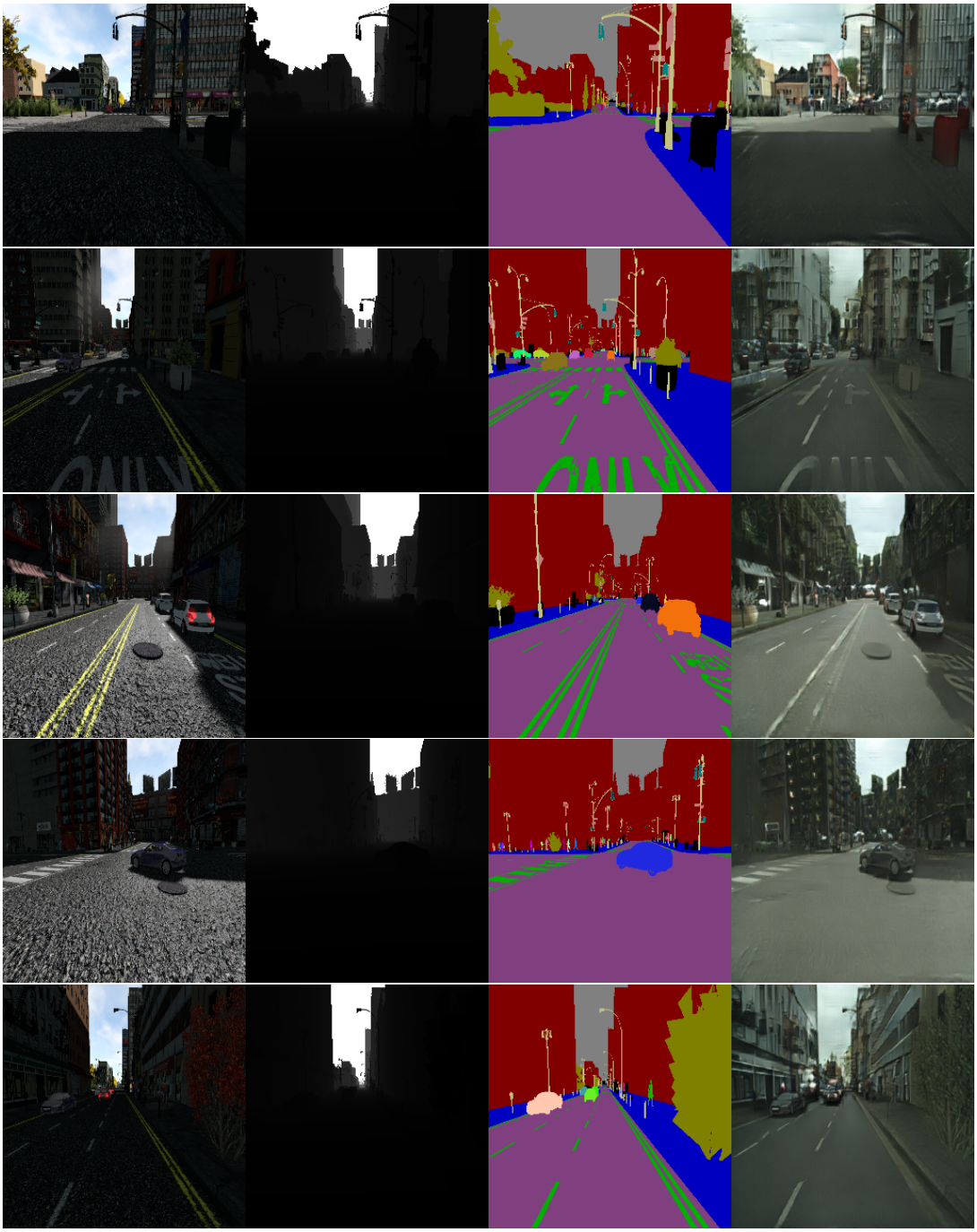
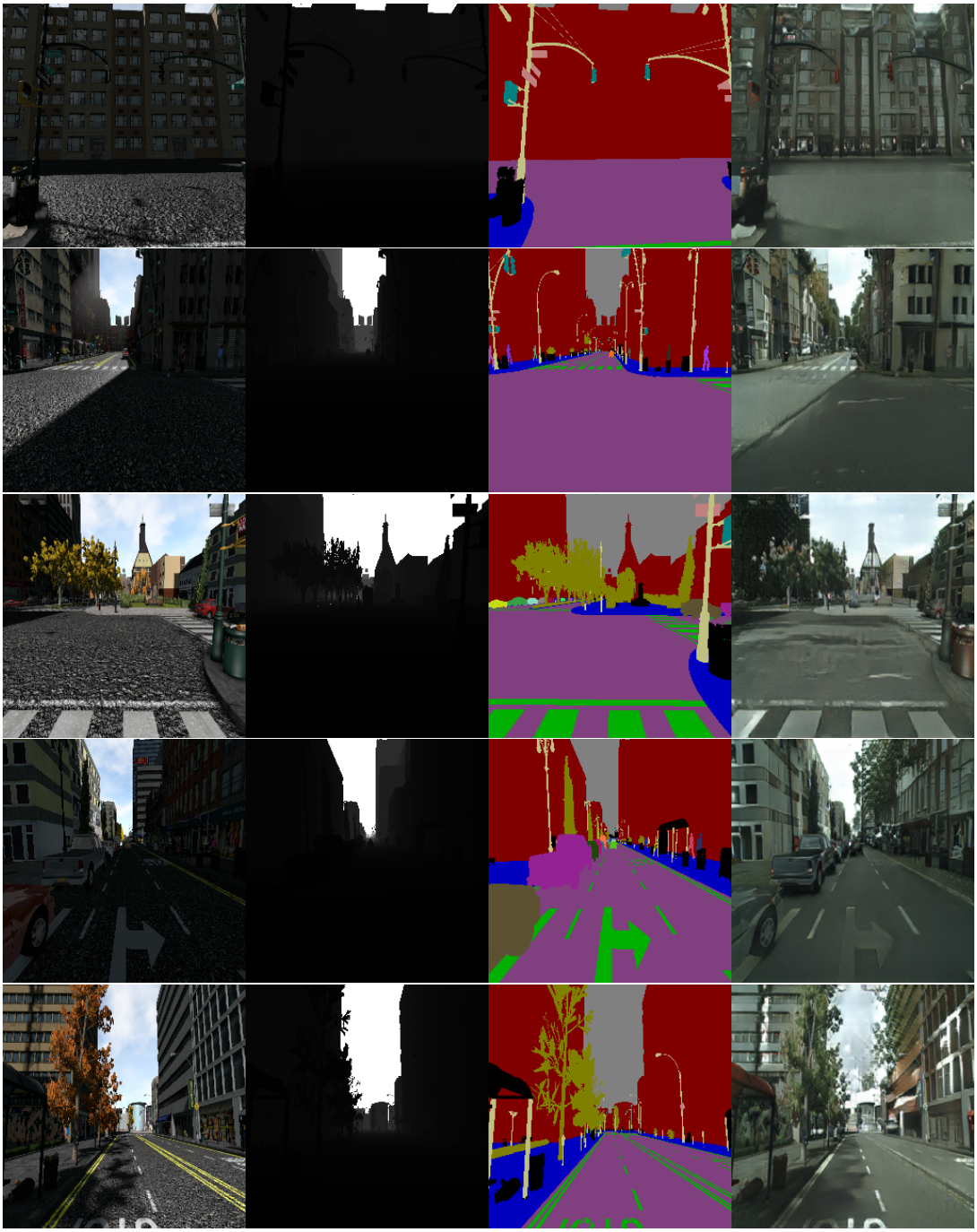


Figure 4.1: Samples of results of the trained model. From left to right, there are respectively the RGB image, depth and semantic labels map (the three input modalities of the synthetic domain), followed by the corresponding refined image.

4.1.2 Second model

For the second model, it has been added the self-regularization loss, putting $\lambda_{sreg} = 2$.
The other parameters are left unchanged and some results are shown in the Figure 4.2 below:



Synthetic image | Depth map | Semantic labels | Refined Image

Figure 4.2: Samples of results of the trained model. From left to right, there are respectively the RGB image, depth and semantic labels map (the three input modalities of the synthetic domain), followed by the corresponding refined image.

4.1.3 Third model

The third model is analogous the the second one, with the weighted cycle loss instead of the original cycle-consistency formulation. Some results are shown in the Figure 4.3 below:

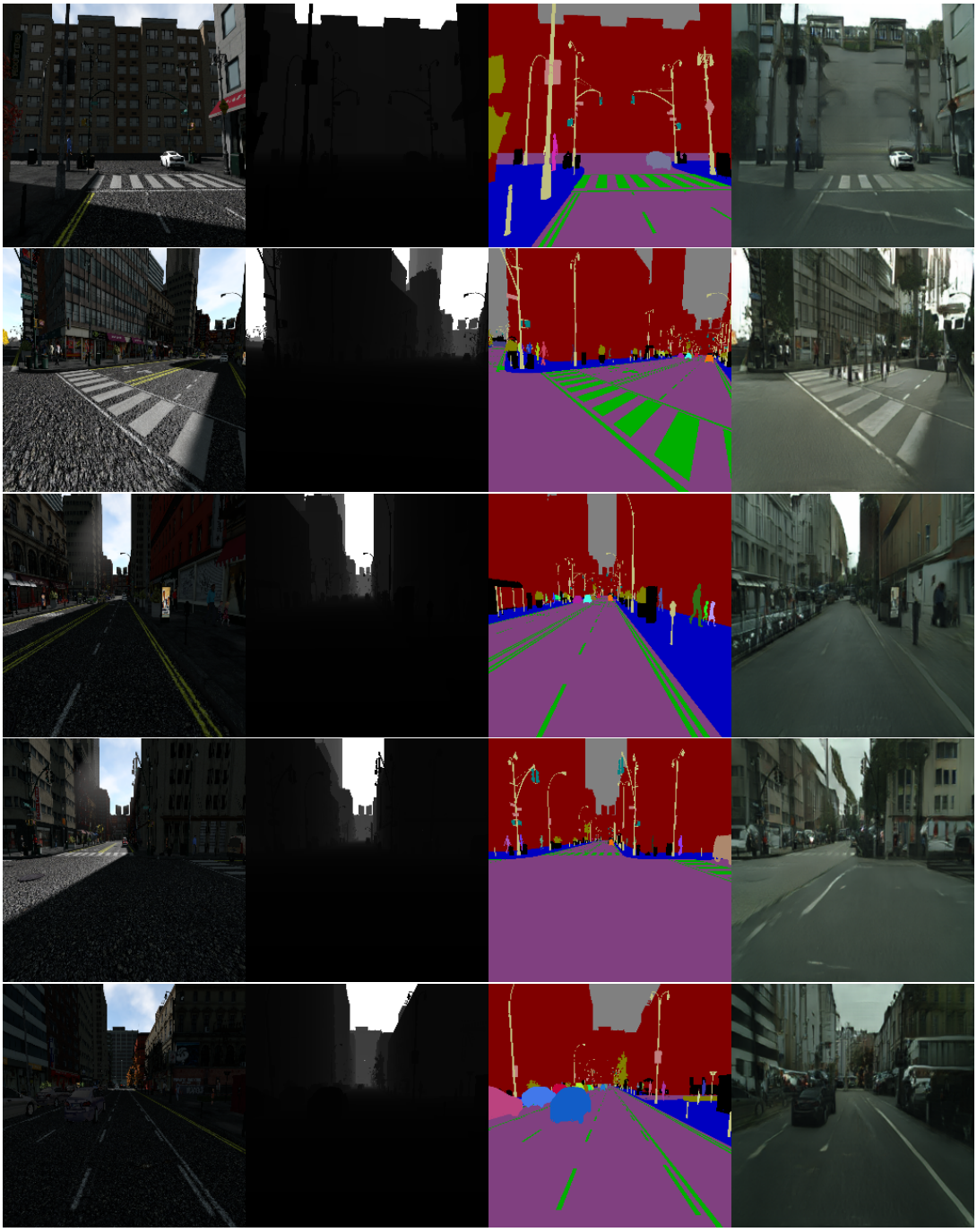


Figure 4.3: Samples of results of the trained model. From left to right, there are respectively the RGB image, depth and semantic labels map (the three input modalities of the synthetic domain), followed by the corresponding refined image.

4.1.4 Fourth model

In order to have an output closer to the original synthetic image, the parameters weights of the Gradient-Sensitive and Self-Regularization losses are increased to respectively $\lambda_{sreg} = 5$ and $\lambda_{grad} = \frac{1}{4 \cdot 10^4}$. The other parameters are kept unchanged from the previous model and some results are shown in Figure 4.4

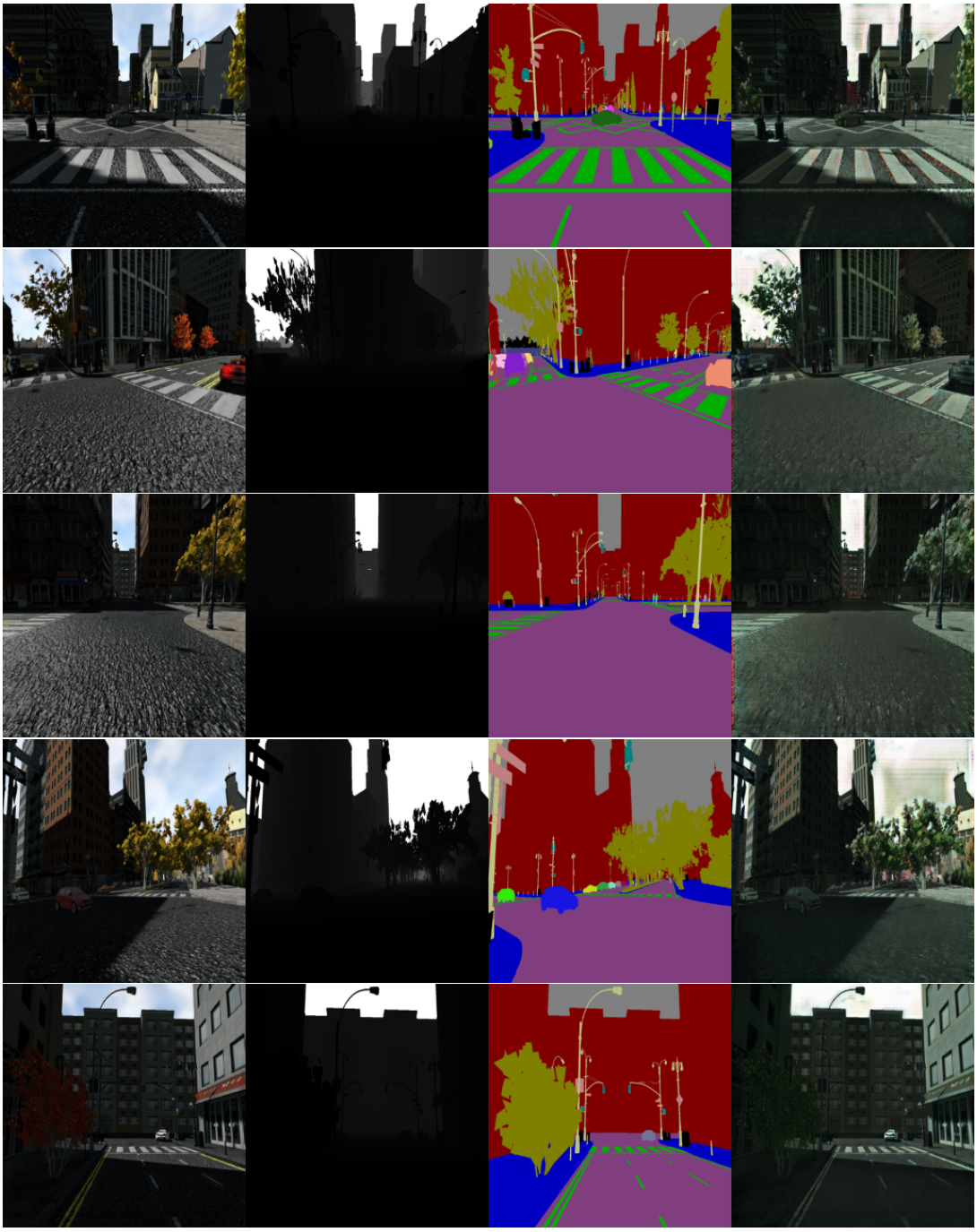


Figure 4.4: Samples of results of the trained model. From left to right, there are respectively the RGB image, depth and semantic labels map, followed by the corresponding refined image

4.2 Measures results

As explained in Section 3.4 , different metrics have been proposed in order to assess the quality of the results and such results are here shown for the different models proposed.

4.2.1 Model-based metrics

4.2.1.1 First model

Using the metrics defined in Equation 3.4.2, for the first model the resulting average m_1 score is $\mu_{m_1} = 0.5221$ (which means that the discriminator on average is not able to distinguish effectively between real and refined images), with a standard deviation of $\sigma_{m_1} = 0.1369$. Some of examples of refined images with high and low scores are shown in Figure 4.5.

(a) Examples of generated images with high m_1 (respectively 0.8412 and 0.8227)

(a) Examples of generated images with low m_1 (respectively 0.0575 and 0.212)

Figure 4.5: Some of the output images with the highest or lowest m_1

The values of the second metrics m_2, defined in Equation 3.13, have a less directly interpretable meaning.
Such metrics results are shown in Figure 4.6 with m_1. The plot might be of interest to both visualize the results of this metrics and to spot any clear relationship with the previously-discussed metrics.
The Pearson correlation coefficient (PCC), which measures the linear correlation of these two metrics results (for the model in exam), is $PCC = 0.127$, with a two-sided p-value (with null hypothesis of independent sample pairs) of $p = 9.06 \cdot 10^{-7}$

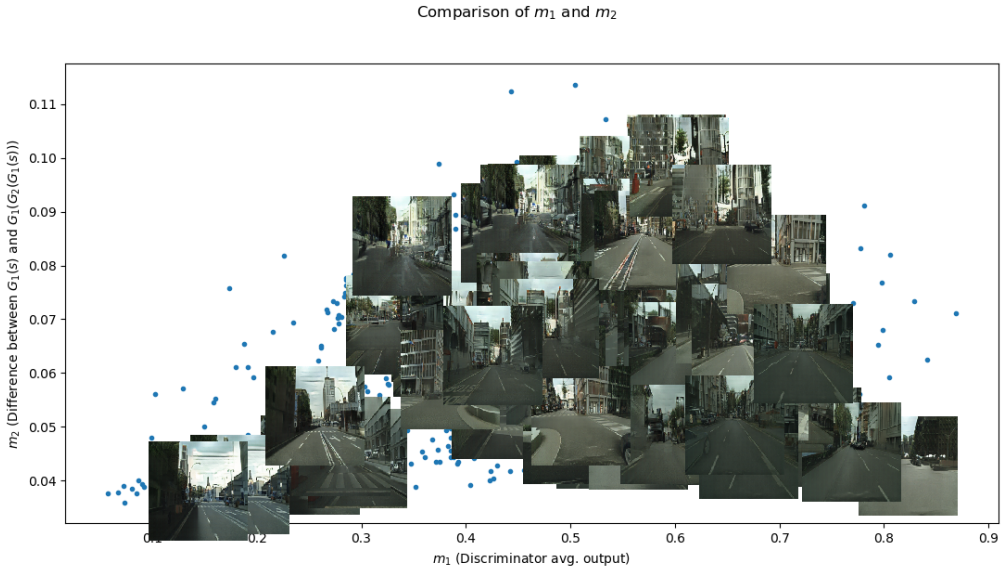


Figure 4.6: Scatter plot of the model results with respect to m_1 and m_2. Some of such points are replaced with the corresponding refined image to better visualize the results

4.2.1.2 Second model

Analogously, for the second model the resulting average m_1 score is $\mu_{m_1} = 0.20659$, with a standard deviation of $\sigma_{m_1} = 0.11412$. Some of examples of refined images with high and low scores are shown in Figure 4.7.

(a) Examples of generated images with high m_1 (respectively 0.5929 and 0.5629)

(a) Examples of generated images with low m_1 (respectively -0.150 and -0.103)

Figure 4.7: Some of the output images with the highest or lowest m_1

The Pearson correlation coefficient in this case is $PCC = 0.342$ with corresponding p-value $p = 6.61 \cdot 10^{-42}$

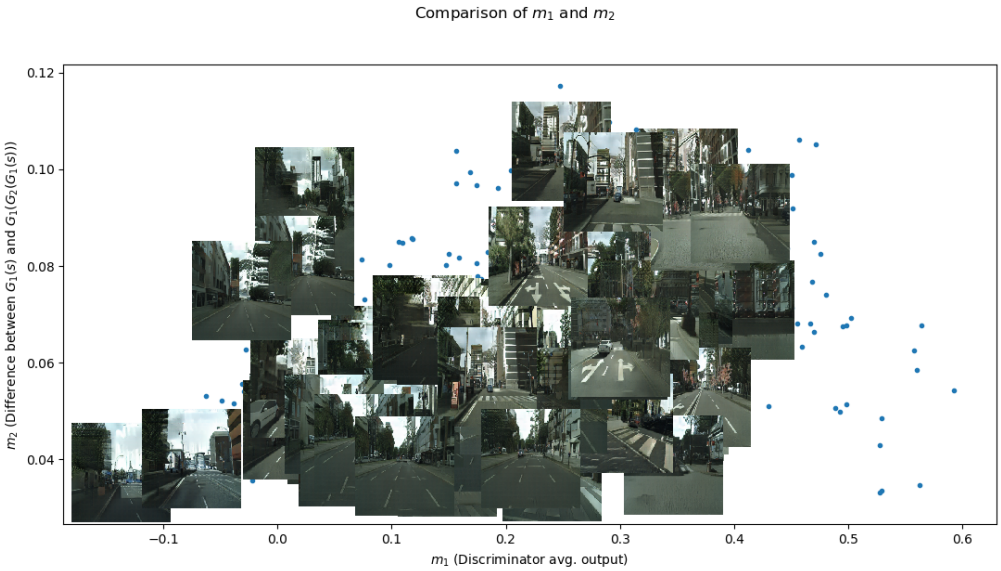
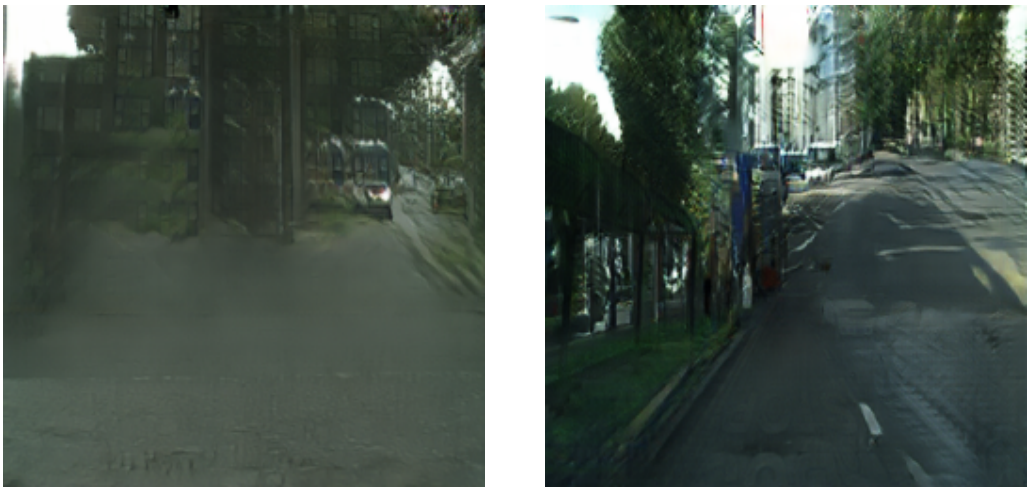


Figure 4.8: Scatter plot of the model results with respect to m_1 and m_2. Some of such points are replaced with the corresponding refined image to better visualize the results

4.2.1.3 Third model

For the third model, the resulting average m_1 score is $\mu_{m_1} = 0.31897$, with a standard deviation of $\sigma_{m_1} = 0.09792$. Some of examples of refined images with high and low scores are shown in Figure 4.9. Contrary to the previous cases, the Pearson correlation coefficient is $PCC = -0.35$ with corresponding p-value $p = 8.114 \cdot 10^{-45}$ as can be seen in figure 4.10



(a) Examples of generated images with high m_1 (respectively 0.5803 and 0.4664)

(a) Examples of generated images with low m_1 (respectively 0.0595 and 0.0137)

Figure 4.9: Some of the output images with the highest or lowest m_1

3.13

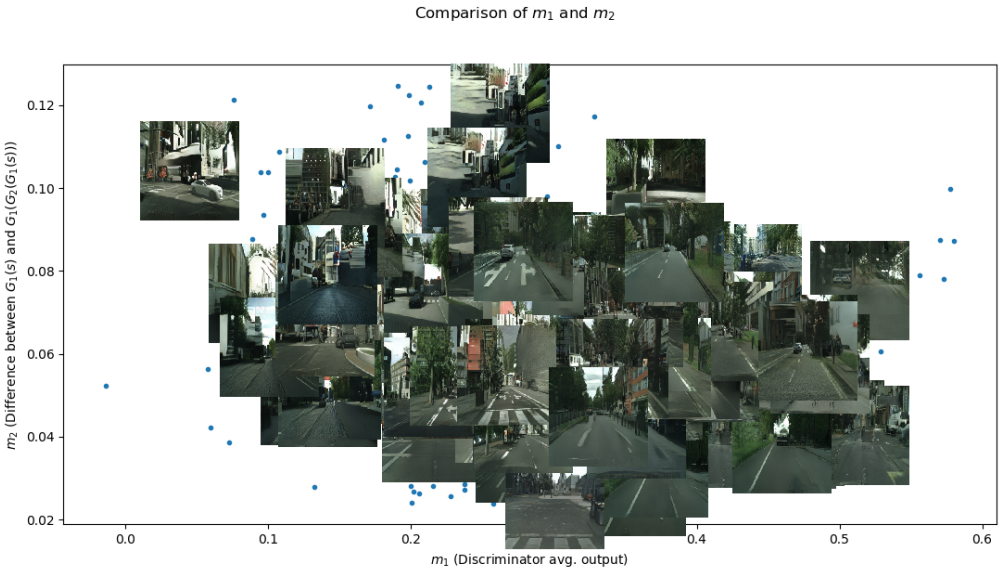


Figure 4.10: Some of the output images with the highest or lowest m_1

4.2.1.4 Fourth model

For the third model, the results differ from the previous ones as resulting average m_1 score is $\mu_{m_1} = 0.08962$, with a standard deviation of $\sigma_{m_1} = 0.07845$. Some of examples of refined images with high and low scores are shown in Figure 4.11. The Pearson correlation coefficient is $PCC = 0.74$ with corresponding p-value $p = 2.228 \cdot 10^{-150}$, such results are shown in Figure 4.12
3.13

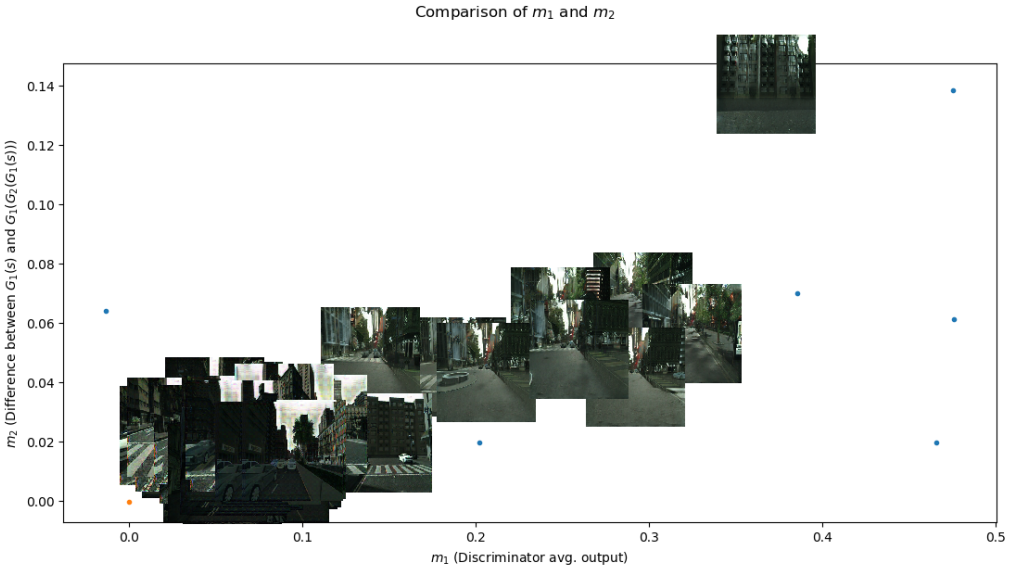


Figure 4.12: Scatter plot of the model results with respect to m_1 and m_2. Some of such points are replaced with the corresponding refined image to better visualize the results. Most of the samples are grouped in the bottom left area.

(a) Examples of generated images with high m_1 (respectively 0.4759 and 0.3855)

(a) Examples of generated images with low m_1 (respectively 0.0481 and 0.0231)

Figure 4.11: Some of the output images with the highest or lowest m_1

4.2.2 Segmentation metrics

Another possible approach proposed is to assess the quality of the refined image based on the segmentation result of a segmentation algorithm trained on the real dataset (using its ground truth semantic labels), as described in Section 3.4.3.

As a further possible term of comparison, such methodology is applied also to the synthetic images. For such case, the average accuracy is $m_3 = 0.7079.$, while the single per-class accuracy is shown below in Table 4.1 :

Segmentation Accuracy	
Class	Accuracy
Sky	0.7684
Road	0.7890
Building	0.8335
Car	0.2666
Pedestrian	0.1795
Vegetation	0.3640

Table 4.1: Resulting segmentation accuracy on the synthetic images for each class

Such process is repeated for the baseline model as well for the four different model previously introduced. The results can be seen in Table 4.2

Pixel Segmentation Accuracy					
Class	Baseline (I^{n}2I)	First	Second	Third	Fourth
Total (m_3)	0.5614	0.61817	0.5716	0.5514	0.6321
Sky	0.1899	0.5139	0.4126	0.3540	0.1105
Road	0.9715	0.9507	0.9196	0.9070	0.8370
Building	0.5132	0.5873	0.4610	0.4881	0.8478
Car	0.2569	0.3578	0.4453	0.3939	0.4260
Pedestrian	0.0947	0.1027	0.1154	0.2163	0.1957
Vegetation	0.3185	0.4387	0.5468	0.5840	0.3799

Table 4.2: Resulting segmentation accuracy for the different models

Overall, there is not an evident correlation between the previous metrics m_1,m_2 with the segmentation accuracy, as shown for example in Figure 4.13 for the fourth model.

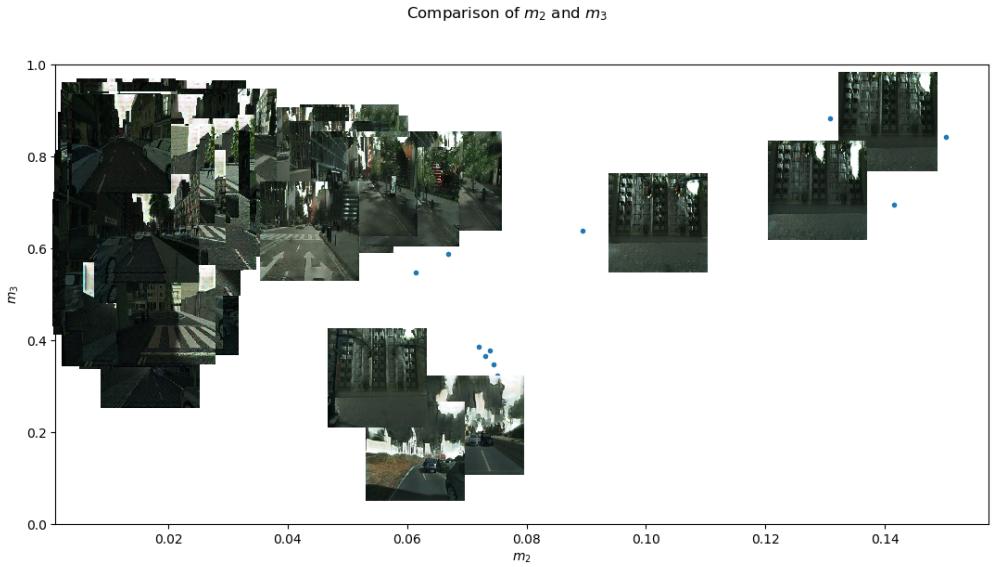


Figure 4.13: Scatter plot of the model results with respect to m_2 and m_3. Some of such points are replaced with the corresponding refined image to better visualize the results. Most of the images lies on the far left of the graph.

4.2.3 Classifier based metrics

As expected, the classification algorithm is able to correctly discriminate the refined images from the real ones in all the cases, although in some cases with higher accuracy, as shown in Table 4.3

Average accuracy	
Model	Average accuracy
Synthetic	0.997
First	0.954
Second	0.956
Third	0.964
Fourth	0.972

Table 4.3: Average classification algorithm accuracy

Furthermore, the classification algorithm output in all the four model reflects the discriminator measure m_1 not bringing further information about the image quality.

4.2.4 Mean opinion score

In order to obtain the Mean Opinion Score, a total of fifty-two synthetic-refined image pairs have been asked to be judged by a diverse pool of people.
In total, 84 people participated to the study. Such group of people varies both in age and occupation. Table 4.4 shows the average MOS for the different models, while Figure 4.14 represents

such results with respect to m_2.

Models Mean Opinion Score		
Model	$\text{MOS}_{average}$	MOS_{std}
First	3.1692	0.3413
Second	2.8281	0.3857
Third	2.8119	0.2948
Fourth	3.4984	0.3887

Table 4.4: Mean opinion score averages (MOS) for the different models

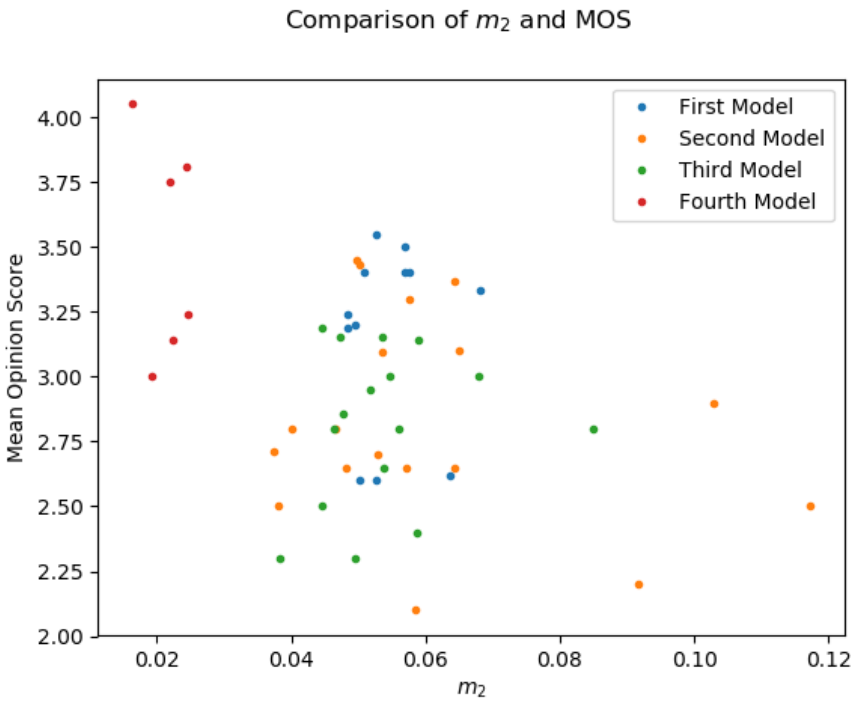


Figure 4.14: Scatter plot of the refined image samples which have been judged by humans. On a scale from 1 to 5, 1 corresponds to a bad result, while 5 to a refined image that looks real (while being coherent with the synthetic counterpart).

4.3 Discussion of the results

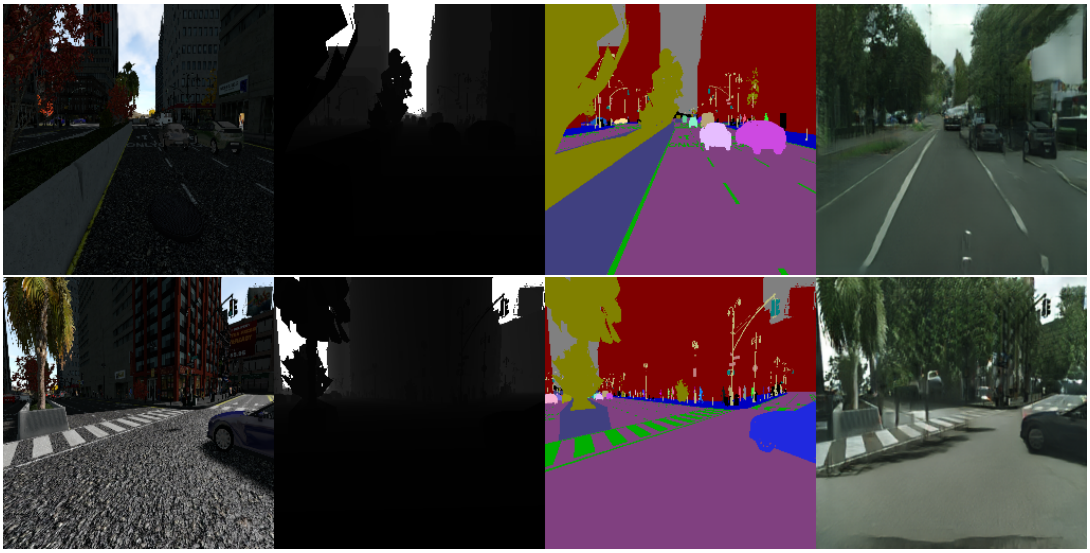
4.3.1 Models comparison

The differences between the models are only based on the loss formulation, and it is possible to see how certain constrains bring to specific outcomes.
The first model, being the least constrained among the four settings, attributes a bigger importance to the output of the discriminator. Reasonably, as can be seen by the average m_1 measure μ_{m1}, the generator (of the first GAN) learns reasonably well to deceive the corresponding discriminator.
However, once imposed the self regularization term, the generator is forced to match closer the synthetic image and the corresponding m_1 values for all the following models decrease. In particular in the fourth model, as such constrains are strengthened, the generator output is always correctly classified as not-real by the discriminator ($\mu_{m1} = 0.08962$).
Analogously, compared to the baseline In2I the results of the proposed models are more coherent with the corresponding synthetic images, as can be seen in Table 4.2 and visually, although as previously discussed this partially impacts the performance on the perceptual loss.
The weighted term addition in the third model has minor repercussions on its behaviour. Mainly, smaller classes objects (such as poles, pedestrian and cars) are less modified while bigger ones, like building and sky, allow for heavier changes.

Overall, it seems to consistently be an underlying trade-off between the fidelity with respect to the original synthetic image and the ability to deceive the discriminator.
A related problem is also the well-known tendency of such GAN models to insert artifacts and unwanted objects in the image. In fact, due to the different classes distribution of the real dataset, the generator is pushed by the discriminator to better match the real images distribution through the addition of unwanted objects or features. In the case in exam, the most notable byproduct of such behaviour is the addition of building in the sky (as usually in the real images only a small portion of the image is sky) and the modification of building to vegetation (as this is a predominant class the in the real dataset but less in the synthetic one). Figure 4.15 shows some examples of refined images with extensive vegetation addition.

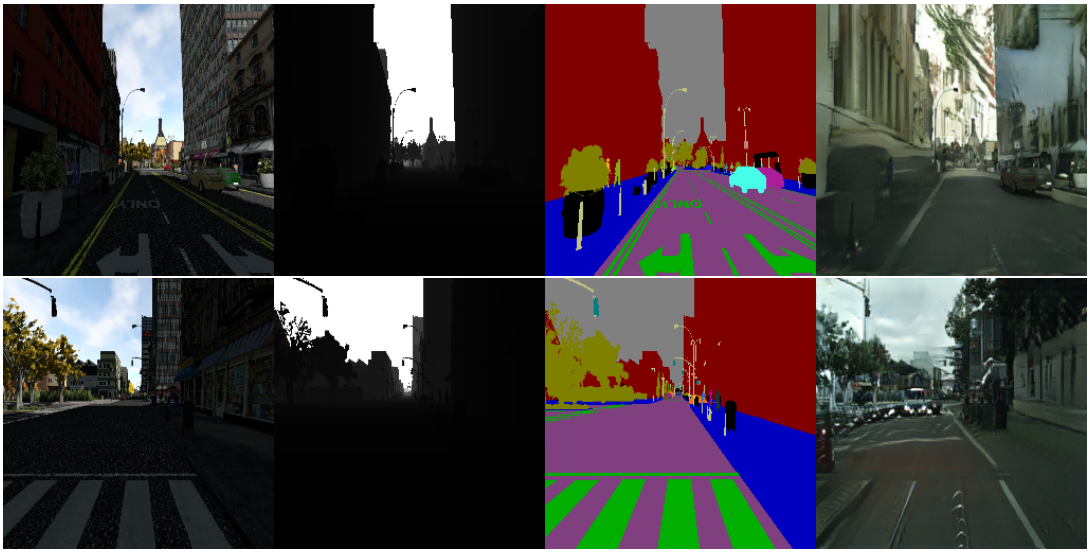
Furthermore, especially for the first three models, another problem to keep in account is the robustness of such generative processes. In fact, the refinement process might sometimes yield to confused images. Such behaviour, shown for example in Figure 4.16, might be attributed to the limited generalizability capacity of generator.

The latter two behaviours are limited in the fourth model by the strong self regularization (and gradient-sensitive) loss. However, in this case, a common problem arising is the addition of finer patter in the sky which are not penalized much by either the self-regularization or gradient sensitive losses, but hinder the segmentation process as those parts are wrongly labeled as building. A possible explanation is that such behaviour is consistent with the attempt of the generator to fill the sky with building-like objects but, due to the $L1$ loss, it can just insert lower-level features of buildings. Figures 4.17 and 4.19 show some example of this phenomenon. In fact, neglecting the cases in which a pixel is wrongly classified as building, the segmentation accuracy of the fourth model would rise to $m_3 = 0.76731$ (outperforming the synthetic images



Synthetic image Depth map Semantic labels Refined Image

Figure 4.15: Samples of results of the trained models (respectively first and third model) in which there is a heavy vegetation modification.



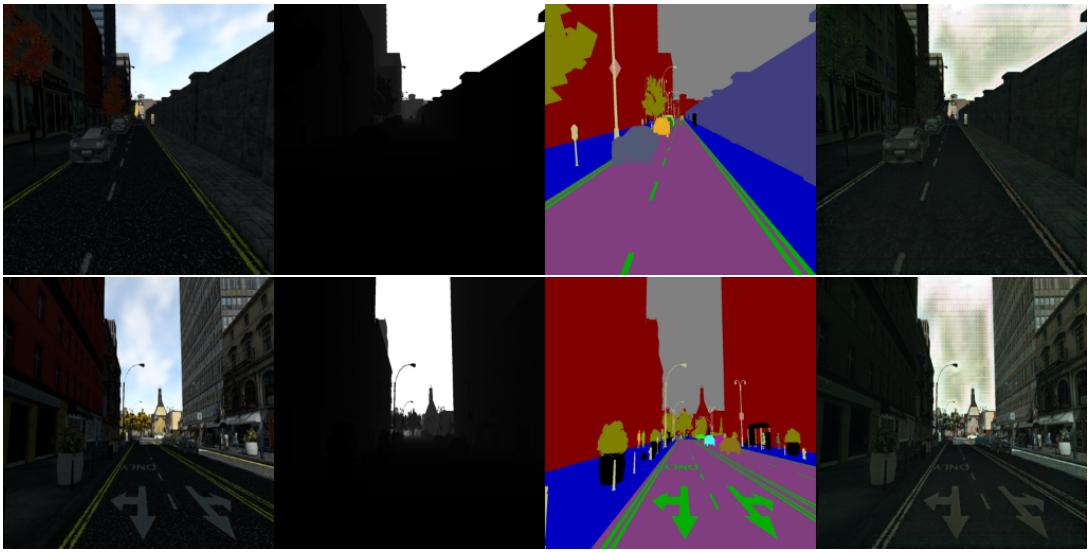
Synthetic image Depth map Semantic labels Refined Image

Figure 4.16: Samples of results of trained models (respectively third and first model) in which there is a heavy modification of the image resulting in a chaotic output.

case).

4.3.2 Quality metrics

It is easy to note from Figures like 4.7 or 4.9 that an high m_1 does not always mean a good quality of the result. In fact, some of the highest m_1 refined images are either chaotic or completely not coherent with the corresponding synthetic image.



Synthetic image | Depth map | Semantic labels | Refined Image

Figure 4.17: Samples of results of the fourth model in which in the image refined there are artifacts in the sky.

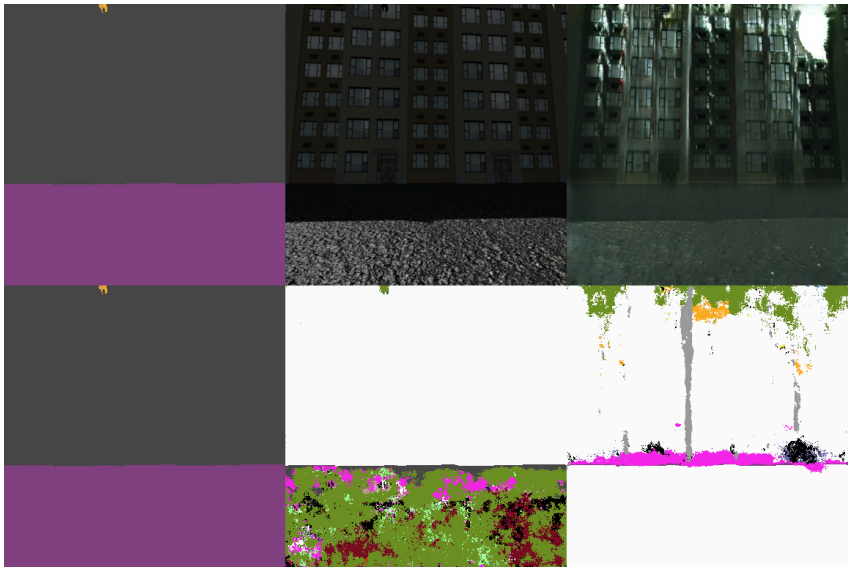


Figure 4.18: Top row: from left to right, the semantic labels map, the synthetic and the refined image. Bottom row, the semantic labels map, the segmentation comparison (w.r.t. the ground truth) of respectively the synthetic and real images. The white color means the segmentation result agrees with the ground truth, otherwise the pixel is colored with the mislabeled class.

Moreover, the highest mean opinion scores are achieved by the fourth model, whose m_1 measures are the lowest among the four settings.

A better measure of the similarity between the synthetic image and the refined one can be obtained using the segmentation accuracy, that it m_3, but some points have to be discussed. First, it is highly dependant on the input semantics. Certain semantics configurations are easier to

extract than others, for example when just a class occupies the whole image, whereas an input scene with many small objects will likely yield a worse segmentation accuracy. Figure 4.18 is an example of bad-quality refined image which, due to simple semantics gets almost entirely correctly classified. Furthermore, as already discussed in the previous chapter, the choice and

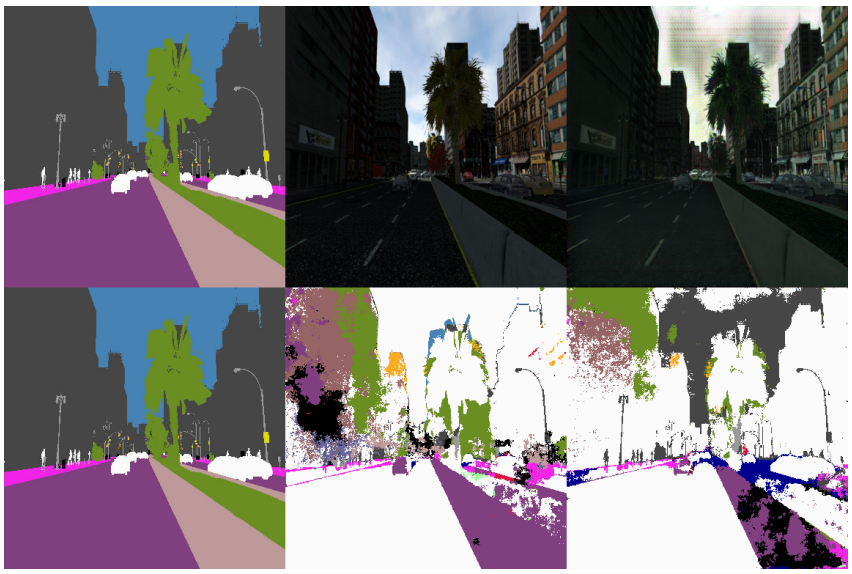


Figure 4.19: Top row: from left to right, the semantic labels map, the synthetic and the refined image. Bottom row, the semantic labels map, the segmentation comparison (w.r.t. the ground truth) of respectively the synthetic and real images. The white color means the segmentation result agrees with the ground truth, otherwise the pixel is colored with the mislabeled class.Note the sky in the refined image is almost completely wrongly classified as building, due to the finer artifacts

training of the segmentation algorithm is critical to achieve a meaningful segmentation accuracy measure. In this case, the segmentation algorithm accuracy on the trained dataset was high on large classes like building,sky and vegetation, but was not accurate on smaller instances like poles, cars or pedestrian. Thus, it might be inaccurate to attribute the cause of the bad segmentation performance on the refinement process.
It is also important to note that the size of the image is critical for the segmentation process. In this work, it has been mainly performed image-to-image translation on small-sized images (256x256 pixels). However, the same fourth model trained on a bigger scale (384x384) yields to an higher accuracy in most of the classes (total accuracy: 0.6751) and usually segmentation processes are performed on bigger images.

Broadly speaking, the segmentation accuracy measure highlights the overall difficulty of the model to correctly attribute at each class its right essence. This is due to the lack of semantic labels for the real images, which only allow the network to learn in an unsupervised way.
Such limit is also notable in the discriminator D_{24} performance, which outperforms the second GAN generator in all four models. In other words, the second generator is not able to correctly learn to segment a real image. It is an understandable outcome, as unsupervised segmentation is still a difficult and mainly unsolved problem.

While not a solid measure of the quality of the image, the reconstruction difference measure m_2 seems to correctly identify instances in which big artifacts are introduced, like the one in Figure 4.18 and other instances are notable in Figure 4.13. In fact, Figure 4.14 suggests that high m_2 values are correlated to low visual-quality images, hence such measure could be considered as an upper-bound limit, however the samples are too small to draw any strong conclusion.

5
Conclusion

In this thesis, different GAN models for synthetic-to-real transformation have been developed and their results have been analyzed through the use of different metrics.
The results for such models vary in perceptual quality and semantics preservation, showing how the fine tuning of different parameters might enhance different model behaviours.

Overall, the quality of the results are on pair with the state-of-the-art approach and offer the user more control on the output semantics and characteristics. The addition of specific input modalities are leveraged (namely semantic labels map) to obtain the desired result.

A single metrics capable of reproducing the human judgment has not been found, rather different measure proposal have been proposed and it has been shown how such metrics can give an insight on different characteristics of the refined image.

5.1 Future work

The unsupervised approach of such models offer great flexibility and will be continued to be developed as it has shown promising results. Different modification and additions could be made.
First, more work could be done with respect to the architecture choice of both the generator and discriminators. Regularization techniques like dropout or normalization procedures could be tested as they shown promising results in contiguous areas.

As mentioned in the previous chapter, the segmentation measure m_3 strongly depends on the image size and the segmentation algorithm. Thus, it might be of interest to train the GAN models using bigger images, along with testing different segmentation algorithm architectures.

Finally, it is of great importance to further test the proposed model on different datasets to study its behaviour and assess the improvements brought in the case in exam. Along with this, a Mean Opinion Score test can be performed at a wider scale to have more informative and precise results.

Bibliography

[1] Frisk, D. (2016) A Chalmers University of Technology Master's thesis template for LaTeX. Unpublished.

[2] Learning from Simulated and Unsupervised Images through Adversarial Training, https://arxiv.org/abs/1612.07828

[3] GAN-Generative Adversarial Nets. https://medium.com/@sairajreddy/gan-generative-adversarial-nets-e8520157ec62

[4] Virtual Worlds as Proxy for Multi-Object Tracking Analysis, https://arxiv.org/abs/1605.06457

[5] Image Quilting for Texture Synthesis and Transfer, A. A. Efros, W. T. Freeman, https://people.eecs.berkeley.edu/ efros/research/quilting/quilting.pdf

[6] Image Analogies, A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, D. H. Salesin, https://mrl.nyu.edu/publications/image-analogies/analogies-72dpi.pdf

[7] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[8] Real-Valued Medical Time Series Generation With Recurrent Conditional GANs Stephanie L. Hyland,Cristóbal Esteban, Gunnar Rätsch

[9] H. B. Mitchell, Image Fusion: Theories, Techniques and Applications.

[10] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A.Ng, Multimodal deep learning

[11] G. Pajares and J. M. de la Cruz. A wavelet-based image fusion tutorial

[12] P. Perera, M. Abavisani, and V. Patel, I n2I : Unsupervised Multi-Image-to-Image Translation Using Generative Adversarial Networks

[13] Methodology for the subjective assessment of the quality of television pictures

[14] Unsupervised Image-to-Image Translation Networks Ming-Yu Liu, Thomas Breuel, Jan Kautz

[15] Coupled Generative Adversarial Networks Ming-Yu Liu, Oncel Tuzel

[16] VQEG, Final report from the video quality experts group on the validation of objective models of video quality assessment, phase ii

[17] K. Thung, P. Raveendran A Survey of Image Quality Measures

[18] Z. Wang, A.C. Bovik, H.R. Sheikh,E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity

[19] Vipin Kamble, K.M. Bhurchandi 1 No-reference image quality assessment algorithms: A survey

[20] Xiaoqiao Chen, Qingyi Zhang , Manhui Lin , Guangyi Yang and Chu He No-Reference Color Image Quality Assessment: From Entropy to Perceptual Qualit https://arxiv.org/pdf/1812.10695.pdf

[21] Khosro Bahrami and Alex C. Kot A Fast Approach for No-Reference Image Sharpness Assessment Based on Maximum Local Variation

[22] Deep Convolutional Neural Models for Picture-Quality Prediction Jongyoo Kim, Hui Zeng, Deepti Ghadiyaram, Sanghoon Lee, Lei Zhang, and Alan C. Bovik

[23] Semantic-aware Grad-GANfor Virtual-to-Real Urban Scene Adaption Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P. Xing

[24] Sensor Modelling with Recurrent Conditional GANs, Henrik Arnelid

[25] Generative Adversarial Nets Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley,Sherjil Ozair,Aaron Courville, Yoshua Bengio

[26] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, Wenzhe Shi

[27] The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes German Ros, Laura Sellart , Joanna Materzynska, David Vazquez , Antonio M. Lopez

[28] Playing for Data: Ground Truth from Computer Games Stephan R. Richter, Vibhav Vineet, Stefan Roth, Vladlen Koltun

[29] Semantic Object Classes in Video: A High-Definition Ground Truth Database, Gabriel J. Brostow and Julien Fauqueur and Roberto Cipolla

[30] The Cityscapes Dataset for Semantic Urban Scene Understanding Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, Bernt Schiele

[31] Adam: A Method for Stochastic Optimization, Diederik P. Kingma, Jimmy Lei Ba

[32] Understanding the difficulty of training deep feedforward neural networks Xavier Glorot, Yoshua Bengio

[33] Image-to-Image Translation with Conditional Adversarial Networks Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros

[34] Precomputed Real-Time Texture Synthesis withMarkovian Generative Adversarial Networks Chuan Li, Michael Wand

[35] Learning from Simulated and Unsupervised Images through Adversarial Training, Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, Russ Webb

[36] Learning Representations for Automatic Colorization, Gustav Larsson, Michael Maire, and Gregory Shakhnarovich

[37] Colorful Image Colorization, Richard Zhang, Phillip Isola, Alexei A. Efros

[38] Photographic Image Synthesis with Cascaded Refinement Networks Qifeng Chen, Vladlen Koltun

[39] A Neural Algorithm of Artistic Style Leon A. Gatys, Alexander S. Ecker, Matthias Bethge

[40] Unsupervised Image Translation, Ròmer Rosales, Kannan Achan, Brendan Frey

[41] High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs, Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, Bryan Catanzaro

[42] Medical Image Synthesis for Data Augmentation and Anonymization using Generative Adversarial Networks, Hoo-Chang Shin, Neil A Tenenholtz, Jameson K Rogers, Christopher G Schwarz, Matthew L Senjem, Jeffrey L Gunter, Katherine Andriole, and Mark Michalski

[43] Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros
[44] The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation Simon Jegou, Michal Drozdzal, David Vazquez, Adriana Romero, Yoshua Bengio
[45] Distinguishing Between Natural and Computer-Generated Images Using Convolutional Neural Networks Weize Quan, Kai Wang, Dong-Ming Yan , Xiaopeng Zhang