T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

THE DEPARTMENT OF COMPUTER ENGINEERING

ENHANCING IMAGE-TO-IMAGE TRANSLATION: A NOVEL CONDITIONAL GENERATIVE ADVERSARIAL NETWORK APPROACH WITH U-NET AND RESNET COMBINATION

MASTER'S THESIS

KHALED AL HARIRI

ISTANBUL 2024

T.C.

BAHCESEHIR UNIVERSITY

GRADUATE SCHOOL

THE DEPARTMENT OF COMPUTER ENGINEERING

ENHANCING IMAGE-TO-IMAGE TRANSLATION: A NOVEL CONDITIONAL GENERATIVE ADVERSARIAL NETWORK APPROACH WITH U-NET AND RESNET COMBINATION

MASTER'S THESIS

THESIS ADVISOR
ASSIST. PROF. ERKUT ARICAN

ISTANBUL 2024

T.C.

BAHÇEŞEHİR UNIVERSITY

GRADUATE SCHOOL

MASTER THESIS APPROVAL FORM

Program Name:	COMPUTER ENGINEERING
Student's Name and Surname:	Khaled AL HARIRI
Name of The Thesis:	Enhancing Image-to-Image Translation: A Novel Conditional Generative Adversarial Network Approach with U-Net and ResNet Combination
Thesis Defense Date	3rd of June 2024

This thesis has been approved by the Graduate School which has fulfilled the necessary conditions as Master thesis.

Assoc. Dr. Yücel Batu SALMAN

Institute Director

This thesis was read by us, quality and content as a Master's thesis has been seen and accepted as sufficient.

	Title, Name	**Institution**	**Signature**
Thesis Advisor:	Assist. Prof. Erkut ARICAN	BAHÇEŞEHİR UNIVERSITY	
2nd Member	Assist. Prof. Tarkan AYDIN	BAHÇEŞEHİR UNIVERSITY	
3rd Member (Outside Institution)	Assist. Prof. Sinem AKYOL	FIRAT UNIVERSITY	

I hereby declare that all information in this document has been obtained and presentedin accordance with academic rules and ethical conduct. I also declare that, as requiredby these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Khaled AL HARIRI

Signature:

ABSTRACT

Enhancing Image-to-Image Translation:
A Novel Conditional Generative Adversarial Network Approach
with U-Net and ResNet Combination

Khaled, AL HARIRI

Master's Program in Computer Engineering

Thesis Advisor: Assist. Prof. Erkut ARICAN

May 2024, 63 pages

Image-to-image translation is a very important task in computer vision that allows transformations between different image domains, allowing for many unique types of applications such as style transfer and image enhancement. The methodology used in this paper includes a novel conditional generative adversarial network model with a pair of generator and discriminator in which the generator has an architecture that combines U-Net and ResNet while the discriminator has an architecture that uses PatchGAN. The performance of the model was evaluated using highly known evaluation metrics such as SSIM and PSNR. Furthermore, we compared the results of the model with other studies using the same evaluation metrics and also by conducting a public survey for human visual assessment in which participants voted for the image that looked most similar to the target. The results have shown that our model exceeds other methods in both the evaluation metrics and the public survey, proving the effectiveness of our image-to-image translation model.

Keywords: Image-to-Image Translation, Computer Vision, Deep Learning, Conditional Generative Adversarial Networks

ÖZ

Görüntüden Görüntüye Çevirinin Geliştirilmesi:
U-Net ve ResNet Kombinasyonu ile Yeni Bir Koşullu
Üretken Çekişmeli Ağ Yaklaşımı

Khaled, AL HARIRI

Bilgisayar Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi. Erkut ARICAN

Mayıs 2024, 63 sayfa

Görüntüden görüntüye çeviri, bilgisayarlı görmede, farklı görüntü alanları arasında dönüşümlere izin veren, stil aktarımı ve görüntü iyileştirme gibi birçok benzersiz uygulama türüne olanak tanıyan çok önemli bir görevdir. Bu çalışmamız, U-Net ve ResNet'i birleştiren mimariye sahip çift-üretici ve PatchGAN kullanan mimariye sahip bir ayırıcının olduğu, yeni bir koşullu üretken çekişmeli ağ modelini içerir. Modelimizin performansı SSIM ve PSNR gibi çok bilinen değerlendirme metrikleri kullanılarak değerlendirilmiştir.. Ayrıca, modelin sonuçlarını aynı değerlendirme metriklerini kullanan diğer çalışmalarla karşılaştırılmasının yanı sıra katılımcıların hedef görüntüye en çok benzeyen görüntüyü seçtikleri anket aracılığıyla da değerlendirilmiştir. Sonuçlar, modelimizin hem değerlendirme ölçütlerinde hem de anket sonucunda diğer yöntemlerden daha iyi sonuçlar vererek görüntüden görüntüye çeviri modelimizin etkinliğini kanıtlamıştır.

Anahtar Kelimeler: Görüntüden Görüntüye Çeviri, Bilgisayarla Görü, Derin Öğrenme, Koşullu Üretken Çekişmeli Ağlar

I dedicate this thesis to my dear and beloved parents. Their encouragement and belief in my potential was extremely important in my journey of pursuing higher education. I am eternally grateful for all the sacrifices they have made for me and their constant belief in my skills and abilities. I also thank my advisor, Assist. Prof. Erkut Arıcan, for his immense help in guiding me throughout my academic career.

With much gratitude,

ACKNOWLEDGMENTS

I want to express my heartfelt gratitude to all of the people that helped me throughout my academic career and led me to this moment of pursuing higher education and writing my master's thesis.

I would like to thank my beloved parents for their support throughout my life and my academic career. Their encouragement and understanding played a vital role in me pursuing higher education. I want to thank you for constantly being my pillar of support that I could rely on and for all of the sacrifices they have made for me.

I would like to also thank my advisor, Assist. Prof. Erkut Arıcan, for his mentorship and constant support in my academic journey and in my master's thesis. Without his mentorship and support, this master's thesis would not have been the same.

With much gratitude,

TABLE OF CONTENTS

ETHICAL CONDUCT iii

ABSTRACT iv

ÖZ v

ACKNOWLEDGMENTS vii

TABLE OF CONTENTS viii

LIST OF TABLES ix

LIST OF FIGURES xi

LIST OF ABBREVIATIONS xiii

Chapter 1 Introduction 1

1.1 Computer Vision and Image-to-Image Translation 1

1.2 Deep Learning Architectures 2

1.3 Generative Adversarial Networks 3

1.4 Potential of Image-to-Image Translation 3

1.5 Possible Challenges 4

1.6 Objectives and Motivation 5

1.7 Scope of Study 5

Chapter 2 Literature Review 6

2.1 Review 7

Chapter 3 Methodology 25

3.1 Data Collection 26

3.2 Data Pre-processing 28

3.3 Model Architecture 30

3.3.1 Generator network architecture. 30

3.3.2 Discriminator network architecture. 34

3.3.3 Training process. 37
Chapter 4 Results & Evaluation Metrics 39
4.1 Results 39
4.2 Quantitative Evaluation 41
4.2.1 Structural similarity index (SSIM). 42
4.2.2 Peak signal-to-noise ratio (PSNR). 43
4.2.3 Evaluation metrics calculation process. 43
4.2.4 Quantitative evaluation results 44
4.3 Training Loss Results 45
4.4 Comparing Results with Original Study 47
4.4.1 Comparing presentation results. 47
4.4.2 Comparing evaluations metrics. 50
4.4.3 Comparing training loss results. 51
4.5 Survey Results 52
4.5.1 Structure of the survey. 53
4.5.2 Results of the survey. 54
4.6 Results Discussion & Interpretation 56
Chapter 5 Conclusion 57
5.1 Key Findings & Contributions 57
5.2 Limitations & Future Work 57
5.3 Closing Remarks 58
REFERENCES 59

LIST OF TABLES

TABLES

Table 1. Classification Performance of the SPA-GAN (Emami et al., 2020) 18
Table 2. Network Structure of DMDIT Framework (Sho et al., 2021) 19
Table 3. Information of Datasets Used (University of California, 2018) 27
Table 4. Comparison of Evaluation Metrics with Isola et al. (2017) 51
Table 5. Comparison of Training Loss with Isola et al. (2017) 52
Table 6. Comparison of Survey Results with Isola et al. (2017) 56

LIST OF FIGURES

FIGURES

Figure 1. Comparison of resolutions in image synthesis (Odena et al., 2017) 8
Figure 2. U-Net architecture (Isola et al., 2017) 8
Figure 3. AdGAN model with six generators (Gan et al., 2018) 10
Figure 4. Multiple cyclic consistency loss with L1 loss (Cho et al., 2019) 10
Figure 5. InstaGAN (instance-aware GAN) architecture (Mo et al., 2019)..... 11
Figure 6. InjectionGAN diversity comparison with BicycleGAN (Xu et al., 2019) . 13
Figure 7. Overview of StarGAN v2 structure (Choi et al., 2020)..... 14
Figure 8. Overview of MedGAN framework with CasNet (Armanious et al., 2020) 15
Figure 9. CycleGAN mappings and cycle consistency losses (Zhu et al., 2020) 16
Figure 10. RealDRR framework for DRR rendering (Dhont et al., 2020)..... 17
Figure 11. Combination of image denoising and translation (Yan et al., 2021)..... 20
Figure 12. Structure of URCA-GAN for image translation (Nie et al., 2021)..... 21
Figure 13. Comparison between Liu et al. and other models (Liu et al., 2021)..... 22
Figure 14. Generator architecture in MISS GAN (Brazilay et al., 2021) 23
Figure 15. Samples of training dataset (University of California, 2018)..... 28
Figure 16. Showcasing dataset after preprocessing 29
Figure 17. Architecture of our generator with U-Net and ResNet..... 33
Figure 18. Architecture of our discriminator with PatchGAN..... 36
Figure 19. Example of training loop output..... 38
Figure 20. Example of our generator results with facades dataset..... 40
Figure 21. Example of our generator results with maps dataset 41
Figure 22. Equations for luminance, contrast, and structure (Wang et al., 2004)..... 42
Figure 23. Equation for calculating SSIM index (Wang et al., 2004)..... 42
Figure 24. SSIM measurement system diagram (Wang et al., 2004)..... 43
Figure 25. Equations for MSE and PSNR (Stéphane, 2009) 43
Figure 26. Line graph of our SSIM values over training steps..... 45
Figure 27. Line graph of our PSNR values over training steps 45
Figure 28. Line graph of our generator loss over training steps 46

Figure 29. Line graph of our discriminator loss over training steps 47
Figure 30. Comparison of results with Isola et al. On facades dataset 48
Figure 31. Comparison of results with Isola et al. On maps dataset 49
Figure 32. Line graph of Isola et al. SSIM values over training steps 50
Figure 33. Line graph of Isola et al. PSNR values over training steps 50
Figure 34. Line graph of Isola et al. Generator loss over training steps 51
Figure 35. Line graph of Isola et al. Discriminator loss over training steps 52
Figure 36. Example showcasing our survey structure 54
Figure 37. Survey results (percentage of total answers) 55
Figure 38. Survey results (number of favored questions) 55

LIST OF ABBREVIATIONS

CV	Computer Vision
AI	Artificial Intelligence
DL	Deep Learning
ML	Machine Learning
CNN	Convolutional Neural Network
GAN	Generative Adversarial Network
DGM	Deep Generative Model
cGAN	Conditional Generative Adversarial Network
DRR	Digitally Reconstructed Radiograph
MB	Megabyte
KB	Kilobyte
ResNet	Residual Neural Network
ReLU	Rectified Linear Unit
Tanh	Hyperbolic Tangent
SSIM	Structural Similarity Index
PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Squared Error

Chapter 1

Introduction

1.1 Computer Vision and Image-to-Image Translation

In this digital age where many innovations and revolutions has occurred, few areas have witnessed as rapid and profound a transformation as Computer Vision (CV). CV is one of the most prominent fields of Artificial Intelligence (AI) in which useful information is gathered from visual inputs (e.g., videos and images) that allow computers to perform an action based on the gathered information (IBM, 2021). Basically, CV grants computers the capability of visual perception therefore making them understand visual cues and perform certain tasks according to what they understood. In order to achieve this, large chunks of image data are needed to allow the model to analyze the contents of the images thus making it able to distinguish between these images (IBM, 2021). Deep Learning (DL), a subset of Machine Learning (ML) that allows computers to learn and perceive, is one of the main technologies used in this process of implementing CV (IBM, 2021). The field of "image-to-image translation" is one area of CV that has seen a lot of growth and interest. By examining its methods, difficulties, and applications, this study aims to explore and dig further into this field and to potentially further revolutionize the way images are perceived and transformed.

According to Isola et al. (2017), the phrase "translating" can be used to describe most of the problems solved by CV in which an input image is translated into a corresponding output image. The study also added that given a large enough dataset, images can be shown in many different representations by translating it to the desired scene of the output image using "image-to-image translation". In other words, the input image is mapped from one domain to the domain of the output image while focusing on specific features that highlight the image.

1.2 Deep Learning Architectures

As stated by Mohammed et al. (2022), Convolutional Neural Networks (CNNs) are considered a DL architecture that primarily focuses on visual analysis and CV tasks while working with image data. In order to achieve the desired results, CNNs work by minimizing a loss function which is a function that is related to the quality of the achieved results (Isola et al., 2017; Mohammed et al., 2022). However, while the process in which a CNN model learns to minimize the loss is done automatically, manual labor is still needed in order to inform the CNN model about what we want to be minimized in our loss function which could potentially result in human error thus resulting in undesired outcomes such as blurry images (Isola et al., 2017). Due to this, it is unadvised to use CNNs for tasks such as image-to-image translation due to how difficult it would be to exactly tell the model what the loss function should be minimizing without the assistance of an expert in the field (Isola et al., 2017). The obvious solution to this problem is to automate the designing process of the loss function so that it is appropriate and works well with the given task the same as how the learning process of the CNNs is automated.

Goodfellow et al. (2014) introduced Generative Adversarial Networks (GANs), which are another DL architecture that is also a Deep Generative Model (DGM) which is generally able to generate data that is very similar to a real dataset and differentiate between fake and real data. The loss function used in GANs adapt to the data thus making it usable in many different types of applications even though each of these applications would require a different loss function if they were using CNNs or similar architectures (Isola et al., 2017). However, GANs are typically used for normal image generation and are not fully suited for a task such as image-to-image translation (Kamil & Shaikh, 2019). Due to this, Conditional GANs (cGANs) are used instead, an extension of GANs, which places a condition on the input image in order to generate an output image that matches that condition (Isola et al., 2017; Kamil & Shaikh, 2019).

1.3 Generative Adversarial Networks

At their core, GANs consist of two distinct yet interconnected neural networks—the generator and the discriminator, each fulfilling their own objective (Goodfellow et al., 2014). As stated previously, GANs are able to generate fake data while also being able to differentiate between fake and real data (Goodfellow et al., 2014). In that scenario, generators are tasked with generating synthetic data, while discriminators are tasked with differentiating between the data generated by the generator and real data (Goodfellow et al., 2014). This means that the two neural networks are at odds with each other in which the generator tries to fool the discriminator, while the discriminator tries to expose the generator. Due to this competitive dynamic, each time one of them beats the other, the loser refines their skill (Goodfellow et al., 2014). For instance, if the generator fools the discriminator with fake data, then the discriminator will update itself and learn from its mistakes. The same thing happens to the generator when the discriminator is able to differentiate between the fake and real data. This cycle will continue until the generator is able to generate data that is virtually indistinguishable from real data.

GANs are very flexible due to the fact that many different architectures can be employed to build the generator and the discriminator. For instance, in the study conducted by Isola et al. (2017), they have used the U-Net architecture for the generator, and the PatchGAN for the discriminator. The choice of architecture for each of the neural networks can be crucial to the results of the model and is highly dependent on the type of data used and the objective of the user. Since this decision is very sensitive to the results, trial and error needs to be employed with many different architectures for both neural networks in order to find the best combination.

1.4 Potential of Image-to-Image Translation

What makes the topic of image-to-image translation so intriguing is the number of ways it can be applied in many different sectors and fields. For instance, it can be used to turn a sketch into a realistic image which could be a great asset to artists and even policemen when they are trying to identify a criminal. It can also be used in

architecture in which basic and simple schematics of a building's facade can be swiftly visualized in real life as a real building. Moreover, it can even be used in the field of cartography in which aerial photographs can get automatically transformed into maps thus saving hours of labor. These are all just a small number of examples out of the infinite pool of ideas that could come into life with the usage of image-to-image translation and GANs.

Yet, with all these possible applications and ideas, the field remains in its starting stage with many possibilities that still need to be explored. This is the case because we are trying to emulate an integral process that occurs in the human mind which is seamless translation of visual cues. When a human brain processes a sketch, it could imagine how it would like as a real image which is exactly what we seek to accomplish with modern algorithms instead of a human brain. These kinds of transformative feats, which would only been seen as science fiction in the past, are becoming a reality in our everyday life by setting the stage for incredible applications and innovations.

1.5 Possible Challenges

Image-to-image translation contains many challenges that need to be addressed during implementation. One of these challenges is the complexity that comes with different types of translations. For instance, some translations such as geometrical transformations are more complex than some other translations thus requiring heavier modifications than usual (Hoyez et al., 2022). Image-to-image translation requires a huge amount of data which could be difficult to gather due to the different number of implementations that could be done with this project. Not only that, but the datasets found need to be of high-quality and diversity that is sufficiently labeled. Furthermore, training the data effectively could also prove to be a tough challenge due to the dataset size, data consistency, and finding the most optimal parameter values. Computational power is also very important in implementing image-to-image translation since it is very computationally expensive, which is another challenge we could face. Ethical

concerns are also to be considered due to the nature of image-to-image translation which is able to manipulate images and faces.

1.6 Objectives and Motivation

While there has been previous literature about the subject of image-to-image translation, there will always be a need for more research and analysis due to the constant advancement of this domain and the exceedingly large number of ways it can be applied. With the way technology keeps evolving rapidly with each passing day, the need for refreshed perspectives, updated methodologies, and refined algorithms is emphasized. This is especially important for a topic such as image-to-image translation due to the numerous ways it can be applied from healthcare to even entertainment.

By deeply examining its underlying algorithms, methodologies, and neural network architectures, this thesis seeks to provide a comprehensive exploration of image-to-image translation in the context of CV. We aim to thoroughly display the potential of this domain and how it can transform the field of CV by showing its capabilities in a variety of fields and applications. Furthermore, we aim to propose a novel implementation for image-to-image translation and compare it to other methods in the same domain.

1.7 Scope of Study

This study will delve into image-to-image translation and how it relates to the realm of CV via the usage of GANs. Furthermore, we will look into a number of architectural combinations that can be used with different types of datasets that are related to image-to-image translation. We will deeply explore past studies that focused on image-to-image translation, and how our research contributes to the realm of CV and the way we process visual data. This research will also showcase a novel implementation in the field that utilizes a specific combination of architectures and evaluate it using highly known evaluation metrics. We will also compare it to other known studies in the field using a variety of comparison methods.

Chapter 2

Literature Review

The field of image-to-image translation and CV as a whole is always expanding and evolving in various sectors and with many studies and literature. Each of these studies has a different implementation and architecture that build upon each other with various use-cases. In today's digital world, visual data is extremely important and can be seen everywhere we go. This prominence explains the very high demand for methodologies that focus on manipulating and interpreting images in many different sectors and fields. Image-to-Image translation is one of the many fields in CV that stands out as one of the most important and powerful fields for image manipulation and interpretation.

As a consequence for the increasingly high demand for this field, many studies and literature are emerging that focus on harnessing the potential of visual data and how to make it as useful as possible by interpreting it correctly and manipulating it as necessary. For image-to-image translation specifically, there are many studies that utilize it in different ways to fit their needs such as for security, healthcare, navigation, architecture, and many more. Due to the extremely fast rate at which this field is expanding, more studies are always needed to keep pace with the evolving technology and to cover all of the different needs in the various sectors now and in the future.

In this section of the study, we are going to go over a multitude of studies and literature that revolve around CV and image-to-image translation. Our objective is to understand the different uses for this field, its different implementations, and how it can be compared to the implementation of our study in terms of the differences and similarities of the methods and results. We also aim to provide great insight into how image-to-image translation grew over the years with many different approaches and architectures thus shedding light on how image-to-image translation could progress and evolve in the future.

2.1 Review

In 2014, GANs were introduced by Goodfellow et al. (2014) where they were made up of 2 models pitted against each other. The generator model generates the images while the discriminator model attempts to figure out if the image is real or fake. Since this is a competition between the models, if any of the models succeed in fooling the other, the loser model adapts and improves itself accordingly. The introduction of GANs opened up the possibilities for many different image generation applications such as image-to-image translation in which many studies were created each explaining different ways of applying GANs in their image generation projects. In this section we are going to go over many of these different studies and analyze them.

GANs are a very good tool for image generation and other visual data tasks. However, image-to-image translation and other tasks as well would perform better if a condition was put in place for the input image in order to generate a corresponding output image from it. Due to this particular reason, in 2014, cGANs were introduced by Mirza & Osindero (2014), which is a conditional version of GANs in which the data given by the user is set as a condition on the 2 models. In 2016, a study published by Perarnau et al. (2016) focused on the usage of cGANs in the domain of image editing which allows them to edit images by applying a condition that depends on some arbitrary attributes via inverting the mapping of the cGAN used. By using invertible cGANs (combination of encoder and GANs), the authors were able to reconstruct and modify real images with image-to-image translation. They also evaluated their results by using an attribute predicate network for the evaluation of the cGAN and testing the modification variable in different cGAN layers. In 2017, new methods were created by Odena et al. (2017) for image synthesis that will improve the training results of GANs. By using label conditioning, the authors were able to create a new variant of GANs with image samples that have a resolution of 128x128 and a global coherence while also comparing it with other image resolutions via 2 analyses. According to their analysis, the discriminability of the 128x128 samples were more than 50% higher than the downsampled 32x32 samples that were used in the analysis.

Figure 1 shows the results of different resolutions in this study compared to each other in terms of discriminability and accuracy (Odena et al., 2017).

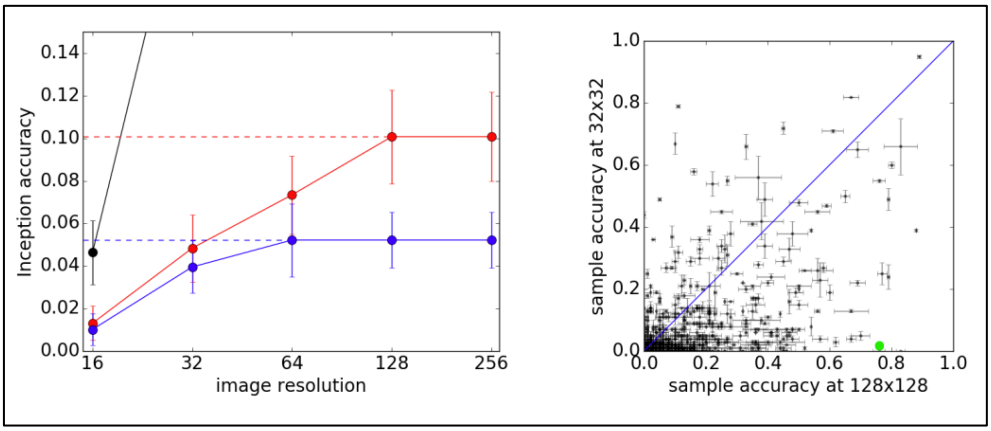

Figure 1. Comparison of resolutions in image synthesis (Odena et al., 2017)

In 2017, an implementation of image-to-image translation using cGANs was published by Isola et al. (2017) in which the U-Net architecture is used for the generator model and the PatchGAN is used for discriminator model. The U-Net architecture which is created by Ronneberger et al. (2015), allows the generator to have skip connections which is used for preserving the information between the mirrored layers from upsampling and downsampling. Figure 2 shows the U-Net architecture which is also an encoder-decoder architecture just with the added addition of skip connections (Isola et al., 2017).

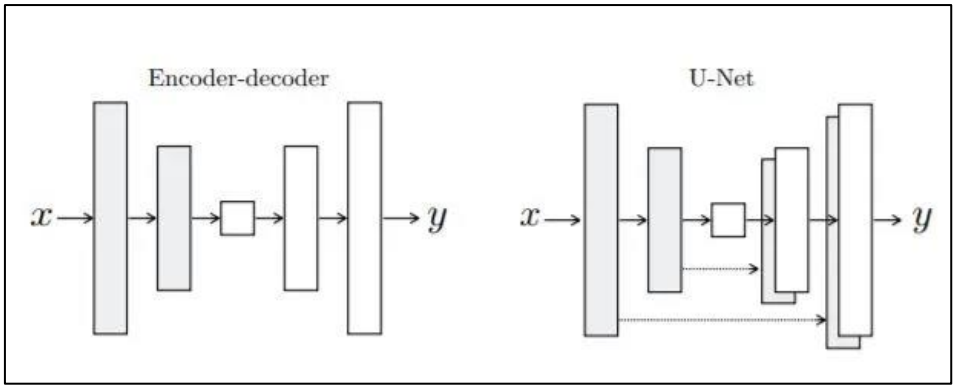

Figure 2. U-Net architecture (Isola et al., 2017)

The study conducted by Ji et al. (2018) performed salience detection also by utilizing cGANs for transforming the prediction of the saliency map as a salience

segmentation task. This is performed via the usage of a type of saliency called the pair-wise image-to-ground-truth saliency (Ji et al., 2018). The authors also used the cGANs to perform the translation from the saliency mask to a real image using saliency-to-image translation. The results from this study showed that the generator with cGANs is able to perform extremely well in terms of saliency segmentation and saliency-to-image translation.

In the same year, a study was conducted by Mao et al. (2018) that used a new method for image-to-image translation with Semantic Invariant GANs which focused on controlling the hierarchical semantics of images. The authors added constraints to both the label and the spatial levels in order to keep the semantic data of the input image in the corresponding generated image. They also created a custom constraint called the attention loss which is able to specify the most important parts of the image for classification which helped them to retain the semantic data as well. The results of the study showed that the generated images retained the semantic data of the original input image while also fulfilling the requirements of the target image.

Shortly thereafter, Gan et al. (2018) published a study that focused on fixing the issue in which the mapping function of image-to-image translation that maps from one domain to another does not have comprehensive and detailed data about the generated images. In order to fix this problem, the authors proposed a novel unpaired generative adversarial networks model that has the ability to combine the domain to be learnt and the augmented auxiliary domain called AdGAN. This is done by designing a multitude of generators and discriminators that will undergo a multitude of adversarial losses alongside full cycle constraint losses in order to reach and attain unpaired cross domain learning (Gan et al. (2018). Figure 3 shows the model with 6 generators (G1, G2, G3, F1, F2, F3) and 3 different domains (X, Y, Z) (Gan et al., 2018). The results of the study showed that their model has achieved better performance in comparison to other methods that are focused on unpaired cross domain learning.

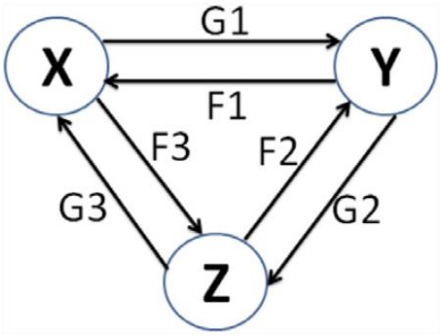

Figure 3. AdGAN model with six generators (Gan et al., 2018)

In the next year, the study conducted by Cho et al. (2019) performed image-to-image translation using GANs for image restoration and dehazing purposes. In order to understand the unique features of underwater haze images, the authors used unpaired image-to-image translation. The results of the network created by the study were great even with images that are severely distorted. For the loss function, they have used multiple cyclic consistency loses with L1 loss for capturing the unique characteristics of the image. Figure 4 shows the loss function they have used alongside the network flow (Cho et al., 2019).

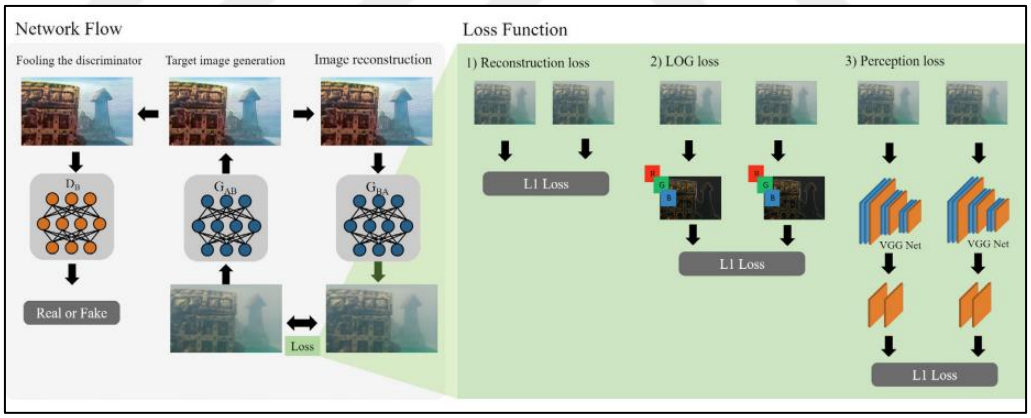

Figure 4. Multiple cyclic consistency loss with L1 loss (Cho et al., 2019)

The study conducted by Mo et al. (2019) created a new implementation of GANs called the InstaGAN in order to combat the issues of unsupervised image-to-image translation in tasks that are considered complex such as translations that involve having the image go through many considerate changes and when there are multiple target instances for the image. According to the authors, by using InstaGAN, not only will the image be translated, but also a set of instance attributes in which each of the

them will have its permutation invariant maintained in order to significantly improve multi-instance image translation. Figure 5 shows the architecture of the InstaGAN (Mo et al., 2019). The authors also used a context preserving loss in order to maintain the context throughout the multiple target instances. Since they are going to use multiple instances, they used a sequential mini-batch training technique. The results of their study showed that their method was very effective in multiple datasets that use multiple instances of target images.

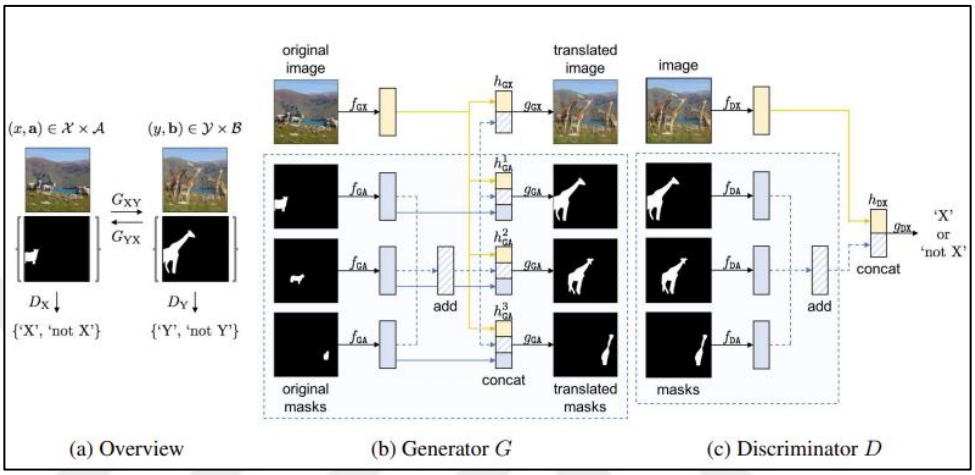

Figure 5. InstaGAN (instance-aware GAN) architecture (Mo et al., 2019)

The study conducted by Yang et al. (2019) focused on fixing the mode-collapse issue that comes with using cGANs. According to the authors, this issue occurs due to the fact that cGANs mostly learn a distribution that is very simple in which it doesn't give any wright to latent code variations and always maps an input to a single output thus not making use of the multi-modal distributions that cGANs should be capable of. Yang et al. (2019) addressed this issue by regularizing the generator with a simple realization algorithm in which the generator must generate outputs with a high degree of diversity that also depend on and take account of the latent codes. By using these diversity-sensitive cGANs, the authors were able to create a good balance in which the generated output images can have a high degree of diversity while also maintaining good visual quality. The study also tested their method in image-to-image translation and found the results to be promising with a diverse set of generated outputs.

Subsequently, the study conducted by Liu et al. (2019) focused on using image-to-image translation with only a small number of examples rather than a large dataset of images. The authors developed a few-shot unsupervised implementation of image-to-image translation in which it uses target classes that were never seen by the model only in test time with only a handful of example images. The goal of the study was to imitate the way humans are able to adapt and understand the true essence of an object by only seeing a handful of examples and creating a generalization to be used for the rest. Their method proved to be very effective in utilizing image-to-image translation with a few examples in the case a large dataset is not available.

In the same year, Xu et al. (2019) conducted a study that aims to find a solution to the high predictability in the generated images of image-to-image translation. The authors proposed a model called the InjectionGAN which is a novel GAN model that has the ability to learn a mapping that is many-to-many rather than the usual one-to-one mappings. The model uses a number of variables that can either be domain-specific variables that focus on the target domain or unspecific random variations that focus on making the generated image less predictable (Xu et al., 2019). According to the authors, all of these variables will be combined with the input image to ensure low predictability. Moreover, the authors continue to state that both of these parts will be regrouped using a unified framework which will then be used for producing generations with a high degree of diversity in each domain. The study states that the results showed that the performance of the InjectionGAN model is better than other approaches in the same domain. Figure 6 shows how the generated images by the InjectionGAN model has more variations and diversity than other approaches.

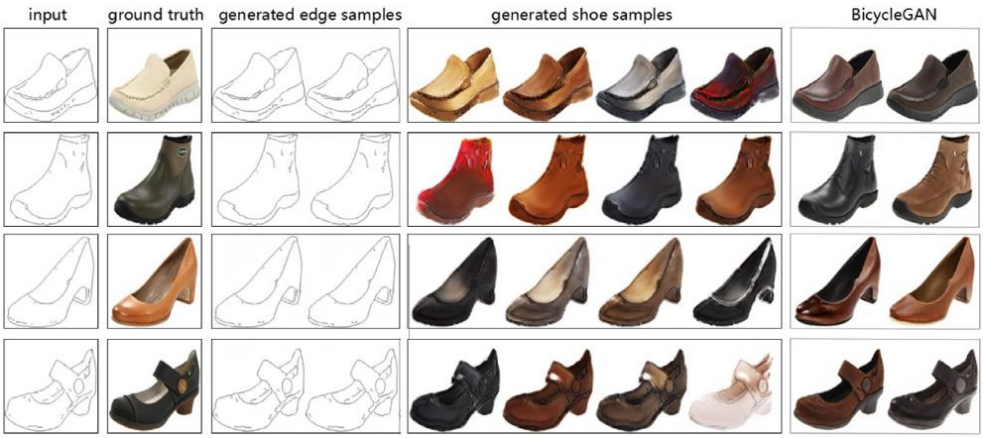

Figure 6. InjectionGAN diversity comparison with BicycleGAN (Xu et al., 2019)

Ye et al. (2019) conducted a study that provided an implementation of a triple translation GAN for their face image synthesis and translation project. According to the authors, this triple translation GAN was inspired by the problems that were facing the CycleGAN architecture. In order to enhance the generated images, the authors used L1-norm representation constraint which helped to preserve the important details in the generated images thus decreasing the error of the reconstruction. Furthermore, the study proposed a triple translation consistency loss which was extremely helpful in making the model more optimized in a much more stable way than before. The results of the study were remarkable and, in many ways, better than the results of the methods used by other studies.

In the next year, the study conducted by Choi et al. (2020) proposed a framework called StarGAN v2 that satisfies two very important criteria for a modern image-to-image translation model which are having generated output images with a great degree of diversity and having the model be scalable over different domains for the target. According to the authors, there existed previously separate methods that tackle these points why is why StarGAN v2 was needed since it fits these criteria as a single framework that can be used. The study experimented with multiple datasets such as celebrity faces and animals in which the results validated the superiority of StarGAN v2 in which the generated images were diverse, scalable, and of very good visual quality. Figure 7 shows an overview of the structure of the StarGAN v2 in terms of its generator, mapping network, encoder, and discriminator (Choi et al., 2020).

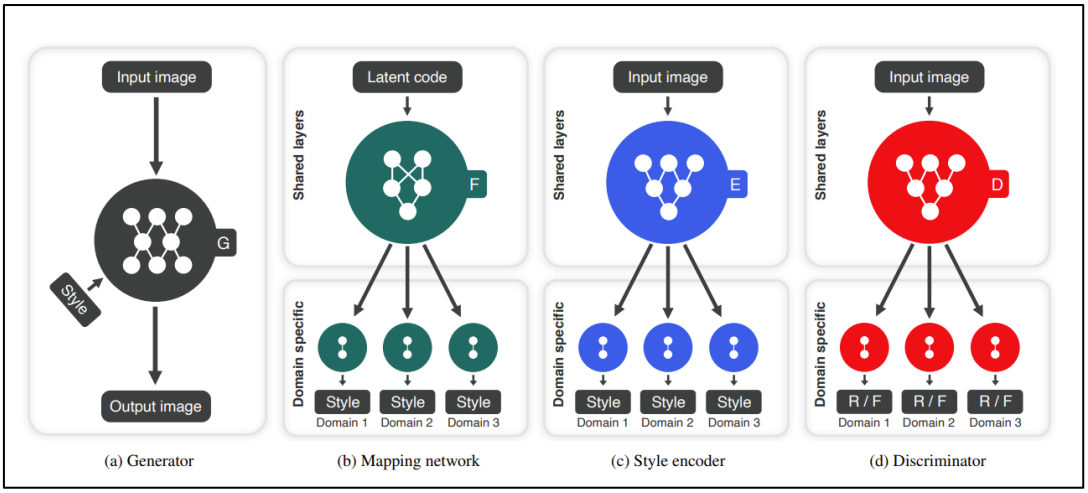

Figure 7. Overview of StarGAN v2 structure (Choi et al., 2020)

Shortly thereafter, the study conducted by Armanious et al. (2020) used image-to-image translation in medical image analysis and created a framework called MedGAN that can be used for medical related tasks in image-to-image translation. This was done by the authors due to the fact that the other methods are mostly used for specific tasks rather than being a general end-to-end framework for medical image-to-image translation. MedGAN was built via combining the adversarial framework of GANs with a set of non-adversarial losses (Armanious et al., 2020). The study also used style-transfer loss for translating the texture and details of the target image to the generated image. For the generator, the authors used a custom architecture called the CasNet which uses encoder-decoder pairs for applying progressive refinement to generate sharp medial generated images. While testing their implementation, the authors applied the MedGAN in PET-CT translation, MR motion artifact correction, and denoising PET images. The MedGAN results were exceptionally well compared to other approaches for medical image-to-image translation according to the perceptual analysis that was done by radiologists in the study. Figure 8 shows the MedGAN framework and how it incorporates the CasNet generator as its generator architecture (Armanious et al., 2020).

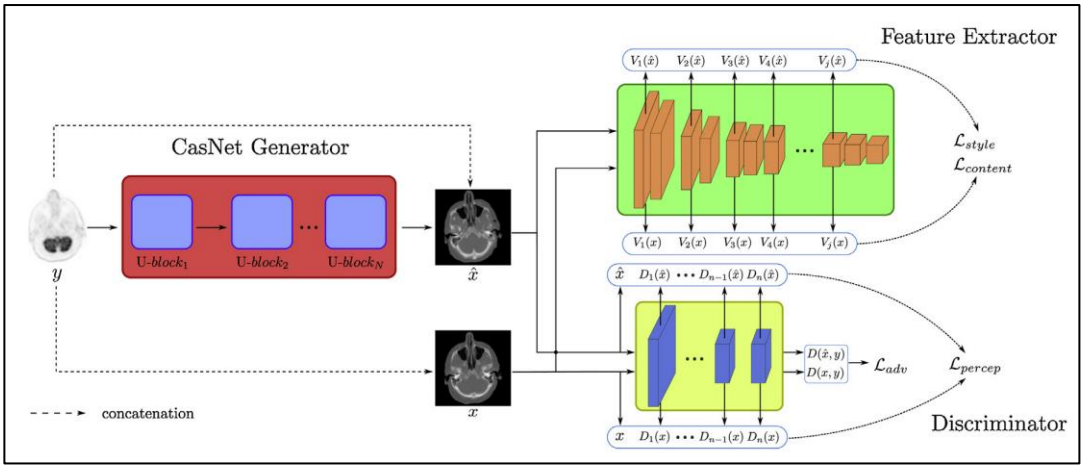

Figure 8. Overview of MedGAN framework with CasNet (Armanious et al., 2020)

The study conducted by Hicsonmez et al. (2020) looked at a different type of image-to-image translation in which images are translated to illustrations in which both the style and the content are transferred from the input image to the generated image. Due to this, 2 images will be given as inputs, one that represents the content and one that represents the style in which the generated image will have the contents of the content image illustrated the style of the style image (Hicsonmez et al., 2020). The authors created a new generator network called GANILLA that will transfer the style and the content to the output image in a very balanced way compared to any other available network. GANILLA consists of a downsampling stage which is a modified ResNet-18 network and an upsampling stage in which skip connections are used similar to the U-Net architecture (Hicsonmez et al., 2020). For the evaluation metrics, the study has created a new framework that considers both the style and content with different classifiers in their image-to-illustration model.

Zhu et al. (2020) looked into a way of implementing image-to-image translation without the need for paired training data which is the regular for other image-to-image translation implementations. This study presented a method for translating an image from the source to the target without paired examples called CycleGAN. This was done via mapping the source image to the domain in which the distributions are indistinguishable from each other (Zhu et al., 2020). Moreover, the authors applied an inverse mapping and a cycle consistency loss due to how the mapping is insufficiently constrained. The results of the study proved to be superior

when compared to other methods in a multitude of image-to-image translation tasks (style transfer, photo enhancement, etc.). Figure 9 shows the mapping functions used in CycleGAN alongside the cycle consistency losses (Zhu et al., 2020).

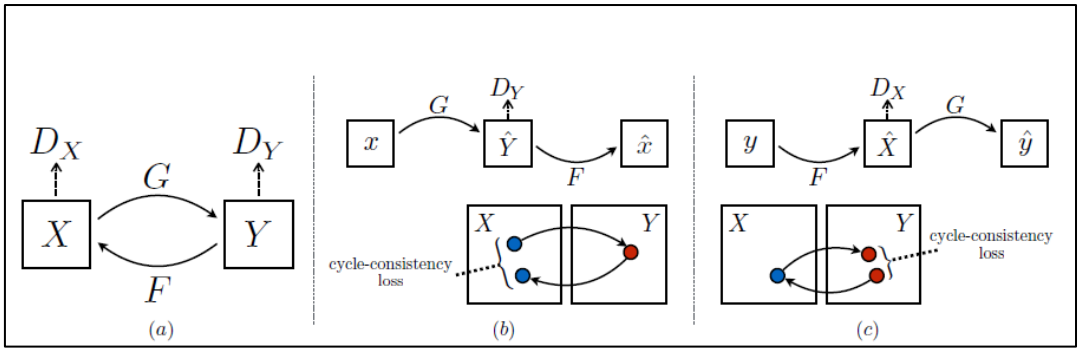

Figure 9. CycleGAN mappings and cycle consistency losses (Zhu et al., 2020)

Xia et al. (2020) conducted a study in the same year to find a solution for the 3 main issues that come with image-to-image translation: lack of paired training data, having many outputs from only 1 single image, not being able to simultaneously train with a single model for multi-domain translation. The study came up with a framework that fixed these issues in which it works with unpaired training data to generate outputs with great diversity and has the ability to simultaneously train multiple domains with a single model. According to the authors, they have also conducted experiments that resulted in them extracting domain-level signal as an explicit supervision due to issues that come with translating the content and the style of the input. The results of the study have shown that their method is superior to other methods in this field.

In the same year, the study conducted by Dhont et al. (2020) looked into how to render digitally reconstructed radiographs (DRR) via the usage of image-to-image translation. The study proposed a framework called RealDRR which is going to be used for DRR rendering in a very fast, robust, and realistic manner via the usage of raytracing and image-to-image translation with deep learning. The main goal of the study was to create a framework that can be used to render DRR and can be trained on site with resources that can already be found in radiotherapy departments all over the world while also maintaining a good degree of flexibility in which it can be modified

according to the local imaging systems. Figure 10 shows the overview RealDRR framework proposed by the study (Dhont et al., 2020).

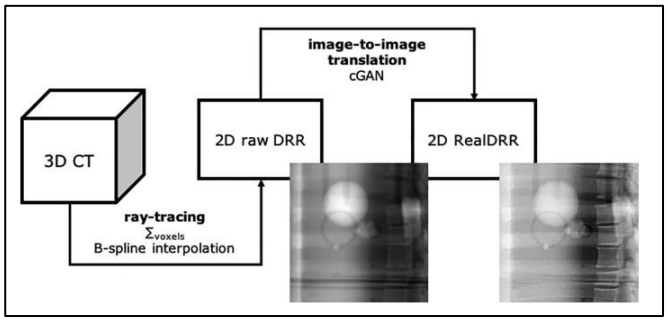

Figure 10. RealDRR framework for DRR rendering (Dhont et al., 2020)

The study conducted by Emami et al. (2020), showcased an attention mechanism that was implemented using GANs in a new model architecture called the SPA-GAN that can be used for image-to-image translation. In the SPA-GAN, the attention is calculated in the discriminator which is then used for assisting the generator to provide more focus and attention to the regions that are the most discriminative between the images (Emami et al., 2020). According to the authors of the study, SPA-GAN also uses a custom feature map loss which is used for preserving the core characteristics and features of the input image during the translation process so they can be passed on to the generated image within the target domain. The study also focused on finding a way to utilize image-to-image translation without the need for paired training datasets due to how expensive and difficult it can be to get them. The results of the study proved that it is superior to its counterparts while also being a lightweight model that doesn't need to be supervised. Table 1 shows the classification performance of the SPA-GAN in comparison to other models and methods (Emami et al., 2020).

Table 1

Classification Performance of the SPA-GAN (Emami et al., 2020)

Method	apple → orange	orange → apple	zebra → horse	horse → zebra	lion → tiger	tiger → lion
Real	97.58	97.36	85.71	97.85	99.63	100
DualGAN [6]	78.57	64.91	41.42	83.33	66.53	39.05
UNIT [13]	80.07	94.75	70.00	82.50	82.95	67.27
MUNIT [14]	67.80	85.70	55.27	82.50	79.60	52.75
DRIT [15]	75.50	76.80	72.50	80.31	84.90	60.38
CycleGAN [5]	71.80	72.93	75.00	83.33	73.48	48.10
Attention-GAN [8]	27.40	35.71	62.86	80.71	78.90	52.00
AGGAN [7]	21.80	34.21	64.28	82.85	87.63	50.54
SPA-GAN	**87.21**	**95.49**	**84.17**	**87.50**	**92.42**	**87.12**

In the next year, a study was conducted by Wang et al. (2021) that aimed to research the issues that could arise in image-to-image translation due to bias and diversity. Bias in image-to-image translation could come from a number of things such as biased datasets that only show the object in a certain way which could hinder the translation process as a consequence (Wang et al., 2021). The authors proposed the usage of semantic constraints that will make sure the specified image properties are preserved. The authors further state that the model they proposed will lessen the surprise changes that could during the translation process while also keeping the changes that are required for the translation process to be complete. The study stated that it has experimented on the model and the results showed that it is very effective in fixing the specified issues.

The study conducted by Marzullo et al. (2021) proposed an image synthesis method called the Minimally Invasive Surgery (MIS) image synthesis in which they trained the Pix2Pix cGAN, created by Isola et al. (2017), so it can generate paired MIS data with the use of segments from surgical instruments and tissues. By applying this, the study by Marzullo et al. (2021) was able to generate laparoscopic images via the usage of image-to-image translation which translates the input sematic label map. The authors also stated that they enhanced the look of the surgical tools and made them more realistic by adding an extra regularization term to the original optimization. The results of the study proved that their method is able to generate realistic MIS images by translating MIS segmentations. Since these generated images can be used as an addition to existing datasets, this method can help in solving the large-scale data shortage for DL in the surgery field (Marzullo et al., 2021).

Shao et al. (2021) conducted a study that focused on creating a unified framework for diverse and multi-domain image-to-image translation that does not require paired datasets for training, is able to translate through multiple domains, and can generate outputs with a great degree of variety and diversity. The framework presented by the study is able to translate through multiple domains while also maintaining the core characteristics of the input image which differentiates it from other known frameworks in the field of image-to-image translation. Moreover, by randomly sampling from the normal distribution for a latent noise, they were able to obtain and generate diverse outputs that greatly vary from each other (Shao et al., 2021). However, the authors needed to modify the discriminator by adding to it a noise separation module due to the mode collapse issue that occurs due to the unconstrained noise. The study compared their results with other similar frameworks and found that their DMDIT framework offered good performance when compared to state-of-the-art methods in the same field. Table 2 shows the network structure of their framework (Shao et al., 2021).

Table 2

Network Structure of DMDIT Framework (Sho et al., 2021)

Encoder(G_{Enc})	Decoder(G_{Dec})	Attention	D	C	S
Conv(64,4,2),BN,LReLU	DeConv(1024,4,2),BN,ReLU		Conv(64,4,2),IN,ReLU		
Conv(128,4,2),BN,LReLU	DeConv(512,4,2),BN,ReLU		Conv(128,4,2),IN,ReLU		
Conv(256,4,2),BN,LReLU	DeConv(256,4,2),BN,ReLU		Conv(256,4,2),IN,ReLU		
Conv(512,4,2),BN,LReLU	DeConv(128,4,2),BN,ReLU		Conv(512,4,2),IN,ReLU		
Conv(1024,4,2),BN,LReLU	DeConv(64,4,2),BN,ReLU		Conv(1024,4,2),IN,ReLU		
	DeConv(32,4,2), BN,ReLU		FC(1024),IN,LReLU		
	Conv(3,7,1), Tanh	Conv(1,7,1), σ	FC(1)	FC(13), σ	FC(8)

Lin et al. (2021) conducted a study that provided a framework called the ZstGAN which is used for training image-to-image translation models that can perform translations from domains without the need for these domains to be observed by the model during the training. Their framework is based on unsupervised zero-shot learning and how it can be used with image-to-image translation. Furthermore, the study used a feature distribution that is specific for the target domain in which that feature distribution follows the same modalities. The results of the study showed how

effective the ZstGAN framework compared to other zero-shot learning methods, especially with its accuracy.

In the same year, Yan et al. (2021) conducted a study that aimed to fix the issue in which failure occurs when translating images from one domain to another due to the image containing noise. The authors proposed an enhanced generative adversarial network which combines image-to-image translation with image denoising techniques in order to fix this issue. Figure 11 shows how the input and output of their method looks like when it combines translation and denoising techniques (Yan et al., 2021). The authors further stated that their model is built upon the Pix2Pix project by Isola et al. (2017) in which they add residual blocks in addition to the original model. This will assist the model in capturing information in a much deeper manner (Yan et al., 2021). Furthermore, in order to enhance the performance of the translation process, the authors proposed a perceptual loss function that will be used during the training process of their model. The study has performed several experiments in order to test their model and the results showed that their model is able to combat the effects that noise has on the images during the translation process. The authors also observed that their model showed better performance than other methods in the same domain.

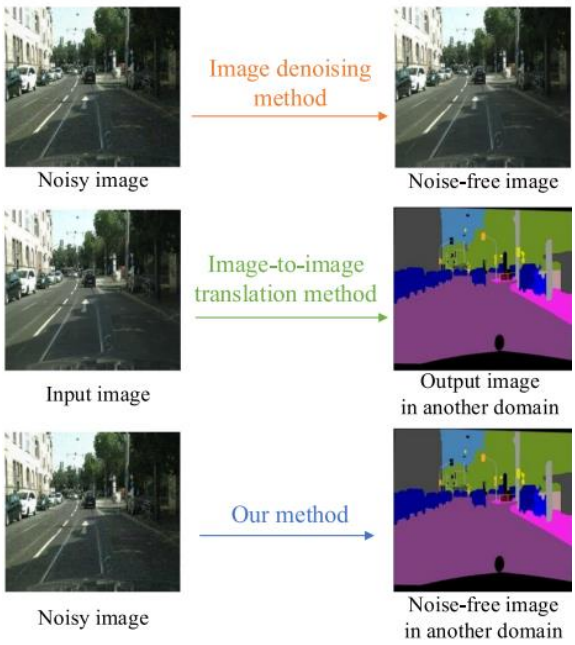

Figure 11. Combination of image denoising and translation (Yan et al., 2021)

Nie et al. (2021) conducted a study that proposed a new DL method for image-to-image translation called the URCA-GAN in which it aims to generate images that are of very high quality and diversity in comparison to other studies that proposed similar methods. Their method is made up of Upsample Residual Channel-wise Attention Blocks which are proposed by the authors for extracting the foreground features from the image. According to the authors, these blocks are modeled after ResNet and SoftMax channel-wise attention which will assist them greatly in extracting the required foreground features. After that, the authors stated that the features extracted from the foreground by the blocks need to be merged which is done by an architecture proposed by the authors called the Upsample Residual Channel-wise Attention Module. Figure 12 shows the structure of the URCA-GAN method (Nie et al., 2021). The study stated that the results of their model showed that their model performed better than other methods in the same domain.

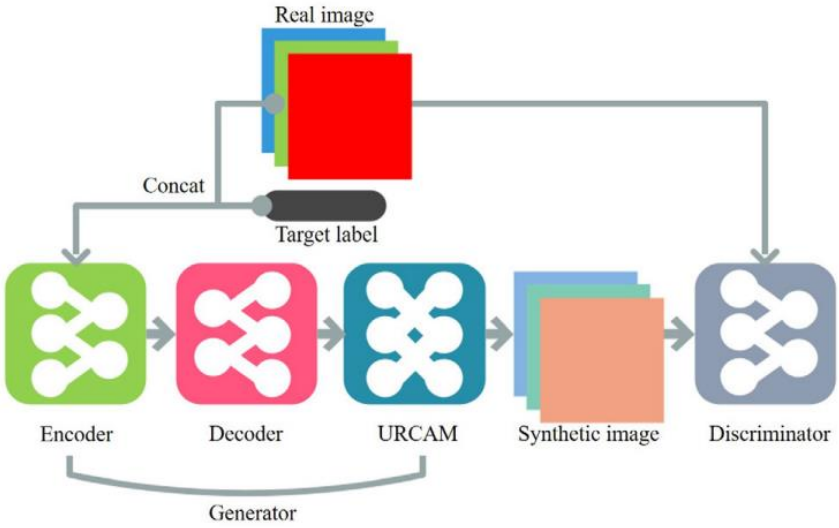

Figure 12. Structure of URCA-GAN for image translation (Nie et al., 2021)

Liu et al. (2021) conducted a study in the same year that focused on achieving multi-domain image-to-image translation. Rather than using multiple generators or labeled datasets, the authors proposed an unsupervised network that has only one generator and one discriminator with an arbitrary number of encoders in order perform multi-domain image-to-image translation. Furthermore, the network of the authors does not require the use of labeled datasets in order to learn the mappings of each

domain. This is crucial because using multiple generators or labeled datasets can be costly and will introduce limitations that could affect the future work in this domain (Liu et al., 2021). The authors further stated that they proposed a loss for extracting the representation details for each domain called the representation loss which could help in improving the performance of the multi-domain translation process. The results of the study showed that their method has better performance than other existing methods in the same domain. Figure 13 shows the results of their model compared to other models (Liu et al., 2021).

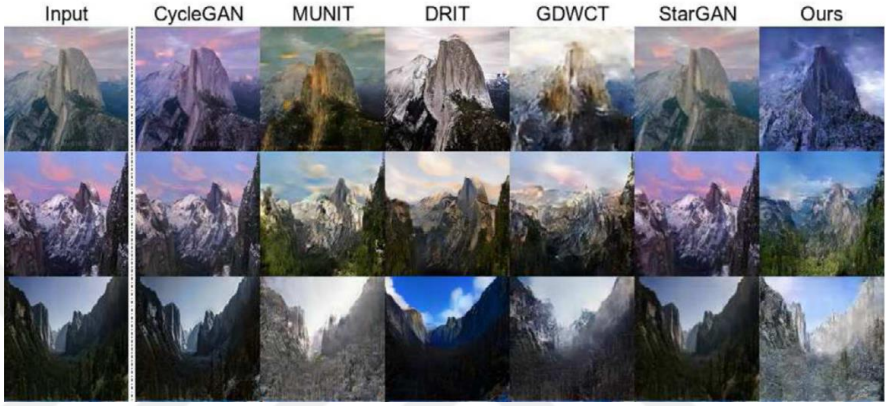

Figure 13. Comparison between Liu et al. and other models (Liu et al., 2021)

Brazilay et al. (2021) conducted a study that focused on image-to-illustration translation with a number of different illustrators using only one trained model. To that end, the authors proposed a network framework called the MISS GAN which is able to generate images that contain both the content and the style required in the translation process. While other methods require the usage of a number of generators depending on the different styles, MISS GAN is able to work with many different styles using only one trained model (Brazilay et al., 2021). Figure 14 shows the architecture of the generator in the MISS GAN framework (Brazilay et al., 2021).

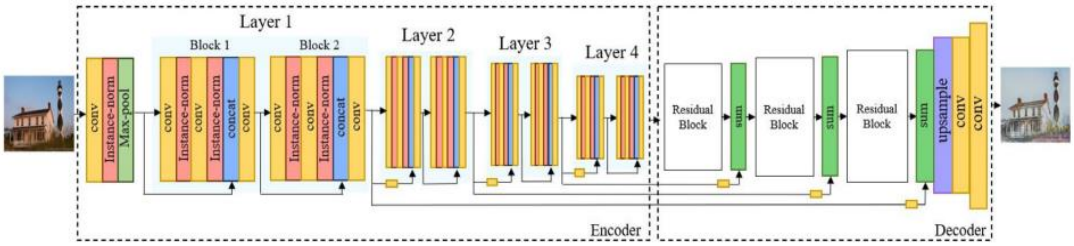

Figure 14. Generator architecture in MISS GAN (Brazilay et al., 2021)

In the next year, the study conducted by Yang et al. (2022) looked at creating a framework by combining image compression with image-to-image translation in multiple domains. The study proposed an implementation of image-to-image translation that works using an image-to-image codec that is made up by quantization and entropy coding. However, they modified the framework due to the need of image compression which required a regular image codec that will be deployed simultaneously with the image-to-image codec (Yang et al., 2022). Due to this, the authors further proposed a unified framework which is now able to translate and perform autoencoding using a single codec with a lot of flexibility due to using adaptive residual blocks. The evaluations of the study proved that their framework is able to perform both image compression and image-to-image translation with impressive results.

The study conducted by Platscher et al. (2022) focused on generating magnetic resonance images using image-to-image translation. These images will be translated from semantic segmentation maps and the brain volumes will either be with stroke lesions or without them. The objective of this study is to generate images that can be used as training datasets in the future for medical deep learning tasks. The authors also stated that they trained a GAN for generating synthetic lesion masks which will then be combined with the generated magnetic resonance images for creating a large dataset of synthetic stroke images. The study evaluated the performance of their generated dataset via the usage of the U-Net with a clinical test set which provided good results compared to just training the model with original data.

In the next year, Varghese & Hoskere (2023) conducted a study that looked at using image-to-image translation for generating structural damage using a dataset

made-up of buildings. The authors created a cGAN architecture called EIGAN which is capable of creating damaged buildings when an input of an undamaged building is given. By using image-to-image translation, they are able to generate structural damage images using images of normal and undamaged buildings (Varghese & Hoskere, 2023). The authors also stated that EIGAN can also do the reverse in which given a damaged building, EIGAN can generate an image that shows the building without any structural damage while also maintaining the characteristics of the building. The study tested their architecture with a dataset that contains images of buildings damaged by an earthquake alongside regular undamaged buildings. The results of the study showed the superiority of their architecture compared to other known architectures in the field, especially in translating the damage of buildings.

The study conducted by Chen et al. (2023) focused on detection of bladder cancer and used image-to-image translation for synthesizing images. The authors are using multi-parametric magnetic resonance imaging for their identification process, but it suffers from noise and artifacts. Due to this reason, the authors suggested the usage of image-to-image translation using existing sequences for substituting any of them that suffers from noise. In order for the generator to synthesize multi-domain images, it will be supported by an improved adaptive instance normalization module which was introduced by the authors in the paper. They also used region-wise semantic segmentation for enhancing the quality of the generated images in a specific region that will be important for the identification process of bladder cancer (Chen et al., 2023).

Grebo et al. (2023) conducted a study that focused on the translating thermography into data for digital image correlation. This translation will be using image-to-image translation techniques with the Pix2Pix and CycleGAN models. The authors further stated that these models were on a dataset that contained two types of images, paired and unpaired (Grebo et al., 2023). The study stated that the results showed that the Pix2Pix model performed better than the CycleGAN model.

The study conducted by Al-Hindawi et al. (2023) used image-to-image translation for generalizing critical heat flux detection models. This was done because

when the critical heat flux is detected using other machine learning models, it gave poor performance when data from different domains was used (Al-Hindawi et al, 2023). In order to combat this, the authors uses image-to-image translation for transforming the images to make them as if they were generated from the same domain that was used for training the model previously. The study further stated that the results of this new approach showed high accuracy in generalizing the models.

In the same year, a study was conducted by Kang et al. (2023) which focused on domain adaptation and preserving the structure of the image when performing image-to-image translation. Domain adaptation is done via using the results of the translation process as segmented samples which could then be used for domain adaptation segmentation training (Kang et al., 2023). The results showed that the method of the study gave better performance than other methods in the same field.

Chapter 3

Methodology

Our implementation of image-to-image translation utilizes cGANs with two neural networks serving distinct roles as the generator and the discriminator. For the generator, we have mixed the U-Net architecture, known for its excellence in detail preservation, with a basic foundational Residual Neural Network (ResNet) architecture, which is also known for its ability to learn complex mappings, in order to leverage both of their complementary advantages to allow the generator to generate high-quality images. On the other hand, for the discriminator, we have applied the PatchGAN architecture in order to focus on every patch of the image rather than only the image's structure. This will allow the discriminator to capture all of the small and intricate details of the image which should assist it in detecting the realism of the image. With this structure in mind, the two neural networks will push each other to improve the generated image quality of the model thus making this image-to-image translation framework extremely robust.

In order to write this implementation, tremendous research was required which involved a certain research strategy to analyze articles from many different academic databases. The research strategy included certain keywords that needed to be included in the selected articles such as "Image-to-Image Translation", "Generative Adversarial Networks (GANs)", "Deep Learning", and other related terms. The inclusion and exclusion criteria focused on many aspects such as the relevancy of the article to the topic of image-to-image translation, the date of publication, and the language the article is written in. For implementation-specific articles, we made sure to select only the latest articles to ensure currency. Articles that did not fit these criteria were excluded from the study. By the end of this research strategy, we were able to identify many articles that we could use in our study in both theoretical and practical manners.

3.1 Data Collection

Image-to-image translation is highly dependent on the image datasets available in both its training and testing processes. Data collection is very important for modern DL applications due to them requiring large amounts of data with high quality compared to traditional applications (Whang et al., 2023). Due to that, data collection was one of the most crucial parts of the development process of our implementation and played a very pivotal role in our result. The images in the used datasets must be of high quality while also remaining diverse, which is why gathering and curating datasets can highly influence the effectiveness of our resultant model. While we needed to collect high-quality datasets, we also needed to make sure the datasets are free to use for our study to follow the data privacy law.

During the data collection process, we had certain criteria that needed to be met by our selected datasets that included image quality, dataset size, and having image pairs for each sample. We have looked at many other studies that implemented image-to-image translation and looked into the kind of datasets they used. This was when we found the same datasets used by Isola et al. (2017) which was also given to the public for free to use by the University of California (2018). There were 6 image-to-image translation datasets that included cityscapes, facades of buildings, edges,

maps, and more. The two datasets our study focused on the most were the facades dataset and the maps datasets due to them having a small a size of only 29 MB and 230 MB respectively making them the easiest to train in terms of time and computing power. These datasets were made up of pairs of input images and ground truths. For example, in the case of the facades dataset, the input images were facades while the ground truths were actual buildings. The pairs were in one image which required them to be cut from each other during the pre-processing phase so they can be read correctly by the model. Table 3 shows the information related to the available datasets such as number of samples and size.

Table 3

Information of Datasets Used (University of California, 2018)

Information	Facades	Maps
Size	29 MB	230 MB
Train Samples	400	1096
Test Samples	206	1098

The datasets are made up of 3 different folders, train, test, and val. Each of these folders contain the pairs that can be used by image-to-image translation projects. The pairs are combined into one jpg image file with a width of 512 pixels and a height of 256 pixels. After the pair image is cut into two images to separate the input and the ground truth, each image in the pair will have an equal width and height of 256 pixels. All image pairs in the datasets are colored and have a size of around 60 KB. Figure 15 shows some samples of the datasets used in our implementation.

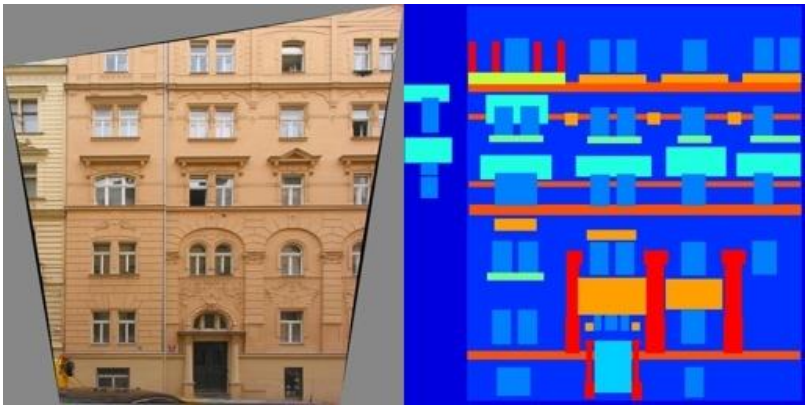

Figure 15. Samples of training dataset (University of California, 2018)

In summary, the data collection process is extremely important for the success of image-to-image translation which is why we prioritized the collection of high-quality datasets in order to ensure the reliability of our model. We also made sure that the selected datasets are free to use and have paired samples of input and target images which we can use as a condition in our cGAN model. The datasets vary in size and contain colored images with equal dimensions for input and target images.

3.2 Data Pre-processing

Data pre-processing is a very important task in any project that involves deep learning because it transforms the raw data into a format that can be used for training our model. In order to make sure that the analysis results of an application are correct and reliable, data pre-processing is a very necessary step applications that rely on large amounts of data (Fan et al., 2021). In our implementation, each pair of images in our dataset is in one image file thus the first step in our pre-processing phase was to separate each pair of images into two images, an input image, and a target image. This was done by simply dividing the width by 2 and using the result to cut the pair image matrix to get each image. We then converted these images into float objects to continue the pre-processing process. This conversion ensured the compatibility of the images with our deep learning framework so we could continue to pre-process the data.

After the images are separated, we performed data augmentation in order to increase the size of our dataset and improve the robustness of our model. Data augmentation is a very useful method for making training data be of higher quality and have higher diversity which should help in achieving good training performance (Mumuni & Mumuni, 2022). One of our data augmentation techniques is resizing the image to larger dimensions and then randomly cropping it back to its original size which will select random regions of the resized images. This method is called random cropping in which a small section of the image is taken randomly and resized to be of the original image's size which can then be used as training data (Maharana et al., 2022). Moreover, we also randomly flip the image horizontally from left to right thus creating a random mirroring effect with a probability of 50% to further diversity the training data. The goal of data augmentation is to create a much more diverse dataset without altering the semantic content of the images and by still maintaining the intricate details of the original image. Figure 16 shows four possible results that can be obtained from the used pre-processing methods on the input images from our dataset.

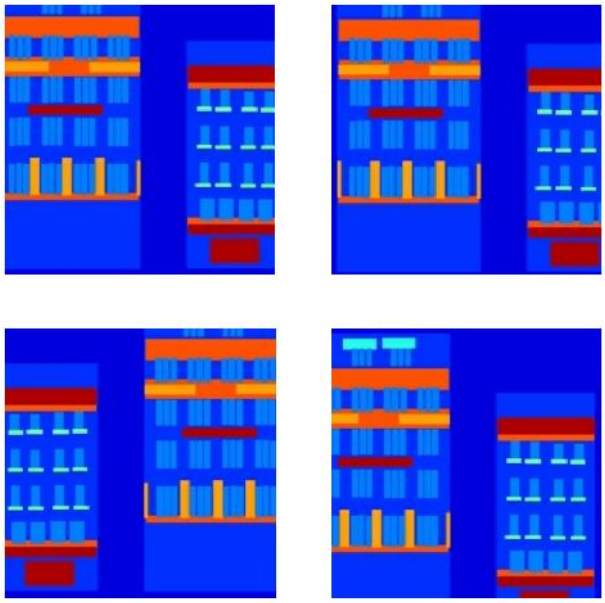

Figure 16. Showcasing dataset after preprocessing

Overall, our pre-processing method involved image pair separation using division, data conversion to ensure compatibility, data augmentation techniques to

ensure data diversity, and quality control measures to ensure the reliability of our dataset. All of these steps contribute to the preparation of a robust and diverse dataset for training our image-to-image translation models.

3.3 Model Architecture

For image related tasks in CV, we usually look into using GANs which consist of two interconnected neural networks: the generator and the discriminator. The goal of the generator is to create data samples that are as realistic as possible while the goal of the discriminator is to identify the generated data samples and distinguish between them and real data. In our case, the generated data samples will be images which will be used for image-to-image translation tasks.

However, normal GANs are not best suited for image-to-image translation tasks due to them not specifying any conditional information on the generated data samples in order to generate them in a very specific way unlike cGANs. For this reason, we have decided to use cGANs as our model architecture for our image-to-image translation tasks due to its ability to learn mappings between different domains via the usage of paired data samples which are included in our collected dataset. This is very crucial for our implementation due to it being designed to be flexible in which it can work with multiple different datasets (facades, maps, etc.) and each of these datasets is considered to be in a different image domain. In this case, the target images in the paired data will be considered the condition to be used in our cGAN model. The following two subsections will explain the generator network architecture and the discriminator network architecture in detail.

3.3.1 Generator network architecture.

The goal of the generator network in our cGAN model is to transform the input images from one image domain to another which is provided by the target image as a condition. In our implementation, this task is done via the usage of a combination of two well-known architectures: the U-Net architecture and the ResNet.

The U-Net architecture is known to have three important parts to it: downsampling, upsampling, and skip connections. Downsampling blocks are used for reducing the spatial dimensions of the input images while also making sure to extract the intricate details of the image so they can be passed on later. Upsampling blocks are used for increasing the spatial dimensions of the feature maps to around the size of the desired output image. Skip connections are used to preserve the details from the downsampling blocks to the upsampling blocks in which each downsampling block is connected with its corresponding upsampling block (Wu et al., 2020).

In the middle of the U-Net architecture, we have included basic ResNet blocks right after the downsampling stage which are used to help in capturing the details of the image and in learning complex image transformations. This combination serves to increase the expressiveness of our model in which the U-Net will focus on preserving the spatial information while the ResNet will focus on enhancing feature learning and thus result in the model being able to learn much more complex mappings.

In our implementation of the generator, all of the layers are initialized via the usage of a random initializer with a mean of 0 and standard deviation of 0.02 in order to stabilize the training process. Furthermore, the input layer receives images with a size of 256x256 and 3 color channels. At the start of our generator, we have 3 downsampling blocks that are made up of convolutional layers with a stride of 2, same padding technique, and Leaky Rectified Linear Unit (ReLU) activation functions. The first block utilizes 64 filters without applying any normalization technique. The subsequent blocks use 128 and 256 filters, respectively, while also adding layer normalization to improve the stability of the training process. We have used a kernel size of 4 in all of the downsampling blocks. After these blocks, we are entering the ResNet part of the architecture to which we have added 3 ResNet blocks for capturing the details of the input in a deeper manner. Each of these blocks contain two convolutional layers with 256 filters, a kernel size of 3, and a stride of 1. They also utilize the batch normalization technique and ReLU activation functions. After that, we have added 2 upsampling blocks that are made up of transposed convolutional layers which are used for increasing the spatial dimensions with a stride of 2, kernel

size of 4, same padding technique, and ReLU activation functions. The blocks utilize 128 and 64 filters respectively. Each upsampling block is concatenated with its corresponding downsampling block for the implementation of skip connections. After that, we have a final transposed convolutional layer that has a kernel size of 4 which will be used for generating the output images with 3 color channels. Moreover, the final layer uses a hyperbolic tangent (Tanh) activation function which will help in generating realistic images due to scaling the pixel values to the range of -1 and 1 as part of its process. All of these steps in order make up the generator architecture of our model: downsampling blocks, ResNet blocks, upsampling blocks, and a final layer. Figure 17 shows our generator architecture in detail with all the different steps and how they connect to each other.

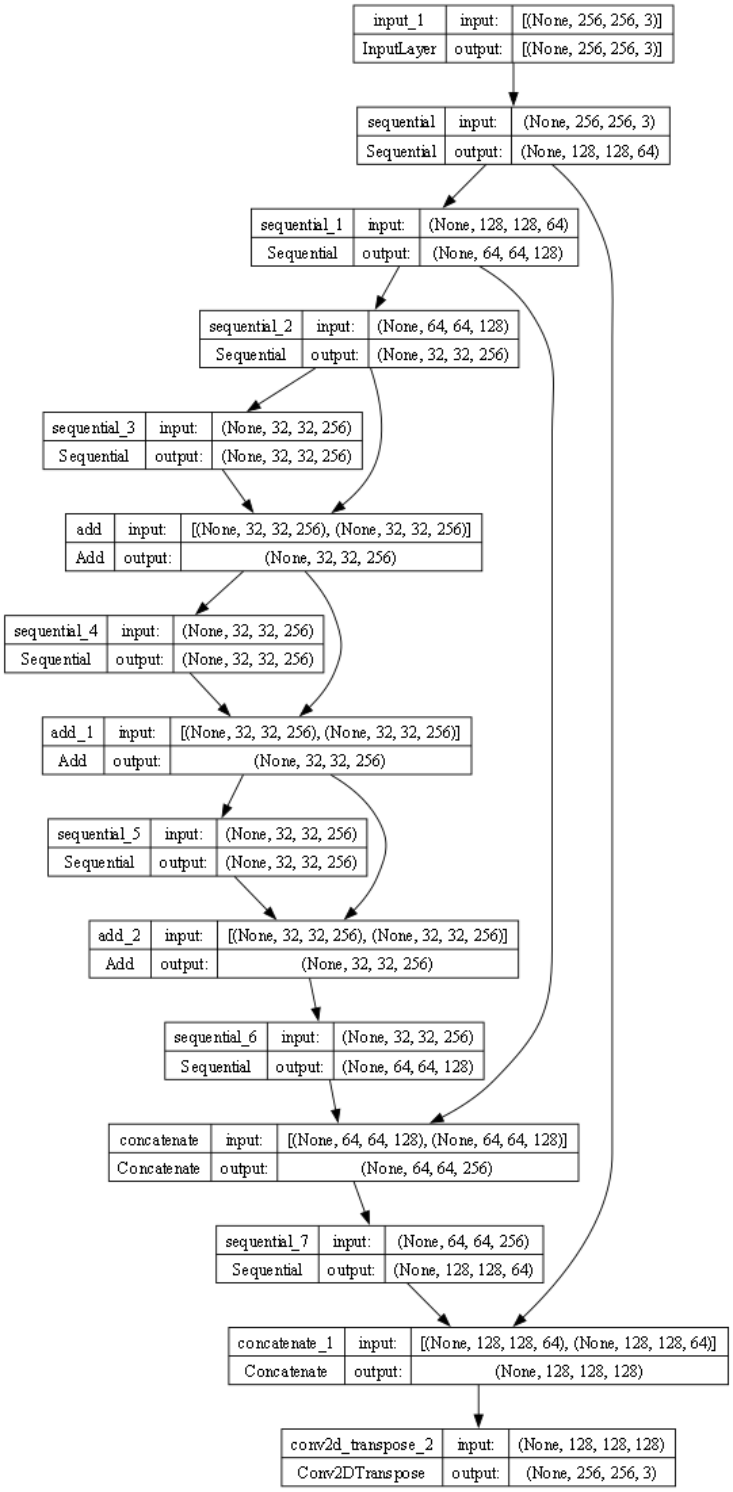

Figure 17. Architecture of our generator with U-Net and ResNet

We have used two loss functions for calculating the generator loss, the adversarial loss and the L1 loss. The adversarial loss, also known as the GAN loss, is

used for calculating and minimizing the difference between the predictions of the discriminator in terms of whether the generated image from the generator is real or not. This will help the generator in generating higher quality images that will be harder to distinguish from real images from the perspective of the discriminator as the generator will aim to minimize the difference of the discriminator's output between the generated images and real images. On the other hand, we have used the L1 loss, also known as the mean absolute error, to measure the difference between the generated image and the target image in terms of pixel values. The main difference between the adversarial loss and the L1 loss is that the L1 loss focuses on the image fidelity rather than the realism of the image. By doing this, the generator will be able to preserve the intricate details of the target image in its generated images. Our generator benefits a lot from the combination of these two loss functions since it will be able to generate images that are realistic while also staying faithful to the target images with detail preservation.

3.3.2 Discriminator network architecture.

The goal of the discriminator network in our cGAN model is to distinguish between real images and the images generated by the generator. In our implementation, this task is done via the usage of the PatchGAN architecture.

The PatchGAN architecture differs from other discriminator architectures in the fact that it focuses on patch-wise discrimination rather than giving a single score for the entire image at once. This approach is significantly more effective for image-to-image translation tasks because it will guide the generator in identifying the parts of the image that need improvement in much greater detail due to it discriminating patch by patch.

In our implementation of the discriminator, all the layers are initialized randomly with a mean of 0 and standard deviation of 0.002 just like the generator. Moreover, the input and target images are both of size 256x256 with 3 color channels and are concatenated to form a single input tensor that will be used by the discriminator in order for it to compare the input and target images. After this, the downsampling

process starts with 3 downsampling blocks with convolutional layers and Leaky ReLU activation functions for reducing the spatial dimensions of the input in order to learn the features for the discrimination process. The first downsampling blocks does not have any normalization technique and has 64 filters. The next 2 blocks have 128 and 256 filters respectively and also utilize layer normalization.

After downsampling is done, zero padding layer is applied which is going to preserve the spatial information and prevent edge artifacts by adding zeros around the borders of the feature maps. After this, a convolutional layer is added with 512 filters and a stride of 1 which is responsible for computing the features from the padded feature maps. After that, layer normalization is applied, and a Leaky ReLU activation function is added followed by another zero padding layer. Finally, a convolutional layer is added at the end which has 1 filter and a kernel size of 4 and is responsible for providing the output for each patch during the discriminator process. Figure 18 shows our discriminator architecture in detail with all the different steps and how they connect to each other.

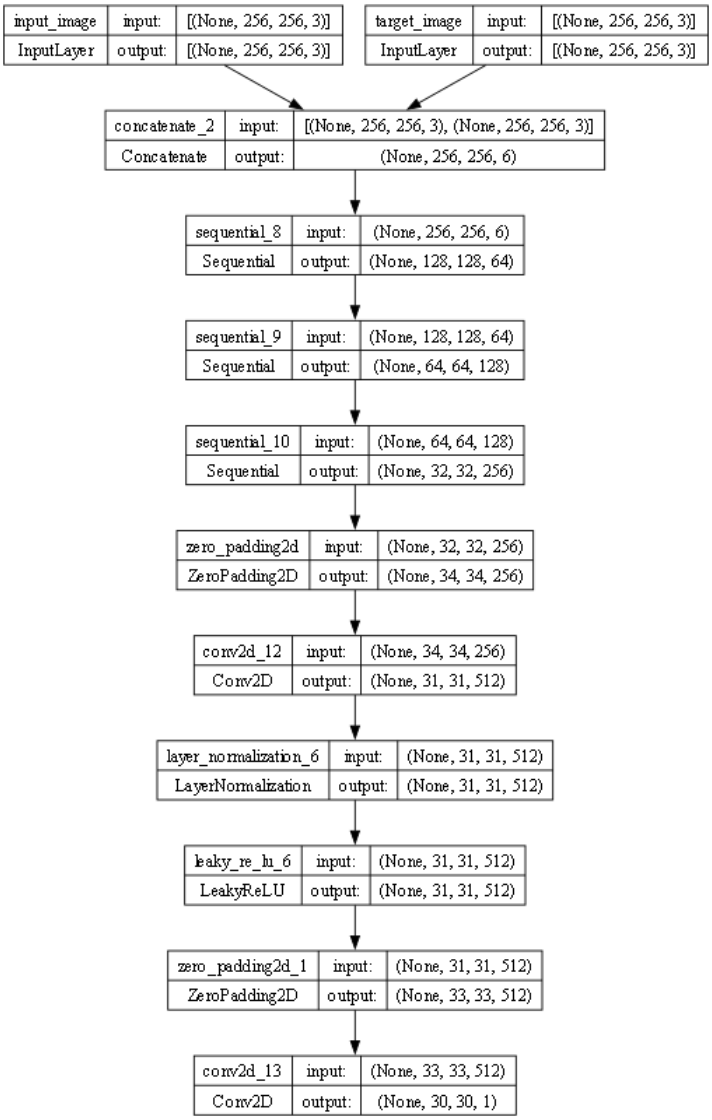

Figure 18. Architecture of our discriminator with PatchGAN

In order for the discriminator to differentiate the real images from the fake ones, we have created a discriminator loss that will be part of the training process. We have used two loss functions for calculating the discriminator loss, the real image loss and the generate image loss. The goal of the real image loss is to check the accuracy of the discriminator in classifying real images as real. It is calculating it by doing a comparison between the output of the discriminator when a real image is evaluated with a target label that shows that these images are real. In our case, the target label is a tensor of ones in which they indicate that the evaluated image is a real image and can be used for comparing with the output of the discriminator. The real image loss

aims to check the result of the comparison between the output and the target label, and to also minimize the difference between them to encourage the discriminator to correctly classify real images as real. On the other hand, the generated image loss works in the opposite direction in which it evaluates the discriminator's ability to distinguish generated images from real images, in other words, to tell if an image is fake or not. The calculation of the generated image loss is similar to the real image loss in which the discriminator's output is compared with a target label that indicates the images are fake. In our case, the target label is a tensor of zeroes that indicate that the valuated images are not real. The generated image loss aims to minimize the difference between the output of the discriminator and the target label, and to also encourage the discriminator to correctly classify generated images as fake.

3.3.3 Training process.

This section will explain the training process used by our implementation and all of the configuration values we used for setting it up. The training process trains both the generator and discriminator via the usage of their loss functions with the goal of minimizing them.

In our training loop, we have used steps instead of epochs in which the loop iterates through the entire training dataset and feeds batches of input and target images to our model. In each step, we are using gradient tapes in order to record the operations to differentiate automatically during the process of backpropagation. After defining these, we provide the generator with an input image and in return it generates an output image depending on that input. Next, the discriminator takes the output image from the generator and evaluates it alongside the target image to compute the loss functions. Both the generator loss and the discriminator loss are calculated after each step is completed and the data used for calculating them depends on the current batch of data used in the loop. We have used the Adam optimizer for both the generator and the discriminator in which the learning rate value is 0.0002 and the beta value is 0.5 for both networks.

During each iteration, the optimizers will update the parameters of the model depending on the value of the loss functions for the generator and the discriminator using methods such as gradient descent for minimizing the loss in each iteration. After that, we calculate the metrics for evaluating the accuracy of our generated images. In this case, we have used the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR) for evaluating our generated images. This calculation will occur after each step is completed. After each 1000 steps, we are showing an example of the generated images and printing the generator loss, the discriminator loss, and the values of the metrics. After each 5000 steps, we are taking a checkpoint to capture the state of the model at these steps so they can be used later for evaluation, testing, or even resuming the training process at a later date. In our case, we have used 50000 steps for training our model and also the model of our comparison study from Isola et al. (2017). The ultimate goal of our training loop is to minimize the result of the loss functions of both the generator and the discriminator. Figure 19 shows an example of the training loop and how its output looks like.

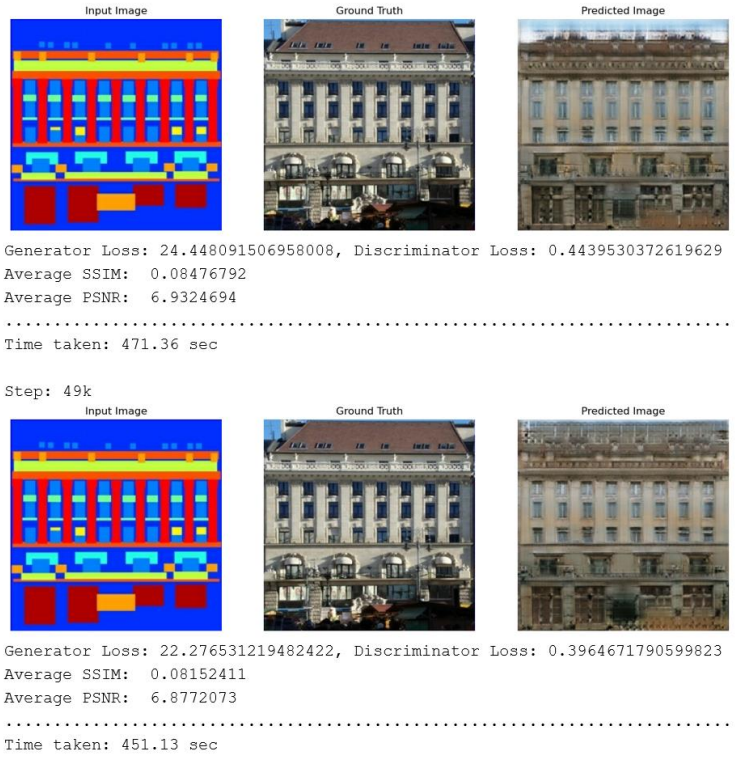

Figure 19. Example of training loop output

Chapter 4

Results & Evaluation Metrics

In this chapter, we are going to discuss the results of our implementation and the evaluation metrics used for evaluating these results. Furthermore, we will delve into a comprehensive analysis of both qualitative and quantitative aspects of our image-to-image translation model. We will showcase the visual results of our model by presenting sample images of input-output pairs alongside ground truth.

We will also make a comparison between the results of our model and the results of the Pix2Pix model by Isola et al. (2017) in order to see which model produces better results. Additionally, we have used the Structural Similarity Index (SSIM) and Peak Signal-to-Noise Ratio (PSNR), which are widely known metrics for evaluating images, to quantitatively assess the quality of our generated images. Other than these metrics, we have also conducted a survey to compare our results to the results of the Pix2Pix model by Isola et al. (2017) in which users select the generated image that is closest to the ground truth. We have opted to use the facades and maps datasets for testing our model which has been trained with a learning rate of 0.0002 for 50000 steps on both of these datasets.

4.1 Results

During this section, we are going to showcase the results of our model alongside the input and target images. In our training, there exists 3 images, the input image, the target image, and the generated image. When using our model for testing, we only provide it with an input image, and it will generate the output image according to the input image and how it gets translated.

Our goal is to generate an image from the input that looks as similar as possible to the target image. The target image is only given to the model during its training period, to let it know the results that we are looking for. Figure 20 shows some examples of our results using our facades test dataset with the input image and the corresponding generated image.

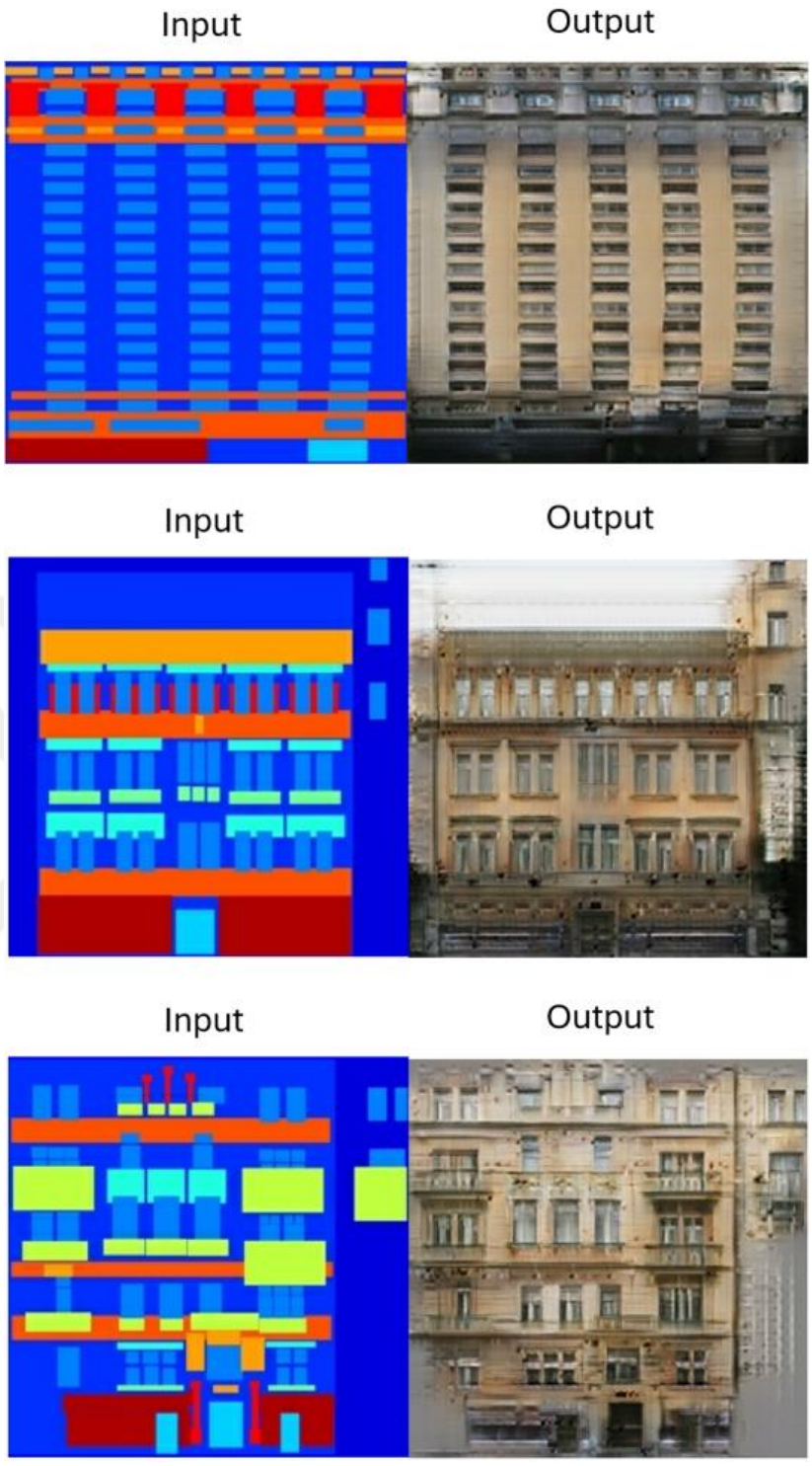

Figure 20. Example of our generator results with facades dataset

Our model is built in a way that it could be trained on a number of different datasets and still provide excellent image-to-image translation. Figure 21 shows some examples of our results using our maps test dataset.

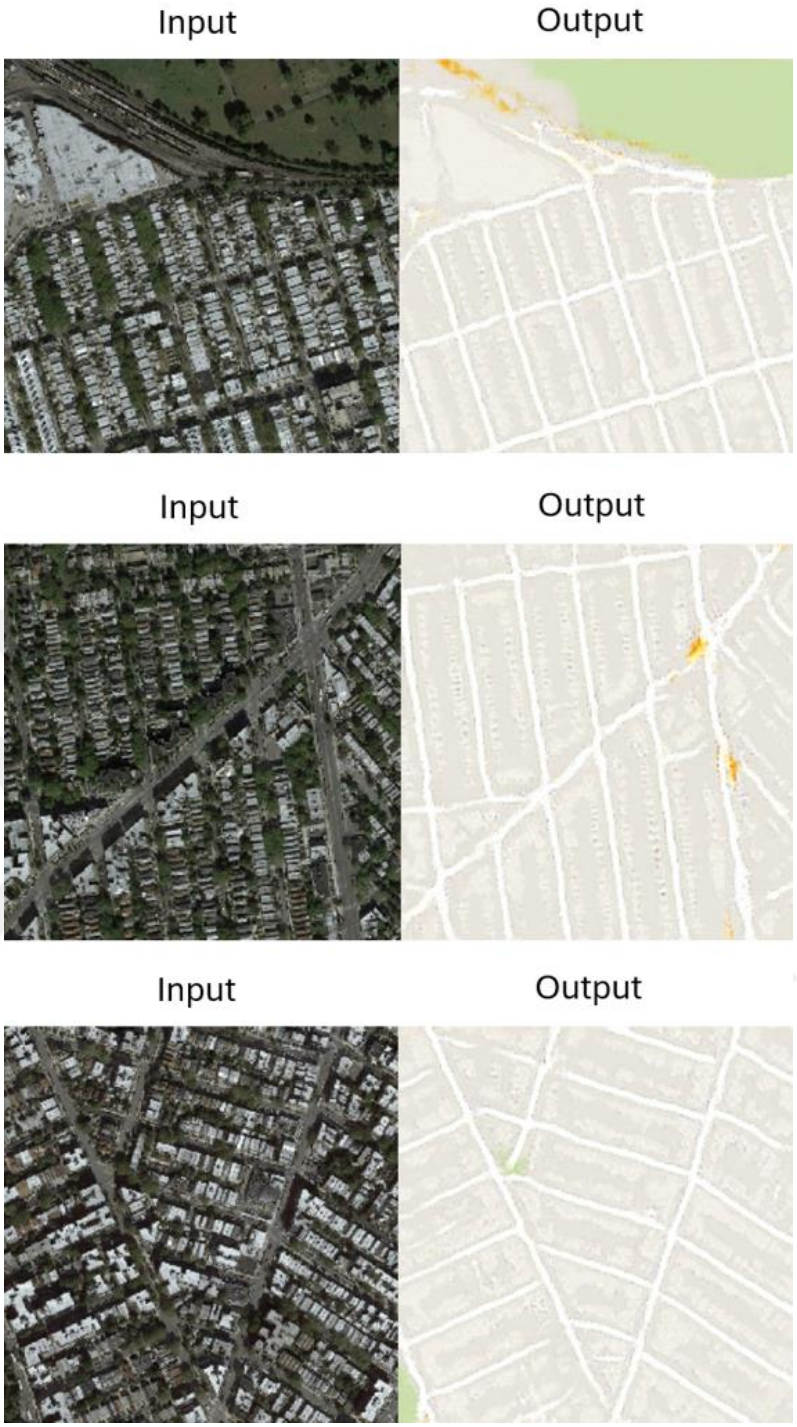

Figure 21. Example of our generator results with maps dataset

4.2 Quantitative Evaluation

In order to evaluate the quality of our generated images, we have decided to use the SSIM and PSNR as evaluation metrics.

4.2.1 Structural similarity index (SSIM). The SSIM is an evaluation metric that can be used to evaluate the similarity between two images, especially between a distorted image and a reference image (Wang et al., 2004). According to Wang et al. (2004), the way the similarity is calculated is by making use of three main components that show the perceptual similarity: luminance, contrast, and structure. The authors also added that the components are all used in its equation for comparing the perceptual similarity of two images. Figure 22 shows the equations for calculating these factors respectively (Wang et al., 2004).

$$l(\mathbf{x}, \mathbf{y}) = \frac{2(1+R)}{1+(1+R)^2+\frac{C_1}{\mu_x^2}}.$$

$$c(\mathbf{x}, \mathbf{y}) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(\mathbf{x}, \mathbf{y}) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}.$$

Figure 22. Equations for luminance, contrast, and structure (Wang et al., 2004)

Once all these factors are calculated separately, they are then combined together to calculate the value of the SSIM index. Figure 23 shows the equation for calculating the SSIM index (Wang et al., 2004).

$$S(\mathbf{x}, \mathbf{y}) = f\left(l(\mathbf{x}, \mathbf{y}), c(\mathbf{x}, \mathbf{y}), s(\mathbf{x}, \mathbf{y})\right).$$

Figure 23. Equation for calculating SSIM index (Wang et al., 2004)

Fundamentally, if two images are to be compared with each other using the SSIM and of them is of perfect quality while the other is an imitation of the first one, then the SSIM value will act as an evaluation metric for the second image's quality depending on how similar it is to the first image via the usage of luminance, contrast, and structure as shown in Figure 24 (Wang et al., 2004).

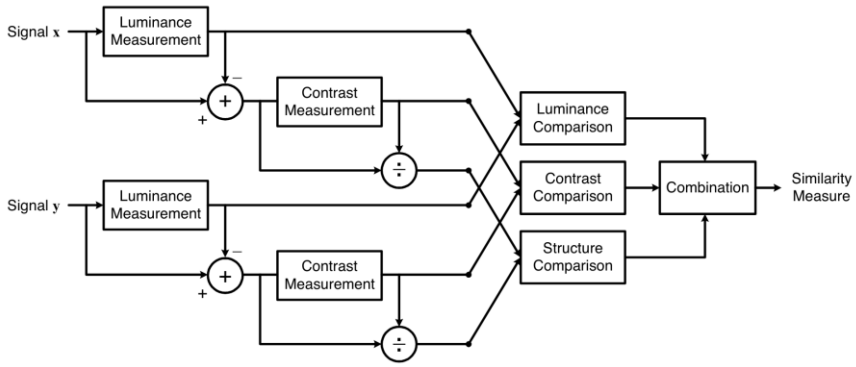

Figure 24. SSIM measurement system diagram (Wang et al., 2004)

4.2.2 Peak signal-to-noise ratio (PSNR). On the other hand, the PSNR is an evaluation metric used for calculating the quality of a reconstructed image in comparison to its original counterpart (Stéphane, 2009). It is further stated that the PSNR measures the quality by comparing the maximum possible intensity of the original image to the intensity of the reconstructed image in a fashion similar to the calculation of the Mean Squared Error (MSE). By comparing a received signal to the original source signal, the PSNR is able to calculate the accuracy of the received signal to the original (Fardo et al., 2016). Figure 25 shows the equation to calculate the MSE and PSNR respectively (Stéphane, 2009).

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

$$PSNR = 10\log_{10}\left(\frac{MAX^2}{MSE}\right),$$

Figure 25. Equations for MSE and PSNR (Stéphane, 2009)

4.2.3 Evaluation metrics calculation process. Our process of calculating the values of the SSIM and PSNR was simply going through all of the images stored in our test dataset and calculating the values using the TensorFlow framework for each image then calculating the average value for both metrics. However, there were some

pre- processing steps for the test images that needed to be followed before calculating the values. First, the images were converted to tensors in order for them to be compatible with the TensorFlow functions used for calculating the metrics. Then, the images were converted to be grayscale as the SSIM and PSNR metrics are usually calculated on greyscale images rather than colored images. Finally, the TensorFlow functions are called for calculating the metrics using the images that were altered from the pre-processing steps.

4.2.4 Quantitative evaluation results. During our training process, we calculate the average SSIM and the PSNR values to test our model after each 1000 steps are completed. This calculation is done on the testing dataset in which we loop through the whole dataset, calculate the metrics after each 1000 steps and record the values for later use. We also calculate the values of the generator loss and the discriminator loss. Since we are training our model in 50000 steps, we will have 50 SSIM and PSNR values after the training process is done.

During the training, the values are saved so they can be used for evaluating the results of the model in terms of quality. After the training process was done, we used the saved values and visualized them using a line graph in order to see how the SSIM and PSNR values improved after each iteration in the training loop. You can see the line graphs for the SSIM and PSNR values during the training process depending on the steps in Figure 26 and Figure 27 respectively in order to see how our metrics improve after each training step is completed when trained on the facades dataset.

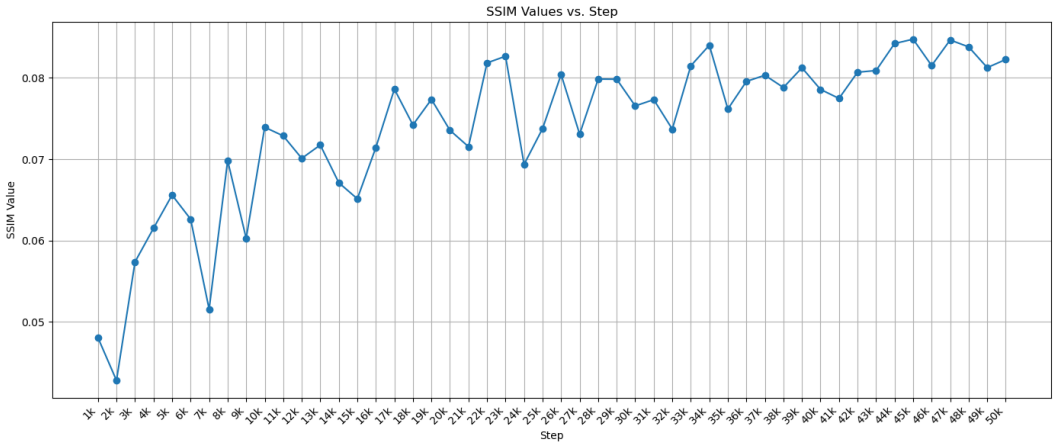

Figure 26. Line graph of our SSIM values over training steps

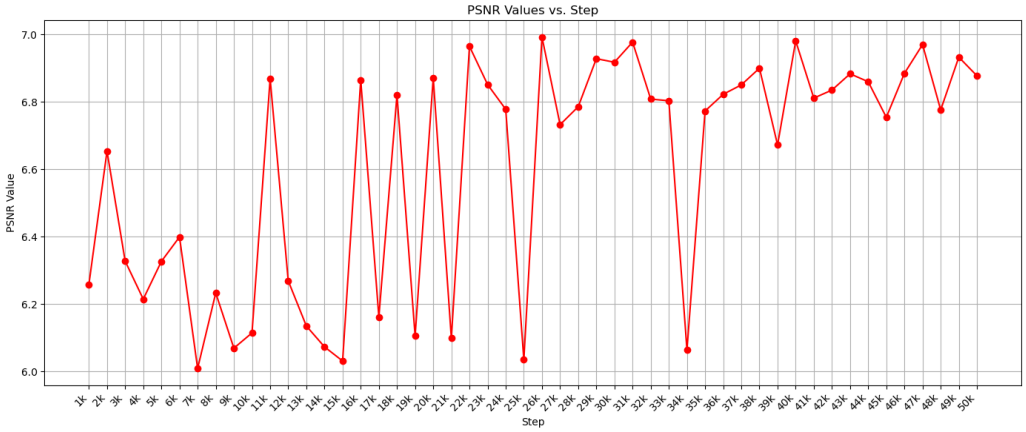

Figure 27. Line graph of our PSNR values over training steps

After the training was done, we went through all of the image pairs inside of the test dataset, generated the output image using the trained generator, and then calculated the SSIM and PSNR values for each one of them in comparison to target image. After that, we calculated the average of both values in which SSIM was 0.082270745 and PSNR was 6.8772073.

4.3 Training Loss Results

In this section, we will look at the generator loss and the discriminator loss of our training loop and how they compare to the loss of the original study. We are also going to graph these losses in a line graph to analyze them visually.

During our training process, we have used two loss functions, one for the generator, and one for the discriminator. The loss of the generator checks on whether the generated images are close to the target or not while the discriminator loss checks on whether the discriminator is able to distinguish fake images from real images. Losses usually start with high values and go down as the training continues. This is because at the start of the training process, the generator and the discriminator will have inadequate training resulting in poor performance in their respective goals. As the training continues and the networks improve, the loss is expected to go down to reflect these improvements. In our case, we have calculated the generator loss and the discriminator loss after each 1000 steps in our training loop. After that, we saved these records and used them to evaluate the performance of our model. Figure 28 and Figure 29 show the generator loss and the discriminator loss depending on the step respectively when trained on the facades dataset.

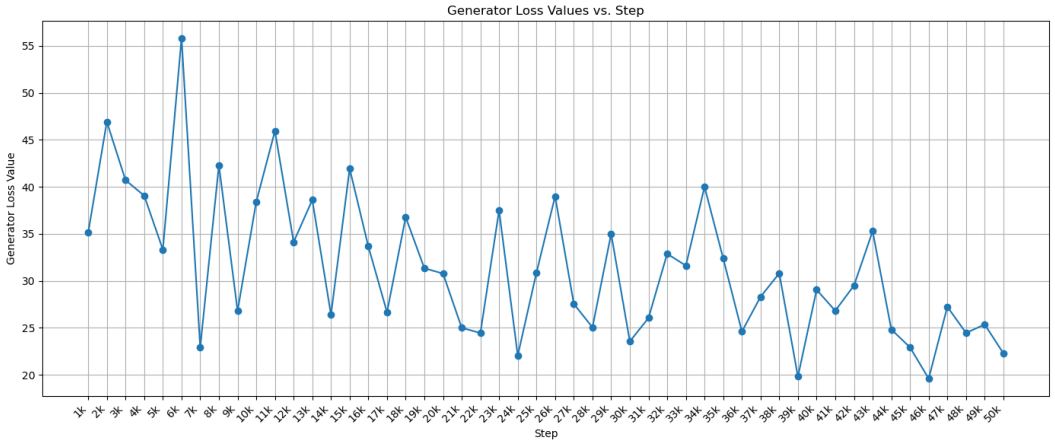

Figure 28. Line graph of our generator loss over training steps

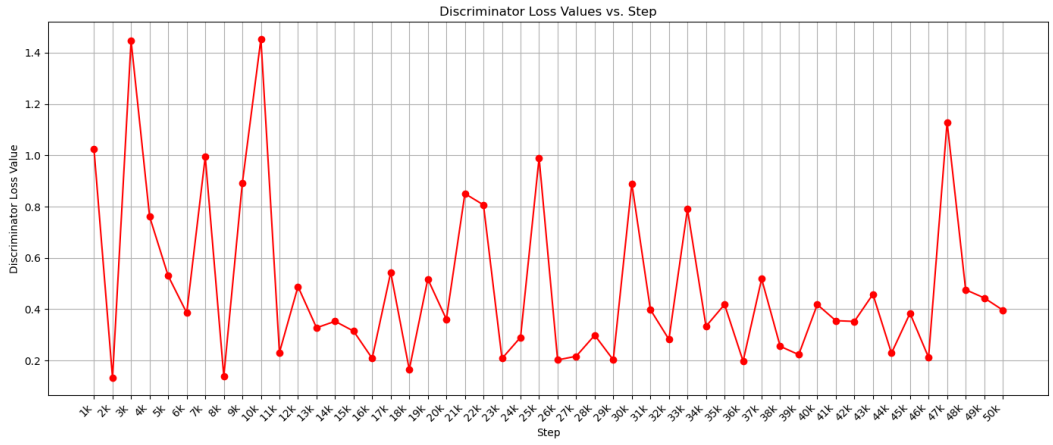

Figure 29. Line graph of our discriminator loss over training steps

As can be seen by the results of Figure 28, our generator loss starts with a value of 35.1 during the first 1000 steps. However, by the end of the training, our generator loss dropped to 22.2. On the other hand, Figure 29 shows our discriminator loss which started with a value of 1.02 during the first 1000 steps, then dropped to a value of 0.39 by the end of the training process. You can see in the figures how the values of the generator loss and discriminator loss contrast each other as the generator and the discriminator fight over whether the generated image is real or not in which each network improves itself according to the result.

4.4 Comparing Results with Original Study

In this section, we are going to compare our results in terms of presentation and evaluation metrics with the results of the original study of Isola et al. (2017). We are already using the same datasets as the datasets they have used in their study, making it easier for us to do the comparison between the models.

4.4.1 Comparing presentation results. We are going to compare the results of our model with the results from the original study of Isola et al. (2017). We are going to compare the generated images from both of the generators and how they compare to each other in similarity to the target image. We will also show the input images that prompted the generation of these images. This comparison will be done on the test dataset in which each sample includes an input image, and a target image.

Figure 30 shows the comparison between the results of our generator and the original study's generator with some examples from the facades testing dataset.

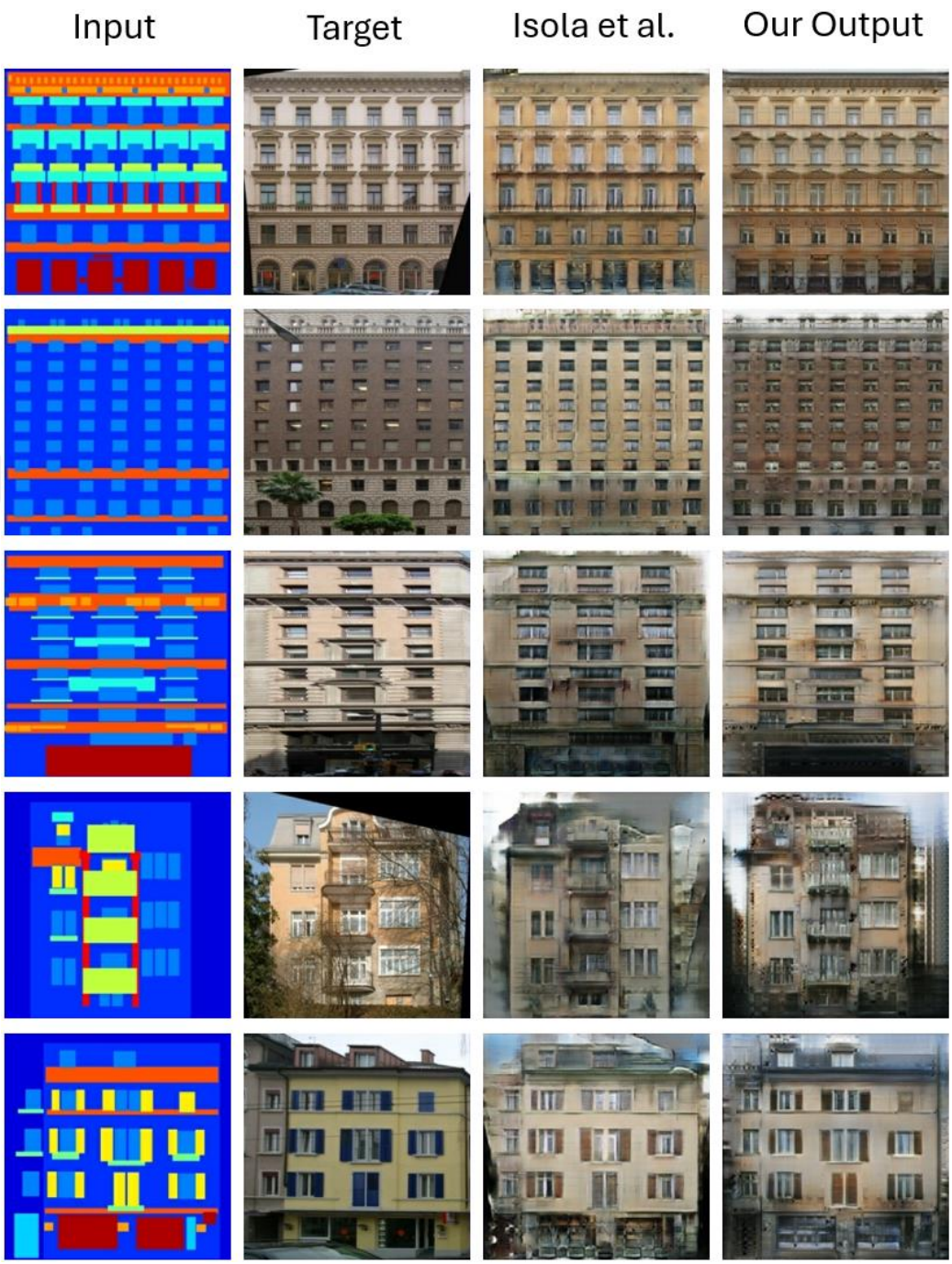

Figure 30. Comparison of results with Isola et al. On facades dataset

Since both models work on multiple datasets, we also decided to showcase comparison between them using the maps dataset in Figure 31.

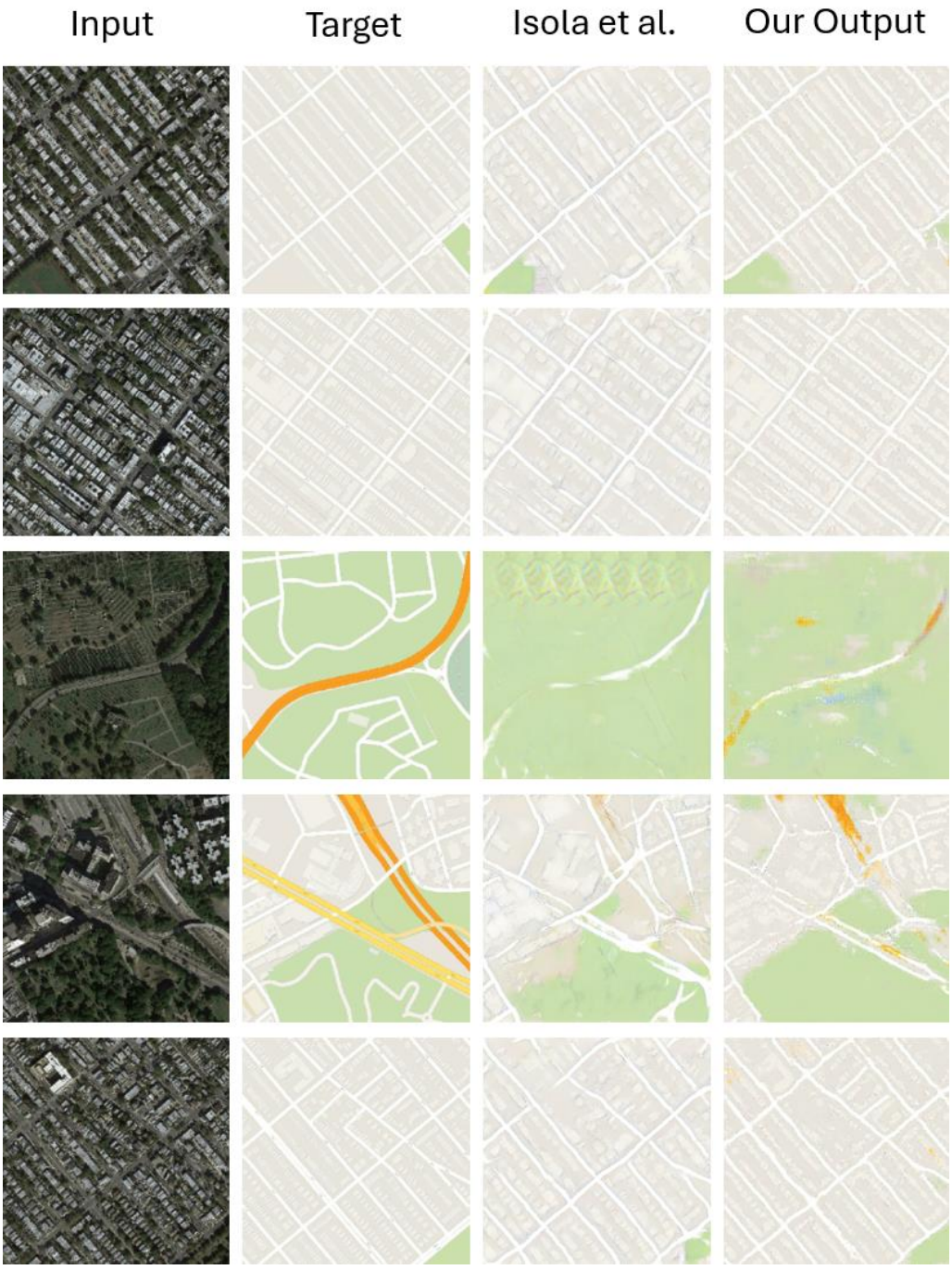

Figure 31. Comparison of results with Isola et al. On maps dataset

4.4.2 Comparing evaluations metrics. In order to compare our results with the Pix2Pix study by Isola et al. (2017), we calculated the SSIM and PSNR values using their model in the same way as in our implementation with 50000 steps. We also graphed the values in the same way as before depending on the steps and they can be seen in Figure 32 and Figure 33 respectively when they are trained on the facades dataset.

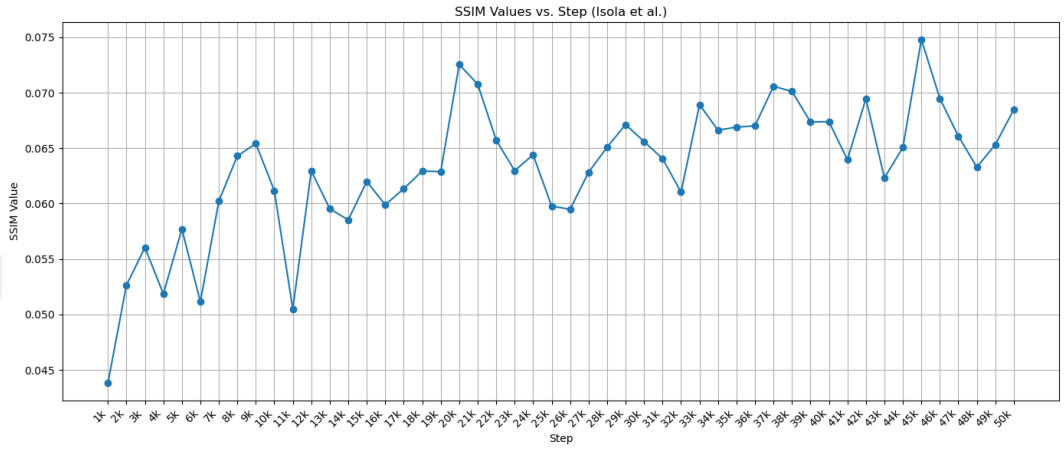

Figure 32. Line graph of Isola et al. SSIM values over training steps

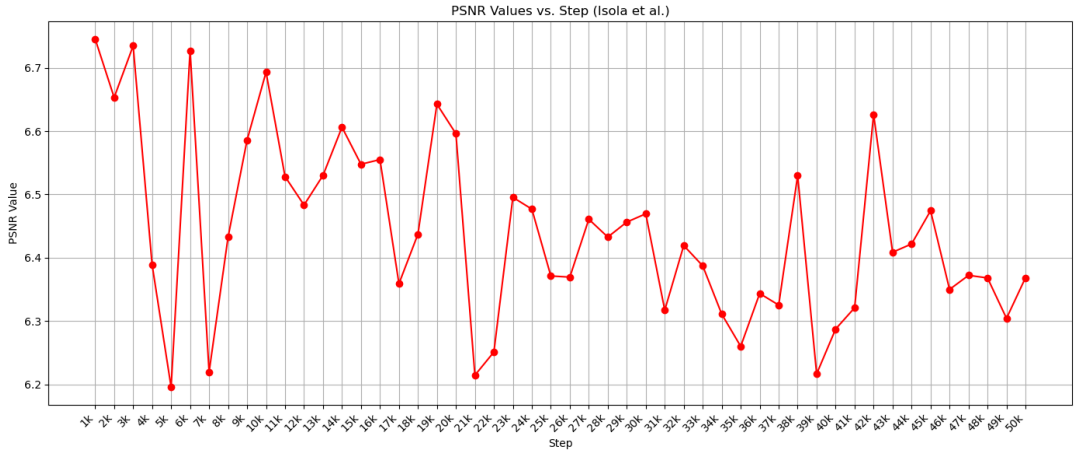

Figure 33. Line graph of Isola et al. PSNR values over training steps

Furthermore, we also calculated the average SSIM and PSNR values on the testing dataset after the training was completed and the results were 0.068490095 and 6.3685718 respectively. As can be seen from the results, our average values are higher thus showing our model is capable of generating images that are of higher quality and

are closer to the target image than in the original study. Table 4 shows the comparison of the evaluation metrics between our model and Isola et al. (2017).

Table 4

Comparison of Evaluation Metrics with Isola et al. (2017)

Evaluation Metric	Isola et al. (2017)	Our Model
Average SSIM	0.068490095	0.082270745
Average PSNR	6.3685718	6.8772073

4.4.3 Comparing training loss results. In order to compare the results of the training loss of our generator and discriminator with the original study of Isola et al. (2017), we also calculated the loss of their model in a similar way which was trained on 50000 steps and the losses are calculated after each 1000 steps. After that, they are saved to use them for the comparison with our model. Figure 34 and Figure 35 are line graphs that show the generator loss and the discriminator loss of the original stupid depending on the step when they are trained on the facades dataset.

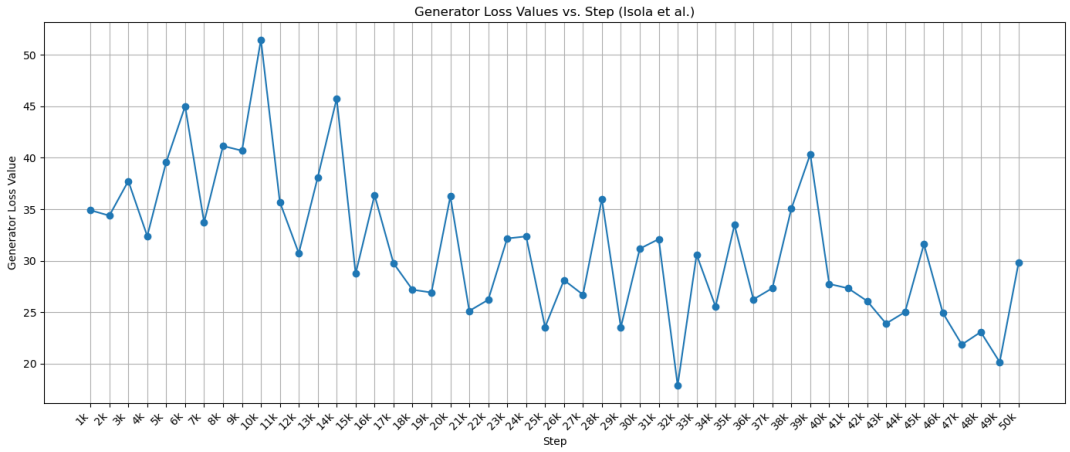

Figure 34. Line graph of Isola et al. Generator loss over training steps

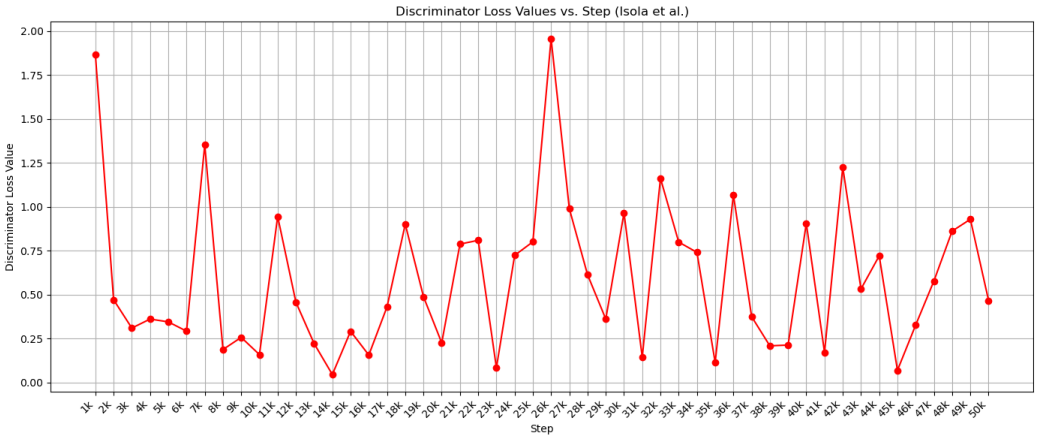

Figure 35. Line graph of Isola et al. Discriminator loss over training steps

As can be seen in Figure 34, the generator loss of the original study started with a value of 34.9 during the first 1000 steps then it dropped to a value of 29.8 by the end of the training process. However, this value is higher than the loss of our generator network, which was equal to a value of 22.2 by the end of its training. This indicates that our generator network has better performance than the original study's generator. On the other hand, Figure 35 shows the discriminator loss of the original study which started with a value of 1.86 during the first 1000 steps but dropped to a value of 0.46 by the end of the training process. This also shows how our discriminator performed better due to it having a loss of 0.39 by the end of the training process. Table 5 shows the comparison of the loss between our model and Isola et al. (2017).

Table 5

Comparison of Training Loss with Isola et al. (2017)

Loss	Isola et al. (2017)	Our Model
Generator Loss	29.8	22.2
Discriminator Loss	0.46	0.39

4.5 Survey Results

In addition to our evaluation metrics, we also decided to evaluate our results via comparing them with the results of the original study in a public survey. This is done in order to take into account human visual assessment which could provide

different results than the results provided by the evaluation metrics. In this section, we are going to explore the conducted survey and showcase the results we got from the participants.

4.5.1 Structure of the survey. In order to remove any bias from the results and make the survey fair to both models, we decided to get 25 random pairs of inputs and targets from the facades testing dataset and then got the corresponding generated images for these inputs from our model and the Pix2Pix model by Isola et al. (2017). This left us with 3 images for each set: the ground truth image, our generated image, and the Pix2Pix generated image from Isola et al. (2017). Each set was stored in a separate folder that was named after a hashed value we got from the matrix of the target image. After that, we got all of these image sets into the survey in order to remove any selection bias from our side.

The survey was conducted between 22nd of April 2024 and 15th of May 2024 on Google Forms with 60 total participants that answered the questions. The survey consisted of 25 multi-choice questions in which all of these questions have the same structure and participants will be limited to selecting only one option for each question. Each question will feature a ground truth image, accompanied by the following title: 'Which of the two image options best matches the image below?". Moreover, there will be only two image options for each question, one generated by our model and one generated by the original study. Both of these image options are generated according to the ground truth image in the question using an input image. The participants are asked to select the generated image that looks the most similar to the ground truth image given in each question. During the course of the survey, no personal information will be collected from the participants. The only data collected is the selection of the image from the options. Figure 36 shows an example question from the survey to showcase the structure of the questions. All of the other questions had the same structure as the one in the figure.

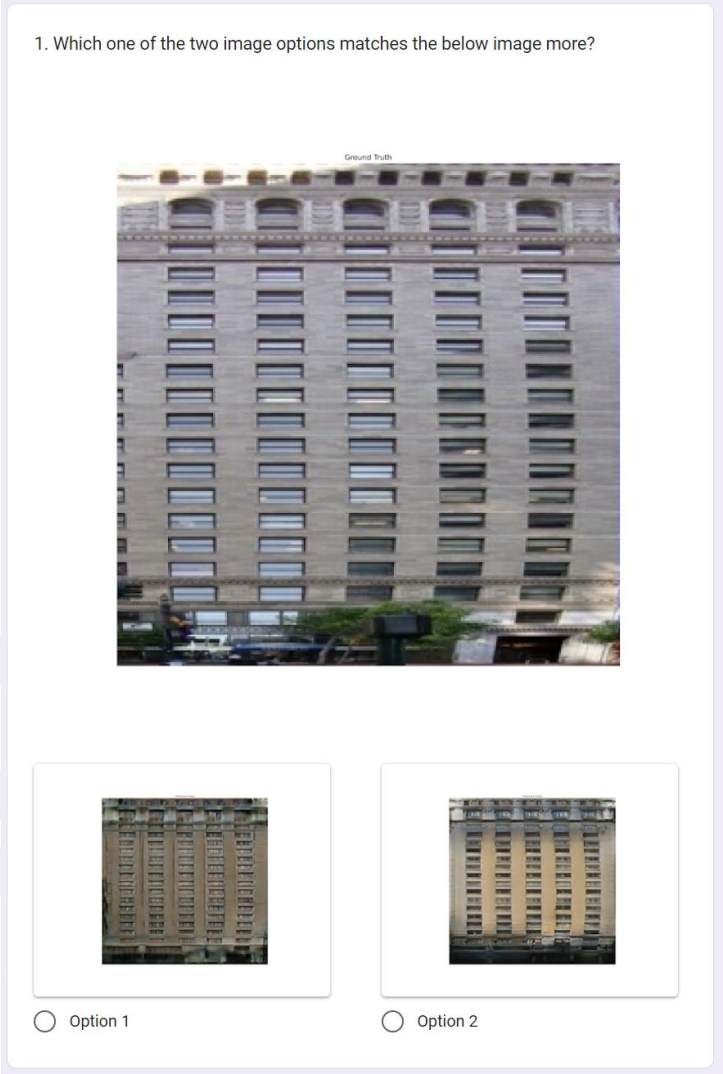

Figure 36. Example showcasing our survey structure

4.5.2 Results of the survey. During the course of the survey, we received 60 different responses in which the participants have each answered all of the questions with their selected images as the ones closest to the ground truth. Overall, the results of the survey were in favor of our model rather than the model from the original of Isola et al. (2017). Out of the 1496 different possible answers for all the questions, 841 of them were for our model while only 655 were for the Pix2Pix model by Isola et al. (2017). This is equivalent to 56.2% of the total answers favored our model. Figure 37 shows the results in terms of the number of answers in a pie chart.

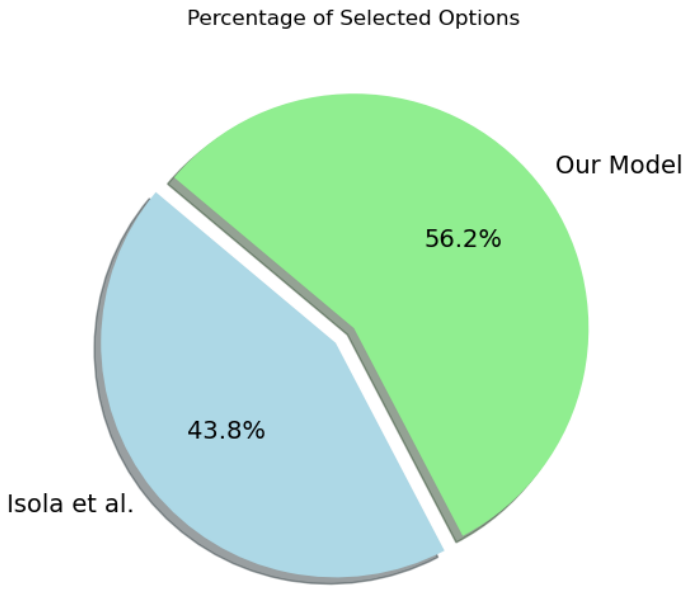

Figure 37. Survey results (percentage of total answers)

Furthermore, out of the total 25 questions that were presented in the survey, 15 of them had results that were mostly in favor of our model, 8 of them were in favor of the other study, and 2 of them were a draw. Figure 38 shows these results as well in a horizontal bar chart.

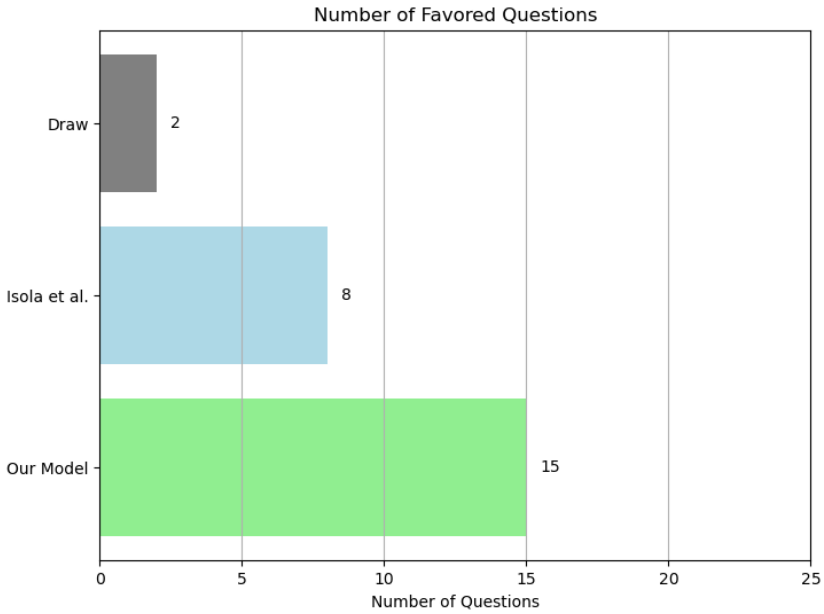

Figure 38. Survey results (number of favored questions)

Table 6 shows a comparison between our model and the original study's model in terms of survey results.

Table 6

Comparison of Survey Results with Isola et al. (2017)

Survey Results	Isola et al. (2017)	Our Model
Selection Percentage	43.8%	56.2%
Favored Questions (Out of 25)	8	15

4.6 Results Discussion & Interpretation

The results shown by our image-to-image translation model turned out to be promising and met all our expectations, in which it demonstrated high efficiency and performance. The visual inspection of the generated images that were generated by our model has shown that they had high resemblance to the target images and were of high quality with accurate color representation and semantic consistency. Furthermore, the evaluation metrics employed by our study, SSIM and PSNR, has yielded better results than when they were used in the original study which proved that our model is able to produce higher quality images that are closer in structure to the target images. Moreover, the generator loss and the discriminator loss were shown to be less than in the original study which proves the high performance of our model.

The successful outcome of our study proves that our proposed methodology for image-to-image translation using cGANs with a generator architecture that combines the U-Net architecture with a basic ResNet architecture, is a viable method that can be utilized in many other applications related to image processing with DL.

Chapter 5

Conclusion

5.1 Key Findings & Contributions

Our study has explored a new implementation of image-to-image translation using a cGAN model with a new hybrid generator architecture. Furthermore, we focused on evaluating our model using several evaluation metrics and also by comparing it other studies that also focused on the same domain of image-to-image translation. The results of our model showed the efficiency and performance of our proposed methodology in image-to-image translation. We have validated our model via the usage of evaluation metrics such as SSIM and PSNR while also comparing it with other studies by also using the same metrics in their models which yielded better results for our model.

We have also contributed to the methodology by creating a new method for implementing image-to-image translation which uses a hybrid architecture for the generator that combines the U-Net architecture with a basic ResNet architecture while also employing a PatchGAN architecture for the discriminator with layer normalization applied in both approaches. These architectural choices have shown improvements in the quality of the generated images and made them look more similar to the target images.

5.2 Limitations & Future Work

Despite meeting all of our expectations, our study acknowledges that our implementation still suffers from certain limitations that need to be addressed in future work. One of these limitations is the fact that our training process takes a lot of computational resources as well as a lot of time for it to be completed. Another limitation is in regard to the difficulty in generating more complex structures that deviate from the rest of the dataset. Future work should focus on overcoming these limitations by optimizing the model for a better training process that takes less resources and time while also being able to generate more complex structures.

5.3 Closing Remarks

In conclusion, our study has provided a comprehensive exploration of image-to-image translation using cGANs with a new methodology involving a hybrid architecture for the generator. Our study explored theoretical perspectives on the field of image-to-image translation in addition to practical applications. We have learned a great deal about the difficulties, and possible directions for future research in this area by applying extensive investigation and testing. We would like to express our gratitude to everyone who has helped with this study and wish to see more advancements and developments in the field of image-to-image translation in the years to come.

REFERENCES

IBM. (2021). What is Computer Vision? Retrieved from https://www.ibm.com/topics/computer-vision

Isola, P., Zhu, J.Y., Zhou, T., & Efros, A.A. (2017). Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1125-1134).

Mohammed, E.U.R., Reddy, S.N., & Waseem, M.S. (2022). A Comprehensive Literature Review on Convolutional Neural Networks.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

Kamil, A., & Shaikh, T. (2019, December). Literature Review of Generative models for Image- to-Image translation problems. In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE) (pp. 340-345). IEEE.

Hoyez, H., Schockaert, C., Rambach, J., Mirbach, B., & Stricker, D. (2022). Unsupervised image-to-image translation: A review. Sensors, 22(21), 8540.

Mirza, M., & Osindero, S. (2014, November 6). *Conditional Generative Adversarial Nets*. arXiv.org. https://doi.org/10.48550/arXiv.1411.1784

Perarnau, G., van de Weijer, J., Raducanu, B., & Álvarez, J. M. (2016, November 19). *Invertible Conditional GANs for image editing*. arXiv.org. https://doi.org/10.48550/arXiv.1611.06355

Odena, A., Olah, C., & Shlens, J. (2017, July 20). *Conditional Image Synthesis With Auxiliary Classifier GANs*. arXiv.org. https://doi.org/10.48550/arXiv.1610.09585

Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv.org. https://doi.org/10.48550/arXiv.1505.04597

Ji, Y., Zhang, H., & Jonathan Wu, Q. M. (2018). Saliency detection via conditional adversarial image-to-image network. *Neurocomputing*, *316*, 357–368. https://doi.org/10.1016/j.neucom.2018.08.013

Mao, X., Wang, S., Zheng, L., & Huang, Q. (2018). Semantic invariant cross-domain image generation with generative adversarial networks. *Neurocomputing*, *293*, 55–63. https://doi.org/10.1016/j.neucom.2018.02.092

Gan, Y., Gong, J., Ye, M., Qian, Y., & Liu, K. (2018). Unpaired cross domain image translation with Augmented Auxiliary Domain Information. *Neurocomputing*, *316*, 112–123. https://doi.org/10.1016/j.neucom.2018.07.057

Cho, Y., Malav, R., Pandey, G., & Kim, A. (2019). DehazeGAN: Underwater haze image restoration using unpaired image-to-image translation. *IFAC-PapersOnLine*, *52*(21), 82–85. https://doi.org/10.1016/j.ifacol.2019.12.287

Mo, S., Cho, M., & Shin, J. (2019, January 2). *InstaGAN: Instance-aware Image-to-Image Translation*. arXiv.org. https://doi.org/10.48550/arXiv.1812.10889

Yang, D., Hong, S., Jang, Y., Zhao, T., & Lee, H. (2019, January 25). *Diversity-Sensitive Conditional Generative Adversarial Networks*. arXiv.org. https://doi.org/10.48550/arXiv.1901.09024

Liu, M.-Y., Huang, X., Mallya, A., Karras, T., Aila, T., Lehtinen, J., & Kautz, J. (2019, September 9). *Few-Shot Unsupervised Image-to-Image Translation*. arXiv.org. https://doi.org/10.48550/arXiv.1905.01723

Xu, W., Shawn, K., & Wang, G. (2019). Toward learning a unified many-to-many mapping for diverse image translation. *Pattern Recognition*, *93*, 570–580. https://doi.org/10.1016/j.patcog.2019.05.017

Ye, L., Zhang, B., Yang, M., & Lian, W. (2019). Triple-translation gan with multi-layer sparse representation for face image synthesis. *Neurocomputing*, *358*, 294–308. https://doi.org/10.1016/j.neucom.2019.04.074

Choi, Y., Uh, Y., Yoo, J., & Ha, J.-W. (2020, April 26). *StarGAN v2: Diverse Image Synthesis for Multiple Domains*. arXiv.org. https://doi.org/10.48550/arXiv.1912.01865

Armanious, K., Jiang, C., Fischer, M., Küstner, T., Hepp, T., Nikolaou, K., Gatidis, S., & Yang, B. (2020). Medgan: Medical image translation using Gans. *Computerized Medical Imaging and Graphics*, *79*, 101684. https://doi.org/10.1016/j.compmedimag.2019.101684

Hicsonmez, S., Samet, N., Akbas, E., & Duygulu, P. (2020a). Ganilla: Generative adversarial networks for image to illustration translation. *Image and Vision Computing*, *95*, 103886. https://doi.org/10.1016/j.imavis.2020.103886

Zhu, J.-Y., Park, T., Isola, P., & Efros, A. A. (2020, August 24). *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. arXiv.org. https://doi.org/10.48550/arXiv.1703.10593

Xia, W., Yang, Y., & Xue, J.-H. (2020). Unsupervised multi-domain multimodal image-to-image translation with explicit domain-constrained disentanglement. *Neural Networks*, *131*, 50–63. https://doi.org/10.1016/j.neunet.2020.07.023

Dhont, J., Verellen, D., Mollaert, I., Vanreusel, V., & Vandemeulebroucke, J. (2020). RealDRR – rendering of realistic digitally reconstructed radiographs using locally trained image-to-image translation. *Radiotherapy and Oncology*, *153*, 213–219. https://doi.org/10.1016/j.radonc.2020.10.004

Emami, H., Aliabadi, M. M., Dong, M., & Chinnam, R. B. (2020, December 30). *Spa-gan: Spatial attention gan for image-to-image translation*. arXiv.org. https://doi.org/10.48550/arXiv.1908.06616

Wang, Y., Gonzalez-Garcia, A., Herranz, L., & van de Weijer, J. (2021). Controlling biases and diversity in diverse image-to-image translation. *Computer Vision and Image Understanding*, *202*, 103082. https://doi.org/10.1016/j.cviu.2020.103082

Marzullo, A., Moccia, S., Catellani, M., Calimeri, F., & Momi, E. D. (2021). Towards realistic laparoscopic image generation using image-domain translation. *Computer Methods and Programs in Biomedicine*, *200*, 105834. https://doi.org/10.1016/j.cmpb.2020.105834

Shao, M., Zhang, Y., Liu, H., Wang, C., Li, L., & Shao, X. (2021). DMDIT: Diverse multi-domain image-to-image translation. *Knowledge-Based Systems*, *229*, 107311. https://doi.org/10.1016/j.knosys.2021.107311

Lin, J., Xia, Y., Liu, S., Zhao, S., & Chen, Z. (2021). Zstgan: An adversarial approach for unsupervised zero-shot image-to-image translation. *Neurocomputing*, *461*, 327–335. https://doi.org/10.1016/j.neucom.2021.07.037

Yan, L., Zheng, W., Wang, F.-Y., & Gou, C. (2021). Joint image-to-image translation with denoising using enhanced generative adversarial networks. *Signal Processing: Image Communication*, *91*, 116072. https://doi.org/10.1016/j.image.2020.116072

Nie, X., Ding, H., Qi, M., Wang, Y., & Wong, E. K. (2021). Urca-Gan: Upsample residual channel-wise attention generative adversarial network for image-to-

image translation. *Neurocomputing*, *443*, 75–84. https://doi.org/10.1016/j.neucom.2021.02.054

Liu, H., Chen, L., Sui, H., Zhu, Q., Lei, D., & Liu, S. (2021). Unsupervised multi-domain image translation with domain representation learning. *Signal Processing: Image Communication*, *99*, 116452. https://doi.org/10.1016/j.image.2021.116452

Barzilay, N., Shalev, T. B., & Giryes, R. (2021). Miss Gan: A multi-illustrator style generative adversarial network for image to illustration translation. *Pattern Recognition Letters*, *151*, 140–147. https://doi.org/10.1016/j.patrec.2021.08.006

Yang, F., Wang, Y., Herranz, L., Cheng, Y., & Mozerov, M. (2022, August 9). *A Novel Framework for Image-to-image Translation and Image Compression*. arXiv.org. https://doi.org/10.48550/arXiv.2111.13105

Platscher, M., Zopes, J., & Federau, C. (2022). Image translation for medical image generation: Ischemic stroke lesion segmentation. *Biomedical Signal Processing and Control*, *72*, 103283. https://doi.org/10.1016/j.bspc.2021.103283

Varghese, S., & Hoskere, V. (2023). Unpaired image-to-image translation of structural damage. *Advanced Engineering Informatics*, *56*, 101940. https://doi.org/10.1016/j.aei.2023.101940

Chen, Z., Cai, L., Chen, C., Fu, X., Yang, X., Yuan, B., Lu, Q., & Zhou, H. (2023). Unsupervised image-to-image translation in multi-parametric MRI of bladder cancer. *Engineering Applications of Artificial Intelligence*, *124*, 106547. https://doi.org/10.1016/j.engappai.2023.106547

Grebo, A., Krstulović-Opara, L., & Domazet, Ž. (2023). Thermal to digital image correlation image to image translation with Cyclegan and pix2pix. *Materials Today: Proceedings*, *93*, 752–760. https://doi.org/10.1016/j.matpr.2023.06.219

Al-Hindawi, F., Soori, T., Hu, H., Siddiquee, Md. M., Yoon, H., Wu, T., & Sun, Y. (2023). A framework for generalizing critical heat flux detection models using unsupervised image-to-image translation. *Expert Systems with Applications*, *227*, 120265. https://doi.org/10.1016/j.eswa.2023.120265

Kang, M., Chikontwe, P., Won, D., Luna, M., & Park, S. H. (2023). Structure-preserving image translation for multi-source medical image domain adaptation. *Pattern Recognition*, *144*, 109840. https://doi.org/10.1016/j.patcog.2023.109840

Whang, S. E., Roh, Y., Song, H., & Lee, J. G. (2023). Data collection and quality challenges in deep learning: A data-centric ai perspective. *The VLDB Journal*, *32*(4), 791-813.

University of California, Berkeley. (2018). Pix2Pix datasets. Retrieved from https://efrosgans.eecs.berkeley.edu/pix2pix/datasets/

Fan, C., Chen, M., Wang, X., Wang, J., & Huang, B. (2021). A review on data preprocessing techniques toward efficient and reliable knowledge discovery from building operational data. *Frontiers in Energy Research, 9*. https://doi.org/10.3389/fenrg.2021.652801

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

Maharana, K., Mondal, S., & Nemade, B. (2022). A review: Data pre-processing and data augmentation techniques. *Global Transitions Proceedings*, *3*(1), 91–99. https://doi.org/10.1016/j.gltp.2022.04.020

Wu, D., Wang, Y., Xia, S. T., Bailey, J., & Ma, X. (2020). Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, *13*(4), 600–612. https://doi.org/10.1109/tip.2003.819861

Stéphane, M. (2009). Compression. *A Wavelet Tour of Signal Processing*, 481–533. https://doi.org/10.1016/b978-0-12-374370-1.00014-8

Fardo, F. A., Conforto, V. H., de Oliveira, F. C., & Rodrigues, P. S. (2016). A formal evaluation of PSNR as quality measurement parameter for image segmentation algorithms. *arXiv preprint arXiv:1605.07116*.