

Fatma Nur TAKIŞ

M.Sc. Thesis

AGU 2024

GENERATING LOST URBAN FABRIC:
EXPLORING GENERATIVE
ADVERSARIAL NETWORKS AS A TOOL
FOR URBAN DESIGN

M.Sc. THESIS

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF ABDULLAH GUL UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Fatma Nur TAKIŞ

May 2025

GENERATING LOST URBAN FABRIC: EXPLORING GENERATIVE ADVERSARIAL NETWORKS AS A TOOL FOR URBAN DESIGN

A THESIS

SUBMITTED TO THE DEPARTMENT OF ARCHITECTURE
AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE OF
ABDULLAH GUL UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE

By

Fatma Nur TAKIŞ

May 2025

SCIENTIFIC ETHICS COMPLIANCE

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

Name-Surname: Fatma Nur TAKIŞ

Signature :

X X X X

REGULATORY COMPLIANCE

M.Sc. thesis titled Generating Lost Urban Fabric: Exploring Generative Adversarial Networks As A Tool For Urban Design has been prepared in accordance with the Thesis Writing Guidelines of the Abdullah Gül University, Graduate School of Engineering & Science.

Prepared By
Fatma Nur TAKIŞ
Signature

Advisor
Asst. Prof. Sinan AKYÜZ
Signature

Head of the Architecture Program
Doc. Dr. Buket METİN
Signature

ACCEPTANCE AND APPROVAL

M.Sc. thesis titled Generating Lost Urban Fabric: Exploring Generative Adversarial Networks As A Tool For Urban Design and prepared by Fatma Nur TAKIŞ has been accepted by the jury in the Architecture Graduate Program at Abdullah Gül University, Graduate School of Engineering & Science.

07 /04 / 2025

(Thesis Defense Exam Date)

JURY:

Advisor: Asst. Prof. Sinan AKYÜZ

Member: Prof. Burak ASİLİSKENDER

Member: Doc. Dr. Orkan Zeynel GÜZELCİ.....

APPROVAL:

The acceptance of this M.Sc. thesis has been approved by the decision of the Abdullah Gül University, Graduate School of Engineering & Science, Executive Board dated /..... / and numbered

..... / /

(Date)

Graduate School Dean
Prof. İrfan ALAN

ABSTRACT

GENERATING LOST URBAN FABRIC: EXPLORING GENERATIVE ADVERSARIAL NETWORKS AS A TOOL FOR URBAN DESIGN

Fatma Nur TAKIŞ
MSc. In Architecture
Advisor: Asst. Prof. Sinan AKYÜZ
May 2025

This research investigates the potential of Generative Adversarial Networks as a design tool for generating the lost urban fabric. Focusing on the city center of Hatay, Türkiye, which experienced extensive structural loss due to the 2023 Türkiye-Syria earthquakes, the study employs a Pix2PixHD architecture to generate urban patterns. Using pre-disaster maps as the training dataset, the research aims to generate urban patterns that reflect Hatay's architectural identity and spatial continuity.

The thesis adopts a hybrid methodology, combining quasi-experimental and case study approaches. It begins with an explanation of the fundamentals of Machine Learning and their relevance to urban studies, followed by detailed documentation of the model training and validation processes. The generated outputs are analyzed using quantitative metrics, including the Fréchet Inception Distance (FID) score to evaluate the model's ability to replicate urban patterns, and the Structural Similarity Index Measure (SSIM) to assess visual realism and structural integrity of the outputs. The study further investigates how different training dataset scales (1:2000, 1:3000, and 1:5000) impact the model's performance.

This research demonstrates the potential of GANs as a tool for urban design in a post-disaster context and provides insights into how dataset scale influences the success of the model in generating realistic urban patterns.

Keywords: Generative Design, Artificial Intelligence, Generative Adversarial Networks, Urban Fabric Loss, Urban Design

ÖZET

KAYIP KENTSEL DOKUYU ÜRETMEK: KENTSEL TASARIMDA BİR ARAÇ OLARAK ÜRETKEN ÇEKİŞMELİ AĞLARI KEŞFETMEK

Fatma Nur TAKIŞ
Mimarlık Anabilim Dalı Yüksek Lisans
Tez Danışmanı: Dr. Öğr. Üyesi Sinan AKYÜZ
Mayıs 2025

Bu araştırma, Generative Adversarial Networks'ün (GAN'ler) kaybolan kentsel dokuyu yeniden üretmek için bir tasarım aracı olması potansiyelini incelemektedir. Çalışma, 2023 Kahramanmaraş depremleri nedeniyle büyük yapısal kayıplar yaşayan Hatay şehir merkezine odaklanmaktadır ve kentsel dokuları üretmek için Pix2PixHD mimarisini kullanmaktadır. Araştırma, eğitim veri seti olarak afet öncesi kentsel haritaları kullanarak, Hatay'ın mekânsal sürekliliğini yansıtan kentsel dokular üretmeyi hedeflemektedir.

Tez, yarı deneysel ve durum çalışması yaklaşımlarını birleştiren hibrit bir metodoloji benimsemektedir. Çalışma, makine öğrenmesinin temel prensipleri ve kentsel çalışmalardaki önemine değinmektedir, ardından modelin eğitim ve doğrulama süreçlerinin ayrıntılı bir şekilde belgelenmesiyle devam etmektedir. Modelin kentsel dokuları yeniden üretme yeteneğini değerlendirmek için Fréchet Inception Distance (FID) skoru ve üretim çıktılarını görsel gerçeklik ile yapısal bütünlüğü değerlendirmek için Structural Similarity Index Measure (SSIM) gibi nicel metrikler kullanılarak analiz edilmektedir. Ayrıca, veri seti ölçeklerinin (1:2000, 1:3000 ve 1:5000) model performansı üzerindeki etkisi araştırılmaktadır.

Bu araştırma, Generative Adversarial Networks'ün afet sonrası kentsel tasarımda bir araç olma potansiyelini ortaya koymakta ve veri seti ölçeğinin gerçekçi kentsel desenler üretmedeki başarısı üzerindeki etkisine dair önemli içgörüler sunmaktadır.

Anahtar kelimeler: Üretken Tasarım, Yapay Zekâ, Üretken Çekişmeli Ağlar, Kentsel Doku Kaybı, Kentsel Tasarım

Acknowledgements

I would like to express my sincere gratitude to TÜBİTAK for supporting my studies through the 2210C Yurt İçi Öncelikli Alanlar Yüksek Lisans Burs Programı and 1002A Hızlı Destek Modülü. I am profoundly grateful to my advisor, Asst. Prof. Sinan Akyüz, for his invaluable guidance, patience, and insights throughout this research journey. His support was instrumental in shaping the direction of my work.

Finally, I wish to express my deepest appreciation to my family for their endless support, encouragement, and understanding. Their presence has been a source of strength, and I am deeply thankful for their belief in me.

TABLE OF CONTENTS

1. INTRODUCTION	1
1.1 RESEARCH BACKGROUND & PROBLEM STATEMENT	1
1.2 RESEARCH AIMS & OBJECTIVES	5
1.3 SCOPE & LIMITATIONS	6
1.4 RESEARCH METHODOLOGY	7
2. GENERATIVE BACKUP OF URBAN MEMORY	14
2.1 URBAN MEMORY	14
2.2 GENERATIVE APPROACHES IN URBAN STUDIES	16
2.3 BEYOND DIGITAL ARCHIVES	19
3. FUNDAMENTALS OF MACHINE LEARNING	21
3.1 KEY CONCEPTS IN MACHINE LEARNING	22
3.1.1 <i>Types of Learning Approaches</i>	23
3.1.2 <i>The Role of Data and Selection of Model Type</i>	24
3.1.3 <i>How Models Learn: The Training Process</i>	25
3.2 DEEP LEARNING AND NEURAL NETWORKS	28
3.2.1 <i>Neural Networks: The Basics</i>	28
3.2.2 <i>Deep Learning: The Power of Layered Learning</i>	31
3.2.3 <i>Types of Neural Networks in Deep Learning</i>	35
4. GENERATIVE ADVERSARIAL NETWORKS	37
4.1 GENERATIVE ALGORITHMS	37
4.2 ADVERSARIAL LEARNING: THE ARCHITECTURE OF GENERATIVE ADVERSARIAL NETWORKS	39
4.3 COMMON CHALLENGES IN TRAINING	40
4.4 GENERATIVE ADVERSARIAL NETWORKS IN URBAN STUDIES	42
5. URBAN FABRIC GENERATION: HATAY CITY CENTER	46
5.1 CONTEXT & HISTORICAL BACKGROUND	46
5.2 DATA PREPARATION	47
5.3 GENERATIVE ADVERSARIAL NETWORKS EXPERIMENTS	47
5.3.1 <i>Trial Experiment</i>	48
5.3.2 <i>Experiments of Scale 1:5000</i>	49
5.3.2.1 <i>Experiment 1</i>	49
5.3.2.2 <i>Experiment 2</i>	51
5.3.2.3 <i>Experiment 3</i>	54
5.3.2.4 <i>Experiment 4</i>	56
5.3.2.5 <i>Experiment 5</i>	58
5.3.2.6 <i>Experiment 6</i>	61
5.3.2.7 <i>Experiment 7</i>	65
5.3.3 <i>Experiments of Scale 1:3000</i>	69
5.3.3.1 <i>Experiment 1</i>	69
5.3.4 <i>Experiments of Scale 1:2000</i>	73
5.3.4.1 <i>Experiment 1</i>	73
5.4 EVALUATION OF THE MODEL AND GENERATED IMAGES	76
6. CONCLUSIONS AND FUTURE PROSPECTS	83

6.1 CONCLUSIONS 83

6.2 SOCIETAL IMPACT AND CONTRIBUTION TO GLOBAL SUSTAINABILITY 85

6.3 FUTURE PROSPECTS 86



LIST OF FIGURES

Figure 1.1 Strengths of Models (based on Gainetdinov, 2023)	3
Figure 1.2 Input comparison, Top - Boim et al. (Jisr Az-Zarqa), Bottom - Research Area (Hatay)	4
Figure 1.3 From left to right: pre-earthquake map, aerial photograph, input map	7
Figure 1.4 25 km ² area used for dataset creation (Created by the author).....	9
Figure 1.5 Example visuals from Dataset 1, Dataset 2, Dataset 3 (Created by the author)	11
Figure 1.6 Flowchart of the experiment workflow (Created by the author).....	12
Figure 3.1 A perceptron node (created by the author)	29
Figure 3.2 Structure of a node (created by the author)	29
Figure 3.3 Representation of typical layers of a neural network (created by the author).....	30
Figure 3.4 Iterative learning process (created by the author)	32
Figure 3.5 A Gradient for two weights (x, y axes) (Based on intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee)	33
Figure 3.6 Difference between an appropriate learning rate (left) and a high learning rate (right) (Based on intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee).....	34
Figure 3.7 More realistic gradient landscape (From intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee)	34
Figure 4.1 Generative Adversarial Networks architecture	40
Figure 4.2 Mode collapse, the doctoral dissertation of Can UZUN (2020).....	41
Figure 4.3 Underfitting and overfitting in a regression model (What Are Overfitting and Underfitting, Dooinn KIm, Medium)	41
Figure 5.1 Effects of different values of scale, threshold, and erosion on the pattern of the mask	50
Figure 5.2 Applying different parameters to understand which one would be more suitable for training	50
Figure 5.3 Change of Generator loss and Discriminator loss throughout the epochs	51
Figure 5.4 Change of Fréchet Inception Distance Score throughout the epochs.....	52
Figure 5.5 Generated image during training epoch 241, a sign of overfitting.....	52
Figure 5.6 Example of the test result at 245 epochs	52

Figure 5.7 Comparison of the test result at 75, 100, 150, and 245 epochs	53
Figure 5.8 Change of Generator loss and Discriminator loss throughout the epochs during experiment 3	54
Figure 5.9 Change of Fréchet Inception Distance Score throughout the epochs during experiment 3	55
Figure 5.10 An example of the test results at epoch 100 during experiment 3	55
Figure 5.11 Generated image at epoch 100 during training	55
Figure 5.12 Change of Generator and Discriminator loss throughout the epochs during experiment 4	56
Figure 5.13 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 4.....	57
Figure 5.14 Examples of the test results at epoch 245 during Experiment 4	57
Figure 5.15 Generated image at epoch 239 during experiment 4.....	58
Figure 5.16 Change of Generator and Discriminator loss throughout the epochs during the experiment 5.....	59
Figure 5.17 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 5.....	59
Figure 5.18 The test results of a 1:5000 scale input map at epoch 45 during experiment 5	60
Figure 5.19 Difference of generation success between labelmap-like input and post-earthquake map input.....	61
Figure 5.20 Steps of creating the new label map	62
Figure 5.21 Outputs throughout the experiments	63
Figure 5.22 Steps of creating the magenta label map	66
Figure 5.23 Change of Generator and Discriminator loss throughout the epochs during the experiment 7.....	66
Figure 5.24 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 7.....	67
Figure 5.25 The test results of a 1:5000 scale input map at epoch 45 during experiment 7	67
Figure 5.26 The test results of a 1:3000 scale input map at epoch 45 during experiment 7	68
Figure 5.27 Change of Generator and Discriminator loss throughout the epochs during experiment 1	69

Figure 5.28 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 1.....	70
Figure 5.29 The test results of a 1:3000 scale input map at epoch 50 during experiment 1	70
Figure 5.30 The test results of a 1:5000 scale input map at epoch 50 during experiment 1	71
Figure 5.31 The test results of an alternative 1:3000 scale input map at epoch 50 during experiment 1	72
Figure 5.32 Change of Generator and Discriminator loss throughout the epochs during experiment 1	73
Figure 5.33 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 1.....	74
Figure 5.34 The test results of a 1:2000 scale input map at epoch 45 during experiment 1	74
Figure 5.35 The test results of a 1:5000 scale input map at epoch 45 during experiment 1	75
Figure 5.36 The test results of a 1:3000 scale input map at epoch 45 during experiment 1	75
Figure 5.37 Types of label map generation methods.....	76
Figure 5.38 Comparison of Fréchet Inception Distance Scores of different experiments.	77
Figure 5.39 Comparison of Structural Similarity Index Measures across input scales of different models.	78
Figure 5.40 SSIM scores of the generated images of different scale inputs- model 1:2000	79
Figure 5.41 SSIM scores of the generated images of different scale inputs- model 1:3000	79
Figure 5.42 SSIM scores of the generated images of different scale inputs- model 1:5000	80
Figure 5.43 Comparison of the generated images with different models	81

LIST OF TABLES

Table 1.1 Research Structure	6
Table 2.1 Eras in Computational Urban Studies.....	18
Table 3.1 Machine Learning Models	25
Table 3.2 Common Evaluation Metrics	27
Table 3.3 Output Layer Activation Functions	31
Table 3.4 Types of Neural Networks.....	35
Table 4.1 GANs in Urban Studies	44
Table 5.1 Description of the Parameters Used in the Experiments	48
Table 5.2 SSIM Scores of The Generated Images.....	81



LIST OF ABBREVIATIONS

AI	Artificial Intelligence
FID	Fréchet Inception Distance
GANs	Generative Adversarial Networks
GIS	Geographic Information System
ML	Machine Learning
NNs	Neural Networks
OSM	OpenStreetMap
QGIS	Quantum Geographic Information System
RGB	Red Green Blue
SSIM	Structural Similarity Index



*To the memory, strength, and hope of the
people affected by the Türkiye-Syria
earthquakes*

Chapter 1

Introduction

Urban environments are dynamic entities shaped by layers of historical, cultural, and social evolution. At the heart of this complexity lies the urban fabric, which not only defines the physical form of a city but also carries the traces of collective memory and identity. However, urban fabrics are not immune to disruption, and when significant loss occurs, it challenges both the continuity and future development of urban life.

Urban fabric plays a vital role in shaping the functionality and sustainability of an urban area while nurturing a unique identity. It envelops not only the physical structure of the area but also its historical, cultural, social, and temporal dynamics. However, urban fabrics are vulnerable to significant disruptions caused by natural disasters, urban warfare, and rapid urban developments. These events might result in mass loss of fabric and leave gaps in the spatial and functional cohesion of the city.

Unfortunately, Hatay's city center is a poignant example of a mass loss of urban fabric. As a historically rich urban area shaped by centuries of diverse influences, Hatay has faced considerable fabric loss due to the 2023 Türkiye-Syria Earthquakes. This devastating loss not only highlights the need for a contemplative rebuilding process but also opens the door for exploring innovative approaches to reimagining urban spaces. At the intersection of these challenges and opportunities, this research investigates the potential of Generative Adversarial Networks (GANs) as a tool in urban design processes.

1.1 Research Background & Problem Statement

Urban space plays a crucial role in forming and preserving urban memory. The physical elements of the city, often referred to as the urban fabric, serve as tangible anchors for collective memory, shaping how people experience and recall the city over time. The term “Urban fabric” refers to the physical characteristics and spatial

organization of an urban area. It includes the arrangement of buildings, streets, and public spaces and their interconnections with one another. However, it is defined not only by the physical layout of the urban area but also by its historical, cultural, and social dimensions, which shape its unique identity and functionality (Lynch, 1981).

Over time, the concept of the urban fabric has evolved beyond its original definition. While the initial definition mainly emphasized the physical dimensions of cities, during the mid-20th century, social and cultural elements began to be incorporated into the term (Levy, 1999). There are also new perspectives that categorize urban fabric based on transport systems and other factors, highlighting the multifaceted influences of urban fabric on urban life (Newman et al., 2016).

A large-scale loss of urban fabric often results in the erosion of urban memory, bringing with it profound social, cultural, and environmental consequences. This disruption can sever connections to collective identity, diminish cultural continuity, and negatively impact the sustainability and livability of urban spaces. Socially, it disrupts communities and diminishes the interconnections that come with shared spaces. Culturally, the destruction of historical or traditional urban fabrics leads to the loss of architectural heritage and unique identities (Ulu & Karakoç, 2004). From an environmental point of view, large-scale demolition and redevelopment projects often result in unsustainable practices and increased carbon emissions (X. Li et al., 2016).

Ultimately, the loss of urban fabric affects the physical layout of cities and erodes the cultural and social frameworks that build urban life. A large-scale loss can be a result of natural disasters such as floods, earthquakes, tsunamis, or storms, and it can also be the result of human-induced causes such as wars or urban development projects. Because of the multi-layered and faceted nature of the urban fabric and urban memory, the rebuilding process after the loss is critically important.

When addressing this type of complex design process, the design tools employed by architects and urban designers are just as important as the approaches they adopt. In the realm of architecture, there is a parallel relationship between the evolution of our tools and the way we conceptualize the discipline. Design tools serve as a vehicle for both thinking and making architecture (García del Castillo y López, 2022).

The evolution of design tools is closely tied to the advent of computers and the continuous progress in computing technology. As these advancements shape the tools we use, the tools, in return, influence the way we conceptualize architecture. Consequently, technological progress indirectly affects architectural thinking. This connection makes

the developments in artificial intelligence and machine learning very significant for the discipline itself.

Architects create a wide array of drawings, sketches, models, and renderings over the course of design processes and construction phases, forming what is essentially image-based data. Therefore, progress in machine learning techniques applied to this type of visual data holds considerable importance, both for the broader field of machine learning and for architecture.

A major breakthrough in processing image-based data occurred in 2014 with the introduction of Generative Adversarial Networks (GANs) by Goodfellow and colleagues. GANs involve two neural networks working in opposition: a generator and a discriminator. The generator learns from the dataset to create new data, while the discriminator determines whether the data it encounters is from the original dataset or generated. This adversarial nature makes GANs very capable of handling complex datasets and generating new data from them (I. J. Goodfellow et al., 2014).

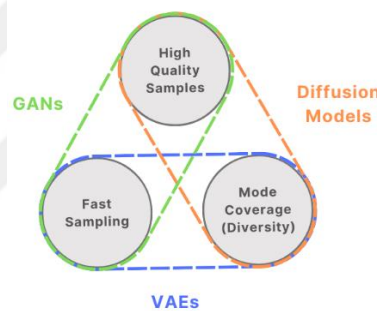


Figure 1.1 Strengths of Models (based on Gainetdinov, 2023)

In addition to this, Generative Adversarial Networks (GANs) possess certain advantages over other deep learning-based generative models. For instance, Variational Autoencoders (VAEs) often produce images that appear blurry due to pixel-based loss, whereas GANs typically generate much sharper, more detailed outputs. Although other methods, such as Diffusion Models, can produce high-quality results, their relatively slow data generation makes them less practical in many applications. In this respect, GANs excel by providing both speed and efficiency (Gainetdinov, 2023).

These advantages have led to the use of GANs in various areas of the architectural discipline. For example, in his study, Stanislas Chaillou (2019) focused on the building scale and employed Generative Adversarial Networks to generate housing floor plans, examining the functionality of GANs in terms of functional zoning and furniture layout.

In Additionally, he proposed parameters for classifying plan layouts produced through the model and categorized the layouts according to parameters.

Another study at the same scale, conducted by Uzun (2020), addressed the generation of Palladian plan schemas. Uzun documented the GAN model's training process and evaluated the newly generated Palladian plan layouts created by the model. In another study published in 2020, a GAN architecture different from the one used in this research was implemented to produce land-use plans. However, that study incorporated predefined parameters deemed important for design during the generation process (D. Wang et al., 2020a).

In a project titled "This Map Does Not Exist", which was conducted at a scale similar to this study, a GANs model was used to generate urban maps. However, this project does not take any inputs, and the resulting maps are randomly generated without being tied to any real-world location (Boates, 2020). Another study at a similar scale explores urban road designs (Chu et al., 2019), yet it employs Neural Turtle Graphics (NTG) rather than a GAN-based method.

In a different study, a GAN model was trained on urban maps to produce block designs (Fedorova, 2021). In this study, researchers used maps from multiple cities, yielding outputs in several distinct styles. Although its approach was similar to this research, the scope was narrower, and only block-level designs were generated.

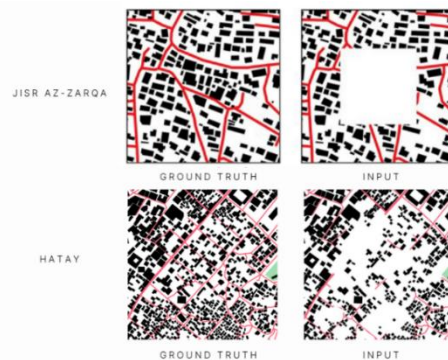


Figure 1.2 Input comparison, Top - Boim et al. (Jisr Az-Zarqa), Bottom - Research Area (Hatay)

The work that most closely aligns with this research was conducted by Boim et al. (2022) on unplanned settlements. Their study involved training a GAN on maps of the Jisr az-Zarqa settlement and removing square-shaped urban fabric segments so the GAN could generate the missing areas (See Figure 1.2). This generation served as a test of how effectively the model had learned the unplanned urban environment. The authors

evaluated the outcomes by examining the GAN-generated segments against the ground truth.

Given the challenges outlined in the problem statement, exploring innovative design approaches to address these issues becomes essential. In response to this context, the next section presents the research aim and objectives, which guide the research into how GANs can contribute to memory-preserving urban regeneration strategies.

1.2 Research Aims & Objectives

This research aims to explore the potential of Generative Adversarial Networks (GANs) in the field of urban design, focusing on their application in generating urban fabric. By integrating GANs into the urban design process, the study seeks to bridge the gap between traditional urban design practices and contemporary AI technologies. Beyond exploring this potential, another core aim is to make the concept of GANs more accessible to architects and urban designers who may not be familiar with machine learning. In doing so, the research aspires to initiate meaningful discussions among professionals, encouraging them to consider the opportunities and implications of these technologies in their design practices.

To achieve these aims, the study has established the following objectives:

- **To explain the relevant literature to professionals unfamiliar with GANs** in an easy-to-understand manner, breaking down technical concepts and presenting them within the context of urban and architectural design.
- **To generate new urban fabrics derived from the datasets** as a means of exploring and evaluating the capabilities of GANs in urban design applications.
- **To assess the efficiency and competence of the Pix2PixHD architecture** in creating realistic urban patterns and contributing to urban design tasks.
- **To analyze the impact of dataset scales** (1:2000, 1:3000, and 1:5000) on GAN training success and the quality of generated urban fabric, providing insights into the relationship between data granularity and model performance.

These aims and objectives shape the overall research structure, anticipating that it provides theoretical contributions to understanding GANs in urban design and practical insights into their application.

Table 1.1 Research Structure

Research Questions	Research Objectives	Research Assumptions
How can generative design tools contribute to the preservation and regeneration of urban memory in post-disaster urban contexts?	To investigate the role of generative approaches in maintaining spatial continuity and supporting the regeneration of urban memory after large-scale urban loss.	Generative models, when trained on a city's spatial characteristics, can support memory-preserving design decisions by synthesizing spatial patterns rooted in local urban identity.
How can Generative Adversarial Networks (GANs) be utilized as a tool in urban design to generate lost urban fabric?	To explore the potential of GANs in reconstructing urban fabric for urban design applications	GANs can generate urban design elements through training on urban maps, providing insights into design potential for urban planners and architects.
How effective is the Pix2PixHD architecture in generating urban patterns?	To assess the efficiency and competence of the Pix2PixHD architecture in urban design tasks.	The Pix2PixHD GAN architecture, when trained with comprehensive datasets, can produce contextually accurate representations of urban fabrics
How do different scales (1:2000, 1:3000, and 1:5000) in training data affect the model's learning outcomes?	To assess the impact of dataset scale on GAN training success and output quality in urban fabric generation.	Variations in map scales in training datasets influence the model's capacity to capture details and patterns effectively.

This research is structured around four core questions that investigate how generative design tools—particularly Generative Adversarial Networks (GANs)—can support the preservation and regeneration of urban memory in post-disaster contexts. The study aims to explore the potential of GANs in reconstructing lost urban fabric, assess the effectiveness of the Pix2PixHD architecture in urban design tasks, and examine how different training dataset scales (1:2000, 1:3000, 1:5000) influence model performance. Underlying these objectives is the assumption that generative models trained on the spatial characteristics of a city can produce contextually appropriate outputs that help inform memory-sensitive design decisions (see Table 1.1).

1.3 Scope & Limitations

This research explores the application of Generative Adversarial Networks (GANs) within the context of urban design, specifically focusing on the generation of two-dimensional urban fabric. The datasets used for training are restricted to two-dimensional representations comprising buildings, roads, rivers, streams, and shared green spaces. The algorithm's outputs are confined to the same scope, with no incorporation of three-dimensional data or visualizations.

The datasets for this study were prepared exclusively from the urban center of Hatay, deliberately excluding rural areas. The training data reflects the city's pre-earthquake state, while the input data for generation is based on post-earthquake conditions. Additionally, industrial zones were excluded from the dataset due to their lack of city-specific morphology and minimal variation across different cities.



Figure 1.3 From left to right: pre-earthquake map, aerial photograph, input map

An Urban fabric consists of multiple layers, including historical, cultural, and economic elements. The datasets in this research, however, focus only on physical features, leaving these broader aspects unrepresented in the generated outputs (see Figure 1.3).

The training data was obtained from OpenStreetMap due to being open-source. While OpenStreetMap data is known to have some inaccuracies, it was deemed sufficient for representing the general urban fabric, and these inaccuracies were overlooked.

The model was trained exclusively on data from Hatay. As a result, its applicability is limited to the specific context of Hatay and cannot be directly used for urban design in other cities without further training or adjustments.

Having defined the scope and acknowledged the limitations of the study, the next section outlines the research methodology adopted to address the identified objectives

1.4 Research Methodology

In this study, a hybrid research design was employed to examine the capacity of Generative Adversarial Networks (GANs) to generate urban maps and the influence of varying dataset scales on this capacity. A *quasi-experimental case study* design was adopted for this purpose.

A case study is a research method involving an empirical investigation into a phenomenon in its real-life context when the distinction between the phenomenon and its context is unclear (Yin, 2002). Within the scope of this research, it can be said that the GANs model itself is the phenomenon, while the task of generating urban maps is the context. The evaluation of the GANs' outputs is the empirical investigation between the phenomenon and the context.

The experimental method is a research design that focuses on examining how changes to one or more independent variables influence a dependent variable. In the context of this research, the independent variables are the datasets utilized and GANs architecture, while the dependent variable is the collection of "urban maps" generated from these datasets. A true experimental setup would include a control group in order to measure the effect of the independent variable, and experimental outcomes would be measured against this control group. However, in this research, there is no control group, and also the independent variables (the datasets) are predetermined rather than randomly assigned to the groups (Rogers & Revesz, 2019). Hence, it would not be accurate to label the approach as a "true experimental" method. Instead, the comparison in this study is made by comparing the outcomes of the different experiments with one another. As a result, the methodology can be characterized as "*quasi-experimental*."

The results obtained from the training experiments are evaluated using analytical generalization rather than statistical generalization, as the research is conducted on a limited sample. Therefore, the generalization of findings is based not on numerical statistics but on their alignment with the theoretical framework established in the existing literature.

While statistical generalization aims to apply findings from large sample groups to a broader population, analytical generalization allows results to be interpreted within a wider theoretical or conceptual framework (Yin, 2002). In this study, the findings obtained from evaluating the capacity of Generative Adversarial Networks (GANs) to generate urban maps using datasets of different scales are analyzed in a way that contributes to the broader theoretical discourse on GANs' sensitivity to data scale and their learning capabilities. Therefore, this study aims to interpret results within a conceptual context by drawing on similar research in the literature. This approach enables the findings to be discussed within a broader framework, contributing to the overall generalizability of the research.

This study examines Generative Adversarial Networks (GANs) as a phenomenon within the context of urban map completion, aiming to explore the relationship between GANs and urban map completion. In this respect, the thesis has been designed using a case study approach. Additionally, since it investigates the interactions between the independent variables (datasets) and the dependent variable (generated outputs), the study incorporates elements of an experimental design. However, due to the absence of a control group and the lack of a predetermined number of experiments, a fully “true experimental” method was not applied, and a “quasi-experimental” approach was adopted instead. As a result, this thesis adopts a hybrid methodology that integrates elements of both quasi-experimental and case study approaches. This hybrid methodology, combining quasi-experimental and case study approaches, has also been successfully employed in previous research. Can UZUN's doctoral dissertation (2020) on generating Palladian plan schemes highlights its applicability to studies requiring the integration of experimental and contextual analyses.

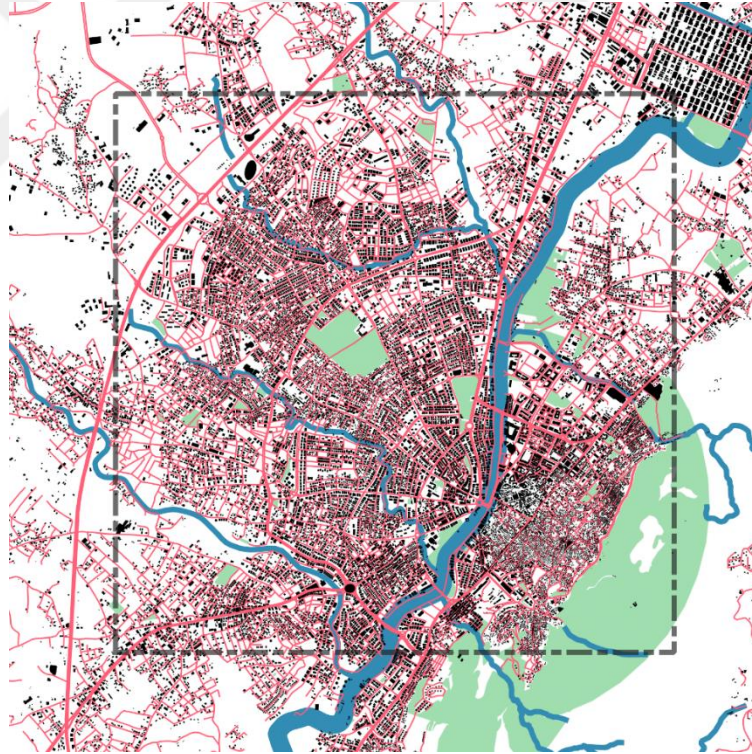


Figure 1.4 25 km² area used for dataset creation (Created by the author)

Dataset Preparation: In the initial phase of the research, the focus was on creating the datasets required for training the Generative Adversarial Network (GAN) model. Accordingly, three different datasets were prepared at scales of 1:2000, 1:3000, and 1:5000 (See Figure 1.5). QGIS was used to create these datasets, as it is open-source

software and is frequently preferred in the literature. While generating the general map of Hatay, OpenStreetMap data was imported into QGIS using the QuickOSM plugin via the “Quickquery” tool. The “key:value” pairs used in the imported data are detailed below:

- For **roads**, a single key with five associated values was used:
highway: trunk, primary, secondary, tertiary, residential
- For **green areas**, three different key values were defined, each with multiple associated values were used:
landuse: recreation_ground, cemetery, meadow, farmland, forest
leisure: garden, pitch, park
natural: wood, scrub, forest
- For **buildings**, two keys with associated values were used:
building: yes
destroyed:building: yes
- For **rivers and streams**, a single key with three associated values was used:
waterway: river, stream, ditch

The dataset was designed to represent structures using a "solid-empty" style (Nolli Map). This approach is expected to enhance the training success (Boim et al., 2022). In this context, buildings were represented in black (RGB 0 0 0), roads in red (RGB 255 95 120), green areas in green (RGB 130 210 150), and rivers and streams in blue (RGB 50 140 180), with a white background (RGB 255 255 255).

A 5 km x 5 km area, centered on a region with high building density in the city center, was selected for dataset creation (see Figure 1.4). Within this area, a 100-meter grid was generated. Then, square frames measuring 300, 450, and 750 meters in length were defined. These frames were systematically shifted along the X and Y axes across the grid using a Python script, producing three different datasets. The datasets were reviewed image by image, and any images deemed unsuitable for training were excluded from the final datasets.

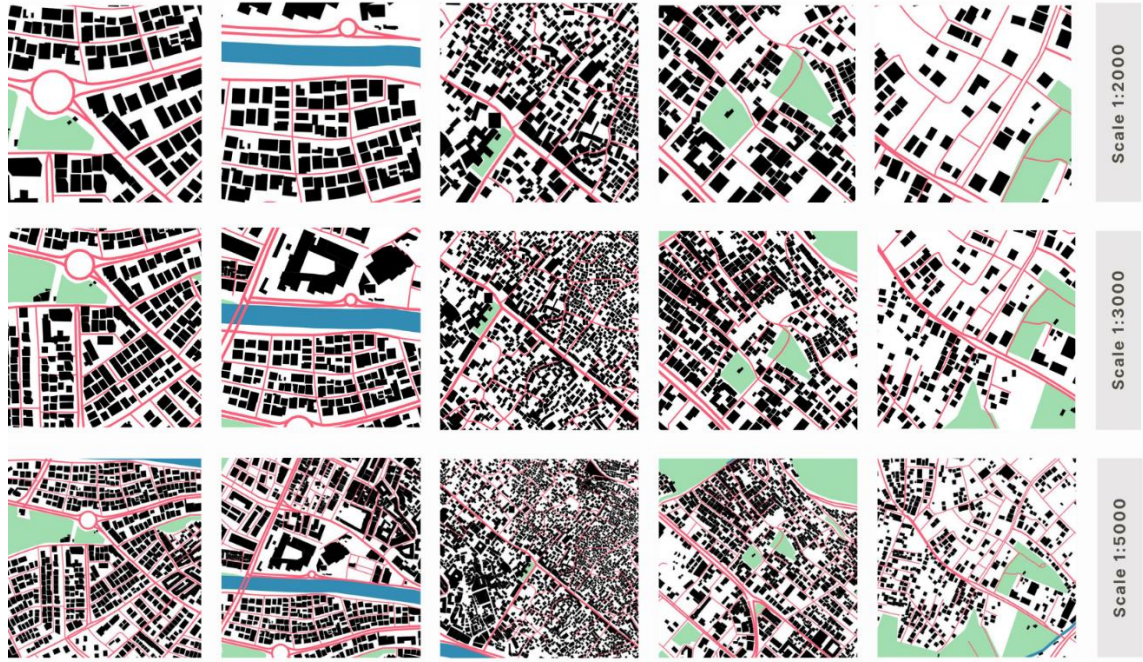


Figure 1.5 Example visuals from Dataset 1, Dataset 2, Dataset 3 (Created by the author)

Experiment Design: Every experiment starts with a preliminary "trial" phase to confirm that the algorithm functions properly. If the trial phase runs without any complications, the process advances to the training stage. During the training process, metrics like FID scores and loss values are continuously monitored and recorded. An experiment concludes for a specific dataset when the model successfully generates satisfactory outputs.

Analysis and Evaluation: The next stage after training is the analysis phase, which involves evaluating both the algorithm and the outputs it generates. For algorithm analysis, the Fréchet Inception Distance (FID), a widely recognized metric in generative model evaluation, will be utilized. The FID score effectively measures the quality and diversity of generated images by comparing their features to those of real images, with lower scores indicating a closer resemblance (Heusel et al., 2018).

For visual evaluation, the model's performance will be evaluated based on its ability to extend existing road networks, propose new connections, accurately identify the topological features of streets, and replicate building morphologies with similar density. These generated outputs will be compared to the original urban fabric (ground truth), and their similarity will be measured using the Structural Similarity Index Measure (SSIM).

SSIM is a commonly used metric that assesses the similarity between two images by analyzing their brightness, contrast, and structural details. It provides a score between 0 and 1, where a value closer to 1 indicates a high degree of similarity, while a value near

0 suggests minimal resemblance (Z. Wang et al., 2004). Brightness measures the overall light intensity in the image, contrast evaluates the differences in light intensity across regions, and structural information captures the shapes, edges, textures, and spatial organization of objects. By comparing these aspects, SSIM evaluates how well the structural patterns and features of the images align, offering insights into the preservation of visually significant details. In applications such as image generation and compression, SSIM is widely recognized as an effective tool for gauging the structural fidelity of images (Z. Wang et al., 2004).

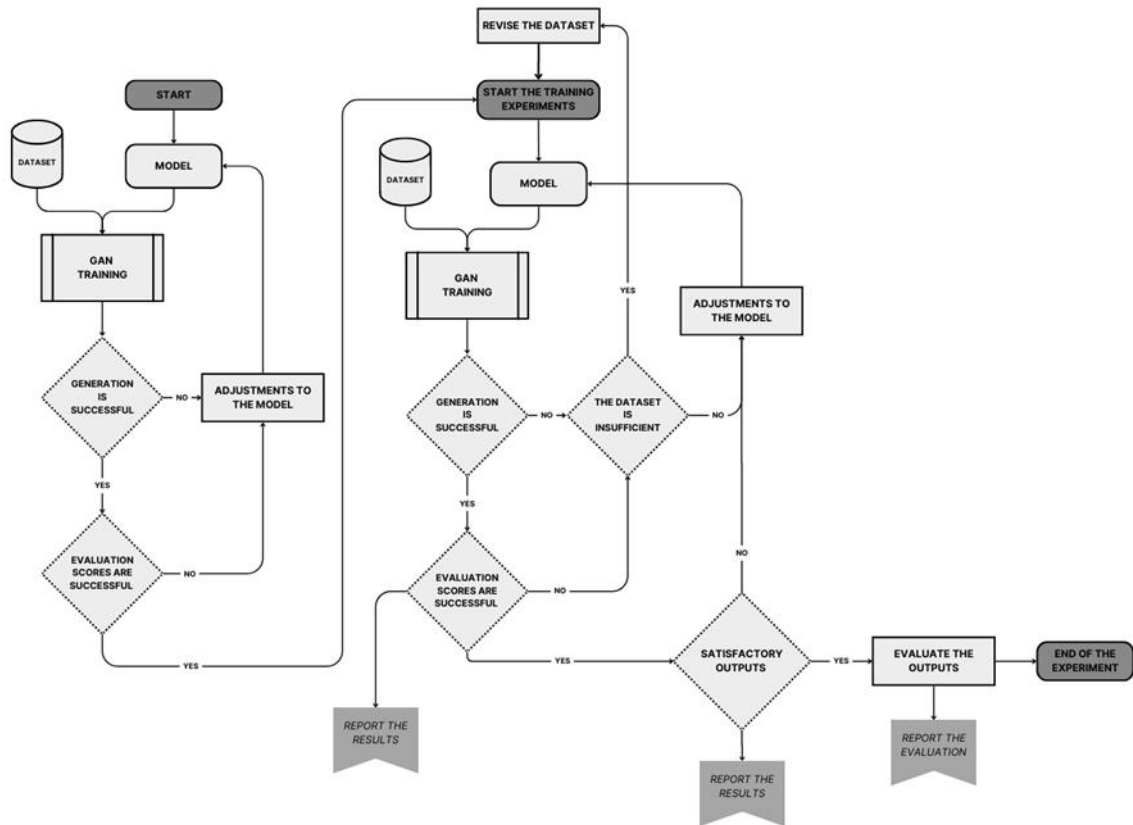


Figure 1.6 Flowchart of the experiment workflow (Created by the author)

Experiment Workflow: The experiments proceed according to the workflow outlined below and depicted in the flowchart (see Figure 1.6):

1. **Trial Phase:** For each dataset, experiments begin with a trial training phase to ensure the algorithm functions smoothly. If no issues are observed during this stage, the main training experiments proceed. If the trial phase fails, necessary adjustments are made to the model, and the process is repeated.
2. **Determining Training Parameters:** Before training begins, values for parameters such as the learning rate, number of residual blocks, and batch size are determined.

3. **Model Training:** The model is trained based on the defined parameters. During training, intermediate outputs and performance metrics are periodically reviewed to monitor progress.
4. **Loss Function and FID Measurement:** Loss function values and Fréchet Inception Distance (FID) scores are continuously tracked throughout the training process. These metrics evaluate the model's learning efficiency and the quality of the generated outputs. If the loss values or FID scores exhibit excessive oscillation or inconsistency, the dataset is revisited and potentially revised, requiring a return to Step 2.
5. **Saving Generated Images:** The output images produced at the conclusion of the training are saved for further analysis and evaluation. These outputs serve as the foundation for assessing the model's performance.
6. **Evaluation of Generated Outputs:** The quality of the generated images is reviewed. If they fail to meet the desired standards, adjustments are made, and the process loops back to the parameter determination phase. If the visuals meet expectations, the workflow progresses.
7. **SSIM Analysis:** For the outputs deemed satisfactory, the Structural Similarity Index Measure (SSIM) is calculated to evaluate the structural integrity and visual realism of the outputs.
8. **Finalizing the Experiment:** Once the desired outcomes are achieved for a particular dataset or scale, the experiment is finalized. The same methodology is then applied to other datasets or scales.

Chapter 2

Generative Backup of Urban Memory

Urban memory plays a crucial role in shaping the identity and continuity of cities. It emerges through the accumulation of spatial, cultural, and social layers embedded in the urban fabric. However, large-scale disruptions such as natural disasters, wars, and rapid urban transformations can severely damage or erase these accumulated layers, leading to a profound loss of memory. Traditional preservation methods often focus on documentation and conservation, but these approaches typically treat urban memory as a static archive rather than a living, evolving phenomenon. With advancements in computational tools and generative models, new possibilities have emerged for preserving and regenerating urban memory. These models can learn and synthesize spatial patterns, offering opportunities to support memory continuity even in disrupted urban environments. This chapter explores the conceptual foundations of urban memory, discusses generative approaches in urban studies, and examines how emerging technologies can contribute to preserving the dynamic relationship between space and memory.

2.1 Urban Memory

The Letter of Lord Chandos (Ein Brief), an essay written in the form of a letter by **Hofmannsthal**(1902), presents a fictional correspondence dated August 1603 from Lord Philip Chandos to Francis Bacon. In it, Chandos discusses his struggles with a language crisis. And He describes how “*the power of our mysterious ancestors accumulated in us*” led to “*the stacking of accumulated collective memory*”, a statement that marks the earliest introduction of the **concept of collective memory** (Olick & Robbins, 1998).

However, after this initial introduction of the concept, Hofmannsthal did not develop the term in a sociological sense. Instead, **Halbwachs** (1925), in “*On Collective Memory*”, formalized the concept and expanded the study of memory beyond the field of psychology into sociology and the social sciences (Z. Liao & Dai, 2020). After conceptualizing memory in a social framework, Halbwachs (1950) examined the

relationships between collective memory and space. In his posthumously published work, it is stated that changes to the physical environment can significantly influence the identity and continuity of the people inhabiting that space.

Even though Halbwachs explored the relationship between space and collective memory, the development of a structured framework within urban studies was shaped by Rossi (1982) in *The Architecture of the City*. Rossi defined the city as a **repository of collective memory** and introduced the concept of **urban artifacts**, which include buildings, streets, public spaces, and so on. These artifacts are recognized not only as physical elements of the city but also as **memory carriers** that shape and preserve its unique identity. During the city's transformation, the continuity of typologies of these artifacts is emphasized as a key factor in preserving the **collective urban memory**.

Thus, **urban memory** can be understood as a form of collective memory, both shaped by the urban environment and daily life while also simultaneously shaping the city and its inhabitants.

In contemporary studies, the understanding of urban memory has expanded beyond the physical form of space to include the abstract dimensions of the city. Alongside this shift, the definition of the city's physical form has also evolved, incorporating new dimensions such as walkability, green infrastructure, and public transportation accessibility (Newman et al., 2016). As a result, both tangible and intangible aspects are considered to play an important role in shaping urban memory and reflecting a more complex and layered perception of the city.

This broader perspective in contemporary literature positions urban memory as a more comprehensive understanding of the city's spatial organization and transformation, acknowledging the multiple layers that shape urban identity. Rather than being solely a repository of memory, urban memory now represents a collective perception, intertwining shared experiences, cultural narratives, and evolving social dynamics, all of which are embedded in and shaped by urban space.

Although urban memory is defined in various ways across the literature, its core elements can be broadly categorized into two main classes. The first category of elements is **subjects**, which refers to individuals or communities that perceive and experience memory. The second category is **objects**, which includes the urban elements that serve as the content of memory. The objects category is further divided into two subgroups (Z. Huang & Huang, 2024):

Material Elements: These are tangible, physically existing components that can be directly observed and spatially perceived. This group includes Spatial forms, Physical typologies, Natural environment, Built environment, and so on.

Immaterial Elements: These do not have a physical form but are embedded in collective memory and constitute a significant part of urban memory. This group includes: Temporal elements, Symbols, Names, and so on.

Urban memory is the relationship between the subject and the object - a dialogue between the one who observes and the one being observed - within the framework of urban space. Therefore, the preservation of urban memory is heavily dependent on the continuity of urban space. And this continuity is always at risk of disruption, whether due to conflicts, natural disasters, or large-scale urban transformation projects. The loss of a significant portion of urban space creates a profound gap in urban memory, leaving lasting scars on the city's inhabitants. Urban space is not merely a physical form; it is the stage where urban life unfolds. When space is lost, a part of urban life disappears with it, and in turn, a part of human life is also diminished.

2.2 Generative Approaches in Urban Studies

Technological advancements have always been an influential factor in shaping generative approaches in urban studies. As computational tools have become more accessible, new methods have gradually been integrated into mainstream practice, and this influence continues to evolve the discipline today.

Before the digital era, urban studies were largely based on conceptual models aimed at defining the ideal city. These models were influenced by various urban theories. Some well-known examples include Howard's (1898) **Garden City** concept and Le Corbusier's modernist urban planning principles.

Alongside these idealized models, there were other frameworks aimed at explaining the underlying patterns of urban growth and social structure. Notable examples include Burgess's (1925) **Concentric Zone Model**, which analyzed urban expansion and social organization, and Christaller's (1933) **Central Place Theory**, a mathematical model that explained the spatial distribution of settlements.

With advancements in computing technology and digitalization, early computational models were built upon the foundations of previous urban theories. These

new developments led to the emergence of simple rule-based models to address the same questions as before like how does a city expand.

Lowry (1964) introduced urban growth models, which aimed to **predict patterns of city expansion**. Tobler (1979) applied cellular automata, a grid-based computational model, to **simulate urbanization patterns**. Around the same time, a **Geographic Information System (GIS)** was introduced by Tomlinson (1974). Later, Hillier and Hanson (1984) developed **Space Syntax**, a method for analyzing spatial relationships. Overall, in the early stages of computational models, there were predictive models, simulations, new analytical tools, and methods for spatial analysis.

A true boom of generative methods in urban design and architecture was experienced only after the widespread availability of personal computers. With this newfound computation power, more algorithmic and procedural methods found their place on the stage of urban studies. Stiny (1980) used Shape Grammars, a rule based method, for generating urban forms. Agent-based models (ABM) were introduced by Batty (1997) to simulate pedestrian movement. Additionally, GIS and parametric design tools became increasingly widespread, broadening computational approaches in urban studies.

With advancements in parallel computing driven by the introduction of multi-core processors in the 2000s and the widespread adoption of cloud computing in the 2010s, computational capabilities have significantly expanded, enabling more efficient data processing, large-scale simulations, and real-time analytics across various fields (Richman et al., 2014).

Advancements in data generation technologies and real-time data analytics have significantly expanded the use of urban simulations and analyses. Among the key approaches emerging from this era is the concept of digital twins. Digital twins provide a virtual replica of urban environments, enabling in-depth spatial analysis and optimization processes and simulating different scenarios at an urban scale. As this approach gained momentum, data-driven urban analysis became more widely adopted, leading to the growth of data-informed urban design strategies (Mendula et al., 2022).

Furthermore, the increasing integration of digitalization into everyday life has led to the generation of vast amounts of data. This influx of data, combined with improved computational power and the widespread availability of cloud computing, has laid a foundation for the rise of advanced learning algorithms. As the volume of data has grown, these algorithms have evolved beyond rule-based processing, shifting toward identifying

underlying patterns and extracting meaningful insights without relying on predefined rules.

While learning algorithms had roots back in the early 20th century, they experienced significant advancements in the 2010s, particularly with the rise of deep learning (Schmidhuber, 2015). This advancement transformed generative algorithms from simple, rule-based systems to sophisticated models that are capable of generating data that reflects the intricacies of real-world datasets. This shift not only revolutionized artificial intelligence but also reshaped generative urban studies, expanding the possibilities for AI-driven urban analysis and design (H. Xu et al., 2024).

Table 2.1 Eras in Computational Urban Studies (Created by the author)

Era	Key Approaches	Technological Influence	Example Applications
Pre-Digital Era (Pre-1950s)	Theoretical urban models (e.g., Garden City, Modernist Planning), handcrafted urban designs	Manual and conceptual design approaches, no computational tools	Traditional city planning (Haussmann's Paris, Howard's Garden City)
Early Computational Models (1950s-1980s)	Urban growth models, Cellular Automata (CA), early computational simulations, Space Syntax	Mainframe computing, early GIS, mathematical urban models	Urban expansion prediction models, land-use simulations
Algorithmic & Rule-Based Approaches (1990s-2000s)	Procedural Modeling, Shape Grammars, Agent-Based Models (ABM), GIS-based Urban Analysis	Rise of personal computing, parametric design, and early AI experimentation	Automated urban form generation, early smart city concepts
AI-Driven Models (2010s- Present)	Machine Learning (ML), Deep Learning (DL), Generative Adversarial Networks (GANs), Digital Twins	Big Data, IoT, AI-enhanced urban simulations, real-time data integration	Real-time urban monitoring, AI-generated designs, Digital Twins

Urban studies have evolved significantly, transitioning from the search for an ideal city form to rule-based expansion models and from geographic information systems to digital twins. This progression has led to the era of powerful generative models. Today, these models are being used in various applications, including urban simulation and data augmentation, automated design processes, preservation efforts, and urban typology studies. However, these advancements have not yet settled into a fixed framework; we are still experimenting and exploring better ways to utilize these newfound technological tools, continuously refining their role in urban studies and design.

2.3 Beyond Digital Archives

The growing complexity of urban environments demands new approaches that go beyond traditional methods of analysis and documentation. While conventional tools have enabled significant advancements in understanding cities, they often fall short in capturing the living, evolving nature of urban spaces. As cities are shaped by continuous transformations, preserving and interpreting urban memory becomes essential for sustaining their identity. In this context, emerging technologies such as generative deep learning models offer promising possibilities for bridging the gap between static documentation and dynamic urban evolution.

GIS systems and digital twins are powerful tools for urban analysis and simulation, enabling detailed mapping, modeling, and monitoring of cities (Villanueva-Merino et al., 2024). However, they primarily function as static archives rather than capturing the dynamism of urban transformation. Traditional computational tools can document these transformations, but cannot interpret the transformation itself or how the transformation accumulates into a unique urban identity.

Urban memory is more than a collection of documentation of urban snapshots; it is a living spatial relationship, movement patterns, and morphological continuity (Cankurt Semiz & Özsoy, 2024). Cities go through layered transformations, with each era contributing new spatial relationships and patterns. These layers and the ways in which they interact create the stage for urban life to unfold while continuously shaping and reshaping one another in an ongoing process.

By learning the spatial DNA of cities, generative deep learning models offer a new horizon beyond static documentation (Kempinska & Murcio, 2019). These models can analyze and synthesize urban space, enabling them to generate entire cities from scratch or integrate new additions into existing urban environments. This capability highlights their potential as a design-thinking tool and their role in preserving urban memory throughout the design process. By understanding and synthesizing spatial patterns, generative deep learning models can help maintain continuity in urban transformations.

The transformation of a city is usually a slow-paced process, allowing governance and designers time to maintain continuity and preserve urban memory. However, when a city experiences large-scale loss due to natural disasters, war, or uncontrolled redevelopment, the transformation process accelerates rapidly due to the urgent need for

reconstruction. These spatial losses can have severe consequences on the functionality of the city, the social fabric, and the psychological well-being of its inhabitants.

The transformation of a city is usually a slow-paced process, allowing governance and designers time to maintain continuity and preserve urban memory. However, when a city experiences large-scale loss due to natural disasters, war, or uncontrolled redevelopment, the transformation process accelerates rapidly due to the urgent need for reconstruction. These spatial losses can have severe consequences on the functionality of the city, the social fabric, and the psychological well-being of its inhabitants.

In fast-paced reconstruction efforts, urban memory and spatial coherence are at significant risk of being permanently lost. Integrating generative models trained on the city's spatial organization can help guide rapid design processes toward continuity and preservation. By learning and synthesizing the spatial DNA of the city, these models offer a way to mitigate absolute loss, ensuring the preservation of the city's identity.

Teaching a machine, the fundamental spatial characteristics of an entire city can lay a needed groundwork for preserving urban memory, not only through human intent but also through the capabilities of our tools. This approach does not simply archive the past of the urban form but creates a dialogue between the history of the urban space and its future possibilities. It provides a mechanism where urban memory is not only preserved but also regenerated.

Chapter 3

Fundamentals of Machine Learning

Machine learning is a field within artificial intelligence (AI) focused on training computers to identify and generalize patterns from data. While “learning” is in the name, it’s not the type of learning humans do, there’s no experience or understanding involved. Instead, machine learning algorithms work through datasets by using mathematical models that pick up on underlying patterns, which they then use to make predictions. This is what we mean by “learning” here. This form of “learning” is deeply tied to the fields of statistics and mathematics. Key concepts began taking shape as early as the 19th century. These early concepts provided the essential groundwork for modern machine learning methods (Schmidhuber, 2022).

The concept of giving machines human-like attributes has roots in the 1950s, but the idea of bridging human thought processes and computation started even earlier. In 1943, neurophysiologist Warren McCulloch and mathematician Walter Pitts published a paper proposing that neurons could be modeled in binary terms: either firing (1) or not firing (0). They represented neuron behavior using basic logical operations, “and,” “or,” and “not”, which allowed them to build a mathematical model to describe how neurons might process information (McCulloch & Pitts, 1943).

In 1947, building on this work, McCulloch and Pitts explored the theoretical mechanisms behind how we perceive audiovisual signals (Pitts & McCulloch, 1947). They developed a mathematical model to explain how the neurons might recognize consistent patterns, referred to as “universals”, in sensory inputs. While it was a theoretical framework, their ideas were foundational for later developments in artificial neural network studies by creating a bridge between how our brains process information and how computers could model those processes.

In 1950, Alan Turing asked a provocative question, "Can machines think?" to explore this, he introduced a concept he referred to as "the imitation game." This game involved three participants: A human (referred to as A), a machine (referred to as B), and

a human judge (referred to as C). Judge (C) communicates with both A and B solely through written text while being in a separate room. The judge's task is to identify which participant is human and which is the machine, while the machine's objective is to persuade the judge that it is human (Turing, 1950).

Turing's approach to this question shifted the focus from evaluating a computer's performance based solely on its computational abilities to considering how humans perceive and interpret the machine's behavior. This shift altered how we view the relationship between humans and machines (García del Castillo y López, 2022). Today, this concept is widely known as the "Turing Test" and remains a foundational idea in AI and machine learning research.

The early theoretical foundations laid by researchers such as McCulloch, Pitts, and Turing shaped the initial understanding of how machines could mimic aspects of human cognition. These pioneering ideas established the conceptual framework that allowed the field of machine learning to emerge. Building on this foundation, modern machine learning focuses on enabling computers to learn patterns from data and improve their performance over time. The following section introduces the key concepts that define contemporary machine learning, providing the necessary groundwork for understanding how these systems operate today.

3.1 Key Concepts in Machine Learning

Machine learning has become a cornerstone of modern computational systems, offering methods that allow machines to learn from data and improve their performance over time without being explicitly programmed for specific tasks. Understanding the fundamental concepts of machine learning is essential for grasping how models are developed, trained, and evaluated. This section introduces key concepts that form the foundation of machine learning, including different learning approaches, the critical role of data, the structure of machine learning models, the training process, and methods for assessing model performance. Together, these topics provide a framework for understanding how machine learning systems operate and how their success can be measured.

3.1.1 Types of Learning Approaches

To emphasize the point highlighted previously, the concept of "learning" in machine learning holds a notably different meaning compared to the learning processes we experience in everyday life. Here, learning refers to a process rooted in mathematical and statistical models. Machine learning models aim to construct a representation that can represent the general characteristics of a given dataset. The higher the generalization capacity of the model, the greater its prediction accuracy.

In the field of machine learning, four fundamental training approaches are commonly used: supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning (Oladipupo, 2010). These approaches vary depending on the nature of the dataset.

In **supervised learning**, the model is trained on a labeled dataset where each input is associated with a specific output. This type of learning is particularly common in regression and classification tasks (Mitchell, 1977). In classification models, the output corresponds to a category, such as a dataset containing emails classified as “spam” and “not spam.” The model learns the patterns associated with each category to be able to correctly classify whether a new email is spam or not. On the other hand, in regression tasks, the dataset is labeled with continuous values, which are noncategorical outputs. For instance, a regression model trained on a dataset containing the house features along with their corresponding prices can predict the price of a house based on its characteristics.

In **unsupervised learning**, the model is trained on an unlabeled dataset, meaning the data lacks predefined output labels. In this method, the model is expected to analyze the dataset to identify underlying patterns. Clustering and dimensionality reduction are particularly common methods within this framework (I. Goodfellow et al., 2016). A common example of unsupervised learning is customer segmentation. The model is trained on a dataset of customers’ purchasing behaviors without any corresponding labels. Then, it analyzes the pattern of the dataset and clusters customers into groups based on the similarity of their behaviors.

Semi-supervised learning works with partially labeled datasets. In practice, the majority of the dataset is typically unlabeled. Although it might initially appear that having a large amount of unlabeled data could reduce the model's prediction accuracy. In fact, having more data points, even if unlabeled, often improves the model's ability to

learn the underlying pattern of the dataset. This method is especially useful in scenarios where labeling the entire dataset is costly or time-consuming (Hady & Schwenker, 2013).

Lastly, **reinforcement learning** is based on the concept of rewards and penalties. The model makes a series of decisions within an environment, resulting in rewards or penalties (Pavel et al., 2023). For example, consider a robot attempting to navigate out of a room. The robot earns a positive point for each step closer to the door and a negative point for each step away from it. Such a scenario is a typical example of reinforcement learning.

3.1.2 The Role of Data and Selection of Model Type

Data plays a crucial role in machine learning; one could even say that **data and the model** are two sides of the same coin. The model's performance is entirely dependent on the quality, quantity, and diversity of the data it is trained on. Without well-prepared data, even the most advanced models will struggle to produce meaningful results (Paullada et al., 2020).

If a dataset contains noise and inconsistent data points, the training process will likely result in poor prediction capabilities. Additionally, the amount of data directly affects the model's ability to generalize to unseen inputs. A dataset that is too small may lead to overfitting, while an excessively large but low-quality dataset may not improve performance.

Moreover, if the training data lacks diversity, the model is more likely to develop biases, leading to inaccurate or skewed predictions. In real-world applications, such a model may fail to perform during different scenarios (Kaur et al., 2020).

To achieve accurate and reliable results, it is essential to find a balance between data quality, dataset size, and diversity during training. This need for balance applies to all types of learning approaches and data formats. Whether the dataset consists of structured data, organized into rows and columns, unstructured data (such as images, texts), or semi-structured data, maintaining this balance remains crucial for ensuring optimal model performance and generalization.

The other side of the coin is **selecting the right model** for the task at hand. Data preparation and model selection are often interdependent processes rather than sequential steps. Typically, a specific task or problem defines the learning approach, which in turn influences the type of data required and the choice of model.

Table 3.1 Machine Learning Models (Created by the author)

Model Type	Tasks	Learning Paradigms	Applications	Common Algorithms
Predictive Models	Learn from labeled data to make predictions about future or unseen data.	Supervised Learning	Classification, regression, forecasting	Linear Regression, Logistic Regression, Decision Trees, Random Forests, Neural Networks
Pattern Discovery Models	Identify hidden patterns or structures in unlabeled data.	Unsupervised Learning	Clustering, anomaly detection, dimensionality reduction	K-Means Clustering, Hierarchical Clustering, PCA
Generative Models	Generate new data samples that resemble a given dataset.	Unsupervised & Supervised Learning	Image synthesis, text generation, data augmentation	GANs, VAEs, Diffusion Models
Decision-Making Models	Learn policies to make sequential decisions to maximize a reward.	Reinforcement Learning	Robotics, self-driving cars, AI planning	Q-Learning, Deep Q-Networks (DQN), Policy Gradient Methods
Sequence Models	Handle data that changes over time or depends on previous steps.	Supervised & Unsupervised Learning	Speech recognition, stock price prediction, NLP	RNNs, LSTMs, Transformers (BERT, GPT-4, ViT)

Machine learning models are categorized based on their purpose and the type of data they process. Different models are designed for tasks such as prediction, pattern recognition, data generation, decision-making, and sequence processing. Understanding these model types is essential for selecting the right approach for a given problem(C. M. Bishop, 2013).

3.1.3 How Models Learn: The Training Process

The training phase is a critical component of machine learning, where the model is expected to learn patterns of the dataset to make predictions. This process involves algorithm selection, data feeding, model parameter tuning, and model evaluation. The first step in the training process is data collection and preparation, a critical phase as the success of the model heavily relies on the quality of the data(Polyzotis et al., 2018). Collected data can come from various sources, such as sensor readings, user information,

or images. While the type of data may vary depending on the objective, the approach to the data preparation processes is generally similar.

The dataset needs to be both accurate and balanced. To achieve this, incorrect or inconsistent data is removed, and a choice of suitable approaches is made to address missing values. For example, missing values can be filled with the mean value or entirely removed from the dataset. This stage, known as **data cleaning**, aims to create a high-quality dataset that enhances the model's prediction accuracy (Borrohou et al., 2023). Following data cleaning, **standardization and normalization** are implemented to minimize bias during training. Numeric features are typically scaled to a standardized range, commonly between 0 and 1, while categorical features are **encoded** into numerical formats to be processed by the model (Chai et al., 2022).

The final step in data preparation involves splitting the dataset into training and test sets. The training set is used for the model to learn, while the test set is reserved for evaluating the model's performance on unseen data. Additionally, a validation set may be used to assess the model's performance more accurately during training.

After splitting the dataset, appropriate parameters are set for training the selected model, and the training process begins. The duration of training depends on the type of model and the size of the dataset. While some models may complete training in a few hours, larger datasets and more complex models can take days. For example, large-scale models like ChatGPT require significantly longer training times.

Throughout the training process, the model's performance is monitored using various metrics. If the changes in these metrics over time are insufficient, training may be paused, and parameters can be adjusted before restarting. In some cases, modifications to the dataset may be necessary, requiring the training process to begin from scratch. Training and model evaluation are dynamic processes, both parameters and datasets can be updated as needed to improve performance.

One of the most critical aspects of the training phase is **evaluating the model's performance**. This evaluation involves multiple metrics and different approaches. While metrics provide a numerical assessment of the model's accuracy, they are not always sufficient on their own. The model's performance is also assessed through visual and qualitative observations to gain insights (Handelman et al., 2019).

In generative models, **visual inspection** plays a particularly important role and is often one of the primary evaluation methods. However, metrics remain essential for

identifying errors in specific tasks, helping to fine-tune training parameters or adjust the dataset as needed.

The choice of evaluation metrics depends on the specific task and application, ensuring that the model's performance is assessed appropriately for its intended use.

Table 3.2 Common Evaluation Metrics (Created by the author)

Model Type	Metrics and Purpose
Predictive Models	Accuracy: Overall correctness of predictions Precision: The proportion of correctly predicted positive cases Recall: Ability to correctly identify all positive cases F1-Score: Balance between Precision and Recall ROC-AUC: How well the model separates classes Mean Absolute Error (MAE): Average absolute error between predicted and actual values Mean Squared Error (MSE): Average squared error between predicted and actual values Root Mean Squared Error (RMSE): Square root of MSE for better interpretability R² Score: Explains how well the model captures variance in data
Pattern Discovery Models	Silhouette Score: Measures cluster separation and compactness Dunn Index: Evaluates the compactness and separation of clusters Inertia (Within-Cluster Sum of Squares): Measures how well data points fit within clusters
Generative Models	Frechet Inception Distance (FID): Measures realism of generated images Inception Score (IS): Evaluates diversity and quality of generated images Structural Similarity Index (SSIM): Measures structural similarity between two images Perceptual Path Length (PPL): Evaluates smoothness of generated image transitions
Decision-Making Models	Reward Function: Measures the cumulative reward earned in decision-making tasks Q-Value: Estimates the value of taking a certain action in reinforcement learning Policy Gradient: Optimizes decision-making by adjusting policy updates
Sequence Models	Perplexity: Measures how well a language model predicts sequences BLEU Score: Evaluates text generation quality Dynamic Time Warping (DTW): Measures similarity between time-series sequences

These metrics help analyze the model's performance from different perspectives and play a crucial role in guiding the training process.

3.2 Deep Learning and Neural Networks

Deep learning has emerged as one of the most transformative areas within machine learning, enabling computers to learn complex patterns directly from raw data. At the core of deep learning are artificial neural networks, computational models inspired by the structure and functioning of the human brain. Unlike traditional machine learning methods that often require manually crafted features, deep learning models are capable of automatically extracting and hierarchically organizing features through multiple layers of abstraction. This capacity allows deep learning models to achieve remarkable success in a wide range of fields, including computer vision, natural language processing, speech recognition, and autonomous systems.

Neural networks consist of interconnected nodes, or "neurons," organized into layers - input, hidden, and output layers- that work together to process information. As data flows through these layers, the network adjusts internal parameters such as weights and biases through training, enabling it to progressively improve its predictions or outputs. Deep learning's power lies in its ability to model highly complex and non-linear relationships, making it particularly well-suited for tasks involving large-scale, high-dimensional datasets.

This section introduces the basic concepts of deep learning and neural networks, laying the foundation for understanding how these architectures function and how they are trained.

3.2.1 Neural Networks: The Basics

Just as the human brain is composed of neurons, artificial neural networks are made up of **nodes**. Simply put, a node receives an **input** (x_i), performs a basic computation, and then passes its **output** (z) to the next node. During this transfer, a process called the **activation function** (ϕ) comes into play (C. Bishop, 2006).

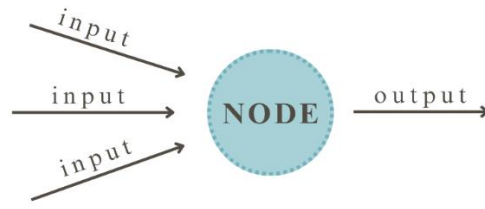


Figure 3.1 A perceptron node (created by the author)

In the traditional approach, the activation function, inspired by neurons, represents a threshold value. For instance, in McCulloch and Pitts' theoretical model, a neuron becomes active and transmits information only if the threshold is surpassed. If the threshold is not exceeded, the neuron remains inactive, and no information is transmitted (McCulloch & Pitts, 1943). However, modern neural network methods do not favor such a binary approach and allow nodes to be partially activated. This flexibility enables smoother information flow and allows nodes to establish non-linear relationships, which makes neural networks capable of solving more complex tasks.

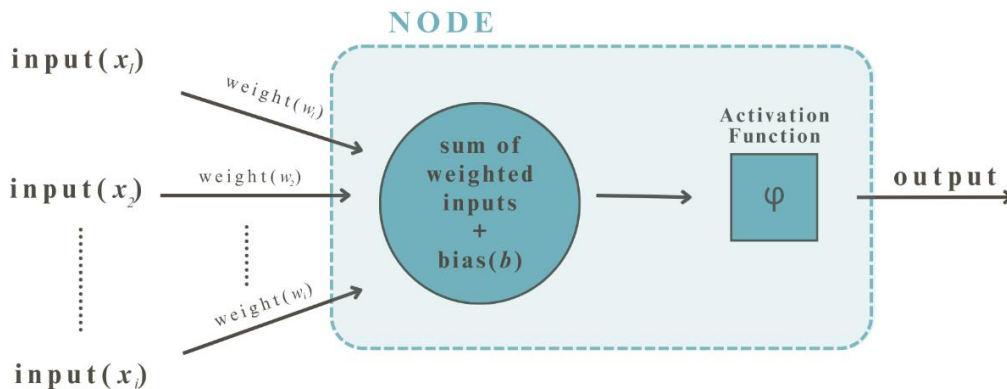


Figure 3.2 Structure of a node (created by the author)

The inputs received by nodes are typically weighted. **Weights** (w_i) are numeric values that determine to what extent an input affects the output. By multiplying each input by its corresponding weight, summing these products, and adding **bias** (b) to that, an intermediate output is calculated. Then, the activation function is applied to produce the final output. This output is passed to the next node, and the same calculations are repeated. Weights are updated during the training process through iterations to minimize loss and improve prediction accuracy.

These weighted computations occur within a structured architecture, where nodes are organized into distinct layers. Each layer's nodes perform computations and transfer

the results to the next layer. Typically, neural networks consist of **input, hidden, and output layers**, each serving a specific role in data transformation. This layered architecture enables neural networks to learn and recognize complex patterns (Kruse et al., 2016).

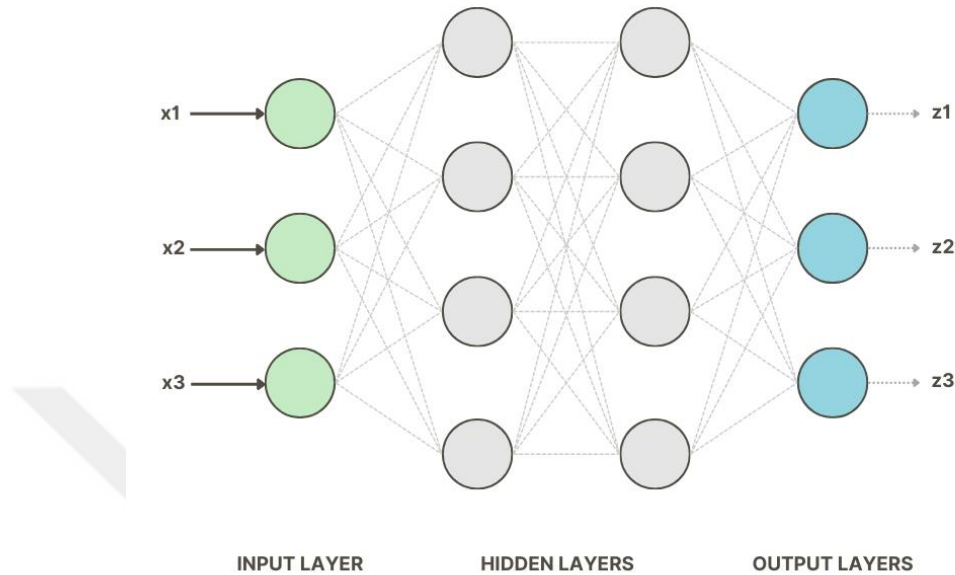


Figure 3.3 Representation of typical layers of a neural network (created by the author)

Input Layer: The input layer is responsible for receiving raw data and passing it to the hidden layers without performing any computations. Its nodes act as data holders, storing values such as pixel values in an image or numerical entries in a table. Unlike hidden and output layers, the input layer does not have weights or activation functions (Grillo, 2019).

Hidden Layers: The hidden layer is positioned between the input and output layers and is where the main computations take place. A neural network can have one or multiple hidden layers. If there is only one hidden layer, it is called a *shallow neural network*, whereas a network with two or more hidden layers is referred to as a *deep neural network*. The number of hidden layers significantly impacts the network's performance. Each node in the hidden layers has weights and an activation function, enabling the extraction of features and the recognition of patterns in the input data (Nakama, 2011).

Table 3.3 Output Layer Activation Functions (Created by the author)

Task Type	Activation Function	Output Format	Example
Binary Classification	Sigmoid (σ)	Value between 0 and 1	Spam detection: <i>Email classified as spam (1) or not spam (0)</i>
Multi-Class Classification	Softmax	Probabilities for each class	Digit recognition: <i>Predicting a handwritten number (0-9)</i>
Regression	Linear (No activation)	Any continuous number	House price prediction: <i>Predicting house price as \$250,000</i>

Output Layer: The output layer is the final layer of the network, responsible for generating the result based on the given task. It receives the transformed data from the last hidden layer and processes it accordingly. The output can vary depending on the task, such as binary or multi-class classification or regression(see Table 3.3). The activation function used in the output layer depends on the specific task (S. Yang et al., 2023).

3.2.2 Deep Learning: The Power of Layered Learning

Deep networks consist of multiple hidden layers between the input and output layers, where each layer contains nodes that process data using weights, biases, and activation functions. This multi-layered structure allows the network to learn patterns directly from raw data, dismissing the need for manually crafted features, which are essential in traditional machine learning (ML)(Kriegeskorte & Golan, 2019).

In traditional ML, what the model needs to learn must be explicitly defined. For instance, in urban map analysis using traditional ML methods, you would need to provide the model with predefined features such as:

- Building density: number of buildings per square kilometer
- Road connectivity: number of road intersections
- Green space ratio: percentage of parks in an area
- Proximity to major roads: distance to highways

However, with multiple hidden layers, a deep learning model automatically extracts these features in a hierarchical manner. As data passes through each layer, the model learns different levels of representation, progressively identifying more complex patterns. For example, in the case of learning urban maps:

- **Shallow Layers:** Detect simple patterns such as edges, colors, and basic shapes in images
- **Mid-Level Layers:** Capture combinations of patterns like textures, roads, and building shapes
- **Deep Layers:** Learn high-level representations, capturing entire urban layouts and structural compositions.

This multi-layered architecture, combined with non-linear activation functions, enables deep neural networks (DNNs) to learn complex patterns hierarchically through an iterative process. Each layer refines the extracted features, allowing the network to recognize increasingly abstract and meaningful representations in the data (Schmidhuber, 2015).

In this iterative process, data flows through the network during each pass to generate a prediction (Forward pass). The network then calculates the error margin using a loss function and adjusts the weights and biases through an optimization algorithm to minimize this error (Back propagation). This cycle repeats continuously, allowing the model to improve its prediction accuracy over time (see Figure 3.4).

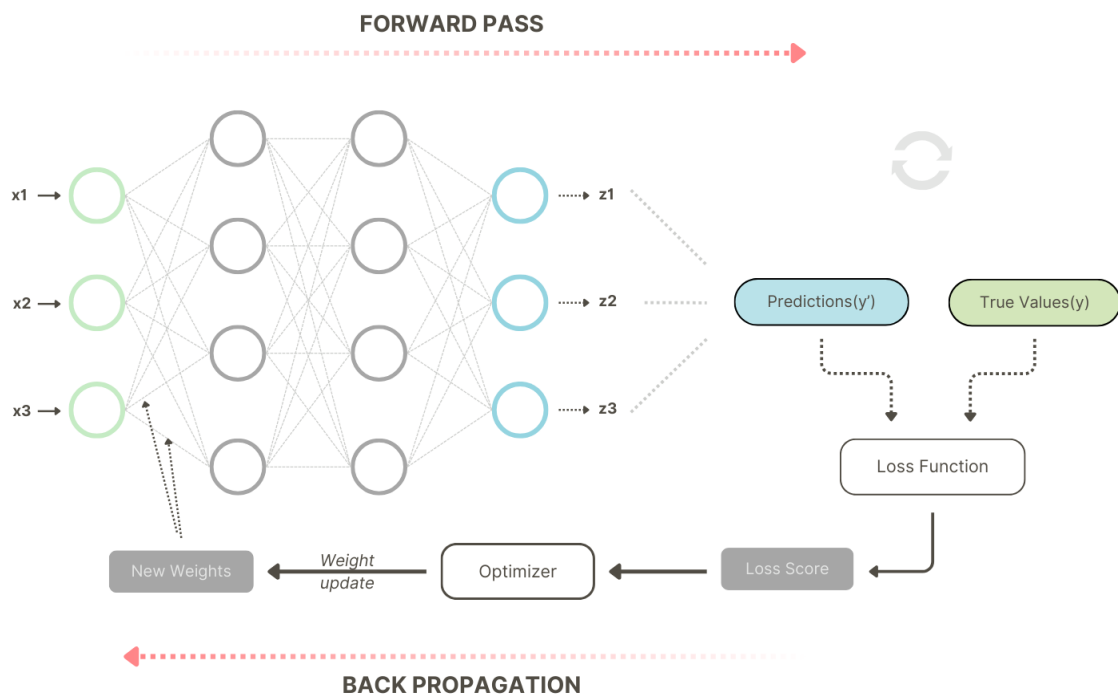


Figure 3.4 Iterative learning process (created by the author)

Forward Pass: The process in which training data flows through the network, undergoing transformations via weights, biases, and activation functions to generate an

output or prediction. This predicted output is then compared to the ground truth (true value) to assess the model's accuracy (Antorán et al., 2020).

Loss Function: A loss function is an algorithm that calculates the difference (error margin) between the model's prediction (output) and the true values (ground truth), generating a loss score. A low loss score indicates a good prediction, while a high score suggests a poor one. The goal of training is to minimize the loss score in order to improve the model's accuracy and produce more reliable predictions. (Q. Wang et al., 2020).

Optimizer: An optimizer is an algorithm that determines how weights and biases should be adjusted to minimize loss during training. It plays a crucial role in training neural networks. There are various types of optimizers, but gradient-based methods are the most commonly used for training neural networks. These optimizers compute the gradient of the loss function with respect to the weights, guiding the model to update its parameters in a way that moves toward the minimum loss value, improving accuracy over time (Abdulkadirov et al., 2023).

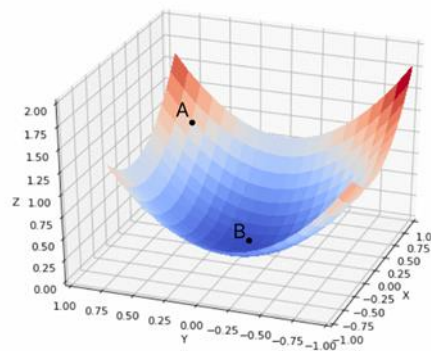


Figure 3.5 A Gradient for two weights (x, y axes) (Based on intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee)

The graphic above represents a loss function gradient for two weights. The X and Y axes correspond to the weight values, while the Z-axis represents the loss function values based on those weights. Point B is known as the **minimum (or minima)**, where the loss function reaches its lowest possible value. The role of an optimizer algorithm is to update the weights effectively, adjusting them in a way that reduces the loss function from Point A to Point B.

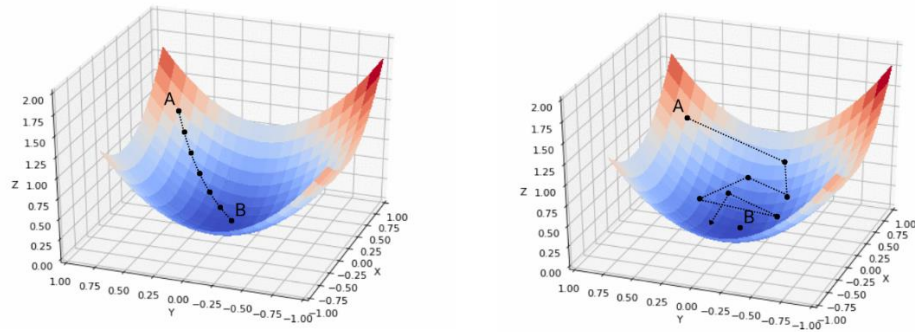


Figure 3.6 Difference between an appropriate learning rate (left) and a high learning rate (right) (Based on intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee)

During training, a parameter called the **learning rate** determines the extent of weight updates, controlling the step size taken in the direction of the negative gradient. A high learning rate may cause the optimizer to overshoot the minima, preventing convergence, while a low learning rate can lead to slow training or the model getting stuck in a saddle point. Therefore, the learning rate must be carefully adjusted (F. He et al., 2019).

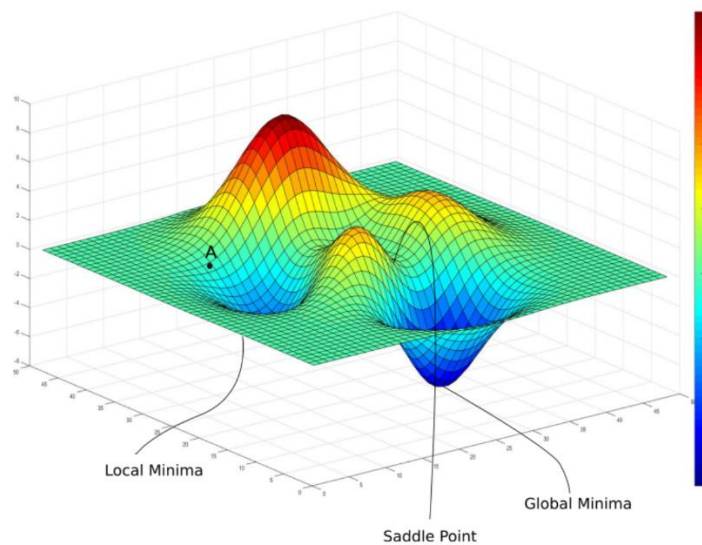


Figure 3.7 More realistic gradient landscape (From intro to optimization in deep learning gradient descent by Ayoosh Kathuria and Shaoni Mukherjee)

These simple representations only illustrate two weights, but in actual training, there are millions of weights. As a result, the gradient is much more complex, and calculating it is computationally expensive. Due to this computational cost, passing the entire dataset through the network in a single forward pass is not preferable. To address this, instead of passing the entire dataset through the network in a single forward pass,

training is controlled by a parameter called **batch size**. Instead of processing the entire dataset, during each iteration, a subset (batch) of the input data is passed through the network to generate outputs, compute gradients, and update weights accordingly (Masters & Luschi, 2018)

Backpropagation: After the loss calculation, the gradient of the loss function with respect to the weights is also computed. This gradient is sent backward from the output layer to the input layer, moving layer by layer in reverse. This means the error signal propagates backward through the network. The optimizer then adjusts the weights in the right direction to minimize the error. This process is called backpropagation, and it is repeated at every iteration to improve the model's performance (Werbos, 1990).

3.2.3 Types of Neural Networks in Deep Learning

Neural networks come in different architectures, which vary based on how nodes (neurons) are organized and how layers interact. Neural networks vary not just in node organization but also in how information flows. Each type of neural network (NN) is designed to handle specific types of data and tasks. Furthermore, even within each type of neural network, there are numerous variations optimized for different applications.

Table 3.4 Types of Neural Networks (Created by the author)

Neural Network Type	Architecture	Type of Data	Types of Tasks	Examples
Feedforward Neural Networks (FNNs)	Layers are sequentially connected, information flows in one direction (input to output), no loops.	Structured data (tabular, numerical) or simple unstructured data	Classification, regression, basic predictive modeling	MLPs (Multi-Layer Perceptrons),
Convolutional Neural Networks (CNNs)	Uses convolutional layers to extract spatial features, followed by pooling and fully connected layers	Image data, spatially structured data, video frames	Image classification, object detection, medical imaging	AlexNet, VGGNet, ResNet, EfficientNet
Recurrent Neural Networks (RNNs)	Includes loops and connections that allow information to persist over time, enabling sequence learning	Sequential data such as time-series, speech, and text	Speech recognition, sentiment analysis, time-series forecasting	LSTMs, GRUs, DeepSpeech

Generative Adversarial Networks (GANs)	Consists of two networks: a Generator (creates fake data) and a Discriminator (distinguishes real vs. fake data)	Images, text, or structured data for generation tasks	Image generation, style transfer, data augmentation	Pix2Pix, StyleGAN, CycleGAN, BigGAN, DALL·E
Transformer Networks	Uses self-attention mechanisms to process input sequences in parallel rather than sequentially	Text, time-series, and high-dimensional data	Machine translation, text generation, NLP applications	BERT, GPT-3, GPT-4, T5, ViT (Vision Transformer)

Table 3.4 summarizes major types of neural networks by outlining their architectures, data types, common tasks, and key examples. Feedforward Neural Networks (FNNs) are used for basic prediction with structured data. Convolutional Neural Networks (CNNs) handle image-based tasks by extracting spatial features. Recurrent Neural Networks (RNNs) are suited for sequence data like text and speech. Generative Adversarial Networks (GANs) generate new data through a generator–discriminator setup. Transformer Networks process sequences using self-attention, excelling in language and high-dimensional data tasks.

Chapter 4

Generative Adversarial Networks

4.1 Generative Algorithms

Generative algorithms represent a category of algorithms designed to produce new synthetic data based on a given dataset or specification. The term "generative" stems from these algorithms' ability to create something novel or original. In the realm of machine learning, generative algorithms are frequently linked to generative models. Generative models are a type of statistical model specifically designed to generate a new set of samples or data that closely resemble the original dataset. In other words, generative models have the capability to examine a dataset, identify underlying patterns, and then create new data based on those observed patterns.

There are two main types of generative models:

1. Explicit density model
2. Implicit density model

Explicit density models explicitly define the probability distribution of the data. They learn the patterns within this probability distribution and leverage these patterns to generate new data. Explicit density models encompass techniques such as Markov Chain methods, Maximum Likelihood Estimation (MLE), and Approximate Inference.

In the context of Maximum Likelihood Estimation (MLE), a common issue arises known as over-smoothing of the generated data. MLE has a tendency to average out details, leading to a less realistic representation of the data. This over-smoothing effect can result in a loss of finer nuances and details in the generated samples.

Generative models trained with approximate inference face challenges in solving the objective function, which quantifies the model's performance and is a mathematical expression optimized during training. The complexity introduced by real-world data makes the objective function highly intricate. Consequently, the approximate inference method merely allows the model to skirt around the complexities of the objective function, reaching only the lower bound instead of directly tackling its complexity. In

essence, the model cannot directly pinpoint the optimal strategy for optimizing the objective function. Instead, it can only approximate a certain level of performance, achieving the best outcome possible without fully addressing the complexities of the actual intricate problem (J. Gui et al., 2023).

The Markov chain method can indeed be employed for training models, but it comes with a significant drawback: it is computationally expensive.

The explicit density model faces difficulties related to computational tractability. This problem often results in an inability to accurately capture the intricate complexity present in the actual data distribution, especially when dealing with learning high-dimensional data distributions.

Implicit models, in contrast to explicit models, do not directly estimate the probability distribution of the data. Consequently, they generate data instances that may not strictly adhere to the underlying probability distribution (Bengio et al., 2014). Essentially, these models produce new data instances without clear rules, operating in a more abstract and flexible manner. Prior to the advent of Generative Adversarial Networks (GANs), training implicit density models often necessitated techniques like ancestral sampling or Markov chain-based sampling (Bengio et al., 2013). However, these methods could be inefficient and constrain the practical applications of implicit density models.

GANs, as a prominent example of implicit density models, have demonstrated success in various applications, including image synthesis, style transfer, and data generation tasks.

GANs were proposed to overcome the limitations of **generative algorithms**, with the primary objective of generating realistic data through an adversarial learning process. The fundamental concept of GANs involves two components: the generator and the discriminator. The generator's role is to create data with the aim of making it as realistic as possible, while the discriminator strives to distinguish between fake and real data examples (I. J. Goodfellow et al., 2014). Consequently, both the generator and the discriminator iteratively improve through this adversarial process. This unique approach provides GANs with a distinct advantage over other generative algorithms (J. Gui et al., 2023):

1. GANs exhibit the capability to parallelize the generation process, enabling the efficient and simultaneous creation of multiple examples. This term implies the ability to generate several instances simultaneously, unlike a sequential or one-

at-a-time process. This starkly contrasts with other generative algorithms like PixelCNN and fully visible belief networks (FVBNs), which lack the ability to parallelize the generation process.

2. GANs offer flexibility in designing the generator, imposing fewer restrictions on its structure than alternative generative methods.
3. GANs are subjectively acknowledged for producing superior examples compared to other generative methods. This superiority may pertain to the visual quality or realism of the generated data.

4.2 Adversarial Learning: The Architecture of Generative Adversarial Networks

As previously mentioned, GANs consist of two distinct neural networks: the generator and the discriminator (see Figure 4.1). The generator's objective is to produce data that closely resembles the samples from the training dataset. Conversely, the discriminator is tasked with examining the data and discerning between real and fake instances. While the generator endeavors to create data that can deceive the discriminator, the discriminator seeks to enhance its capacity to classify data accurately. This dynamic interplay between the generator and discriminator forms the adversarial learning process characteristic of GANs.

GANs are structured as a minimax game, a concept derived from game theory. In this two-player game, one participant strives to minimize the outcome while the other aims to maximize it. The term "minimax" denotes the strategy of minimizing the maximum potential loss, illustrating the adversarial nature of the learning process between the generator and the discriminator in GANs.

In the GANs framework, the generator (minimizer) and discriminator (maximizer) take on players' roles in a minimax game. The generator's objective is to minimize the discriminator's ability to distinguish between fake and real data, while the discriminator strives to maximize its accuracy in distinguishing between the two. This adversarial interaction establishes a continuous learning process, with both models refining their strategies through iterative steps. This iterative process persists until an equilibrium is achieved, a state where neither player can unilaterally alter their strategy to attain a better outcome (I. J. Goodfellow et al., 2014).

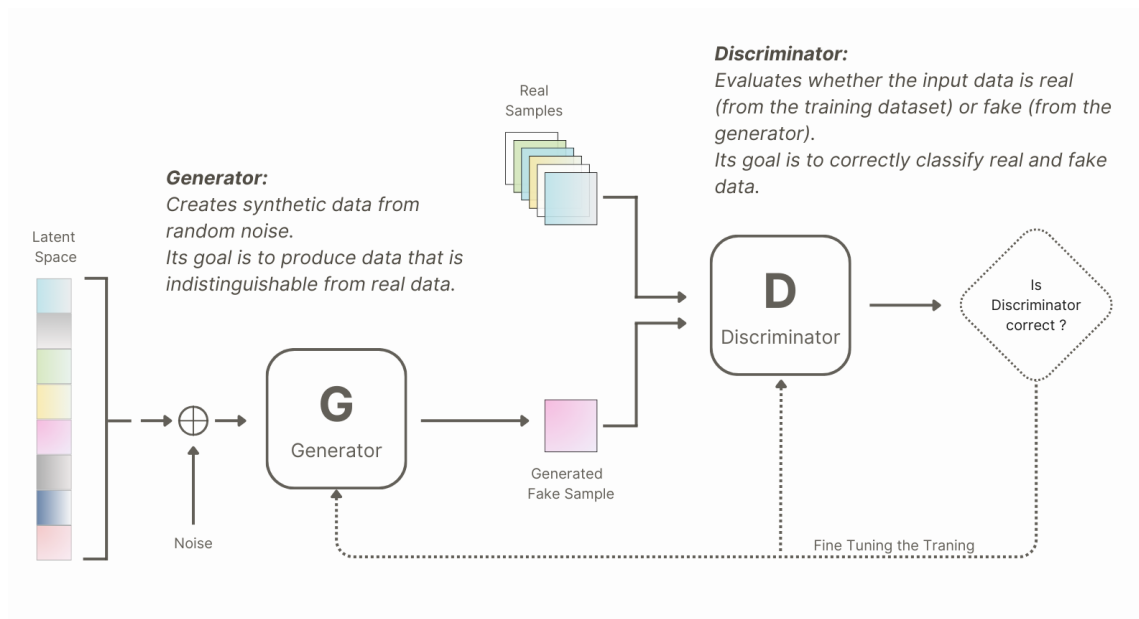


Figure 4.1 Generative Adversarial Networks architecture

In the context of GANs, this equilibrium signifies that the generator produces data realistic enough to confound the discriminator, making it unable to differentiate between the generated and real data. In this state, the distribution of the generated data closely aligns with that of the real dataset, even though the generator does not explicitly model the underlying probability distribution.

Enhancing the strategy is achieved by employing the backpropagation mechanism, which efficiently updates the parameters of both the generator and the discriminator during the learning process. In simpler terms, backpropagation is a method for both the generator and discriminator to learn from their mistakes.

4.3 Common Challenges in Training

During the training of GAN models, several challenges can arise. Some of these challenges are common in general machine learning, while others are unique to GANs due to their adversarial architecture.

One of the unique challenges of GANs is their **hypersensitivity** to hyperparameters. Small changes in parameters like learning rate, batch size, or optimizer settings can significantly affect training stability, sometimes leading to failure to converge or erratic behavior (Dumont et al., 2022).

Another GAN-specific challenge is **mode collapse**, where the generator fails to learn the full data distribution and collapses to producing only a few variations of a few modes rather than generating a diverse set of outputs. In the case of mode collapse, the

model repeatedly generates nearly identical outputs, reducing the variety of generated data (Kossale et al., 2022).

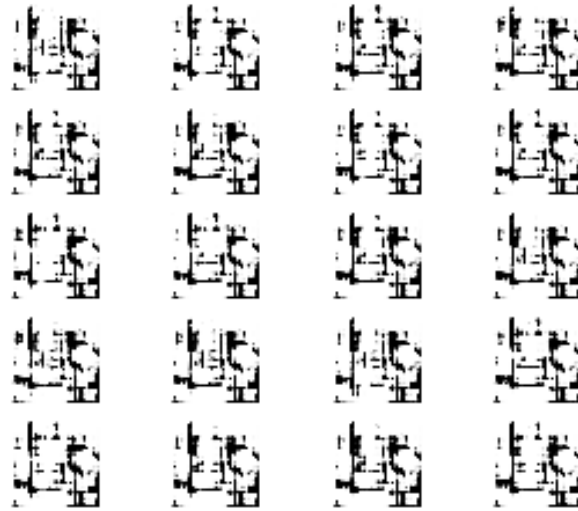


Figure 4.2 Mode collapse, the doctoral dissertation of Can UZUN (2020)

Another challenge commonly encountered in deep learning, but particularly problematic for GANs, is the **vanishing gradient** problem. During training, the gradients become so small that the network can no longer effectively update its weights, slowing or even halting learning (Hochreiter, 1998). Whenever this issue affects the discriminator, it weakens the adversarial feedback mechanism, and the generator receives little useful information for improvement. This imbalance can contribute to mode collapse.

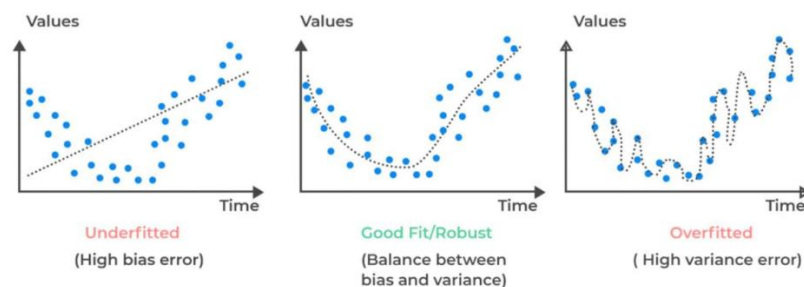


Figure 4.3 Underfitting and overfitting in a regression model (What Are Overfitting and Underfitting, Dooinn KIm, Medium)

Other common issues that can arise with any machine learning model are overfitting and underfitting. **Overfitting** occurs when a model becomes too complex and memorizes the training data instead of learning the underlying patterns. As a result, the model performs exceptionally well on the training data but fails when it encounters new, unseen data. To address overfitting, reducing the complexity of the model can help improve its ability to generalize the data.

In the case of **underfitting**, the model fails to learn the patterns in the data and performs poorly on both the training data and unseen data. This indicates that the model is too simplistic and unable to capture the complexity of the dataset. To address underfitting, the complexity of the model can be increased (Bashir et al., 2020).

4.4 Generative Adversarial Networks in Urban Studies

GANs have been utilized in various studies within the broader field of architectural research. There are multiple perspectives from which the literature can be examined. One way is by considering the size of the dataset used, or the availability and openness of that data. Another is by looking at the scale of application, whether the focus is on the city, district, or building level. Additionally, the type of task for which GANs were applied, such as generation, transformation, or completion, offers another lens through which to categorize and analyze these studies.

A review titled "*Generative Adversarial Networks in the Built Environment*", conducted by Wu et. al. (2022), remains the most comprehensive review work on the subject to date. The researchers systematically screened databases such as Scopus and Web of Science, selecting and analyzing 100 relevant studies. To structure their findings, they proposed a two-part classification method based on the specific tasks GANs were designed to perform. To define the nature of each study, the authors first categorized the tasks into three main groups:

- **Data synthesis**, where the output differs in type or format from the input and is used either qualitatively or quantitatively.
- **Data augmentation**, where the output closely resembles the input but represents a more refined or complete version.
- **Design automation**, where the output differs from the input in format or nature, and while can not be used directly quantitatively, it offers visual or conceptual insights valuable for design processes.

In the second step, the studies were further classified based on their specific application domains within the built environment.

This task-based approach offers a comprehensive and holistic framework for analyzing GAN applications in architectural and urban contexts. For this reason, the classification methodology presented in the review has been adopted within the scope of

this research and used as a primary reference throughout the review of the literature (see Table 4.1).

Data Synthesis: GANs can be used to generate synthetic data that can be used both qualitatively and quantitatively. For example, it is used for constructing detailed 3D urban models from street-level images and segmentation maps. It is also used to predict human behavior in built environments and optimize energy use. There are studies where GANs were utilized for predicting traffic patterns, population density, and land use scenarios. Furthermore, researchers used GANs to generate realistic facades, satellite images, and street views. Other applications include style transfers such as restyling cartography of urban maps.

Data Augmentation: In comparison to data synthesis, data augmentation involves output data that is closely related to the input. Rather than generating entirely new content, the output serves as an enhanced version of the input by improving its quality, filling gaps, or restoring missing information. One of the significant applications of GANs in this category is data upsampling, where GANs are employed to fill in missing values in spatiotemporal datasets, such as traffic records. Another notable use is façade restoration, in which the missing parts of building façades are completed with realistic detail.

It is also used to increase the resolution of satellite imagery, perform street view inpainting, and convert floor plans into machine-readable vector formats.

Design automation: GANs are used to assist and accelerate architectural and urban design processes. In several studies, models have been trained to generate 3D models and interior scenes from inputs such as floor plans, sections, or even raw geometric data. Also used in design analysis support, for example, classifying architectural styles using unsupervised learning. For design assistance, GANs are employed in tasks such as coloring master plans, converting sketches into realistic renderings, and generating three-dimensional mass models from two-dimensional images. These applications help bridge the gap between conceptual visuals and more refined representations. Another use of GANs in this category is design inspiration, where generated visuals serve as a starting point for creative exploration. This research also falls under the category of design automation, with its application domain positioned as a tool that supports design while remaining informed by the spatial memory of the urban context.

Table 4.1 GANs in Urban Studies

TASK	APPLICATION DOMAIN	STUDIES
Data Synthesis	3D Urban Reconstruction	(Du et al., 2022), (Kim et al., 2020), (Kelly et al., 2018)
	Building Performance Simulation/Optimization	(Q. He et al., 2021), (Chokwitthaya et al., 2019), (Chokwitthaya et al., 2020)
	Cross-View Image Translation	(Deng et al., 2018), (Regmi & Borji, 2018), (Regmi & Borji, 2019), (Tang et al., 2019), (Tang, Liu, et al., 2020), (Ding et al., 2020)
	Data Estimation and Urban Simulation	(Albert et al., 2018), (G. Zhang et al., 2020), (Y. Zhang et al., 2022), (Y. Zhang et al., 2019), (Ibrahim et al., 2023), (S. Sun, Mu, Feng, et al., 2021), (Yao et al., 2021), (Johnsen et al., 2022)
	Façade, Satellite, and Street View Generation	(M. Wang et al., 2019), (Tang, Xu, et al., 2020), (Zhu et al., 2020), (Venator et al., 2021), (B. Zhao et al., 2021), (Steinfeld, 2022), (C. Sun et al., 2022)
	Cartographic Style Transfer	(C. Xu & Zhao, 2018), (Kang et al., 2019), (J. Li et al., 2020), (Y. Zhang et al., 2020)
	Semantic Segmentation and Change Detection	(Shi et al., 2018), (W. Wu et al., 2018), (Varia et al., 2018), (Y. Chen et al., 2019), (Hou et al., 2020), (S. Sun, Mu, Wang, et al., 2021), (Alawadhi & Yan, 2021)
	Data Upsampling	(S. Wang et al., 2018), (B. Yang et al., 2022)
	Facade Restoration	(Qin et al., 2017), (H. Zhang et al., 2018), (J. Li et al., 2019), (L. Zhao et al., 2020), (H. Liu et al., 2021), (Güzelci et al., 2024)
	Satellite Image Enhancement	(Pham & Bui, 2021), (Ikeno et al., 2021)
Data Augmentation	Digital Surface Model Enhancement	(Bittner & Korner, 2018), (Pang & Biljecki, 2022), (Y. Wang et al., 2021)
	Street View Cleaning and Inpainting	(J. Zhang et al., 2021), (Hettiarachchi et al., 2021)
	Floorplan Cleaning	(Kim et al., 2018), (Cho et al., 2020), (Dong et al., 2021)
	3D Model/Scene Generation	(S. Zhang et al., 2019), (H. Zhang & Blasetti, 2020)
Design Automation	Design Analysis	(H. Zhao et al., 2019), (N. Wang et al., 2019), (S. M. Wang & Huang, 2019), (Wijnands et al., 2019), (Rachele et al., 2021)
	Design Assistance	(Sharma et al., 2018), (Chan & Spaeth, 2020), (Ye et al., 2022), (Qian et al., 2022), (Navarro-Mateu et al., 2021),

Design Inspiration	(J. Huang et al., 2021), (Danchenko, 2021), (Del Campo, 2021), (J. Chen & Stouffs, 2021)
Design Optimization	(W. Liao et al., 2021), (Sato et al., 2020)
Floorplan Generation	(Newton, 2019), (Uzun, 2020), (Chaillou, 2019), (Y. Liu et al., 2020), (C. Zhao et al., 2021)
Spatial Layout Generation	(As et al., 2018), (Nauata et al., 2020), (Nauata et al., 2021), (Owaki & Machida, 2020), (D. Wang et al., 2020b)
Photo-realistic Rendering	(Y. Gui et al., 2021), (F. Yang et al., 2019)



Chapter 5

Urban Fabric Generation: Hatay City Center

5.1 Context & Historical Background

Hatay is one of Anatolia's oldest settlement centers, with historical evidence of human habitation dating back as far as 100,000 BCE. Throughout its extensive past, the city has been shaped by numerous civilizations and empires, each leaving behind distinct cultural and architectural legacies. From prehistoric times through the Bronze Age and beyond, Hatay has experienced continuous transformation under the influences of the Hittites, Assyrians, Persians, Romans, Byzantines, Islamic dynasties, Crusaders, Seljuks, Mamluks, and Ottomans. This historical richness has layered the city with diverse architectural styles and cultural traditions, creating a vibrant heritage that makes Hatay uniquely representative of Anatolia's multifaceted history.

In February 2023, the city suffered severe damage due to two powerful earthquakes, both registering above magnitude 7, that struck southern Turkey and northern Syria. The region continued to experience intense seismic activity, with over 45,000 aftershocks, some exceeding magnitude 7. The World Health Organization declared a Level 3 emergency in response to the disaster. Hatay was among the cities that suffered the most extensive damage. A significant portion of the city either collapsed during the earthquakes or was later deemed too severely damaged and required demolition. This resulted in a severe loss of urban fabric and a painful disruption to the continuity of urban memory.

5.2 Data Preparation

The first phase of the research is the creation of datasets required for training the GAN model. This phase has been completed, with three datasets created at different scales: 1:2000, 1:3000, and 1:5000 (Figure 1.5). QGIS was chosen for dataset creation due to its open-source nature and frequent use in the literature. While creating Hatay's general map, OpenStreetMap data was imported into QGIS via the QuickOSM plugin using the quick query tool. The key-value pairs for the quick query tool were as follows:

1. **For roads**, a single key with five values was used:
Highway: trunk, primary, secondary, tertiary, residential
2. **For green spaces**, three keys were used, each with multiple values:
Landuse: recreation_ground, cemetery, meadow, farmland, forest
Leisure: garden, pitch, park
Natural: wood, scrub, forest
3. **For buildings**, the "building" key was used to obtain existing structures, and for structures destroyed in the earthquake, the key "Destroyed:building" was added:
Building: yes
Destroyed:building: yes
4. **For rivers and streams**, one key with three values was used:
Waterway: river, stream, ditch

Buildings were represented in a Nolli Map style in the dataset, as this approach was predicted to improve training success (Boim et al., 2022). Buildings were shown in black (RGB 0 0 0), roads in red (RGB 255 95 12), green spaces in green (RGB 130 210 150), and rivers and streams in blue (RGB 50 140 180), with white (RGB 255 255 255) as the background color. Subsequently, a Python script was used to create the datasets.

5.3 Generative Adversarial Networks Experiments

The application of Generative Adversarial Networks (GANs) to generate urban fabric is detailed through a series of experiments. Using three distinct datasets at scales of 1:2000, 1:3000, and 1:5000, GAN models were trained to evaluate how variations in map scale influence the model's ability to generate accurate and realistic urban structures. The experiments highlight both successes and challenges in achieving high-fidelity

representations of the urban environment, providing insights into the factors impacting the quality of GAN-generated outputs in an urban design context.

The experiments were conducted using the Pix2PixHD architecture within the Google Colab environment, which provided access to cloud-based GPU resources. All experimental data, including training datasets and model outputs, were stored and managed through Google Drive to ensure accessibility and efficient file handling throughout the process.

Table 5.1 Description of the Parameters Used in the Experiments

Parameter	Used for	Description
--lr	Generator	Initial learning rate for Adam optimizer (shared by generator and discriminator)
--netG	Generator	Type of generator network (e.g., 'global', 'local', 'resnet_9blocks')
--ngf	Generator	Number of generator filters in the first conv layer
--lambda_feat	Generator	Weight for the discriminator's feature matching loss (used by the generator)
--norm_G	Generator	Normalization method in generator (e.g., batch, instance)
--niter	Training	Number of iterations with initial learning rate
--niter_decay	Training	Number of iterations to decay the learning rate to
--num_D	Discriminator	Number of discriminators to use (multi-scale GANs use more than one)
--n_layers_D	Discriminator	Number of conv layers in the discriminator if 'n_layers' model is selected
--ndf	Discriminator	Number of filters in the first conv layer of the discriminator

5.3.1 Trial Experiment

This trial experiment was conducted to verify if the model was running without an issue. For this trial, the 1:5000 scale urban maps dataset was used. During this process, an error was encountered due to the fact that the Google Colab environment runs Python 3.9, but the Pix2PixHD prerequisites Python 2 or 3. Because the new version of the Python updated `fractions.gcd` to `math.gcd`. To resolve this issue, I imported `math.gcd` into the `train.py` file and replaced all instances of `fractions.gcd` with `math.gcd`. After making this adjustment, the code ran successfully.

In this trial, I also tested the capabilities of the free version of Google Colab to assess its potential for training the Pix2PixHD model. The training lasted for a few epochs

since each epoch took about 75 minutes. Given that training typically should the references include only journal articles, or are conference proceedings and book chapters also acceptable?y requires hundreds of epochs, the free version proved unsuitable for the task at hand. Consequently, I decided to upgrade to the pro version of Google Colab to continue with the training experiments.

5.3.2. Experiments of Scale 1:5000

A total of 15 experiments were conducted at the 1:5000 scale. Of these, 9 were consolidated and presented collectively under the title Experiment 6 for readability due to their similar structure and comparable results. After the final trial (Experiment 7) yielded satisfactory outcomes, no additional experiments were considered necessary for this scale. The training process involved the use of varied parameters to assess their effects on model performance. Two different datasets were utilized, one being an augmented version of the other, to increase data variety and support generalization. Furthermore, three distinct label maps, created using different masking techniques, were employed to examine their influence on the quality of the generated outputs.

5.3.2.1 Experiment 1

In the first experiment, the training process reached 300 epochs, after which the model was tested. During testing, it was noted that the model could not fill the empty spaces in the input images, and the input-output pairs remained almost identical. The primary reason for this outcome was found to be the absence of any blank areas in the images of the label set, preventing the model from learning the task of gap-filling. As a result, it was decided that the label dataset should be reconfigured to include blank areas. In order to introduce erasures to the label set, a Python script was used to employ random noise-based masking. With these modifications, it is anticipated that the model will perform better in completing gaps when trained on the revised label set.

Reconfiguration of the label set: To create the updated label set, a random noise field was first generated at a chosen *scale*, determining the level of detail within the pattern. This noise field served as the foundation for introducing randomness into the labels. Gaussian smoothing was applied to the noise field to ensure smoother and more interconnected structures. The *connectivity* parameter was adjusted to control how

cohesive the structures appeared, ensuring a balance between isolated and interconnected patterns.

Once the smoothing process was completed, a *threshold* was applied to the gradient to transform the continuous noise field into a binary mask. This step identified regions that would remain visible and areas that would be masked. To further refine the mask, a slight *erosion* of the visible regions was introduced. This adjustment helped unify the remaining structures while preventing the visible areas from overwhelming the masked spaces.

Through this systematic process, random patterns were transformed into organic, web-like masks. These masks introduced naturalistic disruptions while maintaining coherent patterns.

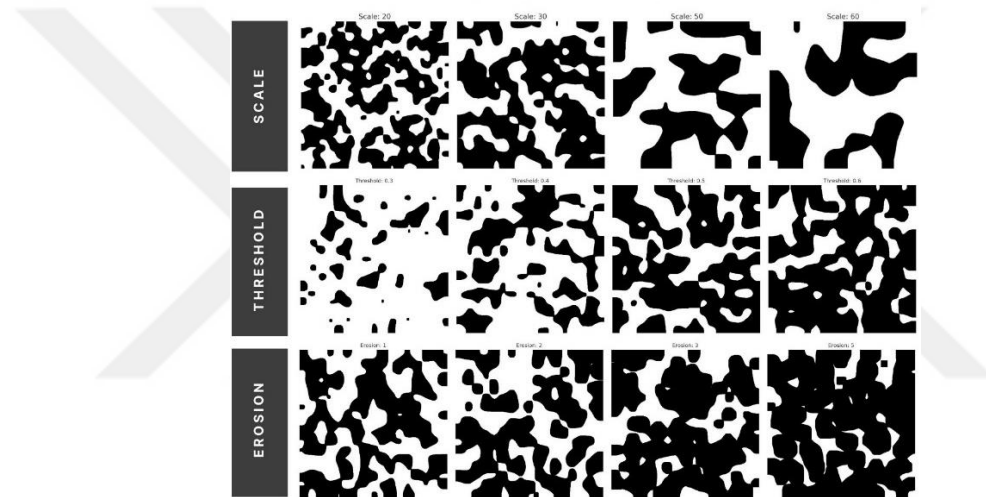


Figure 5.1 Effects of different values of scale, threshold, and erosion on the pattern of the mask

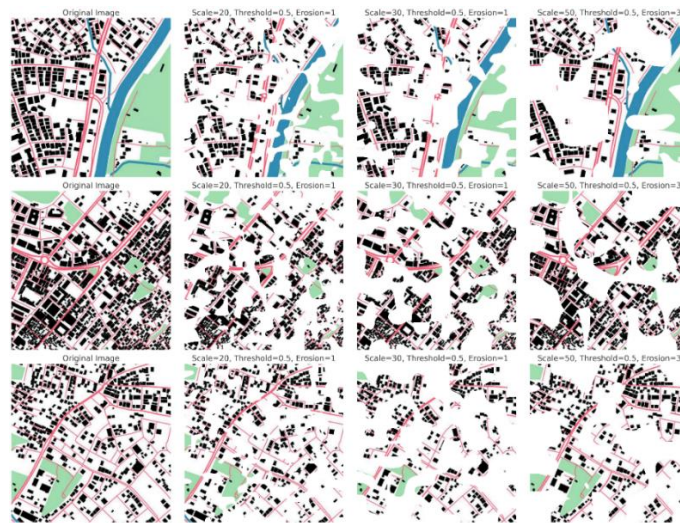


Figure 5.2 Applying different parameters to understand which one would be more suitable for training

Additionally, since calculating the FID score was not part of the default metrics provided by pix2pixHD, the necessary code lines were added to the `train.py` file.

5.3.2.2 Experiment 2

Training Setup: The training session continued for 245 epochs with a batch size of 10. And the training took around 11 hours. The `--niter_decay` parameter was set to 200, and the model checkpoints were saved every 5 epochs. The learning rate remained at its default value of 0.0002, and the feature matching loss, controlled by `--lambda_feat`, was also left at its default value of 10. Additionally, the FID score was calculated every 5 epochs to monitor improvements in the generated outputs.

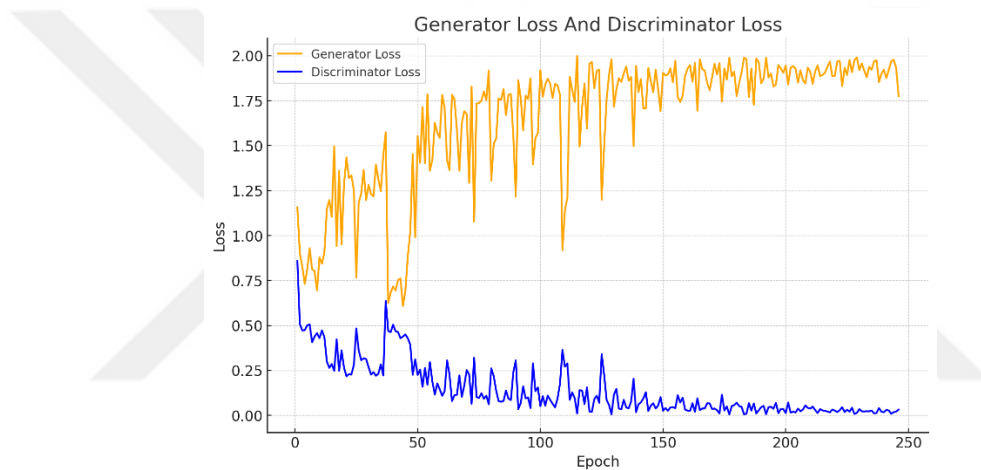


Figure 5.3 Change of Generator loss and Discriminator loss throughout the epochs

Loss Function and FID Score Analysis: The generator and discriminator losses consistently improved until approximately Epoch 150, where they appeared to reach a balance. However, after Epoch 150, the discriminator loss approached zero, indicating that the discriminator became far too strong (see Figure 5.3). This imbalance restricted the generator's ability to take creative freedom, forcing it to overfit the training data instead of learning to generalize. Despite this, the FID score continued to decrease, albeit at a slower rate (see Figure 5.4). A significant slowdown in the rate of FID score improvement was observed after Epoch 100, and the gains became minimal beyond Epoch 150.

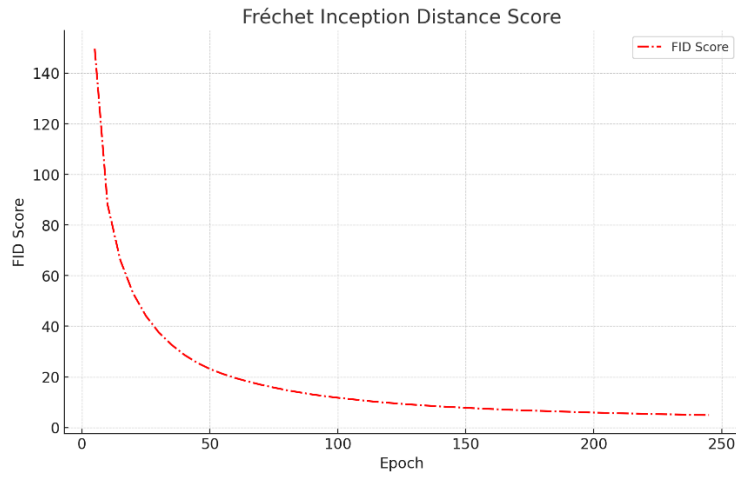


Figure 5.4 Change of Fréchet Inception Distance Score throughout the epochs

Output Quality: Throughout the training process, the generated images showed noticeable improvements, particularly in filling blank areas (see Figure 5.6). However, the generation remained heavily dependent on the input label maps. Early in training, the model struggled to create sharp edges, and the generated images were rough and undefined. As training progressed, the visible areas on the label maps became sharper, and completions grew increasingly close to the real images. Despite this progress, the edges never achieved sharpness before overfitting, and the generator's reliance on label maps resulted in outputs that lacked novelty.



Figure 5.5 Generated image during training epoch 241, a sign of overfitting

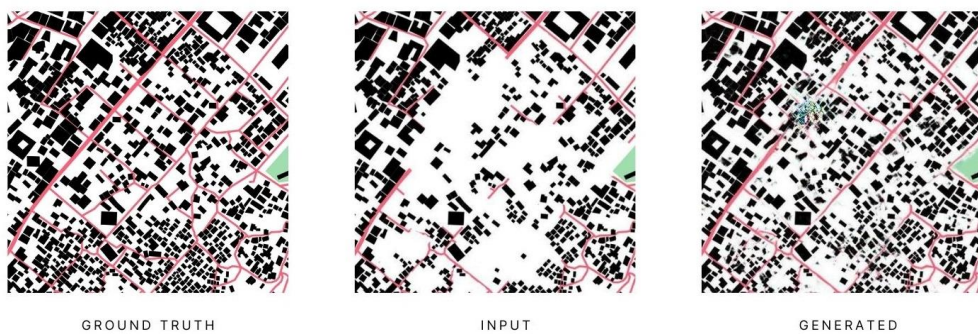


Figure 5.6 Example of the test result at 245 epochs

Evaluation and Observations: The reconfiguration made to the label set helped the model learn to fill gaps more effectively, but this improvement proved insufficient, even before overfitting set in (see Figure 5.7). The main issue appears to be the model’s dependence on pixel-by-pixel matches between the label maps and the dataset, causing the generator to mimic the label set partially and then attempt to fill the gaps. This behavior limited its ability to generalize and likely contributed to overfitting.

To mitigate this issue, introducing slight Gaussian noise to the label set could disrupt the pixel-perfect matches, forcing the generator to generalize its outputs rather than replicating the label data. Additionally, lowering the batch size could help improve edge sharpness by enabling the generator to focus on finer details during training. Reducing the learning rate to 0.0001 could also prevent overfitting by allowing the generator to make smaller, more controlled updates. Given that overfitting appears to have started around Epoch 150, and the rate of FID score improvement slowed significantly after Epoch 100, it would be viable to continue training from Epoch 100 with these adjusted parameters and a refined label set.

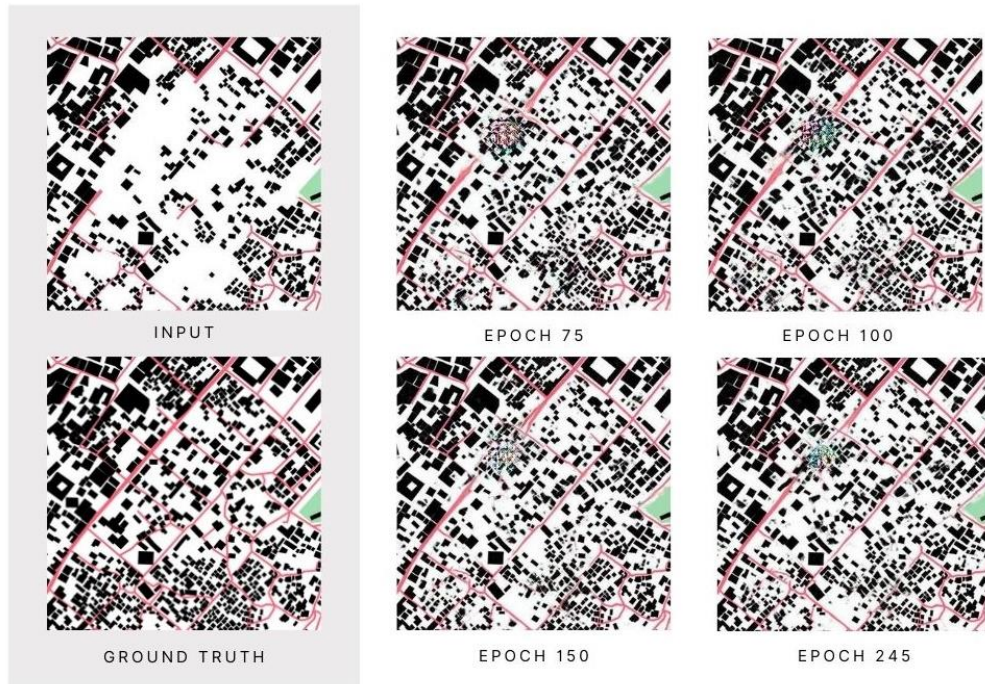


Figure 5.7 Comparison of the test result at 75, 100, 150, and 245 epochs

5.3.2.3 Experiment 3

Training Setup: Training resumed from the previous session at epoch 100. Key adjustments included reducing the batch size to 8 and setting the `--niter_decay` parameter to 50 epochs. The feature matching weight (`--lambda_feat`) and the learning rate were both halved from their default values. After 20 epochs, the number of residual blocks in the generator was increased to 12. The training ran for a total of 100 epochs, taking approximately 4 hours to complete.

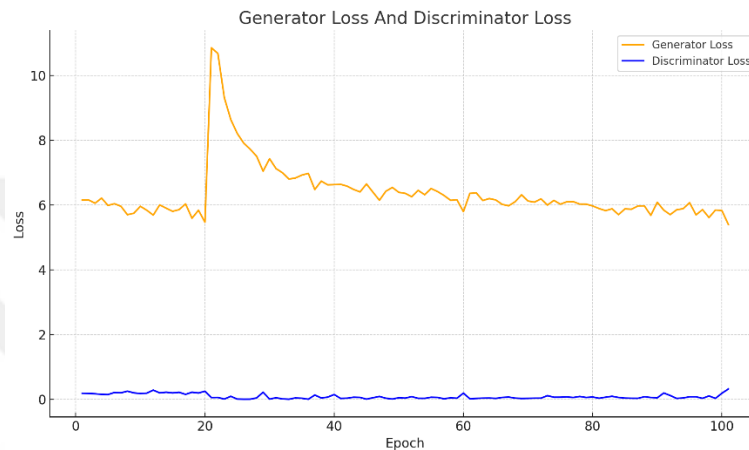


Figure 5.8 Change of Generator loss and Discriminator loss throughout the epochs during experiment 3

Loss Function and FID Score Analysis: During the first 20 epochs, the generator's loss decreased gradually. However, at epoch 20, when the number of residual blocks was increased to 12, the generator loss temporarily doubled (see Figure 5.8). As training progressed, the generator loss resumed its downward trend. A similar pattern was observed in the FID score, which experienced a spike due to the generator architecture change (see Figure 5.9). This modification initially reduced the quality of the generated images but ultimately led to improved performance as the training continued. The discriminator architecture remained unchanged, leading to stable loss values after an initial decline. Discriminator loss decreased for 20 epochs and balanced its value around 0, potentially overpowering the generator.

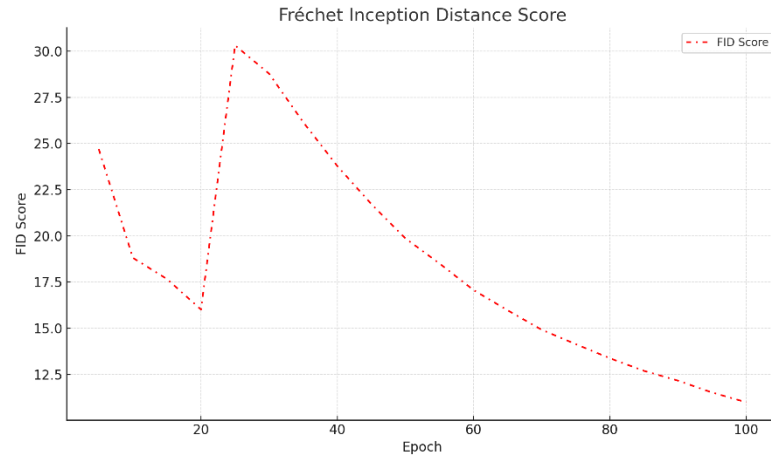


Figure 5.9 Change of Fréchet Inception Distance Score throughout the epochs during experiment 3



Figure 5.10 An example of the test results at epoch 100 during experiment 3

Output Quality: After increasing the number of residual blocks, the generator's performance showed significant improvement, with a noticeable reduction in the areas it had previously struggled to complete during training (see Figure 5.10). The model demonstrated enhanced pattern recognition, particularly in its ability to extend roads more effectively. However, challenges remained, as the generator continued to struggle with creating sharp edges and populating empty spaces with buildings, particularly during testing. While the results were better than the previous session, the overall improvement was relatively modest.



Figure 5.11 Generated image at epoch 100 during training

Evaluation and Observations: While no signs of overfitting were observed at epoch 100 in the previous session, the discriminator appeared too strong, potentially limiting the generator's ability to generalize effectively. Raising the number of residual blocks in the generator during this session had a positive impact, leading to better visual results.

Future training sessions might benefit from restarting the training from scratch with 12 residual blocks in the generator and increasing the number of discriminators to 3 for better balance.

5.3.2.4 Experiment 4

Training Setup: The training was initiated from scratch and lasted for 245 epochs. A batch size of 8 was used, with the `--niter_decay` value set to 50. Both the learning rate and the lambda values were initially reduced to half their default values. The model incorporated 12 residual blocks and 3 discriminators. At epoch 50, the learning rate was restored to its default value, followed by the lambda value at epoch 200.

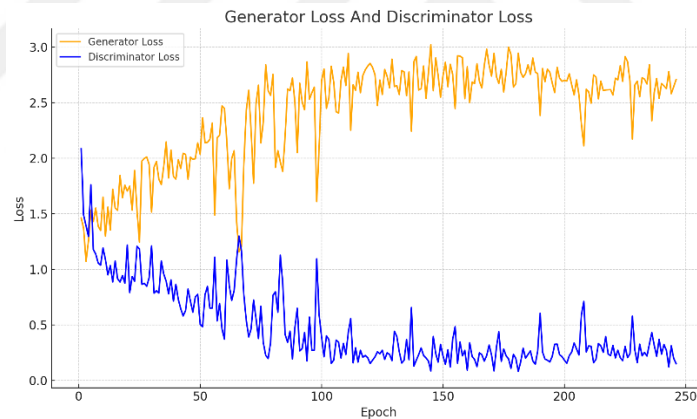


Figure 5.12 Change of Generator and Discriminator loss throughout the epochs during experiment 4

Loss Function and FID Score Analysis: After epoch 150, the loss values for the discriminator and generator reached a relatively balanced state (see Figure 5.12). However, the FID score continued to decrease, which justified extending the training beyond 200 epochs. Despite this, the decline in FID score significantly slowed after epoch 200 (see Figure 5.13).

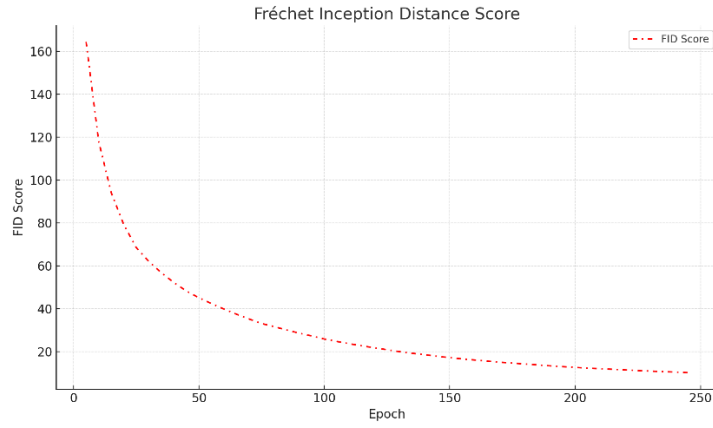


Figure 5.13 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 4

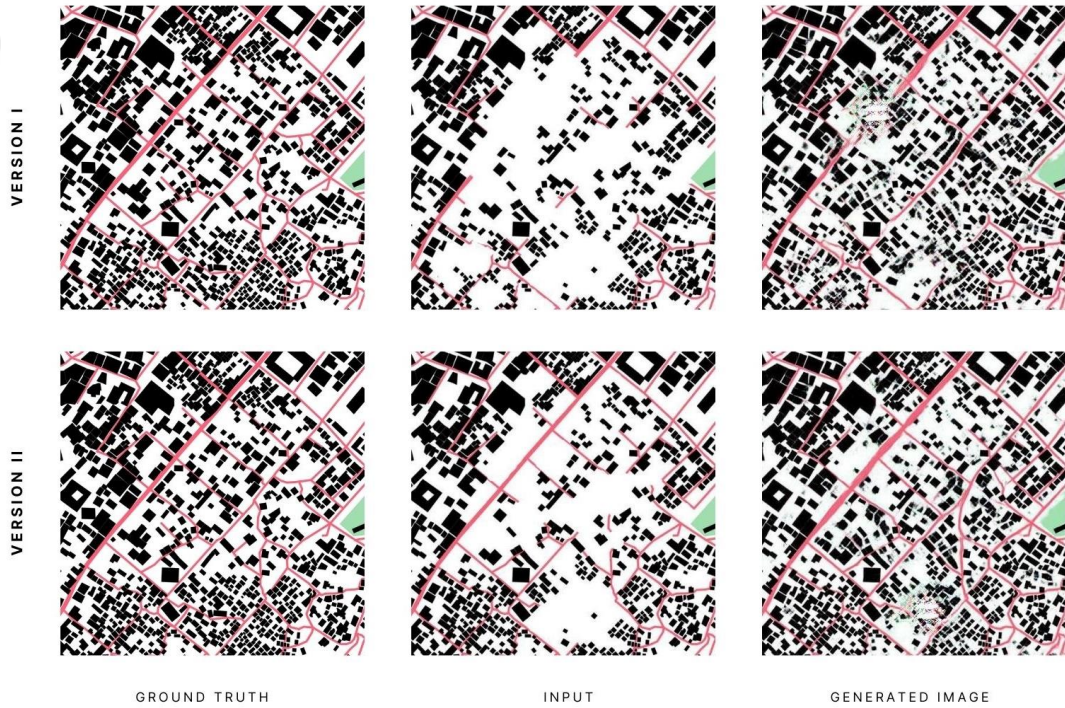


Figure 5.14 Examples of the test results at epoch 245 during Experiment 4

Output Quality: In this experiment, the model seems to add more buildings to the test image compared to previous models. While the density of structures in the generated areas did not match the ground truth, it showed a resemblance to the surrounding urban fabric. However, a different version of the same input image did not produce a similar result. This might be an indication of potential sensitivity to the size of the building blocks in the image. First version of the input, the generated image featured buildings with sharper edges and larger sizes, alongside smudged black pixels, highlighting a variability of quality in the model's outputs (see Figure 5.14).

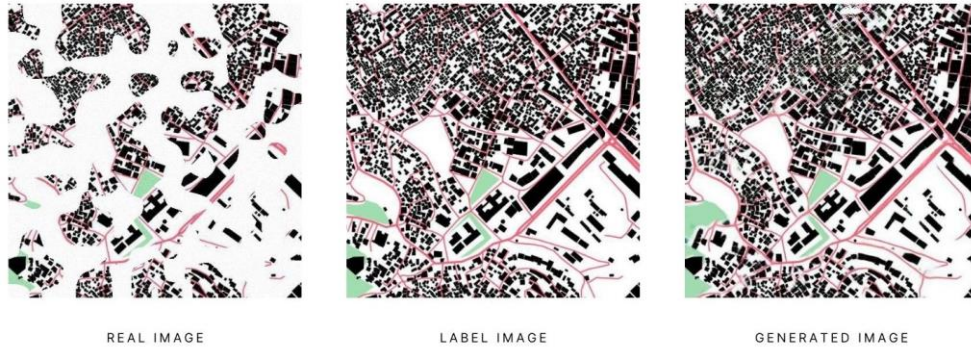


Figure 5.15 Generated image at epoch 239 during experiment 4

Evaluation and Observations: The architectural changes implemented in this experiment contributed to an improvement in image quality. Increasing the number of residual blocks and discriminators further in future experiments could enhance the results. While reducing the batch size and learning rate addressed the issue of sharp edges to some extent, the improvement was minimal. To tackle this limitation, increasing the number of local enhancers could be explored in future experiments.

5.3.2.5 Experiment 5

Dataset Augmentation: Before conducting this training experiment, expert insights were sought from Anna Boim, the lead author of "A Machine-Learning Approach to Urban Design Interventions in Non-Planned Settlements," to refine model performance strategies. These discussions highlighted the importance of dataset augmentation in improving training efficiency. To enhance variability and strengthen the model's learning process, the dataset was expanded by applying transformations such as mirroring and rotations of 90, 180, and 270 degrees. As a result, the total number of images increased to 12 150, forming the final dataset used in this experiment.

Training Setup: The training experiment was conducted using the hyperparameters from the previous experiment along with the newly augmented dataset. Initially, the training process was observed to be inefficient, with a single epoch requiring more than two hours to complete. To address this, modifications were applied to the training parameters. The batch size was increased to 12, and the number of data loading threads (`--nThreads`) was set to 16 to improve data processing speed. The number of residual blocks and discriminators was reverted to their default values. Both the learning rate and `--lambda_feat` was reduced to half of their default values. Additionally, the local enhancer

was set to 3. These adjustments resulted in a significant reduction in training time, bringing the duration of a single epoch down to approximately 10 minutes.

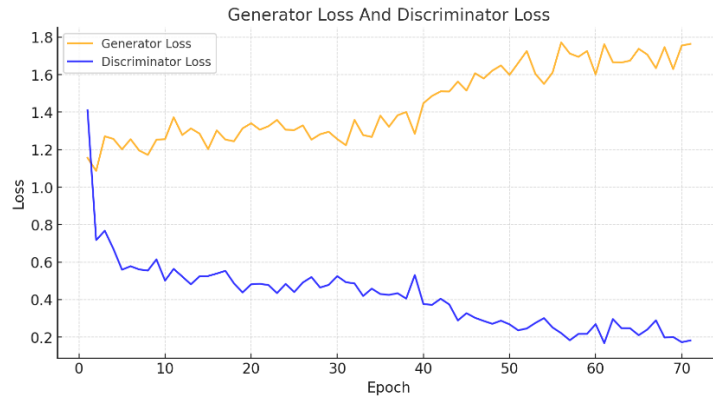


Figure 5.16 Change of Generator and Discriminator loss throughout the epochs during the experiment 5

Loss Function and FID Score Analysis: This training experiment was significantly shorter than previous ones. By epoch 60, the loss values for both the generator and discriminator reached a relative balance (see Figure 5.16). However, after epoch 40, the generator loss began to increase, suggesting that the generator struggled to deceive the discriminator effectively, indicating potential training instability. At epoch 70, the FID score dropped below 10 (see Figure 5.17). For comparison, in the previous experiment, loss values stabilized only after 150 epochs, and the FID score fell below 10 after 200 epochs. To prevent overfitting, training was finalized at epoch 70. Both the FID score and loss value graphs demonstrate that training became more efficient with the augmented dataset. The trends observed in both the loss and FID graphs confirm that the augmentation helped accelerate convergence and enhance training stability.

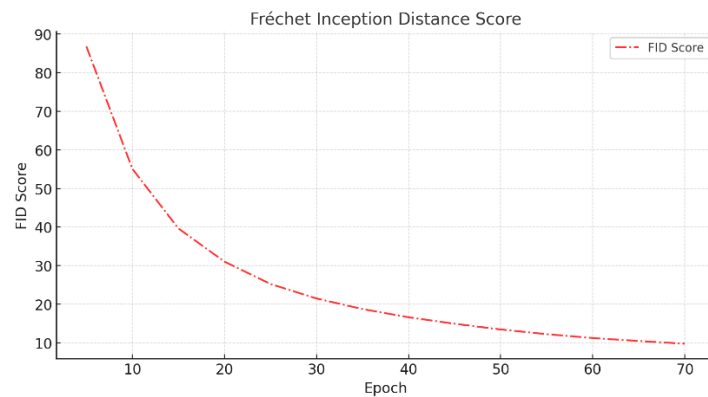


Figure 5.17 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 5

Output Quality: In this experiment, the model produced outputs with greater clarity compared to previous runs. Smudged pixels were significantly reduced, and overall edge quality improved, resulting in sharper and more defined structures. However, the generated structures did not fully match the density of the ground truth. While road continuity was highly successful, the model struggled with building morphology from an architectural perspective, failing to accurately replicate the spatial patterns of real urban fabric.

Evaluation and Observations: The model demonstrated a notable improvement in learning the geometric shapes of buildings, with enhanced edge clarity and a better ability to complete road networks (see Figure 5.18). These advancements can be attributed to the augmented dataset. However, one critical aspect that remained lacking was morphological accuracy. Despite the increase in edge sharpness and the significant reduction in smudged pixels, the model still struggled to fill empty spaces in alignment with pre-earthquake morphological patterns appropriately. This issue may be attributed to the low value of `--lambda_feat`, as this parameter determines the model's emphasis on structural features. Increasing `--lambda_feat` could potentially enhance the model's ability to learn and replicate morphological patterns more accurately.

However, the most probable cause of this issue is that the label maps do not fully align with the images in the test set. This discrepancy may indicate that the model may not be learning a structure that accurately reflects the patterns it needs to generate. Before the next experiment, additional tests can be conducted to verify this hypothesis, and the label map set can be adjusted to ensure better consistency with the test set.

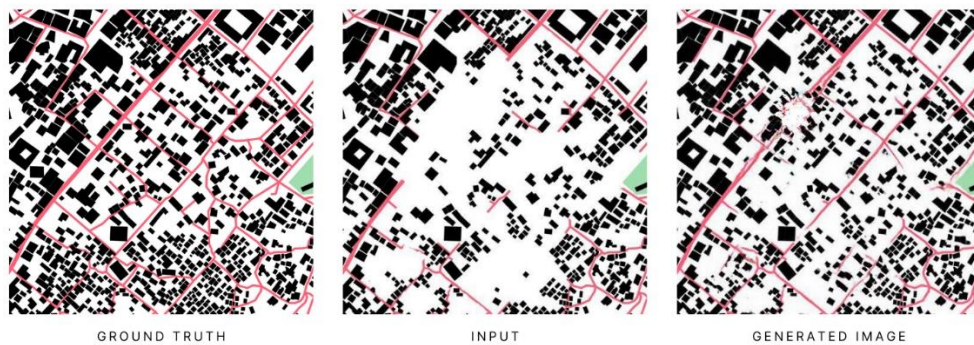


Figure 5.18 The test results of a 1:5000 scale input map at epoch 45 during experiment 5

5.3.2.6 Experiment 6

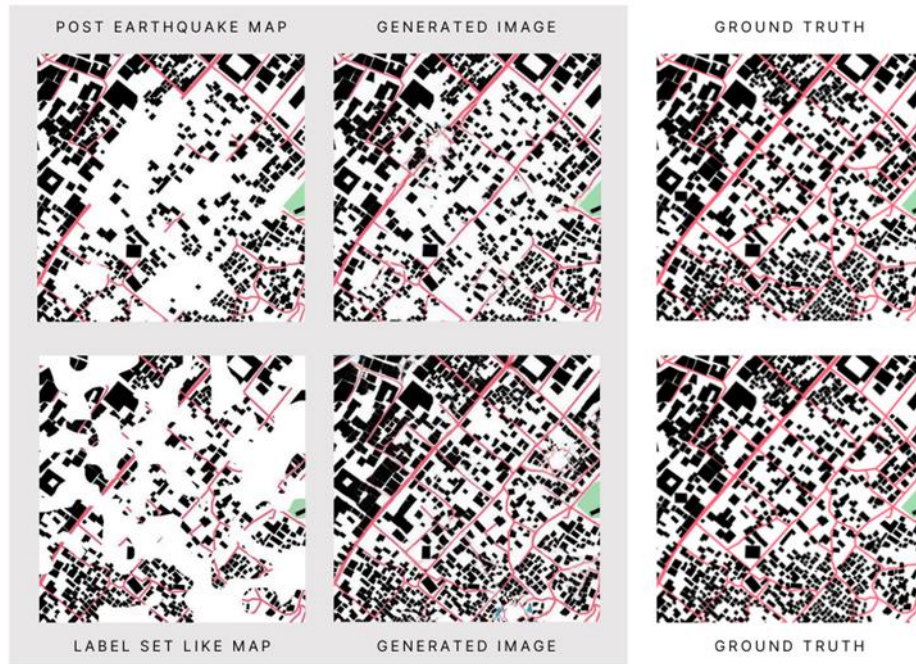


Figure 5.19 Difference of generation success between labelmap-like input and post-earthquake map input

Before starting the experiment, a validation test was conducted to check the consistency between the label map and test sets. As part of this process, the same masks used to create the label maps were applied to the pre-earthquake maps in the test set using the same Python script that generated the label maps. This masked test set was then fed into the model for completion. The model successfully generated the missing areas, demonstrating its ability to complete the urban fabric successfully.

However, the results revealed that while the model had learned the task of urban fabric completion, the label set did not sufficiently simulate the expected scenario in the test set. Consequently, the label set was restructured to better represent post-earthquake building loss. To achieve this, a two-step masking method was applied (see Figure 5.20). **Step 1: Random Removal of Buildings:** The first step is the random removal of buildings to simulate earthquake-induced destruction. To achieve this, buildings in the images were identified based on their RGB values, and connected components analysis was used to identify each independent building as a separate object.

To ensure a random removal process, each building was assigned a random value between 0 and 1, and a threshold-based decision mechanism was implemented. Buildings with values below the threshold were removed, while those above it were retained. By

adjusting the threshold value, the probability of buildings being removed could be controlled, offering a flexible approach to realistically modeling post-earthquake building loss.

Step 2: Random Erosion of Roads: The second phase involved the erosion of roads, which required a different approach than building removal. Unlike buildings, roads could not be separated into independent objects using connected components analysis, necessitating an alternative method.

To overcome this, a noise-based "web-like" pattern (512x512 in size) was generated, following the same principles used in label map creation. This pattern was selectively applied to road areas, identified based on their RGB values.

By using this mask, only roads were targeted for irregular removal, leaving buildings untouched. This method effectively simulated disruptions in the transportation network, producing a more accurate representation of post-earthquake urban damage.



Figure 5.20 Steps of creating the new label map

A total of **nine experiments** were conducted using this label map. In these trials, various combinations were tested by altering both the label map generation process and the model's training parameters. However, none of the experiments resulted in a meaningful improvement over the previously achieved outcomes. To enhance readability and avoid unnecessary repetition, all experiments related to this label map are presented together in a single consolidated section.

While evaluating loss function values and FID scores individually is useful for deciding whether training should be continued, it was concluded that analysis of these metrics would not significantly contribute to the overall report. Therefore, report metrics are addressed collectively and summarized briefly.

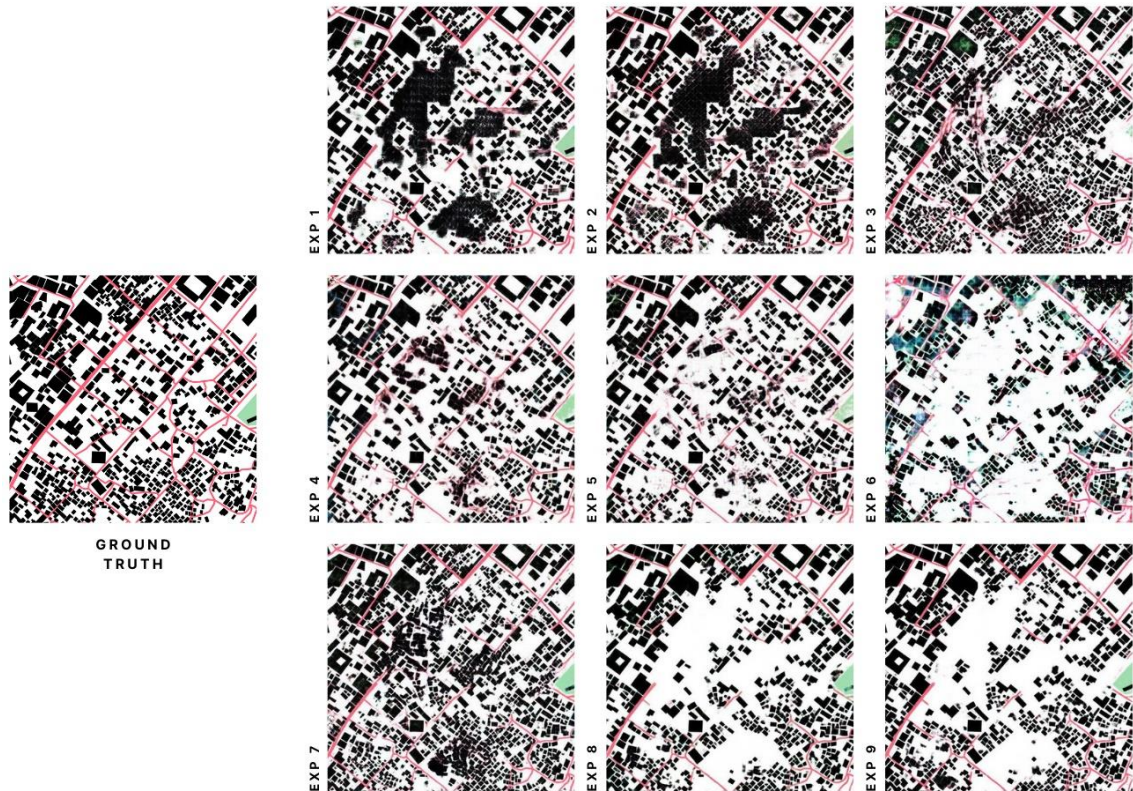


Figure 5.21 Outputs throughout the experiments

Training Setup: The first experiment was conducted using only the default training parameters to observe how the model responds to the newly introduced label map set. Following the unsatisfactory results, a series of experiments were carried out in which various training parameters were modified. The key findings derived from these trials are as follows:

- The use of **dropout** in the generator had a **positive effect** on model performance.
- Increasing the **number of filters** in the first convolutional layer of **the generator** (`--ngf`) **improved the quality** of the results.
- Conversely, reducing **the number of filters** in the first convolutional layer of the **discriminator** (`--ndf`) had a **negative effect** on performance.
- Increasing the **number of discriminators** led to **better results**, enhancing the model's ability to detect and guide generation.
- While batch normalization did not show a noticeable effect, **instance normalization** yielded **positive improvements**, particularly in image consistency.
- Setting the **lambda_feat** value to **higher levels** (e.g., 100 or 150) contributed **positively** to the structural integrity of the generated images.

- Lowering **the learning rate** from the default setting did not result in any significant positive or negative change in performance.
- Across all experiments, it was observed that training beyond 50 epochs did not yield additional improvements, suggesting that **a shorter training period** is sufficient when using this label map structure.
- High values of batch size caused memory issues due to the increased number of filters in the generator.

Output Quality: The output quality remained consistently low across the experiments. In the **first two experiments**, the model primarily generated checkered black pixels and red or black smudges in the missing areas, with no meaningful reconstruction. **The third experiment** showed slight improvement, it was able to add building-like forms; however, these were not limited to the masked regions and appeared scattered across the entire background (white areas). Additionally, the green area on the right edge of the image disappeared, and some of the generated buildings contained misplaced green pixels, indicating color inconsistencies and spatial inaccuracies in the output.

In the output of **the fourth experiment**, the number of additions decreased, but the generated forms still did not resemble proper buildings. The model continued to struggle with extending existing roads. Additionally, the green area appeared deformed, and scattered green and blue pixels were visible between existing buildings, indicating a lack of spatial coherence and further issues with color placement. The same issues are present at the **fifth experiment** with no improvement.

In the **sixth experiment**, the results took an unexpected turn. The model generated blue and green pixels around black areas but failed to add any buildings or continue existing roads. Once again, the green area in the corner disappeared. This recurring issue suggests a possible connection between the disappearance of green spaces and the appearance of random green or blue pixels in unrelated parts of the image, indicating a breakdown in the model's ability to interpret and reconstruct specific spatial elements correctly.

The **seventh experiment** produced results similar to those of experiment three. However, unlike in earlier trials, the green area remained intact, and the random placement of green pixels was minimal. This suggests a slight improvement in the model's handling of green space and color consistency, although overall performance remained limited.

Experiments eight and nine produced similar outputs. In both cases, the models were unable to properly fill in the missing urban fabric. Instead, they added small buildings and smudged black pixels around existing structures, and failed to continue the road networks. However, the clarity of the additions was noticeably higher compared to the earlier experiments, indicating a slight improvement in visual sharpness despite the persistent structural issues.

Evaluation and Observations: None of the experiments yielded promising outcomes. In fact, the results were noticeably worse than those from previous trials. None of the models were able to learn how to continue road segments, let alone connect them into a coherent network. This complete lack of spatial recognition indicates that the new label map is insufficient for the goals of this research. In contrast, earlier experiments using the previous label map showed at least some ability to continue roads and establish network connections. These latest results reflect a clear regression in model performance.

5.3.2.7 Experiment 7

Label map: Following the unsatisfactory results of the previous label map, different label map preparation techniques used in GAN-based inpainting studies were re-evaluated through a review of the literature (J. Li et al., 2019; Pathak et al., 2016; Yeh et al., 2017). These reviews revealed that, in successful examples, the removed regions in the input data were not blended into the background but were instead marked with distinct colors, such as white in some cases and black in others, ensuring clear visual separation from the surrounding context.

Based on these findings, and taking into account that the first label map produced more successful results compared to the second version, it was decided to apply the mask used in the first label map using the magenta color (RGB: 255, 0, 255). Similarly, the damaged or removed areas in the input images were also marked with this color (see Figure 5.22).

This approach allowed the missing regions to be clearly separated from the background in visual terms. As a result, the model was directly guided toward which areas needed to be completed, eliminating the need for it to identify missing regions on its own. The model thus shifted its focus entirely to completing predefined gaps, detaching from the task of detecting spatial loss. **This shift signifies a meaningful change in the model's task definition during training.**

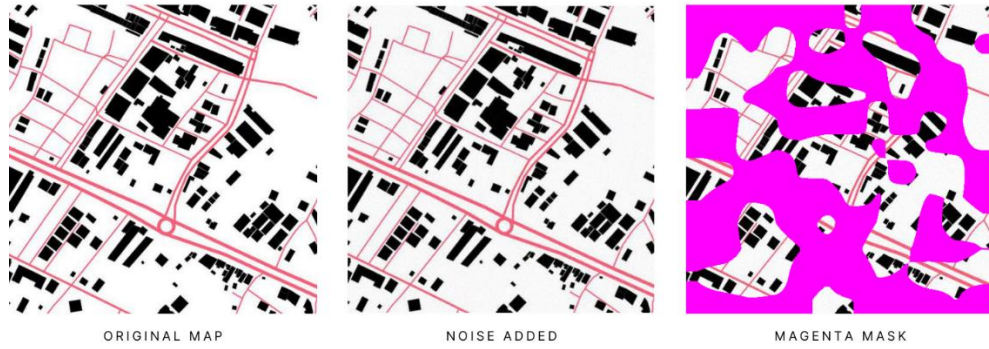


Figure 5.22 Steps of creating the magenta label map

Training Setup: This training experiment lasted for 45 epochs. The generator used 128 filters in the first convolutional layer and included dropout for regularization. Instance normalization was applied instead of batch normalization. The batch size was set to 8, and the number of data-loading threads was 4. The model employed 3 discriminators. The lambda feature loss was set to 100, and the learning rate was kept at its default value. The learning rate remained constant for the first 15 epochs and then decayed over the following 35 epochs.

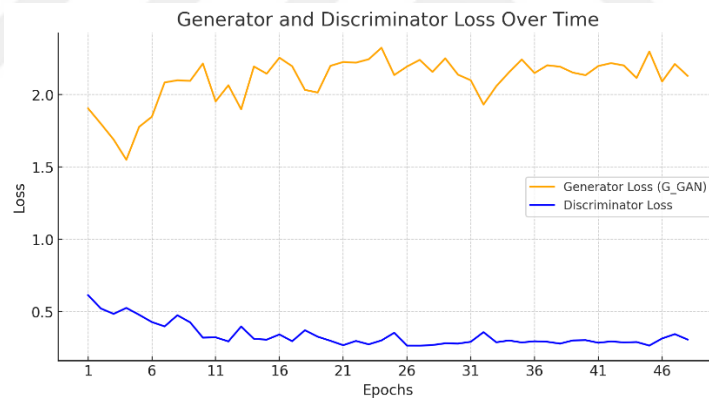


Figure 5.23 Change of Generator and Discriminator loss throughout the epochs during the experiment 7

Loss Function and FID Score Analysis: The generator and discriminator appeared to reach a balance as early as 15 epochs (see Figure 5.23). Additionally, the model started with a relatively low FID score, which continued to decrease rapidly over the first 20 epochs, approaching a value of 10 by epoch 45 (see Figure 5.24). This consistent improvement led to the decision to stop training at that point, as the results indicated a successful and stable learning process.

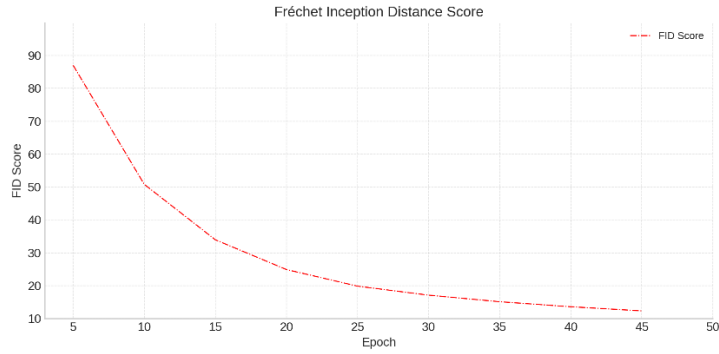


Figure 5.24 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 7

Output Quality: The output quality significantly improved compared to the results of earlier experiments. Edge sharpness increased noticeably, and overall visual clarity is much higher. The colors produced closely match the ground truth, with only minimal occurrences of out-of-place color pixels.

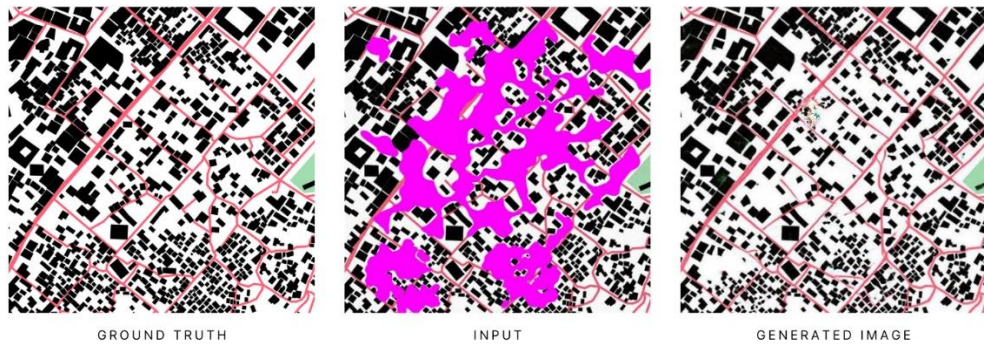


Figure 5.25 The test results of a 1:5000 scale input map at epoch 45 during experiment 7

Evaluation and Observations: The models appear to be highly successful at continuing roads and connecting networks, demonstrating a notable degree of creativity in the process. This creative capacity remains consistent across different input scales (see Figure 5.25). However, while the overall spatial coherence remains strong, the visual clarity of outputs tends to decline at lower input scales (see Figure 5.26).

Explicitly marking the missing areas with magenta appears to have significantly aided the model's performance. This experiment marked the first time such a high level of color consistency was achieved in the generated images. However, the model occasionally struggled with distinguishing between blue and green, sometimes blending the two. This issue is likely related to the dataset creation process. Blue and green appear less frequently in the dataset, and their visual similarity may have led the model to confuse them, making it more difficult to learn distinct patterns associated with these specific elements.

The loss graph and FID score indicate a successful training process, and the visual assessment of the generated images is equally satisfactory. The combination of dataset size, label map style, and hyperparameter settings appears to have achieved an effective balance, resulting in improved model performance and output quality.

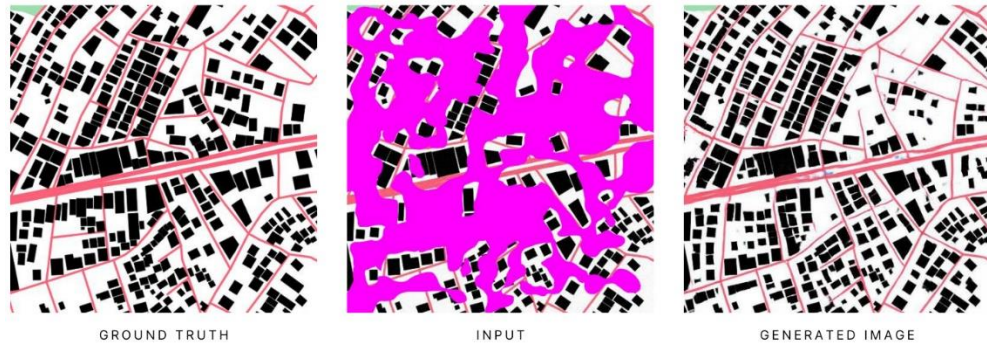


Figure 5.26 The test results of a 1:3000 scale input map at epoch 45 during experiment 7

5.3.3. Experiments of Scale 1:3000

A single experiment was conducted at the 1:3000 scale. Since the results obtained from this trial were deemed satisfactory, no additional experiments were considered necessary for this scale. The training process employed the same training parameters that had previously yielded successful results at the 1:5000 scale.

The label map used in this experiment combined magenta-colored masking with a noise-based random deletion method, allowing for clear separation of missing areas. The parameters used in generating the label map were as follows:

- Scale: 40
- Connectivity: 11
- Noise standard deviation: 10
- Threshold: 0.3
- Erosion iterations: 1

Trail Experiment:

No issues were encountered during the trial run, and the experiments were initiated subsequently.

5.3.3.1 Experiment 1

Training Setup: The training lasted for 51 epochs, with a batch size of 8 and 4 data-loading threads. Dropout was applied in the generator to help prevent overfitting. The first convolutional layer of the generator used 128 filters. Instance normalization was used throughout the network. The lambda feature loss was set to 100, and the learning rate was kept at its default value. The model was trained using 3 discriminators. The learning rate remained constant for the first 15 epochs and then linearly decayed over the following 40 epochs.

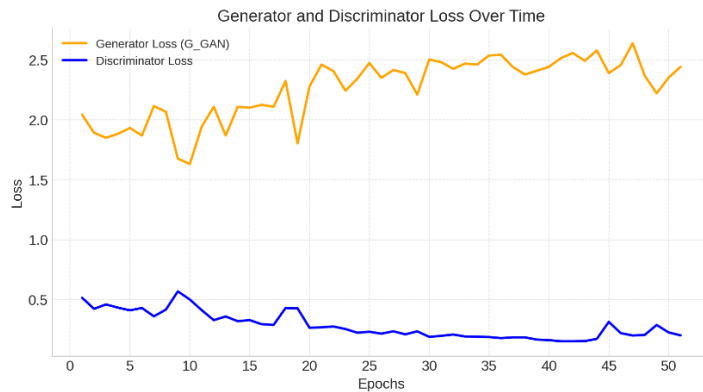


Figure 5.27 Change of Generator and Discriminator loss throughout the epochs during experiment 1

Loss Function and FID Score Analysis: The loss values reached a balanced state after epoch 30 (see Figure 5.27). Although the GAN loss remained slightly higher compared to previous experiments, the decrease in the FID score indicates a successful learning process. The FID score dropped rapidly during the first 15 epochs, then continued to decline steadily (see Figure 5.28). However, after epoch 45, the rate of decrease became very slow, which led to the decision to stop training at epoch 51.

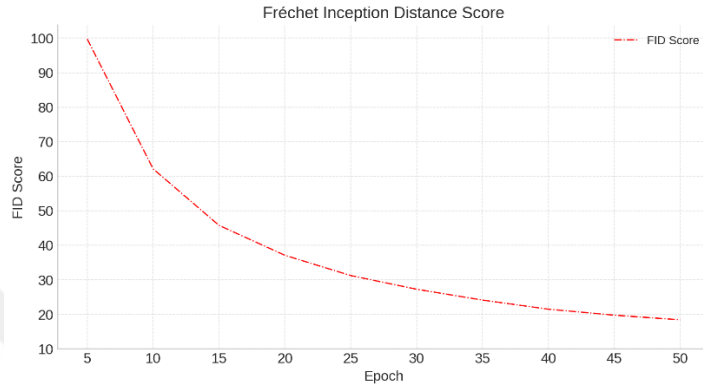


Figure 5.28 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 1

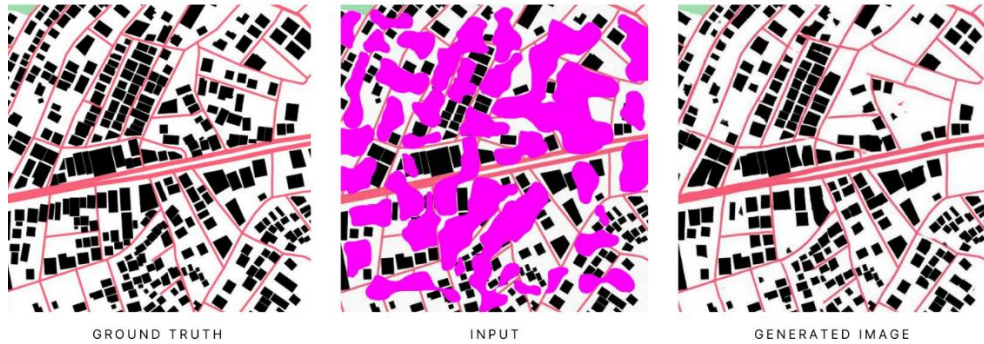


Figure 5.29 The test results of a 1:3000 scale input map at epoch 50 during experiment 1

Output Quality: The overall clarity of the generated images is high. The edge definition is satisfactory, although some buildings appear slightly concave (see Figure 5.29). The model successfully reproduces colors that closely match the ground truth, contributing to a visually coherent output. Quality is consistent throughout multiple images.

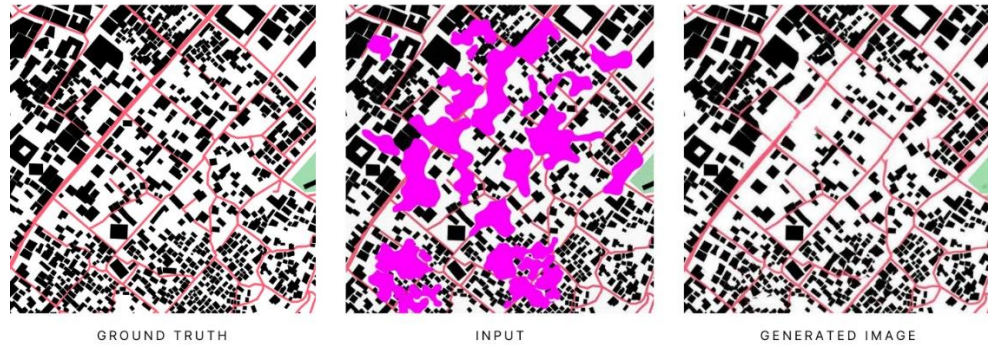


Figure 5.30 The test results of a 1:5000 scale input map at epoch 50 during experiment 1

Evaluation and Observations: The model applied inpainting to areas marked with magenta, generating distinct and well-defined structures within these regions. However, occasional losses were also observed in unmarked areas, indicating that the model's focus on completion may influence the overall structure of the map beyond the explicitly targeted zones (see Figure 5.30).

In the inpainting process, the model demonstrated success in continuing road networks, but it struggled to connect roads across large empty spaces, resulting in some roads ending as dead-ends. On the other hand, the model appeared to develop a strategy for building placement based on surrounding environmental data. For instance, when the area surrounding the magenta-marked region had high building density, the model produced a correspondingly dense configuration. In contrast, when building density was low, it added fewer buildings or none at all. This suggests that the model generates context-sensitive outputs, responding to the spatial logic of its surroundings.

Moreover, the footprint of the newly added buildings often resembled those of nearby structures, indicating an attempt to maintain local morphological consistency. These buildings were generally placed in novel yet structurally coherent arrangements that showed similarity to the ground truth. This generative approach yielded successful results across input maps at varying scales (see Figure 5.31).

However, at the 1:5000 scale, particularly in areas with organic urban fabric or regions characterized by small building footprints, the model's ability to maintain morphological coherence declined. In such cases, it occasionally generated buildings with slightly larger footprints than their surroundings, and minor concavities were observed in the building outlines.

Despite these limitations, the model performed well in terms of visual clarity and edge sharpness in the newly generated areas. Overall, it demonstrated strong generative performance, both in structural alignment and visual quality.

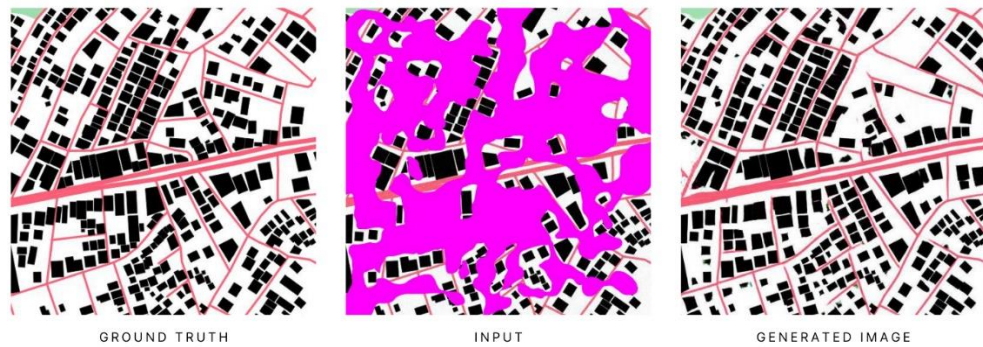


Figure 5.31 The test results of an alternative 1:3000 scale input map at epoch 50 during experiment 1

5.3.4. Experiments of Scale 1:2000

One experiment was conducted with the 1:2000 scale dataset. The training process employed the same training parameters that had previously resulted in successful outputs with the 1:3000 scale.

The label map used in this experiment combined magenta-colored masking with a noise-based random deletion method, allowing for clear separation of missing areas. The parameters used in generating the label map were as follows:

- Scale: 40
- Connectivity: 11
- Noise standard deviation: 10
- Threshold: 0.3
- Erosion iterations: 1

Trail Experiment:

An issue arose due to a mismatch in the image filenames. Once the problem was resolved, the code ran without any further issues, and the experiments were able to begin.

5.3.4.1 Experiment 1

Training Setup: The training spanned 50 epochs, using a batch size of 8 and 4 data-loading threads. To reduce the risk of overfitting, dropout was applied in the generator. The generator's first convolutional layer included 128 filters, and instance normalization was used throughout the network. The lambda feature loss was set to 100, and the learning rate was left at its default setting. The model was trained with 3 discriminators. The learning rate remained unchanged for the initial 15 epochs, followed by a linear decay over the subsequent 35 epochs.

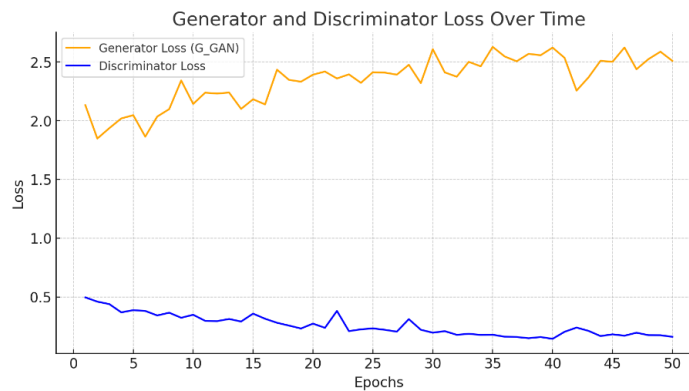


Figure 5.32 Change of Generator and Discriminator loss throughout the epochs during experiment 1

Loss Function and FID Score Analysis: The loss values reached equilibrium after epoch 35 (see Figure 5.32). Overall, the generator loss appeared similar to that observed in the 1:3000 scale experiment. The decrease in FID score slowed significantly after epoch 35, and by the end of epoch 50, it had dropped below 25—slightly higher than in previous experiments (see Figure 5.33).

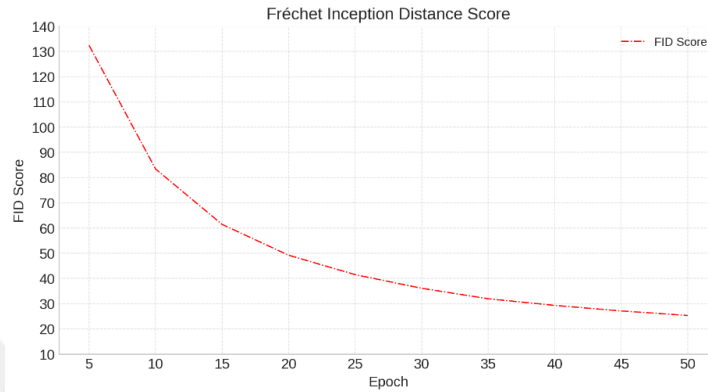


Figure 5.33 Change of Fréchet Inception Distance Score throughout the epochs during Experiment 1

Output Quality: The model's overall output quality can be considered high. The edge definition is satisfactory and consistent with earlier experiments, and the model successfully reproduces colors that closely match the ground truth (see Figure 5.34). However, minor issues were observed, such as the presence of a few misplaced green or blue pixels.

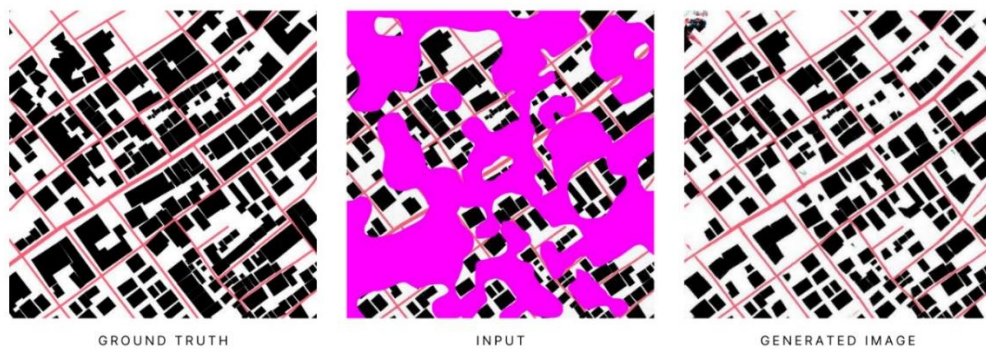


Figure 5.34 The test results of a 1:2000 scale input map at epoch 45 during experiment 1

Evaluation and Observations: The model does not produce satisfactory results on input data at varying scales. In particular, it has been observed that the model struggles to connect road networks in larger-scale inputs (see Figure 5.35). This issue is expected, given that the training dataset was based on a smaller scale, limiting the model's ability to effectively generate coherent road networks. Similarly, inadequacies have been observed in the generation of buildings with small footprints.

In contrast, the model performs more successfully at the scale it was trained on (1:2000), producing consistent building footprints and continuous road networks. However, when operating outside this scale, especially with larger-scale input data, the model tends to generate buildings with larger footprints compared to surrounding structures. This indicates a limited ability to perceive and interpret surrounding spatial patterns.

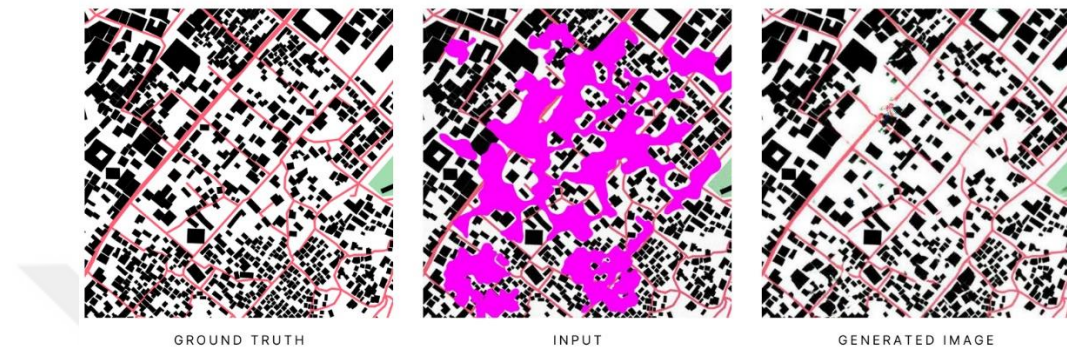


Figure 5.35 The test results of a 1:5000 scale input map at epoch 45 during experiment 1

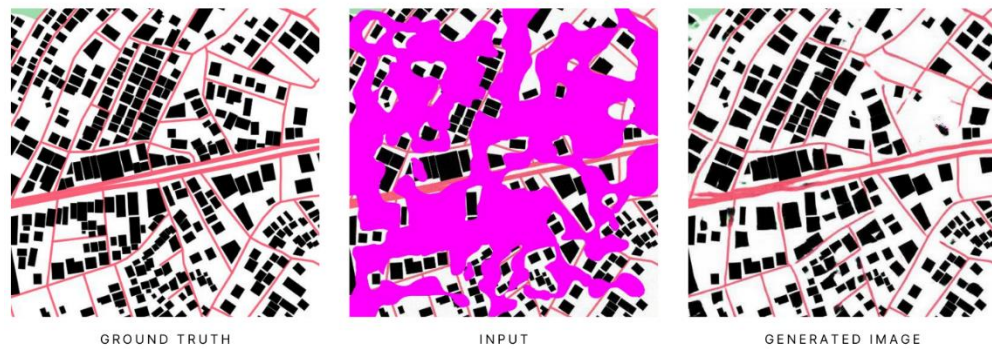


Figure 5.36 The test results of a 1:3000 scale input map at epoch 45 during experiment 1

5.4 Evaluation of the Model and Generated Images

To assess the performance of the model and the quality of its outputs, a systematic evaluation was carried out using both visual inspection and quantitative metrics. This section presents the results of that evaluation, focusing on two key measures: the Structural Similarity Index Measure (SSIM) and the Fréchet Inception Distance (FID) score. SSIM was used to evaluate the visual accuracy and structural integrity of the generated images, while FID provided insight into the model’s ability to produce realistic and coherent outputs. Before evaluating the model’s performance, it is important to highlight that one of the key factors influencing the success of the training process was the design of the **label maps**. Three different label maps were used throughout the experiments (see Figure 5.37). Two were generated using a **procedural noise-based masking** method, while the third relied on a **stochastic deletion** approach.

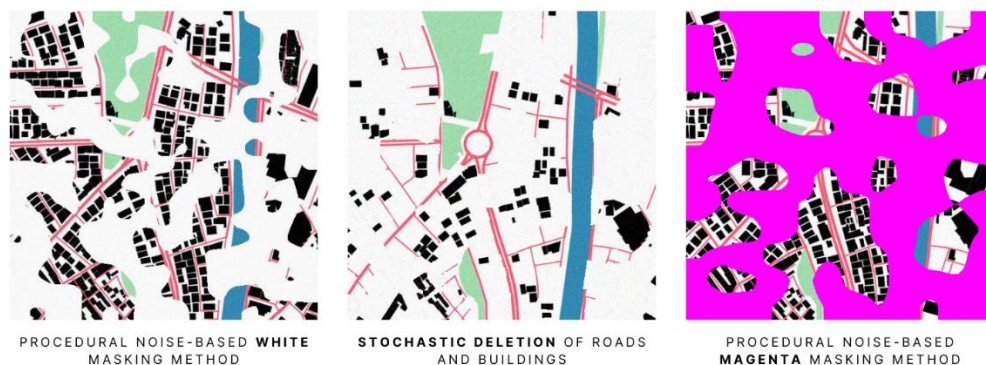


Figure 5.37 Types of label map generation methods

Chronologically, the first label map used was a procedural noise-based white mask. Models trained with this version demonstrated some ability to connect road networks; however, they struggled with edge sharpness and visual clarity and failed to generate clearly defined, rectangular building forms.

Subsequently, based on the assumption that there was a mismatch between the label map and the input set, stochastic deletion-type label maps were created. However, the experiments conducted using this label map consistently failed to continue roads or connect road networks. Additionally, nearly all models struggled to generate buildings with sharp edges and were unable to effectively inpaint the missing areas in the input images.

Lastly, a magenta-colored Gaussian blur-based mask was used to create the third label map, which yielded the most successful results among the three. Experiments

conducted with this version consistently produced higher quality images, showing notable improvements in both clarity and edge sharpness. In contrast to earlier label maps, where the output colors appeared faded or muted, this version resulted in color reproduction that closely matched the input images.

The key difference with this label map is that the magenta mask does not blend with the background or with any of the dataset's existing colors. As a result, the model could clearly identify which areas were missing in both the label maps and the input set. This explicit visual cue is believed to be one of the main reasons for the improved training outcomes. By removing the need to infer missing regions, the model was able to focus more effectively on learning color relationships, edge definition, and spatial patterns.

Fréchet Inception Distance Scores were calculated once every 5 epochs during training. When different scale experiment scores are compared to one another, they follow very similar curves (see Figure 5.38). This indicates that all the scales have a similar learning pace. But comparison graphs still show a difference in level in the model's ability to produce realistic images.

The model that appears to be the most successful in generating the most realistic images is the one trained at the 1:5000 scale, followed by the experiments conducted at the 1:3000 scale. This outcome suggests a clear trend: as the scale of the training dataset increases, the model tends to produce more realistic and coherent outputs. The larger-scale datasets likely allow the model to better understand broader spatial patterns and contextual relationships, which in turn enhances the quality and believability of the generated images.

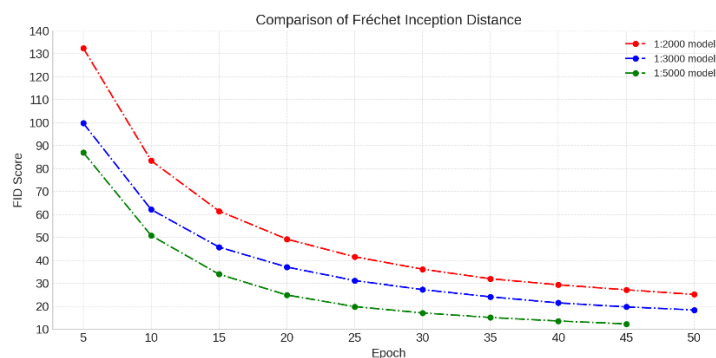


Figure 5.38 Comparison of Fréchet Inception Distance Scores of different experiments.

This trend is also evident when visually inspecting the images generated during training. As the dataset scale increases, the outputs display greater spatial coherence, clearer structural details, and a more realistic overall appearance. The images produced

by the 1:5000, in particular, exhibit a higher level of refinement and consistency compared to those generated at smaller scales, reinforcing the correlation between dataset scale and output quality.

When comparing the **Structural Similarity Index Measure (SSIM)** across different models, an inverse trend becomes apparent: the model trained on the 1:5000 dataset consistently achieves lower SSIM values compared to those trained on smaller-scale datasets (see Figure 5.39). However, when visually examining the outputs of the 1:5000 model, the generated images, particularly the novel additions, appear quite successful. In light of this, the lower SSIM scores may be attributed not to poor performance but to the model producing more divergent or creative generations that deviate from the ground truth while still maintaining contextual plausibility.

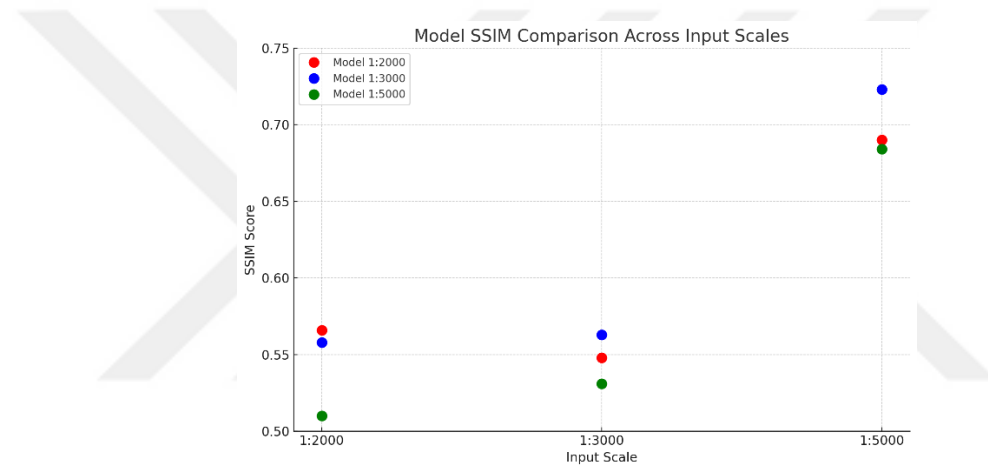


Figure 5.39 Comparison of Structural Similarity Index Measures across input scales of different models.

Comparing SSIM values between the generated images of the same model does not provide a clear understanding of which input scale the model performed better on. This is primarily due to the varying amount of masked area in each input image, as the masking is applied based on earthquake damage. In larger-scale inputs, the spatial frame is wider, which causes the damaged areas to appear proportionally smaller. As a result, the masked regions occupy less of the overall image, leading to naturally higher SSIM scores that may not accurately reflect the model's true performance across different scales. A visual inspection of the images generated by the same model across different input scales proves to be a more reliable evaluation method.

SSIM RESULT OF THE MODEL TRAINED ON 1:2000 SCALE MAPS	SCALE	GROUND TRUTH	GENERATED IMAGE	SSIM
	1:2000			0.566
	1:3000			0.548
	1:5000			0.690

Figure 5.40 SSIM scores of the generated images of different scale inputs- model 1:2000

Although the model trained at the 1:2000 scale did not achieve the highest SSIM score when tested with 1:2000 inputs, it still produced the most contextually appropriate and natural-looking urban fabric(see Figure 5.40). Visual analysis shows that, despite lower numerical similarity, the model better captured the spatial logic and continuity of the urban environment at this scale. As a result, its outputs blended more smoothly with the surrounding areas, indicating a stronger understanding of the real-world spatial relationships beyond what the SSIM metric alone could measure.

SSIM RESULT OF THE MODEL TRAINED ON 1:3000 SCALE MAPS	SCALE	GROUND TRUTH	GENERATED IMAGE	SSIM
	1:2000			0.558
	1:3000			0.563
	1:5000			0.723

Figure 5.41 SSIM scores of the generated images of different scale inputs- model 1:3000

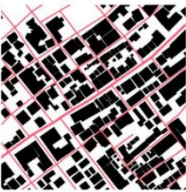
SSIM RESULT OF THE MODEL TRAINED ON 1:5000 SCALE MAPS			
SCALE	GROUND TRUTH	GENERATED IMAGE	SSIM
1:2000			0.510
1:3000			0.531
1:5000			0.684

Figure 5.42 SSIM scores of the generated images of different scale inputs- model 1:5000

The model trained on 1:3000 images demonstrates the highest visual success when tested with 1:3000 input data (see Figure 5.41). The footprints of the added buildings appear smaller than those generated by the 1:2000 model but larger than those from the 1:5000 model. This suggests that the model does not determine building footprint size solely based on the input image, but rather operates within a range of sizes learned from its training dataset.

A similar pattern can be observed in the road network generation of the model trained on the 1:5000 dataset. When applied to 1:2000 and 1:3000 input images, the generated roads appear thinner than expected, likely because, at the 1:5000 training scale, roads are represented with less visual thickness (see Figure 5.42). This suggests that the model internalizes scale-specific characteristics during training, which then influences how it reconstructs elements across different input scales. As a result, each model performs best at the scale it was trained on, indicating that scale consistency between training and input data plays a key role in generation accuracy.

However, when considering creative performance, the 1:5000 model exhibits the highest level of divergence from the ground truth while still preserving the key characteristics of spatial organization. It appears to perform better not only in continuing road segments but also in successfully connecting the overall road network. This is likely due to the dataset containing larger frames of road networks, which may have enabled the model to better understand broader connectivity patterns during training.

When it comes to the geometry and edge sharpness of the buildings, the models trained on the 1:2000 and 1:3000 datasets appear to be more successful. This is likely due to the smaller scales providing more detailed visual information. Increasing the resolution of the 1:5000 dataset could help improve edge sharpness and enhance the geometric clarity of the generated buildings.

Table 5.2 SSIM Scores of The Generated Images

	INPUT 1:2000	INPUT 1:3000	INPUT 1:5000
MODEL 1:2000	0.566	0.548	0.690
MODEL 1:3000	0.558	0.563	0.723
MODEL 1:5000	0.510	0.531	0.684



Figure 5.43 Comparison of the generated images with different models

The overall results of the experiments demonstrate that GANs architecture holds significant potential for learning and generating spatial data in a meaningful and context-sensitive way. Across various trials, it became evident that even complex urban spatial relationships can be understood and reproduced by the model with notable accuracy (see Figure 5.43). These outcomes affirm both the capabilities of GANs and the effectiveness of the Pix2PixHD model, which is an accessible and open-source architecture.

Despite the computational expense of the training process, the use of cloud-based platforms such as Google Colab has proven to be a practical solution. By offering access to GPU resources, Google Colab enables deep learning experiments to be conducted without the need for high-end local hardware. This accessibility highlights the potential for such technologies to become integral tools in future design workflows.



Chapter 6

Conclusions and Future Prospects

6.1 Conclusions

In the context of large-scale urban loss, the preservation of spatial continuity emerges as a fundamental challenge for ensuring the survival of collective urban memory. Beyond the physical reconstruction of cities, safeguarding the subtle patterns, relationships, and identities embedded in the urban fabric is essential for restoring a sense of belonging and continuity. This research examines the potential of generative deep learning models as tools to support this process, proposing a method where the spatial characteristics of a city are learned and reinterpreted through artificial intelligence. By utilizing accessible resources and platforms, the study demonstrates that sophisticated urban regeneration strategies can be developed without the need for high-end technical infrastructures. Focusing on the intersection of technology, design, and cultural memory, the research aims to expand the possibilities for post-disaster urban recovery. It also highlights how data-driven approaches can democratize the regeneration process, making memory-sensitive design methods available to a broader community of practitioners and stakeholders.

The primary aim of this research was to explore whether a generative tool can maintain spatial continuity and support the regeneration of urban memory following large-scale urban loss. The experiments demonstrate that a model trained on the spatial characteristics of a city can contribute meaningfully to memory-preserving design decision processes. Since urban space serves as the stage for all aspects of urban life, preserving its continuity plays a crucial role in maintaining the collective memory embedded within it. Therefore, protecting spatial patterns and relationships is not only a design concern but also a cultural and social imperative in post-loss urban regeneration.

This research also demonstrates that the process of training a deep neural network on a city's spatial organization can be implemented with relative ease and minimal

resources. Both the creation of the dataset and the training of the model require only basic facilities, especially with the support of accessible platforms like Google Colab. This shows that such advanced generative approaches are not limited to high-end research environments and can be utilized by a broader range of researchers, designers, and local governments.

The overall results of the experiments show that the GAN architecture is capable of learning the spatial organization characteristics of a city and regenerating urban fabric in a way that aligns with the city's contextual identity. The findings also highlight the importance of the dataset. In this research, relatively simple datasets and label sets were used, yet the GAN model was still able to learn and reproduce a significant amount of spatial patterns. This demonstrates both the potential of GANs in urban design applications and the value of even basic data when carefully prepared and structured.

Pix2PixHD, originally designed as an open-source tool for translating segmentation maps into photorealistic images, has proven its potential in generating urban patterns. Its accessibility not only enables the development of custom tools but also allows for experimentation with different types of datasets, making it a valuable resource for researchers and designers alike. Furthermore, its success in this context suggests that other types of GAN architectures may also hold significant potential in the field of generative urban studies, opening new directions for data-driven urban memory preservation research.

The findings of this research indicate a clear relationship between the scale of the training data and the model's ability to learn spatial patterns. In particular, the model trained on 1:5000 scale maps not only performed well at its own scale but also demonstrated the highest level of cross-scale performance compared to models trained at other scales. In the case of Hatay city center, this model was the most successful in generating realistic and contextually appropriate urban fabric across different input scales, highlighting the advantage of training at broader scales for generalization.

There appears to be a connection between the scale of the training data and the scale of the input during inference. Models trained on larger-scale data tend to perform well when generating outputs at their own scale or smaller. However, when presented with input at a larger scale than they were trained on, these models often struggle to generate precise spatial elements, such as road networks or building footprints. Despite this, they still show an understanding of broader spatial relationships, such as how buildings relate to one another, indicating partial generalization beyond the training scale.

Another important factor in terms of generalization is the proportion of specific urban fabrics within the overall area of the city. Since the dataset is created directly from the city map, if a particular fabric type occupies only a small portion of the urban area, it results in fewer data points representing that pattern in the dataset. This limited representation affects the model's ability to learn the spatial characteristics of that specific fabric. The impact of this imbalance is evident across all training scales, as the model consistently struggles the most with generating organic urban fabric, which typically covers a smaller portion of the city.

When it comes to quantitative evaluation methods, FID and SSIM scores are reliable measures of performance. However, for assessing success in urban pattern generation, visual inspection remains the most important and insightful method. It allows for a deeper understanding of spatial coherence, contextual appropriateness, and the overall realism of the generated urban fabric - factors that may not be fully captured by numerical metrics alone.

6.2 Societal Impact and Contribution to Global Sustainability

This research directly aligns with the objectives outlined in the United Nations Sustainable Development Goals (SDGs), with a particular emphasis on SDG 11: Sustainable Cities and Communities. By investigating the application of artificial intelligence in urban design, the study advances sustainable urbanization strategies that prioritize the preservation of existing urban fabric, the responsible management of land resources, and the reinforcement of cultural identity. The integration of generative adversarial networks (GANs) into urban planning processes introduces innovative methodologies that support the development of more resilient, adaptable, and context-sensitive urban areas. This technological approach not only aids in mitigating the adverse effects of rapid urbanization but also fosters the creation of inclusive, safe, and sustainable communities. Furthermore, by enabling more intelligent use of available resources and promoting efficient design strategies, AI-driven methods contribute to reducing the environmental footprint of urban development. In doing so, the research highlights the potential for generative technologies to serve as powerful tools for both cultural preservation and sustainable growth, reinforcing the essential link between technological innovation and global sustainability efforts.

6.3 Future Prospects

As the field of deep learning continues to advance at an accelerated pace, its intersection with architectural and urban design presents increasingly rich and promising opportunities for innovation. The findings of this research demonstrate that Generative Adversarial Networks (GANs) are capable not only of generating plausible urban patterns but also of signaling broader possibilities for reshaping design methodologies in the context of urban regeneration, memory preservation, and sustainable development. Beyond their immediate application in producing spatial patterns, GANs offer a powerful means to explore, reimagine, and extend the principles that have historically underpinned urban form and morphology.

Future research could focus on tailoring GAN architectures specifically to the unique challenges of urban design. This might include developing models better suited to recognize complex spatial hierarchies, land use distributions, and multi-scale relationships inherent in real cities. Datasets could be further enriched with more nuanced layers of information, including building typologies, functional zoning, demographic data, green space distributions, road networks, and infrastructure hierarchies. Introducing temporal datasets could even enable the generation of urban forms that reflect growth patterns over time, offering predictive or retrospective design capabilities.

Another compelling research trajectory involves integrating multiple GAN models sequentially. For instance, generated urban plans could be translated into realistic satellite imagery, three-dimensional models, or even street-level perspectives, offering planners and designers a more immersive understanding of proposed urban fabrics. Such multi-step workflows could simulate how generated layouts might materialize in the real world, bridging the gap between abstract spatial design and tangible urban experience.

Ultimately, the practical adoption of these generative tools by local governments, urban planning authorities, and community stakeholders holds transformative potential. By enabling the reconstruction of urban forms that resonate with a city's historical and cultural memory while responding flexibly to contemporary needs, these technologies could fundamentally alter how we envision, negotiate, and build the future of our cities, creating spaces that honor the past while innovating toward more resilient, inclusive, and sustainable urban environments.

BIBLIOGRAPHY

Abdulkadirov, R., Lyakhov, P., & Nagornov, N. (2023). Survey of Optimization Algorithms in Modern Neural Networks. *Mathematics*, 11(11), Article 11. <https://doi.org/10.3390/math11112466>

Alawadhi, M., & Yan, W. (2021). *BIM Hyperreality: Data Synthesis Using BIM and Hyperrealistic Rendering for Deep Learning* (No. arXiv:2105.04103). arXiv. <https://doi.org/10.48550/arXiv.2105.04103>

Albert, A., Strano, E., Kaur, J., & Gonzalez, M. (2018). *Modeling urbanization patterns with generative adversarial networks* (No. arXiv:1801.02710). arXiv. <https://doi.org/10.48550/arXiv.1801.02710>

Antorán, J., Allingham, J., & Hernández-Lobato, J. (2020). *Depth uncertainty in neural networks*. <https://doi.org/10.17863/CAM.69875>

As, I., Pal, S., & Basu, P. (2018). Artificial intelligence in architecture: Generating conceptual design via deep learning. *International Journal of Architectural Computing*, 16(4), 306–327. <https://doi.org/10.1177/1478077118800982>

Bashir, D., Montañez, G. D., Sehra, S., Segura, P. S., & Lauw, J. (2020). An Information-Theoretic Perspective on Overfitting and Underfitting. In M. Gallagher, N. Moustafa, & E. Lakshika (Eds.), *Proceedings 33* (pp. 347–358). Springer International Publishing. https://doi.org/10.1007/978-3-030-64984-5_27

Batty, M. (1997). Cellular Automata and Urban Form: A Primer. *Journal of the American Planning Association*, 63(2), 266–274. <https://doi.org/10.1080/01944369708975918>

Bengio, Y., Thibodeau-Laufer, É. ric, Alain, G., & Yosinski, J. (2014). Deep generative stochastic networks trainable by backprop. *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, II-226-II–234.

Bengio, Y., Yao, L., Alain, G., & Vincent, P. (2013). Generalized Denoising Auto-Encoders as Generative Models. *Advances in Neural Information Processing Systems*, 26.

https://papers.nips.cc/paper_files/paper/2013/hash/559cb990c9dffd8675f6bc2186971dc2-Abstract.html

Bishop, C. (2006). Pattern Recognition and Machine Learning. In *Journal of Electronic Imaging* (Vol. 16, pp. 140–155). <https://doi.org/10.1117/1.2819119>

Bishop, C. M. (2013). Model-based machine learning. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1984), 20120222. <https://doi.org/10.1098/rsta.2012.0222>

Bittner, K., & Korner, M. (2018). Automatic Large-Scale 3D Building Shape Refinement Using Conditional Generative Adversarial Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1968–1982. <https://doi.org/10.1109/CVPRW.2018.00249>

Boates, I. (2020, October 11). Generating OSM maps using StyleGAN2. *Medium*. <https://iboates.medium.com/generating-osm-maps-using-stylegan2-72868539fe70>

Boim, A., Dortheimer, J., & Sprecher, A. (2022). A Machine-Learning Approach to Urban Design Interventions In Non-Planned Settlements. *Jeroen van Ameijde, Nicole Gardner, Kyung Hoon Hyun, Dan Luo, Urvi Sheth (Eds.), POST-CARBON - Proceedings of the 27th CAADRIA Conference, Sydney, 9-15 April 2022, Pp. 223-232*, 223–232. https://papers.cumincad.org/cgi-bin/works/paper/caadria2022_45

Borrohou, S., Fissoune, R., & Badir, H. (2023). Data cleaning survey and challenges – improving outlier detection algorithm in machine learning. *Journal of Smart Cities and Society*, 2(3), 125–140. <https://doi.org/10.3233/SCS-230008>

Burgess, E. (1925). The Growth of the City: An Introduction to a Research Project. In *Publications of the American Sociological Society* (Vol. 18, pp. 35–41). https://doi.org/10.1007/978-0-387-73412-5_5

Cankurt Semiz, S. N., & Özsoy, F. A. (2024). Transmission of Spatial Experience in the Context of Sustainability of Urban Memory. *Sustainability*, 16(22), 9910. <https://doi.org/10.3390/su16229910>

Chai, C., Wang, J., Luo, Y., Niu, Z., & Li, G. (2022). Data Management for Machine Learning: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 1–1. <https://doi.org/10.1109/TKDE.2022.3148237>

Chaillou, S. (2019). *AI + Architecture | Towards a New Approach* [masterThesis, Harvard University]. https://www.academia.edu/39599650/AI_Architecture_Towards_a_New_Approach

Chan, Y. H. E., & Spaeth, B. (2020). *Architectural Visualisation with Conditional Generative Adversarial Networks (cGAN). - What machines read in architectural sketches*. 299–308. <https://doi.org/10.52842/conf.ecaade.2020.2.299>

Chen, J., & Stouffs, R. (2021). *From Exploration to Interpretation—Adopting Deep Representation Learning Models to Latent Space Interpretation of Architectural Design Alternatives*. 131–140. <https://doi.org/10.52842/conf.caadria.2021.1.131>

Chen, Y., Ouyang, X., & Agam, G. (2019). ChangeNet: Learning to Detect Changes in Satellite Images. *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, 24–31. <https://doi.org/10.1145/3356471.3365232>

Cho, D., Kim, J., Shin, E., Choi, J., & Lee, J.-K. (2020). *Recognizing Architectural Objects in Floor-plan Drawings Using Deep-learning Style-transfer Algorithms*. 717–725. <https://doi.org/10.52842/conf.caadria.2020.2.717>

Chokwitthaya, C., Collier, E., Zhu, Y., & Mukhopadhyay, S. (2019). Improving Prediction Accuracy in Building Performance Models Using Generative Adversarial Networks (GANs). *2019 International Joint Conference on Neural Networks (IJCNN)*, 1–9. <https://doi.org/10.1109/IJCNN.2019.8852411>

Chokwitthaya, C., Zhu, Y., Mukhopadhyay, S., & Collier, E. (2020). Augmenting building performance predictions during design using generative adversarial networks and immersive virtual environments. *Automation in Construction*, 119, 103350. <https://doi.org/10.1016/j.autcon.2020.103350>

Christaller, W. (1933). *Central Places in Southern Germany*. <https://doi.org/10.1177/000271626636800132>

Chu, H., Li, D., Acuna, D., Kar, A., Shugrina, M., Wei, X., Liu, M.-Y., Torralba, A., & Fidler, S. (2019). Neural Turtle Graphics for Modeling City Road Layouts. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4521–4529. <https://doi.org/10.1109/ICCV.2019.00462>

Danchenko, E. (2021). The AI-teration Method and the Role of AI in Architectural Design. In K. Arai, S. Kapoor, & R. Bhatia (Eds.), *Proceedings of the Future Technologies Conference (FTC) 2020, Volume 1* (Vol. 1288, pp. 525–538). Springer International Publishing. https://doi.org/10.1007/978-3-030-63128-4_40

Del Campo, M. (2021). *Architecture, Language and AI - Language, Attentional Generative Adversarial Networks (AttnGAN) and Architecture Design*. 211–220. <https://doi.org/10.52842/conf.caadria.2021.1.211>

Deng, X., Zhu, Y., & Newsam, S. (2018). What is it like down there?: Generating dense ground-level views and image features from overhead imagery using conditional generative adversarial networks. *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 43–52. <https://doi.org/10.1145/3274895.3274969>

Ding, H., Wu, S., Tang, H., Wu, F., Gao, G., & Jing, X.-Y. (2020). Cross-View Image Synthesis with Deformable Convolution and Attention Mechanism. In Y. Peng, Q. Liu, H. Lu, Z. Sun, C. Liu, X. Chen, H. Zha, & J. Yang (Eds.), *Pattern Recognition and Computer Vision* (Vol. 12305, pp. 386–397). Springer International Publishing. https://doi.org/10.1007/978-3-030-60633-6_32

Dong, S., Wang, W., Li, W., & Zou, K. (2021). Vectorization of Floor Plans Based on EdgeGAN. *Information*, 12(5), 206. <https://doi.org/10.3390/info12050206>

Du, Z., Shen, H., Li, X., & Wang, M. (2022). 3D building fabrication with geometry and texture coordination via hybrid GAN. *Journal of Ambient Intelligence and Humanized Computing*, 13(11), 5177–5188. <https://doi.org/10.1007/s12652-020-02488-9>

Dumont, V., Ju, X., & Mueller, J. (2022). Hyperparameter Optimization of Generative Adversarial Network Models for High-Energy Physics Simulations. *ArXiv, abs/2208.07715*. <https://doi.org/10.21203/rs.3.rs-2181360/v1>

Fedorova, S. (2021). *Generative Adversarial Networks for Urban Block Design* (No. arXiv:2105.01727). arXiv. <http://arxiv.org/abs/2105.01727>

Gainetdinov, A. (2023, May 12). *Diffusion Models vs GANs vs VAEs: Comparison of Deep Generative Models*. Medium. <https://pub.towardsai.net/diffusion-models-vs-gans-vs-vaes-comparison-of-deep-generative-models-67ab93e0d9ae>

García del Castillo y López, J. (2022). *Artificial intelligence for all perspectives and outlooks on the role of machine learning in architectural design*. 18. https://assets.cmhc-schl.gc.ca/sf/project/archive/housing_organizations3/harvard_jchs_digitalization_panel_1_garciadelcastillo.pdf

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. The MIT press.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., & Bengio, Y. (2014). Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2672–2680.

Grillo, S. A. (2019). A linear relation between input and first layer in neural networks. *Annals of Mathematics and Artificial Intelligence*, 87(4), 361–372. <https://doi.org/10.1007/s10472-019-09657-3>

Gui, J., Sun, Z., Wen, Y., Tao, D., & Ye, J. (2023). A Review on Generative Adversarial Networks: Algorithms, Theory, and Applications. *IEEE Transactions on Knowledge and Data Engineering*, 35(4), 3313–3332. <https://doi.org/10.1109/TKDE.2021.3130191>

Gui, Y., Zhou, B., Xie, X., Li, W., & Zhou, X. (2021). GAN-based method for generative design of visual comfort in underground space. *IOP Conference Series: Earth and Environmental Science*, 861(7), 072015. <https://doi.org/10.1088/1755-1315/861/7/072015>

Güzelci, O. Z., Alaçam, S., Bekiroğlu, B., & Karadag, I. (2024). A machine learning-based prediction model for architectural heritage: The case of domed Sinan mosques. *Digital Applications in Archaeology and Cultural Heritage*, 35, e00370. <https://doi.org/10.1016/j.daach.2024.e00370>

- Hady, M. F. A., & Schwenker, F. (2013). Semi-supervised Learning. In M. Bianchini, M. Maggini, & L. C. Jain (Eds.), *Handbook on Neural Information Processing* (Vol. 49, pp. 215–239). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-36657-4_7
- Halbwachs, M. (1925). *On Collective Memory*. University of Chicago Press.
- Halbwachs, M. (with Internet Archive). (1950). *The collective memory*. Harper & Row. <http://archive.org/details/collectivememory00halb>
- Handelman, G. S., Kok, H. K., Chandra, R. V., Razavi, A. H., Huang, S., Brooks, M., Lee, M. J., & Asadi, H. (2019). Peering Into the Black Box of Artificial Intelligence: Evaluation Metrics of Machine Learning Methods. *American Journal of Roentgenology*, 212(1), 38–43. <https://doi.org/10.2214/AJR.18.20224>
- He, F., Liu, T., & Tao, D. (2019). Control Batch Size and Learning Rate to Generalize Well: Theoretical and Empirical Evidence. *Advances in Neural Information Processing Systems*, 32, 1141–1150.
- He, Q., Li, Z., Gao, W., Chen, H., Wu, X., Cheng, X., & Lin, B. (2021). Predictive models for daylight performance of general floorplans based on CNN and GAN: A proof-of-concept study. *Building and Environment*, 206, 108346. <https://doi.org/10.1016/j.buildenv.2021.108346>
- Hettiarachchi, P., Nawaratne, R., Alahakoon, D., De Silva, D., & Chilamkurti, N. (2021). Rain Streak Removal for Single Images Using Conditional Generative Adversarial Networks. *Applied Sciences*, 11(5), 2214. <https://doi.org/10.3390/app11052214>
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., & Hochreiter, S. (2018). GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium (No. arXiv:1706.08500). arXiv. <https://doi.org/10.48550/arXiv.1706.08500>
- Hillier, B., & Hanson, J. (1984). *The Social Logic of Space*. Cambridge University Press. <https://doi.org/10.1017/CBO9780511597237>
- Hochreiter, S. (1998). The Vanishing Gradient Problem During Learning Recurrent Neural Nets and Problem Solutions. *International Journal of Uncertainty, Fuzziness and*

Knowledge-Based Systems, 06(02), 107–116.
<https://doi.org/10.1142/S0218488598000094>

Hofmannsthal, H. von. (1902). *Letter to Lord Chandos*.
https://depts.washington.edu/vienna/documents/Hofmannsthal/Hofmannsthal_Chandos.htm

Hou, B., Liu, Q., Wang, H., & Wang, Y. (2020). From W-Net to CDGAN: Bitemporal Change Detection via Deep Learning Techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3), 1790–1802. <https://doi.org/10.1109/TGRS.2019.2948659>

Howard, E. (1898). *To-morrow: A peaceful path to real reform*. Swan Sonnenschein & Co. <https://archive.org/details/tomorrowpeaceful00howa/page/6/mode/2up>

Huang, J., Johanes, M., Kim, F. C., Doumptioti, C., & Holz, G.-C. (2021). On GANs, NLP and Architecture: Combining Human and Machine Intelligences for the Generation and Evaluation of Meaningful Designs. *Technology|Architecture + Design*, 5(2), 207–224. <https://doi.org/10.1080/24751448.2021.1967060>

Huang, Z., & Huang, L. (2024). Research Progress and Prospects of “Urban Memory” Renewal Planning Paths—Based on International Comparative Perspective. *Scientific and Social Research*, 6(8), 98–106. <https://doi.org/10.26689/ssr.v6i8.7927>

Ibrahim, H., Khattab, Z., Khattab, T., & Abraham, R. (2023). Expatriates’ Housing Dispersal Outlook in a Rapidly Developing Metropolis Based on Urban Growth Predicted Using a Machine Learning Algorithm. *Housing Policy Debate*, 33(3), 641–661. <https://doi.org/10.1080/10511482.2021.1962939>

Ikeno, K., Fukuda, T., & Yabuki, N. (2021). An enhanced 3D model and generative adversarial network for automated generation of horizontal building mask images and cloudless aerial photographs. *Advanced Engineering Informatics*, 50, 101380. <https://doi.org/10.1016/j.aei.2021.101380>

Johnsen, M., Brandt, O., Garrido, S., & Pereira, F. (2022). Population synthesis for urban resident modeling using deep generative models. *Neural Computing and Applications*, 34(6), 4677–4692. <https://doi.org/10.1007/s00521-021-06622-2>

- Kang, Y., Gao, S., & Roth, R. E. (2019). Transferring multiscale map styles using generative adversarial networks. *International Journal of Cartography*, 5(2–3), 115–141. <https://doi.org/10.1080/23729333.2019.1615729>
- Kaur, H., Pannu, H. S., & Malhi, A. K. (2020). A Systematic Review on Imbalanced Data Challenges in Machine Learning: Applications and Solutions. *ACM Computing Surveys*, 52(4), 1–36. <https://doi.org/10.1145/3343440>
- Kelly, T., Guerrero, P., Steed, A., Wonka, P., & Mitra, N. J. (2018). FrankenGAN: Guided detail synthesis for building mass models using style-Synchronized Gans. *ACM Transactions on Graphics*, 37, 1–14. <https://doi.org/10.1145/3272127.327506>
- Kempinska, K., & Murcio, R. (2019). Modelling urban networks using Variational Autoencoders. *Applied Network Science*, 4(1), 114. <https://doi.org/10.1007/s41109-019-0234-0>
- Kim, S., Kim, D., & Choi, S. (2020). CityCraft: 3D virtual city creation from a single image. *The Visual Computer*, 36, 911–924. <https://doi.org/10.1007/s00371-019-01701-x>
- Kim, S., Park, S., & Yu, K. (2018). Application of Style Transfer in the Vectorization Process of Floorplans (Short Paper) [Application/pdf]. *LIPICs, Volume 114, GIScience 2018, 114*, 39:1-39:6. <https://doi.org/10.4230/LIPICS.GISCIENCE.2018.39>
- Kossale, Y., Airaj, M., & Darouichi, A. (2022). Mode Collapse in Generative Adversarial Networks: An Overview. *2022 8th International Conference on Optimization and Applications (ICOA)*, 1–6. <https://doi.org/10.1109/ICOA55659.2022.9934291>
- Kriegeskorte, N., & Golan, T. (2019). Neural network models and deep learning. *Current Biology*, 29(7), R231–R236. <https://doi.org/10.1016/j.cub.2019.02.034>
- Kruse, R., Borgelt, C., Braune, C., Mostaghim, S., & Steinbrecher, M. (2016). Introduction to Neural Networks. In R. Kruse, C. Borgelt, C. Braune, S. Mostaghim, & M. Steinbrecher (Eds.), *Computational Intelligence: A Methodological Introduction* (pp. 9–13). Springer. https://doi.org/10.1007/978-1-4471-7296-3_2
- Levy, A. (1999). Urban morphology and the problem of the modern urban fabric: Some questions for research. *Urban Morphology*. <https://doi.org/10.51347/jum.v3i2.3885>

- Li, J., Chen, Z., Zhao, X., & Shao, L. (2020). MapGAN: An Intelligent Generation Model for Network Tile Maps. *Sensors*, 20(11), 3119. <https://doi.org/10.3390/s20113119>
- Li, J., He, F., Zhang, L., Du, B., & Tao, D. (2019). Progressive Reconstruction of Visual Structure for Image Inpainting. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 5961–5970. <https://doi.org/10.1109/ICCV.2019.00606>
- Li, X., Lv, Z., Hijazi, I., Jiao, H., Li, L., & Li, K. (2016). Assessment of Urban Fabric for Smart Cities. *IEEE Access*, 4, 373–382. <https://doi.org/10.1109/ACCESS.2016.2517072>
- Liao, W., Lu, X., Huang, Y., Zheng, Z., & Lin, Y. (2021). Automated structural design of shear wall residential buildings using generative adversarial networks. *Automation in Construction*, 132, 103931. <https://doi.org/10.1016/j.autcon.2021.103931>
- Liao, Z., & Dai, G. (2020). Inheritance and Dissemination of Cultural Collective Memory: An Analysis of a Traditional Festival. *SAGE Open*, 10(1), 2158244020901601. <https://doi.org/10.1177/2158244020901601>
- Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., & Liao, J. (2021). PD-GAN: Probabilistic Diverse GAN for Image Inpainting. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9367–9376. <https://doi.org/10.1109/CVPR46437.2021.00925>
- Liu, Y., Lai, Y., Chen, J., Liang, L., & Deng, Q. (2020). SCUT-AutoALP: A Diverse Benchmark Dataset for Automatic Architectural Layout Parsing. *IEICE Transactions on Information and Systems*, E103.D(12), 2725–2729. <https://doi.org/10.1587/transinf.2020EDL8076>
- Lowry, I. S. (1964). *A model of metropolis*. Rand Corporation. <https://www.semanticscholar.org/paper/A-model-of-metropolis-Lowry/9e805610931906efdd5396923eebacc13abf6205>
- Lynch, K. (1981). *A theory of good city form*. MIT Press. <https://search.worldcat.org/title/977272771>
- Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *ArXiv*, abs/1804.07612. <https://consensus.app/papers/revisiting-small-batch->

training-for-deep-neural-networks-masters-
luschi/524fe2c7e97e59b6ab6f4c0470f2a68b/

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The Bulletin of Mathematical Biophysics*, 5(4), 115–133. <https://doi.org/10.1007/BF02478259>

Mendula, M., Bujari, A., Foschini, L., & Bellavista, P. (2022). A Data-Driven Digital Twin for Urban Activity Monitoring. *2022 IEEE Symposium on Computers and Communications (ISCC)*, 1–6. <https://doi.org/10.1109/ISCC55528.2022.9912914>

Mitchell, T. M. (1977). *Machine learning* (Nachdr.). McGraw-Hill.

Nakama, T. (2011). Comparisons of Single- and Multiple-Hidden-Layer Neural Networks. In D. Liu, H. Zhang, M. Polycarpou, C. Alippi, & H. He (Eds.), *Advances in Neural Networks – ISNN 2011* (pp. 270–279). Springer. https://doi.org/10.1007/978-3-642-21105-8_32

Nauata, N., Chang, K.-H., Cheng, C.-Y., Mori, G., & Furukawa, Y. (2020). House-GAN: Relational Generative Adversarial Networks for Graph-Constrained House Layout Generation. In A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Eds.), *Computer Vision – ECCV 2020* (Vol. 12346, pp. 162–177). Springer International Publishing. https://doi.org/10.1007/978-3-030-58452-8_10

Nauata, N., Hosseini, S., Chang, K.-H., Chu, H., Cheng, C.-Y., & Furukawa, Y. (2021). House-GAN++: Generative Adversarial Layout Refinement Network towards Intelligent Computational Agent for Professional Architects. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13627–13636. <https://doi.org/10.1109/CVPR46437.2021.01342>

Navarro-Mateu, D., Carrasco, O., & Cortes Nieves, P. (2021). Color-Patterns to Architecture Conversion through Conditional Generative Adversarial Networks. *Biomimetics*, 6(1), 16. <https://doi.org/10.3390/biomimetics6010016>

Newman, P., Kosonen, L., & Kenworthy, J. (2016). Theory of urban fabrics: Planning the walking, transit/public transport and automobile/motor car cities for reduced car dependency. *Town Planning Review*, 87, 429–458. <https://doi.org/10.3828/tpr.2016.28>

Newton, D. (2019). Generative Deep Learning in Architectural Design. *Technology|Architecture + Design*, 3(2), 176–189. <https://doi.org/10.1080/24751448.2019.1640536>

Oladipupo, T. (2010). *Types of Machine Learning Algorithms* (Y. Zhang, Ed.). InTech. <https://doi.org/10.5772/9385>

Olick, J. K., & Robbins, J. (1998). Social Memory Studies: From “Collective Memory” to the Historical Sociology of Mnemonic Practices. *Annual Review of Sociology*, 24, 105–140.

Owaki, T., & Machida, T. (2020). RoadNetGAN: Generating Road Networks in Planar Graph Representation. In H. Yang, K. Pasupa, A. C.-S. Leung, J. T. Kwok, J. H. Chan, & I. King (Eds.), *Neural Information Processing* (Vol. 1332, pp. 535–543). Springer International Publishing. https://doi.org/10.1007/978-3-030-63820-7_61

Pang, H. E., & Biljecki, F. (2022). 3D building reconstruction from single street view images using deep learning. *International Journal of Applied Earth Observation and Geoinformation*, 112, 102859. <https://doi.org/10.1016/j.jag.2022.102859>

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context Encoders: Feature Learning by Inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2536–2544. <https://doi.org/10.1109/CVPR.2016.278>

Paullada, A., Raji, I. D., Bender, E. M., Denton, E. L., & Hanna, A. (2020). Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2. <https://doi.org/10.1016/j.patter.2021.100336>

Pavel, M.-D., Roşioru, S., Arghira, N., & Stamatescu, G. (2023). Control of Open Mobile Robotic Platform Using Deep Reinforcement Learning. In T. Borangiu, D. Trentesaux, & P. Leitão (Eds.), *Service Oriented, Holonic and Multi-Agent Manufacturing Systems for Industry of the Future* (Vol. 1083, pp. 368–379). Springer International Publishing. https://doi.org/10.1007/978-3-031-24291-5_29

Pham, V.-D., & Bui, Q.-T. (2021). Spatial resolution enhancement method for Landsat imagery using a Generative Adversarial Network. *Remote Sensing Letters*, 12(7), 654–665. <https://doi.org/10.1080/2150704X.2021.1918789>

- Pitts, W., & McCulloch, W. S. (1947). How we know universals the perception of auditory and visual forms. *The Bulletin of Mathematical Biophysics*, 9(3), 127–147. <https://doi.org/10.1007/BF02478291>
- Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data Lifecycle Challenges in Production Machine Learning: A Survey. *SIGMOD Rec.*, 47(2), 17–28. <https://doi.org/10.1145/3299887.3299891>
- Qian, W., Xu, Y., & Li, H. (2022). A self-sparse generative adversarial network for autonomous early-stage design of architectural sketches. *Computer-Aided Civil and Infrastructure Engineering*, 37(5), 612–628. <https://doi.org/10.1111/mice.12759>
- Qin, X., Chen, W., Shen, Q., Jiang, J., & Feng, G. (2017). Image Inpainting: A Contextual Consistent and Deep Generative Adversarial Training Approach. *2017 4th IAPR Asian Conference on Pattern Recognition (ACPR)*, 588–593. <https://doi.org/10.1109/ACPR.2017.120>
- Rachele, J. N., Wang, J., Wijnands, J. S., Zhao, H., Bentley, R., & Stevenson, M. (2021). Using machine learning to examine associations between the built environment and physical function: A feasibility study. *Health & Place*, 70, 102601. <https://doi.org/10.1016/j.healthplace.2021.102601>
- Regmi, K., & Borji, A. (2018). Cross-View Image Synthesis Using Conditional GANs. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3501–3510. <https://doi.org/10.1109/CVPR.2018.00369>
- Regmi, K., & Borji, A. (2019). Cross-view image synthesis using geometry-guided conditional GANs. *Computer Vision and Image Understanding*, 187, 102788. <https://doi.org/10.1016/j.cviu.2019.07.008>
- Richman, R., Zirnelt, H., & Fix, S. (2014). Large-scale building simulation using cloud computing for estimating lifecycle energy consumption. *Canadian Journal of Civil Engineering*, 41(3), 252–262. <https://doi.org/10.1139/cjce-2013-0235>
- Rogers, J., & Revesz, A. (2019). Experimental and quasi-experimental designs. In *The Routledge Handbook of Research Methods in Applied Linguistics* (1st ed.). Routledge.

rossi, aldo. (1982). *The Architecture of the City*. MIT Press.
<https://mitpress.mit.edu/9780262680431/the-architecture-of-the-city/>

Sato, G., Ishizawa, T., Iseda, H., & Kitahara, H. (2020). *Automatic Generation of the Schematic Mechanical System Drawing by Generative Adversarial Network*. 403–410.
<https://doi.org/10.52842/conf.ecaade.2020.1.403>

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117. <https://doi.org/10.1016/j.neunet.2014.09.003>

Schmidhuber, J. (2022). *Annotated History of Modern AI and Deep Learning*.
<http://arxiv.org/abs/2212.11279>

Sharma, D., Gupta, N., Chattopadhyay, C., & Mehta, S. (2018). REXplore: A Sketch Based Interactive Explorer for Real Estates Using Building Floor Plan Images. *2018 IEEE International Symposium on Multimedia (ISM)*, 61–64.
<https://doi.org/10.1109/ISM.2018.00018>

Shi, Q., Liu, X., & Li, X. (2018). Road Detection From Remote Sensing Images by Generative Adversarial Networks. *IEEE Access*, 6, 25486–25494.
<https://doi.org/10.1109/ACCESS.2017.2773142>

Steinfeld, K. (2022). Imaging Place Using Generative Adversarial Networks (GAN Loci). In S. Carta (Ed.), *Machine Learning and the City* (1st ed., pp. 513–516). Wiley.
<https://doi.org/10.1002/9781119815075.ch42>

Stiny, G. (1980). Introduction to Shape and Shape Grammars. *Environment and Planning B: Planning and Design*, 7(3), 343–351. <https://doi.org/10.1068/b070343>

Sun, C., Zhou, Y., & Han, Y. (2022). Automatic generation of architecture facade for historical urban renovation using generative adversarial network. *Building and Environment*, 212, 108781. <https://doi.org/10.1016/j.buildenv.2022.108781>

Sun, S., Mu, L., Feng, R., Wang, L., & He, J. (2021). GAN-Based LUCC Prediction via the Combination of Prior City Planning Information and Land-Use Probability. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, 10189–10198. <https://doi.org/10.1109/JSTARS.2021.3106481>

- Sun, S., Mu, L., Wang, L., Liu, P., Liu, X., & Zhang, Y. (2021). Semantic Segmentation for Buildings of Large Intra-Class Variation in Remote Sensing Images with O-GAN. *Remote Sensing*, 13(3), 475. <https://doi.org/10.3390/rs13030475>
- Tang, H., Liu, H., & Sebe, N. (2020). Unified Generative Adversarial Networks for Controllable Image-to-Image Translation. *IEEE Transactions on Image Processing*, 29, 8916–8929. <https://doi.org/10.1109/TIP.2020.3021789>
- Tang, H., Xu, D., Sebe, N., Wang, Y., Corso, J. J., & Yan, Y. (2019). *Multi-Channel Attention Selection GAN with Cascaded Semantic Guidance for Cross-View Image Translation* (No. arXiv:1904.06807). arXiv. <https://doi.org/10.48550/arXiv.1904.06807>
- Tang, H., Xu, D., Yan, Y., Torr, P. H. S., & Sebe, N. (2020). *Local Class-Specific and Global Image-Level Generative Adversarial Networks for Semantic-Guided Scene Generation* (No. arXiv:1912.12215). arXiv. <https://doi.org/10.48550/arXiv.1912.12215>
- Tobler, W. R. (1979). Cellular Geography. In S. Gale & G. Olsson (Eds.), *Philosophy in Geography* (pp. 379–386). Springer Netherlands. https://doi.org/10.1007/978-94-009-9394-5_18
- Tomlinson, R. F. (1974). *Application of electronic computing methods and techniques to the storage, compilation and assessment of mapped data: Ph. D. thesis* | WorldCat.org [university of london]. <https://search.worldcat.org/title/85954223>
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 59(236), 433–460.
- Ulu, A., & Karakoç, İ. (2004). Kentsel Değişimin Kent Kimliğine Etkisi. *TMMOB ŞEHİR PLANCILARI ODASI YAYINI*, 3(29), 59–66.
- Uzun, C. (2020). *Yapay zeka ve mimarlık etkileşimi üzerine bir çalışma: Üretken çekişmeli ağ algoritması ile otonom mimari plan üretimi ve değerlendirmesi* [Doktora, İstanbul Teknik Üniversitesi]. <http://hdl.handle.net/11527/20011>
- Varia, N., Dokania, A., & Senthilnath, J. (2018). DeepExt: A Convolution Neural Network for Road Extraction using RGB images captured by UAV. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1890–1895. <https://doi.org/10.1109/SSCI.2018.8628717>

- Venator, M., Aklanoglu, S., Bruns, E., & Maier, A. (2021). Enhancing collaborative road scene reconstruction with unsupervised domain alignment. *Machine Vision and Applications*, 32(1), 13. <https://doi.org/10.1007/s00138-020-01144-8>
- Villanueva-Merino, A., Urra-Uriarte, S., Izkara, J. L., Campos-Cordobes, S., Aranguren, A., & Molina-Costa, P. (2024). Leveraging Local Digital Twins for planning age-friendly urban environments. *Cities*, 155, 105458. <https://doi.org/10.1016/j.cities.2024.105458>
- Wang, D., Fu, Y., Wang, P., Huang, B., & Lu, C.-T. (2020a). Reimagining City Configuration: Automated Urban Planning via Adversarial Learning. *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 497–506. <https://doi.org/10.1145/3397536.3422268>
- Wang, D., Fu, Y., Wang, P., Huang, B., & Lu, C.-T. (2020b). Reimagining City Configuration: Automated Urban Planning via Adversarial Learning. *Proceedings of the 28th International Conference on Advances in Geographic Information Systems*, 497–506. <https://doi.org/10.1145/3397536.3422268>
- Wang, M., Yang, G.-Y., Li, R., Liang, R.-Z., Zhang, S.-H., Hall, P. M., & Hu, S.-M. (2019). Example-Guided Style-Consistent Image Synthesis From Semantic Labeling. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1495–1504. <https://doi.org/10.1109/CVPR.2019.00159>
- Wang, N., Zeng, X., Xie, R., Gao, Z., Zheng, Y., Liao, Z., Yang, J., & Wang, Q. (2019). *Unsupervised Classification of Street Architectures Based on InfoGAN* (No. arXiv:1905.12844). arXiv. <https://doi.org/10.48550/arXiv.1905.12844>
- Wang, Q., Ma, Y., Zhao, K., & Tian, Y. (2020). A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*, 9, 187–212. <https://doi.org/10.1007/s40745-020-00253-5>
- Wang, S., Cao, D., Lin, D., & Chao, F. (2018). Traffic Condition Analysis Based on Users Emotion Tendency of Microblog. In F. Chao, S. Schockaert, & Q. Zhang (Eds.), *Advances in Computational Intelligence Systems* (Vol. 650, pp. 299–311). Springer International Publishing. https://doi.org/10.1007/978-3-319-66939-7_26

- Wang, S. M., & Huang, C.-J. (2019). Using Space Syntax and Information Visualization for Spatial Behavior Analysis and Simulation. *International Journal of Advanced Computer Science and Applications*, 10(4). <https://doi.org/10.14569/IJACSA.2019.0100463>
- Wang, Y., Zorzi, S., & Bittner, K. (2021). Machine-learned 3D Building Vectorization from Satellite Imagery. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1072–1081. <https://doi.org/10.1109/CVPRW53098.2021.00118>
- 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>
- Werbos, P. (1990). Backpropagation through time: What it does and how to do it. *Proceedings of the IEEE*, 78, 1550–1560. <https://doi.org/10.1109/5.58337>
- Wijnands, J. S., Nice, K. A., Thompson, J., Zhao, H., & Stevenson, M. (2019). Streetscape augmentation using generative adversarial networks: Insights related to health and wellbeing. *Sustainable Cities and Society*, 49, 101602. <https://doi.org/10.1016/j.scs.2019.101602>
- Wu, A., Stouffs, R., & Biljecki, F. (2022). Generative Adversarial Networks in the built environment: A comprehensive review of the application of GANs across data types and scales. *Building and Environment*, 223, 109477. <https://doi.org/10.1016/j.buildenv.2022.109477>
- Wu, W., Qi, H., Rong, Z., Liu, L., & Su, H. (2018). Scribble-Supervised Segmentation of Aerial Building Footprints Using Adversarial Learning. *IEEE Access*, 6, 58898–58911. <https://doi.org/10.1109/ACCESS.2018.2874544>
- Xu, C., & Zhao, B. (2018). Satellite Image Spoofing: Creating Remote Sensing Dataset with Generative Adversarial Networks (Short Paper) [Application/pdf]. *LIPICs, Volume 114, GIScience 2018*, 114, 67:1-67:6. <https://doi.org/10.4230/LIPICS.GISCIENCE.2018.67>

- Xu, H., Omitaomu, F., Sabri, S., Zlatanova, S., Li, X., & Song, Y. (2024). Leveraging generative AI for urban digital twins: A scoping review on the autonomous generation of urban data, scenarios, designs, and 3D city models for smart city advancement. *Urban Informatics*, 3(1), 29. <https://doi.org/10.1007/s44212-024-00060-w>
- Yang, B., Kang, Y., Yuan, Y., Li, H., & Wang, F. (2022). ST-FVGAN: Filling series traffic missing values with generative adversarial network. *Transportation Letters*, 14(4), 407–415. <https://doi.org/10.1080/19427867.2021.1879624>
- Yang, F., Lu, Z., Qiu, G., Lin, J., & Zhang, Q. (2019). Capsule Based Image Synthesis for Interior Design Effect Rendering. In C. V. Jawahar, H. Li, G. Mori, & K. Schindler (Eds.), *Computer Vision – ACCV 2018* (Vol. 11365, pp. 183–198). Springer International Publishing. https://doi.org/10.1007/978-3-030-20873-8_12
- Yang, S., Wang, S., Sun, L., Luo, Z., & Bao, Y. (2023). Output Layer Structure Optimization for Weighted Regularized Extreme Learning Machine Based on Binary Method. *Symmetry*, 15(1), 244. <https://doi.org/10.3390/sym15010244>
- Yao, J., Huang, C., Peng, X., & Yuan, P. F. (2021). Generative design method of building group—Based on generative adversarial network and genetic algorithm. *Proceedings of the 26th International Conference of the Association for Computer-Aided Architectural Design Research in Asia*, 61–70. <https://doi.org/10.52842/conf.caadria.2021.1.061>
- Ye, X., Du, J., & Ye, Y. (2022). MasterplanGAN: Facilitating the smart rendering of urban master plans via generative adversarial networks. *Environment and Planning B: Urban Analytics and City Science*, 49(3), 794–814. <https://doi.org/10.1177/23998083211023516>
- Yeh, R. A., Chen, C., Lim, T. Y., Schwing, A. G., Hasegawa-Johnson, M., & Do, M. N. (2017). Semantic Image Inpainting with Deep Generative Models. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6882–6890. <https://doi.org/10.1109/CVPR.2017.728>
- Yin, R. K. (2002). *Case Study Research: Design and Methods* (3rd ed.). SAGE.
- Zhang, G., Rui, X., Poslad, S., Song, X., Fan, Y., & Wu, B. (2020). A Method for the Estimation of Finely-Grained Temporal Spatial Human Population Density Distributions

Based on Cell Phone Call Detail Records. *Remote Sensing*, 12(16), 2572. <https://doi.org/10.3390/rs12162572>

Zhang, H., & Blasetti, E. (2020). 3D Architectural Form Style Transfer through Machine Learning (Full Version). *Design in the Age of Humans - Proceedings of the 25th International Conference on Computer-Aided Architectural Design Research in Asia*, 661-670. <https://doi.org/10.13140/RG.2.2.16791.52645>

Zhang, H., Hu, Z., Luo, C., Zuo, W., & Wang, M. (2018). Semantic Image Inpainting with Progressive Generative Networks. *Proceedings of the 26th ACM International Conference on Multimedia*, 1939–1947. <https://doi.org/10.1145/3240508.3240625>

Zhang, J., Fukuda, T., & Yabuki, N. (2021). Automatic Object Removal With Obstructed Façades Completion Using Semantic Segmentation and Generative Adversarial Inpainting. *IEEE Access*, 9, 117486–117495. <https://doi.org/10.1109/ACCESS.2021.3106124>

Zhang, S., Han, Z., Lai, Y.-K., Zwicker, M., & Zhang, H. (2019). Stylistic scene enhancement GAN: Mixed stylistic enhancement generation for 3D indoor scenes. *The Visual Computer*, 35(6–8), 1157–1169. <https://doi.org/10.1007/s00371-019-01691-w>

Zhang, Y., Li, Y., Zhou, X., Kong, X., & Luo, J. (2019). TrafficGAN: Off-Deployment Traffic Estimation with Traffic Generative Adversarial Networks. *2019 IEEE International Conference on Data Mining (ICDM)*, 1474–1479. <https://doi.org/10.1109/ICDM.2019.00193>

Zhang, Y., Li, Y., Zhou, X., Kong, X., & Luo, J. (2022). Off-Deployment Traffic Estimation—A Traffic Generative Adversarial Networks Approach. *IEEE Transactions on Big Data*, 8(4), 1084–1095. <https://doi.org/10.1109/TBDATA.2020.3014511>

Zhang, Y., Yin, Y., Zimmermann, R., Wang, G., Varadarajan, J., & Ng, S.-K. (2020). An Enhanced GAN Model for Automatic Satellite-to-Map Image Conversion. *IEEE Access*, 8, 176704–176716. <https://doi.org/10.1109/ACCESS.2020.3025008>

Zhao, B., Zhang, S., Xu, C., Sun, Y., & Deng, C. (2021). Deep fake geography? When geospatial data encounter Artificial Intelligence. *Cartography and Geographic Information Science*, 48(4), 338–352. <https://doi.org/10.1080/15230406.2021.1910075>

Zhao, C., Yang, J., Xiong, W., & Li, J. (2021). Two Generative Design Methods of Hospital Operating Department Layouts Based on Healthcare Systematic Layout Planning and Generative Adversarial Network. *Journal of Shanghai Jiaotong University (Science)*, 26(1), 103–115. <https://doi.org/10.1007/s12204-021-2265-9>

Zhao, H., Wijnands, J. S., Nice, K. A., Thompson, J., Aschwanden, G. D. P. A., Stevenson, M., & Guo, J. (2019). Unsupervised Deep Learning to Explore Streetscape Factors Associated with Urban Cyclist Safety. In X. Qu, L. Zhen, R. J. Howlett, & L. C. Jain (Eds.), *Smart Transportation Systems 2019* (Vol. 149, pp. 155–164). Springer Singapore. https://doi.org/10.1007/978-981-13-8683-1_16

Zhao, L., Mo, Q., Lin, S., Wang, Z., Zuo, Z., Chen, H., Xing, W., & Lu, D. (2020). UCTGAN: Diverse Image Inpainting Based on Unsupervised Cross-Space Translation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5740–5749. <https://doi.org/10.1109/CVPR42600.2020.00578>

Zhu, P., Abdal, R., Qin, Y., & Wonka, P. (2020). SEAN: Image Synthesis With Semantic Region-Adaptive Normalization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 5103–5112. <https://doi.org/10.1109/CVPR42600.2020.00515>

CURRICULUM VITAE

20016 – 2021

B.Sc., Architecture, Selçuk University, Konya,
TÜRKİYE

