

Cilt 1 Sayı 3

Eylül

2020

Vol 1 No 3

September

2020

ISSN 2687-4318

**J**

journal of

**CO**

computational

**DE**

design

mimarlıkta yapay zeka

artificial intelligence in architecture

**JCODE**  
journal of Computational Design



Cilt 1 Sayı 3

Vol 1 No 3

Eylül

September

2020

2020

**J**

journal of

**CO**

computational

**DE**

design

ISSN 2687-4318

mimarlıkta yapay zeka

artificial intelligence in architecture

**Cilt 1 Sayı 3 | Eylül 2020**

Vol 1 No 3 | September 2020

**İstanbul Teknik Üniversitesi Mimarlık Fakültesi E-Dergisi**

Istanbul Technical University Faculty of Architecture E-Journal

**Yılda iki kez yayınlanır.** | Published two issues in one year.

**Yayıncı** | Publisher

İstanbul Teknik Üniversitesi Rektörlüğü | Istanbul Technical University Rectorate

**Editör** | Editor

Prof. Dr. Gülen Çağdaş

Doç. Dr. Sema Alaçam

Dr. Öğr. Üyesi Ethem Gürer

**Yayın Kurulu** | Editorial Board

Prof. Dr. Leman Figen Gül

Prof. Dr. Mine Özkar

Prof. Dr. Hakan Yaman

Doç. Dr. Meltem Aksoy

Doç. Dr. Hasan Serdar Kaya

Doç. Dr. Gülten Manioğlu

Dr. Öğr. Üyesi Bahriye İlhan Jones

Dr. Öğr. Üyesi Aslı Kanan

Dr. Öğr. Üyesi Muhammed Ali Örnek

Dr. Öğr. Üyesi Sevil Yazıcı

Öğr. Gör. Dr. Elif Sezen Yağmur Kilimci

Araş. Gör. Dr. Hakan Tong

**Danışma Kurulu** | Advisory Board

Prof. Dr. Rahmi Nurhan Çelik (İstanbul Teknik Üniversitesi)

Doç. Dr. Gülay Öke Günel (İstanbul Teknik Üniversitesi)

Dr. Öğr. Üyesi Mehmet Tahir Sandıkkaya (İstanbul Teknik Üniversitesi)

Prof. Dr. Özgür Ediz (Uludağ Üniversitesi)

Doç. Dr. Neşe Çakıcı Alp (Kocaeli Üniversitesi)

Doç. Dr. Güzden Varinlioğlu (İzmir Ekonomi Üniversitesi)

Dr. Öğr. Üyesi Yazgı Badem Aksoy (Medipol Üniversitesi)

Dr. Öğr. Üyesi Güven Çatak (Bahçeşehir Üniversitesi)

Dr. Öğr. Üyesi Ahmet Emre Dinçer (Karabük Üniversitesi)

Dr. Öğr. Üyesi Özlem Atak Doğan (Erciyes Üniversitesi)

Dr. Öğr. Üyesi Şehnaz Cenani Durmazoğlu (Medipol Üniversitesi)

Dr. Öğr. Üyesi Esra Gürbüz Yıldırım (Gaziantep Üniversitesi)

Dr. Öğr. Üyesi Ömer Halil Çavuşoğlu (Erzurum Teknik Üniversitesi)

Dr. Öğr. Üyesi Serdar Aydın (Mardin Artuklu Üniversitesi)

Fernando García Amen (Universidad de la República, Uruguay)

**Bölge Temsilcisi** | Regional Representative

Benay Gürsoy Toykoç (Pennsylvania State University, ABD)

**Editorial Sekreteryası** | Editorial Assistance

Gülce Kırdar

Özlem Çavuş

Begüm Hamzaoğlu

**Dizgi** | Typesetting

Gülce Kırdar

Özlem Çavuş

**Logo** | Logo

Melis Dağ

**Kapak** | Cover

İlke Yıldan

Varlık Yücel

Begüm Hamzaoğlu

**Web** | Web

Begüm Hamzaoğlu

Gülce Kırdar

Özlem Çavuş



ISSN 2687-4318

**İletişim** | Contact

JCoDe: Journal of Computational Design

Yayın Sekreterliği

İstanbul Teknik Üniversitesi

Mimarlık Fakültesi

Taşkışla, Taksim, 34437

İstanbul Türkiye

email: jcode@itu.edu.tr

web: jcode.itu.edu.tr

# Mimarlıkta Yapay Zeka

## Editörden

JCoDe'un üçüncü sayısı, mimari tasarım ve yapım/üretim süreçlerinde son yıllarda etkin biçimde kullanılmaya başlanan "Mimarlıkta Yapay Zeka"yı (YZ) odak olarak almaktadır. Zeka, insanın düşünme, akıl yürütme/yargılama, objektif gerçekleri algılama ve sonuç çıkarma yeteneklerinin tamamı olarak tanımlanabilir. YZ, zeka kavramından yola çıkarak, insanlar gibi düşünen, akıl yürüten, algılayan ve hareket edebilen, öğrenebilen ve tek başlarına problem çözmek için dağarcığındaki bilgilerini kullanarak sonuç üreten bilgisayar modeli oluşturma çalışmalarıdır.

YZ çalışmalarının ilk döneminde formel dillere dayanan temsiller ve modeller ön plandayken, 1990'lara gelindiğinde kuramsal düzlemde yapılmış olan gelecek öngörülerinin, uygulamada geline aşama ile birlikte değerlendirilerek yeniden gözden geçirilmesine ihtiyaç duyulmuştur. Günümüzde ise bilginin formel dillere dayanan modelleri ve algılamanın farklı modlarına (görsel, işitsel, harekete dayalı, sinyale dayalı gibi) ilaveten, eylemlerin geçtiği çevre ve bütün yönleriyle temsil edilemeyen bir çevrede "formel olmayan" davranışların çözümlenmesi de yapay zekanın konusu haline gelmiştir.

Mimarlıkta kullanılan geleneksel bilgisayar modelleri, problem çözümünde algoritmik yöntemlerle sayısal ve mantıksal işlemlere dayanan bir süreç izler. Bu modeller, genellikle iyi tanımlanmış problemlere uygun çözümler getirebilmektedir.

İyi tanımlanmamış tasarım problemlerinin çözümünde ve uzmanlık bilgisi gerektiren alanlarda, çözümler üretip önerilerde bulunmak amacıyla YZ'nin ilk uygulama alanı olarak uzman sistemler kullanılmıştır. 1990'lı yıllardan itibaren tasarımcının nasıl düşündüğünü ve tasarım bilgisini dışsallaştırmayı hedefleyen örnek tabanlı akıl yürütme, bilgi tabanlı akıl yürütme ve görsel akıl yürütme arayışları ivme kazanmıştır. İyi tanımlanmamış mimarlık problemlerinin çözümünde, buluşsal/sezgisel arama yöntemleri kullanılmaktadır. Güncel ve gelişmekte olan YZ yöntemleri, mimarlığa parçaların aritmetik toplamı olmanın ötesinde bütün bileşenlerinin birbiri ile etkileşim halinde olduğu, dinamik ve öğrenen yeni bir "kara kutu" olanağı sunmaktadır.

YZ çalışmaları, Uzman Sistemler, Karar Destek Sistemleri, Etmen Sistemler, Genetik Algoritmalar, Sürü Zekası, Bulanık Mantık, Yapay Sinir Ağları (makine öğrenmesi, derin öğrenme), Doğal Dil İşleme, Görüntü İşleme, Veri Madenciliği, Robotik, Nesnelerin İnterneti ve Paylaşımlı YZ gibi konularla mimarlıkta uygulama alanı bulmaktadır. Bu bağlamda JCoDe'un üçüncü sayısında, YZ'nin, mimari tasarım ve üretim süreçlerindeki kullanımları; disiplinler arası işbirliği olanakları; mimarlık eğitimine entegrasyonu ve mimari tasarımın geleceğine dair etkileri tartışmaya açılmaktadır.

Sevil Yazıcı, “Doğal Sistemlerle Bilgilendirilmiş Hesaplamalı Tasarım” başlıklı yazısında mimarlık eğitiminde hesaplamalı tasarım süreçlerinin ele alınması için doğal sistemlerin analitik bir bakış açısıyla çözümlenmesine dayalı bir model önermekte ve 2018-2020 yılları arasında uygulanan model kapsamındaki çıktıları organizasyon-, performans-, süreç- ve hareket-tabanlı hesaplama ölçütleri üzerinden değerlendirmektedir.

Sena Kaynarkaya ve Gülen Çağdaş, “Sürü Zekâsı Yaklaşımı İle Metro Hatlarının Değerlendirilmesi” başlıklı makalelerinde Göztepe-Ümraniye metro hattının sürü zekâsı yaklaşımı ile cıvık mantar davranışlarını sayısal ortamda modelleyerek değerlendirmekte ve kentsel ölçekte sürü zekâsı davranışına dayalı sayısal modellerin potansiyellerini tartışmaya açmaktadır. Esranur Demirtaş ve Şehnaz Cenani ise, “Boğaz Hattındaki Trafik Optimizasyonuna Yönelik Bir Karar Destek Sistemi Önerisi: E-Bosphorus” başlıklı makalelerinde deniz trafiğini düzenleyici bir karar destek sistemi için kavramsal çerçeve önermekte ve İstanbul’da şehir içi akıllı deniz yolu ulaşımında optimum çözüm önerisi geliştirmenin olanaklılığını sorgulamaktadır.

Derin öğrenme yönteminin bir uygulaması olan Çekişmeli Üretici Ağlar (Cycle-GAN) Esra Yağdır Çeliker, Gizem Efendioğlu ve Özgün Balaban’ın “Cycle-GAN ile Modern İç Mekânların Bilim Kurgu Ortamları Olarak Yeniden Üretilmesi” başlıklı makalesinde iç mekân ölçeğinde; Can Uzun “GAN ile Mimari Plan Üretimlerinin Değerlendirilmesi Üzerine Bir Durum Çalışması” başlıklı makalesinde mimari plan şeması özelinde; Ozan Balcı, Şemsi Barış Terzi ve Özgün Balaban’ın “Çekişmeli Üretici Ağlar (GAN) ile Harita Üretimi ve Manipülasyonu” başlıklı makalesinde ise kentsel ölçekte kullanılarak, farklı veri setlerinin işlenebilirliğini ortaya koymaktadır. Sevgi Altun ve Cem Güneş ise “Evrışimli Sinirsel Ağ (CNN)” Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması” çalışmaları ile bir derin öğrenme yöntemini tarihsel bağlam ile ilişkilendirmektedir.

Tolga Bilbey ve Mehmet Tahir Sandıkkaya, “Oyunlaştırmanın Kişisel Veri Toplanmasına Etkisi” ni deneysel bir çalışma üzerinden eleştirel bir perspektiften sorgularken, Lale Başarır “Yapay Zeka Çırağları İnsan Emsallerinden Daha İyi Performans Gösterirlerse?” başlıklı çalışması ile yapay zeka ile üretilen çıktılarının, hala “tasarım” olarak kabul edilip edilemeyeğini tartışmaya açmaktadır.

Alandaki ulusal kaynak eksikliğini gidermeye çabalamak, YZ kavramı ile yeni tanışanlara farklı perspektiflerden rehberlik etmek, Türkiye’de ve dünyada gün geçtikçe gelişen hesaplamalı düşünme ve tasarım alanında YZ farkındalığını artırmak, bu sayının temel amaçları arasındadır.

# Artificial Intelligence in Architecture

## Editorial

The third issue of JCoDe focuses on “Artificial Intelligence (AI) in Architecture” which has been used effectively in architectural design and construction/production processes in recent years. Intelligence can be defined as all of the human abilities of thinking, reasoning, perception, and conclusion. Based on the concept of intelligence, AI is the work of creating a computer model that thinks, reasons, perceives and acts, learns, and produces results like humans by using their knowledge in their repertory to solve problems on their own.

While representations and models based on formal languages were at the forefront in the first period of AI studies, when it came to the 1990s, it was necessary to review the future predictions made on the theoretical level with the phase reached in practice. Today, in addition to models of information based on formal languages and different modes of perception (such as visual, auditory, movement-based, signal-based), the situation in which the actions take place and the analysis of “formal” behaviors in an environment that cannot be represented in all aspects are also included by AI.

Conventional computer models used in architecture follow a process based on numerical and logical operations with algorithmic methods in problem-solving. These models are generally able to provide appropriate solutions to well-defined problems.

Expert systems have been used as the first application area of AI in order to produce solutions and make suggestions in solving ill-defined design problems and in areas that require expertise. Since the 1990s, the search for case-based reasoning, knowledge-based reasoning, and visual reasoning, which aims to externalize design knowledge and how the designer thinks has gained acceleration. Heuristic search methods are used to solve architectural problems that are not well defined. Current and developing AI methods, beyond being the arithmetic sum of parts, offer architecture a new dynamic and learning “black box” where all its components interact with each other.

AI studies find application in architecture together with Expert Systems, Decision Support Systems, Agent Systems, Genetic Algorithms, Swarm Intelligence, Fuzzy Logic, Artificial Neural Networks (machine learning, deep learning), Natural Language Processing, Image Processing, Data Mining, Robotics, Internet of Things, and Shared distributed AI. In this context, in the third issue of JCoDe, the uses of AI in architectural design and production processes, interdisciplinary cooperation opportunities, its integration into architectural education, and its effects on the future of architectural design are opened to discussion.

Sevil Yazıcı, in her article titled “Computational Design Informed by Natural Systems” proposes a model based on analytical analysis of natural systems to address computational design processes in architectural education, and evaluates the outputs in which are the scope of the model applied between 2018 and 2020, depending on the organizational-, performance-, process-, and motion-based calculation criteria.

Sena Kaynarkaya and Gülen Çağdaş, in their article titled “Evaluation of Metro Lines with Swarm Intelligence Approach”, evaluate slime-mould behaviors by modeling them in digital environment with the swarm intelligence approach of Göztepe-Ümraniye metro line and open to discussion the potentials of numerical models based on swarm intelligence behavior in urban scale. Moreover, Esranur Demirtaş and Şehnaz Cenani propose a conceptual framework for a decision support system regulating maritime traffic in their article titled “A Decision Support System Proposal for the Optimization of Traffic on the Bosphorus Line: E-Bosphorus.” The authors also question the possibility of developing an optimum solution proposal in urban smart sea transportation in Istanbul.

Cycle-GAN, which is an application of deep learning method, is studied in the article titled “Reproduction of Modern Interiors as Science Fiction Environments with Cycle-GAN” in interior design by Esra Yağdır Çeliker, Gizem Efendioğlu, and Özgün Balaban. The same method is used in the architectural plan scheme specific in the article titled “A Case Study on the Evaluation of Architectural Plan Production with GAN”, studied by Can Uzun. In the “Map Production and Manipulation with Contested Producer Networks (GAN)” by Ozan Balcı, Şemsi Barış Terzi and Özgün Balaban, it is revealed the processability of different data sets by using it on an urban scale. Sevgi Altun and Cem Güneş, on the other hand, associate a deep learning method with the historical context with their “Classification of Anatolian Seljuk Patterns Using Convolutional Neural Network (CNN).”

While Tolga Bilbey and Mehmet Tahir Sandıkçaya question “Effects of Gamification to Private Data Collection” in an experimental study from a critical perspective, Lale Başarır opens to discussion whether the outputs produced with artificial intelligence can still be accepted as “design” with the title “What if AI Apprentices Outperform Their Human Counterparts?”.

Striving to resolve the national lack of resources in the area, guiding the newcomers to the concept of AI from different perspectives, improving the AI awareness in the fields of computational thinking and design which develop in Turkey and in the world day by day are among the main objectives of this issue.



Computational Design Informed by Natural Systems Doğal Sistemlerle Bilgilendirilmiş Hesaplamalı Tasarım Sevil Yazıcı	01
Sürü Zekâsı Yaklaşımı İle Metro Hatlarının Değerlendirilmesi Evaluation of Metro Lines with Swarm Intelligence Approach Sena Kaynarkaya, Gülen Çağdaş	17
Boğaz Hattındaki Trafiğin Optimizasyonuna Yönelik Bir Karar Destek Sistemi Önerisi: E-Bosphorus A Decision Support System Proposal for the Optimization of Traffic on the Bosphorus Line: E-Bosphorus Esranur Demirtaş, Şehnaz Cenani	49
Cycle-GAN ile Modern İç Mekânların Bilim Kurgu Ortamları Olarak Yeniden Üretilmesi Regenerating Modern Interiors into Science Fiction Environments via Cycle-GAN Esra Yağdır Çeliker, Gizem Efendioğlu, Özgün Balaban	71
Çekişmeli Üretici Ağlar (GAN) ile Harita Üretimi ve Manipülasyonu Map Generation & Manipulation with Generative Adversarial Networks Ozan Balcı, Şemsi Barış Terzi, Özgün Balaban	95
Evrişimli Sinirsel Ağ Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması Classification of Historic Ornaments with CNN Sevgi Altun, Cem Güneş	115
Effects of Gamification to Private Data Collection Oyunlaştırmanın Kişisel Veri Toplanmasına Etkisi Tolga Bilbey, Tahir Sandıkkaya	131
What if AI Apprentices Outperform Their Human Counterparts? Yapay Zeka Çırakları İnsan Emsallerinden Daha İyi Performans Gösterirlerse? Lale Başarır	153
GAN ile Mimari Plan Üretimlerinin Değerlendirilmesi Üzerine Bir Durum Çalışması A Case Study on Evaluation of Architectural Plan Production with GAN Can Uzun	167

# Computational Design Informed by Natural Systems

Sevil Yazıcı<sup>1</sup>

0000-0002-0664-4494<sup>1</sup>

<sup>1</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

Animate and inanimate matters in nature are evaluated within systems' unity. Natural systems represent order and balance defined by computation. Mathematicians, biologists, material scientists and professionals from other fields investigate natural systems for problem-solving. Although nature has been used as a reference since historical periods, there are improper uses on how to integrate nature efficiently into the design process. The term biomimicry is used today by transcending its meaning from imitating nature towards learning from its intelligence. There is a necessity that architectural design students develop their skills on computation and evaluate natural systems from an analytical point of view to apply their finding in creative design solutions. This research is about an elective course on biomimicry developed for undergraduate architectural design education. The methodology consists of three stages, including investigation of the natural systems (1), the abstraction of the natural systems and extracting the system parameters (2), implementing the parameters in the computational design model (3). The proposed study was implemented from 2018 to 2020 into the student projects, of which outputs are discussed and grouped under four categories, including organization-, performance-, process- and motion-based computation. By examining the results, it is determined that the students gained skills in computational design and their awareness related to the natural systems were increased.

**Received:** 24.08.2020

**Accepted:** 25.09.2020

**Corresponding Author:**

sevilyazici@itu.edu.tr

Yazici, S. (2020). Computational Design Informed by Natural Systems. JCoDe: Journal of Computational Design, 1(3), 01-16.

**Keywords:** Natural Systems, Computation, Algorithms, Architectural Design Education.

# Doğal Sistemlerle Bilgilendirilmiş Hesaplamalı Tasarım

**Sevil Yazıcı<sup>1</sup>**

**0000-0002-0664-4494<sup>1</sup>**

<sup>1</sup>İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Doğada canlı ve cansız varlıkların tümü sistem bütünlüğü içinde değerlendirilmektedir. Doğal sistemler hesaplama ile tanımlanan düzen ve dengeyi temsil eder. Matematikçiler, biyologlar, malzeme bilimcileri ve farklı alanlardan profesyoneller problem çözme amaçlı olarak doğal sistemleri araştırmaktadır. Doğa, tarihsel dönemlerden beri tasarım sürecinde referans olarak kullanılsa da, doğanın tasarım süreciyle verimli bir şekilde nasıl bütünleştirileceğine dair yanlış kullanımlar bulunmaktadır. Biyomimesis terimi, günümüzde doğayı taklit etme anlamını aşarak, doğanın zekasından öğrenme anlamında kullanılmaktadır. Mimari tasarım öğrencilerinin hesaplama becerilerini geliştirerek, doğal sistemleri analitik bakış açısıyla inceleme ve bulguları yaratıcı tasarım çözümlerinde uygulama ihtiyacı bulunmaktadır. Bu araştırma, mimarlık lisans eğitiminde uygulanmak üzere geliştirilmiş biyomimesis konulu bir seçmeli ders üzerinedir. Yöntem, doğal sistemlerin incelenmesi (1), doğal sistemlerin soyutlanması ve sistem parametrelerinin çıkarılması (2), parametrelerin hesaplamalı tasarım modelinde uygulanması (3) dahil olmak üzere üç aşamadan oluşmaktadır. Önerilen çalışma, 2018 yılından 2020 yılına dek öğrenci projelerinde uygulanmış olup, çıktılar organizasyon-, performans-, süreç- ve hareket-tabanlı hesaplama olmak üzere dört grupta incelenmiştir. Sonuçlar irdelendiğinde, öğrencilerin hesaplamalı tasarım konusunda beceri kazandıkları ve doğada bulunan sistemler hakkında farkındalıklarının arttığı belirlenmiştir.

**Teslim Tarihi:** 24.08.2020

**Kabul Tarihi:** 25.09.2020

**Sorumlu Yazar:**

sevilyazici@itu.edu.tr

Yazıcı, S. (2020). Doğal Sistemlerle Bilgilendirilmiş Hesaplamalı Tasarım. JCoDe: Journal of Computational Design, 1(3), 01-16.

**Anahtar Kelimeler:** Doğal Sistemler, Hesaplama, Algoritmalar, Mimari Tasarım Eğitimi.

## 1. INTRODUCTION

Architects commonly use metaphors or analogies based on nature during their design processes. There have been continuous attempts to use nature as a reference as seen in well-known examples from historical periods, such as Crystal Palace by Joseph Paxton built in 1851 and columns of Johnson Wax Building by Frank Lloyd Wright built in 1939. Paxton and Wright were both inspired by water lily pads (Navarro-Mateu and Cocho-Bermejo, 2020). There were also some pioneers in the past, such as Gaudi, Fuller and Otto, who observed and interpreted nature wisely by developing analytical solutions for design problems. Physical models were mostly used to understand the behavior of the systems holistically.

The term biomimicry became popular and spread by J. Benyus (1997). Despite biomimicry is coined by two words as bio and mimicking, the meaning of the word goes beyond mimicking biological systems only. It is about learning from the intelligence of nature. Thus, the relationship between architecture and nature can be defined by two main approaches, including imitating the natural forms or learning from nature, in terms of its generative and behavioral properties (Selçuk and Sorguç, 2009). Biomimetic architecture is filled with a significant number of projects that are mostly imitating the formal features of natural systems. This approach does not obtain a deeper understanding of natural systems.

Architectural design is widely influenced by information technologies today. The intent is achieving a tectonic unity in design, where form, performance and material are integrated into one system, as observed in nature (Kolarevich and Klinger, 2008; Oxman, 2009). Design process obtains various stages, which differ by their details. During the process, problem-solving is identified by ill-defined problems that involve uncertainty, and well-defined ones that are for specific goals (Reitman, 1964; Suwa et. al., 1999). The computational design process is rational and critical parameters affecting design should be identified well at the beginning of the process. The use of biomimetic principles in the computational design process offers systems with high effectiveness and performance. These types of principles were implemented in a

series of built projects and tested with state-of-the-art material assemblies (Schwinn, et. al, 2012; Krieg et al. 2012; Brugnaro et al. 2016; Castriotto, et. al, 2019).

An algorithm is defined by a process, consisting of a finite number of steps. Oxman (2017) described the term algorithmic thinking as a rule set, which uses computation for designing. Algorithms are generated for specific intents and solutions (Terzidis, 2006). Integration of computation, algorithmic logic, as well as scripting and visual programming languages into the architectural education were explored in the past (Kvan, et. al. 2004; Çolakoğlu and Yazar, 2007; Celani and Vaz, 2012). Digital design models and techniques, design theory and architectural discourse were also investigated, in terms of design pedagogy (Oxman, 2008). Since natural systems represent order and balance defined by computation, there were additional attempts to incorporate multi-faceted biomimetic principles with computational design into architectural curricula by using systems' thinking (Yazici, 2015) and by generating alternatives for space architecture (Varınlıoğlu, et. al. 2018). More studies should be undertaken in architectural design education, in terms of evaluating natural principles, understanding their computational logic and learning from them for their implementation into design problems. The research question remains unanswered: How can natural systems be used analytically for problem-solving in architecture and design, by improving computational design skills, as well as algorithmic thinking skills of undergraduate architecture students? Creativity in architectural design can be supported by using natural systems as a source for learning by using computation.

## **2. METHODOLOGY**

The methodology of the course consists of three stages, including (1) investigation of the natural systems, (2) the abstraction of the natural system and extracting the critical parameters, and (3) implementing the parameters into the computational design model.

## **2.1 Investigation of the Natural Systems**

In the first stage of the course, nature should be observed and analyzed based on its properties initially. As a part of the course, lectures are given covering the state-of-the-art research and application projects in the field of biomimetic design systems. Students need to interpret nature, in terms of systems' thinking by discussing the terms, such as inputs and outputs, the transformation of inputs into the outputs, holism, goal seeking, differentiation and hierarchy (Yazici, 2011). Afterwards, students are asked to prepare a research project for a formation or an organization found in nature, such as an organism, inanimate entity or a phenomenon, by identifying the reason of their selection, parameters and rules of the system and by aiming to respond the question of, whether it can be applied to a design problem at the conceptual level.

## **2.2 The Abstraction of the Natural Systems and Extracting the Critical Parameters**

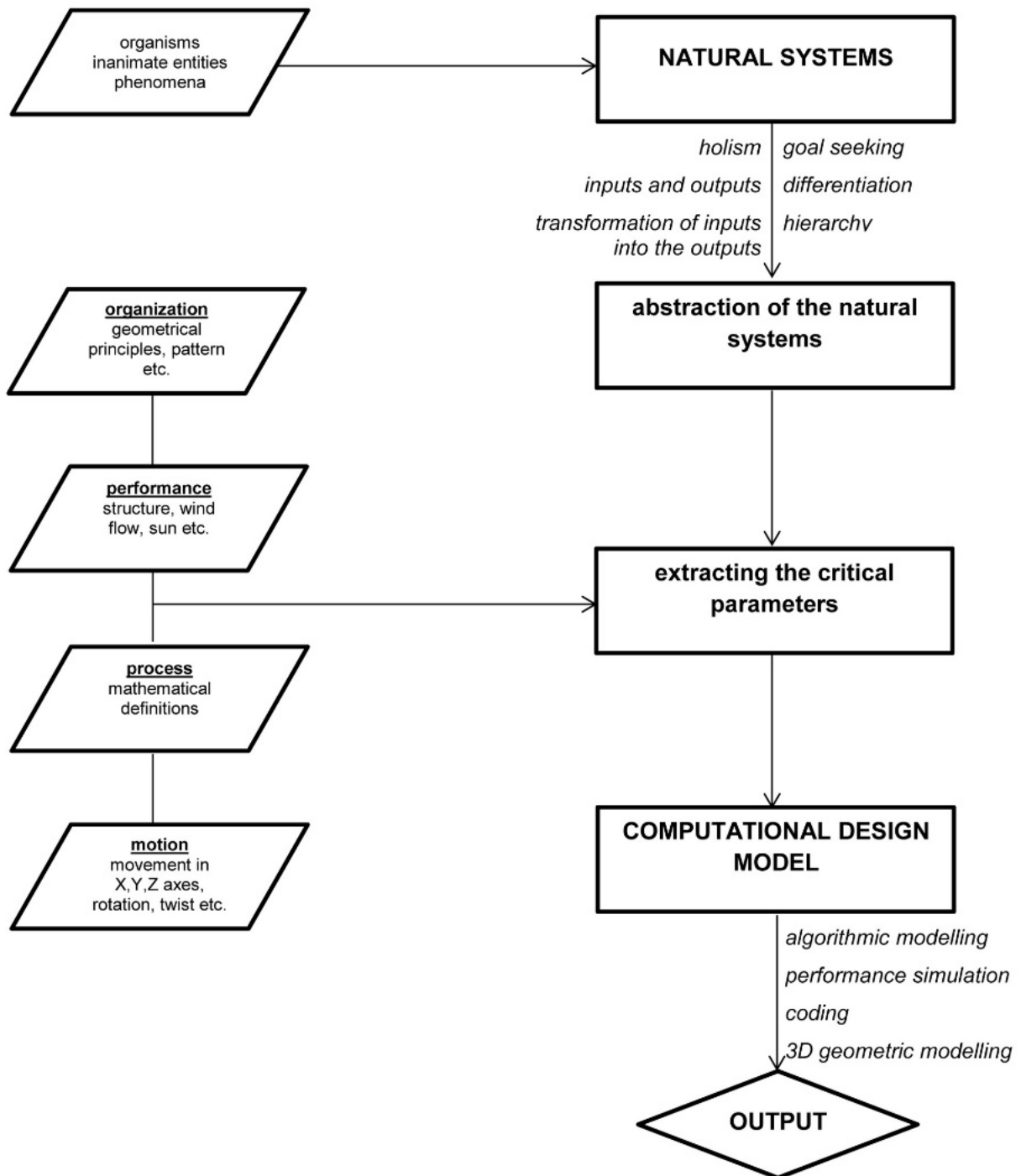
In the second stage, students are required to investigate nature from a new perspective, in which parameters, rules and relationships of the system are identified. Students should explain selected research projects with diagrams; describe the system from an analytical point of view by introducing a computational framework. This process requires a certain level of abstraction, as well as simplification of the system since natural systems are highly complex. Thus, only critical parameters of the system are extracted, by serving to the goal of individual students.

As a part of the course, some precedent work and algorithms are investigated that are extensively used in the field, such as Voronoi and Fractal algorithms (Mandelbrot, 1977, p: 4), driven by the cell morphologies and growth in nature respectively. However, students are expected to investigate all possibilities in natural systems without providing prior constraints. The selected critical parameters may be associated with features related to the organization, such as geometrical principles and pattern; or performance, related to the structure and environment, such as wind flow and sun; or process, such as mathematical definitions; or motion, such as movement in X, Y, Z axes, rotation and twist.

## 2.2 Implementing the Parameters into the Computational Design Model

In the final stage, the mathematical properties of the natural systems should be understood and re-generated by the use of computation. By investigation and abstraction of the natural systems, computational design models can be established with critical parameters. The outputs may differ according to the principles extracted from nature, related to the organization, performance, process, and motion.

Introduction on the capabilities of the computer-aided design (CAD) methods, including algorithmic modelling, performance simulation, coding and three-dimensional (3D) geometric modelling, are given and related methods are discussed in the course. There is neither a specific computational tool nor a tutorial on particular software is provided. However, participants are required to explore tool ecologies flexibly and use them creatively in their processes. The research project in the first stage needs to be implemented into a computational framework. **Figure 1** depicts the flowchart, which includes the stages of the methodology.



**Figure 1:** The methodology flowchart.



### 3. OUTPUTS: STUDENT PROJECTS

The methodology was implemented from the year 2018 to 2020 as an elective course. Since the course was open to the Faculty of Architecture and Design, students from different departments were able to join the course. Although the majority of the students were from the department of architecture, there were also participants from the departments of communication design and interior architecture. Selected seven output projects of students (S.1-S.7) were grouped under four, including organization-based, performance-based, process-based and motion-based computation. The groups were formed according to their characteristics on, how they translate natural principles into the computational design models. The computational tools, used in the student projects, were Rhinoceros for geometric modelling, Grasshopper (GH) for algorithmic modelling, Kangaroo GH and RhinoVAULT for form-finding, Autodesk Flow for Computational Fluid Dynamics (CFD) simulation, Processing for two-dimensional (2D) visual design and C# coding.

#### 3.1 Organization-based Computation

Organization-based computation is driven by organizational principles found in natural systems. S.1 used sea-corals, particularly brain corals, formed by the movements of the waves at the shores, which contain sedimental rocks. By erosion of water on the rocks, gaps are generated. They are filled with micro-creatures that use these parts for laying eggs and interbreeding. Corals act as buffers against large wave movements. The investigation of S.1 was based on the geometry and growth of corals, identified by certain rules. The computational model was developed by interpreting the growth by algorithms, which resulted in the looped generations. The code developed at C# programming language enabled direct integration to the Rhinoceros 3D geometric modelling and Grasshopper (GH) algorithmic design environments. The application was implemented into subdivision logic of a 3D surface, which started from a flat surface and transformed sequentially into a subdivided mesh without having intersecting surfaces. The two-dimensional (2D) organization was translated into a plan layout to test as well.

### **3.2 Performance-based Computation**

The projects under this group are based on the behaviour of the natural systems related to the performance criteria, such as structural or environmental performances, as presented by **S.2** and **S.3**.

Geomimicry uses geological process as a source for learning, similar to the term biomimicry that is used to examine nature for problem-solving. Landforms may vary from mountains, plains, valleys, to the canyons and deserts. Formation of land may continue thousands of years, as being informed by activities, such as erosion, plate tectonics, weathering and Aeolian processes. Dunes are made out of mixed size sand particles, formed by Aeolian or weathering processes. The formations of dunes occur in deserts and coast, where wind forces are active. The goal of **S.2** was to use environmental forces and landforms to extract critical parameters to generate a dwelling, which could be adapted to its surroundings, in extreme conditions, in deserts more specifically, Since dunes act as barriers against forces of wind and waves, the geometry of the dwelling was created accordingly, to increase the resistance of the mass against severe wind forces. The computational model was generated by using Kangaroo for GH, as physics-based spring system as the form-finding method. Additionally, Computational Fluid Dynamics (CFD) simulations were undertaken by Autodesk Flow software to assess the effects of wind on the geometry. **S.3** used spiral aloe, a type of succulent, of which leaves symmetrically grow in spirals. The plant can grow by obtaining up to 150 individual leaves, which can support and carry each other and create a self-supporting system. The participant asked the question of whether this feature could be translated into structural systems, in which neither additional supports nor fixtures might be necessary. The masonry structures were investigated further and tested by the use of RhinoVAULT, a plug-in developed for funicular form-finding of thin-shell structures in compression. The definition presented that the force distributions and optimization process could be controlled parametrically. The relationship between the geometry and active forces affecting the shell were explored through a series of tests.

### **3.3 Process-based Computation**

This group investigates the natural processes, by aiming to understand its underlying mathematical principles, as presented by **S.4** and **S.5**.

The growth of multicellular systems represents similarities in nature. Complex systems found in nature are simplified by the use of Lindenmayer systems (L-systems), used by biologist, mathematicians and computer scientists widely. Fractals are associated with the L-systems and defined by fragmented shapes that can split into parts, in which each part contains the same geometrical information as the whole. By investigation of the morphological development of the plants by **S.4**, Fractals and Pentigree L-systems were explored further in Processing environment (URL1 and URL 2). The algorithm was translated into a code in the Processing Visual Design software environment to generate 2D graphics, in which the colour of the 2D geometry can be altered by the code, as well as the iterations of the system.

A rose is also called as a rhodonea curve, defined by polar coordinates. The development of a rose flower contains several stages. In the growing process, the petals and sepals gradually open and form a blossom. The blooming process of the rose flowers, informed by a combination of the sun, wind, water, temperature, was investigated by **S.5**. A mathematical model based on a radial table was developed to identify the process, in which the diameter of the flower was associated with the sin and cos values. The ratio provided the exact stage of the blooming process. The mathematical formula was translated into a code in the Processing software environment to generate the 2D graphical pattern.

### **3.4 Motion-based Computation**

The projects under this group investigate the dynamic behaviour in nature, as presented in the projects of **S.6** and **S.7**.

Climbing plants use trees or other objects as supports to grow vertically. They are consisting of a root, stem, leaves and tendrils. Tendrils obtain a coiling morphology, which informs their behaviour to hold the hosts and stabilize their position by coiling around. Tendrils can easily stretch or contract, which provide flexibility to adapt to the surfaces of the host objects. Their length gets shorten, by curling into spirals or twisting into helixes. **S.6** focused on the capabilities of the movements of the tendrils.

The adaptive properties of tendrils were examined in an architectural design project developed by Rhinoceros 3D geometric and GH algorithmic modelling tools. The movement and geometrical features of the tendrils were interpreted and abstracted in the generation of a responsive wall, in which each module could translate in one direction. Arachnids are evolved by enabling independent movements between their legs, chest and head and by accelerating their reflexes by environmental conditions. The skeletons and movements of these insects were investigated by S.7. The system was based on abstracting of the kinetic behaviour of the arachnids and translating it into a mechanical system for the operation of a responsive facade, in which window openings could be controlled. While the conveyors in the design proposal worked in a circular motion, the openings on the facade could be controlled in one axis. The circular motion was converted into direct motion by using inverse kinematics logic.

Table 1 shows the general evaluation of student projects. Additionally, Figure 2 and 3 depict the outputs according to the computational design models, tools and type of computation. The output projects underlined that natural systems can be used as a source for learning, in terms of organization, performance, process and motion-based computations. Thus, the study extends beyond that nature is commonly used for formal resemblance in design projects.

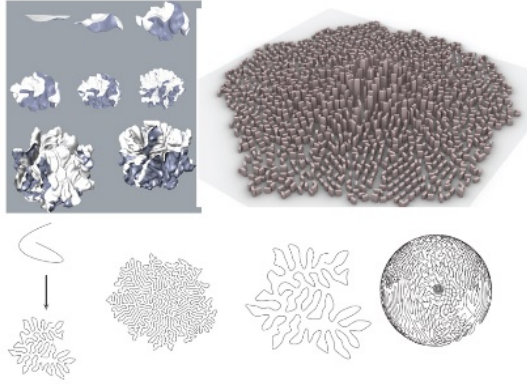
Table 1: The evaluation of student projects.

Figure 2: The projects from S.1 to S.4, computational design models, tools and type of computation (Image credits from “a” to “d”: URL 3-5; Shiffman, 2012, chapter 8).

Student ID	Natural systems	Abstraction of the natural system into the computational model	Tools	Output	Type of computation
S.1	Corals	Growth algorithm	Rhinoceros,GH, C# Code	Mesh surface	Organization-based computation
S.2	Dune formations	Physics-based spring systems/ CFD analysis	Rhinoceros,GH Kangaroo, Autodesk flow	Dwelling	Performance-based computation
S.3	Spiral aloe	Funicular form-finding	Rhinoceros, RhinoVAULT	Compressive shell structure	Performance-based computation
S.4	Growth of plants	Fractals/ L-systems algorithm	Processing	2D graphics	Process-based computation
S.5	Roses	Blooming algorithm	Processing	2D graphics	Process-based computation
S.6	Tendrils	Kinetic movement	Rhinoceros,GH: Translation of components	Adaptive wall	Motion-based computation
S.7	Arachnids: Insect skeletons	Kinetic movement	Rhinoceros, GH: Attractor curves	Mechanical system of an adaptive facade	Motion-based computation

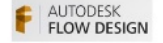
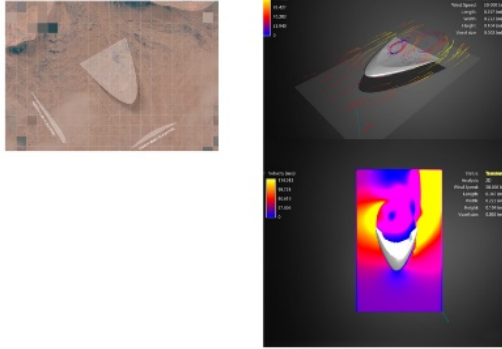
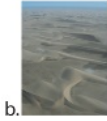
**Student ID    Natural Systems    Computational Design Models    Tools / Type of Computation**

S.1



**organization-based computation**

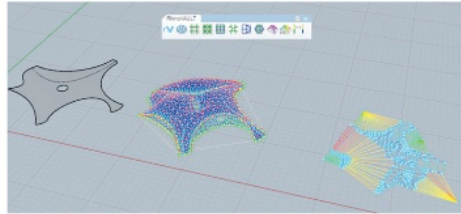
S.2



**CFD analysis**

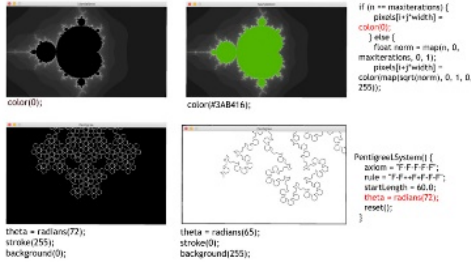
**performance-based computation**

S.3



**performance-based computation**

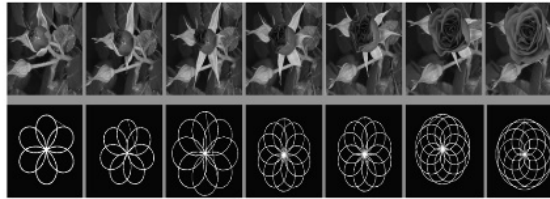
S.4



**process-based computation**

**Student ID    Natural Systems    Computational Design Models    Tools / Type of Computation**

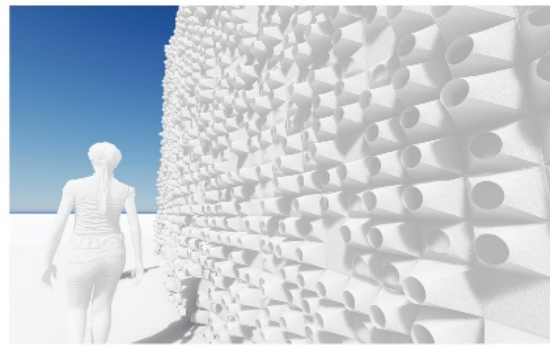
S.5



**Processing**  
*visual design*

**process-based  
computation**

S.6



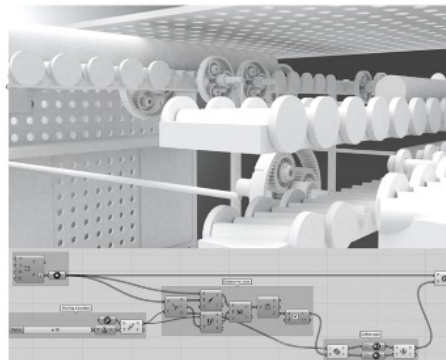
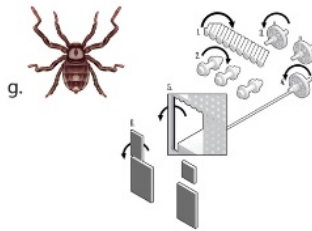
**Rhino**  
*geometric modelling*



**Grasshopper**  
*algorithmic modelling*

**motion-based  
computation**

S.7



**Rhino**  
*geometric modelling*



**Grasshopper**  
*algorithmic modelling*

**motion-based  
computation**

#### 4. CONCLUSIONS

**Figure 3:** The projects from S.5 to S.7, computational design models, tools and type of computation (Image credits from “e” to “g”: URL 6-8).

All matters in nature are evaluated within systems’ unity. The intelligence of nature is considered as a source for problem-solving used by scientists, engineers, architects and designers. Although architects have been using nature as a reference since historical periods, there is a large amount of misunderstanding on, how to use

natural principles in the design process efficient and not to imitate its aesthetics only. There is a necessity to increase awareness of architects, as well as architectural design students for this issue, by teaching the computational logic behind the natural systems.

This research shares the methodology and outputs of an elective course related to the biomimicry, which was implemented in undergraduate architectural education from the year 2018 to 2020. By evaluating the output projects, four groups were identified, differ in how natural principles are used and translated into the design process by using computation. This enabled to transcend the task beyond, how nature is used in the common architectural design process. The main difficulties encountered in the process for the students were to investigate natural systems with an analytical point of view and to generate an abstraction of the system by simplifying the system parameters because nature is highly complex. The other difficulty was that most of the students were not skilled, in terms of computational design. For future research, training on the computational tools and techniques might be given a priori to the course, to increase the capabilities of the students for establishing computational design systems efficiently and rapidly.

### **Acknowledgements**

The studies were applied at Özyeğin University, Faculty of Architecture and Design, from the year 2018 to 2020. Special thanks to the students for their contribution to the course (Student names from S.1 to S.7 respectively: Benson Sanga, Tamer Kumaş, Özgün Şimşek, Burak Can Çatan, Ayşe Yasemin Şenalp, Doğa Çakmakçı, Mert Aydın).

### **References**

- Benyus, J. (1997). *Biomimicry: Innovation Inspired by Nature*, William Morrow and Company, New York.
- Brugnarò, G., Baharlou, E., Vasey, L., and Menges, A. (2016). Robotic softness: An adaptive robotic fabrication process for woven structures. In: *36th the Association for Computer Aided Design in Architecture (ACADIA) Conference Proceedings, Ann Arbor* (pp. 154–163).
- Castriotto, C., Giantini, G. and Celani, G. (2019). Biomimetic Reciprocal Frames A design investigation on bird's nests and spatial structures, *eCAADe 37 / SIGraDi 23 Conference Proceedings*, Volume 1, 613-620.
- Celani, G. and Vaz, C., E., V. (2012). CAD Scripting And Visual Programming Languages For Implementing Computational Design Concepts:A Comparison

From A Pedagogical Point Of View, *International Journal of Architectural Computing* 1(10), 121-137.

Çolakoğlu, B. and Yazar, T. (2007). Mimarlık Eğitiminde Algoritma: Stüdyo Uygulamaları, *Journal of the Faculty of Engineering and Architecture of Gazi University* 22(3), 379-385.

Kolarevich, B., and K. Klinger. (2008). Manufacturing Material Effects Rethinking Design and Making Architecture, 196–198. New York: Routledge.

Krieg, O. D., Mihaylov, B., Schwinn, T., Reichert, S., and Menges, A. (2012). Computational Design of Robotically Manufactured Plate Structures Based on Biomimetic Design Principles Derived from Clypeasteroidea, *Digital Physicality, 30th Education and Research in Computer Aided Architectural Design in Europe (eCAADe) conference proceedings* (pp. 531–540). Prague.

Kvan, T., Mark, E., Oxman, R. and Martins, B. (2004). Ditching the Dinosaur: Redefining the Role of Digital Media in Education. *International Journal of Design Computing*.

Mandelbrot, B., B. (1977). *The Fractal Geometry of Nature*, W. H. Freeman and Company, New York.

Navarro-Mateu, D. and Cocho-Bermejo, A. (2020). Evo-Devo Strategies for Generative Architecture: Colour-Based Patterns in Polygon Meshes, *Biomimetics*, 5, 23; doi:10.3390/biomimetics5020023.

Oxman, N. (2009). Material-based Design Computation: Tiling Behavior. *ReForm: Building a Better Tomorrow, Proceedings of the 29th Annual Conference of the Association for Computer Aided Design in Architecture*. Chicago, pp. 122-129.

Oxman, R. (2008). Digital Architecture as a Challenge for Design Pedagogy: Theory, Knowledge, Models and Medium. *Design Studies*, 29, 99-120.

Oxman, R. (2017). Thinking difference: Theories and Models of Parametric Design Thinking. *Design Studies* 52(2017), 4–39.

Reitman, W. (1964). Heuristic Decision Procedures, Open Constraints, and the Structure of Ill-Defined Problems, in: M. Shelly and G. L. Bryan (eds), *Human Judgement and Optimality*, (pp. 282–315). New York: John Wiley and Sons

Schwinn, T, Krieg, O D, Menges, A, Mihaylov, B and Reichert, S (2012). Machinic Morphospaces: Biomimetic Design Strategies for the Computational Exploration of Robot Constraint Spaces for Wood Fabrication, *Proceedings of the 32nd Annual Conference of the ACADIA*.

Shiffman, D. (2012). *The Nature of Code*. Chapter 8. ISBN-13: 978-0985930806.

Suwa, M., Gero, J., and Purcell, T. (1999). Unexpected Discoveries and Inventions of Design Requirements: A Key to Creative Designs, in: *Computational Models of Creative Design IV*.

Selçuk, S. A and Sorguç, A. G. (2009). Exploring Complex Forms in Nature Through Mathematical Modeling: A Case on *Turritella Terebra*. *Proceedings of eCAADe 27*.



Terzidis, K. (2006). Algorithmic architecture. Oxford: Elsevier.

Yazici, S. (2011). Computing through Holistic Systems Design Method: Material Formations Workshop, *Dearq Journal of Architecture*, Universidad de Los Andes, 90-101.

Yazici, S. (2015). A Course on Biomimetic Design Strategies, *33rd eCAADe 2015 Conference*, Vienna University of Technology, 16-18 September 2015, Vienna, Austria, 111-118.

URL 1 <<https://processing.org/examples/mandelbrot.html>>

URL 2 <<https://processing.org/examples/pentigree.html>>

URL 3 <<https://www.nationalgeographic.org/encyclopedia/dune/#coastal-dunes>>

URL 4 <<https://oceanservice.noaa.gov/facts/brain-coral.html>>

URL 5 <<https://garden.org/plants/view/117109/Spiral-Aloe-Aloe-polyphylla/>>

URL 6 <<https://www.britannica.com/plant/rose-plant>>

URL 7 <<https://www.britannica.com/science/tendril>>

URL 8 < <https://www.britannica.com/animal/arachnid>>

# Evaluation of Metro Lines with Swarm Intelligence Approach

Sena Kaynarkaya<sup>1</sup>, Gülen Çağdaş<sup>2</sup>

0000-0001-2345-6789<sup>1</sup>, 0000-0001-8853-4207<sup>2</sup>

<sup>1</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

<sup>2</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

Computer technologies have an important place in the fields of architecture and urban design as in every field today. It is possible to see that they are largely inspired by the life cycles of living organisms, when we look closely at the development processes of these technologies in the fields of architecture and urban design.

Slime mould which are examples of emerging living systems; shows an example of swarm intelligence, which arises from basic needs and produces results very different from individual behavior. Rail systems that significantly reduce the burden of today's urban transportation networks, are constantly changing and renewed, especially in large cities. When we look at the examples of which slime mould are used, it is seen that they have a very important role in the design of metro lines and highway systems.

In the article, it is aimed to evaluate an existing metro line by modeling slime mould behaviors with swarm intelligence approach in digital environment. Based on the studies conducted to evaluate the conformity of the metro lines and highways design criteria, this study will work on the Göztepe-Ümraniye metro line in Istanbul, which is still under construction. The stations on the line were kept constant and the model was expected to create a route, and the compliance of the route with the metro design criteria was evaluated. At points that do not comply with the design criteria, commands are introduced on the model via Python code, and the similarities and differences between the route created by the swarm and the existing route are compared.

**Keywords:** Swarm Intelligence, Slime Mould, Railway Design.

**Received:** 24.08.2020

**Accepted:** 30.09.2020

**Corresponding Author:**

senakurkcuoglu@hotmail.com

Kaynarkaya, S. & Çağdaş, G. (2020). Evaluation of Metro Lines with Swarm Intelligence Approach. JCoDe: Journal of Computational Design, 1(3), 17-48.

# Sürü Zekâsı Yaklaşımı İle Metro Hatlarının Değerlendirilmesi

Sena Kaynarkaya<sup>1</sup>, Gülen Çağdaş<sup>2</sup>

0000-0001-2345-6789<sup>1</sup>, 0000-0001-8853-4207<sup>2</sup>

<sup>1</sup> İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

<sup>2</sup> İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Bilgisayar teknolojileri günümüzde her alanda olduğu gibi mimari ve kentsel tasarım alanlarında da önemli yer tutmaktadır. Bu teknolojilerin mimari ve kentsel tasarım alanlarında gelişim süreçlerine daha yakından bakıldığında büyük ölçüde canlı organizmaların yaşam döngülerinden esinlenildiğini görmek mümkündür.

Beliren canlı sistem örneklerinden olan cıvık mantarlar; temel ihtiyaçlarından doğan ve bireysel davranışlarından çok farklı sonuçlar doğuran sürü zekası örneği sergilemektedir. Günümüz kent içi ulaşım ağlarının yükünü önemli ölçüde hafifleten raylı sistemler, özellikle büyük kentlerde sürekli olarak değişmekte ve yenilenmektedir. Cıvık mantarların kullanıldığı örnekler bakıldığında, metro hatları ve otoyol sistemlerinin tasarımında oldukça önemli bir rolü olduğu görülmektedir.

Makalede sürü zekası yaklaşımı ile cıvık mantar davranışları sayısal ortamda modellenerek mevcut bir metro hattının değerlendirilmesi amaçlanmıştır. Metro hatlarının ve otoyolların tasarım kriterlerine uygunluğunu değerlendirmek için yapılan çalışmalardan yola çıkılarak, İstanbul kentinde yapımı devam etmekte olan Göztepe-Ümraniye metro hattı üzerinde çalışılmıştır. Hat üzerindeki istasyonlar sabit tutulup modelin bir güzergâh oluşturması beklenerek ve oluşan güzergâhın metro tasarım kriterlerine uygunluğu değerlendirilmiştir. Tasarım kriterlerine uygunluk göstermeyen noktalarda model üzerinde Python kodu aracılığı ile komutlar tanıtılıp, sürünün oluşturduğu güzergâh ile mevcut güzergâh arasındaki benzerlik ve farklılıklar karşılaştırılmıştır.

**Anahtar Kelimeler:** Sürü Zekası, Cıvık Mantarlar, Metro Hattı Tasarımı.

**Teslim Tarihi:** 24.08.2020

**Kabul Tarihi:** 30.09.2020

**Sorumlu Yazar:**

senakurkcuoglu@hotmail.com

Kaynarkaya, S. & Çağdaş, G. (2020). Sürü Zekası Yaklaşımı ile Metro Hatlarının Değerlendirilmesi. JCoDe: Journal of Computational Design, 1(3), 17-48.

## 1. GİRİŞ

Mimarlık alanında biyolojiden esinlenme her zaman söz konusu olmuştur. Bilgi ve iletişim teknolojilerinin mimari ve kentsel tasarım alanlarında kullandığı yaklaşımlarda ve modellerde canlı organizmaların yaşam döngülerinden ve davranışlarından esinlendiğini görmek mümkündür. Hem biçimsel hem de ilişkisel bazda karşılaştırıldıklarında, mimarlık ve biyolojinin morfolojiyle yakından ilişkili olduğu görülmektedir. Ayrıca her iki disiplin de bütün halinde çalışan bireysel öğelerden var olur ve bu öğeler arasında sürekli bir etkileşim söz konusudur. Tüm bu benzer özelliklerinin yanı sıra biyolojik süreçler daha hareketli ve üretkendir. Gödel'in (1931) Eksiklik Teorisinde kullandığı "Bazen belirli bir disiplindeki bir sorunu çözmek için tamamen farklı bir bölgeye geçmek zorundasınız." ifadesinden yola çıkıldığında, disiplinler arasındaki iş birliği çoğu zaman kaçınılmazdır.

Doğadaki canlıların, özellikle sürü davranışı sergileyen kuşlar, balıklar ve karıncaların davranışlarının bilgisayar modelleriyle temsili, pek çok kentsel ve mimari tasarım probleminin çözümünde önemli bir esin kaynağı olmuştur.

Makalede, sürü zekâsı yaklaşımıyla, bilgisayar ortamında geliştirilen bir modelle mevcut metro hatlarının güzergahının değerlendirilmesi amaçlanmıştır. Bu amaçla, öncelikle literatürde sürü davranışının fiziksel ortamda benzetimini yapan araştırmalar incelenmiş; daha sonra da sürü davranışı ve bu davranış bağlamında yaşam döngüsüne sahip civik mantar davranışları incelenerek bilgisayarda sayısal bir model geliştirilmiştir. Modelin geliştirilmesinde Rhinoceros / Grasshopper ve Python kodu kullanılmıştır. Model mevcut bir metro hattının verileri ile uygulanmış ve sonuçlar kıyaslanarak değerlendirilmiştir.

## 2. LİTERATÜR

Günümüzde biyolojik organizasyonların yaşam döngüleri birçok alanda araştırmacılara ilham kaynağı olmuştur. Sürü davranışı sergileyen canlıların yaşamlarını sürdürmek adına gerçekleştirdikleri temel ihtiyaca dayalı basit davranışların, karmaşık sonuçlara neden olduğu gözlemlenmektedir. Bu sistemlerin kendini örgütleyerek sürü zekâsı yaklaşımıyla merkezi bir otoriteye sahip olmadan problem çözme davranışında bulunmaları, ulaşım ve iletişim gibi mühendislik hesapları,

finansal yapılanmalar, benzetim modelleri ve yapay zekâ teknolojilerine referans olmuştur.

## 2.1 Sürü Zekâsı

Sürü zekâsı; çok sayıda etmenin ortaklaşa gerçekleştirdikleri davranışlar sonucu meydana gelen davranıştır. Sürü zekâsı, ilk olarak 1898 yılında Gerardo Beni, Susan Hackwood ve Jing Wang tarafından beliren fenomenleri tanımlamak için oluşturulan robotik sistemlerde kullanılmıştır (Beni, Wang, 1990).

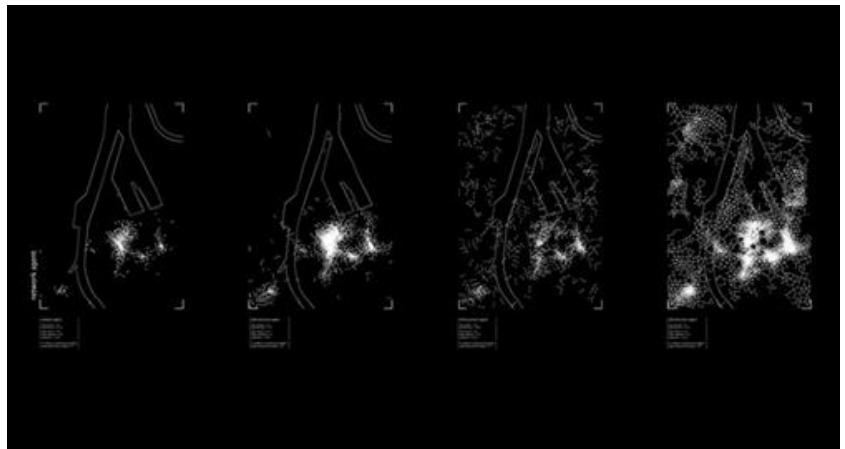
Sürü zekâsı; çevrelerinin farkında, kendini etraftan soyutlamamış, özerk veya yarı özerk olan ve küresel davranışlar ortaya çıkarabilecek etmenlerden oluşur. Bu davranış ile merkezi bir otorite gerekmeden, sadece etmenlerin etkileşimi ile tanımlanan problemin çözümü mümkündür. Sürü zekâsı öz-örgütlenme ve Stimerji kavramları ile ilişkilidir. Stimerji; karınca kolonilerinin ve cıvık mantarların yaşam döngülerinin alt yapısını oluştururken, aynı zamanda sürüdeki bireylerin iletişim kurmalarını sağlayan etkileşimlerin bütünüdür (**Şekil 1**). Stimerji kavramında bilginin dağıtımı esastır. Değişen çevre koşullarından etkilenen etmenin bıraktığı iz sayesinde, tüm sistem bu etkilenmeden haberdar olur ve kendini yeniden organize eder. Karıncaların bıraktıkları feromon, Stimerji örneklerinden birisidir. Öz-örgütlenme, herhangi bir dış kontrol ve yönlendirme olmadan bir sistemin iç düzenine ait belirli bir fonksiyonunun gelişmesi için çevreye adapte olma sürecine denir (Hensel, v.d., 2004). Öz-örgütlü sistemlerde gelişim; rastgelelik, geri bildirim ve etkileşimlerle olur. Dolaylı etkileşim barındıran Stimerji, öz-örgütlenmeyi besleyen bir kavramdır. Sürünün kendini örgütleyerek çalışması ile merkezi otorite ihtiyacı kalmamış olur. Yerel etmenlerin örgütlü etkileşimleri küresel sonuçlar doğururken aynı zamanda sistemi dengede tutar. Sürüyü oluşturan bireylerin her biri özdeştir. Otonom davranışlara sınırlı sahip olmaları ve grup olarak hedefe hizmet edebilmeleri küresel davranışı ortaya çıkarır (**Şekil 1**).

**Şekil 1:** Öz-Örgütlülük: Kuş sürüleri (Url-1).



Günümüzde belirme teriminin dijital hesaplamalarda ve sosyal davranışları açıklamalardaki kullanımı oldukça hız kazanmıştır. Şehirlerin beliren akıllı sistemler olarak görülmesi ve bir tasarımcıya bağlı kalmadan şekillendiği düşüncesi ile sürü istihbaratının merkezi bir kontrol mekanizması olmadan kendini organize eden yapısının gösterdiği benzerlik, şehircilik için önemli bir referans kaynağı olmuştur.

Sürü zekâsı ve bu istihbarata dayanan algoritmaların şehircilikte kullanımının artması tasarımcı iradesine dayanan planlamanın yerini bireylerin etkileşimine bırakmıştır. Bu bağlamda master plan üretmek yerine master algoritma üretme fikri ile yola çıkılarak, Roland Snooks ve Robert Stuart-Smith tarafından Melbourne’ de tasarlanan Swarm Urbanism projesinin amacı öncelikle ulaşım ve altyapı ağlarını tasarlamaktır. Çalışma ile tasarımı haritalandırmaktan ziyade, etmenlerle elde edilen kolektif tasarım bir sisteme dönüştürülmek istenmektedir (Şekil 2).



**Şekil 2:** Swarm Urbanism.

Sistemi organize etmek için genel etmen türü kullanılmamıştır. Bunun yerine etmenlerle küresel özelliklerin elde edilebileceği bir ekoloji

oluşturulmuştur. Etmelerin özelliklerinin yanı sıra yoğunluklarının sağladığı hiyerarşik yapı da tasarım sürecinde etkin rol oynamıştır.

Sürü zekasının en kritik noktalarından olan etmen sayısının fazlalığı, kent sistemlerinin barındırdığı birçok unsurun, kentin şekillenmesine olan etkisini açıklamayı kolaylaştırmaktadır.

## **2.2 Cıvık Mantarların Davranışı ve Uygulama Örnekleri**

Doğada, pek çok canlı yaşam faaliyetlerini sürdürebilmek için besin arayışına girerler. Bu arayışı gerçekleştirirken çevrelerine adapte olan ve büyüyen stratejiler geliştirirler. Sahip oldukları basit yaşam döngüleri ile daha küresel sistemler oluşturan ve çevresiyle etkileşim kurarak durumunu değiştiren canlı türlerinden biri de 'Physarum polycephalum' türü olan 'Cıvık Mantarlar'dır. Mantarlar dağınık besin kaynakları arasındaki iletimi yapmak için doğrudan besinler arasında ana bağlantılar ve ek bağlantılar sağlayan ağ hattına sahiptirler.

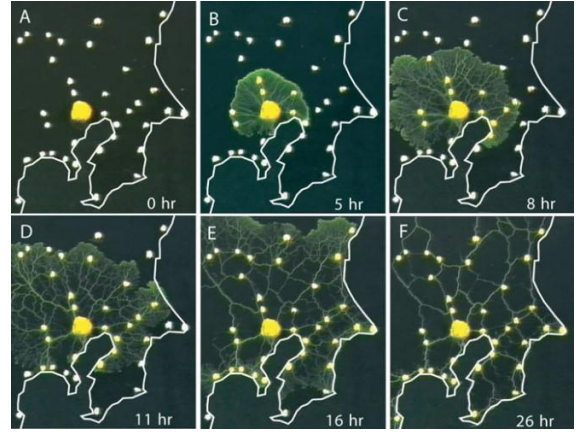
Cıvık mantarlarda besin aktarımı için gelişmiş olan ağ yapısı; taşınan besin miktarı ve taşıma performansı bakımından en verimli olan tüplerden oluşur. Verimliliğin yanı sıra bu bağlantıların en önemli özelliği, besin kaynakları arasındaki en kısa mesafeyi kurmalarıdır.

Tek hücreli canlı organizmalar olan cıvık mantarlar; herhangi bir sinir sistemine sahip değildir. Merkezi bir yönlendirme ve beyin

fonksiyonu barındırmamalarına rağmen, öğrenme ve öğrendikleri bilgileri birleştikleri anda diğer hücrelere aktarma potansiyelleri ile birçok araştırmaya konu olmuşlardır.

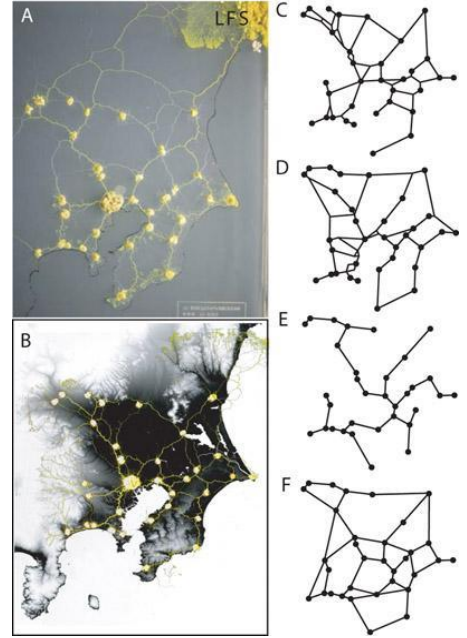
Cıvık mantar hücrelerinin, Nagakaki ve Toth (2000) tarafından yapılan labirent çözme deneyindeki başarısından sonra Tokyo metrosu üzerine bir grup araştırmacı çalışma başlatmıştır (American Association, 2010). Mühendislerin uzun mesailer harcayarak tasarladıkları bu ağ haritalarını cıvık mantar kolonilerinin üretip üretemeyeceği soruları üzerinden gidilerek deneysel birtakım çalışmalar yapılmıştır. Tokyo ve çevresindeki şehirlerin haritası üzerinde cıvık mantar yayılımı gözlemlenmiştir (**Şekil 3**).

**Şekil 3:** Tokyo metrosu üzerinde zamana bağlı cıvık mantar gelişimi (Url-2).



Cıvık mantarlar tarafından üretilen ağın Tokyo metrosuna oldukça benzer şekilde oluştuğu gözlemlenmiştir. Metro ağı ile mantar ağı arasındaki benzerliğin bu kadar fazla olması, cıvık mantar davranışının matematiksel bir model barındırdığını düşündürmüştür. Tokyo metro hattı üzerinden incelemelere devam edildiğinde, ilk etapta mantar hücrelerinin ağlarını her yere gönderdiği gözlemlenmiştir. Zamanla besin kaynakları arasındaki ağların en sağlam ve kısa olanı korunarak diğer tüpler geri çekilmiştir (**Şekil 4**).

**Şekil 4:** Zamana bağlı olarak Plazmodyum tüplerinin değişimi (Url-3).



Çalışmalar sonucunda, bu basit organizmaların uzun hesaplamalar ile üretilen bir hattı çok kısa bir sürede oluşturmuş olması, tasarım aşamasının yanında, gelecekte, doğal afet ya da savaş gibi durumlarda



farklı ağ arayışlarında hızlı bir şekilde alternatif geliştirebilecek bir alt yapı sunmaktadır. Ayrıca merkezi olmayan bir irade ile gelişen sistem, yapay zekâ gibi teknolojilere algoritmik bir altlık sağlayabilecektir.

Adamatzky, dünya üzerindeki çeşitli ülkelerin otoyol ağlarının oluşumlarını civık mantarlar üzerinden değerlendiren birçok çalışma yapmıştır. Çalışmalarda belirlenen ülkelerin yoğun nüfuslu, ekonomik potansiyeli yüksek ve ulaşım açısından kritik olan şehirleri seçilerek, civık mantar ağları ile bu ülkelerin otoyol ağları karşılaştırılmıştır. Otoyol ağlarının karşılaştırılması için yapılacak deneye başlamadan önce civık mantarların gelişimi için uygun ortam hazırlanmıştır. 120x120 mm boyutlarında kare ya da 90 mm çapında yuvarlak polistiren petri kaplarına içme suyu serpilmiş kâğıt havlu yerleştirilmiştir. Yulaf gevreği konularak civık mantarlar yetiştirilmek üzere gevreklerin üzerine yerleştirilmiştir. Deneyi yapılacak bölgenin haritası belirli ölçekte küçültülerek agar plakalar halinde kesilmiştir (**Şekil 5**). Hattın geçtiği şehirlerin yerlerine yulaf gevreği yerleştirilmiştir. Deneyin görselleştirildiği zamanlar dışında, deney ortamı karanlık ve sıcaklığı 22-25° C dir. Deney, sonuçlara göre ortalama 20-25 sefer tekrarlanmıştır. Deney kapları tarayıcıda taranıp doygunluk seviyesi yükselttilerek görselleştirilmiştir (Adamatzky, Akl, 2011).



**Şekil 5:** İngiltere'nin agarlardan kesilmiş haritası (Url-4).

Kesilen agar plakalar üzerinde seçilen başlangıç şehrindeki yulaf üzerine civık mantar aşılansak deney başlatılmıştır. Toplanma eylemini tetikleyen yulafın yanı sıra engel olacak bölgelerden mantarların geçmemesi için, civık mantarları uzaklaştırıcı etkisi bulunan kristal tuz kullanılmıştır. Otoyol hattının mevcut hali H grafiği olarak kabul edilerek deneye başlanmış ve civık mantar grafiği tanımlanırken aşağıdaki kabuller yapılmıştır:

- U: Kentsel alan kümesi.
- E: Set kenarı.
- P: Cıvık mantar grafiği
- $w: E[0,1]$  : Kenarların olası ağırlığı. E' nin her bir kenarı bir olasılıkla ilişkilendirilir.
- P(Q): Eşik değeri. U bölgesindeki noktalar arasında oluşan tüplerin kalınlığıdır. Tüp kalınlıkları ve ağırlıkları ile orantılıdır. Q değeri değiştiğinde noktaların bağlanma şekillerinde değişiklik gözlenmektedir.
  - $P(Q) = 0$  iken H grafiği ham haldedir.
- $T(E) = e$  ve  $E: w(e) > Q$  iken Q' ya eşit veya daha az ağırlığa sahip olan tüm kenarları kaldırır.

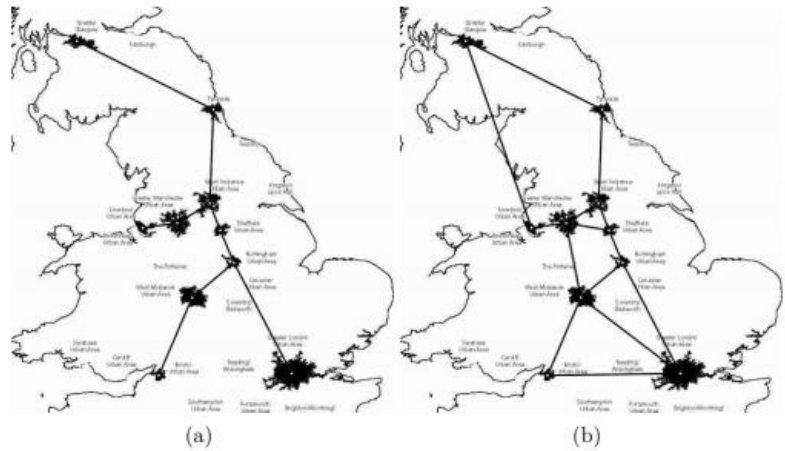
Cıvık mantar grafiği şu şekildedir:

$$P(Q) = (U, T(E), w)$$

Cıvık mantar grafiğinden elde edilen sonuçlar otoyol yapımlarında kullanılan Düzlemsel Yakınlık Grafikleri (Gabriel Grafiği, Göreceli Mahalle Grafiği) ile karşılaştırılarak sonuçlar elde edilir (Adamatzky, Prokopenkob, 2011).

Bu çalışmalar, İngiltere (Adamatzky, Jones, 2009), (**Şekil 6**), Meksika (Adamatzky, A., 2010), Kanada (Adamatzky, Akl, 2011), Avustralya (Adamatzky, Prokopenkob, 2011), Hollanda (Adamatzky, v.d., 2012), Çin (Adamatzky, v.d., 2013), gibi ülkelerin otoyolları için tekrar edilmiş ve büyük oranda benzerlikler olduğu belirtilmiştir.

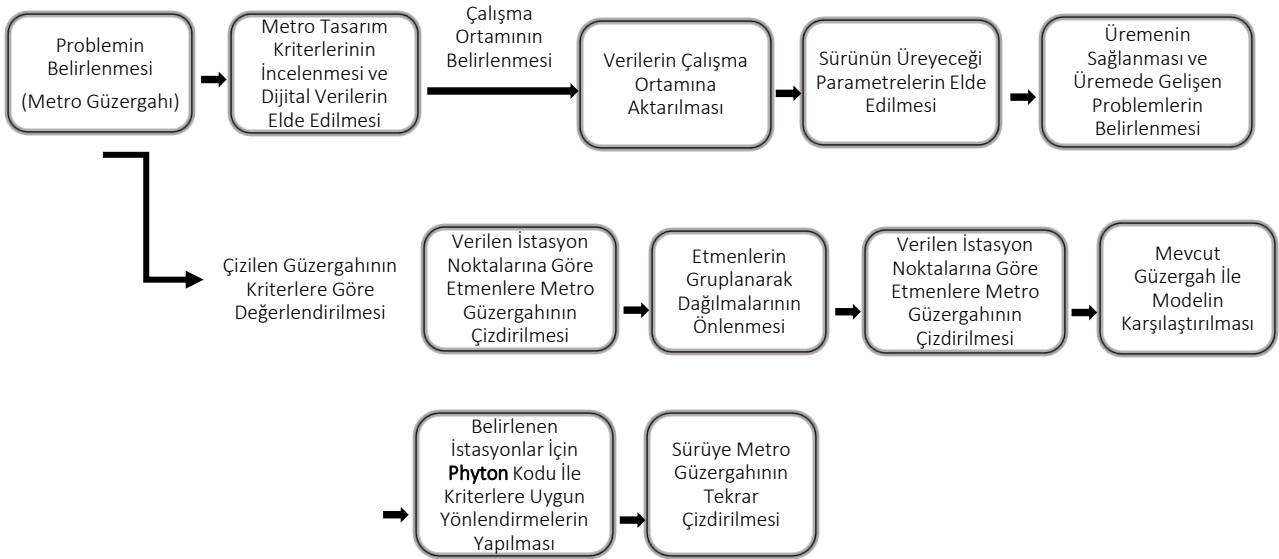
**Şekil 6:** (a) Canlı organizmaların dağılımını incelemek için kullanılan İngiltere haritası (b) Gabriel grafiği karşılaştırması (Url-5).



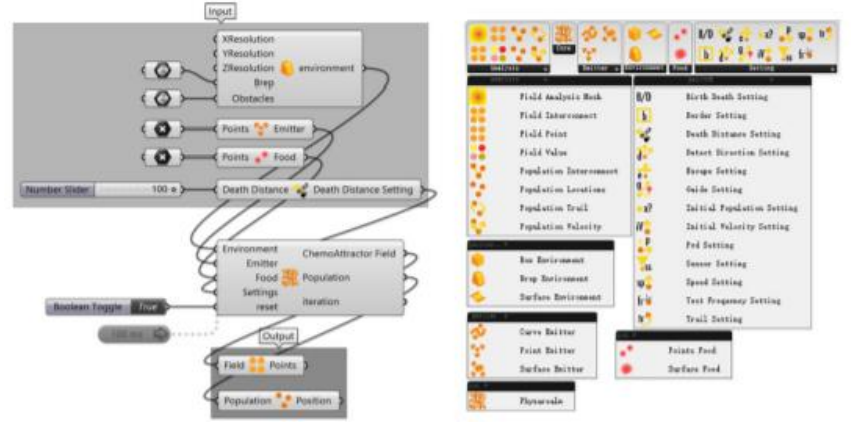
### 3. METRO HATTI DEĞERLENDİRME YÖNTEMİ

Yürütülen çalışmada sonuç ürüne ulaşırken çeşitli aşamalardan geçilmiştir. Problemin belirlenmesi ve probleme nasıl yaklaşıp çözüme nasıl ulaştırılacağı gibi kararlar süreçle beraber şekillenmiştir. Bu nedenle tezin anlaşılabilirliğini artırabilmek adına Şekil 7 de verilen diyagram oluşturulmuştur. Tez çalışması ve buna bağlı model üretilirken belirme kavramı üzerinden gidilerek, bu kavrama referans veren bir yaşam döngüsüne sahip olan civik mantarların, tez çalışmasındaki benzer birçok çalışmada kullanılması üzerinden gidilmiştir. Alınmak istenen sonuç ürüne verimli şekilde ulaşılacak bir çalışma ortamı belirlenerek eklenti içerisinde bulunan parametreler kontrollü şekilde değiştirilmiştir. Değişimlerden elde edilen geri dönüşler sonucunda parametreler güncellenerek sürekli tekrar eden bir süreç izlenmiştir. Sürünün üremesi için uygun parametreler elde edildikten sonra, popülasyon çoğalırken ortaya çıkan problemler belirlenerek, problemlerin çözümleri üzerine gidilmiştir. Sürünün çoğalırken gösterdiği fazla yayılım ve metro tasarım kriterleri de göz önüne alınarak civik mantar sürüsünün çizdiği metro güzergahı gözden geçirilerek, belirli kısıtlar ve yönlendirmeler ile güzergâh etmenlere tekrar tekrar çizdirilmiştir.

Şekil 7: Çalışmada İzlenen Yöntem.



Etmenlerin aralarındaki uzaklık, konumları arasındaki açılar ve iz miktarının yoğunluğuna göre farklı üreme şekilleri ve doğum oranı göstermeleri öncelikle kullanılacak programda geri bildirim ihtiyacı doğurmaktadır. Bu nedenle çalışmanın modelini oluşturmak için ortam olarak; eş zamanlı çıktı alma ve parametrik modelleme fırsatı sunan Rhinoceros-Grasshopper tercih edilmiştir. Grasshopper, Rhinoceros'a entegre edilmiş bir modelleme aracıdır. David Rutten tarafından geliştirilen Grasshopper, sürekli olarak parametreleri güncelleme ve güncelleme sonucu ortaya çıkan değişiklikleri anında görüntüleme imkânı sunar. Programın verdiği eşzamanlı geri bildirimler sayesinde geometri üzerinde yapılan değişiklikler, kaydırıcılar sayesinde sürekli olarak güncellenmektedir. Oluşturulan algoritma setleri adım adım birbirine bağlanarak modelin çalışması sağlanmaktadır. Modelleme Rhinoceros'ta yapılabileceği gibi, doğrudan Grasshopper üzerinden de yapılabilmektedir. Modelin geliştirilmesinde, Grasshopper'a gömülerek çalışan ve civik mantarlar için üretilmiş olan Physarealm eklentisi kullanılmıştır. Physarealm; etmen tabanlı algoritmalar ile üretilmiş bir eklentidir. Civik mantarların yaşam döngüsünü açıklayan kuralları kullanarak laboratuvar ortamında yapılan deneylerin, dijital ortamda benzetiminin yapılabilmesi için kullanılan bir yazılımdır. Altı bileşen grubundan oluşmaktadır: Analysis, Core, Emitter, Environment, Food, Settings (Şekil 8).

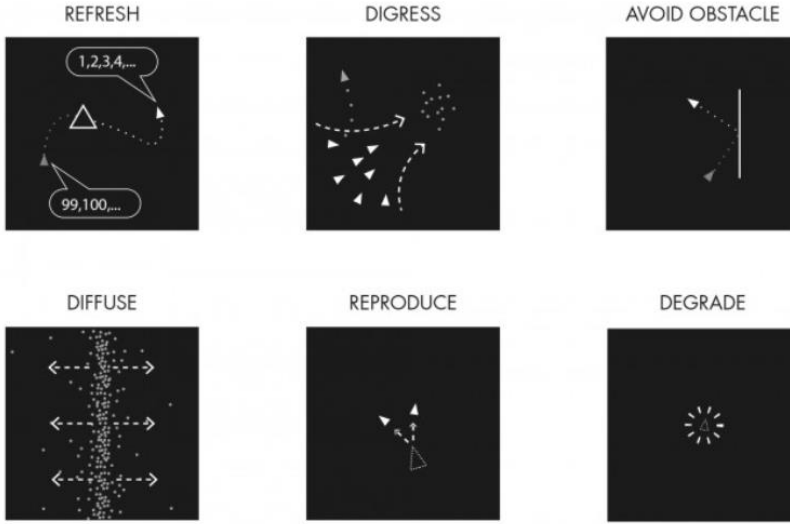


Şekil 8: Physarealm Bileşenleri (Url-6).

Physarealm eklentisini oluşturan bileşenlerden Environment bileşeni; etmenlerin üreyeceği ortamı tanımlarken, Emitter bileşeni; koloninin üremesi için ortama bırakıldığı ilk noktayı ve metro hattının başlangıç istasyonunu temsil etmektedir. Box, brep ya da point ile temsil edilebilen Emitter bileşeni ile mantarların yöneleceği diğer besin kaynaklarını, metro hattının diğer istasyon noktalarını temsil eden Food

bileşeni de Core noduna beraber bağlanarak sistemin çalışmasını sağlamaktadır.

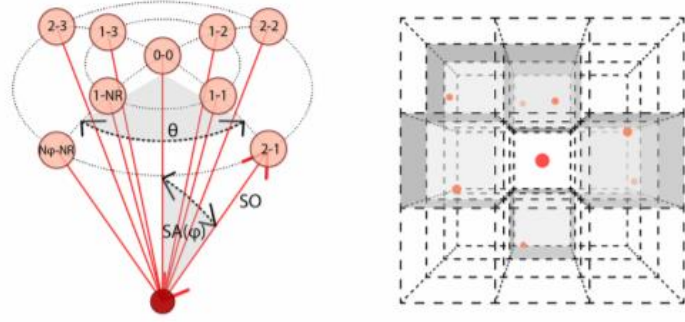
Settings bileşeni; mantar davranışlarını kontrol eden parametreleri ekleyebileceğimiz alt bileşenleri içermektedir. Cıvık mantar davranışlarını formülleştiren Jeff Jones'un (2010) çalışmaları referans alınarak, Setting bileşeninin içindeki parametrelere optimum başlangıç değerleri atanmıştır. Physarealm eklentisi ayarlarının temelini birkaç davranış modeli oluşturmaktadır (**Şekil 9**).



**Şekil 9:** Physarealm davranış modelleri (Url-7).

Jeff Jones'ın (2010) modelinden esinlenerek üretilse de sonuç olarak iki model birbirinden oldukça farklıdır. Sürüyü oluşturan etmenlerin üreme ortamları, kafeslere bölünmüş bir küptür (**Şekil 10**). Etmenlerin davranış modelleri, bu küpün içerisinde şekillenir. Birbirlerine göre konumları ve bu konumlar arasındaki açılar Physarealm eklentisindeki parametrelerin temelini oluşturmaktadır. Sensör davranışında; etmenler birbirlerinin salgıladığı kemo-atraktif maddeyi algılamalarını sağlayan sensörlere sahiplerdir. Davranış bileşenini dört parametre oluşturur: Etmenin Z ekseninde hareket ettiği bir koordinat düzlemi, her sensöre radyal mesafede olan Algılama Dengesi (SO), maksimum kutup açısında olan Algılama Açısı (SA) ve aynı Z koordinatında bulunan etmenleri algılamayı sağlayan sensör sayısından oluşan Yön Algılama (NPhy)dir.

**Şekil 10:** Davranışın gerçekleştiği ortam ve davranış açıları (Url-8).


















Doğum ve ölüm davranışında (Adaptasyon); Etmen sayısı çok fazlaştığında ortamdaki etmenlerin bir kısmı ölür. Buna göre algoritma yeni etmen sayısına göre davranışı günceller. Kurallar yakın yarıçap içerisindeki etmenlere göre çalışır. Doğum için yarıçap  $DvR$ , ölüm için ise  $DeR$ 'dir (**Tablo 3**). Alanın yarıçapı  $(2 * DvR + 1)$  ve  $(2 * DeR + 1)$ 'dir. Bu nedenle, kontrol alanı, yarıçap=1 ise  $3*3*3=27$ 'dir. Komşu etmenlerin sayısı  $N$ 'dir.  $DvMin > 0$  ve  $DvMax < (2 * DvR + 1)^3$  ise büyüme gerçekleşmez.  $DeMin > 0$  ve  $DeMax < (2 * DeR + 1)^3$  ise etmen sayısı azalır. Örneğin, algoritma 26 komşu hücreyi sayar ve  $DeR = 1$  olduğunda başka herhangi bir etmenin bunları işgal edip etmediğini kontrol eder. Komşu hücrelerin sayısı  $DvMin$  ve  $DvMax$  arasında ise etmen kendini çoğaltır. Kemoatraktan iz davranışında; küpü oluşturan kafesler  $(3*3*3)$  ölçülerindedir.  $TrRat$  parametresi besin üzerinden geçen etmenin bıraktığı izin miktarıdır. İzin miktarına göre ortaya çıkan davranış ve buna bağlı olarak oluşan kurgu da değişir. Ana davranış parametreleri dışında sürü davranışını değiştiren birkaç parametre daha bulunmaktadır:

**Dikey Kılavuz Faktörü (VGF):** Etmenin hareketini etkileyen parametredir. Yüksek bir VGF ile etmenler daha dikey olarak hareket etme eğilimindedir; bu da sütunlar gibi formları bulmak için yararlıdır.

**Başlangıç Hızı (IV):** Bu parametre, akın gibi çok-etmenli sistemlerde etkisi daha fazla olan bir faktördür. Cıvık mantarlarda akın durumu görülme de bu davranış yönlü etmenlerin başlatılması için önemlidir.

**Kaçış Olasılığı (EscP):** Parametre değerinin büyük olması etmenlerin belirlenen sınırlardan kaçmasına veya belirlenen engellere girmesine neden olur.

Fiziksel deney ortamlarında kullanılan materyallere benzerlik ve metro hattının temsiline uygunluğu açısından Physarealm içerisinde üreme ortamı olarak “Surface Environment”, aşılama ve besin kaynaklarını göstermesi için ise “Point Food ve Point Emitter” seçilmiştir. Geri kalan bileşenler üreme kuralları, ortam boyutları ve istasyonlar arası mesafeye uygun olarak belirlenmiştir. Modelin çalışacağı yüzey, etmenlerin hareket edeceği noktalar ve ilişkileri düzenlemek için Grasshopper komutları kullanılmıştır (Tablo 1).

	BİLEŞEN	GÖRSEL	İSİM	ID	TANIMLAMA	YER
G R A S S H O P P E R	PARAMS		POINT	Pt	Bu bileşen nokta veya noktaları temsil eder.	Geometry
			SURFACE	Srf	Bu bileşen yüzey veya yüzeyleri temsil eder.	
			PANEL	-	Bileşenleri değerlendirir ve bu değerlendirmeyi bağlanma sırasına göre listeler.	Input
			NUMBER SLIDER	-	Sayı değerini değiştiren kaydırıcıdır.	Primitive
			BOOLEAN TOGGLE	Toggle	Tek bir doğru ve yanlış değer arasında hızlı bir şekilde geçiş yapmanızı sağlar.	
			COLOUR SWATCH	Swatch	Renk kartelası.	
			TIMER	-	Parametreleri bazı aralıklarla değiştirebilir ve sonucu animasyonlu bir film gibi görebilirsiniz.	Util
	MATHS		LARGER THAN	Larger	Büyüktür bileşeni iki veri listesi alır ve A Listesinin ilk öğesinin B Listesinin ilk öğesinden büyük olup olmadığını belirler. İki çıktı, iki listeyi daha büyük bir değere göre değerlendirmek isteyip istemediğinizi belirlemenizi sağlar.	Operators
			SMALLER THAN	Smaller	Küçüktür bileşeni, Büyüktür bileşenin ters hareketini gerçekleştirir. Küçüktür bileşeni A listesinin B listesinden küçük olup olmadığını belirler ve bir boole değerleri listesi döndürür.	
	SETS		CULL PATTERN	Cull	Bir kafesden alınacak yüzler.	Sequence
			GRAFT TREE	Graft	Her öğeye fazladan bir dal ekleyerek bir veri ağacı oluşturur.	Tree
	VECTOR		CLOSEST POINTS	CPs	Bir nokta koleksiyonunda en yakın noktaları bulur.	Point
	SURFACE		SPHERE	Sph	Küresel bir yüzey oluşturun.	Primitive
			BREP CLOSEST POINT	BrepCP	Bir brep üzerindeki en yakın noktayı bulur.	Analysis
DISPLAY		CUSTOM PREVIEW	Preview	Oluşturulan şeklin renklendirilmesini sağlar.	Preview	

Tablo 1: Grasshopper’ da kullanılan bileşenlerin listesi.

### 3.1 Değerlendirilecek Metro Hattına Ait Veriler

Makale kapsamında yapılacak çalışma için, elde edilebilecek verilerin güncelliği nedeni ile Göztepe-Ümraniye metro hattı pilot hat olarak belirlenmiştir. İstanbul metrosu tasarım kriterleri ve standartları kitapçığından, metro hattı tasarlanırken dikkat edilmesi gereken hususlar ile ilgili veriler elde edilmiştir. İstanbul Metropolitan Alanı Kentsel Ulaşım Ana Planı doğrultusunda; ilk durağı, son durağı ve durak sayısı belirli olan hat bu kriterlere göre şekillenmektedir. İki durak arası mesafenin maksimum uzunluğu, aracın çıkabileceği en yüksek eğim

değeri ve dönebileceği virajın yarıçapı gibi, her hat için uygulanan genel kurallar ve hattın geçtiği çevrenin durumu, istasyon noktalarını belirlerken dikkat edilen unsurlardır.

Metro A.Ş Etüt Projeler Müdürlüğünden alınan iki adet avan proje ve kesin proje arasında yapılan karşılaştırmalar ile istasyon noktalarını belirlerken dikkat edilen kriterlerin öncelik sıralaması yapılmıştır. Göztepe-Ümraniye hattı tasarlanırken belirlenen kriterler şunlardır: Entegrasyon, Yoğunluk (Okul, alışveriş merkezi, hastane), İmar durumu (Tablo 2).

	İstasyon	Kriter
1	Göztepe 60. Yıl Parkı	Caddebostan sahili lastikli araç yolu entegrasyonu
2	Göztepe	Marmaray entegrasyonu
3	Sahrayıcedid	Minibüs caddesi lastikli araç yolu entegrasyonu
4	Yeni Sahra	Kadıköy-Tavşantepe metro entegrasyonu
5	Ataşehir	İki durak arası maksimum mesafe ve imar durumu
6	Finans Merkezi	Cazibe merkezi
7	Soyak Yenişehir	İki durak arası maksimum mesafe ve imar durumu
8	Atakent	İki durak arası maksimum mesafe ve imar durumu
9	Çarşı	Üsküdar-Ümraniye metro entegrasyonu
10	Hastane	Ümraniye E. A. Hastanesi
11	Kazım Karabekir	Son durak

**Tablo 2:** Göztepe-Ümraniye hattı tasarım kriterleri.

### 3.2 Metro Hattı Değerlendirme Modeli ve Uygulanması

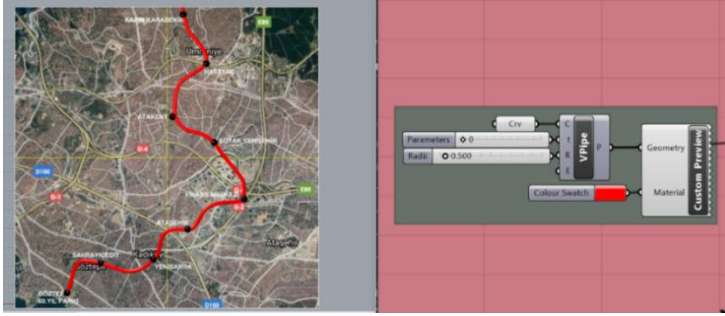
Bu çalışmada referans canlı olarak belirlenen civık mantarların sürü zekâsı ile ortaya çıkardıkları davranışların esas dayanağı komşuluk ilişkileri ve bu ilişkiden doğan etkileşimlerdir. Civık mantarları temsil eden etmenlerin ortamdaki üreme durumları, ulaşabilecekleri en yüksek sayı ve birbirlerine olan uzaklıkları nedeni ile gerçekleşecek ölümleri, beliren fenomenlerde ortaya çıkan komşuluk ilkesiyle doğrudan ilişkilidir. Makale kapsamında geliştirilen model, metro güzergahları üzerinden yapıldığı için üreme ortamı yüzey olarak seçilmiştir. Eğitim göz ardı edilerek çalışma yapılmıştır. Jeff Jones' un (2010) çalışmasında ve diğer uygulamalarda piksel olarak belirlenen ortam boyutu, bu çalışmalardan farklı olarak santimetre cinsinden



verilmiştir. Parametre ve denklem hesapları piksele dayalı yapıldığı için bazı parametreler bu çalışmaya birebir uymamıştır. Modelin geliştirildiği ortam 90x90cm boyutlarında 1/10.000 ölçeğinde küçültülerek oluşturulmuştur. Metro istasyon yerleri, Metro AŞ.' den alınan veriler doğrultusunda aslına uygun olarak yerleştirilmiştir (Şekil 11). İstasyon ve güzergâh temsilleri şöyledir:

● İstasyon noktaları

■ Güzergâh



Şekil 11: Mevcut hattın Grasshopper' da temsili.

Metro AŞ.' den alınan projenin AutoCad çizimleri Rhinoceros ortamına aktararak istasyon noktaları ve hat görsel olarak daha okunaklı hale getirilmiştir. AutoCad çizimi üzerinden, Grasshopper'da bulunan 'Pipe' bileşeni ile hattı oluşturan eğri, bir tüpe temsil edilmiştir. Yapılan sayısallaştırma üzerinden sürünün oluşturduğu hattın mevcut hatta olan benzerliği ve mevcut hatla arasındaki farklılıklar gözlemlenmiştir. İlk olarak mevcut hat ile Grasshopper'da oluşturulan modeldeki sürünün başlangıç parametreleri dışında herhangi bir müdahale bulunmaksızın oluşturdukları metro hattı karşılaştırılmıştır. Daha sonra belirli noktalara konulan engeller ile sürünün yönelimine bağlı olarak hattın gelişimi gözlemlenmiştir.

Physarealm eklentisinde "Surface Environment" bileşeni için kullanılan başlıca beş ayar bileşeni bulunmaktadır. Her ayar bileşeni kendi içinde üremeyi etkileyen birçok parametre içermektedir. Bu parametreler kemoatraktör izin diğer etmenler tarafından algılanabilmesi ve davranışlarının değişmesi üzerinden oluşturulmuştur (Tablo 3, Tablo 4). Benzetim sürecinin başlangıcında belirlenen etmen sayısı 70'tir. Sürü nüfusunun davranışı etkilemesi nedeniyle parametrelere farklı değerler atanarak elde edilen değerler üzerinden davranışlar gözlemlenmiştir.

**Tablo 3:** B/D Setting Parametre Tanımları.

---

### B/D Setting

---

DvR = Doğum mesafesi yarıçapı

DvMin = DvR içinde olan etmenlerin birbiri ile olan en kısa mesafesi

DvMax = DvR içinde olan etmenlerin birbiri ile olan uzun mesafesi

DeR = Ölüm mesafesi yarıçapı

DeMin = DeR içinde olan etmenlerin birbiri ile olan en kısa mesafesi

DeMax = DeR içinde olan etmenlerin birbiri ile olan en uzun mesafesi

---

---

### Speed Setting

---

Max Speed = Etmenlerin hareket edeceği en yüksek hız

---

### Death Distance Setting

---

Death Distance = Pozitif değerdeki ölüm uzaklığıdır

---

### Sensör Setting

---

Sensor Angle (SA)= Etmenlerin algıladığı maksimum kutup açısıdır.

Rotate Angle (RA) = Etmen en yüksek kemoatraktör miktarının olduğu yere değil dönme açısının olduğu yöne gider.

Sensor Offset (SO)= Sensörün algıladığı aralığın en yüksek değeridir.

---

### Detect Direction Setting

---

DDirR = Algılanan konik yapının alt çemberinin bölüneceği nokta miktardır.

DDirP = Algılanan konik yapının alt çemberinin bölüneceği açıdır.

---

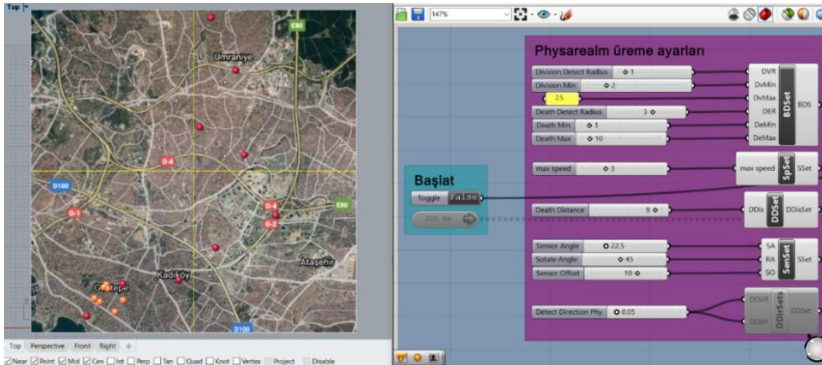
**Tablo 4:** Speed Setting, DDis. Setting, Sensör Setting ve Detect D. Setting Parametre Tanımları.

Eklentinin geliştiricilerinin verdiği başlangıç verileri üzerinden model geliştirilmeye başlanmıştır. Ortam boyutunun büyüklüğü etmen sayısını ve kullanılan parametreleri oldukça değiştirmektedir. B/D Setting içerisinde bulunan Dvmin ve DvMax parametreleri, ortam boyutuna uygun olarak yapılandırılarak üremenin sağlıklı bir şekilde gerçekleşmesi hedeflenmiştir. Simülasyonun rahatça takip edilebileceği en uygun hız seçilmiştir. Sensor Setting değerleri geliştiricilerin önerdiği değerlere yakın olarak belirlenmiştir (**Tablo 5**).

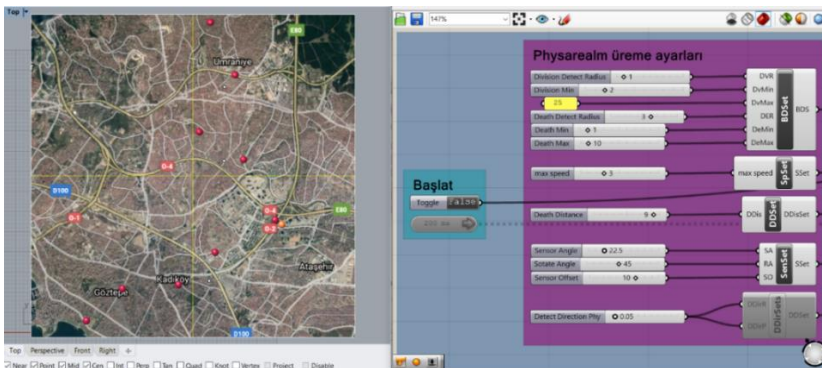
B/D Setting	Speed Setting	Death Distance Setting	Sensor Setting
DvR=1			
DvMin=2			SA=22.5
DvMax=25	Max Speed=3	Death Distance=9	RA=45
DeR=3			SO=10
DeMin=1			
DeMax=10			

**Tablo 5:** Simülasyon-1' in parametreleri

Değerler tanımlanıp simülasyon başlatıldığında etmenler başlangıç istasyonundan diğer istasyonlara doğru çoğalarak ilerlemeye başlamıştır (**Şekil 12**). Modelin ilk çalışma aşamasında az miktarda çoğalma eğilimi gösterebilir de dördüncü istasyona ulaşmadan sayıları azalmış ve tüm etmenler ölmüştür (**Şekil 13**). DvMax parametresinin yüksek olması etmenlerin uzak mesafelerde de üremesine sürünün bir anda kontrolsüzce büyümesine ve bir anda ölmesine neden olmuştur.



**Şekil 12:** Simülasyon 1-a



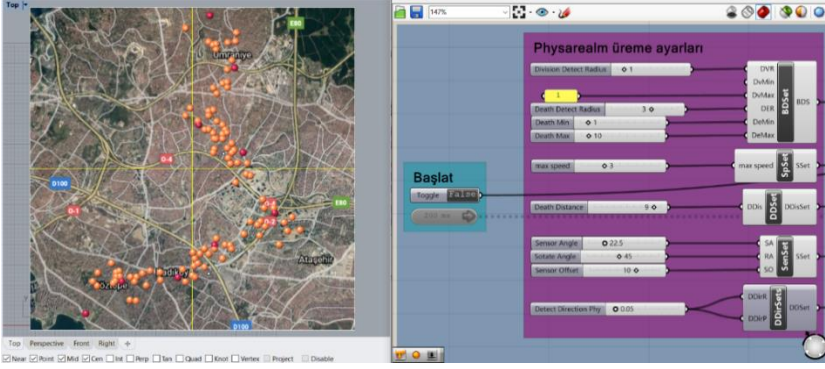
**Şekil 13:** Simülasyon 1-b

Modelin çalışması sürecinde, parametrelerin bazıları sabit tutularak, bazılarının değerleri değiştirilerek simülasyon sonuçları izlenmiştir; bu süreçte yedi simülasyon gerçekleştirilmiştir. Modelin ana parametreleri olan DvMin, DeMin ve SA parametrelerinin değeri etmenlerin kemoatraktör izi takip edebilmesi açısından oldukça önemlidir. Bu yüzden DvMin, DeMin ve SA parametreleri yedinci simülasyonda değiştirilerek etmenlerin davranışları tekrarlanmıştır (**Tablo 6**). Tanımlanan son parametreler ile etmenlerin düzenli bir şekilde üremeye devam ettikleri ve son istasyona kadar ulaştıkları görülmüştür (**Şekil 14**).

B/D Setting	Speed Setting	Death Distance Setting	Sensör Setting	Detect Direction Setting
DvR=1				
DvMin=0			SA=22.5	
DvMax=1	Max Speed=3	Death Distance=9	RA=45	DDirR=0.05
DeR=3			SO=10	DDirP=0.05
DeMin=1				
DeMax=10				

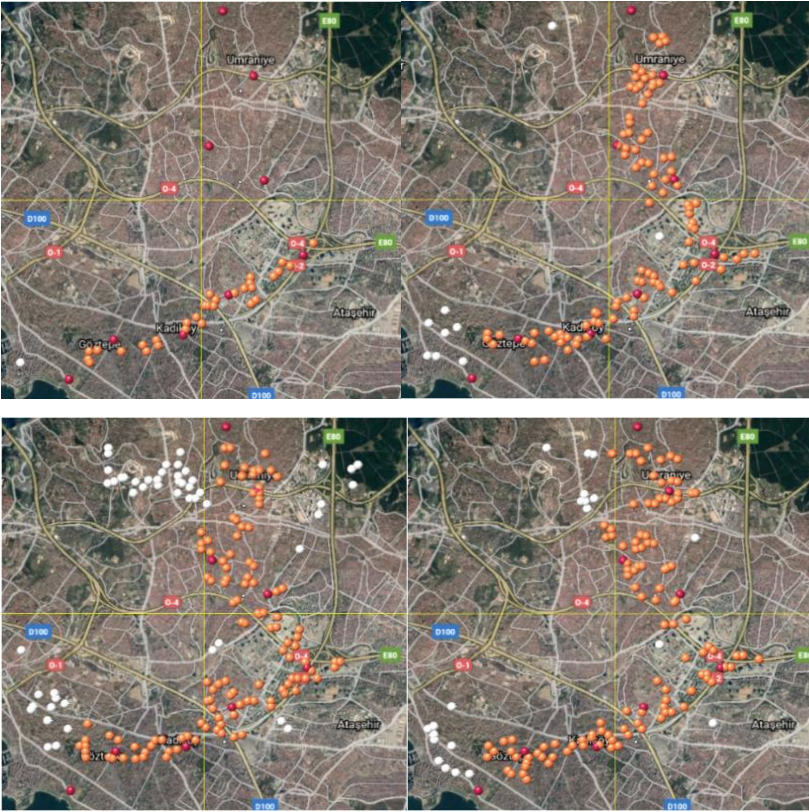
**Tablo 6:** Simülasyon-7' in parametreleri

Parametreler değiştirildikten sonra, modelin tekrarlı çalıştırılması sonucunda civık mantarların istasyonlar arası düzenli bir ilerleme kaydettiği ilmiştir. B/D Setting parametreleri arasında çok fazla fark olmaması dağılım ve buna bağlı olarak sürünün yok olmaması için önemlidir. Çünkü doğum yarıçapının en az ve en fazla değeri arasındaki fark arttığı zaman etmenlerin birbirinden uzaklaşarak izleri algılayamamasına neden olmuştur. Fakat kemoatraktör miktarının çok yüksek olması etmenlerin hareketi için tek başına en önemli parametre değildir. SA parametresi bu yüzden iz miktarı ile beraber çalışarak etmenlerin üremesine yön verir.



Şekil 14: Simülasyon-7.

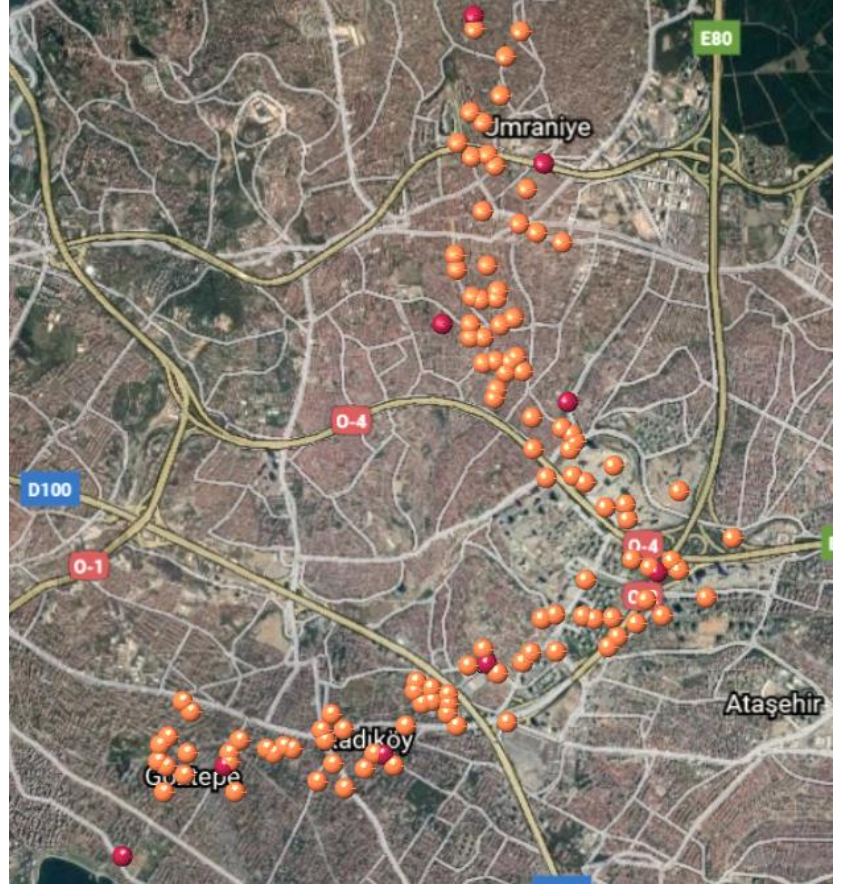
Simülasyon başladıktan sonra modelin çalışmasında bazı eksiklikler olduğu saptanmıştır. Tüm etmenler istasyonlara sırayla uğrarken aynı zamanda dağılmalar meydana gelmeye başlamıştır (Şekil 15).



Şekil 15: Etmenlerin zamana bağlı olarak ilerlemesi.

Etmenlerin zamana bağlı olarak ilerlemesinde tespit edilen dağılımları giderebilmek için öncelikle aralarında ne kadar mesafe olduğunda dağılımların yaşandığı tespit edilmiştir. Bunun için öncelikle sürüyü oluşturan etmen yoğunluğu azaltılıp Grasshopper'da bulunan operatörlerin yardımı ile gruplamalar yapılmıştır. Closest Points

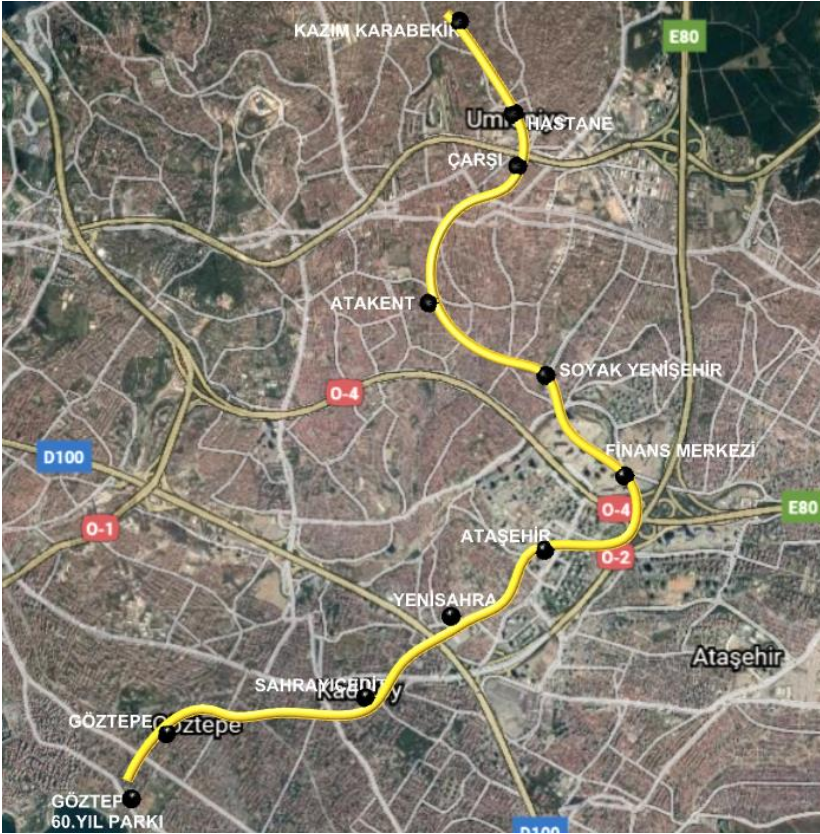
parametresi, bir nokta bulutu içerisinde belirli bir noktaya yakın olan elemanlarla grup oluşturur. Gruptaki eleman sayısı opsiyoneldir. Sürü Closest Points bileşeni ile sadeleştirilmiş ve sonrasında Smaller Than bileşeni ile aralarındaki mesafeye göre kalan etmenler tekrar gruplanmıştır. Mesafesi 4.00 birimden küçük olan etmenler simülasyonda bırakıldığında dağılımların olmadığı gözlemlenmiştir (Şekil 16).



Şekil 16: Gruplanmış etmenlerin yayılımı.

Yayılmının metro güzergahı üzerinden araştırıldığı düşünüldüğünde, anlaşılmasını kolaylaştırmak adına temsili değiştirilmiştir. Çok fazla etmen bulunması ve güzergâh incelemesi yapılabilmesi için sürü elemanları daha da sadeleştirilerek, belirlenen parametreler ile civık mantar sürüsüne tekrar güzergâh çizdirilmiştir. Smaller Than bileşeninden gelen noktalar kendi içlerinde önce gruplanmış, listesi çıkarılmış ve bu listedeki noktalardan çokgen oluşturularak, çokgenlerin merkez noktaları seçtirilmiştir. Daha sonra merkez noktaları tekrar listeye dönüştürülüp, nokta olarak çıktı vermesi sağlanmıştır.

Elde edilen daha az sayıda noktasal etmenler Interpolate Curve ile birbirine bağlanıp Pipe komutu ile kalınlık verilerek, durak noktaları arasında cıvık mantarların kendi davranış şekilleri ile oluşan metro güzergahı ortaya çıkarılmıştır (Şekil 17).



Şekil 17: Modelde üretilen metro güzergahı.

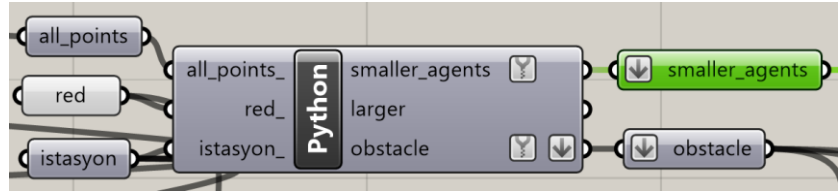
Modelde etmenlerin oluşturduğu güzergâhın geçtiği bölgelere bakıldığında, istasyon çıkışları, metro güzergahı, raylı sistem tasarım kriterleri ile beraber değerlendirildiğinde üç istasyon için modeldeki etmenlere müdahale edilmesi gerekliliği ortaya çıkmıştır:

- Göztepe durağı: Metro çıkışlarının kamusal alanlara yakınlığı söz konusu olduğu için metro güzergahının Göztepe Parkına doğru kaydırılması gerekmektedir.
- Sahrayı Cedit durağı: Entegrasyon ilkesi metro tasarımının ana ilkelerinden olduğu için Minibüs caddesindeki lastikli araç yolu ile metro güzergahının entegre olması kaçınılmaz bir durumdur.
- Yeni Sahra durağı: Metro istasyonları maliyet açısından değerlendirildiğinde, kamulaştırmanın az olduğu bölgelere yapılmaktadır. Bu nedenle Metro Marketler Zincirinin binasının

bulunduğu yerden geçiyor olması maliyeti artıracığı için güzergahın başka doğrultuya kaydırılması gerekmektedir.

Belirlenen istasyonlarda yapılacak müdahalelerin tek adımda olabilmesi için de Grasshopper içerisinde bulunan GhPython bileşeninden faydalanılmıştır. Model içerisine Python kodu eklemeye olanak sağlayan GhPython bileşeni isteğe göre artırılıp azaltılabilen iki giriş ve iki çıkış noktası bulundurmaktadır. Giriş ve çıkış noktaları yazılacak kod parçasına göre isimlendirilmektedir. Gereksiz düğüm yükü oluşturmak yerine GhPython bileşenine bağlanan girdiler ile bileşenin çıktı düğümlerinden daha kolay sonuç elde edilmiştir (**Şekil 18**).

**Şekil 18:** Python bileşeni, girdi ve çıktıları.

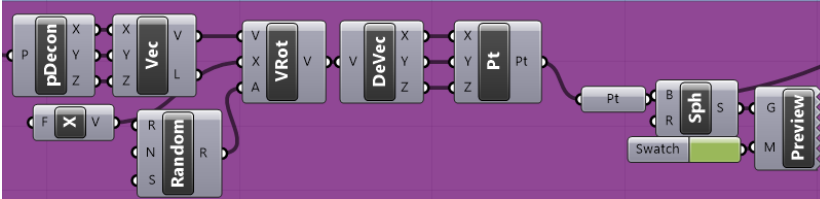


Yapılan bu çalışmada mevcut güzergâh üzerinden gidildiğinde, civık mantarların laboratuvar deneylerinde alınan verimin karşılığının bilgisayar ortamında da alınabilmesi için sürü zekasının çalışma mantığının gerektirdiği kurallara ihtiyaç duyulmuştur. Etmenlerin dağılımlarını önlemek, engellerden uzaklaşmalarını sağlamak ve yönelim kuvveti uygulayabilmek için birtakım kuralların belirtilmesi gerekmiştir. Bu nedenle Python bileşeni ile belirlenen kurallar modele dahil edilmiş ve elde edilen çıktılar üzerinden model görselleştirilmiştir. Öncelikle GhPython bileşeninde yazılan kod, sürü elemanlarının birbirlerinden uzaklaşma mesafesi belirlenirken kullanılan Smaller Than bileşeni ile aynı görevi görerek; etmenlerin “noktadan listeye, listeden noktaya” dönüştürülmesi ve mesafelerin karşılaştırılmasını sağlamıştır. Bileşen içerisindeki kod ile etmenler istasyonlara uzaklıklarına göre gruplandıktan sonra, Yeni Sahra, Göztepe ve Sahrayı Cedit istasyonlarında ne gibi müdahaleler yapılacağına karar verilmiştir. Hattın Göztepe parkından ve Minibüs caddesinden geçmesinin tasarım kriterlerine daha uygun olacağı düşünüldüğü için bu bölgelere etmenlerin yönlendirilmesi gerekmektedir. Ayrıca özel mülk bölgesinden geçen güzergahın, Yeni Sahra istasyonunda o noktadan geçmemesi daha uygun olacağı için hattın oradan uzaklaştırılması gerekmektedir. Bu nedenle yönlenecek ve uzaklaşılacak bölgeler



belirlendikten sonra, bölgelere “red ve green” adlarında nokta grupları yerleştirilmiştir.

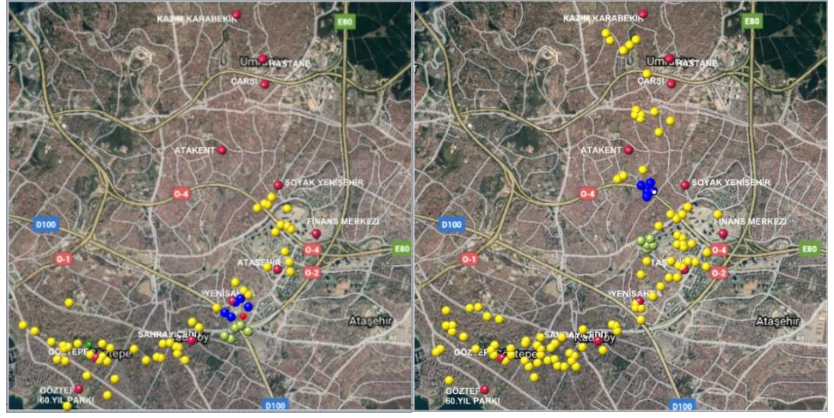
Engel noktaları için kural seti, öncelikle red noktaların yukarıda yazılan kodun düğüm noktalarından bağlanarak bileşene dahil edilmesiyle oluşturulmuştur. Bileşen içerisindeki kodun tüm etmenler ile çalıştırılarak obstacle ve smaller\_agents olmak üzere iki ayrı grup şeklinde çıktı vermesi sağlanmıştır. Engel olacak red nokta bileşeninin tüm etmenlere olan uzaklığı hesaplanarak obstacle çıktısı elde edilmiştir. Obstacle bileşeni içerisindeki noktalar, red noktalara olan mesafeleri 4.00 birimden az olduğu zaman rastgele olarak yönleri değiştirilerek simülasyona tekrar dahil edilmiştir. Simülasyon çalıştığı sürece kural seti tekrar tekrar devreye girerek her etmen için mesafeleri hesaplayıp yakın olanların yönünü her seferinde değiştirmektedir. Noktaların yönlerinin değiştirilebilmesi için öncelikle noktaların X, Y ve Z koordinatları çıkartılmıştır. Daha sonra koordinatlardan vektör oluşturulmuştur. Vektörler Rotate bileşenine bağlanarak; yön düğümüne Random bileşeni bağlanmıştır. Yönleri değişen vektörler tekrar koordinatlarına ayrılarak noktaya dönüştürülmüştür (**Şekil 19**). Yönleri değiştikten sonra etmenler, Physarealm kuralları ile çalışmaya devam ederek hedef istasyonlara ilerlemeye devam etmiştir. Engelden uzaklaşan etmenler dışında her seferinde doğan etmenler için de kurallar uygulanmıştır.



**Şekil 19:** Obstacle bileşenindeki noktaların yönlerinin değiştirilmesi.

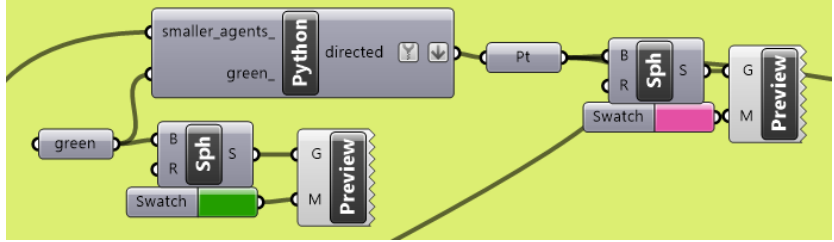
Red bileşenine bağlı engel noktalarının yeri değiştirildiğinde; her seferinde tüm etmenlerin engellere olan uzaklıkları hesaplanarak, uzaklığı 4 birimden az olan etmenlerin yönleri rastgele olarak değiştirilmiştir (**Şekil 20**). Yönü değişen etmenler yeşil renk ile temsil edilirken, değişmeden önceki etmenlerin rengi lacivert ile temsil edilmiştir. Lacivert etmenler yönlerini daha rahat tespit edilebilmesi için ayrı bir grup olarak saklanmıştır.

Şekil 20: Yönleri değişen etmenleri yayılımı.



Yönelim yapılacak noktaları içerisinde barındıran green bileşeni GhPython kodunun çıktısı olan smaller\_agents' lar ile yeni bir GhPython bileşenine bağlanmıştır (Şekil 21).

Şekil 21: Yönelim için oluşturulan düğüm dizisi.



Bileşenin içerisinde bulunan kod ve işlevi aşağıdaki şekildedir:

- Çalışılacak kütüphane çağırılır.
- Düğüm noktalarından bağlanan girdiler nokta olarak tanıtılır.
- İstasyonlara olan uzaklıklarına göre etmenler gruplandıktan sonra engelle yönelinilecek noktalara olan uzaklıklar ölçülür.

“Eğer etmenlerin uzaklıkları 8.00 birimden az ise directed listesi içerisine yazdır, değil ise geri dön ve liste dışı bırak.” Uzaklıkları 8.00 den küçük olan etmenler listelenip, green noktalar ve smaller\_agents noktaları arasında vektörler oluşturulup, bu vektörler vasıtası ile smaller\_agents noktaları green noktalara taşınır.

Green noktalara yaklaşan her etmenin uzaklığı hesaplanıp, 8.00'den az mesafede bulunanlar her seferinde green noktalara yaklaştırılmıştır. Simülasyon çalıştığı sürece, doğan her etmen için tekrar tekrar kurallar uygulanmıştır. Directed çıktısı noktasallaştırılıp görsel olarak kürelere bağlanıp simülasyon başlatıldığında, Python kodu, smaller\_agents' ların

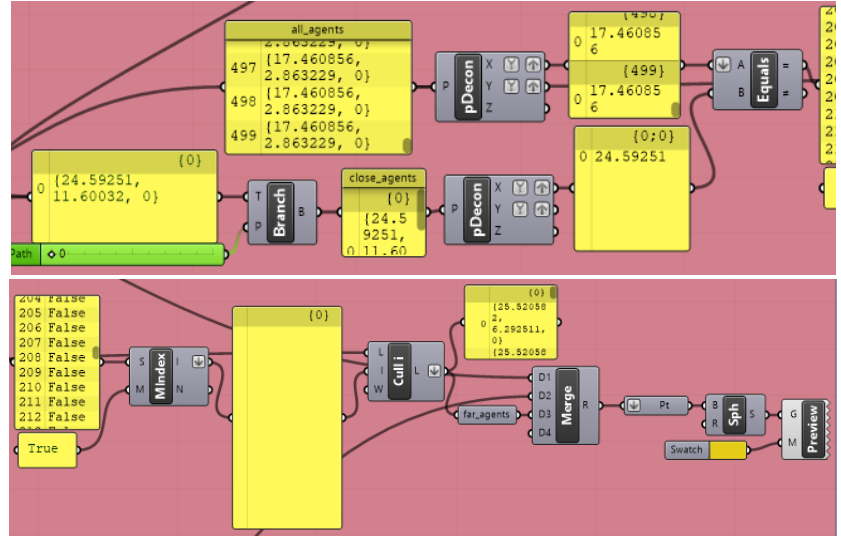
green bileşeninde tanımlı yeşil noktalara olan uzaklığı 8.00 den az olduğunda bu etmenlerin green bileşeninde tanımlı noktalara taşınmasını sağlamıştır (Şekil 22).



Şekil 22: Yönelimi yapılan etmenlerin dağılımı.

Green bileşeninde tanımlı noktalara yaklaştığında sarı renkli kürelerin rengi pembeye dönüşmektedir. Yönü değişmesi gereken etmenler, yönü değişen etmenler, yönelimi yapılacak etmenler ve yönelmiş etmenlerin tamamı son sürünün içerisinde çalıştırılmaktadır. Fakat bu son durum içerisinde, etmenler iki kere çalıştırılıp aksaklık oluşturacağı için obstacle ve directed bileşenlerinde bulunan etmenlerin, tüm etmen sürüsünden ayrıştırılması gerekmektedir. Bu nedenle tüm sürü bir "ağaç olarak düşünülüp" tekrar eden dallar bu ağaçtan çıkarılmıştır (Şekil 23).

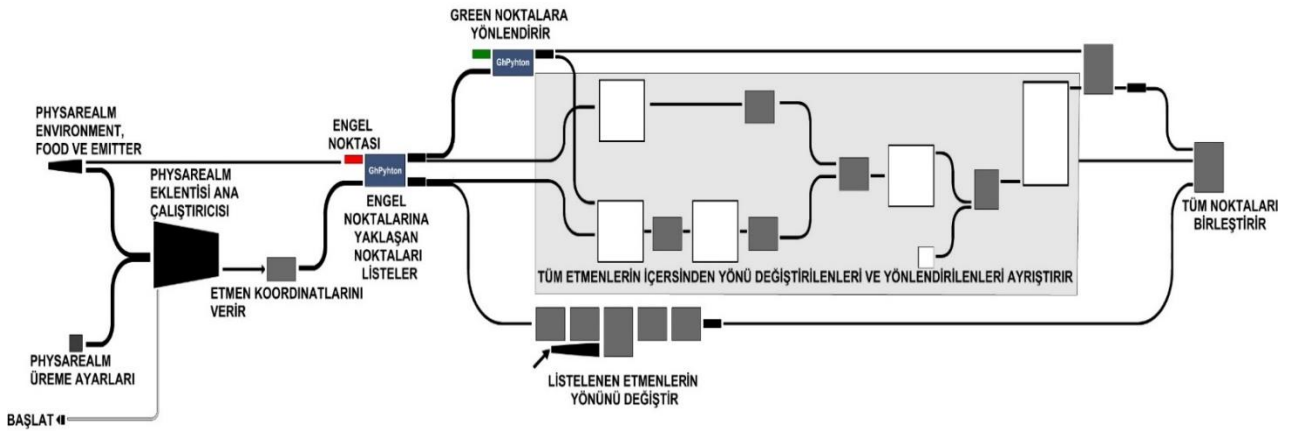
Şekil 23: Etmen Listelerinin Ayırıştırılması.



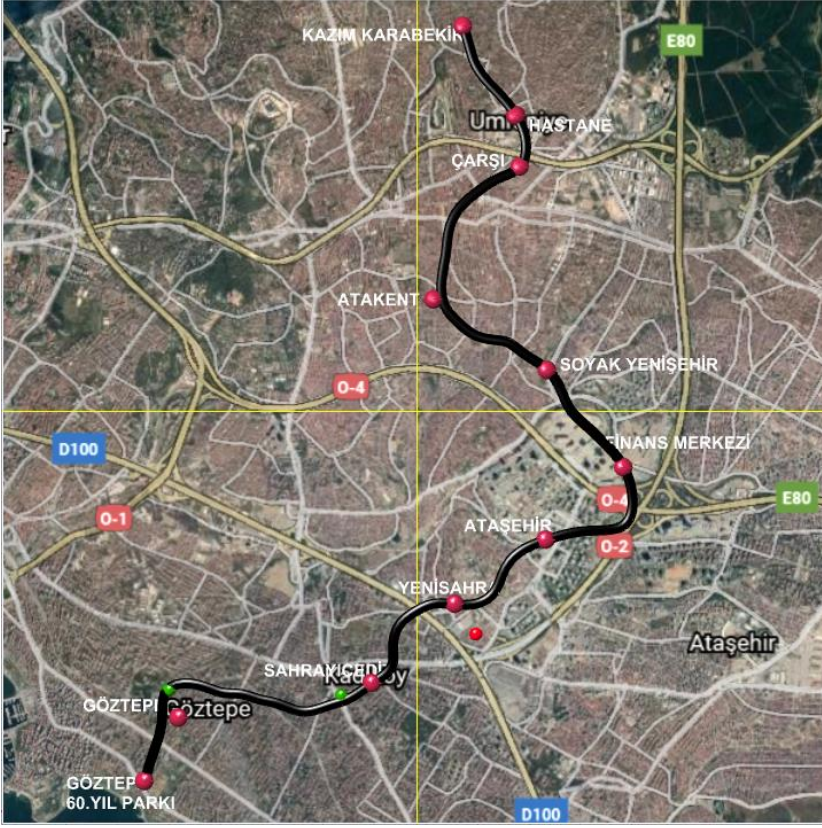
Cıvık mantar simülasyonunu oluşturan etmenlerden, yönü değiştirilmiş ve belirli bir koordinata yönlendirilmiş olan etmenlerin listeleri tüm listeden çıkarıldıktan sonra, Merge komutu ile tüm etmenler tekrar tek bir listede toplanmıştır. Liste çıktısı önce noktasal olarak alınmıştır. Son durumda sürüde çok sayıda etmen bulunduğu için noktaların eğrisel olarak ifadesi Grasshopper ortamında mümkün olmadığı için etmenler, Rhinoceros ortamına yapıştırılıp, noktalar tek bir eğriye dönüştürülmüştür.

Etmenler ile Grasshopper içerisinde oluşturulan model sadeleştirilerek bir diyagram olarak ifade edilmiştir (Şekil 24). Çok fazla bileşen kullanılması ve bileşenlerin düğüm noktalarının fazla olmasından doğan karmaşıklığın en aza indirgenmesi için Grasshopper ortamındaki ifadelerine benzer olmasına dikkat edilmiştir.

Şekil 24: Model akış diyagramı.



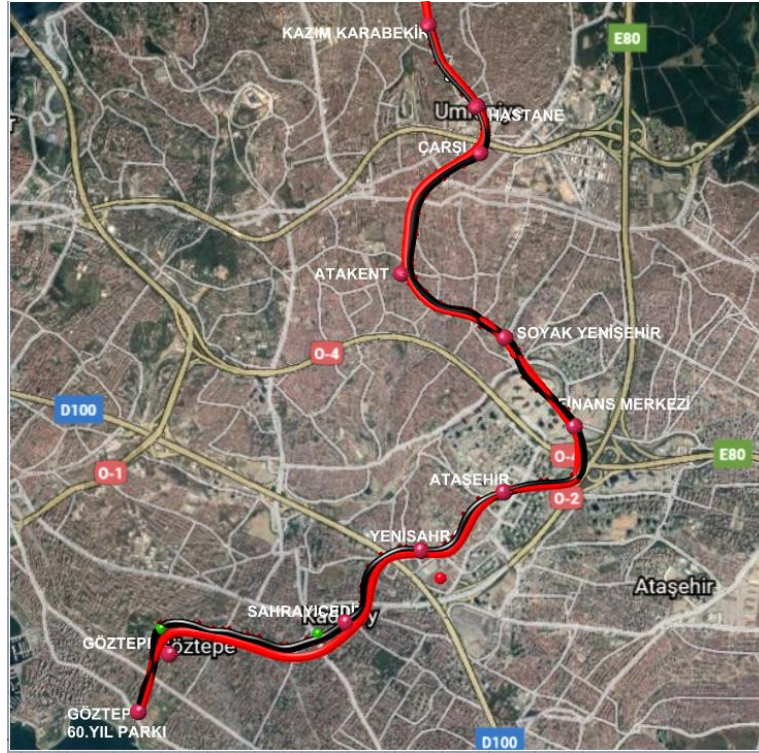
Geliştirilen model üzerinden, etmenlerin üreme parametreleri saptandıktan sonra müdahale olmaksızın metro güzergahı tasarımını gerçekleştirdikleri gözlemlenmiştir (Şekil 25).



Şekil 25: Simülasyon sonucu elde edilen metro hattı.

Canlı organizmaların belirme kavramı üzerinden şekillenen yaşam döngüleri ışığında, mevcut bir metro hattının istasyon yerlerinin bulunacağı bölgelere bağlı kalınarak metro güzergahı mevcuttan bağımsız olarak model aracılığıyla çizdirilmiştir. Belirlenen parametreler ve metro tasarım kısıtları göz önüne alınarak, ortaya çıkan güzergaha üç istasyon bölgesinde müdahale edilmesi gerektiği düşünülmüştür. Yeni Sahra istasyonundan geçtikleri noktaların uygunsuzluğundan dolayı o bölgeden uzaklaştırılırken, Sahrayı Cedit ve Göztepe istasyonlarında entegrasyon göz önüne alınarak başka bölgelere yönelimleri sağlanmıştır. Etmenlerin bu müdahaleler sonucunda oluşturdukları metro güzergahı ve mevcutta yapımı devam eden Göztepe-Ümraniye metro güzergahı arasında büyük ölçüde benzerlik olduğu gözlenmiştir (Şekil 26).

**Şekil 26:** Mevcut güzergâh ile model sonucu elde edilen güzergahın karşılaştırılması (Mevcut hat: kırmızı, Model hattı: siyah).



### 3.3 Çalışmadan Elde Edilen Çıkarımlar

Bu çalışma kapsamında yapılan literatür incelemesinde, civık mantarların metro hatlarının veya otoyolların doğruluklarını sınamak için kullanıldığı görülmüştür. Fakat bu deneyler genel olarak laboratuvar ortamında, agar kalıplar ve haritalar gibi fiziki materyaller ile gerçekleştirilmiştir. Physarealm eklentisi her ne kadar bu organizmaların yaşam döngüleri üzerine kurulu algoritmalar ile geliştirilmiş olsa da eklentinin bu tarz bir çalışmada kullanımına rastlanmamıştır. Bu anlamda yapılan deneylerin fiziki şartlardan bağımsız olarak geliştirilebilecek olması, hem tasarım aşamasında hem de mevcut hatların uygunluğunun sınanmasında daha özgür bir süreç sağlayacaktır.

Kentsel tasarımın önemli aktörlerinden olan ulaşım ağlarının şekillenmesi oldukça zahmetli süreçler gerektirmektedir. Çok sayıda matematik işlemi ve hesaplama yükü barındıran raylı sistem tasarımları, birçok mühendislik çalışması sonucu ortaya çıkmaktadır. Uzun hesaplama sürelerinin hem yeni bir metro hattının tasarımında hem de mevcut bir hatta ekleme veya hattan çıkarmalar yapılacağı durumlarda tasarımın hızlı olması gerekmektedir. Özellikle mevcut hatlar üzerinde

çalışıldığında gerek diğer ulaşım ağlarıyla entegre olduğu daha lokal bölgelerin verimliliği açısından, gerekse ulaşımın sürekliliğinin sağlanabilmesi için daha hızlı ama bir o kadar da güvenilir sonuçların elde edilmesi gerekir. Bu çalışmanın ışığında elde edilen veriler ile metro güzergahları tasarlanırken dijital ortamlarda yapılacak simülasyon modellerinin, zamansal ve parasal anlamda birçok kazanımlar sağlayacağı düşünülmektedir.

#### 4. SONUÇLAR

Meta-düzyey davranışların ortaya çıktığı sürü davranışı, kendini meydana getiren alt etmenlerin işbirlikçi davranışlarına dayanan etkileşimlerden oluşur. Etkileşen birimlerin çokluğu da en az etkileşimler kadar önemlidir. Karmaşık ve beliren sistemlerin temelini oluşturan bu toplu davranış biçimi, sürü zekâsı olarak adlandırılır. Bu davranış biçimi geri bildirim ve tekrar esaslıdır; birçok matematiksel modele de örnek olurken, aynı zamanda tasarım için yaratıcı olanaklar sunmaktadır. Basit yapıların daha karmaşık sistemler oluşturması, kentsel tasarım ve bu çalışmanın konusu olan metro güzergahlarının tasarımı üzerinden bakıldığında, tasarım aşamasında süreci planlarken avantaj sağlayabilmektedir. Cıvık mantarların bireysel olarak ulaşamadıkları besin kaynaklarına aralarında kemoatraktör iz ile iletişim kurarak yönelmeleri ve bu yönelimin çevresel faktörlerden oldukça etkileniyor olması tasarım sorunlarını tanımlarken de yol gösterici olabilmektedir.

Kullanıcı geribildirimlerinin ve birçok unsurun etkileşiminin önemli olduğu metro güzergahları, her ne kadar planlansa da insan unsurunu barındırdığı sürece makro düzeyde sonuçlar doğuran kentlerin önemli bileşenlerindedir. Cıvık mantarların ve kentlerin önemli beliren sistem örnekleri olması ve cıvık mantar davranışlarının metro güzergahlarının tasarımında kullanılması kentlerin planlanmasında verimli sonuçlar elde edilmesini sağlamaktadır.

Bu çalışmada, cıvık mantarların beliren davranışlarının altında yatan algoritmik temeller kurullarla tanımlanıp sürünün oluşturduğu güzergahlar incelenmiştir. Cıvık mantar etmenlerinin iletişimlerini sağlayan kemoatraktör iz miktarı bağlamında üretilen Physarealm eklentisi çalışma ortamı olarak seçilerek mantar davranışlarındaki değişimlerin gözlemlenmesini sağlamıştır. Model çıktıları, dijital ortamların, canlı organizmaların ve yaşam döngülerinin ışığında tasarım

aşamalarına yaptığı katkılar açıkça göstermektedir. Cıvık mantar sürülerinin fiziksel ihtiyaçları doğrultusunda açığa çıkan bu davranışların metro güzergahlarının tasarım mantığı ile bu denli örtüşmesi, beliren sistemlerin tasarım aşamasında birçok katkı sağlayacağını göstermektedir. Cıvık mantar davranışları referans alınarak üretilen model ile metro güzergahlarının tasarım aşamasının uzun süreler gerektiren hesaplama yükünün azaltılması sağlanabilmektedir.

Yapılan bu çalışma ile istasyon yerleri ve güzergâh uzunlukları gibi tasarım kriterleri belirlenen metro hatlarının, program üzerinde belirlenen parametreler ile hızlı bir şekilde oluşturulabileceği görülmüştür. Metro tasarımlarının daha hızlı gerçekleşmesi beliren sistemlerin önemli örneklerinden olan kentlerin oluşum hızlarına ulaşım sistemlerinin daha rahat ayak uydurmasını sağlayacaktır. Sayısal modellemenin sürekli gelişerek her ortama dahil olduğu günümüz dünyasında, biyolojik sistemlerin kendini organize eden yapıları ve bu yapıların altında yatan algoritmaların kullanımı her alanda olduğu gibi tasarımcıya üretkenlik konusunda çeşitli alternatifler sağlayacaktır.

## Referanslar

Adamatzky, A., Martinez, G. J., Chapa-Vergara, S. V., Asomoza-Palacio, R., Stephens, C. R., (2010). Approximating Mexican Highways with Slime Mould. *Advances in Complex Systems* 10 (3), 1195-1214.

Adamatzky, A., Akl, S. G., (2011). Trans-canada Slimeways: Slime Mould Imitates the Canadian Transport Network. doi: 10.4018/978-1-4666-4253-9.ch017. arXiv:1105.5084.

Adamatzky, A., Yang, X., Zhao, Y., (2013). Slime Mould Imitates Transport Networks in China. *International Journal of Intelligent Computing and Cybernetics* 6 (3), 232-251.

Adamatzky, A., Jones, J., (2009). Road Planning with Slime Mould: If Physarum Built Motorways It Would Route M6/M74 through Newcastle. *International Journal of Bifurcation and Chaos* 20(10), 3065-3084. doi:10.1142/S0218127410027568.

Adamatzky, A., Martinez, G., Lees, M., Slood, P. M. A. (2012). Bio-Development of Motorway Networks in The Netherlands: A Slime Mould Approach. *Advances in Complex Systems* 16 (02-03). doi:10.1142/S0219525912500348.

Adamatzky, A., Prokopenkob, M. (2011). Slime Mould Evaluation of Australian Motorways. *International Journal of Parallel Emergent and Distributed Systems* 4, 275-295.



American Association for the Advancement of Science. (2010, January 22). Slime Design Mimics Tokyo's Rail System: Efficient Methods of a Slime Mold Could Inform Human Engineers [website]. *ScienceDaily*. Retrieved from: [www.sciencedaily.com/releases/2010/01/100121141051.html](http://www.sciencedaily.com/releases/2010/01/100121141051.html)

Beni, G., Wang, J. (1990). Self-Organizing Sensor Systems. *NATO ASI Series F: Computer and Systems Sciences* 58, 251-261.

Evangelidis, V., Jones, J., Dourvas, N., Tsompanas, M. A., Sirakoulis, G. Ch., Adamatzkya, A. (2017). Physarum Machines Imitating a Roman Road Network: the 3D Approach. *Scientific Reports* 7(1). doi: 10.1038/s41598-017-06961-y.

Hensel, M., Menges, A., Weinstock, M. (2004). *Emergence: Morphogenetic Design Strategies*. Argentina: Wiley.

Hofstadter, D. (1979). *Godel, Escher, Bach: An Eternal Golden Braid*. New York: Basic Books.

Jones, J. (2010). Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks. *Artificial Life* 16, 127-153.

Leach, N. (2009). Swarm Urbanism. *Architectural Design* 79, 56-63.

Toht, A., Nakagaki, T. (2000). Intelligence: Maze- Solving by an Amoeboid Organism. *Nature* 407, (s.470-470).

Url-1<<http://archieboytom.blogspot.com>>, erişim tarihi 15.02.2020

Url-2< <https://discovermagazine.com/>>, erişim tarihi 22.03.2020

Url-3< <https://discovermagazine.com/>>, erişim tarihi 23.03.2020

Url-4< <https://www.researchgate.net/>>, erişim tarihi 17.03.2020

Url-5< <https://www.semanticscholar.org/>>, erişim tarihi 23.03.2020

Url-6< <https://www.researchgate.net/>>, erişim tarihi 17.03.2020

Url-7< <https://www.semanticscholar.org/>>, erişim tarihi 02.03.2020

Url-8< <https://www.semanticscholar.org/>>, erişim tarihi 19.03.2020

# A Decision Support System Proposal for the Optimization of Traffic on the Bosphorus Line: E-Bosphorus

Esranur Demirtaş<sup>1</sup>, Şehnaz Cenani Durmazoğlu<sup>2</sup>

0000-0002-3173-8687<sup>1</sup>, 0000-0001-8111-586X<sup>2</sup>

<sup>1</sup> Istanbul Medipol University, Graduate School of Fine Arts, Design and Architecture, Department of Architecture, Istanbul, Turkey

<sup>2</sup> Istanbul Medipol University, Graduate School of Fine Arts, Design and Architecture, Department of Architecture, Istanbul, Turkey

The Bosphorus Line, one of the important regions of Istanbul, is faced with problems such as traffic congestion and noise pollution caused by this problem in the recreation areas on this line. The main reason for these problems is the intensity of transportation by individual vehicles. However, the traffic density on this line can be reduced as a result of the fact that public transportation is provided by sea transportation, which is a characteristic mode of transportation in Istanbul inner city transportation, which is exactly parallel to this route, and is preferred for reasons such as comfort, speed and price. The inefficient use of sea transport depending on factors such as route planning and expedition times poses a problem in choosing this type of transportation. Considering the available possibilities, it is possible to optimize the road traffic on the bosphorus line by encouraging individuals using sea transportation as a result of the smart synchronization of the routes and the time schedules connected to these routes. Therefore, it is aimed to show a way to make a better regulation by revealing the shortcomings of the current (maritime) traffic regulation. It is aimed to develop the conceptual framework of a decision support system that regulates maritime traffic and the benefits of the proposed system are stated. Decision tree and shortest path finding algorithms are used as methods in the study. In addition to the shortest path problem, finding the shortest time is one of the criteria discussed in this study. As a result, the shortest route and the routes to be reached in the shortest time were determined when going from a pier on the bosphorus line to another one. In addition, the interface design of an application (E-Bosphorus) has been developed to make the current bosphorus line ferry timetables more efficient.

**Keywords:** Intelligent Seaborne Transportation, Bosphorus Line, Decision Support System, Route Finding.

**Received:** 23.08.2020

**Accepted:** 27.09.2020

**Corresponding Author:**

esranurdemirtastr@gmail.com

Demirtaş, E & Cenani, Ş. (2020). A Decision Support System Proposal for the Optimization of Traffic on the Bosphorus Line: E-Bosphorus, JCoDe: Journal of Computational Design, 1(3), 49-70.

# Boğaz Hattındaki Trafiğin Optimizasyonuna Yönelik Bir Karar Destek Sistemi Önerisi: E-Bosphorus

Esranur Demirtaş<sup>1</sup>, Şehnaz Cenani Durmazoğlu<sup>2</sup>

0000-0002-3173-8687<sup>1</sup>, 0000-0001-8111-586X<sup>2</sup>

<sup>1</sup> İstanbul Medipol Üniversitesi, Güzel Sanatlar Tasarım ve Mimarlık Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

<sup>2</sup> İstanbul Medipol Üniversitesi, Güzel Sanatlar Tasarım ve Mimarlık Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

İstanbul'un önemli bölgelerinden biri olan Boğaz Hattı, trafik sıkışıklığı problemi ve bu problemin bu hattaki rekreasyon alanlarında meydana getirdiği gürültü kirliliği gibi problemlerle karşı karşıyadır. Bu problemlerin ana sebebi bireysel araçlarla yapılan ulaşımın yoğunluğudur. Ancak toplu taşımanın tam olarak bu güzergahın paralelinde bulunan ve İstanbul şehir içi ulaşımında karakteristik bir ulaşım şekli olan deniz yolu ulaşımı ile sağlanması, konfor, hız, ücret gibi nedenlerle tercih edilmesi sonucunda bu hattaki trafik yoğunluğu azaltılabilir. Deniz yolu ulaşımının rota planlaması ve sefer saatleri gibi faktörlere bağlı olarak verimli olmayan kullanımı bu ulaşım türünün tercih edilmesi noktasında sorun teşkil etmektedir. Mevcut imkanlar göz önüne alındığında rotaların ve bu rotalara bağlı zaman çizelgelerinin akıllı senkronizasyonu sonucunda, boğaz hattını kullanan bireylerin deniz yolu ulaşımını kullanmalarını teşvik etmek yoluyla boğaz hattındaki kara yolu trafiğinin optimizasyonun sağlanması mümkündür. Bu nedenle, çalışmada mevcut (deniz) trafik düzenlemesinin eksikliklerini ortaya koyarak daha iyi bir düzenlemenin nasıl yapılacağına dair bir yol gösterilmesi hedeflenmiştir. Deniz trafiğini düzenleyici bir karar destek sisteminin kavramsal çerçevesinin geliştirilmesi amaçlanmış ve önerilen sistemin faydaları belirtilmiştir. Çalışmada yöntem olarak karar ağacı ve en kısa yolu bulma algoritmaları kullanılmıştır. En kısa yol problemine ek olarak en kısa zamanı bulma da bu çalışmada ele alınan kriterlerdendir. Sonuç olarak, boğaz hattındaki bir iskeleden yine boğaz hattındaki bir başka iskeleye giderken en kısa yol ve en kısa zamanda gidilecek rotalar belirlenmiştir. Ayrıca, mevcut boğaz hattı vapur sefer tarifelerinin daha verimli olmasına yönelik bir uygulamanın (E-Bosphorus) arayüz tasarımı geliştirilmiştir.

**Anahtar Kelimeler:** Akıllı Deniz Yolu Ulaşımı, Boğaz Hattı, Karar Destek Sistemi, Rota Bulma.

**Teslim Tarihi:** 23.08.2020

**Kabul Tarihi:** 27.09.2020

**Sorumlu Yazar:**

esranurdemirtastr@gmail.com

Demirtaş, E., Cenani, Ş. (2020). Boğaz Hattındaki Trafiğin Optimizasyonuna Yönelik Bir Karar Destek Sistemi Önerisi: E-Bosphorus. JCoDe: Journal of Computational Design, 1(3), 49-70.

## 1. GİRİŞ

Günümüzde bireysel araçların kullanımı ile artan trafik yoğunluğu, teknolojinin de sunduğu olanaklarla optimize edilebilir bir hale gelmiştir. Türkiye özelinde trafiğin en problemlisi olduğu şehirlerden biri İstanbul'dur. İstanbul'a daha yakından bakmak gerekirse, özellikle rekreasyonel alanların yoğunlukta olduğu, insanların boş vakit geçirme, eğlenme, spor yapma gibi amaçlarla sıklıkla kullandığı Boğaz Hattı'nın kıyıya paralel aksında uygun bir toplu taşıma olanağı sağlanabilecek bir sistemin geliştirilmesi gerektiği görülür. Bu aksta geliştirilecek olan bir toplu taşıma ağıyla kullanıcıların bireysel araçlardan ziyade konfor, dinlenme, kalabalık olmayan ulaşım, uygun ücretlendirme gibi nedenlerle deniz ulaşımını tercih etmeleri sağlanabilir. Ayrıca kentlinin İstanbul'u deniz yolu ulaşımı ile deneyimlemesi sağlanacaktır.

İstanbul dünyada önemli deniz geçişlerinin biri üzerinde kurulu bir şehirdir. Çancı ve diğ. göre (2015) diğer deniz geçişlerinden farkı sadece deniz yolu üzerindeki transit geçişlerle sınırlı olmamasıdır. Aynı zamanda kara yolunun köprülerle, demiryollarının da boğaz tüp geçidi şeklinde bu su yolu üzerinden geçmesi diğer bir özelliği olmaktadır. Şehir ölçeğinde; İstanbul şehir içi yolcu taşımacılığında deniz yolu taşımacılığının % 3 (ortalama 250.000 yolcu/gün)<sup>1</sup> kadarlık bir payı bulunmaktadır. Bu da taşınan günlük yolcu sayısı bakımından İstanbul deniz yolu ulaşımının ayrıcalıklı bir konumda olduğunu göstermektedir. Hızla artan nüfusu ile şehir içi yolcu taşımacılığı yönünden İstanbul'da iki kıtayı birleştiren ulaşım türü olarak deniz taşımacılığının verimli hale getirilme ihtiyacı daha da önem kazanacaktır (Çancı, Önden, Çakmak, Gürel ve Tuzla, 2015). Deniz yolu taşımacılığını diğer sistemlerden ayıran ucuz ve yüksek kapasiteli olması, emniyetli olması gibi yönleridir. Ancak ülkemizde gereken rağbeti görmemekte ve deniz yolu ulaşımı teşvik edilmemektedir. Buna ek olarak, diğer toplu taşıma ağlarıyla modlar arası entegrasyon ve diğer modlardan deniz yoluna geçişin teşvik edilmesi sağlanmamaktadır. Bu da var olan sistemin verimsiz olarak kullanılması şeklinde bir sonuçla karşılaşmamıza neden olur.

Bu noktada Akıllı Ulaşım Sistemlerinin (AUS) kullanımı ile verimli, sürdürülebilir bir yapının oluşması sağlanabilir. Özellikle veriye dayalı karar destek sistemlerinin kullanımı ile ulaşım ağı optimize

---

<sup>1</sup> 2019 yılında <https://data.ibb.gov.tr/> verilerine göre 808.278 yolcu/gün sayısına ulaşmıştır. 2019 yılında bu oran %4.8'e çıkmıştır.

edilebilecektir. Bu sistemler kara yolu, deniz yolu, demir yolu gibi taşımacılık sistemlerinde kullanılabileceği gibi, elektrikli kaykay (e-scooter), elektrikli bisiklet vb. bireysel taşımacılık ağlarıyla da entegre edilerek optimize edilebilir. Bunun için geçmişe yönelik veriler, belirli bir zaman aralığındaki durumlar, davranış modelleri analiz edilebilir. Günümüz teknolojik alt yapısı bu verileri analiz etmeye, verilerden çıkarılan sonuçlarla bu tür karar destek sistemlerini oluşturmaya uygundur. Özellikle sefer saatleri, güzergahların belirlenmesi gibi dinamik, kullanıcı verilerine dayalı durumlar için kullanılması sürdürülebilir bir kent yaşamı için bir gereklilik haline gelmiştir.

Mobiett, Moovitapp, Trafi, Google (*Url 1-2-3-4*) gibi çeşitli uygulamalarda vapur saatleri ve güzergahları Şehir Hatları, Turyol, Dentur gibi firmalar tarafından aktarmalı olarak yapılan vapur seferleri aktarmasız bir şekilde, doğrudan iskeleden iskeleye yapılan seferler olarak kabul edilerek rota önerisinde bulunmaktadır. Aksi takdirde mevcut sistemden çıkarılacak sonuç yetersiz kalacaktır. Örneğin Rumeli Kavağı-Eminönü hattı ile Beşiktaş'tan Arnavutköy'e gitmek mümkünken rota önerilerinde bu hat yalnızca Rumeli Kavağı ve Eminönü arasında hareket eden bir hat olarak gözükmemektedir. Bu nedenle vapur kullanıcısı da hangi noktadan hangi noktalara varabileceğinin bilincinde olamamaktadır. Buna bağlı olarak ayrıca en kısa yolu kullanarak, en kısa zamanda varış bilgisine de kullanıcı erişememektedir; çünkü verilen bilgi hiçbir iskelede durulmadan varılan, tek yönlü bir rotayı içermektedir. Bu sorunun sefer tarifelerinin verimli bir şekilde optimizasyonunun sağlandığı bir sistemle düzenlenebileceği düşünülmektedir. Çalışmanın amacı İstanbul Boğazında su taşımacılığıyla en kısa yol ve en kısa zamanda bir iskeleden başka bir iskeleye gidebilmek için güzergah belirlenmesidir. Bu çalışmada mevcut (deniz) trafik düzenlemesinin eksikliklerini ortaya koyarak daha iyi bir düzenlemenin nasıl yapılacağına dair bir yol gösterilmesi hedeflenmiştir. Deniz trafiğini düzenleyici bir karar destek sisteminin kavramsal çerçevesinin geliştirilmesi amaçlanmış ve önerilen sistemin faydaları belirtilmiştir. Ayrıca, çalışma optimizasyonun sağlandığı, önerilecek bir mobil uygulamanın varlığıyla çözülmesi olası problemlere vurgu yapar.

Çalışmanın ikinci bölümünde literatür araştırması üst bir kavram olan Karar Destek Sistemlerinde Veri Madenciliği, Karar Ağacı ve En Kısa Yol Bulma Problemi olmak üzere üç başlık altında incelenmiştir. Üçüncü bölümde çalışmada kullanılan yöntem açıklanmıştır. Yöntem

bölümünde öncelikle çalışma alanının bilgisi detaylı olarak verilmiştir. Daha sonra çalışma kapsamında gerçekleştirilen veri toplama süreci açıklanmıştır. Yöntem çalışmasının son kısmında deniz yolu ulaşımında karar destek sistemi baz alınarak arayüzü geliştirilen bir mobil uygulama önerisi trafiği düzenleyen bir araç olarak sunulmuştur. Çalışmanın son bölümünde ise sonuçlar ortaya koyulmuş ve tartışılmıştır.

## **2. KARAR DESTEK SİSTEMLERİNDE VERİ MADENCİLİĞİ**

Karar destek sistemleri bir karar aşamasında, verilerden faydalanarak karar vermeyi kolaylaştıran sistemlerdir. Verimliliğin artırılması, karar verme aşamasının hızlandırılması, karar verenin veriye erişiminin ve kavramasının hızlandırılması gibi önemli avantajları bulunur. Ham veri tek başına anlamlı bir sonuca ulaşılmasına olanak tanımayacağı için verilerin kullanılabilir, anlamlı, birbirleriyle ilişkilendirilebilir yararlı bilgiler haline getirilmesi gerekmektedir. Veri madenciliği, büyük miktardaki verilerden kuralların, örüntülerin ve modellerin ortaya çıkarılmasıdır (Emel ve Taşkın, 2005). Veri madenciliğinde bilgi keşfi Han, Kamber ve Pei'ye göre (2012) verinin temizlenmesi (data cleaning), veri birleştirme (data integration), veri seçme (data selection), veri dönüştürme (data transformation), veri madenciliği uygulaması (data mining), örüntü değerlendirme (pattern evaluation), bilgi sunumu (data presentation) süreçlerinden geçer.

Veri madenciliğinde tahmin edici modeller ile örüntü tanıma işi, sınıflama, regresyon ve zaman serileri yaklaşımlarını içerir. Eldeki verinin gruplarını bulan kümeleme, ilişkilendirme ve ardışıklık kurallarını elde etmeyi kapsayan ilişkilendirme analizi ve ardışıklık keşfi davranışı ise tanımlama amaçlı kullanılır (Emel ve Taşkın, 2005).

Veri madenciliği modellerini işlevlerine göre üç ana grup altında toplamak mümkündür: 1. Sınıflama (Classification) ve Regresyon (Regression) 2. Kümeleme (Clustering) 3. İlişkilendirme Kuralları (Association Rules) ve Ardışık Zamanlı Örüntüler (Sequential Patterns)dir. Sınıflama ve regresyon modelleri, kestirimci, kümeleme; ilişkilendirme kuralları ve ardışık zamanlı örüntü modelleri ise tanımlayıcı modellerdir (Albayrak ve Yılmaz, 2009).

Sınıflama ve regresyon, önemli veri sınıflarını ortaya koyan veya gelecek veri eğilimlerini tahmin eden modelleri kurabilen analiz yöntemleridir. Sınıflama ve regresyon modellerinde kullanılan başlıca teknikler: Yapay Sinir Ağları (Artificial Neural Networks), Genetik Algoritmalar (Genetic

Algorithms), K-En Yakın Komşu (K-Nearest Neighbour-KNN), Naive-Bayes sınıflayıcısı, Lojistik Regresyon ve Karar Ağaçları (Decision Trees)'dir (Çalış, Kayapınar ve Çetinyokuş, 2014). Bu çalışmada Karar Ağaçları verinin sınıflandırılmasında, En kısa yol bulma probleminde ise Dijkstra Algoritması kullanılmıştır.

## 2.1 Karar Ağacı

Karar ağaçları bilgi keşfi sırasında pek çok test gerçekleştirerek, hedefi tahmin etmede en iyi sırayı bulmaya çalışırlar. Her bir test karar ağacındaki dalları oluşturur ve bu dallar da diğer testlerin gerçekleşmesine neden olur (Emel ve Taşkın, 2005).

Ağaç üzerindeki koşul ifadeleri 'düğüm' olarak adlandırılırken ilk düğüm 'kök düğüm' olarak ifade edilir. Düğümler arası bağlantı 'kenar' olarak adlandırılır. 'Yaprak' olarak isimlendirilen koşul belirtmeyen ifadeler ise sınıf etiketlerini gösterir. Bir karar ağacı, basit karar verme adımları uygulanarak, büyük miktarlardaki verileri, çok küçük veri gruplarına bölerek verinin analizini sağlayan bir yapıdır. "If-Else" veya "Switch-Case" basamak yapıları ile algoritmaları oluşturulabilir.

## 2.2 En Kısa Yol Bulma Problemi

En kısa yolu bulma (shortest path) konusunda çeşitli algoritmalar kullanılmaktadır. Bunlardan en yaygınları Bellman, Dijkstra, Dreyfus, Floyd Warschall tarafından geliştirilen algoritmalarlardır. Ayrıca genetik algoritmalar (Dener ve Ark., 2011) ve A star algoritması da en kısa yolu/rotayı bulma çalışmalarında sıkça kullanılmıştır. Yine bu bağlamda gezgin satıcı problemi de yaygın olarak kullanılmaktadır.

Genetik algoritmalar, doğada gözlemlenen evrimsel sürece benzer bir şekilde çalışan arama ve eniyileme yöntemidir. Karmaşık çok boyutlu arama uzayında en iyinin hayatta kalması ilkesine göre en iyi çözümü aramaktadır (Beasley, Bull ve Martin, 1993). Genetik algoritmalar problemlerin çözümü için evrimsel süreci taklit ederler. Diğer eniyileme yöntemlerinde olduğu gibi çözüm için tek bir yapının geliştirilmesi yerine, böyle yapılardan meydana gelen bir küme oluştururlar. Kümedeki bireyler evrimsel süreç içinde genetik algoritma işlemleri tarafından belirlenirler (Bingul, Sekmen ve Zein, 1999).

Dijkstra'nın algoritması, en kısa yol problemini çözmek için genişlik öncelikli aramadan (Breadth First Search - BFS) yararlanır. Sadece pozitif yol değerlerine sahip bir graf için, verilen bir başlangıç düğümünden (source node), diğer bütün düğümlere olan en kısa yolları bulmak için

kullanılmaktadır. Yol bulma işlemi sırasında hangi düğümle yola devam edileceğine karar vermek için açgözlü yaklaşımı (greedy approach) kullanılmaktadır. Herhangi bir yol negatif değere sahipse algoritma başarılı bir şekilde çalışmaz (Dijkstra, 1959).

En iyi öneriye ulaşma amacı ile genetik algoritmalar, düğümün bir adım sonraki düğümle ilişkisi bağlamında ise Dijkstra algoritması çalışmanın yöntem aşamasında seyahatin başlangıç ve bitişi arasında en kısa yolun hesaplanmasında kullanılmıştır.

En kısa yol bulma konusunda yeterince kaynağa ulaşılabilmemiş, ancak deniz yolu ulaşımında en kısa yolu ve en kısa süreyi bulma konusunda İstanbul denizyolu toplu ulaşımında sınırlı kaynağa rastlanmıştır.

Ceder A. ve Sarvi M. (2007) tarafından yapılan çalışmada var olan deniz yolu ulaşım rotaları, aday iskeleler/rotalar, ortalama seyahat süresi, uzaklık, deniz otobüslerinin en yoğun kullanılan ve en az kullanılan zamanlardaki yolcu kapasiteleri, deniz otobüslerinin hızları, sefer saatleri, deniz otobüslerinin türleri, filoların büyüklüğü, deniz ulaşımına yönelik yönetmelikler gibi veriler toplanmıştır. Çalışmada hedef kitle yolcu, işletmeci ve belediyedir. Tüm tarafların yararına olabilecek bir sistem önerisinin geliştirilmesi hedeflenmiştir.

### 3. YÖNTEM

#### 3.1 Çalışma Alanı

Çalışmada İstanbul'un Avrupa yakasından 12 ve Anadolu yakasından 13 adet olmak üzere toplamda 25 adet iskeledeki sefer saatleri ve bu iskelelere ulaşımın sağlanabildiği diğer iskelelerin ilişkiselliği öncelikli olarak incelenmiştir. Avrupa yakasındaki iskeleler Eminönü, Karaköy, Kabataş, Beşiktaş, Ortaköy, Arnavutköy, Bebek, Emirgan, İstinye, Yeniköy, Sarıyer, Rumeli Kavağı iskeleleridir. Anadolu yakasındaki iskelelerse Kadıköy, Üsküdar, Kuzguncuk, Beylerbeyi, Çengelköy, Kandilli, Küçüksu, Anadolu Hisarı, Kanlıca, Çubuklu, Paşabahçe, Beykoz ve Anadolu Kavağı iskeleleridir (**Şekil 1**). Trafiğin en yoğun olduğu hafta içi 07:00-09:00 ve 17:00-19:00 saatleri arası çalışmada odaklanılan zaman aralıkları olmuştur.



Şekil 1: Çalışma kapsamında ele alınan iskeleler.

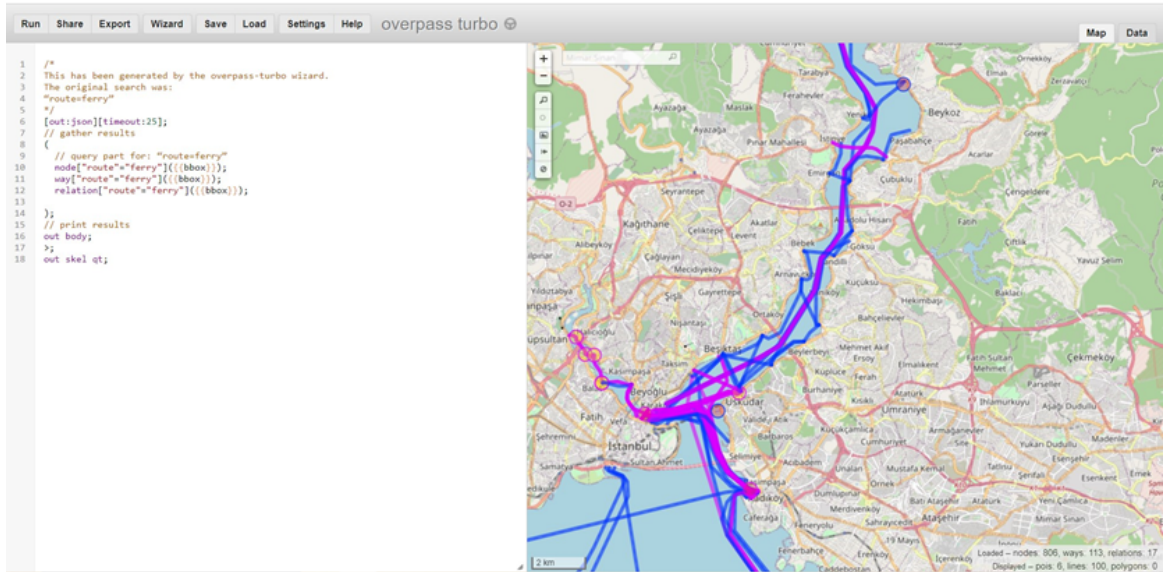


### 3.2 Veri Madenciliği Bilgi Keşif Süreci

Çalışmanın veri toplama ve toplanan verinin işlenmesi adımlarında; en iyi öneriye ulaşma amacı ile genetik algoritmalar, düğümün bir adım sonraki düğümle ilişkisi bağlamında ise Dijkstra algoritması seyahat başlangıç ve bitiş noktaları arasındaki en kısa yolun hesaplanmasında kullanılmıştır. Çalışmada, İstanbul Boğazı'nın tek kıyısında, aktarmasız

olarak yapılan seferler ile Avrupa ve Anadolu yakaları arasındaki seferler bir arada incelenecektir. Bu aşamada açık ve gönüllü veri üretimi ile oluşturulan OpenStreetMap (Url-5) sitesinin veri madenciliği ve filtreleme aracı olarak kullanılabilen Massachusetts Institute of Technology lisanslı overpass-turbo.eu sitesine aşağıdaki kod yazılarak iskelelerin ilişkilerine yönelik bir harita çıkarılmıştır (**Şekil 2**). Boğaz hattında yapılan tüm seferler için iskelelerden tek gidiş veya tek dönüş seçenekleri ele alınmıştır. Örneğin Şehir Hatları'nın Çengelköy-İstinye vapuru aktarmalı olarak Çengelköy-Arnautköy-Bebek-Kandilli-Anadolu Hisarı-Kanlıca-Emirgan-İstinye iskelelerinden geçmektedir (**Şekil 3**). Bu rota Çengelköy-Arnautköy, Arnautköy-Bebek, Bebek-Kandilli vb. şekilde bölünmüştür. Böylelikle aktarmaların da dahil olabileceği ve daha fazla alternatifin mümkün kılındığı bir ulaşım ağı oluşturulmuştur.

**Şekil 2:** Overpass-turbo.eu sitesinden çekilen veri (Url-6).



Yazılan kod aşağıdaki gibidir:

```

[out:json][timeout:25];
(
// query part for: "route=ferry"
node["route"="ferry"]({{bbox}});
way["route"="ferry"]({{bbox}});
relation["route"="ferry"]({{bbox}});
);
out body;
>;
out skel qt;

```

ÇENGELKÖY - İSTİNYE HATTI / Line							
HAFTA İÇİ ve CUMARTESİ GÜNLERİ / Working Days and Saturdays							
Ç. KÖY	A. KÖY	BEBEK	KANDİLLİ	A. HİSARI	KANLICA	EMİRGAN	İSTİNYE
KALKIŞ / Departure							VARIŞ Arrival
07:25	07:35	07:40	07:45	07:55	08:05	08:10	08:20
09:25	09:35	09:40	09:45	09:55	10:05	10:10	10:20
12:25	12:35	12:40	12:45	12:55	13:05	13:10	13:20
15:25	15:35	15:40	15:45	15:55	16:05	16:10	-
-	16:50	16:55	17:00	17:10	17:20	17:25	17:35
18:40	18:50	18:55	19:00	19:10	19:20	19:25	-
İSTİNYE	EMİRGAN	KANLICA	A. HİSARI	KANDİLLİ	BEBEK	A. KÖY	Ç. KÖY
KALKIŞ / Departure							VARIŞ Arrival
08:20	(08:10)	08:30	08:40	08:45	08:50	08:55	09:05
10:20	(10:10)	10:30	10:40	10:45	10:50	10:55	11:05
13:20	(13:10)	13:30	13:40	13:45	13:50	13:55	14:05
-	16:15	16:20	16:30	16:35	16:40	16:45	-
17:40	17:50	17:55	18:05	18:15	18:20	18:25	18:35
-	19:30*	19:35*	19:45*	-	19:55*	20:00*	20:10*

Şekil 3: Şehir Hatları Çengelköy-İstinye hattı tarifesini (Url-7).

Overpass-turbo.eu sitesindeki verilerin güncelliği ve doğruluğu, açık ve gönüllü veri üretimi ile oluşturulduğu için sorgulanmış ve ek olarak sehirhatlari.istanbul, ido.com.tr, turyol.com, denturavrasya.com, trafi.com, moovitapp.com sitelerinden de bu veri teyit edilerek tablo bilgisi olarak üretilmiş ve ayrıca Beşiktaş örneği özelinde grafik olarak bir araya getirilmiştir (Şekil 4). Ancak overpass-turbo.eu sitesi genel bir bakış sunması ve veriyi görselleştirerek aktarması açısından çalışma kapsamında olumlu bir katkı sağlamıştır.

Şekil 4: Beşiktaş'a bir durak bağlantılı hatların 07:00-07:55 saatleri arasındaki kalkış durumları.

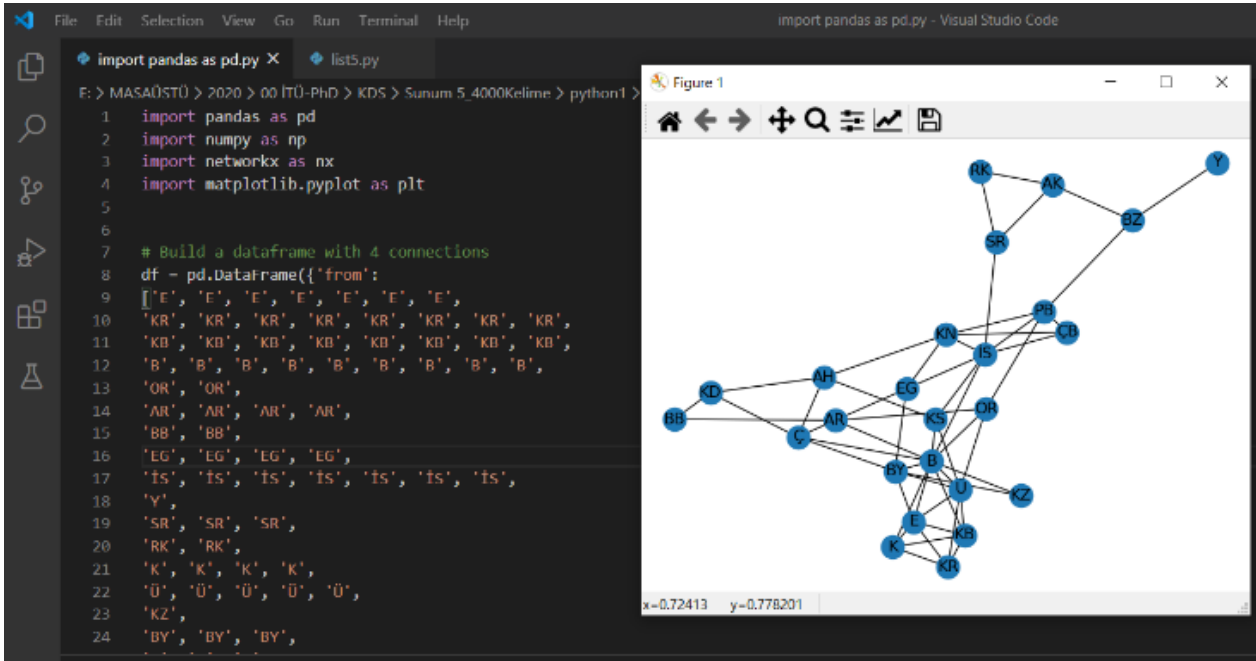
KOD	HAT BAŞLANGICI	HAT BİTİŞİ	İŞLETME	07:00	07:05	07:10	07:15	07:20	07:25	07:30	07:35	07:40	07:45	07:50	07:55
B-01	BEŞİKTAŞ	KABATAŞ	Şehir Hatları											VAR	
B-02	BEŞİKTAŞ	KABATAŞ (-KÜÇÜKSU)	Şehir Hatları												VAR
B-03	BEŞİKTAŞ	KABATAŞ (-B. Hattı)	Şehir Hatları												
B-04	BEŞİKTAŞ	EMİNÖNÜ (-B. Hattı)	Şehir Hatları								VAR		VAR	VAR	VAR
B-05	BEŞİKTAŞ	EMİNÖNÜ (-R. Kavağı)	Şehir Hatları												
B-06	BEŞİKTAŞ	KADIKÖY	Şehir Hatları				VAR						VAR		
B-07	BEŞİKTAŞ	BEYLERBEYİ (-B. Hattı)	Şehir Hatları												
B-08	BEŞİKTAŞ	KADIKÖY	Turyol	VAR						VAR					
B-09	BEŞİKTAŞ	ORTAKÖY (-B. Hattı)	Şehir Hatları												
B-10	BEŞİKTAŞ	ÜSKÜDAR	Dentur					VAR							
B-11	BEŞİKTAŞ	KUZGUNCUK (-B. Hattı)	Şehir Hatları												
B-12	BEŞİKTAŞ	İSTİNYE (-Sarıyer)	Şehir Hatları												
B-13	BEŞİKTAŞ	KABATAŞ (-Adalar)	Dentur												
B-14	BEŞİKTAŞ	KADIKÖY	Dentur												
B-15	BEŞİKTAŞ	ORTAKÖY (-Bebek)	Dentur												
B-16	BEŞİKTAŞ	ATNAVUTKÖY (-R. Kavağı)	Şehir Hatları												

İlişkilerin daha net kurulabilmesi adına öncelikle 07:00-09:00 ve 17:00-19:00 saatleri veri kaybı olmaması için beşer dakikalık aralıklara bölünmüştür ve kalkış saatleri işlenmiştir. Böylece Avrupa ve Anadolu yakasında, yukarıda belirtilen tüm iskelelerden ayrı ayrı kalkış saatlerinin işlenmesi mümkün olmuştur. 07:00-09:00 ve 17:00-19:00 saatleri arasındaki sefer saat aralıkları ve güzergahları değişkenlik gösterdiği için ve çalışmadaki zaman kısıtlılığı nedeniyle 17:00-19:00 saatleri arasındaki veriler bu aşamada popülasyon büyüklüğünü daraltmak adına elenmiştir. Seferlerin mevcut olduğu saatler 1’le, mevcut olmadığı saatler ise 0 ile gösterilmiştir (**Şekil 5**).

HAT BAŞLANGICI	HAT BİTİŞİ	SÜRE	07:10:00	07:30:00	07:35:00
EMİNÖNÜ	KADIKÖY	25	0	0	1
EMİNÖNÜ	KADIKÖY	25	0	0	0
EMİNÖNÜ	KADIKÖY	25	0	0	0
EMİNÖNÜ	ÜSKÜDAR	20	1	1	0
EMİNÖNÜ	ÜSKÜDAR	20	0	1	0
EMİNÖNÜ	KABATAŞ	15	0	0	0
EMİNÖNÜ	BEYLERBEYİ	30	0	0	0

**Şekil 5:** Saatlere göre seferler

Ayrıca hat başlangıçları doğrudan hattın son noktası ile ilişkilendirilmemiş, düğüm noktaları olarak işaretlenerek hattaki bir sonraki durak ile ilişkilendirilmiştir. Böylece üzerinden geçilen hatlar da optimizasyon işlemine ve diğer iskelelerle ilişkileri ile sisteme dahil edilebilecektir (**Şekil 6**). İskele adları bir veya isimlerin çakışması durumunda (Kadıköy:K, Kabataş: KB, Kandilli:KN gibi) iki harf ile gösterilmiştir. Bahsi geçen Dijkstra algoritmasına göre düğümlerin bir sonraki düğümlerle ilişkisi zaman faktörü baz alınmadan ele alındığında, bu çizgeye göre örneğin Beşiktaş iskelesinden Yeniköy iskelesine varış yapılabileceği sonucuna varılır. Ancak zaman faktörü dahil edildiğinde bu durumlar değişkenlik gösterebilmektedir. Bu çizgenin amacı düğümün (iskelenin) bir sonraki düğümle ilişkisi bağlamında başlangıç noktasından varış noktasına kullanılabilecek alternatiflerin görülebilmesini sağlamaktır.



Rapidminer yazılımı üzerinden yapılan veri analizinde hem kalkış hem de varış açısından en yoğun olarak kullanılan iskelelerin Beşiktaş olduğu tespit edilmiş ve çalışma Beşiktaş üzerinden örneklendirilmiştir.

Şekil 6: Python ile oluşturulan iskeleler arası ilişki ağı.

### 3.3 Deniz Yolu Ulaşımında Karar Destek Sistemi Mobil Uygulama Önerisi

İstanbul gibi bir şehirde kentin deniz yolu ulaşımı ile deneyimlenmesi önemlidir. Ancak bu konuda bir bilincin oluşabilmesi kentlinin deniz yolu ulaşımına dair verilere kolaylıkla erişmesi ile mümkün olacaktır. Buna yönelik olarak bu çalışma kapsamında karar destek sistemi olarak oluşturulan mobil uygulama ile kullanıcı hem başlangıç ve varış noktasını kendi belirleyebilecek hem de başlangıç noktasını belirleyip, varış noktasına dair mobil uygulamanın önerileri üzerinden rota alternatiflerine erişebilecektir. Rota alternatiflerinin yanı sıra bu rotalara yönelik saat aralıklarına ulaşabilecektir. Çalışmanın ana çıktısı olarak İstanbul Boğazı şehir içi yolcu ve araç taşımacılığı hatlarındaki bir iskeleden yine bu hattaki bir başka iskeleye giderken en kısa yol ve en kısa zamanda gidilecek rota/rotalar belirlenmiştir. Buna dair oluşturulan akış şeması Şekil 7'deki gibidir. Akış şemasına göre uygulama başlatıldıktan sonra başlangıç noktası ve saati, daha sonra varış noktası ve saati girilir veya varış noktası alternatifleri E-Bosphorus Karar Destek Sistemi'nden alınır. Öncelikli olarak aktarmasız seçenekler sunulur. Bu

seeneklerin olmaması durumunda aktarmalar birer artırılarak kullanıcıya öneriler sunulur. Sistemde öneri sunulmaması durumunda saat ve iskele ihtiyacı bilgisinin İstanbul Büyükşehir Belediyesi Akıllı Şehir Müdürlüğü ile paylaşılması ve bu doğrultuda bir sistemsel iyileştirme önerisinin iletilmesi planlanmıştır.

Kullanıcının bilgiye hızlıca erişimi, kullanım kolaylığı, hedef kitleye erişebilmesi gibi nedenlerle önerilen karar destek sistemi bir mobil uygulama aracılığıyla oluşturulmuştur.

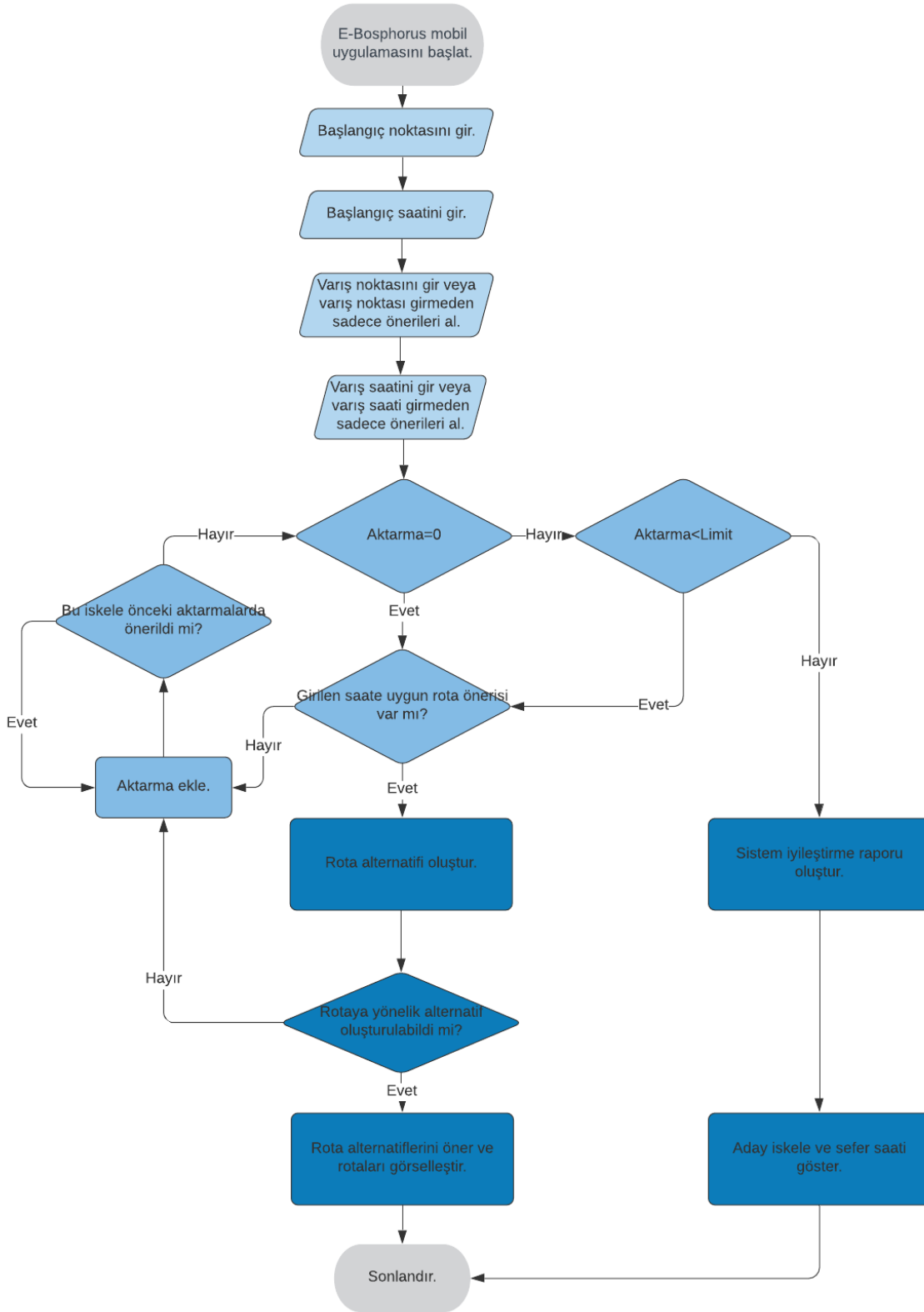
Daha detaylı bir çalışma için daha önce belirtildiği gibi Beşiktaş örneğine yoğunlaşmıştır. Beşiktaş örneğinde ilk aşamada, aktarmasız olarak Kabataş, Eminönü, Kadıköy, Üsküdar ve İstinye olmak üzere beş adet rotanın önerildiği görülmektedir. Bunlardan tekrar eden rotalar elenmiştir. İkinci aşamada, tek aktarma ile saat faktörü de göz önünde bulundurulduğunda 07:00-09:00 saat aralığında Eminönü üzerinden Beylerbeyi'ne, Üsküdar üzerinden Ortaköy'e ve İstinye üzerinden Sarıyer'e ulaşmak mümkündür. Üçüncü ve son elemelerde ise yolcunun yalnızca Sarıyer'den Rumeli Kavağı ve Anadolu Kavağı'na geçiş yapabileceği önerilmiştir. Böylece 07:00-09:00 saat aralığında Beşiktaş'tan 11 farklı iskeleye varışın sağlanabildiği görülmektedir (**Şekil 8**).

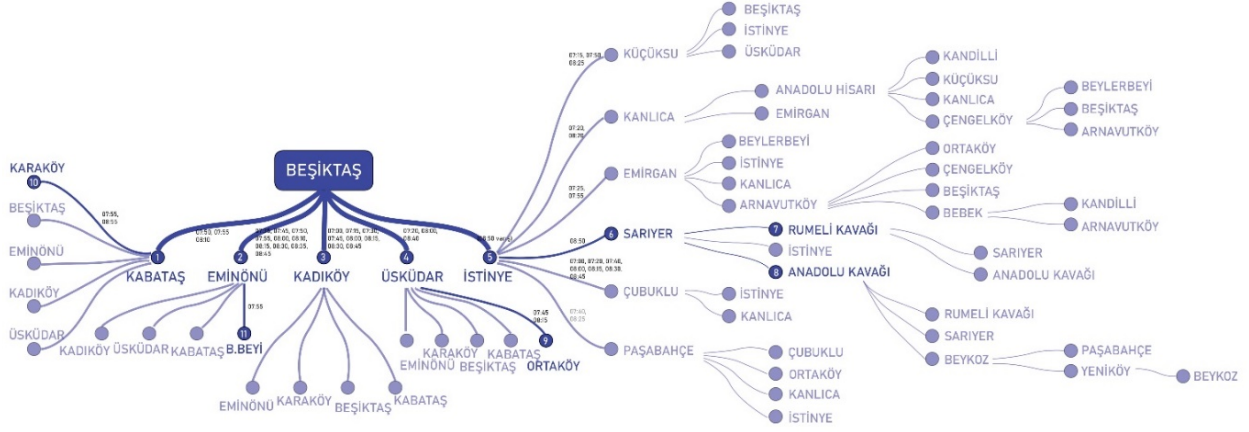
Beşiktaş kök düğüm noktası olarak kabul edilirse aktarmasız olarak Beşiktaş'tan varış sağlanabilen 5 farklı düğüm noktası bulunmaktadır. En iyi öneriye ulaşma amacıyla kullanılan genetik algoritma yöntemindeki gezgin satıcı problemi baz alındığında, 07:00-09:00 saatleri arasında, örneğin Beşiktaş'ın ilk düğüm noktası olarak kabul ettiğimiz Kabataş'tan yalnızca Karaköy'e varışın mümkün olacağı görülür. Kabataş iskelesinden aktarmasız olarak gidilen Beşiktaş iskelesi başlangıç iskelesi olduğu için; Eminönü, Kadıköy, Üsküdar iskeleleri ise Beşiktaş'tan aktarmasız olarak varılabilen iskeleler oldukları için popülasyondan uzaklaştırılırlar.

**Şekil 7**'de de görüldüğü gibi bu en iyileme probleminin ilk aşamasında belirlenen başlangıç ve/veya varış saatine göre ilk alternatif aktarmasız olmak üzere en uygun alternatif arayışında aktarmalar birer artırıldığında bazı seçeneklerin de elenmesi söz konusudur.

**Şekil 7:** E-Bosphorus karar destek sisteminin akış şeması.

## E-BOSPHORUS KARAR DESTEK SİSTEMİ





**Şekil 8:** Beşiktaş'tan 07:00-09:00 saatleri arasında kullanıcıya önerilen alternatifler, karar ağacı.

1 numaralı hat (Düğüm 1: Kabataş) Beşiktaş'tan Kabataş'a 07:50, 07:55 ve 08:10 saatlerinde hareket etmektedir. Yolculuk süresi 5 dakikadır. Kabataş'tan 7:50'de kalkan diğer vapur ise 7:55'te yapılacak olan Kabataş-Karaköy vapuruna sınırda yetişebilecektir. Bu nedenle Beşiktaş'tan Karaköy'e vapurla ulaşmak isteyen bir yolcu için bu vapur sefer saatleri verimsiz olmaktadır. Burada İstanbul Büyükşehir Belediyesi tarafından bir düzenleme yapılması gerektiği ortaya çıkmaktadır. Beşiktaş'tan Kabataş'a 08:10'da kalkan vapurla ise 08:15'te varış sağlanacaktır. Ancak bir sonraki vapur 08:55'te kalkacağı için yolcu 30 dakika beklemek durumunda kalacaktır, bu da birbirine yakın mesafede olan bu güzergah için optimum bir sonuç olmayacaktır.

2 numaralı hat (Düğüm 2: Eminönü) Beşiktaş'tan Eminönü'ne 07:35, 07:45, 07:50, 07:55, 08:00, 08:10, 08:15, 08:30, 08:35, 08:45 saatlerinde hareket etmektedir. Yolculuk süresi 15 dakikadır. Eminönü'nden Beylerbeyi'ne geçiş yapabilmek için 07:00-09:00 saatleri arasındaki en ideal saat ise 07:55'tir. Bu durumda Beşiktaş'tan Eminönü'ne giden 07:35'te kalkan vapur Beylerbeyi'ne geçiş için tek seçenektir. Buradan Beylerbeyi ise 30 dakika sürmektedir. Eminönü'ndeki beş dakikalık bekleme süresi de hesaba dahil edilirse karar destek sisteminin önerdiği Beşiktaş-Eminönü-Beylerbeyi rotasında toplam yolculuk süresi 50 dakikadır.

3 numaralı (Düğüm 3: Kadıköy) Beşiktaş-Kadıköy hattında 07:00, 07:15, 07:30, 07:45, 08:00, 08:15, 08:30, 08:45 saatlerinde vapur seferleri bulunmaktadır. Ancak Beşiktaş-Kadıköy hattından sonraki aktarmasız seferler algoritmadan çıkarıldığı için 07:00-09:00 arasında, Boğaz



Hattında Beşiktaş'tan Kadıköy'e aktarmalı bir vapur seferi bulunmamaktadır.

4 numaralı hat (Düğüm 4: Üsküdar) Beşiktaş'tan Üsküdar'a 07:20, 08:00, 08:40 saatlerinde vapur seferleri bulunmaktadır. Üsküdar'dan ise algoritmadan elenen hatlar dışında tek alternatif olarak Ortaköy hattı kalmaktadır. Üsküdar-Ortaköy seferleri ise 07:45 ve 08:15'te yapılmaktadır. Beşiktaş-Üsküdar arası yolculuk süresi 20 dakikadır. Bu durumda yalnızca 07:20 seferiyle Beşiktaş'tan Ortaköy'e varış mümkündür. Üsküdar-Ortaköy vapur seferinin süresi 15 dakikadır. Beşiktaş'tan Ortaköy'e toplam yolculuk süresi 40 dakika sürecektir. Yapılan mobil uygulamada bu rota öneri olarak sunulacaktır. Trafik yoğunluğuna bağlı olarak yolcu bu rotayı tercih edebilir ya da karar destek sistemi belediyenin Beşiktaş-Ortaköy rotasındaki yolcu yoğunluğuna bağlı olarak sefer açmasına yönelik bir öneri oluşturabilir.

5 numaralı (Düğüm 5: İstinye) Beşiktaş-İstinye hattında 08:15'te bir vapur seferi bulunmaktadır. Beşiktaş'tan bu hatla İstinye'ye varış 08:50'dir. Yine 08:50'de İstinye'den Sarıyer'e varış mümkündür. Sarıyer'den Anadolu Kavağı'na bir sonraki sefer 09:00'dadır ve yolculuk süresi 10 dakikadır. Daha geç bir vakitte Rumeli Kavağı'na da buradan geçiş mümkündür (Tablo 1).

**Tablo 1:** Beşiktaş'tan 07:00-09:00 saatleri arasında kullanıcıya önerilen alternatifler (R. Kavağı hariç)

NO 1	HAT BAŞLANGICI	ROTALAR	BAŞLANGIÇ SAATİ	YOLCULUK SÜRESİ
R1	BEŞİKTAŞ	KABATAŞ	07:50, 07:55, 08:10	5 Dakika
R2	BEŞİKTAŞ	EMİNÖNÜ	07:35, 07:45, 07:50, 07:55, 08:00, 08:10, 08:15, 08:30, 08:35, 08:45	15 Dakika
R3	BEŞİKTAŞ	KADIKÖY	07:00, 07:15, 07:30, 07:45, 08:00, 08:15, 08:30, 08:45	25 Dakika
R4	BEŞİKTAŞ	ÜSKÜDAR	07:20, 08:00, 08:40	20 Dakika
R5	BEŞİKTAŞ	İSTİNYE	08:15	35 Dakika
R6	BEŞİKTAŞ	İSTİNYE SARIYER	08:50	35 + 25 Dakika
R7	BEŞİKTAŞ	İSTİNYE SARIYER R. KAVAĞI	12:05, 12:25	35 + 25 + 10 Dakika
R8	BEŞİKTAŞ	İSTİNYE SARIYER A. KAVAĞI	09:00	35 + 25 + 10 Dakika
R9	BEŞİKTAŞ	ÜSKÜDAR ORTAKÖY	07:45, 08:15	20 + 15 Dakika
R10	BEŞİKTAŞ	KABATAŞ KARAKÖY	07:55, 08:55	5 + 10 Dakika
R11	BEŞİKTAŞ	EMİNÖNÜ BEYLERBEYİ	07:55	15 + 30 Dakika



**Şekil 9:** Beşiktaş'tan 07:00-09:00 saatleri arasında kullanıcıya önerilen alternatifler, haritalama.

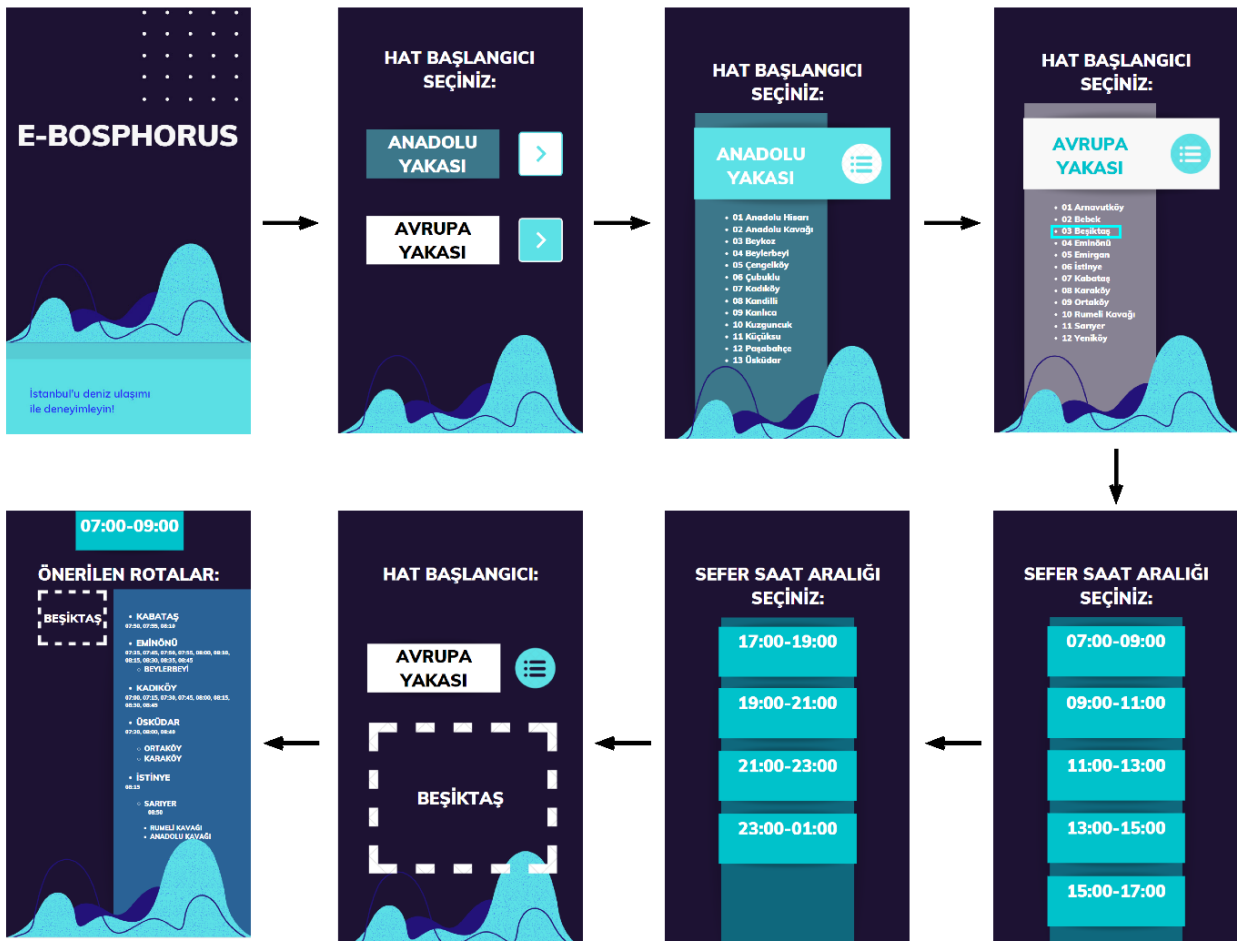
Kullanıcıya 07:00-09:00 saatleri arasında hat başlangıcı Beşiktaş olarak sunulan alternatifler **Şekil 9**'daki gibidir. **Şekil 9**'daki koyu mavi renkteki 1, 2, 3, 4 ve 5 numaralı oklar yolcunun tek aktarmayla, kırmızı renkteki 6, 9, 10, 11 numaralı oklar iki aktarmayla, yeşil renkteki 7 ve 8 numaralı oklar ise üç aktarmayla yolcunun belirtilen saat aralığında hat başlangıç noktası Beşiktaş olan vapur vb. deniz ulaşım araçları ile kullanabileceği rotaları göstermektedir.

Bu öneriler 07:00-09:00 zaman aralığı için kısıtlı sayıda olsa da diğer saatlerde daha çok sayıda alternatiflerle karşılaşılmıştır. Bu nedenle çalışmada kurgulanan sistemin diğer saatleri de kapsamının model önerisi açısından daha gelişmiş bir sonuç vereceği sonucuna varılmıştır. Ancak çalışmanın sınırları çerçevesinde bu saat aralıkları test edilmiştir.

Geliştirilen mobil uygulama önerisine göre kullanıcıdan hat başlangıcı bilgisi alınır. Bu bilgi Avrupa Yakası ve Anadolu Yakası olarak kategorize edilmiş bir arayüz üzerinden alınır. Kullanıcının diğer hatlara yönelik olarak da bilgi sahibi olabilmesi adına bu yakalardaki iskeleler listelenir. Kullanıcı hat başlangıcını bu listeden seçer. Buna ek olarak kullanıcı saat aralığı seçimi yapar. Burada ise ikişer saatlik aralıklara bölünerek kullanıcıya belirlediği saat aralıklarında ve seçtiği hat başlangıcına yönelik olarak rota önerilerinde bulunulur (Şekil 10).

Ayrıca kullanıcıların oylaması ile de İstanbul Büyükşehir Belediyesi'ne vapurların sefer sıklığına yönelik oy verisi iletilebilir. Böylece bu uygulama yoluyla, katılımcı bir yaklaşımla, kentlinin vapur seferlerine yönelik talebi alınabilir.

Şekil 10: Mobil uygulama arayüzü önerisi.



#### 4. SONUÇ VE TARTIŞMALAR

İstanbul Boğazı şartları itibariyle ulaşımda son derece dikkat gerektiren önemli bir su yoludur. Bu su yolundaki ulaşımın verimli kullanımı Boğaz Hattındaki kara yolu trafiğinin minimize edilmesini sağlayacaktır. Mevcut araç trafiğini ve trafikte kaybedilen zamanı azaltmak, kara yolunu kullanan araç sayısının azaltılması ile karbon salınımını da azaltmış olacaktır. Deniz yolu ulaşımının teşvik edilmesi aynı zamanda düşük karbonlu hareketliliği de destekleyecektir. Bu doğrultuda bir karar destek sisteminin kurgulanması ile rotalara ve sefer saatlerine yönelik önemli bir öneri sistemi elde edilir. Boğaz hattında, işe gidiş ve işten çıkış saatlerinde; 07:00-09:00 ve 17:00-19:00 saatleri arasında sefer sayılarının artırılması gereklidir. Bu hatta kıyıya paralel bir deniz yolu ulaşımının verimli bir şekilde sağlanması, bu güzergahtaki kara yolu erişimini de rahatlatacaktır.

Bu çalışmada bahsi geçen mobil uygulamaya benzer uygulamalar Google Maps, Mobiett, Trafi, Moovitapp gibi uygulamalardır. Ancak bu uygulamalarda vapur seferleri özelinde yeterli bilgi bulunmamaktadır. Önerilen karar destek sisteminde aktarmalar da sisteme dahil edilerek çeşitli rotalar önerilebilecektir. Uygulamada kullanıcı oyuyla veri toplanarak belediyeye de paylaşılabılır. Deniz yolu ulaşımını teşvik eden bir mobil uygulama olması ve kentlinin oyuyla akıllı ulaşımda katılımıcılığın desteklenmesi bu uygulama önerisinin inovatif yönlerinden biridir.

Bu çalışma, İstanbul Büyükşehir Belediyesi'nin 2020-2024 amaç ve hedefleri kapsamında (Url-8), 'Ulaşılabilir İstanbul' teması başlığı altında, 'Sürdürülebilir Hareketlilik Kapsamında Kentsel Ulaşımı Geliştirmek' amacının alt maddelerinde yer alan 'Akıllı Ulaşım Sistemlerini ve Ulaşım Altyapı Uygulamalarını Artırarak Trafiği Etkin Yönetmek (H4)', 'Deniz Yolu Taşımacılığının Kapasitesini ve Toplum İçindeki Payını Artırmak (H3)' ve 'Toplu Taşımada Entegrasyonu, Erişilebilirliği ve Kaliteyi Artırmak (H2)' hedefleriyle ilişkilidir ve bu tür uygulamalar kamusal yarar sağlamaktadır. Buna ek olarak güncel gelişmelerden olan, Cumhurbaşkanlığı tarafından 05.08.2020 tarihli 31204 sayılı Resmi Gazete'de yayımlanan Ulusal Akıllı Ulaşım Sistemleri Strateji Belgesi ve Eylem Planı bağlamında çalışma; seyahat sürelerinin kısaltılması, mevcut yol kapasitelerinin etkin ve verimli kullanılması, enerji verimliliği sağlanarak ülke ekonomisine katkı sağlanması ve

çevreye verilen zararların azaltılması gibi amaçlara da uygunluk göstermesi açısından önemlidir.

Çalışma sonucunda, şehir içi akıllı deniz yolu ulaşımında optimum çözümün elde edilebilmesi için İstanbul Büyükşehir Belediyesi (İBB) Ulaşım Koordinasyon Müdürlüğü, Şehir Hatları A.Ş., Turyol, Dentur vb. kurum/kuruluş ve şirketler tarafından sefer saatlerinin belirlenmesinde ortak bir karar destek sistemi kullanımının gerekliliği ortaya çıkmaktadır.

İstanbul şehir içi ulaşımında kara yolu, raylı sistemler gibi çözümlerin yanı sıra deniz yolu ulaşımından da yararlanarak entegre bir ulaşım ağının oluşması gereklidir. Oluşturulan deniz yolu ulaşım ağı ile işe gidiş ve işten dönüş amaçlı kullanımının yanı sıra kentlinin gezme, etkinliğe katılma vb. çeşitli yollarla da kullanımı da mümkün olabilecektir.

Gelecekte yapılacak olan çalışmalarda boğaz hattındaki trafik yoğunluğuna ve iskeledeki yolcu yoğunluğuna bağlı olarak vapur sefer sıklığının belirlenmesi ile iskele tasarımları, büyüklükleri ve yine bunlarla orantılı olarak vapur büyüklüklerinin önerilmesi mümkündür. Ayrıca yalnızca 07:00-09:00 ve 17:00-19:00 saat aralıklarında yapılan bu araştırmanın sınırları dışında, tüm saat aralıklarının (06:00-01:00) bu sisteme entegre edilmesiyle de kentlinin daha fazla aktarma alternatiflerini görerek kendi rotasını belirlemesi mümkün olabilecektir. Bunlara ek olarak deniz yolu toplu ulaşımını kara yolu toplu ulaşımına, elektrikli bisiklet duraklarına, elektrikli kaykayların konumlarına (e-scooter) entegre etmek mümkün olabilir. Sistem derin öğrenme ile saatleri ve rotaları sabitleyebilir ya da kullanıcı önerisiyle sisteme yeni rotalar eklenebilir. Ücretlendirme faktörü de bu bağlamda ileride yapılacak olan çalışmalarda değerlendirilebilir.

## **Teşekkür**

Bu çalışma İstanbul Teknik Üniversitesi, Mimari Tasarımda Bilişim Lisansüstü Programı, Mimari Tasarımda Karar Destek Sistemleri doktora dersi kapsamında gerçekleştirilmiştir. Çalışmada emeği geçen ders yürütücüsü Prof. Dr. Gülen Çağdaş'a yaptıkları katkı ve ayırdıkları değerli zaman için en içten teşekkürlerimizi sunarız.

## Referanslar

- Albayrak, A. S., ve Yılmaz, Ş. K. (2009). Veri Madenciliği: Karar Ağacı Algoritmaları ve İMKB Verileri Üzerine Bir Uygulama. *Süleyman Demirel Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi* 14 (1), 31-52.
- Beasley, D., Bull, D. ve Martin, R. (1993). An Overview of Genetic Algorithms: Part 1&2. *Fundamentals. University Computing* 15 (2), 58–69.
- Bingul, Z., Sekmen, A. ve Zein, S. (1999). An Application of Multi-Dimensional Optimization Problems Using Genetic Algorithms. In *Proceedings of the International Conference Intelligent Systems and Control (IASTED)*. Santa Barbara, CA.
- Ceder, A. ve Sarvi, M. (2007). Design and Evaluation of Passenger Ferry Routes. *Journal of Public Transportation* 10 (1) , 59-79. doi: 10.5038/2375-0901.10.1.3.
- Çalış, A., Kayapınar, S. ve Çetinyokuş, T. (2014). Veri Madenciliğinde Karar Ağacı Algoritmaları ile Bilgisayar ve İnternet Güvenliği Üzerine Bir Uygulama. *Endüstri Mühendisliği Dergisi* 25 (3-4), 2-19.
- Çancı, M., Önden, İ., Çakmak, E., Gürel, Ö. ve Tuzla, H. (2015). İstanbul'da Deniz Ulaşımının Geleceğinin Değerlendirilmesi. İstanbul: İstanbul Kalkınma Ajansı. Erişim adresi: <<https://www.istka.org.tr/media/20863/%C4%B0stanbul-dadeniz-ula%C5%9F%C4%B1m%C4%B1n%C4%B1n-gelece%C4%9Fininde%C4%9Ferlendirilmesi.pdf?/>>, erişim tarihi: 18.04.2020.
- Dener, M., Akcayol, M. A., Toklu, S. ve Bay, Ö. F. (2011). Zamana Bağlı Dinamik En Kısa Yol Problemi İçin Genetik Algoritma Tabanlı Yeni Bir Algoritma. *Gazi Üniversitesi Mühendislik Mimarlık Fakültesi Dergisi* 26 (4), 915-928.
- Dijkstra, E. W. (1959). A Note On Two Problems in Connexion With Graphs. *Numerische Mathematik* 1 (1), 269-271.
- Emel, G. G. ve Taşkın, Ç. (2005). Veri Madenciliğinde Karar Ağaçları ve Bir Satış Analizi Uygulaması. *Eskişehir Osmangazi Üniversitesi Sosyal Bilimler Dergisi* 6, 221-239.
- Han, J., Pei, J., & Kamber, M. (2011). *Data Mining: Concepts and Techniques* (3rd Ed.). Burnaby: Morgan Kaufman.
- Url-1 <<https://www.iETT.istanbul/tr/main/pages/mobiETT/775>>, erişim tarihi 18.04.2020.
- Url-2 <<https://moovitapp.com/>>, erişim tarihi 18.04.2020.
- Url-3 <<https://web.trafi.com/tr/istanbul/vapur>>, erişim tarihi 18.04.2020.
- Url-4 <<https://www.google.com/maps/>>, erişim tarihi 18.04.2020.
- Url-5 <<https://www.openstreetmap.org/>>, erişim tarihi 18.04.2020.
- Url-6 <<http://overpass-turbo.eu/>>, erişim tarihi 23.04.2020.
- Url-7 <<https://www.sehirhatlari.istanbul/tr/seferler/ic-hatlar/istanbul-ici-hatlar-26>>, erişim tarihi 23.04.2020.

Url-8 <<https://www.ibb.istanbul/Uploads/2020/2/IBB-STRATEJIK-PLAN-2020-2024.pdf>>, erişim tarihi 30.04.2020.

# Regenerating Modern Interiors into Science Fiction Environments via Cycle-GAN

Esra Yağdır Çeliker<sup>1</sup>, Gizem Efendioğlu<sup>2</sup>, Özgün Balaban<sup>3</sup>

0000-0002-1817-3829<sup>1</sup>, 0000-0002-0586-6167<sup>2</sup>, 0000-0002-7270-2058<sup>3</sup>

<sup>1</sup>Istanbul Bilgi University, Faculty of Architecture, Department of Architecture, Istanbul, Turkey

<sup>2</sup>Özyeğin University, Faculty of Architecture and Design, Department of Architecture, Istanbul, Turkey

<sup>3</sup>Delft University of Technology, Faculty of Architecture and the Built Environment, Chair of Design Informatics. Delft. Netherlands

Deep learning is a machine learning method that uses artificial neural networks for complex tasks and problems that require the processing of large data sets. Deep learning has shown that it is possible to process the properties of the data that previously needed to be transferred to the computer by an expert person, only by a computer. Generative Adversarial Networks (GAN) algorithm, one of the subsystems of deep learning, takes advantage of the contention of two neural networks working opposite each other. While the Generator produces fake images, the Discriminator evaluates the images and generates the information that the image is fake or real. This contentious situation between two networks repeats until the Discriminator cannot distinguish the image is fake. For this reason, researchers prefer to use the GAN especially in image processing and image translation problems. With the image processing techniques offered by deep learning, it is possible to process complex spatial data and to reproduce spatial fictions through images. The study aims to investigate the new spatial potentials of interior spaces with different characteristics. In this context, modern interiors are reinterpreted as dystopic science fiction spaces by using the GAN algorithm, which is a suitable technique for image processing. In this study, we created two different data sets from modern interior photographs and science fiction movies. Thus, we tried to investigate how modern interiors can change morphologically when they become a part of science fiction movies.

**Received:** 24.08.2020

**Accepted:** 27.09.2020

**Corresponding Author:**

yagdir.celiker@bilgi.edu.tr

Çeliker, Y. E., Efendioğlu, G., Balaban, Ö. (2020). Regenerating Modern Interiors into Science Fiction Environments via Cycle-GAN, JCoDe: Journal of Computational Design, 1(3), 71-94.

**Keywords:** Cycle-GAN, Deep Learning, Machine Learning, Spatial Image Processing, Artificial Intelligence in Architecture.



# Cycle-GAN ile Modern İç Mekânların Bilim Kurgu Ortamları Olarak Yeniden Üretilmesi

Esra Yağdır Çeliker<sup>1</sup>, Gizem Efendioğlu<sup>2</sup>, Özgün Balaban<sup>3</sup>

0000-0002-1817-3829<sup>1</sup>, 0000-0002-0586-6167<sup>2</sup>, 0000-0002-7270-2058<sup>3</sup>

<sup>1</sup>İstanbul Bilgi Üniversitesi, Mimarlık Fakültesi, Mimarlık Bölümü İstanbul, Türkiye

<sup>2</sup>Özyeğin Üniversitesi, Mimarlık ve Tasarım Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

<sup>3</sup>Delft University of Technology, Faculty of Architecture and the Built Environment, Chair of Design Informatics, Delft, Netherlands

Derin öğrenme, karmaşık görevlerin ve büyük veri setlerinin işlenmesi gereken problemlerde, yapay sinir ağlarını kullanan bir makine öğrenmesi yöntemidir. Derin öğrenme ile daha önceden uzman bir insan tarafından bilgisayara aktarılması gereken veriye ait özelliklerin, salt bilgisayar tarafından işlenmesi mümkün hale gelmiştir. Derin öğrenmenin alt sistemlerinden biri olan Üretken Rakip Yapay Sinir Ağları (GAN) algoritması, birbirine zıt çalışan iki sinir ağının birbiri ile çekişmesinden faydalanmaktadır. Üretici ağ gerçek olmayan görseller üretirken, ayırt edici ağ, üretilen görselleri değerlendirmekte ve görselin sahte veya gerçek olduğu bilgisini üretmektedir. İki ağ arasındaki bu çekişmeli durum ayırt edici ağın gerçek ile sahteyi ayıramayacağı kadar kaliteli görseller üretilene kadar tekrarlanmaktadır. Bu nedenle GAN algoritması özellikle görüntü işleme ve görüntü çeviri problemlerinde tercih edilmektedir. Derin öğrenmenin sunduğu görüntü işleme teknikleri ile karmaşık mekânsal verilerin kurgulanması ve mekânsal kurguların görüntüler üzerinden tekrar üretimi mümkündür. Bu çalışmanın amacı, farklı özelliklere sahip iç mekânların bir bilim kurgu filminin parçası olma durumunu ve bu durumdan türeyen yeni mekânsal potansiyelleri araştırmaktır. Bu bağlamda görüntü işleme için uygun bir teknik olan GAN algoritması kullanılarak, modern iç mekânlar distopik bilim kurgu mekânları olarak yeniden yorumlanmıştır. Çalışmada modern iç mekân fotoğraflarından ve bilim kurgu filmlerinden olmak üzere iki farklı veri seti oluşturulmuştur. Böylece modern iç mekânların bilim kurgu filmlerinde yer alması durumunda söz konusu mekânların morfolojik olarak nasıl yorumlanabileceği araştırılmıştır.

**Anahtar Kelimeler:** Cycle-GAN, Derin Öğrenme, Makine Öğrenmesi, Mekânsal Görüntü İşleme, Mimarlıkta Yapay Zekâ.

**Teslim Tarihi:** 24.08.2020

**Kabul Tarihi:** 27.09.2020

**Sorumlu Yazar:**

yagdir.celiker@bilgi.edu.tr

Çeliker, Y. E., Efendioğlu, G., Balaban, Ö. (2020). Cycle-GAN ile Modern İç Mekânların Bilim Kurgu Ortamları Olarak Yeniden Üretilmesi, JCoDe: Journal of Computational Design, 1(3), 71-94.

## 1. GİRİŞ

Günümüzde bilgisayar biliminin alt çalışma alanlarından bir olan yapay zekâ (artificial intelligence) uygulamaları birçok bilimsel araştırma ve meslek pratiği uygulamalarında olduğu gibi mimari tasarım araştırmalarında ve uygulamalarında da gün geçtikçe aktif bir biçimde kullanılmaktadır. Simon'un (1972) tasarımı kötü tanımlanmış (ill-defined) bir problem olarak tanımlamasıyla, tasarım odaklı düşünme süreci, problemin tanımı ve elde edilmesi planlanan çıktılar arasında sürdürülecek kurallar dizisi olarak kurgulanabilmiş, algoritma tabanlı yaklaşım mimari tasarım süreçlerine entegre edilebilmiştir. Böylece hesaplamalı mimari tasarım yöntem ve uygulamaları güncel mimarlık teorisi ve pratiğinde etkin bir biçimde yer alarak insan tasarımcının işleyemeyeceği yoğunlukta, çeşitlilikte ve büyüklükteki veriler yapay zekâ araştırmalarının bir dalı olan makine öğrenmesi uygulamaları kullanılarak işlenebilir hale gelmiştir. Bilgisayar bilimindeki bu gelişmeler bir yandan mimari tasarımda biliş araştırmalarından beslenmiş, insanın bilgi işleme mekanizması makinalara yapay sinir ağları olarak aktarılabilmiştir. Böylece veri işleme süreci statik ve öngörülebilir çıktılar olarak işleyen makinalar yerine, özgün ve yeni bilgi üretebilen, başka bir deyişle öğrenebilen makinalar ile gerçekleştirilmeye başlanmıştır. Yapay zekâ bilgisayarlara zekice hareket edebilmeleri için yeterli çok yönlülüğün sağlanması olarak tanımlanırken (Nilsson, 1980), makine öğrenmesi (machine learning) matematiksel ve istatistiksel işlemler kullanılarak bilgisayarın çıkarımlar yapmasını ve deneyimler ile öğrenme yetisini kazanmasını sağlayan mekanizma olarak ifade edilmektedir (Şafak, 2017). Yapay zekâ uygulamalarındaki çok yönlülüğün sağlanmasıyla bilgisayarın tek başına problem çözebilecek bilgi işleme kapasitesine sahip olması amaçlanmış, böylece hızlı ve mantıksal çıkarımlar yapabilen makinalar ile veri işleme sürecinin gerçekleşmesi hedeflenmiştir. Söz konusu hedefler göz önüne alındığında problem çözümünde gerekli olan zihinsel yetkinlikler ile öğrenme ve strateji belirleme gibi tekniklerin bilgisayara kazandırılması yapay zekânın temel amaçları haline gelmiştir.

Yukarıda bahsedilen yapay zekâ ve makine öğrenmesi çalışmalarının kapsamı, mimari tasarım araştırmalarında yapma bilgisi (techne) ve gerçek/öz teorik bilgi (episteme) üretiminde de insan tasarımcıya destek olmakta ve özgün tasarım yaklaşımlarını ortaya çıkarma, büyük veriyi işleme olanağı sağlamaktadır. Özellikle tasarımda optimizasyon

süreçlerinin hızlandırılması, gömülü ilişkilerin açığa çıkarılması bağlamında yapay zekâ ve makine öğrenmesi uygulamaları insan tasarımcıya geçmişe kıyasla oldukça geniş bir yelpazede yaratıcı ve özgün üretim yapabilme olanağını sağlamaktadır. Makine öğrenmesi aynı zamanda görüntü işleme (image processing) uygulamalarında sıklıkla kullanılmakta ve mekânsal verinin insan tasarımcının elde edemeyeceği bir karmaşıklıkta işlenebilmesine olanak sağlamaktadır. Böylece mekânın görüntüler üzerinden tekrar yorumu ve değerlendirilmesi yapılabilirken, özgün mekânsal kurguların üretimi olanaklı hale gelmektedir. Bu bağlamda çalışma, güncel hesaplamalı tasarım araştırmalarında sıklıkla kullanılmaya başlanılan makine öğrenmesinin bir alt çalışma alanı olan derin öğrenme (deep learning) araçlarından biri olan Üretken Rakip Yapay Sinir Ağları (GAN) algoritmaları ile farklı morfolojik ve topolojik özelliklere sahip iç mekânların bir bilim kurgu filminin parçası olma durumunu ve bu durumdan türeyen yeni mekânsallaşma potansiyellerini araştırmaktadır. Söz konusu mekânsallaşma potansiyellerinin araştırılmasıyla verinin yaratıcı ve özgün tasarım çıktıları elde edilmesi süreçlerine kazandırılacak olası katkılar üzerinde tartışılması da gelecekte yapılacak araştırmalar için oldukça aydınlatıcı olacaktır. Çalışmanın ana araştırma sorusu “İç mekânların kendi bünyelerinde bulundurduğu gömülü (embedded) veriler Üretken Rakip Yapay Sinir Ağları kullanılarak yeni mekânsal formasyonlar doğurabilir mi?” olarak tanımlanabilir. Söz konusu “yeni mekân” makinenin iki farklı atmosfere ait mekân görüntülerini işleyerek özgün çıkarımlar yapması ve bu çıkarımların sonucunda iki farklı veri setinden yeni mekânsal bilgiler elde etmek olarak tanımlanabilir. Söz konusu çalışmada geçmişten günümüze kadar Royal Institute of British Architects (RIBA) dijital mimarlık kütüphanesinde (Url-2) bulunan, inşa edilmiş iç mekân tasarımlarından oluşmuş bir veri seti ve distopik mekânsal referanslara sahip olan dört farklı bilim kurgu filminin iç mekân fragmanlarından hazırlanmış veri setleri kullanılarak, modern iç mekânların bilim kurgu filmlerinin atmosferlerinin bir parçası olacak şekilde tekrar üretimleri yapılmıştır. Böylece modern iç mekân tasarımlarının bilim kurgu filmine entegre edilmesi durumunda söz konusu mekânların morfolojik olarak nasıl evrileceği araştırılmıştır. Araştırma sırasında Üretken Çelişmeli Yapay Sinir Ağı (GAN) algoritmalarından biri olan Cycle-GAN algoritması kullanılmış ve çalışma için sinema tarihinin kült bilim kurgu filmlerinden Blade Runner, Metropolis, 2001: A Space Odyssey ve Star Wars Serisi seçilmiştir.

## 2. LİTERATÜR ARAŞTIRMASI

Literatür araştırması yapay zekâ, makine öğrenmesi ve derin öğrenme kavramları üzerine yapılan çalışmaların mimari tasarım çalışmaları ile olan ilişkilerine odaklanmadır. Böylece genetik algoritmaların mimari tasarım süreçlerine olan katkılarının belirlenmesi amaçlanmıştır.

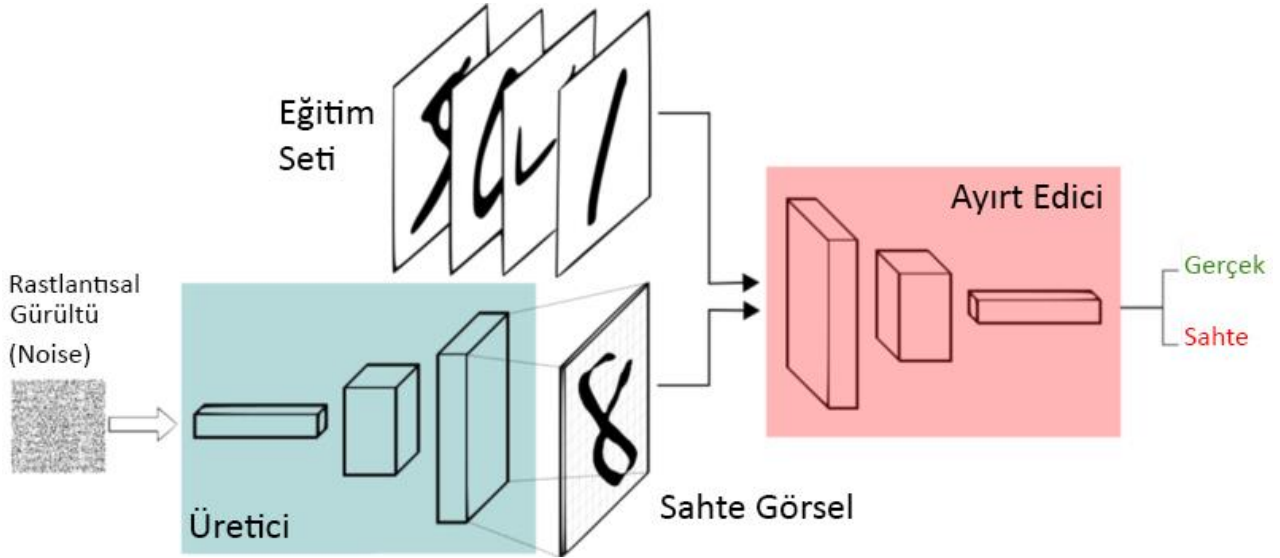
Yapay zekâ araştırmalarının önemli bir alt çalışma alanı olan genetik algoritmalar kullanılarak geliştirilmiş derin öğrenme, karmaşık görevlere yeni çözümler üretebilmek için sinir ağlarını (Neural Networks) kullanan bir makine öğrenmesi yöntemi olarak tanımlanabilir (Hitaj ve ark., 2017). Derin öğrenme ile veriye ait özelliklerin (features) bir insan tarafından öğretilmesinin zorunlu olmadığı, bilgisayarın da bu işlemi gerçekleştirebileceği bir sistemin var olabileceği görülmüştür. Böylece büyük verilerin kullanıldığı karmaşık problemlerin çözümünde bu yöntemden faydalanılmaktadır. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki Şekil 1’de görülmektedir.



Şekil 1: Yapay Zekâ, Makine Öğrenmesi ve Derin Öğrenme arasındaki ilişki (Zbigniew ve ark., 2018).

Goodfellow ve ark. (2014), derin öğrenmedeki sinir ağları teorisini kullanarak iki farklı sinir ağının birbirine zıt şekilde çalıştığı Üretken Rakip Yapay Sinir Ağları (GAN, Generative Adversarial Networks) algoritmasını geliştirmiştir. Bu algoritma ile bilgisayar, eğitim seti (training set) istatistiklerini kullanarak yeni bir set oluşturmaktadır. Görüntü bölümlenme, soyutlama ve biçimlendirme gibi birçok görüntü işleme görevi, görüntüden görüntüye çeviri (image-to-image translation) problemi olarak tanımlanabilir (Isola ve ark., 2017). Bu nedenle görüntüden görüntüye çeviri problemlerinde, birbirine zıt iki farklı sinir ağı gibi çalışan GAN algoritması uygun yöntemi sağlamaktadır. GAN

algoritmasına üretici (Generator), baştaki verilerden rastgele bilgiler olarak gerçek olmayan görseller üretmeye çalışmaktadır (Goodfellow ve ark., 2014). Ayırt edici (Discriminator) ise sahte görselleri değerlendirerek bunların gerçek veya sahte olduğunun bilgisini üretmektedir. Bu işlem ayırt edici birimin (discriminator) gerçek ile sahteyi ayıramayacağı kadar kaliteli görseller üretilene kadar gerçekleştirilmektedir (Şekil 2).



GAN optimizasyonun matematiksel karşılığı ise aşağıda tanımlanan **Denklem 1** ile açıklanmıştır (Abdulkader ve ark., 2016).

**Şekil 2:** GAN optimizasyonu çalışma prensibi (Silva, 2018).

$$\min_{\theta_G} \max_{\theta_D} \sum_{i=1}^{n^+} \log f(x_i; \theta_D) + \sum_{j=1}^{n^-} \log (1-f(g(z_j; \theta_G); \theta_D)) \quad (1)$$

$x_i$  baştaki veriden alınan gerçek görselleri temsil ederken,  $z_j$  rastgele üretilen görselleri tanımlamaktadır.  $f(x_i; \theta_D)$ , karşılaştığı görsellere göre farklı etiketler üreten ayırt edici sinir ağı olarak kabul edilirse,  $g(z_j; \theta_G)$  rastgele verilerden görüntü üreten üretici sinir ağını tanımlamaktadır. Çalışma esnasında ayırt edici sinir ağının performansını en üst düzeye çıkarabilmek için  $\theta_D$  gradyanı hesaplanmaktadır. Böylece  $f(x_i; \theta_D)$ , orijinal görsel ile üretilen görsel arasındaki farkı değerlendirerek görselin gerçek veya sahte olduğunu değerlendirebilmektedir. Devamında  $\theta_G$  gradyanı hesaplanarak, orijinal görsele en yakın sahte görselin üretilmesini sağlamaktadır (Hitaj ve ark., 2017).

GAN algoritmasının görsel verilerin üretiminde sağladığı verimlilik, algoritmanın görsel odaklı birçok farklı çalışmada tercih edilmesini sağlamıştır. Zebra ve at (Zhu ve ark, 2017) veya kedi, köpek gibi hayvan görsellerinin dönüşümleri ile çizimlerden gerçek aksesuarlara veya kıyafetlere dönüşümler (Huang ve ark., 2018), belirli bir alanın yaz ve kış mevsimine göre dönüşümleri (Zhu ve ark, 2017), fotoğraflardan ressamların stillerine göre tablo dönüşümleri (Lee ve ark, 2018), fotoğraflarda insanların cinsiyetlerini değiştirme veya duygu durumlarına göre yüz ifadelerinin değişimi (Choi ve ark., 2018), el yazısı ile sayı görsellerinin üretimi (Goodfellow, 2014) gibi görüntüden görüntüye çeviri problemlerinde GAN algoritması sıklıkla kullanılmaktadır.



Şekil 3: Büyük kedi, ev kedisi ve köpek görselleri ile oluşturulmuş GAN algoritmasının uygulandığı bir çalışma (Huang ve ark., 2017).

## 2.1 Mimari Tasarımda Üretken Rakip Yapay Sinir Ağları (GAN)'ın Kullanımı

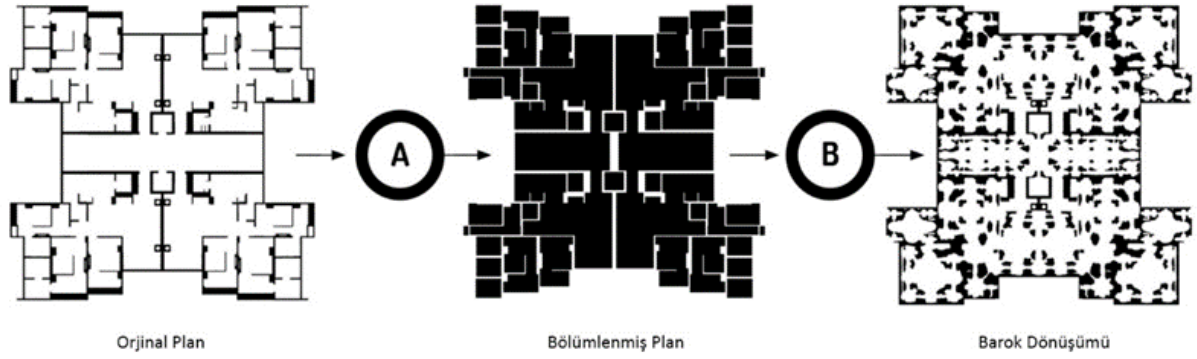
Literatürde mimari tasarım alanında büyük veri kullanılarak özgün ve bütüncül bir mimari üretimin ilk örneği İspanyol Mimar Vignola tarafından 16.yy'da gerçekleştirilmiştir (De Haan akt., As ve diğ., 2018). Vignola bir manastır tasarımı yarışmasından çıkan proje önerilerini kullanarak "yeni" bir manastır tasarımı geliştirmiş ve bu yaklaşım, büyük veriyi kullanarak kolektif mimari üretimin ilk örneklerinden biri olmuştur (As ve diğ., 2018). Söz konusu çalışma, yarışma katılımcılarının

hakları açısından bazı tartışmaların önünü açmasına rağmen açık kaynak kullanımının da öncüsü olarak tanımlanabilir.

1960'lı yıllara gelinmesiyle yapay zekâ araştırmaları hız kazanmış ve makine öğrenmesi, derin öğrenme, genetik algoritmalar geliştirilmeye başlanmış ve mimari tasarım araştırmalarında kullanımı yıllar içinde artış göstermiştir. Hopfield (1982) makine öğrenmesini genetik algoritmalar gibi çeşitli kural dizileri kullanan üretken hesaplamalı tasarım yaklaşımı bütünü olarak tanımlamıştır. Genetik algoritmalar biyolojik evrim operasyonları olarak tanımlanabilecek mutasyon, doğal seçim ve gen aktarımı prensiplerine dayanan, özellikle yüksek çözünürlüklü görüntü işleme ve optimizasyon problemlerinde sıklıkla kullanılan bir algoritma çeşidi olarak tanımlanabilir (Eiben, 1994).

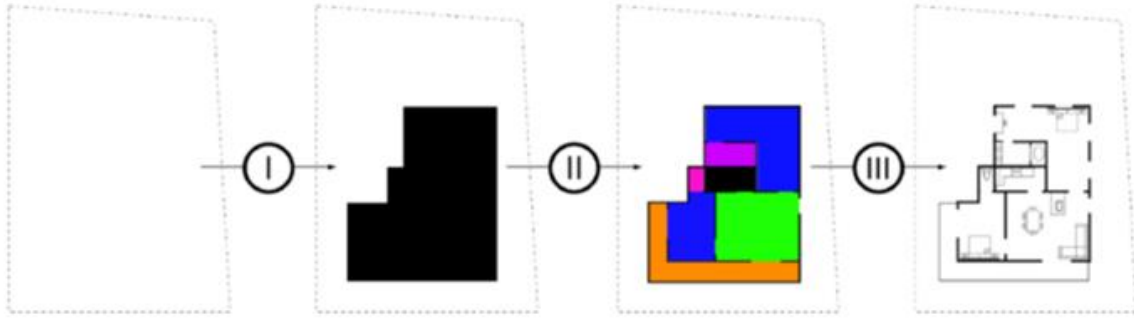
Genetik algoritmalar ilk olarak 1960 yılında John Holland tarafından Darwin'in Evrim Teorisi baz alınarak geliştirilmiş, ardından Holland'ın öğrencisi David E. Goldberg 1989 yılında genetik algoritmaların kullanım alanlarını genişleten çalışmada bulunmuştur (Mitchell, 1997). John Frazer (1995), Reptile Structure adını verdiği çalışmada evrimsel algoritmaları kullanarak mimari strüktürler geliştirmeye çalışmıştır. John Gero, 1996 yılında evrimsel sistemler ve sinir ağlarını kapsayan çalışmasını paylaşmıştır.

Yapay zekâ çalışmalarının gelişimi ile mimarlıkta makine öğrenmesinin kullanımı öncelikli olarak tasarım üretimi, biçim tanımlama, tasarım alanının keşfi ve sınıflandırması alanlarına odaklanmıştır (Tamke ve ark., 2018). Chaillou (2019), tez çalışması kapsamında GAN ile mimari tasarım üretimine katkıda bulunmayı amaçlamıştır. Tezindeki çalışmalar plan organizasyonu ve stil aktarımına odaklanmıştır. Modern planlar ile barok planlar arasındaki farklılıklara değinen Chaillou, modern planlardan barok üslupta plan tiplerinin üretildiği bir GAN algoritması geliştirmiştir (**Şekil 4**).



**Şekil 4:** GAN ile Modern Plandan Barok Plana Dönüşüm (Chaillou, 2019).

Chaillou mimari plan organizasyonunda, boş parsel üzerinde yapı sınırının oluşturulması, yapı sınırından duvarların ve pencere boşluklarının planlanması, devamında kat planının mobilyalar ile donatılmasını makine öğrenmesi ile geliştirmeyi başarmıştır (**Şekil 5**).



**Şekil 5:** GAN ile plan organizasyonu planlaması (Chaillou, 2019).

Araştırmacı, **Şekil 5**'te görülen parsel üzerinde yapı izinin tanımlanmasında Boston'da bulunan yapıların vaziyet planlarından oluşturdukları büyük bir veri setini öğrenme seti olarak kullanmıştır. Bu vaziyet planlarını, ticari, konut ve endüstriyel plan tipleri olarak etiketlemiş ve 700'den fazla plan tipiyle veri setini oluşturmuştur. Planları tek tek bölümlerine ayırarak renk kodlarıyla işaretlemiştir. Mobilya yerleşimlerinde, önceki aşamada kullanılan renk kodlarına göre o mekânın kullanım amacı belirlenmekte ve o alanda kullanılacak mobilyalar plana yerleştirilmektedir. Örneğin, renk kodunun mavi olduğu alanlar yatak odası olarak belirlenmekte ve böylece bu alanlara mobilya olarak yatak yerleştirildiği görülmektedir.

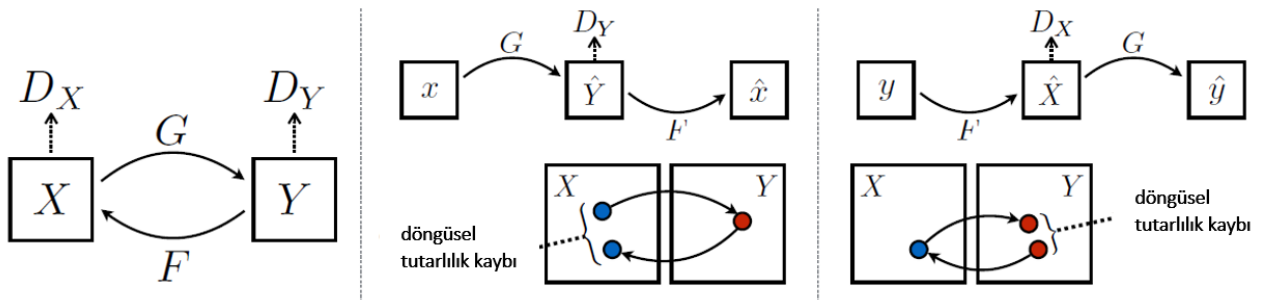
Literatür araştırması bölümünde özetlenen çalışmalarda görüldüğü üzere, mimari tasarımda yapay zekâ uygulamaları farklı amaçlara hizmet edebilecek şekilde kurgulanabilen, esnek, istenildiğinde tasarım kısıtlarına göre evrilebilen bir yapıya sahip bir bilgi işleme aracı olarak



tanımlanabilir. Bu bağlamda özellikle GAN gibi genetik algoritmaların var olan veriyi mimari tasarım kriterlere göre işleyebilmesi ve bu işlemi mantıksal çıkarımlar eşliğinde yapabiliyor olması sayesinde insan tasarımcının öngöremediği potansiyel çözümleri sağlayabildiği ve nihai tasarım çıktılarını çok daha nitelikli hale getirebileceği öngörülebilir.

### 3. YÖNTEM VE ARAÇLAR

Çalışma genellikle iki farklı veri setini birbirine dönüştürmek için kullanılan Cycle-GAN algoritması kullanılarak Gözetimsiz Öğrenme (Unsupervised Learning) yöntemi ile farklı tipoloji ve morfolojilere sahip modern iç mekân görüntülerinin işlenmesini hedeflemektedir. Gözetimsiz Öğrenme, veri setindeki özelliklerin algoritmaya önceden tanımlanmadığı, görüntüdeki herhangi bir bileşenin etiketlenerek derin öğrenme sürecine dahil edilmediği makine öğrenmesi yöntemidir (Duda ve ark. 2001). Böylece etiketlenmemiş/tanımlanmamış veriler arasındaki ilişkilerin algoritmadaki nöronlar tarafından kurulması beklenir. Cycle-GAN ise, temel olarak görüntülerden oluşan veri setinin karakteristik özelliklerini yakalamayı ve bu özelliklerin diğer görüntü veri setine nasıl çevrilebileceğini bulmayı amaçlar (Zhu ve diğerleri, 2018). Bu bağlamda çalışmanın ilk adımı olarak Royal Institute of British Architects (RIBA)'nın açık veri kaynağı olarak kullanıcılara sunduğu dijital mimarlık kütüphanesinden (Url-2) elde edilen modern iç mekân görüntülerinden bir veri seti oluşturulmuş, seçilen dört adet bilim kurgu filmi için söz konusu modern iç mekânlar Cycle-GAN algoritması kullanılarak yeniden üretilmiştir. Cycle-GAN algoritmasının temel yapısı ve çalışma prensibi Şekil 6'da özetlenmiştir.



Şekil 6'de görüldüğü gibi Cycle-GAN algoritması modern iç mekânların bulunduğu X alanına (domain X) tanımlanmış veri seti ve her bir bilimkurgu filminden oluşan birbirinden bağımsız Y alanına (domain Y)

Şekil 6: Cycle-GAN algoritmasının yapısı (Zhu et al.,2018).

tanımlanmış veri setlerine uygulanmak üzere G ve F olarak adlandırılan iki adet eşleşme fonksiyonu (mapping function) içermektedir. Bu iki eşleşme fonksiyonunun X ve Y olarak adlandırılmış veri setleri ile olan ilişkisi aşağıdaki bağıntılar ile tanımlanmıştır (**Denklem 2** ve **Denklem 3**).

$$G:X \rightarrow Y \quad (2)$$

$$F:Y \rightarrow X \quad (3)$$

Çalışmadan kullanılan Cycle-GAN algoritmasında **Denklem 2** ve **Denklem 3** olarak tanımlanan eşleşme fonksiyonlarının yanı sıra, **Şekil 6**'da görülen DX ve DY olarak tanımlanmış olan rakip ayırt ediciler (adversarial discriminators) da X ve Y alanları ile ilişki kurmaktadır (Zhu et al.,2018). Böylece ayırt ediciler, eşleme fonksiyonlarının işlemleri sırasında görüntülerin gerçek veya sahte/makine tarafından yaratılmış olup olmadığını tanımlama işlevini üstlenmektedir (**Şekil 2**). Ayırt edicilerin dahil olduğu ana fonksiyon (**Şekil 2**) aynı zamanda tersine çalışarak birbirinden ayırt edilemeyen çıktılarını saptamaya da olanak vermektedir (Zhu ve diğerleri, 2018).

**Denklem 4** ve **Denklem 5**'de ise eşleşme fonksiyonlarındaki döngüsel tutarlılık kayıplarının (cycle-consistency loss) saptanması hedeflenmektedir (Zhu ve diğerleri, 2018). Söz konusu döngüsel tutarlılık kayıplarının 0'a yaklaşması algoritmanın maksimum verimle verileri işleme anlamına gelmektedir.

$$x \rightarrow G(x) \rightarrow F(G(x)) \approx x \quad (4)$$

$$y \rightarrow F(y) \rightarrow G(F(y)) \approx y \quad (5)$$

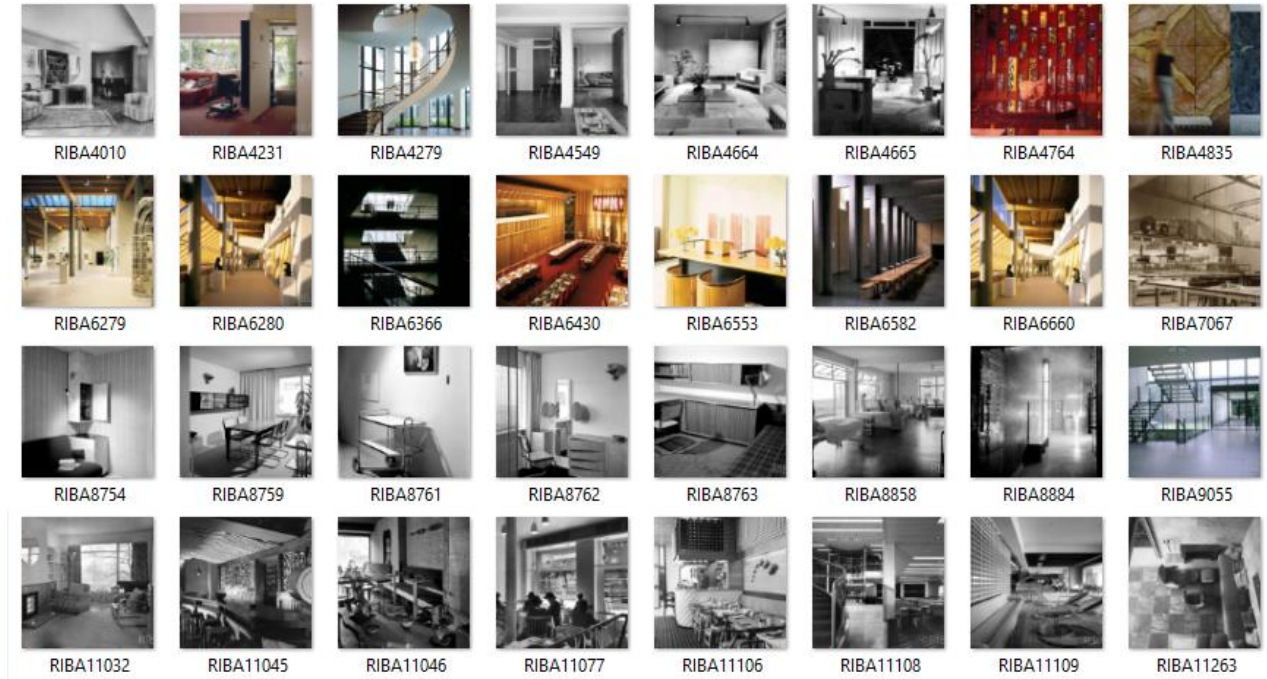
Çalışmanın ana aracı olarak Google ColabPro kod yazım ortamı kullanılmış ve algoritmanın çalıştırılması süresince Google'ın sunduğu Grafik İşleme Ünitesi (GPU) hizmetinden yararlanılmıştır. X alanına karşılık gelen modern iç mekânların .jpeg formatında elde edilen imajları kullanılırken, Y alanı için oluşturulacak veri setlerinin oluşturulmasında "Kawping" isimli çevrimiçi video bölümleri aracı kullanılarak öncelikle seçilen bilimkurgu film videoları istenilen fragmanlara ayrılmıştır. Film fragmanların seçimindeki en öncül kriter, çalışmada kullanılacak bilim kurgu filmi fragmanlarının iç mekân görüntüleri içermesidir. Böylece Cycle-GAN algoritmasının çıktılarındaki döngüsel tutarlılık kayıplarının minimize edilmesi amaçlanmıştır. Video fragmanlarının hazırlanmasının ardından söz konusu .avi uzantılı dosyalar VLC Player kullanılarak .jpeg formatına dönüştürülmüştür. Sırasıyla Blade Runner, Metropolis, 2001: A Space Odyssey ve Star Wars Serisi olmak üzere 4 film için ayrı ayrı olarak .jpeg formatında hazırlanan

Y alanına ait veri setlerine ait imajlar aynı boyuta getirilmek üzere Adobe Bridge aracıyla işlenerek 128x128x3 ve 256x256x3 piksel olarak tekrar boyutlandırılmıştır. Ardından iki farklı boyut olarak hazırlanmış, X ve Y alanlarına ait veri setleri kullanılarak Cycle-GAN algoritması çalıştırılmıştır. Veri setlerinin hazırlanması aşamasında çalışmaların büyük çoğunluğunda Pareto prensibi (80-20 ilkesi) kullanılmaktadır (Url-1). Bu ilkeden yola çıkarak her bir veri setinin yaklaşık olarak %20'si test, %80'i algoritmanın eğitilmesi (training) için ayrılmıştır.

#### 4. MODERN İÇ MEKÂNLARIN CYCLE-GAN İLE TEKRAR ÜRETİLMESİ

Çalışmada Cycle-GAN algoritması kullanılarak RIBA'nın açık kaynak olarak paylaştığı modern iç mekân görüntülerinin (Url-2), sinema tarihinin kültürel bilim kurgu filmlerine ait iç mekânları için tekrar üretimi yapılmıştır. Modern iç mekân görüntüleri X alanı olarak algoritmaya tanımlanmıştır (Şekil 7).

**Şekil 7:** RIBA'nın modern iç mekân görüntülerinden oluşturulmuş veri setinden örnekler (Url-2).



X alanı için hazırlanmış olan veri setinde 795 adet görüntü algoritmanın eğitilmesi, 192 görüntü ise test için kullanılmıştır. Şekil 7'de görüldüğü üzere RIBA dijital kütüphanesinden elde edilmiş modern iç mekân görüntüleri farklı tipoloji ve morfolojilere ait iç mekânların

fotoğraflarından oluşmaktadır (Url-2). Mekânların tipolojik ve morfolojik farklılıklarının yanı sıra, mekânların fotoğraflanma açıları da belirgin farklar bulunmaktadır.

Y alanına ait veri setleri ise bilim kurgu sinemasının kült filmlerinden Blade Runner, Metropolis, 2001: A Space Odyssey, Star Wars Serisi kullanılarak üretilmiştir. Her film için hazırlanmış olan test ve eğitim veri setlerinde bulunan veri sayıları **Tablo 1**'de görülmektedir.

**Tablo 1:** Y alanı (domain) için hazırlanmış test ve eğitim verileri

Veri Setleri	Blade Runner	2001: A Space Odyssey	Metropolis	Star Wars Serisi
Eğitim veri sayısı	624	1221	1249	450
Test veri sayısı	156	404	325	125

Çalışmada Blade Runner ve Star Wars serisi için ortalama 500 görüntü eğitim için kullanılırken, yaklaşık 120-150 kadar görüntü ise algoritmanın testi için kullanılmıştır. Bununla birlikte 2001: A Space Odyssey ve Metropolis filmleri için daha fazla sayıda veri, eğitim ve test için ayrılmıştır. Söz konusu veri sayısı değişimi ile algoritmanın çalıştırılmasıyla birlikte veri seti hacminin algoritmanın çıktı niteliğine olan etkisinin belirlenmesi de amaçlanmıştır. **Şekil 8**'de Y alanına ait veri seti örneklerinde görüldüğü üzere, bilimkurgu filmlerinden elde edilmiş olan iç mekân görüntüleri X alanında olduğu gibi farklı morfoloji ve tipolojide iç mekânların görüntülerini içermektedir. Buna ek olarak filmlerde iç mekân görüntülerinde sadece mekân değil kullanıcılar da görüntülerde bulunabilmektedir. Ayrıca, görüntülerdeki aydınlık düzeyleri de birbirinden oldukça farklılık göstermesi mekânsal algı da değişimler yaratmaktadır.

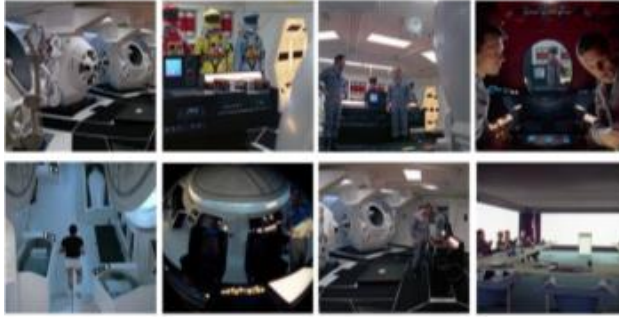
Blade Runner Veri Seti



Metropolis Veri Seti



2001: A Space Odyssey Veri Seti



StarWars Serisi Veri Seti



Tüm veri setlerindeki görüntüler 128x128x3 ve 256x256x3 piksel olarak iki farklı boyutta hazırlanmış, hedeflenen sonuçları elde etmek adına algoritmanın ana fonksiyonları farklı iterasyon sayılarına karşılık gelen dönem (epoch) sayıları ile çalıştırılmıştır. Öncelikle kontrol grubu çalışması olarak tanımlanabilecek bir ön çalışma yapılmış, ana çalışma için optimum görüntü boyutu ve dönem sayısı belirlenmiştir. Ardından ana çalışmada aynı dönem sayısı ve görüntü boyutu ile algoritma her bir film için tekrar çalıştırılarak, sonuçlar değerlendirilmiştir.

**Şekil 8:** Y alanı (domain) için hazırlanmış test ve eğitim verilerinden örnekler

#### 4.1 Kontrol Grubu Çalışması

Kontrol grubu çalışmasında Metropolis filminden üretilmiş olan veri seti kullanılarak Cycle-GAN algoritması 1000 ve 10000 dönem sayılarında çalıştırılmış, 128x128x3 piksel boyutundaki görüntülerden elde edilmiş olan sonuçlar incelenmiştir (Şekil 9).



**Şekil 9:** Metropolis 128x128x3 piksel, yukarıdan aşağıya 1000 ve 10.000 dönem için çıktılar.

**Şekil 9**'da görüldüğü üzere, 128x128x3 piksel boyutundaki verilerin işlenmesinde dönem sayısının artırılması çıktılarının niteliğinde belirgin düzeyde azalmaya neden olmuştur. Her iki deneme için dögüsel tutarlılık kaybının hesaplanmasında **Denklem 4** ve **5**'deki bağıntılara ek

olarak Ortalama Mutlak Hata Fonksiyonu (OMH)'den yararlanılmıştır (Denklemler 6).

$$OMH = \frac{1}{n} \sum_{j=1}^n |x_j - y_j| \quad (6)$$

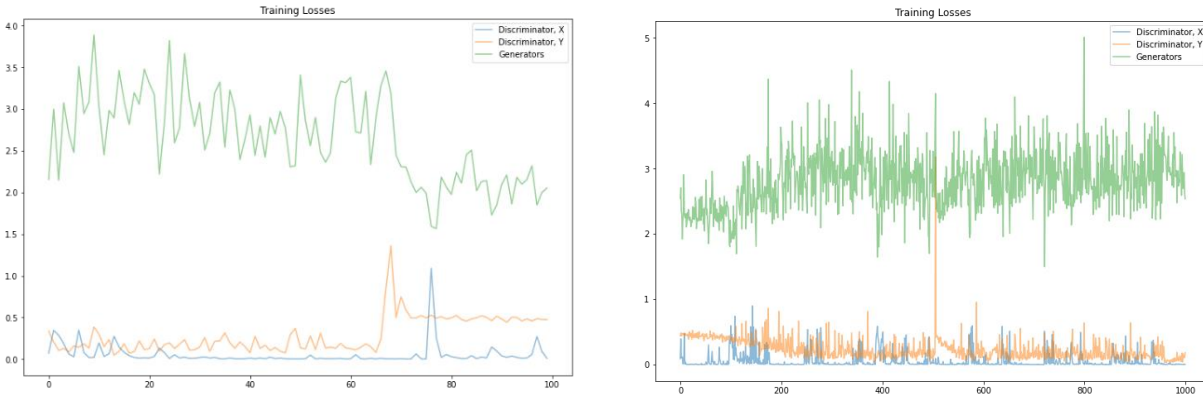
Denklemler 6'te  $x_j$  üretilen (sahte) imajların vektörünü,  $y_j$  ise gerçek imajların vektörünü temsil ederken;  $n$  ise veri setindeki toplam veri sayısına karşılık gelmektedir (Qi ve diğ., 2020). Yukarıda belirtilen hata fonksiyonu kullanılarak, algoritmanın dögüsel tutarlılık kayıpları belirlenmiştir (Tablo 2).

Dönem sayısı	Dögüsel tutarlılık kaybı $D_x$	Dögüsel tutarlılık kaybı $D_y$	Toplam Dögüsel Tutarlılık Kaybı
1000	0.0093	0.4739	2.0535
10.000	0.0047	0.1843	2.5300

Tablo 2: Metropolis filmi için dögüsel tutarlılık kayıpları

Tablo 2'de görüldüğü üzere, dönem sayısını artması toplam dögüsel tutarlılık kaybını arttırmaktadır. Ancak, X ve Y alanlarının ayırt edicileri ayrı ayrı olarak değerlendirildiğinde, dönem sayısının yükselmesiyle ayırt edicilerin dögüsel tutarlılık değerlerini kayıplarını azaltmıştır. Çalışmada X alanına ait verilerden Y alanına benzer verilerin elde edilmesinde odaklanıldığından, araştırmada  $D_y$  ayırt edicisinin tutarlılık kayıpları ve toplam tutarlılık kayıplarına göz önüne alınmış ve ileriki değerlendirmeler bu parametrelere göre yapılmıştır. Söz konusu iki denemenin de dögüsel tutarlılık kayıplarının görüldüğü grafiklere bakıldığında ise;  $D_y$  için 1000 dönem ile yapılan denemede en yüksek tutarlılığın 600-800 dönem aralığında, 10.000 dönemlik deneme için ise 4000-6000 aralığında elde edildiği görülmektedir (Şekil 10).

Şekil 10: Soldan sağa sırasıyla 1000 dönem ve 10.000 dönem için dögüsel tutarlılık kayıpları.



Söz konusu sonuçların elde edilmesinden sonra görüntü boyutlarının değişiminin tutarlılık kayıplarına etkisini belirlemek ve çıktı niteliğinin

değişimini görebilmek adına, 1000 dönem sayısı ve 256x256x3 piksel boyutundaki Metropolis veri setiyle algoritma tekrar çalıştırılmıştır. **Şekil 11**'de, verilerin piksel değerlerinin büyümesiyle görüntüdeki gürültülülüğün (noise) 128x128x3 piksel boyutundaki veri seti sonuçlarına göre oldukça azaldığı görülmektedir. Ayrıca 128x128x3 piksel ile elde edilmeyen renk çeşitliliği, 256x256x3 piksel ile elde edilmiştir. Mekânların morfolojik olarak değişimi gözlemlenmezken, mekândaki aydınlık seviyesi değişerek, gerçek görüntülerde daha açık renk tonlarına sahip yüzeyler, üretilmiş görüntülerde en koyu yüzeyler haline gelmiştir.

**Şekil 11:** Metropolis 256x256x3 piksel, 1000 dönem sayısı



**Şekil 11**'deki çıktılar analiz edildiğinde ise, toplam dögüsel tutarlılık kaybı 1000 dönem sayısı için 3.1797 olarak saptanırken;  $D_Y$  için 0.3504,  $D_X$  için 0.0504 olarak belirlenmiştir. Söz konusu değerler 128x128x3 piksel ve 1000 dönem ile karşılaştırıldığında, dögüsel tutarlılık kaybı açısından düşük çözünürlüklü görüntülerin daha tutarlı sonuçlar verdiği görülmektedir. Kontrol grubu analizleri sonucunda, 128x128x3 piksel boyutu ve 6000 dönem sayısı ile çalışmanın ilerletilmesine karar verilmiştir.

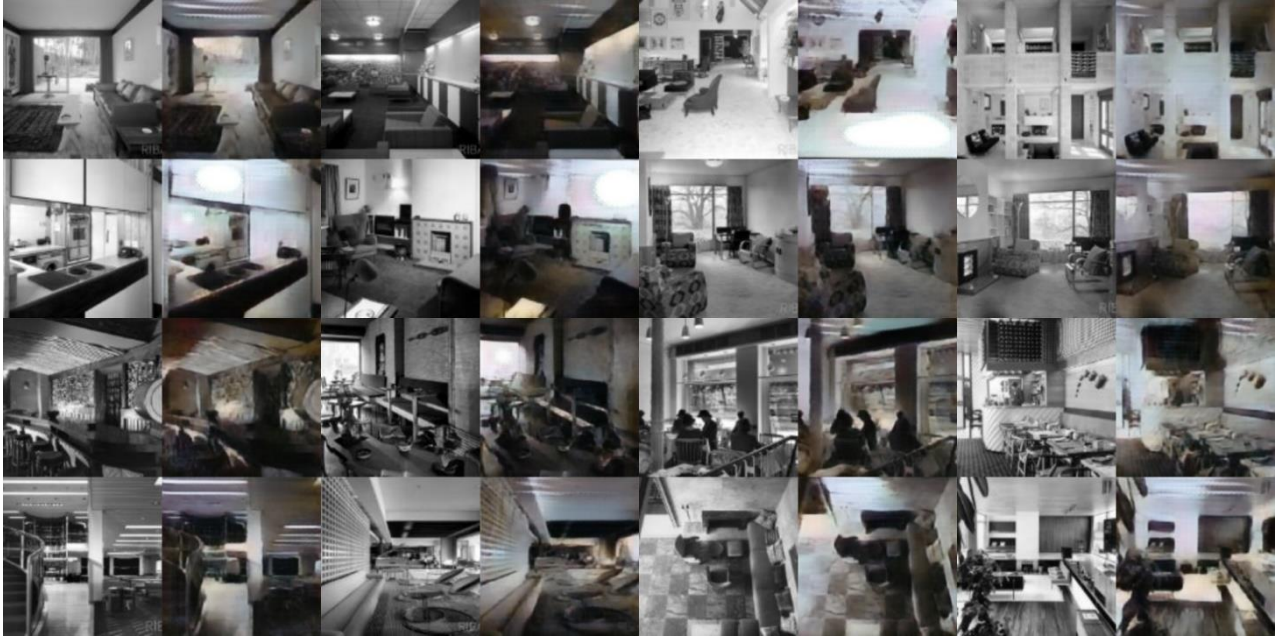
#### 4.2 Bilim Kurgu Ortamları için Modern İç Mekân Üretimi

Çalışmanın ana bölümünde Blade Runner, Metropolis, 2001: A Space Odyssey ve Star Wars Serisi filmleri görüntüleri ile hazırlanmış olan



128x128x3 boyutundaki veriler kullanılarak Cycle-GAN algoritması 6000 dönem sayısı ile çalıştırılmıştır (Şekil 12).





X

Y

X

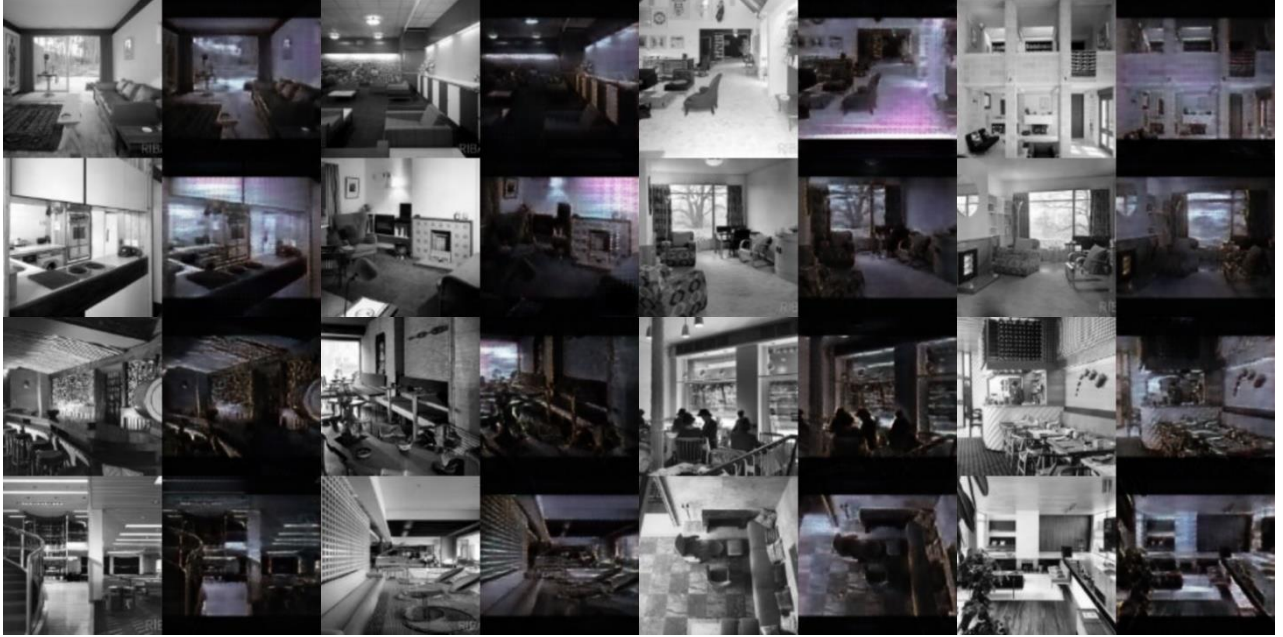
Y

X

Y

X

Y



**Şekil 12:** Yukarıdan aşağıya; Blade Runner, Metropolis, 2001: A Space Odyssey, Star Wars Serisi. 128x128x3 piksel, 6000 dönem sayısı.

**Şekil 12'**de görüldüğü üzere, X alanına ait veri setindeki modern iç mekân görüntüleri, sırasıyla Blade Runner, Metropolis, 2001: Space Odyssey ve Star Wars serisine ait verilerin bulunduğu Y alanına Cycle-GAN algoritması kullanılarak dönüştürülmüştür. 6000 dönem sayısı ve 128x128x3 piksel boyundaki görüntüler ile elde edilmiş sonuçların dögüsel tutarlılık kayıpları da **Tablo 3'**te görülmektedir.

Bilim Kurgu Filmleri	Döngüsel tutarlılık kaybı $D_x$	Döngüsel tutarlılık kaybı $D_y$	Toplam Döngüsel Tutarlılık Kaybı
Blade Runner	0.0037	0.0397	2.8882
Metropolis	0.0455	0.0476	2.9085
2001: Space Odyssey	0.0840	0.1085	3.1845
StarWars	0.0267	0.2658	2.2267

**Tablo 3:** 128x128x3 piksel, 6000 dönem sayısı için döngüsel tutarlılık kayıpları

Döngüsel tutarlılık kayıpları incelendiğinde, en yüksek değer 3.1845 ile 2001: A Space Odyssey Filminde görülürken, en düşük değer ise 2.2267 ile Star Wars Serisi için belirlenmiştir. Bölüm 4.1'deki ön çalışmalardaki değerler bağlamında Metropolis için toplam kayıplar karşılaştırıldığında, 6000 dönem sayısı ile elde edilen kayıp değerinin 1000 ve 10.000 dönem sayısında elde edilen kayıp değerlerinden daha fazla olduğu görülmektedir.

## 5. SONUÇ

Yapılan çalışma Cycle-GAN algoritmasıyla gözetimsiz derin öğrenme araştırmaları yaparak modern iç mekân görüntülerinden bilim kurgu ortamlarına ait yeni mekânlar üretmeyi amaçlamakta ve üretilen mekânların tipolojik ve morfolojik özelliklerinde gerçekleşen değişimlerin analizini kapsamaktadır. Mimari analizlerin sağlıklı olarak yapılabilmesi amacıyla, araştırmada Metropolis filmi verileri kullanılarak kontrol grubu çalışması yapılmış, ana çalışma için gereken optimum görüntü boyutu ve dönem sayısının belirlenmesi hedeflenmiştir. Böylece Cycle-GAN algoritmasından optimum düzeyde verim alınması planlanmıştır. Kontrol grubu deneyi sonuçlarına göre 128x128x3 piksel boyutuna sahip görüntülerin döngüsel tutarlılık değerlerinin, 256x256x3 piksel boyutuna göre daha az olduğu belirlenmiştir. Bu bağlamda, algoritmanın verimi ve piksel boyutu arasında bu çalışma çerçevesinde ters orantı gözlemlenmiştir. Bu durum, gözetimsiz derin öğrenme yönteminin kullanılması ve görüntülerde bulunan mimari bileşenlerin birbirinden oldukça farklı olması nedeniyle gerçekleşebileceği yorumu yapılabilir. Ayrıca, en yüksek verimin 4000-6000 dönem sayısı aralığında alındığı ön kontrol çalışmasında belirlenmesine ve ana çalışmanın da bu bilgiye dayanarak 6000 dönem sayısı ile yapılmasına rağmen, toplam kayıp değerleri özellikle

Metropolis örneği için öngörülen değer üzerindedir. Bu sonucun da algoritmanın gözetimsiz olarak eğitilmesi nedeniyle olabileceği öngörülmektedir. Bu bağlamda ileriki çalışmalarda farklı morfoloji ve tipolojideki mekânlara ait mimari bileşenlerin etiketlenerek gözetimli derin öğrenme ile eğitilmesi, sonuç ürünlerdeki mimari bileşenlerin biçimsel olarak da değişmesini sağlamak için uygun bir yöntem olarak değerlendirilebilir. Ancak, bu çalışmada, biçimsel olarak değişimden ziyade mekânın ait olduğu atmosfer ile olan ilişkisi açısından oldukça belirgin mimari değişim ve dönüşümler elde edilmiştir. Bu bağlamda çalışma, özellikle kavramsal tasarım süreçleri, mimari tasarımda biliş ve mimarlık kuramı çalışmaları için oldukça ufuk açıcı perspektifler sağlamaktadır. Araştırmanın ilgili çalışma alanlarına katkısını özetlemek adına, sonuç ürünlerin mekânsal potansiyellerinin değişimi ve dönüşümü, seçilmiş olan her bir film için sırasıyla incelenmiş ve yorumlanmıştır.

Çalışmanın ana ürünü olan iç mekân görüntüleri (**Şekil 12**), mimari mekân üretimi bağlamında analiz edildiğinde, görüntülerde dramatik seviyede tipolojik ve morfolojik değişimler gözlemlenmediği, ancak mekânların aydınlık seviyesi, ışık rengi ve oluşan gölgeler bağlamında oldukça belirgin dönüşümlerin olduğu görülmektedir.

Blade Runner ve Metropolis veri setleriyle yapılan çalışmalarda mekânların sahip olduğu renklerin birbirine oldukça benzer olduğu görülmekteyken, bu yakınlığın Star Wars Serisi ile 2001: A Space Odyssey arasında da bulunduğu söylenebilir. Ayrıca, Metropolis çıktılarında açık tonlardaki yüzeylerin daha koyu, koyu tonlara sahip yüzeylerin ise daha açık renk yüzeyler olarak yer değiştirmiş olduğu görülmektedir. Bu bağlamda Metropolis veri seti Cycle-GAN'a ışık kaynaklarının yer değiştirdiği bir iç mekân yaratma olanağı sağlamıştır. Böylece görüntülerdeki mimari bileşenlerin iç-dış ilişkisi, ön ve arkada olma durumu değişmiş ve mekân, gözlemciye Blade Runner çıktılarına göre çok daha farklı varyasyonlarda mekânsal deneyimler sunabilmiştir. Örneğin, orijinal görüntüde zeminde bulunan daha aydınlık bir bölüm gözlemciye tavanda bulunabilecek bir ışıklık öğesinin varlığını sorgulatabilir. Bu bağlamda, Metropolis, iç mekânı saran yapı kabuğunun morfolojik değişimleri ile ilgili olasılıkları da arttıran, tasarımcıya mekânın var olduğu atmosfer ile ilgili soru sordurtan mimari ürünlerin ortaya çıkmasını sağlamıştır.

2001: A Space Odyssey ise özellikle iç mekânda, orijinal görüntülerde olmayan yapay ışıklandırma öğelerinin arttığı çıktılar sunmuştur. Ayrıca algoritma ağaç, pencereden görünen gökyüzü, desenli yüzeyler gibi daha karmaşık yapıları pürüzsüzleştirerek, daha sade ve yalın iç mekânlar üretilmesini sağlamıştır. Bu bağlamda filmin genel atmosferine hâkim olan steril mekân algısı, yeniden üretilmiş modern iç mekânlarında etkin bir biçimde hissedilmektedir.

Star Wars Serisi çıktılarına bakıldığında ise; orijinal iç mekân görüntülerinde var olan gölge tonlarının oldukça koyulaştığı ve aydınlık bölgelerdeki ışık kaynağı renklerinin farklı tonlara sahip neon efektlere dönüştüğü, başka bir deyişle mekân görüntüsünün saturasyon (doyma, rengin saflık derecesi) değerlerinde dramatik değişimler gerçekleştiği görülmektedir.

Yukarıdaki analizler bağlamında, mekânların kendilerini çevreleyen ortama ait olma durumu, söz konusu çalışmada ışık, gölge, yüzey dokusu ve renk üzerinden analiz edilmiş ve gözetimsiz derin öğrenme ile gerçekleştirilen üretimlerin bu noktada oldukça yararlı olduğu görülmüştür. Çalışmanın ileriki adımlarında ise mekânlarda bulunan mobilyaların ve/veya yapı elemanlarının biçimsel değişimlerine odaklanılarak, araştırmanın uluslararası literatüre katkı sağlaması hedeflenmektedir.

## Referanslar

Abdulkader A., Lakshmiratan A., and Zhang J. (2016, Jun 1). *Introducing DeepText: Facebook's Text Understanding Engine* [Blog]. Facebook Engineering. Erişim adresi: <https://tinyurl.com/jj359dv>, erişim tarihi 26.08.2020.

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.

Borysiuk, Z. & Konieczny, M. & Kręcis, K. & Pakosz, P. (2018). Application of sEMG and Posturography as Tools in the Analysis of Biosignals of Aging Process of Subjects in the Post-production Age. In Hunek, W., P. & Paszkiel, S. (Eds), *Proceedings of the 3<sup>rd</sup> International Scientific Conference on Brain-Computer Interfaces, BCI 2018* (pp. 23-29). Cham: Springer International Publishing. <https://doi.org/10.1007/978-3-319-75025-5>

Chailou, S. (2019). *AI & Architecture Towards A New Approach* [Thesis Project]. Harvard University Graduate School of Design. Erişim Adresi: <https://towardsdatascience.com/ai-architecture-f9d78c6958e0>, erişim tarihi: 26.08.2020.

Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S., & Choo, J. (2018). Stargan: Unified Generative Adversarial Networks for Multi-Domain Image-To-Image Translation. In *Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition* (pp. 8789-8797).

Duda, R. O., Hart, P. E. & Stork, D. G. (2001). *Pattern Classification* (2<sup>nd</sup> ed.) (pp. 16-17). New York: John Wiley & Sons.

Eiben A.E., Raué P.E., Ruttkay Z. (1994). Genetic algorithms with multi-parent recombination. In Davidor Y., Schwefel HP., Männer R. (Eds), *Parallel Problem Solving from Nature — PPSN III.*. Berlin: Springer. [https://doi.org/10.1007/3-540-58484-6\\_252](https://doi.org/10.1007/3-540-58484-6_252)

Frazer, J. (1995). *An Evolutionary Architecture* (pp. 71-85). London: Architectural Association.

Gero, J. S. (1996). Artificial Intelligence in Computer-Aided Design: Progress and Prognosis. *Computer Aided Design* 27(3), 153-154. Great Britain: Elsevier.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, 2672-2680.

Hitaj, B., Ateniese, G., & Perez-Cruz, F. (2017). Deep Models Under The GAN: Information Leakage From Collaborative Deep Learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security* (pp. 603-618).

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. In *Proceedings of The National Academy Of Sciences* 79(8), 2554-2558.

Huang, X., Liu, M. Y., Belongie, S., & Kautz, J. (2018). Multimodal Unsupervised Image-To-Image Translation. In Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds), *Proceedings of the European Conference on Computer Vision (ECCV)*, 172-189. Cham: Springer. <https://doi.org/10.1007/978-3-030-01246-5>.

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-To-Image Translation With Conditional Adversarial Networks. In *Proceedings of the IEEE Conference On Computer Vision And Pattern Recognition* (pp. 1125-1134).

Mitchell, T. M. (1997). *Machine Learning* (pp. 269-270). Singapore: McGraw-Hill Science/ Engineering/ Math.

Nilsson, N. T. (1980). *Machine Learning Principles of Artificial Intelligence* (pp. 1-3). California: Morgan Kaufmann Publishers.

Qi, J. & Du, J. & Siniscalchi, M. & Ma, X. & Lee, C. (2020). On Mean Absolute Error for Deep Neural Network Based Vector-to-Vector Regression. In *IEEE Signal Processing Letters* 27 (pp. 1485-1498). doi: 10.1109/LSP.2020.3016837

Silva, T. (2018, January 7). *An intuitive Introduction to Generative Adversarial Networks (Gans)* [Blog]. FreeCodeCamp. Erişim Adresi: <https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/>, erişim adresi: 26.08.2020.

Simon, Herbert A. (1973). The Structure of Ill-Defined Problems. *Artificial Intelligence* 4 (3-4), 181-201. [https://doi.org/10.1016/0004-3702\(73\)90011-8](https://doi.org/10.1016/0004-3702(73)90011-8).

Tamke, M., Nicholas, P., & Zwierzycki, M. (2018). Machine Learning for Architectural Design: Practices and Infrastructure. *International Journal of Architectural Computing*, 16(2), 123-143.

Zhu, J. Y., Park, T., Isola, P., & Efros, A. A. (2017). Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)* (pp. 2223-2232).

Url-1 <<https://towardsdatascience.com/finally-why-we-use-an-80-20-split-for-training-and-test-data-plus-an-alternative-method-oh-yes-edc77e96295d>>, erişim tarihi 26.08.2020.

Url-2 <<https://www.architecture.com/imagelibrary/ribapix.html?keywords=modern%20interiors>>, erişim tarihi 26.08.2020.

# Map Generation & Manipulation with Generative Adversarial Networks

Ozan Balci<sup>1</sup>, Şemsi Barış Terzi<sup>2</sup>, Özgün Balaban<sup>3</sup>

0000-0002-9265-2262<sup>1</sup>, 0000-0002-3400-7839<sup>2</sup>, 0000-0002-7270-2058<sup>3</sup>

<sup>1</sup>Istanbul Technical University, Faculty of Architecture, Department of Architecture, Istanbul, Turkey

<sup>2</sup>Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

<sup>3</sup>Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Hollanda

Artificial intelligence makes the Big Data processable as the spread of the Internet. Machine learning, which is a type of artificial intelligence, has made it easier to classify, interpret and process data. On the other hand, the development of graphics processing units has facilitated and accelerated the work done by providing the necessary infrastructure for processing data consisting of images. The development of image processing operation will enable the processing and production of architectural data sets consisting of images. This study aims to show the processability of the image set in map scale and to encourage to think and examine how the achievements in this process can be processed in different scales. In accordance with this purpose, satellite images and map images of a game are handled as data using a GAN model. The model, which is trained with satellite and game maps images contain different textures of nature, different types of transportation networks, settlements of different densities and qualities, can perform two-way transformations (game map from satellite image and satellite image from game map). The study focused on the generation of game maps from satellite images, and during the process, tests for the generation of game maps from satellite images of different locations around the world and analyzes based on these tests were conducted. As the last step of the study, to obtain the most successful results, the model inputs were optimized with the findings obtained from the analyzes and by using satellite images of these regions, the game maps of 'Istanbul Historical Peninsula' and 'Istanbul Göksu' regions were produced.

**Keywords:** Generative Adversarial Networks, Machine Learning, Map Generation

**Received:** 25.08.2020

**Accepted:** 22.09.2020

**Corresponding Author:**

baris@virtualogie.com

Balci, O., Terzi, Ş.B., Balaban, Ö. (2020). Map Generation & Manipulation with Generative Adversarial Networks. JCoDe: Journal of Computational Design, 1(3), 95-114.



# Çekişmeli Üretici Ağlar (Gan) ile Harita Üretimi ve Manipülasyonu

Ozan Balcı<sup>1</sup>, Şemsi Barış Terzi<sup>2</sup>, Özgün Balaban<sup>3</sup>

0000-0002-9265-2262<sup>1</sup>, 0000-0002-3400-7839<sup>2</sup>, 0000-0002-7270-2058<sup>3</sup>

<sup>1</sup> İstanbul Teknik Üniversitesi, Mimarlık Fakültesi, Mimarlık Bölümü, İstanbul, Türkiye

<sup>2</sup> İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

<sup>3</sup> Chair of Design Informatics, Faculty of Architecture and the Built Environment, Delft University of Technology, Delft, Hollanda

Yapay zeka, internetin yaygınlaşması ile yoğunlaşan veri birikimini işlenebilir kılmıştır. Bir yapay zeka türü olan makine öğrenmesi verilerin sınıflandırılması, anlamlandırılması ve işlenmesini kolaylaştırmıştır. Grafik işleme ünitelerinin gelişmesi ise görüntülerden oluşan verilerin işlenmesinde gerekli altyapıyı sağlayarak, yapılan çalışmaları kolaylaştırmış ve hızlandırmıştır. Görüntü işleme süreçlerinin gelişimi görüntülerden oluşan mimari veri setlerinin işlenmesini ve üretilmesini sağlayabilecektir. Bu çalışmada amaç; harita ölçeğindeki görüntü setinin işlenebilirliğinin gösterilmesidir. Bu gösterimden hareketle görüntü işlemenin mimari gösterim tekniklerinde ve farklı ölçeklerde nasıl olabileceğine dair tartışma açabilmektedir. Bu amaç doğrultusunda bir çekişmeli üretici ağ modeli kullanılarak uydu görüntüleri ile bir oyuna ait harita görüntüleri veri olarak ele alınmıştır.

Uydu ve oyun haritasında bulunan, doğaya ait farklı dokular, farklı tip ulaşım ağları, farklı yoğunlukta ve nitelikte olan yerleşim yerleri içeren görüntüler ile eğitilen model, çift yönlü (uydu görüntüsünden oyun haritası ve oyun haritasından uydu görüntüsü) olarak dönüşümler gerçekleştirebilmektedir. Çalışmada, uydu görüntülerinden oyun haritaları üretimine odaklanılmış olup, süreç boyunca dünya üzerinde farklı lokasyonlara ait uydu görüntülerinden oyun haritaları üretimine yönelik testler ve bu testleri baz alan analizler yapılmıştır. Çalışmanın son aşaması olarak analizlerden elde edilen bulgular ile model girdileri en başarılı sonuçlar alınacak şekilde optimize edilmiş ve 'İstanbul Tarihi Yarımada' ve 'İstanbul Göksu' bölgelerine ait uydu görüntüleri kullanılarak bu bölgelerin oyun haritaları üretilmiştir.

**Anahtar Kelimeler:** Çekişmeli Üretici Ağlar, Makine Öğrenmesi, Harita üretimi

**Teslim Tarihi:** 25.08.2020

**Kabul Tarihi:** 27.09.2020

**Sorumlu Yazar:**

baris@virtualogie.com

Balcı, O., Terzi, Ş.B., Balaban, Ö. (2020). Çekişmeli Üretici Ağlar (Gan) İle Harita Üretimi Ve Manipülasyonu. JCoDe: Journal of Computational Design, 1(3), 95-114.

## 1. GİRİŞ

İnternetin yaygınlaşması ile içinden çıkılmaz bir yoğunluğa ulaşan veri birikimi, günümüzde büyük veri denilen kavramı oluşturmuştur. Bu veri birikiminin karmaşıklığı, sınıflandırılma ve işleme zorluğu büyük verinin değerlendirilmesinin önünde bir engel olarak görülmüştür. Büyük verinin anlamlandırılıp işlenmesi, makine öğreniminin gelişimi ile son yıllarda mümkün hale gelmiştir. Büyük verinin bir kısmını oluşturan görüntülerin sayısal görüntülere dönüştürülmesi ve günümüzde bilgisayarların grafik işlem ünitelerinin hızlı gelişimi sayesinde makine öğrenmesi algoritmaları tarafından işlenmesinin mümkün kılınması, çalışmaların hızlanmasını ve kolaylaşmasını sağlamıştır. Bu çalışmada verilerin işlenebilmesi için Python kodlama dili ve Google Colab operatörü kullanılmıştır.

### 1.1 Görüntü İşleme

Görüntü işleme görüntülerin sayısal verilere dönüştürülüp üzerinde işlemler yapılmasını sağlayarak anlamlandırılmasını, manipule edilmesini ve yeni görüntüler elde edilmesini sağlamaktadır. Bu görüntüler makine öğrenmesi yöntemleri ile de işlenebilir. Python kodlama dilinde birçok modül ile görüntü işleme yapılabilir. 'Numpy', 'OpenCV', 'matplotlib', 'PIL' bu modüllerden bazılarıdır. Bu modüller sayesinde görüntüler, her pikselinin RGB (red – green - blue) veya HSV (hue – saturation - value) gibi renk uzaylarındaki karşılıklarından oluşan matrislere dönüştürülebilir, bu verilerden grafikler oluşturulabilir ve geri görüntülere dönüştürülebilir (Gezer, 2018).

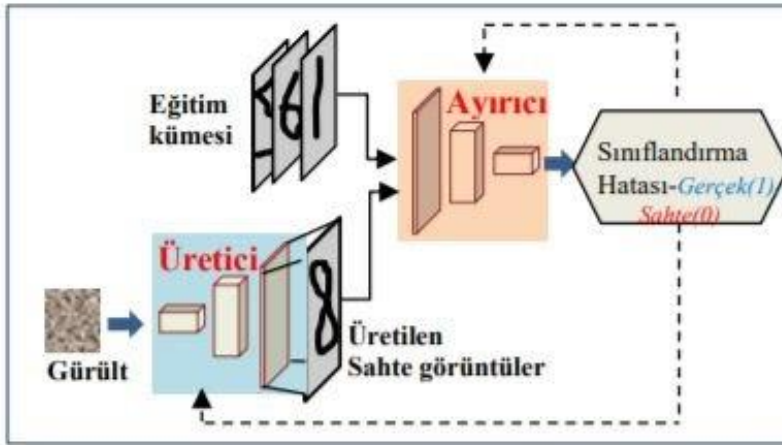
### 1.2 Derin Öğrenme, Makine Öğrenmesi Ve Yapay Zeka

Makine öğrenmesi örnekler ve verilerle öğretilen algoritmaların veriler üzerinden çıkarımlar yaparak yeni veriler üretmesi olarak tanımlanabilir (Mitchel, 1997). Yapay zeka ise 'Doğal sistemlerin yapabildiği (zekice olsun veya olmasın) her bilişsel etkinliği (gerekirse bedenleri olan) yapay sistemlere, daha da yüksek başarımlı düzeylerinde nasıl yaptırabileceğimizi inceleyen bilim dalıdır.'(Say, 2018) Dolayısıyla makine öğrenimi, bir yapay zeka yöntemi olarak ele alınabilir. Bir makine öğrenmesi sınıfı olan derin öğrenme ise çıkarımda bulunmak ve yeni veriler üretmek için öğrenme işleminde kullanılan verilerin birçok katmanının doğrusal olmayan bir şekilde kullanılmasıdır. Derin öğrenmede girdiler üzerindeki birden çok

katmanın öğrenilmesi ve bu katmanlardan oluşturulan hiyerarşik bir düzen söz konusudur (Şeker ve diğ., 2017).

### 1.3 Çekişmeli Üretici Ağlar Ve Türleri

Öğrenme için kullanılan girdilerin karmaşık yapılarını öğrenen derin öğrenme teknikleri son yıllarda büyük bir ilerleme katetmiştir. (Wu ve diğ., 2017) İlan Goodfellow ve ekibi tarafından 2014 yılında üretilen görüntü işleme yeteneğine sahip ilk model olan çekişmeli üretici ağlar (generative adversarial network, GAN), derin öğrenmeye görüntü işleme konusunda büyük yetenek kazandırmıştır. (Wason, 2018) (Goodfellow ve diğ., 2014) Çekişmeli üretici ağlarda üretici (generative) ve ayırıcı (discriminator) olarak iki algoritma çalışır (**Şekil 1**). Üretici öğrenim verilerinden aldığı verilere benzer veriler üretmeye çalışırken ayırıcı bu verilerin gerçek veya sahte olduğunu anlamaya çalışır. Üreticinin üretmiş olduğu veriler ayırıcı tarafından değerlendirilir ve ayırıcının üreticiye verdiği puanlar doğrultusunda üretilen yeni veriler birçok tekrar (epoch) sonrasında gerçek verilere benzemeye başlar. Bu modelin önerilmesinden sonra birçok GAN modeli oluşturulmaya başlanmıştır. Üreticiye verilen bir koşulla üretim yapabilen cGAN, veride havuzlamanın kullanılmadığı DCGAN, üreticinin oluşturduğu gürültüye gizli kod veya ek bilgi eklenen InfoGAN çekişmeli üretici ağ modellerinden bazılarıdır. Ayrıca bu makalede daha detaylı bahsedilecek pix2pix ve CycleGAN da çekişmeli üretici ağ modelleridir.

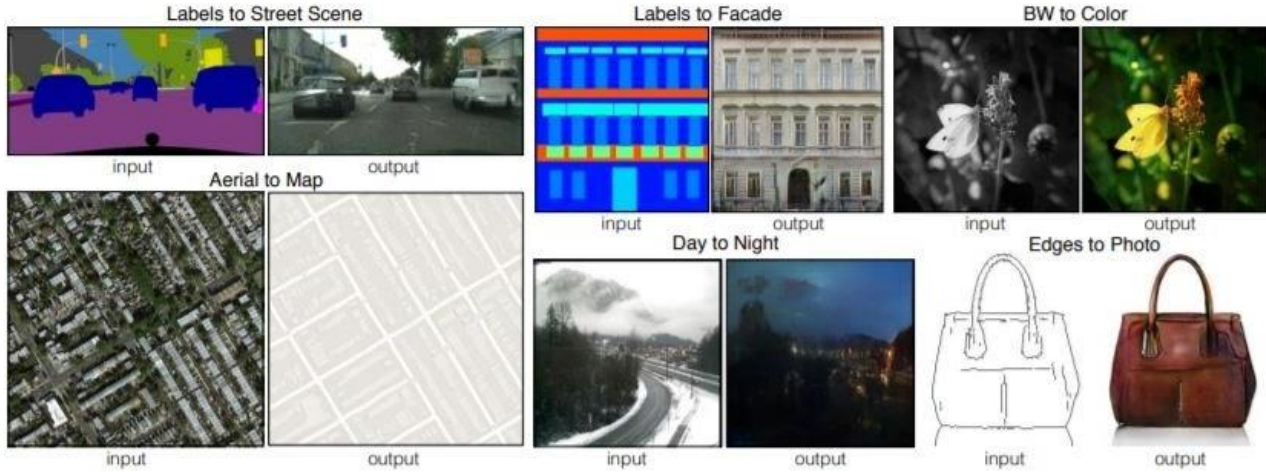


**Şekil 1:** Çekişmeli üretici ağ çalışma prensibi (Çelik & Talu, 2019)

## 2. ÇEKİŞMELİ ÜRETİCİ AĞLAR VE ÇÖZÜMLEME SÜRECİ

Görüntü işleme ve çözümlemede yapay zeka kullanımında çığır açan çekişmeli üretici ağlar girdi olarak üretilmek istenen çıktılara benzer görüntülere ihtiyaç duymaktadır. GAN modelleri arasında bulunan pix2pix ve CycleGAN, Berkeley Yapay Zeka Araştırma (BAIR) Laboratuvarından bir ekibin ürettiği modellerdir. Pix2pix görüntüden görüntüye dönüştürme modelidir (**Şekil 2**). Görüntülerdeki pikselleri birebir eşleştirerek çalışmaktadır. Bu modelde öğrenme için görüntüler sınıflandırılarak birbirini karşılayan -aynı görselin siyah beyazı, bir ürünün çizimi ve fotoğrafı veya aynı noktadan çekilmiş bir fotoğrafın gecesı gündüzü- veri kümeleri oluşturulmaktadır. Bu veri setleri ile eğitilen model yeni görüntüler üretirken bir koşula ihtiyaç duymaktadır. Bu model ile bir gündüz görselinin gecesı, bir çizimin veya boyamanın gerçekçi görüntüsü oluşturulabilmektedir (Isola ve diğ., 2018).

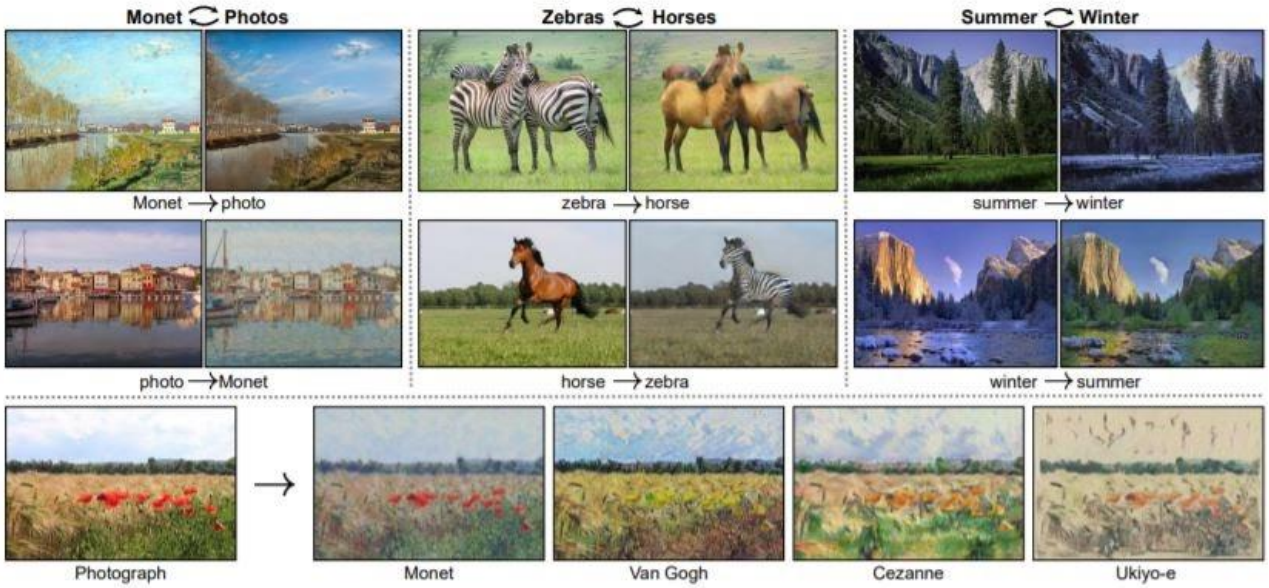
**Şekil 2:** Pix2pix örnek uygulamaları (Isola ve diğ., 2018)



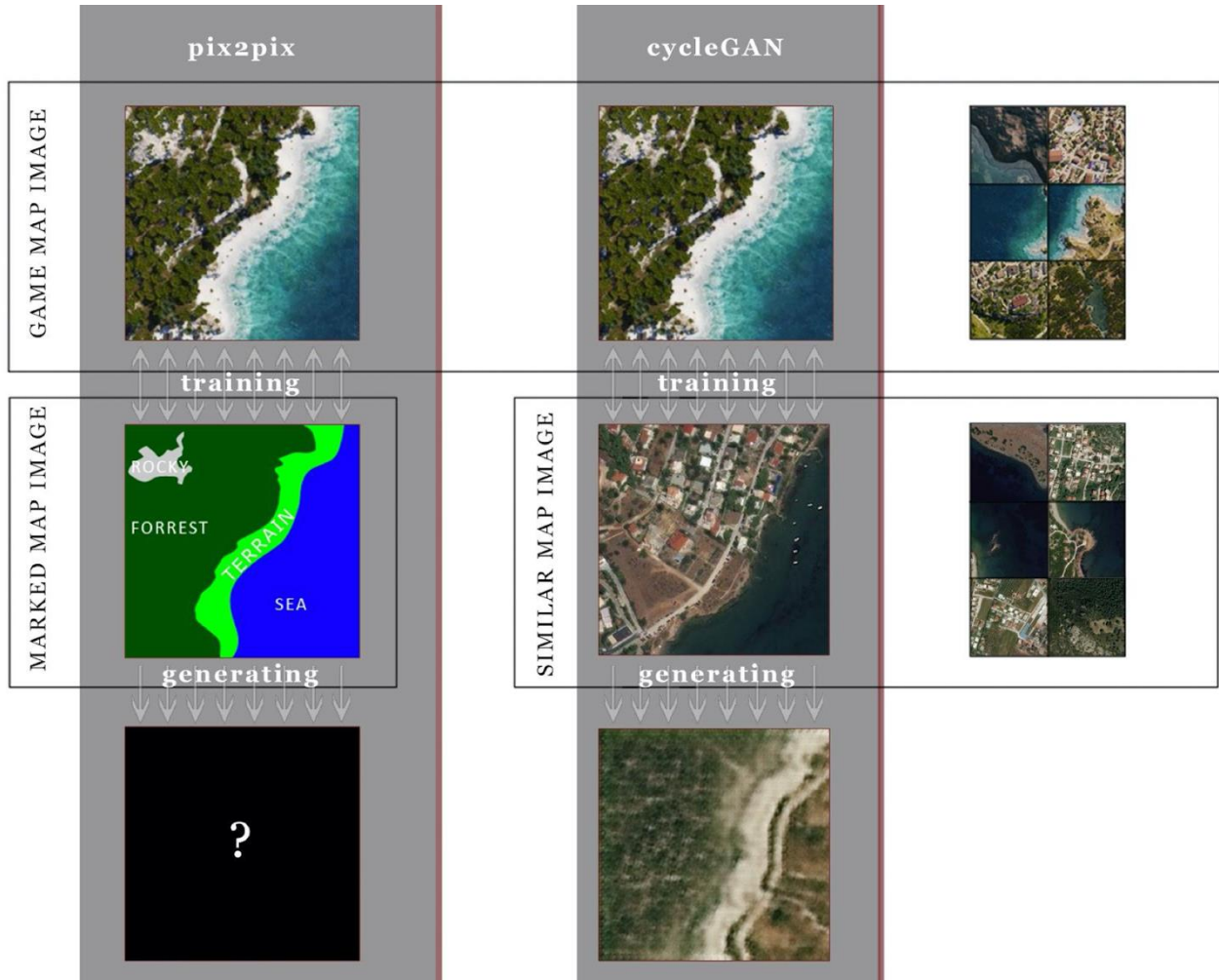
CycleGAN ise pikselleri birebir eşleşen veri setlerine değil sadece benzer veri setlerine ihtiyaç duyan bir GAN modelidir (**Şekil 3**). CycleGAN'ın anlaşılması için verilebilecek en temel örnek zebra – at (zebra - horse) çalışmasıdır. Zebra ve at görüntüleri ile eğitilen algoritma test veri kümelerindeki at görüntülerini zebra, zebra görüntülerini ise ata dönüştürebilmektedir. CycleGAN, pix2pix modelinin aksine çift taraflı çalışabilmektedir. Böylece tersine uygulanan işlemlerde algoritmanın yeniden ve başka verilerle eğitilmesi gerekmemektedir.

Algoritma iç içe iki GAN modelini kullanarak görüntüyü önce ulaşılmak istenen görüntüye sonra ise tekrar eski haline dönüştürmektedir. Dönüştürülen ulaşılmak istenen görüntünün, üreticinin ürettiği görüntüye ne kadar uyum sağladığını gösteren ve asıl haline ne kadar benzediğini gösteren iki farklı kayıp oluşturulmaktadır. Daha sonra ayırıcının etkisi ile bu kayıplar yapılan tekrarlar sonucu azaltılarak görüntünün dönüşümü sağlanmaktadır. Fotoğrafları ünlü ressamların eserlerine dönüştüren veya tam tersini; yaz fotoğrafını kış fotoğrafına dönüştüren veya tam tersini yapan algoritmalar da CycleGAN modellerine örnek olarak verilebilir (Isola ve diğ., 2018).

**Şekil 3:** CycleGAN örnek uygulamaları (Isola ve diğ., 2018)



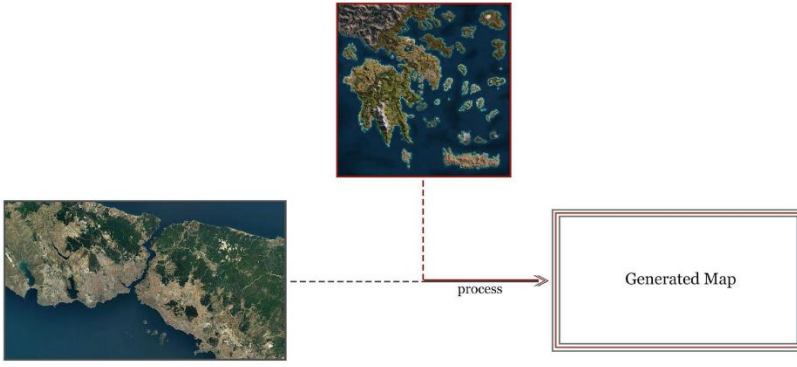
Bu çalışmada kullanılması hedeflenen GAN modeli belirlenirken bu iki model değerlendirilmiştir. Pix2pix modelinin eşlenmiş veriye ihtiyaç duyması, eğitimde kullanılacak verilerin önceden işlenmesi gerekliliğini doğurmuştur. Eğitim ve üretilmesi istenen test verilerinin işlenmesinin zorluğu ve modelin eğitilmesinde eşlenmiş veriye ihtiyaç duyulmaması nedeniyle CycleGAN kullanılması kararlaştırılmıştır (Şekil 4).



### 3. İŞLEM SÜRECİ

**Şekil 4:** Çekişmeli üretici ağlardan cycleGAN ve pix2pix karşılaştırma diyagramı

Çalışmada, belirlenmiş bir ölçekte elde edilen uydu görüntülerini kullanarak oyun haritaları üretilmesi hedeflenmiştir (**Şekil 5**). Bir diğer deyişle çalışmanın amacı; dünya üzerinde bulunan bir lokasyonun haritasının, seçilen oyuna ait harita stilinde yeniden üretilmesidir. Oyun olarak Ubisoft tarafından geliştirilen 'Assassin's Creed: Odyssey' tercih edilmiş ve bahsi geçen oyunun 'Greece' isimli haritası kullanılmıştır.



Şekil 5: Harita üretim diyagramı

Çalışma, bir CycleGAN modeli olan 'Summer to Winter Yosemite' kodu üzerinden geliştirilmiş olup, süreç esnasında veri seti olarak Google Haritalar ve Bing Haritalara ait uydu görüntüleri ve bahsi geçen oyuna ait 'Greece' haritası kullanılmıştır.

### 3.1 Çalışma Planı

Çalışmanın geliştirilmesi için tamamlanması gereken adımlar, çalışma takviminde 3 farklı iş paketi altında toplanmıştır. Bunlar:

1. Veri seti oluşturulması
2. Modelin eğitilmesi
3. Çıktıların değerlendirilmesi

Bahsi geçen iş paketleri, eş zamanlı değil, yukarıda verilen sıra ile ardışık olarak çalışılmış ve hedeflenen sonuç elde edilene kadar süreç devam etmiştir.

### 3.2 Veri Seti Oluşturulması

CycleGAN modelinin ihtiyacı olan iki ayrı veri kümesi bu çalışma özelinde 'Oyun Haritası' ve 'Uydu Haritası' olarak isimlendirilmiştir.

'Oyun Haritası' isimli veri kümesi, daha önce belirtilen oyunun haritasını oluşturan görüntüleri içermektedir. İstenen harita, çeşitli interaktif oyun haritalarını bünyesinde barındıran Mapgenie.io web sitesinde saptanmış ve veri kazıma (data scraping) için yöntem geliştirilmiştir.

İlk olarak haritanın, koordinat sistemi üzerine yerleşmiş 256\*256 piksel boyutlarında görüntülerden oluştuğu tespit edilmiştir. Bu görüntülerin bağlantı adreslerinde, koordinat sistemindeki yerlerine karşılık gelen değerlerinin örüntüsü tespit edilip, Python dilinde yazılan bir komut dizisi ile tüm görüntüler elde edilmiştir.

Örnek olarak 'https://cdn.mapgenie.io/images/tiles/ac-odyssey/greece/11/585/563.jpg' bağlantı adresinde '11' görüntünün ölçeğini, '585' görüntünün X koordinatını, '563' ise Y koordinatını temsil etmektedir.

Aynı metot, Python kodu ile veri kazıma, diğer veri kümesi olan 'Uydu Haritası' oluşturulurken de kullanılmıştır. 'Oyun Haritası' veri kümesinden farklı olarak 'Uydu Haritası' için iki farklı kaynağa ait veriler test edilmiştir. İlk çalışmalar Google Haritalar üzerinden yapılmıştır. Fakat veriler elde edilirken belirli periyotlarda işlemin Google tarafından engellenmesi, bölgeye göre uydu görüntü kalitelerinin farklılaşması, deniz görüntülerinde Google etiketlerinin bulunması gibi birtakım sebeplerden dolayı 'Uydu Haritası' veri kümesi için yeni bir kaynak arayışı gerekli olmuştur. İncelenen alternatif kaynaklar arasında Bing Haritalar'da karar kılınmış ve veri kazıma çalışmaları Bing üzerinden devam etmiştir. Bu verileri oluşturan uydu görüntüleri 2020 yılına aittir ve Bing Haritaların tanımladığı 17. yakınlaştırma seviyesindedir. Örnek olarak 'https://www.bing.com/maps ... cp=37.936172~23.950817&lvl=17&style=h&v=2&sV=2&form=S00027' bağlantı adresinde 'cp=37.936172~23.950817' koordinatları, 'lvl=17' yakınlaştırma seviyesini belirtmektedir.

İki veri kümesi ('Oyun Haritası' ve 'Uydu Haritası') için üç farklı kaynaktan 50.000'in üzerinde veri elde edilmiş ve bu veriler niteliklerine göre iki ana başlık altında sınıflandırılmıştır;

- Tip A: Deniz içermeyen (sadece kara içeren) görüntüler
- Tip B: Deniz içeren görüntüler
  - Tip B.1: Sadece deniz içeren görüntüler
  - Tip B.2: Kara ve denizi aynı anda içeren görüntüler

Yukarıda belirtildiği gibi sınıflandırılan veriler farklı kombinasyonlarda kullanılarak farklı veri setleri elde edilmiş ve bu veri setlerinin detaylarına aşağıdaki tabloda (**Tablo 1**) yer verilmiştir.



No	Versiyon	'Oyun Haritası' Veri Kümesi		'Uydu Haritası' Veri Kümesi	
		Kaynak	Kullanılan Veri Tipi ve Adedi	Kaynak	Kullanılan Veri Tipi ve Adedi
1.	Versiyon 1.0	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A ve Tip B	Google Maps	1500 adet Tip A ve Tip B
2.	Versiyon 2.0	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A ve Tip B	Bing Maps	1500 adet Tip A ve Tip B
3.	Versiyon 2.1 1. Modül	Mapgenie.io (AC: Odyssey, Greece Map)	1500 adet Tip A	Bing Maps	1500 adet Tip A
4.	Versiyon 2.1 2. Modül	Mapgenie.io (AC: Odyssey, Greece Map)	1300 adet Tip B.1 200 adet Tip B.2	Bing Maps	1300 adet Tip B.1 200 adet Tip B.2

**Tablo 1:** Veri setleri içerik tablosu

### 3.3 Modelin Eğitilmesi

Veri setlerinin yanı sıra, tekrar değeri, veri olarak kullanılan görüntülerin boyutları ve yineleme (iteration) değeri çıkan sonuçlar üzerinde etkisi olan başlıca girdilerdir. Çalışma sürecinde bahsi geçen girdilere aşağıda bulunan tablodaki (**Tablo 2**) değerler verilip sonuçları analiz edilmiştir.

Tekrar Değeri	Görüntü Boyutu	Veri Seti	Yineleme Değeri
3000	128 * 128 piksel	Versiyon 1.0	200
-	256 * 256 piksel	Versiyon 2.0	500
-	-	Versiyon 2.1 (1. Modül)	1000
-	-	Versiyon 2.1 (2. Modül)	2500
-	-	-	3000

**Tablo 2:** Girdiler tablosu

Özet olarak model 8 farklı düzende eğitilmiş ve bu 8 eğitimden elde edilen 40 farklı sonuç tablosu incelenip başarılı kabul edilen model kullanılarak çalışmanın sonuç ürünleri olan 'İstanbul - Tarihi Yarımada' ve 'İstanbul - Göksu' haritaları üretilmiştir.

### 3.4 Çıktıların Analizi

Elde edilen çıktılar analiz edilirken her bir girdinin öğrenimi nasıl etkilediğini tespit etmek amacıyla analizlerde tüm girdilerin aynı fakat sadece analiz başlığında yazan girdinin farklı olduğu sonuçlar karşılaştırılmıştır. İncelemeler, modelin 'Uydu Haritası' veri kümesinden 'Oyun Haritası' veri kümesine uyarlayarak ürettiği

sonuçları kapsamaktadır. Diğer (oyun haritasından uydu görüntüleri elde edilen) sonuçlar göz önünde bulundurulmamıştır.

Değerlendirme başlığı altında kullanılan şekiller 4\*8'lik bir ızgara yapı üzerine yerleşmiş 32 adet görüntüden oluşmaktadır. Birinci, üçüncü, beşinci ve yedinci sütunlarda bulunan 16 görüntü 'Uydu Haritası' veri setine ait görüntüler; ikinci, dördüncü, altıncı ve sekizinci sütunlarda bulunan 16 görüntü ise modelin ürettiği görüntüler olmaktadır.

### 3.4.1 Görüntü Çözünürlüğünün Öğrenime Etkisi

Eş veri sayısı ve veri içeriğine sahip fakat sadece görüntü boyutlarında farklılık olan veri setleri kullanılarak 3000 tekrar değerinde eğitilmiş iki modelin 1000 yineleme değerinde elde edilen görüntüleri aşağıda verilmiştir.

**Şekil 6:** 128\*128 piksel boyutlarında veriler kullanılarak eğitilen model





Görüntü boyutunun 128\*128 pikselden 256\*256 piksele yükseltmesi, sonuçları çözünürlük ve detay seviyesi açısından olumlu etkilemiştir. Fakat bunun yanı sıra tespit edilen bir diğer farklılık, çözünürlüğün artması ile eğitilen modelin öğrenmesi gereken doku tipi miktarı artmıştır ve bu da dönüşüm esnasında bazı dokuların kimliğini kaybetmesine sebep olmuştur. Örnek olarak 128\*128 piksel boyutlarında elde edilen sonuçlarda orman, tarla ve toprak dokuları korunmuşken, 256\*256 piksel boyutlarında elde edilen sonuçlarda ormanların kayalık dokusuna, tarlaların toprak dokusuna ve toprakların da karanlık bir dokuya dönüştüğü görülmektedir (**Şekil 6**), (**Şekil 7**). Modelin eğitilmesinde işlem gücü yetersizliği ve eğitim sürelerinin uzaması nedenleriyle daha yüksek çözünürlüklerde veriler kullanılmamıştır; çözünürlük, işlem gücü ve işlem süresi optimizasyonu sağlanmıştır.

**Şekil 7:** 256\*256 piksel boyutlarında veriler kullanılarak eğitilen model

### 3.4.2 Tekrar Değerinin Öğrenime Etkisi

Tekrar değerinin artması, daha fazla yineleme imkanı vermektedir. Bu durum doğrultusunda aynı veri seti kullanılarak yüksek tekrar değeri ile eğitilen algoritmanın hata değeri düşüş göstermektedir. Aşağıda verilen grafiklerde tekrar sayısının artması hata değerinin gittikçe düştüğünü ve yeterli tekrardan sonra belirli bir aralıkta seyrettiğini göstermektedir. Bu nedenle çalışmada yeterli tekrar değeri 3000 olarak belirlenmiş ve eğitim bu tekrar değeri ile yapılmıştır. Girdi verileri ile eğitilen algoritmada ayırıcı, üreticinin ürettiği gürültünün hata değerini puanlamış ve üretici tekrar arttıkça parabolik bir azalma gösteren hata değeri ile yeni veriler üretmiştir.



Şekil 8: Hata değer grafiği

### 3.4.3 Verileri Sınıflandırmanın Öğrenime Etkisi

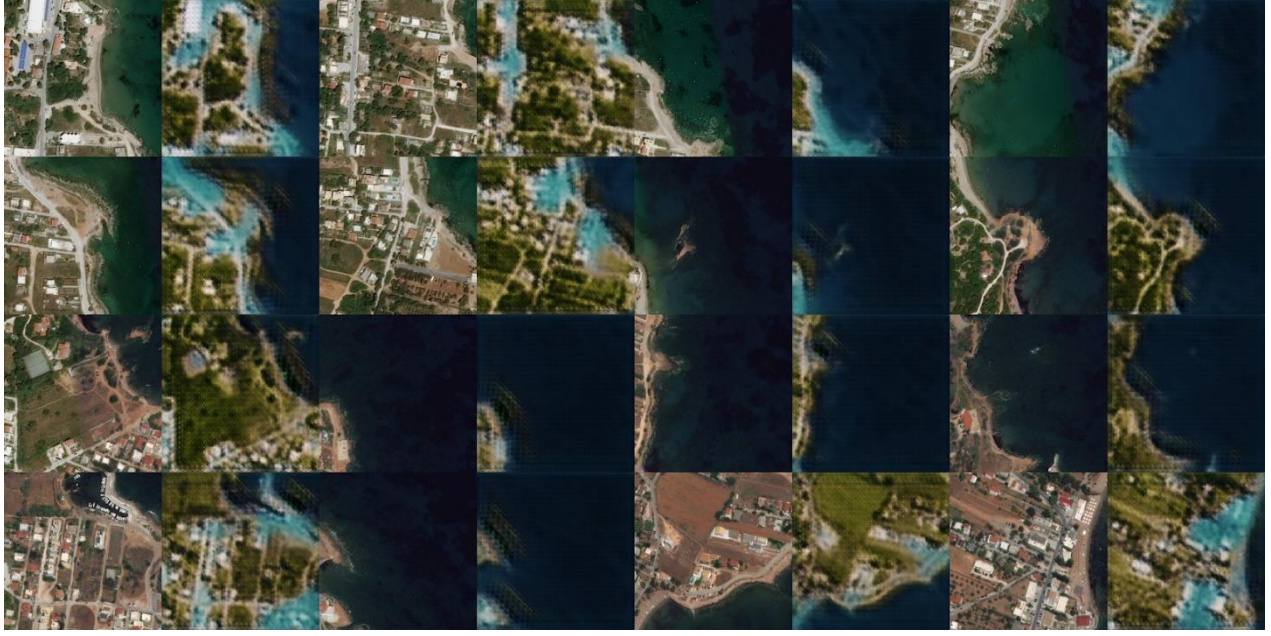
Veri kümelerinin birbirine benzerliğinin sonuç üzerindeki etkisini gözlemlemek amacıyla tüm girdilerin aynı olduğu fakat veri kümelerinin farklı olduğu denemeler yapılmıştır. İlk olarak **Tablo 1**'de Versiyon 2.0 isimli veri setleri kullanılmış, Tip A (deniz içermeyen) ve Tip B (deniz içeren) verileri aynı modele öğretilmiştir. Ardından Versiyon 2.1 isimli veri setleri kullanılarak Tip A (deniz içermeyen) verilerini barındıran 1. modül bir modele, Tip B (deniz içeren) verilerini barındıran 2. modül bir diğer modele öğretilmiştir. 256\*256 piksel boyutlarında görüntüler kullanılarak 1000 tekrar ve 1000 yineleme değerlerinde elde edilen sonuçlar aşağıdadır.



Şekil 9: Versiyon 2.0 isimli veri setleri ile eğitilen model



Şekil 10: Versiyon 2.1 - 1. Modül isimli veri setleri ile eğitilen model



**Şekil 11:** Versiyon 2.1 - 2. Modül isimli veri setleri ile eğitilen model

Elde edilen sonuçlar göz önünde bulundurulduğunda verilerin sınıflandırılması çıkan sonuçları olumlu yönde etkilemiştir. Versiyon 2.0 veri setleri ile yapılan çalışmada kara dokusu olması gereken yerlerde denize rastlanmıştır (**Şekil 9**). Versiyon 2.1 - 1. Modül ile yapılan çalışmada iki veri kümesinde de deniz görüntüleri bulunmadığından dolayı, deniz - kara dokularının birbirlerine karışması ile ilgili bir hataya rastlanılmamıştır (**Şekil 10**). Fakat aynı versiyonun 2. modülü, kara ve deniz görüntülerini içeren, ile yapılan çalışmada deniz - kara dokularının birbirlerine karışmasından doğan hatalar mevcuttur (**Şekil 11**). Söz konusu hataların Versiyon 2.0 ile yapılan çalışmalara kıyasla daha az olduğu görülmektedir.

### 3.5 Örnek Çalışma: İstanbul Haritası Üretimi

Süreç boyunca edinilen bilgiler ışığında, çalışmanın son aşaması olarak İstanbul - Tarihi Yarımada ve İstanbul - Göksu haritaları üretilmesi planlanmıştır. Gerekli olan uydu görüntüleri Bing Haritalar'dan elde edilmiş (**Şekil 12a**) (**Şekil 13a**), Versiyon 2.1 veri setleri kullanılarak 2 farklı modül ile eğitilen modeller üzerinden haritalar üretilmiştir (**Şekil 12b**) (**Şekil 13b**). 3000 tekrar değerinde, 128\*128 piksel boyutunda görüntüler kullanılarak eğitilen modelin 1000 yineleme değerindeki çıktıları aşağıda verilmiştir.



**Şekil 12 (a):** Tarihi Yarımada uydu görüntüsü

**Şekil 12 (b):** Tarihi Yarımada üretilen görüntü



**Şekil 13 (a):** Göksu uydu görüntüsü

**Şekil 13 (b):** Göksu üretilen görüntü

Örnek çalışmalarda üretilen haritalar değerlendirildiğinde bina ve orman dokularının, kıyı şeritlerinin başarılı bir şekilde oluşturulduğu görülmektedir. Bunların yanısıra Göksu örneğinde görüldüğü üzere model, oyun haritasında olmayan otoyolları ve köprüleri, aynı amaca (ulaşım ağı) hizmet eden patikalar olarak üretmiştir. Tarihi Yarımada örneğinde ise yine oyun haritasında bulunmayan teknelerin ve suda bıraktıkları izlerin başarılı bir şekilde üretildiği görülmektedir. Ancak yine aynı örnekte uydu görüntüsünde deniz ve yeşil alanların renk

değerlerinin yakınlığı nedeniyle model, yoğun yeşil alan olan yerleri deniz olarak algılayıp hatalı bir üretim gerçekleştirmiştir.

#### 4. SONUÇ

Bu çalışma harita ölçeğindeki görüntü setinin işlenebilirliğini göstermek ve bu süreçteki kazanımların farklı ölçekteki mimari gösterim tekniklerini nasıl işleyebileceğini düşündürmek için yapılmıştır. Bu süreçte model farklı veri setleri, tekrar değerleri ve görüntü boyutları ile eğitilmiş, 3.4 bölümünde anlatıldığı gibi çıktılar analiz edilmiştir.

Elde edilen sonuçlar doğrultusunda,

- 1000 - 2000 arası tekrar değeri;
- 256\*256 piksel boyutlarındaki görüntülerden oluşan veri seti;
- 1500 - 2000 adet görüntü içeren, verileri Tip A ve Tip B biçiminde sınıflandırılarak oluşturulan veri kümeleri;

ile öğrenimi yapılacak modelin, harita üretimi için başarılı sonuçlar vereceği öngörülmektedir.

Kullanılan CycleGan modelinde, eğitimde kullanılan 'Oyun Haritası' ve 'Uydu Haritası' veri tiplerinin birebir eşleşmesine gerek duyulmamıştır. Model dokular arası renk geçişlerinde başarılı olmuş, eğitim verisinde karşılığı olmayan objeleri üretebilmiştir. Bu olumlu sonuçlar karşısında bazı test veri setlerinde benzer dokuları birbirine karıştırmış, farklı açılarda ve şiddetlerdeki gölgeleri farklı dokular olarak üretmiştir. Ayrıca farklı ölçeklerdeki 'Oyun Haritası' ve 'Uydu Haritası' veri tiplerinde başarısız sonuçlar ortaya koymuştur. Her ne kadar CycleGAN modeli olumsuz yönlere sahip olsa da çalışmayı belirli bir noktaya taşımıştır.

Çalışma süreci boyunca verilerin, veri setinin ve diğer girdilerin CycleGAN modelinde sonuç ürün üzerindeki etkileri tespit edilmiş; girdilerin optimizasyonu konusunda bilgiler elde edilmiştir. Bunun yanı sıra çeşitli veri kazıma metotlarının öğrenilmesi de çalışma kazanımları arasında yerini almıştır.



Yapılan çalışmalar sonucunda, CycleGAN modelinin görüntü işleme yeteneğinin haritalar üzerindeki etkinliği gözlemlenmiştir. Kullanılan veri setlerinde değişiklik gösterilerek farklı görüntüler üretilmesine olanak sağlayan bu modelde iki farklı harita seti ile üretilen görüntüler, görüntü işlemenin mimari gösterim tekniklerindeki potansiyeli hakkında fikir vermektedir. Harita ölçeğinde yapılmış olan bu çalışma, kent veya yapı gibi farklı ölçeklerde geleneksel gösterim tekniklerinin makine öğrenmesi ile işlenebilirliğini göstermektedir. Mimari teknik çizimlerin, diyagramatik çizimlere; uydu görüntülerinin farklı haritalamalara dönüştürülebileceği çalışmalar bunlara örnek olarak gösterilebilir. Bu model ile, yine harita ölçeğinde, tarihin farklı dönemlerine ait varsayımsal haritalamalar, gelecekte farklı gezegenlerdeki kolonileşmelere yönelik simülasyonlar, geçmişin farklı seyrettiği varsayımlar olarak geliştirilen günümüz senaryolarına ait kentleşmeleri konu edinen çalışmalar; mimari ölçeğe indiğimizde ise canlandırım için belirlenen stile uygun hızlı ve sınırsız sayıda plan ve cephe dokuları üretilebilir.

### **Teşekkür**

Bu çalışma İstanbul Teknik Üniversitesi, Mimari Tasarımda Bilişim Yüksek Lisans programı 2019-2020 Bahar Yarıyılı dersi olan Mimari Tasarımda Özel Konular: Makine Öğrenmesi kapsamında yapılan bir projeden üretilmiştir.

### **Referanslar**

Çelik, G., Talu, M. F., (2019). Çekişmeli Üretken Ağ Modellerinin Görüntü Üretme Performanslarının İncelenmesi. *BAUN Fen Bil. Enst. Dergisi*, 22(1), 181-192. doi:<https://doi.org/10.25092/baunfbed.679608>

Gezer, M. (2017). "Python ile Görüntü İşlemede Örnek Bir Uygulama", Ed.: Uçar, M. Kürşad., Mühendislikte Yapay Zeka Uygulamaları, Sakarya, *Sakarya Üniversitesi Kütüphanesi Yayınları*, 65-79. isbn: 978-605-4735-98-3

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. ve Bengio, Y, (2014). "Generative Adversarial Networks", In *Proc. 27th Int. Conf. Neural Information Processing Systems*, (pp. 2672–2680), Montreal. arXiv:1406.2661

Isola, P., Zhu, J., Park, T., Efros, A. A., (2018) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. arXiv:1703.10593v6

Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks. arXiv:1611.07004v3

Mitchell, Tom M. (1997). Machine Learning, New York, McGraw-Hill Science isbn: 0070428077

Şeker, A., Diri, B., Balık, H. H., (2017). Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme. *Gazi Mühendislik Bilimleri Dergisi*, 3(3), 47-64.

Wason, R., (2018). Deep Learning: Evolution and Expansion, *Cognitive Systems Research*, 52, 701-708. doi: <https://doi.org/10.1016/j.cogsys.2018.08.023>

Wu, X., Xu, K. ve Hall, P., (2017). A survey of Image Synthesis and Editing with Generative Adversarial Networks, *Tsinghua Science and Technology*, 22(6), 660–674. doi: <http://doi.org/10.23919/TST.2017.8195348>



# Classification of Historic Ornaments with CNN

Sevgi Altun<sup>1</sup>, Cem Güneş<sup>2</sup>

0000-0002-5872-6985<sup>1</sup>, 0000-0003-1260-9852<sup>2</sup>

<sup>1</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

<sup>2</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

This paper is a critical assessment of an exploration of computer vision and deep learning methods in an architectural heritage context. Convolutional neural network, a type of deep learning is implemented to classify a group of Anatolian Seljuk ornamental patterns. The field of computer vision offers the potentials to assist studies in the field of architectural heritage. However, there are limited studies that combine knowledge across the two fields. One frequently studied topic is image classification based on features. In this study, we took on the task of classifying Anatolian Seljuk ornamental patterns to investigate the potential. The project focused on carved ornamental patterns on flat surfaces due to ease of data collection. The group of images is collected and arranged as two different yet related datasets. The classes are floral and geometrical, and subclasses are sparse and dense for both. Two different CNN architectures are used to train models for predictions. The process and effect of dataset creation on the implementation are explained. Results are discussed from both the technical and architectural points of view, providing a basis for further interdisciplinary studies.

**Received:** 20.08.2020

**Accepted:** 25.09.2020

**Corresponding Author:**

altunse19@itu.edu.tr

Altun, S., & Güneş, C. (2020). Classification of Historic Ornaments with CNN, JCoDe: Journal of Computational Design, 1(3), 115-130.

**Keywords:** Convolutional Neural Networks, Architectural Ornaments, Anatolian Seljuk Ornaments, Architectural Heritage, Artificial Intelligence

# Evrişimli Sinirsel Ağ Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması

Sevgi Altun<sup>1</sup>, Cem Güneş<sup>2</sup>

0000-0002-5872-6985<sup>1</sup>, 0000-0003-1260-9852<sup>2</sup>

<sup>1</sup>İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

<sup>2</sup>İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Bu metin, mimari miras bağlamında bilgisayarla görü ve derin öğrenme yöntemlerinin kullanımına ilişkin bir çalışmanın değerlendirmesidir. Bir tür derin öğrenme yöntemi olan evrişimli sinirsel ağ(CNN), Anadolu Selçuklu süs desenlerini sınıflandırma amacı ile uygulanmıştır. Bilgisayarla görü, mimari miras alanında bilgi sağlama ve çalışmalara yardımcı olma kapasitesine sahip olsa da, her iki alandaki bilgileri bir araya getiren sınırlı sayıda çalışma vardır. Mimarlık tarihini çalışmalarında sıkça karşılaşılan konulardan biri olan Anadolu Selçuklu süsleme desenlerinin sınıflandırılması, söz konusu potansiyeli araştırmak için bir örnek olarak seçilmiştir. Proje, veri toplama kolaylığı nedeniyle düz yüzeylerde oyma ile edilen süsleme desenlerine odaklanmıştır. Çalışma için kullanılacak fotoğraflar bir araya getirilmiş ve iki farklı ancak birbirleriyle ilişkili veri kümesi oluşturacak şekilde işlenmiştir. Sınıflar ve alt sınıflar bitkisel (seyrek / yoğun), geometrik (seyrek / yoğun) olarak belirlenmiştir. Daha sonra derin öğrenme modellerini eğitmek ve süsleme sınıfı öngörülerini elde etmek adına iki farklı evrişimli sinirsel ağ(CNN) mimarisi kullanılmıştır. Bu çalışmanın sonuçları hem teknik hem de mimari açıdan incelenmiştir. Veri kümesi oluşturmanın hem uygulama hem süreç üzerindeki etkisi incelenmiştir. Böylece çalışma, gelecekteki kültürel miras ve yapay zeka konularında disiplinler arası araştırmalara temel oluşturmayı amaçlamıştır.

**Teslim Tarihi:** 20.08.2020

**Kabul Tarihi:** 25.09.2020

**Sorumlu Yazar:**

altunse19@itu.edu.tr

Altun, S., Güneş, M. C. (2020). Evrişimli Sinirsel Ağ Kullanarak Anadolu Selçuklu Desenlerinin Sınıflandırılması. JCoDe: Journal of Computational Design, 1(3), 115-130.

**Anahtar Kelimeler:** Evrişimli Sinirsel Ağ, Mimari Süslemeler, Anadolu Selçuklu Süslemeleri, Mimari Miras, Yapay Zeka

## 1. INTRODUCTION

In this paper, the results of a machine learning experiment on the classification of architectural ornaments are used as an example to discuss the process of computer vision implementation for architectural heritage studies. The importance of problem definition and datasets are highlighted. Benefits and deficiencies are critically reviewed.

Anatolian Seljuk ornamental patterns are one of the popular topics of research in both architectural design computing and the history of architecture. Beginning from the design process part and whole relations of these patterns depend on the geometrical rules and production methods. The analysis can be done focusing on various features, semantic, or physical. In the scope of this study, we focus on the basic visual aspects of the ornaments and try to classify them concerning their figure-ground relation and curviness.

We are using convolutional neural network architectures (CNN), a type of deep neural network that is mainly used for computer vision studies. Implementation of CNN for classification may provide us a beneficial analysis of the physical and semantic relations of the ornaments. Also, it may be used for defining features of architectural ornaments in different periods.

Two datasets are created for the study. For the initial experiments, a simpler dataset with 1010 images of 2 classes is created and used. The second dataset consists of 1400 images with 4 classes. The reason for the difference between numbers is the insufficient number of images acquired for the additional two classes. The experiments are done using two different CNN architectures, details of which are explained in section 5.

## 2. RELATED WORK

In the Anatolian Seljuk Period (1077- 1308) geometric ornaments are widely used in monumental architectures such as mosques, caravanserais, and hospital.

Studies on the classification of this type of historical ornaments focus generally on their visual characteristics. The most common classification is based on the distinction between geometric and floral patterns. Subclasses used for this kind of classification are star systems, slip layouts, badges, and domes for geometric patterns and palmed, lotus, rumi, and acanthus for the floral patterns (Algan, 2008). An alternate sub classification of the patterns that are deemed as star systems relies on symmetry operations for classification (Kaplan & Salesin, 2004). While ornaments can also be classified as unit-based and line-based by defining the continuities (Bulut, 2017), one other study classifies them based on the number of ornamental strips that are brought together (Ertunç, 2016).

Adhering to a limited number of images, we use a general and simple classification, namely the distinction of floral and geometric aspects. Even though there are no studies on figure-ground relationships of the ornaments, we also try to classify the patterns based on their figure-ground relations in subclasses in a novel attempt.

Our ultimate motivation in this is to guide future studies about classifications that are informed by the making process, materials, tools of designs, styles, and their relation to the function of structures with respect to time. It is known that master builders of the Anatolian Seljuk period traveled through cities building various structures and there are researches to detect their traces on the craftsmanship for the architecture history studies (Ödekan, 1977). Our classification study can provide clues for this kind of research since figure-ground relation is can be interpreted in relation to making methods and tools.

### **3. DATA PREPARATION**

#### **3.1 Source**

The source of the images used in datasets is a research project supported by TÜBİTAK, “Computer-Aided Analysis of Design Processes of Two-Dimensional Geometric Patterns in Anatolian Seljuk Architecture” (114K283) coordinated by Prof. Dr. Mine Özkar, in 2014-2016. 92 images taken from the archive of the project are processed and classified by hand. The images of ornaments are from Anatolian Seljuk structures from Sivas, Kayseri, Konya, Erzurum, Amasya, Kütahya, Tokat, Erzincan.

The images taken from the source are rescaled and clipped from 3008 x 2000 pixels to 256 x 256 pixels. During this process, we zoomed into the related part of the image and centered the pattern in the square. Each image after preparation consist of only one ornamental pattern and the pattern occupies more than 80% of the image. Approximately 15 dataset images are extracted from each source image. The RGB values of the images are kept as they are at this stage.

### **3.2 Datasets**

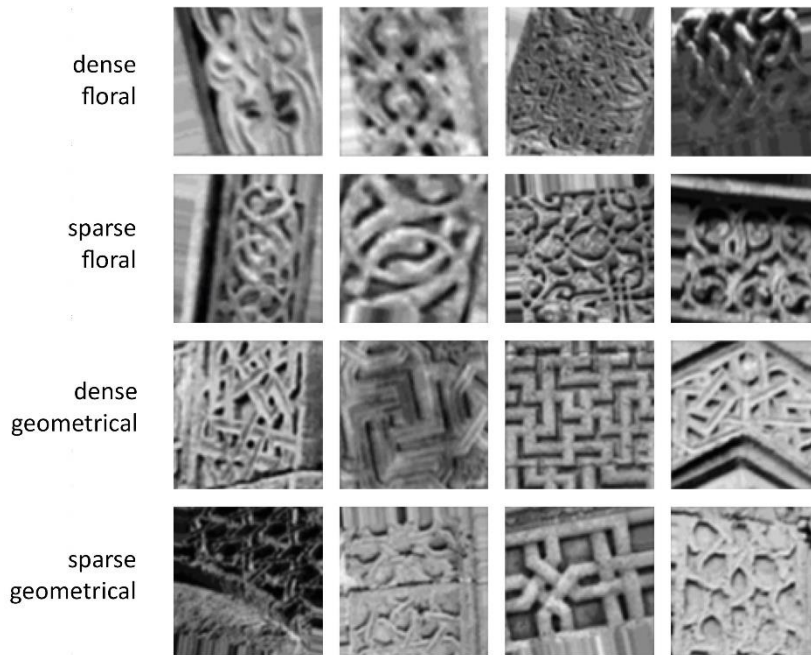
Two different datasets are used for the study. The first dataset consists of 1010 images in total with size 256 x 256 pixels divided into 2 classes: floral and geometrical. 10 % of these images are considered as validation data which is used for evaluation during training and another 5 % is the test data for evaluation after training. The second dataset consists of 1400 images in total with size 256 x 256 pixels divided into 4 classes: dense floral, sparse floral, dense geometrical, sparse geometrical (**Figure 1**). 10% of these images are validation data and another 10% is test data. As it is seen in the names this dataset consists of sub-categories of the first dataset (**Figure 2**).

### **3.3 Data Augmentation**

One of the common problems encountered in machine learning studies is overfitting. When the number of images is too limited, or the range of the data is too narrow models start to remember specific examples instead of learning how to classify new images. This is called overfitting. Since we have a limited number of images, we applied data augmentation to training data to enlarge the dataset to avoid overfitting problems. Some transformations such as shear, zoom, rotation, width shift, height shift, vertical flip, horizontal flip are applied to the images. These types of transformation methods are compatible with ornament creation methods as they also use flipping, axial symmetries, and rotations (Kaplan & Salesin, 2004). In addition to translations, images are converted from RGB to grayscale. Since only stone ornaments are used within the scope of this project and in most of the examples, these show similar characteristics, color is not one of the identifiers. The sizes of the input images are reduced from 256 x 256 x 3 pixels to 224 x 224 x 3 pixels since this is the procedure that was followed in AlexNet architecture. Smaller sized images reduce the number of parameters and helps to save GPU power. Validation data



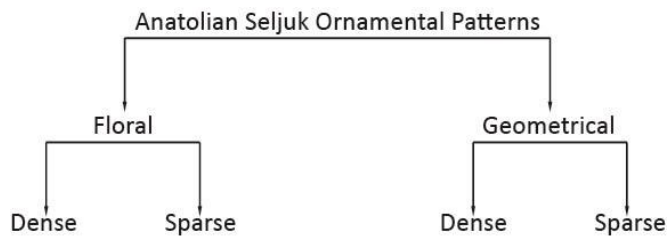
sizes are also reduced to 224x224x1 pixel. Both training and validation data are arranged in random batches of 64 images.



**Figure 1:** Data augmentation applied samples from the dataset (224 x 224 x 1 pixels)

### 3.4 Normalization

RGB values of input images are rescaled from between the range of 0 to 255 to between 0 and 1 to keep pixel values always in the dynamic range of the data type. This makes our model consistent and helps us avoid any data corruption by keeping the results of calculations in the computable interval.



**Figure 2:** Class structure of the dataset

## 4. ENVIRONMENT

The working environment for the project is Google Colab. It is used for implementations and as a coding environment. It provided a free and powerful GPU for us. It enables us to work together on files and share easily. TensorFlow Keras open-source framework is used. The dataset

is saved as .jpeg files to folders considering their classes and imported from Google Drive into Google Colab.

## 5. CNN ARCHITECTURES

We experimented with two different architectures for the project. One based on AlexNet architecture and the other is based on ResNet architecture (Figure 3-4).

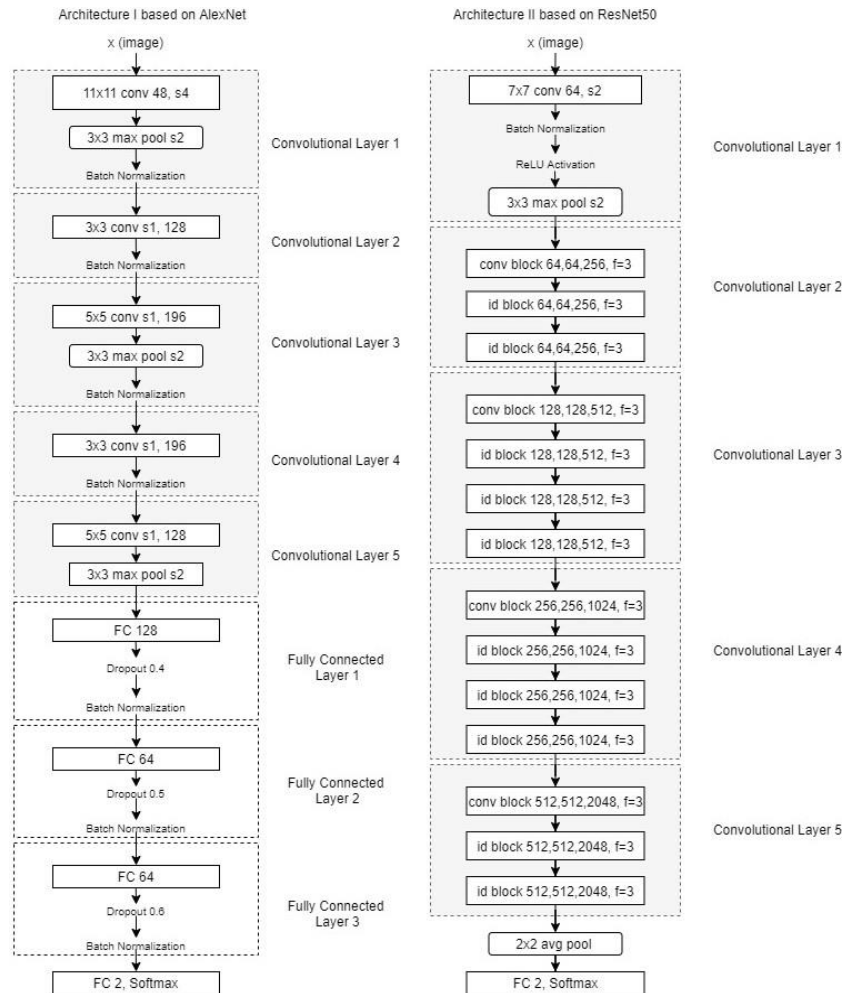


Figure 3: Details of CNN architectures

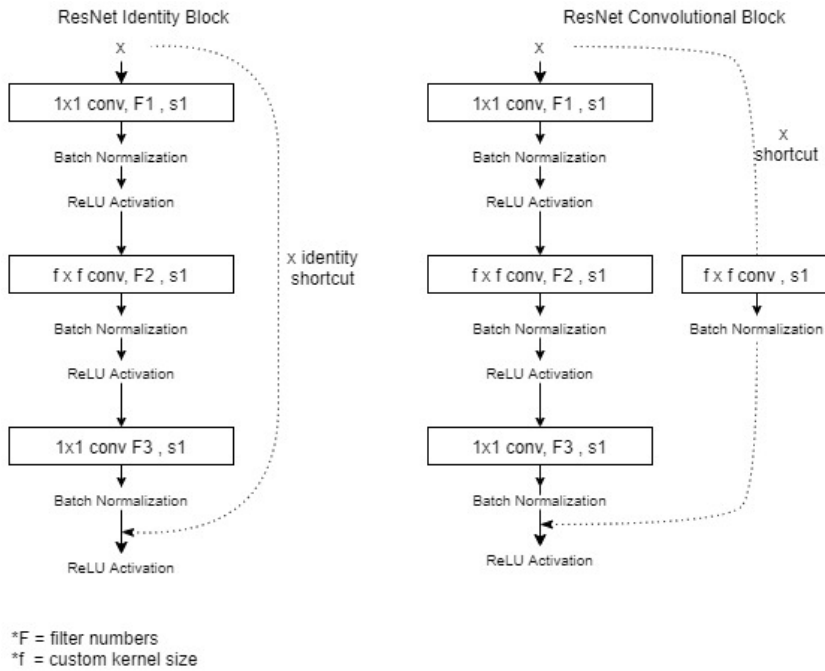


Figure 4: Details of ResNet architecture blocks

## 5.1 CNN Architecture I

The first one of the architectures is based on AlexNet architecture which is the primary successful CNN implementation example in computer vision studies. (Krizhevsky et al., 2012).

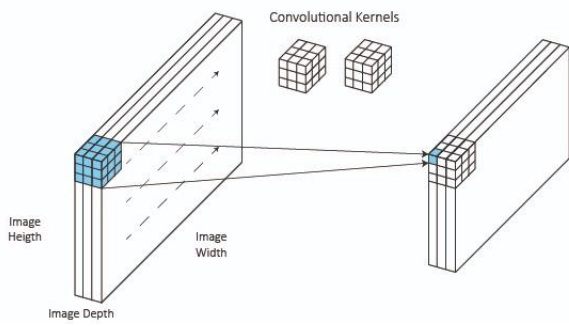
### 5.1.1 Overall Architecture

The first convolutional layer filters the  $224 \times 224 \times 1$  pixels input image with 48 kernels that are each the size of  $11 \times 11$  with the stride of  $4 \times 4$ , then we apply overlapped max pooling and batch normalization. Kernels in other words filters are the feature extractors of given input (Figure 5). They are unique square-based-arrays which can process different characteristics of the image on the other hand stride (s) is a value of the shifting of the convolution or pooling filter over the n-dimensional array after each calculation step. Max pooling is a method to extract the maximum values of initial inputs in the pooling filter zone and map them in smaller sized array (Figure 6). This process reduces the computational cost by decreasing parameter numbers. Batch Normalization (BN) is a normalization method which organizes the data to be easily processable without losing information by scaling it between 0 and 1. Mean and variance values of mini batches are used to create functional distribution after normalization. It helps to prevent overfitting and make the networks more stable (Ioffe & Szegedy, 2015). The second convolutional layer filters the  $27 \times 27 \times 48$  pixels input with

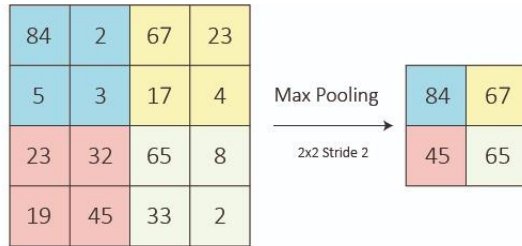
128 kernels the size of 3×3 with the stride of 1×1. After batch normalization, the third layer takes the 27×27×128 pixels input and filters with 196 kernels the size of 5×5 with the stride of 1×1. We applied max pooling and batch normalization. The fourth convolutional layer takes input 13×13×196 pixels and filters it with 196 kernels with the size of 3×3 and stride 1×1. We applied batch normalization. The fifth and final convolutional layer takes the 13×13×196 pixels input and filters with 128 kernels with the size of 3×3 and stride of 1×1. Max pooling and batch normalization are followed by a Gaussian noise layer. Three fully connected layers have 128, 64, 64 filters and 1e-4, 1e-5, 1e-5 regularization in given order. Dropout, batch normalization is applied after each fully connected layer and Gaussian noise is applied in between fully connected layers. Softmax loss which is a classifier returns probabilities of the class predictions of given image and suitable for classification of multiple images is used for the output. ReLU (Rectifier Linear Unit) activation is an activation function which prevents vanishing gradient problem and has low computational cost. Basically, based on  $f(x) = \max(0, x)$  function permits positive values to next calculation step. (Glorot, Bordes & Bengio, 2011). It is used for every convolutional and fully connected layer. Adam update which is a state-of-art stochastic optimizer and categorical cross-entropy are used (Kingma & Ba, 2015).

### **5.1.2 Reducing the Overfitting**

Since the size of our dataset is small, there was no need for a deep network. So, we reduced the filter sizes in both convolutional and fully connected layers. We tried various learning rates to fine-tune. The best performance is obtained with a 0.00005 learning rate. Dropout is the method for dropping some connections randomly between layers to intensify the learning process over different connections and promote different neurons to learn. We applied dropout with various possibilities to find the least overfitting solutions which are 0.4, 0.5, 0.6 for each dropout layer in the given order. Overlapped max pooling is applied to reduce overfitting with the pool size 3×3 with stride 2×2. Gaussian Noise is used before passing fully connected nets to reduce overfitting. 'He' is used as an initializer for the convolutional layers.



**Figure 5:** Visualization of convolution with the use of kernels



**Figure 6:** Visualization of max pooling

## 5.2 CNN Architecture II

The second architecture used for the project is mainly based on ResNet architecture (He et al., 2015). ResNet is the state of art convolutional neural network architecture which uses residual blocks to obtain better accuracy with its deep structure.

### 5.2.1 Overall Architecture

The architecture based on ResNet50 consists of two main blocks, namely the identity block and the convolutional block. The architecture consists of a series of ordered procedures: a convolutional layer, a batch normalization layer, ReLU activation, max pooling, second convolutional block, 2 identity blocks, third convolutional block, 3 identity blocks, fourth convolutional block, 5 identity blocks, 50 convolutional blocks, 2 identity blocks, an average pooling layer and an output layer with Softmax loss in the given order. Mainly the default values of ResNet50 are used for the architecture. Adam update and categorical cross-entropy are used.

### 5.2.2 Reducing the Overfitting

Because of the size of the dataset, a deeper network which has more filters resulted in overfitting. We reduced the filter sizes to  $\frac{1}{4}$  of the default to prevent overfitting. We tried various learning rates to fine-

tune the model, best performance is obtained with a 0.0001 learning rate. We applied dropout before the output layer to reduce overfitting, but it resulted in a worse case of overfitting. 'Xavier' is used as an initializer for the convolutional layers.

## 6. RESULTS

The results of the study provide both architectural and technical insights for the implementation of CNN in an architectural context (Figure 7). The classification of ornaments as floral and geometrical seem to provide accurate results. On the other hand, classification based on their figure-ground relations is more complicated and not well defined enough. Technical results are strongly related to the quality of the definition of the dataset classes.

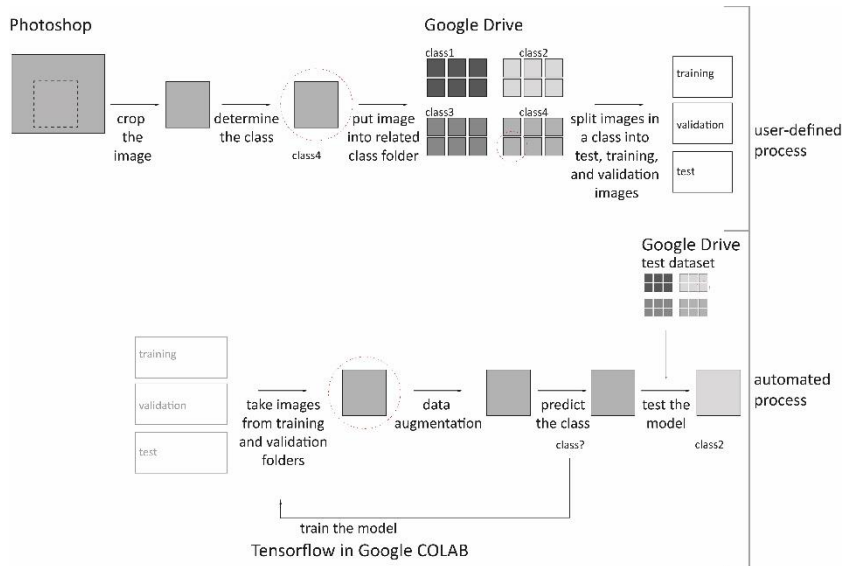
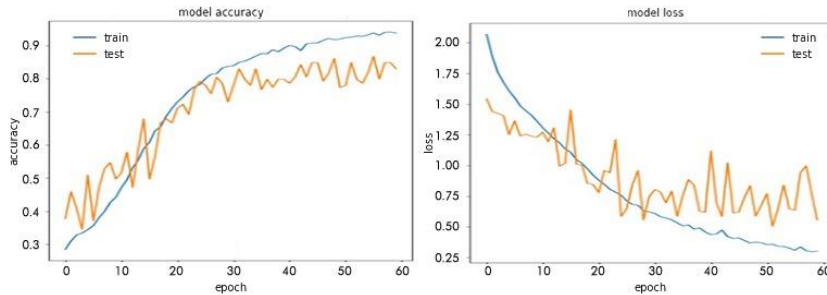


Figure 7: Diagram of the process

### 6.1 CNN Architecture I

During the training process, we saw that the training accuracy reached 0.9639 (high) over 1 and validation accuracy was reached 0.975 (high). Validation loss started with the value 1.52 and dropped to 0.61, training loss commenced at 2.24 and dropped to 0.34. Still, particular values show that the results are acceptable and successful; we see fluctuation and unstable results on accuracy and loss graphs (Figure 8). We tried to eliminate this type of result by canceling dropout and tuning the hyperparameters. However, we have not been able to make our model perform better. On the other hand, in the case with 4-classes, we encountered an overfitting problem. Even though we tried several

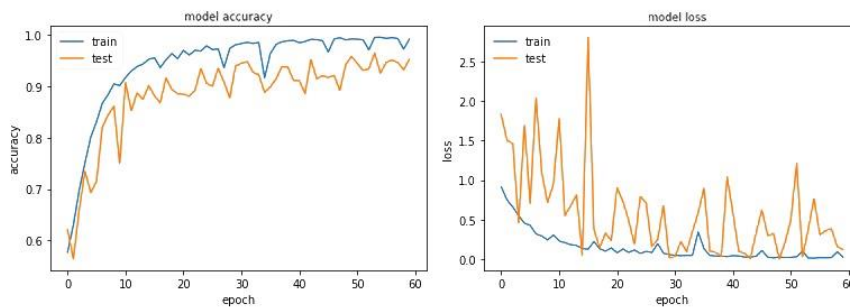
options for fine-tuning we got a minimum %15 percent difference between the validation accuracy and the training accuracy. The test accuracy is 0.90 and the test loss is 0.21. These results are compatible with our training results.



**Figure 8:** The training and the validation accuracies & the training and the validation losses graph of the Architecture I trained with the dataset of 2 classes.

## 6.2 CNN Architecture II

Training process with ResNet based architecture of 2 class dataset reached 0.99 over 1 (high) training accuracy and 0.94 (high) validation accuracy. Training loss started at 0.93 and validation loss commenced at 1.82. These respectively descended to 0.15 and 0.25. Even so, the results are acceptable for classification. The values are not equal, and we can observe immense movements in loss values in different training cycles. Additionally, the optimization process is applied to the model with 4 classes but because of the potential problems in the dataset, training results in a minimum %15 difference between training and validation accuracies. This difference shows a clear overfitting problem. As a result, we observed better stability of the accuracies in sequential training cycles and acceptable results in the 2-classes model but in the model with 4-classes we behold similar types of problems when we compared to AlexNet based architecture (**Figure 9**). The test accuracy is 0.90 and the test loss is 0.24. These results are compatible with our training results.



**Figure 9:** The training and the validation accuracies & the training and the validation losses graph of the Architecture II trained with the dataset of 2 classes.

## 7. CRITICAL REVIEW

Even though the results of the project are acceptable for the two classes we studied, from an architectural perspective the setup of the dataset lacks some desired features.

To start with, the ornaments which are subject to the study have many characteristics that are ignored during the classification of the dataset. The symmetry properties of ornaments, number of repetitions, location-dependent features, and material properties are some of these characteristics. These are ignored because they result in many classes, and generally validate the visual observation that each ornamental pattern is unique in some respects. Besides, while working on this kind of a comprehensive analysis and classification, the interdependencies between features can be observed. But these kinds of findings would fit the underlying purposes of the study, so their neglect is a key issue.

Another issue is that even if we classify patterns based only on the figure types and figure-ground ratios there are still subclasses and neglected properties. One of the biggest problems we encountered during classification was that the average figure-ground ratios are different for the floral and geometrical ornaments. While floral patterns are usually dense, the geometrical patterns are sparse. As a result, a ratio that can be classified as sparse in the context of floral patterns is dense in the context of geometrical patterns. This creates conflicts while deciding the classes. Another problem is when the pattern is carved into multiple layers or its main figures have grooves. The classification of these kinds of details requires deeper subclasses.

Finally, there is a possibility of human error in our classification. Figure-ground ratios of some of the patterns are close to 1 for the human eye. For that type of pattern, it is hard to decide to class for the dataset and they can create inconsistencies. Thus, even if the accuracy of CNN models is high, many features and values of individual ornaments are not represented and recognized.



## 8. CONCLUSION and FURTHER STUDIES

The study described here successfully classifies a group of historical architectural ornaments by using two state-of-the-art CNN architectures and two different datasets. The outputs show us using 2 classes instead of 4 gave better results due to the advantages of using more discriminative visual characteristics. The ResNet architecture was more effective and more stable compared to AlexNet due to its residual blocks.

Based on our study on the implementation of CNN in the context of architectural ornaments, we concluded that the success but acknowledge the limitations caused by the definition of the problem and the dataset. It is possible to successfully implement CNN for architectural purposes. However, it requires elaborate preliminary studies on the collection of images for setting up the dataset and the definition of the classes. Thereafter, the learning process would be easier to manipulate and the results enable us to perform comprehensive studies on architectural heritage. However, as a response to a raised number of subclasses, the computational cost would increase, and initial accuracy would get lower. This requires more work on fine-tuning of the model.

This study is an introductory example of the usage of neural networks for architectural purposes. There are various possibilities to move forward this study on architectural ornaments. As stated, only some selected properties are used for this study while there are many can be used to classify ornamental patterns and each study has potential to create information about the date, designer, location, and even semantics of the ornament. This kind of approach can also be used as the next step of the studies focusing on radial and linear symmetries of the ornaments. The classification based on the number of symmetry axes would reveal deeper information about the ornament. Another approach can be working on the number of corners and concave or convex properties of the geometry. In addition to this, another classification based on the materials (stone, marble, wood, etc.) can be carried out after creating a dataset. Combining this information with the symmetry classification, analysis can be carried on the effect of material on design and making of 2D ornaments. These classifications could not be carried out in the scope of the course project that this paper is based on since it requires a specific dataset. For the original sources used to create the data set, it would have been ideal to have

access to goal-directed photographs such as consistent lighting, angles, and distance.

Implementation of computer vision in the architectural context is an interdisciplinary area of study and it brings up new conversations between the two disciplines.

### **Acknowledgment**

The project mentioned in this paper is a term project for spring 2019 BLG506E Computer Vision course given by Hazım Kemal Ekenel in ITU, completed by the authors.

A very special thanks to Prof. Dr. Mine Özkar for providing images for the dataset and feedback throughout this project. Also, we would like to thank Res. Assist. Begüm Hamzaoğlu and Asst. Prof. Dr. Sibel Yasemin Özgan for their valuable contribution to our research on classes.

### **References**

Algan, N. (2008). *Anadolu Selçuklu Dönemi Mimarisi Taş Yüzey Süslemelerinin İncelenmesi ve Seramik Yorumları* (Unpublished doctoral dissertation or master's thesis). Dokuz Eylül Üniversitesi Güzel Sanatlar Enstitüsü Seramik Anasanat Dalı, İzmir, Turkey.

Bulut, M. (2017). Geometrik Sistemin Çözümlemesi - *Selçuklu Örnekleri Üzerine Birkaç Girişim*. *Sanat Tarihi Dergisi*. 26. 27-44. doi: 10.29135/std.292044.

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. AISTATS.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (pp. 770-778).

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ArXiv, abs/1502.03167.

Ödekan, A. (1977). *Osmanlı Öncesi Anadolu Türk Mimarisinde Mukarnaslı Portal Örtüleri*. İstanbul, Turkey: İ.T.Ü. Mimarlık Fakültesi Baskı Atölyesi.

Kaplan, C., & Salesin, D. (2004). Islamic Star Patterns in Absolute Geometry. *ACM Trans. Graph.* 23, 97-119.

Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. CoRR, arXiv: 1412.6980.

Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. CACM.

Önkol Ertunç, Ç., (2016). Anadolu Selçuklu Dönemi Taçkapıları Süsleme Şeritlerinde Tezyinat. *Pamukkale Üniversitesi İlahiyat Fakültesi Dergisi*, 0(5), 114-131. Retrieved from <https://dergipark.org.tr/tr/pub/pauifd/issue/17359/181261>

# Effects of Gamification to Private Data Collection

Tolga Bilbey<sup>1</sup>, Mehmet Tahir Sandikkaya<sup>2</sup>

0000-0002-5614-2337<sup>1</sup>, 0000-0002-9756-603X<sup>2</sup>

<sup>1</sup>Istanbul Technical University, Game and Interaction Technologies Program, Istanbul, Turkey

<sup>2</sup>Istanbul Technical University, Department of Computer Engineering, Istanbul, Turkey

Gamification is used to motivate people to complete arbitrary tasks. It gains more attention recently as the advertisement companies started using gamification globally on the Internet. The Internet infrastructure is mostly funded by advertisements globally right after its commercialization in the early-nineties. Global advertisement agencies tend to collect as much personal data as possible to better target audiences and increase profits. This leads to the discussion of whether people's privacy is at the sake to continue using the Internet. The aim of the study is to demonstrate the effects of gamification to private data collection. The effects are measured in a questionnaire in a gamified context. People's statements and their behavior is observed together with their demographic distribution. The most influential result indicates that it is possible to deviate people's behavior unwittingly. Therefore, practitioners in the area should take utmost care building ethical gamified setups.

**Received:** 05.07.2020

**Accepted:** 28.09.2020

**Corresponding Author:**

sandikkaya@itu.edu.tr

Bilbey, T. & Sandikkaya, M. T. (2020). The Effects of Gamification to Private Data Collection, JCoDe: Journal of Computational Design, 1(3), 131-152.

**Keywords:** Gamification, Questionnaire, Privacy.

# Oyunlařtırmanın Kiřisel Veri Toplanmasına Etkisi

**Tolga Bilbey<sup>1</sup>, Mehmet Tahir Sandikkaya<sup>2</sup>**

**0000-0002-5614-2337<sup>1</sup>, 0000-0002-9756-603X<sup>2</sup>**

<sup>1</sup>İstanbul Teknik Üniversitesi, Oyun ve Etkileřim Teknolojileri Programı, İstanbul, Türkiye

<sup>2</sup>İstanbul Teknik Üniversitesi, Bilgisayar Mühendislięi Bölümü, İstanbul, Türkiye

Oyunlařtırma, kiřilerin geliřigüzel görevleri tamamlamasını güdülemekte kullanılır. İnternet altyapısının doksanlı yılların bařında ticarileřmesiyle birlikte reklam gelirleri ile bu altyapı fonlandı. Böylece küresel reklam řirketleri, hedef kitleye daha iyi odaklanmak ve karlılıęı artırmak için olabildięince çok kiřisel veri toplamayı hedefler. Bu amaçla uyumlu olarak, reklam řirketlerinin İnternet üzerinden küresel çapta kullanmaya bařlamalarıyla oyunlařtırmaya ilgi artmıřtır. Bu yaklařım kiřilerin İnternet'i kullanmayı sürdürmek için özlük bilgilerinin gizlilięinden ödün vermesi gerektięi tartıřmasını ortaya çikarmaktadır. Bu çalıřma oyunlařtırmanın kiřisel veri toplanmasına etkisinin gösterilmesini amaçlar. Etkiler, oyunlařtırılmıř bir ortamda yürütölen bir anketle ölçölmüřtür. Kiřilerin beyanları ve demografik bilgileri, davranıřları ile örtüřtörelerek gözlenmiřtir. Çalıřmanın ortaya koyduęu en dikkat çekici sonuç; kiřiler farkına varmadan davranıřlarının deęiřtirilebileceęidir. Bu durumda, oyunlařtırma uygulamalarını sürdüren kiřilerin etik deęerlerle çeliřmemek için son derece dikkatli olmaları gerekir.

**Teslim Tarihi:** 05.07.2020

**Kabul Tarihi:** 28.09.2020

**Sorumlu Yazar:**

sandikkaya@itu.edu.tr

Bilbey, T. & Sandikkaya, M. T. (2020). Oyunlařtırmanın Kiřisel Veri Toplanmasına Etkisi. JCoDe: Journal of Computational Design, 1(3), 131-152.

**Anahtar Kelimeler:** Oyunlařtırma, Özlük Bilgilerinin Gizlilięi, Anket.

## 1. INTRODUCTION

Gamification is the usage of game concepts, such as mechanism design, in non-game areas to benefit from game relevant knowledge base. Gamification is used in many sectors including advertising, physical training and banking. Many of these applications aim to make boring tasks enjoyable by introducing simple challenges in a game-like concept. On the other hand, one may misuse gamification to joyfully steer people to do some task. The aim of this paper is measuring the effect of gamification to private data collection. In this manner, this study exemplifies a straightforward experimental setup to collect private data by manipulating people's behavior. The main contributions of the paper are as follows:

- Presenting private data collection violations on a gamified setup;
- Compiling data on whether people are aware of gamification or else could recognize gamified setups;
- Discussing potentially harmful usage of gamification.

The definition of gamification is using game design elements in non-game contexts (Fernández-Luna et al., 2014). These game design elements include leaderboards, points, badges, challenges, time constraints and more. The main aim of such elements in the games is introducing a challenge to a person to motivate them to continue doing a simple task more often and better. Hence, users of a game could enjoy completing tasks and self-motivate to go further. Moreover, computerized games could keep track of users' progress and report these results neatly on online social platforms to attract more users. Similarly, gamified contexts may utilize the game concepts to motivate their users. For instance, completing a daily sports routine for a long period of time may be visible on online leaderboards to motivate the users to continue training daily to not to lose their position on the leaderboard.

Related work on gamification and privacy violations in gamified systems are given in Section 2. Psychological background is presented in Section 3. The reasoning of the utilized experimental setup is explained in Section 4. Section 5 is spared to discuss the methodology of knowledge

extraction from the collected data. Finally, results are presented in Section 6 and interpreted in Section 7.

## 2. RELATED WORK

Gamification is helpful to motivate users in a context. Therefore, there are examples where it is used in customer services. Notifications of success are shown to customer service agents throughout the day as a positive feedback. Moreover, made up competitions in between customer agents are used. As a result, such a setup creates mood for competition and appreciation of success (Makanawala et al., 2013).

Tourism is another area where gamification is visible. Loyalty cards and frequent flying programs are designed with gamification in mind. Users gain some fraction of their costs as points to spend later. This motivates people to use the services more frequently, or at least stay loyal to a company. This behavior is observed in a study that discusses the effects of loyalty programs on customers (Meyer-Waarden, 2007). These approaches have a secondary use as the presented gamified setup is used in advertisements as a feature. Brands build their popularity and prestige, in the meantime people stay loyal to the now prestigious brand. Hence, gamification is mutually beneficial for both the business and the customers (Xu et al., 2013).

One interesting use of gamification is protein analysis. *Foldit* utilizes a gamification effort to solve protein folding puzzles which is an important and difficult problem for genetic engineering. The collaborative effort on solving these puzzles may help to prevent or treat diseases. The leaderboards as a competitive motivator is used (Pirker and Gütl, 2015).

E-learning platforms, or in general, most of all learning activities uses gamification to motivate students. Specific to e-learning applications, the aim is to introduce more enjoyable challenges to motivate students stay in the e-learning environment. Badges, progress bars, points, positive feedback are typical game design elements used in e-learning environments. The interactivity of the gamified learning experience

must be comparable to a traditional classroom, if not more than that (da Rocha Sexias et al., 2016).

Gamification is also used in the area of relationships. Kahnoodle was the first mobile application that supports couples to give presents, do considerate actions to collect points to fill the Love Tank. Also, in this application, couples can use these points to get a redeemable coupon in the real life. Even though, it was the first application focused on this topic, it does not exist right now (Bohyun, 2015).

Gamification can be a motivator to use earphones to reduce the possible risk of Radio Frequency (RF) emissions. BrainSaver is an Android mobile application that tracks the user behavior and gives feedback according to earphone usage. The joyful dog wallpaper is changing to a wallpaper with negative mood if the user is not wearing earphones (Burigat and Chittaro, 2014).

Everchanging Microsoft Office featured Microsoft RibbonHero. This application designed to teach users the beneficial and unknown features of the Microsoft Office. It gives points to the users that complete the given actions (Xu, 2012).

Stack Overflow is known as the biggest question and answer platform for developers. It uses gamification to regulate the content of the platform. A user can earn points by editing the content, answering the questions and getting up or down votes. Also, user can earn badges after some interaction series (Hunter and Werbach, 2015).

Gamification app called "Classroom Live" is used for undergraduate students of computer science to enhance their engagement to the course by using experience points, levels and rewards (Nah et al., 2014).

NikeFuel which is used in Nike+ platform as points generated by measurements gathered from sensors on Nike shoes and Apple iPod or iPhone regarding the sport activities of the user. Hence, users can visualize their progress and compare their performances with others (Blohm and Leimeister, 2013).



Duolingo is a gamified language learning website that is basically a paid text translation crowdsourcing platform. Students are earning badges and experience points by progressing in languages (Lamprinou and Paraskeva, 2015).

MySugr is a health-care gamified application for diabetes. Users of this application keep a virtual monster which can be tamed by logging data, being active and staying in range. Also, the monsters give feedback based on the health data given by the users (Tóth and Tóvölgyi, 2016).

iXiGo.com is a guess the destination website that shows pictures of destinations in India to the users. In the limited time, users try to guess which picture relates to which destination to get points and badges. Also, users can share their progress on social media (Sigala, 2015).

Blockholm is based on MineCraft that uses real cartography. Users can create the next generation city in their minds, so that their ideas can be real because this platform supported by Swedish Center of Architecture and Design (Fonseca et al., 2017).

Khan Academy is another e-learning platform using gamification elements. Users earn points, badges and energy points as they progress in the lectures (Morales et al., 2016).

Fitocracy is a sport assistant gamified application that application sets goals for training and users can monitor their progress using it (Huotari and Hamari, 2017).

Emotional Flower is a gamification system in work that each employee has one virtual flower that grows by positive facial expressions, and flowers are displayed at shared screen (Kuramoto et al., 2013).

Zombies, Run! is a mobile application which is available on both Android and Apple. Players are acting as a survivor from an apocalypse, and if the players run more, the players get more rewards as supplies in the story (Morford et al., 2014).

Yahoo Answers is a gamified question and answer platform like Stack Overflow, but the audience do not need to be registered users. It has leaderboard, points and badges system for gamification purposes (Schacht and Maedche, 2014).

Freshdesk is a gamified platform for employees. It converts everyday tasks like commenting to Facebook/Twitter posts, receiving telephone calls to virtual tickets and assign them to employees. Employee can earn points by completing these tickets. Also, employees can be on the leaderboard and get badges for their work (McCarthy et al., 2014).

SuperBetter is a gamified application which sets goals to overcome for people recovering from concussion. Also, it encourages the people using it (Wylie, 2014).

Opower is a gamified application that encourages to save energy. It sets goal for users, and display comparisons of related users (Zica et al., 2018).

Contrary to many benefits, gamified systems pose a major danger. The owner of a gamified system has access to users' actions. On one hand, these actions are required to be collected and interpreted to design a fair game; on the other, they disclose detailed user behavior and intentions. Given that, the system owner would like to compile such information, the likelihood of privacy invasions is high. Worse, a malicious system owner could use the gamified motivations to collect more private data. A study suggests that, a malicious party can guess the swiped unlock pattern of a mobile device at first try 62\% better if the user data is collected beforehand through a swiping game. The success rate is 76\% for three successive estimates (Acharya et al., 2019).

One typical example of gamification misuse is a discount card named *Payback* that appeared around 2000. The card offers shopping discounts based on users shopping habits. The purchasing behavior of users are collected and sold without the consumers consent (Walz and Deterding, 2015).

Another counter example is the *HealthSeeker* application. In contrast to exercise motivating health applications, HealthSeeker collects private health data for the pharmaceutical industry where the collected data is used for advertisements or passed to third party companies (Walz and Deterding, 2015).

In the light of the current discussion, one may conclude that wearable devices that keep track of daily activities of consumers could be problematic as the devices could detect a users' location continuously, their daily habits, sleeping periods or trips (Goodman, 2015).

FourSquare is a concrete example of location-based gamification usage. It helps people to socialize by featuring badges and virtual rewards. On the other hand, it attracts burglars and stalkers to infer data about their (Walz and Deterding, 2015).

Privacy concerns of people are increasing as recent technologies are continuously presented to better refine user data. A website called *Boundless AI* uses reinforcement learning, an effective machine learning algorithm, to classify enjoyable content for their users. The website suggests actions to take, then rewards users with cute cat videos or a funny animation that make users laugh. So that, users release dopamine, which is the hormone that feedbacks the neural system on enjoyable events, and become more addict. Even though this is an innovative and fun gamification setup, it could be quite dangerous as user behavior and preferences are determined in depth. In a malicious setup, a third party may manipulate users after they become addicted.

### **3. PSYCHOLOGICAL FOUNDATIONS OF GAMIFICATION**

Psychological studies suggest that gamification is a powerful tool as it is a strong motivator (Sailer et al., 2017). Business domain utilizes this tool as it enables to steer consumers to start and then continue the desired action. Recent studies on the success of gamification suggests six perspectives of the types of players. These are described below (Sailer et al., 2014):

**Trait Perspective** These players are stable over time and in changing contexts. Achievements, status reports, memberships and progress bars are the gamification elements that motivates people with trait perspective.

**Behaviorist Learning Perspective** These players are motivated with the rewards and immediate feedback.

**Cognitive Perspective** For this type of players, individual goals are important. Goal orientation, mastery orientation or performance orientation are attractive for these players. Clear and achievable goals, resulting consequences of user actions are attractive motivators.

**Perspective of Self-determination** Competence, autonomy and social relatedness are the focus of attracting this kind of players.

**Perspective of Interest** These players are heavily depending on individual preferences and contents. Adapting the difficulty level, direct feedback, providing clear goals are preferable gamification mechanisms for people with this perspective.

**Perspective of Emotion** Roles of emotions in cognitive and motivational processes are the main concern of these players. Decreasing negative feelings and increasing positive feelings are key points of gamification.

The provided perspectives could be mapped to actual gamification elements. This mapping is described below (Sailer et al., 2014):

**Points** Behaviorist learning perspective and perspective of interest.

**Badges** Trait perspective, cognitive perspective, perspective of self-determination and perspective of interest.

**Leaderboards** Trait perspective and perspective of self-determination.

**Progress bars** Cognitive perspective and perspective of interest.

**Quests** Cognitive perspective and perspective of interest.

**Meaningful stories** Perspective of self-determination, perspective of interest and perspective of emotion.

**Avatars** Perspective of self-determination and perspective of interest.

#### **4. EXPERIMENTAL STUDY**

The aim of the designed experiment is to measure people's private data disclosing habits. A basic set of survey questions is prepared in this manner. Then, a game setup is prepared to observe the effects of an

environment that contains gamification elements to people's behavior. Finally, their statements are compared with the observed behavior.

The experimental setup includes a video game as its main asset. The video game includes a simplified car driving virtualization in a virtual environment. Therefore, the experiment is not performed in a gamified environment, but directly in a game. This setup is chosen as it is extremely familiar for most people. The simplified car driving controls flattens the learning curve to a bare minimum so that the game is readily playable in the first second.

The countdown timer is intuitively set up to a shorter time span that most of the players face difficulties finishing the tour within the given interval. This approach enables to re-run the survey questions one more time or offering a bait to continue the game after the players experience the game. Therefore, comparisons of people's private data disclosure preferences could be observed by asking them direct questions and also by trying to steer them off of their initial preferences in a gamified context.

The developed experiment and the survey questions could be found online (Review, 2019). Experiments took four months with a total of 256 participants.

The video game has the clearest goal that is driving as fast as possible to meet the girlfriend that is waiting in front of her house for a date. This goal is interesting for players with cognitive perspective, perspective of interest and perspective of emotion, since it provides a clear achievable goal that promotes positive feelings inside the player. Also, using countdown timer provides continuous feedback which appeals players with trait perspective and behaviorist learning perspective. Moreover, by storing the time left in the leaderboard, players can compete, so this game mechanic is interesting for players with perspective of self-determination. To sum up, the video game has game mechanics that appeals all 6 types of players with different perspectives.

## 5. METHODOLOGY

After publishing the game created for the experiment, social media is used as a tool to promote it. There are four types of flow in the game to get information from four group of users:

G1 Demographic questions before game and private questions inside the game. All questions are optional to answer.

G2 Demographic questions before game and private questions inside the game. Only demographic questions are optional to answer.

G3 Demographic questions and private questions inside game. All questions are optional to answer.

G4 Demographic questions and private questions inside game. Only demographic questions are optional to answer.

Two sets of questions are prepared in the experiment; a set of demographic questions and another set of privacy-invasive questions. Privacy invasive questions also ask for a claim whether participants prefer to disclose private information.

The demographic data is collected to validate that the sample is composed of a diverse set of people based on age, gender, income and profession. And, private questions are based on gamification materials used by companies such as supermarket cards and coffee cards. These questions could be answered by choosing among a set of predefined values.

Experimental setup is carefully designed to observe players to support the contributions mentioned in § 1. First two hypotheses in the experiment is placed to test if gamification might cause any data collection violation. The third hypothesis is placed to show people's awareness of gamified setups. Harmful usage of gamification is discussed in § 7, after presenting the experiment results.

H1 The number of answered questions should be greater in groups G2 and G4 than in groups G1 and G3. In G2 and G4 groups, participants are forced to answer questions to continue the game. Therefore, the effect of gamification is expected to be more visible. If this

hypothesis holds, this indicates forcing players to disclose data in a gamified environment results in more data being collected.

H2 The number of answered questions should be greater in groups G3 and G4 than in groups G1 and G2. In G3 and G4 groups, questions are asked within the game. Therefore, it is expected that participants are already motivated to continue the game even if they are asked to disclose more information. If this hypothesis holds, this indicates gamified setup motivates players to disclose their personal data more.

H3 There must be participants who disclose personal data during the game even if they state the opposite. The number of participants who change their opinion is determined within the game. If this hypothesis holds, this indicates gamified setups could be utilized to change people's opinion, at least temporarily.

The reason why demographic questions are optional to answer is to get more information from the users without frustration, and let them enjoy the game without thinking about privacy. Groups G2 and G4 are experimented in this way to find out if winning the game is a strong tool enough to make user share their private information. If ratio of information sharing in G2 combined with G4 is greater than G1 combined with G3, then it can be said that forcing users to share information is an effective method. This supports the H1 in the experiment.

Another result can be extracted by comparison of G1 combined with G2 and G3 combined with G4. If the users of G3 combined with G4 share more information, then it can be said that the reason behind it is gamification. This supports the H2 in the experiment.

Finally, people are expected to keep their opinions stable if there exist no environmental motivator. The only motivator in the experimental setup is the game itself. Therefore, we could conclude if a deviation in people's opinions exists, the cause is the gamified setup.

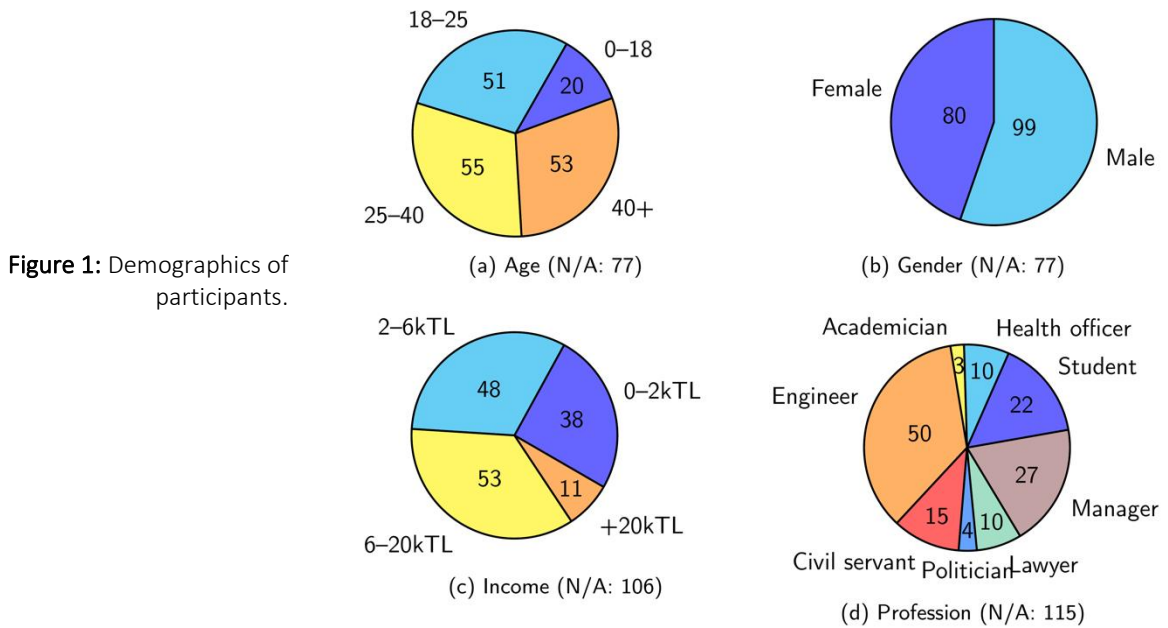
This experimental setup could shed light on whether the aforementioned hypotheses holds. It is expected that participants are tend to share their personal data more in a gamified environment and

especially when they are forced to. Additionally, it is expected that gamification will cause deviations in people's opinions.

## 6. RESULTS

The results are obtained after four months of observation with a total of 256 participants, including four groups. Even if the collected data set does not include personal identifiers, the results are presented in aggregated fashion to eliminate further inference<sup>1</sup>.

**Figure 1** includes the demographics of the participants in all four axes. Even though the participants could leave questions unanswered, it is observed that majority of them willingly answered all of the demographic questions.

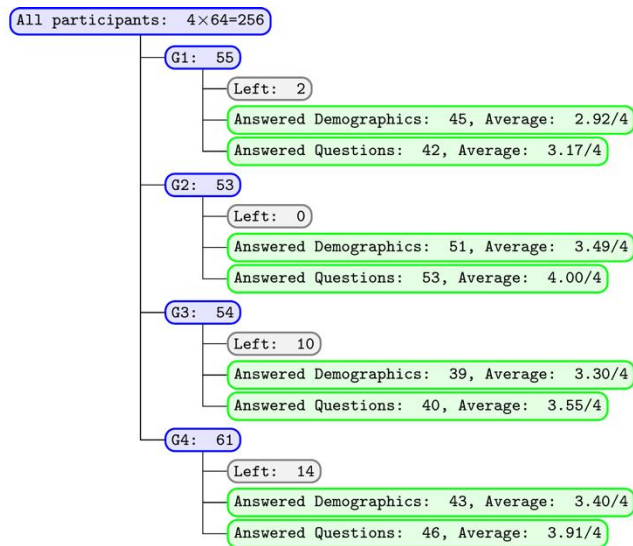


**Figure 1:** Demographics of participants.

**Figure 2** includes the summary of the experiment that will be used to determine if hypotheses H1 and H2 are valid. In this figure, the answer rates of the questions per group and question type can be seen as well as the total participants and participants who left the experiment.

<sup>1</sup> The collected anonymous data is available for researchers upon contact to the corresponding author.





**Figure 2:** Distribution of people who joined the questionnaire.

The number of participants and demographic diversity of each group is in balance as presented in **Figure 2**. The number of participating people in groups G3 and G4 is less than the number of participants in groups G1 and G2. This observation was expected as the joyful game people played is interrupted with questions. This acts as a demotivator and make people leave the experiment. Still, the number of initial participants and the number of people that participated the questionnaire are still in balance. This observation indicates that the participation bias in the experiment could be omitted.

Participants are also asked about their privacy preferences. These questions are considered privacy-invasive as privacy preferences are private itself. The set of questions are presented below:

- Do you prefer to share your personal information?
- Do you use loyalty cards of coffee shops (E.g. Coffee franchise card to have one cup of coffee for free after ten purchase)?
- Do you use supermarket loyalty cards for discounted purchase?
- Would you like to share your personal information when you get a service in return?

**Figure 3** reveals the statistics of the first question in demographics manner. People older than 25 pay more attention to their privacy than younger. Also, men are more conservative than women about their personal life. People with higher salaries are willing to share their

private data. From the professions' perspective, the less careful people are politicians where the most careful ones are the engineers.

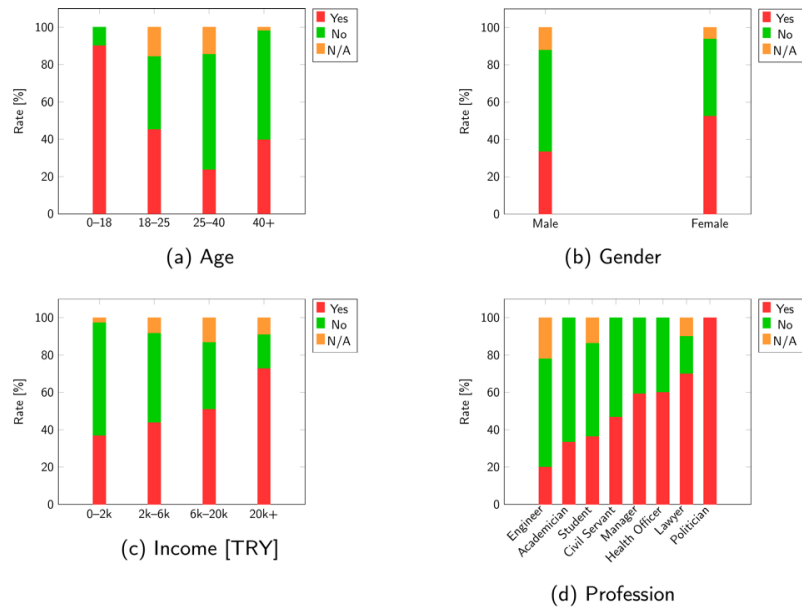


Figure 3: Willingness to share.

Figure 4 unearths the details of the second question in four demographic axes. Gift cards are popular for 0-18 age group and women use gift cards more than men. Also, people earning in the range of 6K-20K use gift cards more than others. The profession axis shows that lawyers are using gift cards more than others.

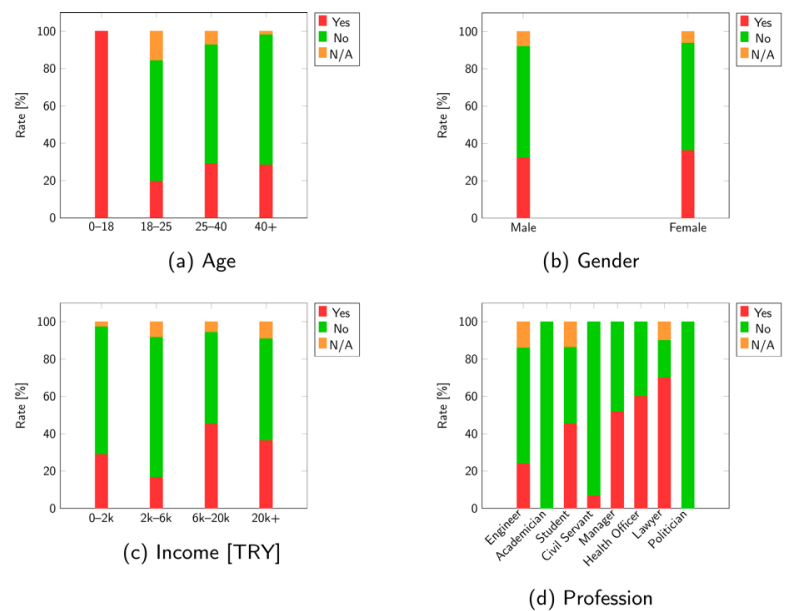


Figure 4: Gift card usage.

Figure 5 includes the details of the third question per four demographic group. Discount card usage is increasing with age, and men are using discount cards more than women. People with the income range of 6K-20K are using discount cards more than others. All of the politicians participated the experiment prefer to use discount cards.

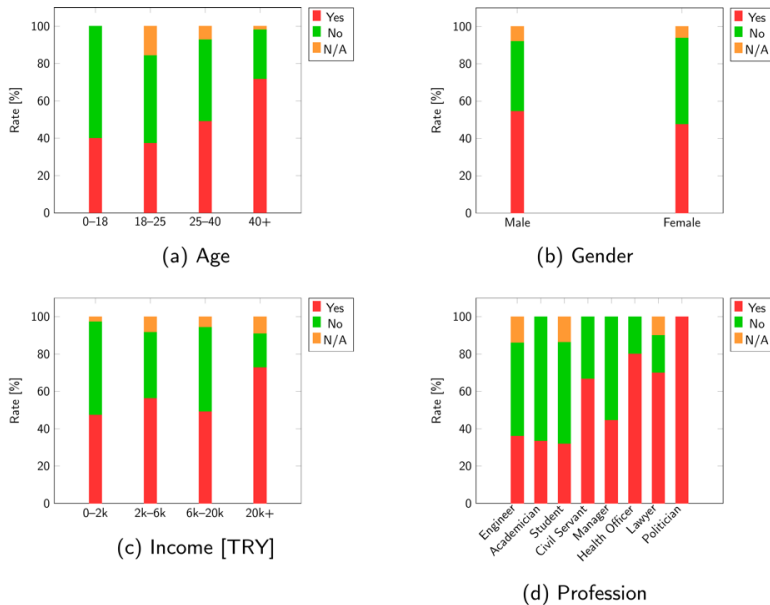
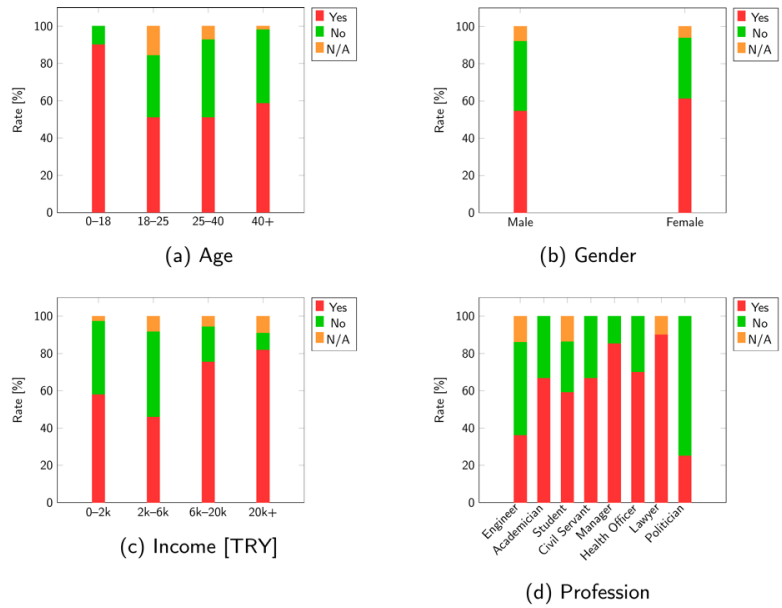


Figure 5: Discount card usage.

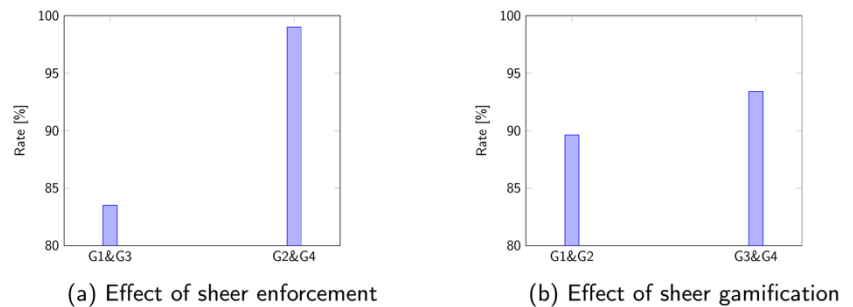
Figure 6 shows the details of the last question in demographic manner. Age group of 0-18 is more tendentious to get private data in exchange for a service. Also, women are apt to share more data in exchange for a service than men. Wealthier people share their private data to get benefits. The results in the profession axis suggests that the lawyers accept to get benefits in exchange for disclosing personal data.



**Figure 6:** Willingness to exchange private data for benefit.

In order to show the hypotheses holds, total number of collected personal items are determined and presented in **Figure 7**. The raw observations are as follows:

- G1 A total of 168 out of 212 questions have been answered.
- G2 A total of 212 out of 212 questions have been answered.
- G3 A total of 156 out of 176 questions have been answered.
- G4 A total of 184 out of 188 questions have been answered.



**Figure 7:** Percentage of shared personal data in a gamified context with respect to motivators.

Considering **Figure 7a**, the participants of the experiment answer more questions when forced. Also, **Figure 7b** indicates that the participants of the experiment are affected from gamification and disclosed more data. The results clearly indicate that the first two hypotheses, H1 and H2 holds. It should be noted that all of the answers to the personal questions are able to be collected in group G2. This group utilizes both

the game itself as the motivator and also enforces users to disclose more information to continue to play.

Participants actual behavior is compared with their answers to reveal the effects of gamification to people's self-conscious opinion. It is observed that 99 people out of 115 changed their minds to continue to play. Initially, they state that they are not willing to share their personal data. However, in a gamified environment more than 86 % of them disclose information. Again, 100 % of the participants change their mind for G2, where both gamification and enforcement are used as motivators.

The experiment tries to materialize the main hypotheses by showing that people are willing to share their personal information in a gamified environment. Also, they disclose more when they are motivated with game elements. Moreover, it is shown that people tend to modify their opinions in a gamified context.

## **7. DISCUSSION**

According to the results, the experiment reached to a diverse set of participants based on age, gender, income and profession. The number of participants is considerable to cancel out oddities in the presented statistics. Moreover, it is observed that the participants are evenly distributed among the groups; thus, participation bias is negligible.

One can conclude that majority of the people release their demographic information easily. Besides, the results indicate that people tend to release their personal information in a gamified environment more than they believe. This is an important inference as this is an evidence that people could be steered to behave against their basic preferences unconsciously by gamification.

This study concludes that users of gamified environments are less careful about protecting their private information. This conclusion is in line with Payback and FourSquare examples (Walz and Deterding, 2015).

The most successful setup is achieved by luring the participants into the gamified environment without any hassle at first, then enforcing user

to disclose personal information to go on with the game. The experiment shows 100 % success rate of collecting information and changing people's mind in this setup. This setup is in line with several legal or illegal applications in practice such as social networks, drug addiction or sect memberships.

Therefore, as basically non-preferred actions could be realized with the joyful mood of gamified virtual benefits, gamification practitioners should take utmost care to not to induce privacy violations. The intentionally malicious gamification designs are hard to recognize for individuals. Therefore, privacy violating malicious collection should be observed carefully by regulatory bodies to protect the community.

## References

- Acharya, S., Matovu, R., Serwadda, A., and Griswold-Steiner, I. (2019). Gamification of Wearable Data Collection: A Tool for Both Friend and Foe. In *Proceedings of the 2019 3rd International Conference on Compute and Data Analysis, ICCDA 2019* (pp. 68-77). New York: Association for Computing Machinery.
- Billbey, T. (2019). Oyunlaştırma Anketi. Retrieved from: <<http://game.tolgabilbey.com/>>
- Blohm, I. and Leimeister, J. M. (2013). Design of It-Based Enhancing Services for Motivational Support and Behavioral Change. *Business and Information Systems Engineering* 5, 275–278.
- Bohyun, K. (2015). Gamification, Examples, Definitions and Related Concepts. *Understanding Gamification* 51, 5–9.
- Burigat, S. and Chittaro, L. (2014). De- Signing A Mobile Persuasive Application to Encourage Reduction of Users' Exposure to Cell Phone of Emissions. *(Re)Defining Gamification: A Process Approach* 8462, 13–24.
- da Rocha Seixas, L., Sandro Gomes, A., and de Melo Filho, I. J. (2016). Effectiveness of Gamification in The Engagement of Students. *Computers in Human Behavior* 58, 48–63.
- Fernández-Luna, J. M., Huete, J. F., Rodríguez-Avila, H., and Rodríguez-Cano, J. C. (2014). Enhancing Collaborative Search Systems Engagement through Gamification. In *Proceedings of the First International Workshop on Gamification for Information Retrieval, GamifIR '14* (pp. 42–45), New York: Association for Computing Machinery.
- Fonseca, D., Villagrasa, S., Navarro, I., Re- dondo, E., Valls, F., and Sánchez, A. (2017). Urban Gamification in Architecture Education. In Rocha, Á., Correia, A.

M., Adeli, H., Reis, L. P., and Costanzo, S., (Eds), *Recent Advances in Information Systems and Technologies* (pp. 335–341). Cham: Springer International Publishing.

Goodman, M. (2015). *Future Crimes: Inside the Digital Underground and The Battle for Our Connected World*. New York: Knopf Doubleday Publishing Group.

Hunter, D. and Werbach, K. (2015). *Practical Gamification. The Gamification Toolkit: Dynamics, Mechanics, and Components for the Win*. Philadelphia: Werbach School Press.

Huotari, K. and Hamari, J. (2017). A Definition for Gamification: Anchoring Gamification in The Service Marketing Literature. *Electronic Markets* 27, 21-31. doi: 10.1007/s12525-015-0212-z.

Kuramoto, I., Ishibashi, T., Yamamoto, K., and Tsujino, Y. (2013). Stand Up, Heroes!: Gamification For Standing People On Crowded Public Transportation. In Marcus, A. (Ed), *Design, User Experience, and Usability DUXU 2013: Health, Learning, Playing, Cultural, and Cross-Cultural User Experience Proceedings* (pp. 538– 547). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-39241-2\\_59](https://doi.org/10.1007/978-3-642-39241-2_59).

Lamprinou, D. and Paraskeva, F. (2015). Gamification Design Framework Based on SDT For Student Motivation. In *2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)* (pp. 406–410). IEEE. doi:10.1109/IMCTL.2015.7359631.

Makanawala, P., Godara, J., Goldwasser, E., and Le, H. (2013). Applying Gamification in Customer Service Application to Improve Agents' Efficiency and Satisfaction. In Marcus, A. (Ed), *Design, User Experience, and Usability DUXU 2013: Health, Learning, Playing, Cultural, and Cross-Cultural User Experience Proceedings* (pp. 548– 557). Berlin, Heidelberg: Springer. [https://doi.org/10.1007/978-3-642-39241-2\\_59](https://doi.org/10.1007/978-3-642-39241-2_59).

McCarthy, I., Kietzmann, J., Robson, K., Plangger, K., and Leyland, P. (2014). Understanding Gamification of Consumer Experiences. *Advances in Consumer Research* 42, 325-356.

Meyer-Waarden, L. (2007). The Effects of Loyalty Programs on Customer Lifetime Duration and Share of Wallet. *Journal of Retailing* 83, 223–236. doi: 10.1016/j.jretai.2007.01.002

Morales, M., Amado-Salvatierra, H. R., Hernández, R., Pirker, J., and Gütl, C. (2016). A practical experience on the use of gamification in mooc courses as a strategy to increase motivation. In Uden, L., Liberona, D., and Feldmann, B. (Eds), *Learning Technology for Education in Cloud – The Changing Face of Education* (pp. 139–149). Cham: Springer International Publishing.

- Morford, Z. H., Witts, B. N., Killingsworth, K. J., and Alavosius, M. P. (2014). Gamification: The Intersection Between Behavior Analysis and Game Design Technologies. *The Behavior Analyst* 37, 25–40.
- Nah, F. F.-H., Zeng, Q., Telaprolu, V. R., Ayyappa, A. P., and Eschenbrenner, B. (2014). Gamification of Education: A Review of Literature. *HCI in Business, LNCS 8527*. Cham: Springer International Publishing.
- Pirker, J. and Gütl, C. (2015). Educational Gamified Science Simulations. In Reiners, T. and Wood, L. C. (Eds). *Gamification in Education and Business* (pp. 253–275). Cham: Springer International Publishing.
- Sailer, M., Hense, J., Mandl, J., and Klevers, M. (2014). Psychological Perspectives on Motivation Through Gamification. *Interaction Design and Architecture Journal* 19, 28–37.
- Sailer, M., Hense, J., Mayr, S., and Mandl, H. (2017). How Gamification Motivates: An Experimental Study of The Effects of Specific Game Design Elements on Psychological Need Satisfaction. *Computers in Human Behavior* 69, 371–380.
- Schacht, S. and Maedche, A. (2015). Project Knowledge Management While Simply Playing! Gaming Mechanics in Project Knowledge Systems. In Reiners, T. and Wood, L. C. (Eds), *Gamification in Education and Business* (pp. 593–614). Cham: Springer International Publishing.
- Sigala, M. (2015). Gamification for Crowdsourcing Marketing Practices: Applications and Benefits in Tourism. In Garrigos-Simon, F. J., Gil-Pechuán, I., and Estelles-Miguel, S. (Eds), *Advances in Crowdsourcing* (pp. 129–145). Cham: Springer International Publishing.
- Tóth, Á. and Tövölgyi, S. (2016). The Introduction of Gamification: A Review Paper About the Applied Gamification in The Smartphone Applications. In *7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom)* (pp. 213–218). IEEE.
- Walz, S. P. and Deterding, S. (Eds) (2015). *The Gameful World: Approaches, Issues, Applications*. Cambridge: Mit Press.
- Wylie, J. (2014). *Fitness Gamification: Concepts, Characteristics, And Applications* [Technical report]. Elon University.
- Xu, F., Weber, J., and Buhalis, D. (2013). Gamification in Tourism. In Xiang, Z. and Tussyadiah, I. (Eds), *Information and Communication Technologies in Tourism 2014* (pp. 525–537). Cham: Springer International Publishing.
- Xu, Y. (2012). *Literature Review on Web Application Gamification and Analytics* [Technical report]. University of Hawaii.



Zica, M. R., Ionica, A. C., and Leba, M. (2018). Gamification in the Context of Smart Cities. In *IOP Conference Series: Materials Science and Engineering* 294 (pp. 12-45).

# What if AI Apprentices Outperform Their Human Counterparts?

Lale Başarır<sup>1</sup>

0000-0001-8620-6429<sup>1</sup>

<sup>1</sup>Izmir University of Economics, Faculty of Fine Arts, Department of Architecture, Izmir, Turkey

The focus of this study is to draw a vision of how computer aided architecture may now evolve into artificial intelligence aided architecture (AIAA). Architects have been augmented by computers in the last decades. Therefore, the vision of such an architecture is depicted in correspondence with the ages of human evolution. Implicit knowledge of architecture is therefore explored in connection with the hierarchy of data/information/knowledge and wisdom. Therefore, the conceptual levels of AI as the narrow AI, general AI and superintelligent AI are introduced to the reader in the context of defining the current state and the possible future of AI applications. The narrow AI applications are independently worked at several different domains. This work introduces a hypothetical architect AI that learns all the knowledge of architecture during the knowledge age and later links itself to Artificial General Intelligence (AGI) in the wisdom age. An emphasis is put on occupant centric approach that architects should take on if they would want to train their future apprentices for the best and customized space creation practices. Within this context the design outcomes which are produced by AI are discussed in terms of whether they may still be considered “design”.

**Received:** 25.08.2020

**Accepted:** 30.09.2020

**Corresponding Author:**

lale.basarir@ieu.edu.tr

**Keywords:** Artificial Intelligence, Artificial General Intelligence, Artificial Intelligence Aided Architect, Architectural Knowledge, Occupant-centric Design

153

Başarır, B. (2020). What if AI Apprentices Outperform Their Human Counterparts?, JCoDe: Journal of Computational Design, 1(3), 153-166.

# Yapay Zeka Çırakları İnsan Emsallerinden Daha İyi Performans Gösterirlerse?

Lale Başarır<sup>1</sup>

0000-0001-8620-6429<sup>1</sup>

<sup>1</sup>İzmir Ekonomi Üniversitesi, Güzel Sanatlar Fakültesi, Mimarlık Bölümü, İzmir, Türkiye

Bu çalışmanın odak noktası, mimarinin bilgi tabanı bir yapay zekaya teslim edildiğinde bilgisayar destekli mimarinin artık yapay zeka destekli mimariye (AIAA) nasıl dönüşebileceğine dair bir vizyon çizmektir. Bu nedenle böyle bir mimarinin vizyonu, insanın evriminin çağlarına uygun olarak tasvir edilmiştir. Bu nedenle, mimarlığın örtük bilgisi, veri / bilgi / bilgi birikimi ve bilgelik hiyerarşisi ile bağlantılı olarak incelenmektedir. Bu nedenle, AI uygulamalarının mevcut durumunu ve olası geleceğini tanımlama bağlamında, yapay dar zekâ (YDZ), yapay genel zekâ (YGZ) ve yapay süper zekâ (YSZ) olarak kavramsal seviyeleri okuyucuya tanıtılmaktadır. Dar yapay zeka uygulamaları, farklı alanlarda birbirinden bağımsız olarak çalışılmaktadır. Bu çalışma, bilgi çağında mimarlığın tüm bilgisini öğrenen ve daha sonra kendisini bilgelik çağında YGZ'ye bağlayan varsayımsal bir mimar yapay zekayı okuyucuya sunulmaktadır. Mimarların, gelecekteki çıraklarını en iyi ve özelleştirilmiş mekanları yaratma uygulamaları için eğitmek istiyorlarsa üstlenmeleri gereken, kullanıcı odaklı yaklaşıma vurgu yapılmaktadır. Bu bağlamda, yapay zeka ile üretilen çıktıların, hala "tasarım" olarak kabul edilip edilemeyeceği tartışılmaktadır.

**Teslim Tarihi:** 25.08.2020

**Kabul Tarihi:** 30.09.2020

**Sorumlu Yazar:**

lale.basarir@ieu.edu.tr

Başarır, L. (2020). Yapay Zeka Çırakları İnsan Emsallerinden Daha İyi Performans Gösterirlerse?

JCoDe: Journal of Computational Design, 1(3), 153-166.

**Anahtar Kelimeler:** Yapay Zeka, Yapay Genel Zeka, Yapay Zeka Destekli Mimar, Mimarlık Bilgisi, Kullanıcı Odaklı Tasarım

## 1. INTRODUCTION

*Digitally mediated design and construction* did not simply provide labor-wise assistance for the industry but also evolved into a digital content that allowed for elaboration at several levels of complexity (Mitchell, 2005). Have digitally produced architectural designs revealed a new type of architecture with novel principles or methodology? Have computers changed the built environment through digital architectural design and construction? With these yet-to-be answered questions architectural design computing has been analyzed and modeled in various research work (Mitchell, 1990; Aish, 2003; Cross, 1984).

The primary concern of this work, however, is to explore whether digital architectural design is evolving from computer aided design (CAD) into artificial intelligence aided design (AIAD) as its next step. Following this concern is the next question whether architecture may continue to be a human task when the rapid pace of computational progress is taken into account. The latter question has been explored within the framework of the evolution of architectural knowledge. The standpoint for this framework is based on who handles the tacit knowledge of architecture since its inception. Whether the contemporary mainstream of architectural design is digital design or not is not a primary concern of this research. However, the new generation of digital design specialists called the *digerati* (*digital literati*) are considered as the contemporary practitioners of today's architecture (Oxman, 2006).

## 2. WHO IS IN CHARGE OF ARCHITECTURAL DESIGN?


Architectural design is a process consisting of a set of operations such as *definition of the desired objectives*, *production of alternative design solutions* and *evaluation of the expected performances* (Carrara et al., 1994).

Who holds the knowledge of architecture? The answer to this question seems to have changed several times throughout the history of humanity. The framework of viewing the shift of authorship and expertise of architectural design is depicted on a timeline **Table 1** along with a modest future forecast on who will be the next generation of architects in the midst of the century. Architectural designer's identity has changed due to occupants' needs. The nomadic, neolithic, industrial, information, knowledge and the hypothetical wisdom ages

are included in the graphic to help define designer's identity based on who the master builders of that age were.

Within this occupant-centric framework the architecture's knowledge shifted from the nature itself to the occupant during the Neolithic age. This knowledge came from the nature that is considered to be the first sheltering institution in nomadic times to the occupant who decided to settle and took on the decision making role for how and where they wanted their shelter to perform. The next main shift was during the industrial age when the architect took over the space-making specialization and seemed to somewhat dictate and or reflect upon where and how life should be enclosed in the built environment. The knowledge of architecture until this time was created by purely biological intelligence.

**Table 1:** Architectural designers' identity through ages. Developed by the author.

Architectural Designers' Identity Through Ages		Nomadic Age	Neolithic Age	Industrial Age	Information Age	Knowledge Age	Wisdom Age
WHAT		Huge Areas, Caves	Settlements Social Buildings & Houses for Extended Families &	Cities Industrial, Social buildings Transportation hubs, Homes(Houses, Apartments etc.)	Cities Physical Space, Virtual Space	Homes, Blended Cities Mixed-Reality Space	Galactic Settlements, Interplanetary Portation Space
	HOW	Carving	Masonry Stone, Mud Brick	Steam Engines Electrical Machines Iron, Steel & Glass Regulations	File-to-Factory, Construction Robots, Computer Regulations	File-to-Site, Digital twins, 3D Printers, Construction Robots Compliance Automation	Nanotechnology Programmable Matter, Microscopic / Nanoscopic drones
WHO	OCCUPANT	Hunter Gatherer	Agriculturer	Ruler & Worker	Ruler & Worker & Avatar	Ruler & Worker & Avatar	Avatar
	DESIGNER	Nature	The Occupant	The Architect	Computer Aided Architect	AI Aided Architect	The Avatar Occupant
WHEN		200,000BC	10,000BC- 18th Century	1760 1870	1960s	2000 	
		Biological			Hybrid		Non-Biological
<b>Intelligence</b>							

However, 1960s brought a whole new paradigm of the augmented architect thus creating a hybrid form of intelligence between the biological and the non-biological (Kurzweil, 2006). The shift revealed itself in the way architectural design was produced. The computers took on the role of enablers in cases requiring complex computational tasks, visualizations, and representations.

## 2.1. The Augmented Architect

Looking back at the periods until seventeenth century there was little distinction between professions and disciplines, or between architects and mathematicians (Williams and Ostwald, 2015) the master builders were still expected to master at several other disciplines. However, the eighteenth century witnessed gradual separation of professions from disciplines while the guilds and technical colleges began to emerge. Therefore, architecture began to distance from mathematics due to increased specialization. To emphasize, this distancing was the gradual result of practical moves for vocational and educational institutions educating the next generations. While specialization brought about deeper knowledge to both areas, engineering practices embraced technical and mathematical aspects and architecture gravitated towards being a profession of a comparably wider base of tacit knowledge by the twentieth century. However, with mathematical thinking models architects could clarify and/or quantify the underlying mechanisms of design making the processes more explicit and reaching optimum precision. Nevertheless, the last decades implied an era of gradually growing augmentation of the architects through integration of mathematical improvements (Williams, 2015).

A timeline that a recent research suggests (Chaillou, 2019) four consecutive and interpenetrating periods which distinguishes two levels of creation, in the last decades. The four periods are *Modularity*, *Computational Design*, *Parametricism* and *Artificial Intelligence*. Inventions coming mainly from academic research leading to innovations pave the way to a practice in constant progress. Therefore, Chaillou introduces the age of architectural AI as a culminating point resulting from the spiral feed of the architectural and the computational fields. The timeline starts with *Modularity* pinned to 1930s and lists *Gropius-Baukasten Concept*, *Le Corbusier-Le Modulor('45)*, *Buckminster Fuller-Dymaxion House('46)* and the *Dartmouth Conference-AI('56)* leading to innovations of the same period ending in 1960s. Computational design period therefore witnesses Christopher Alexander- *A Pattern Language ('68)*, N.Negroponte-*The Architecture Machine ('70)*, *Architecture Machine Group- Urban II and V.* (Chaillou, 2019, p18-19). However, as stated earlier, computation and mathematical presence in architecture dates

back to far earlier until the times of stonemasons calculating the topology and geometry of stones etc.

## 2.2. Computer Aided Architect (CAA)

Architectural design projects no longer have the sole authorship of architects. They require collective decision-making (Eastman, 2008). When the complexity of building projects raised beyond the architects' control, they nestled to their own silos within the industry sharing the workload and responsibility to engineering and construction specialists. Earlier, in the seventies, Eastman et al (1974) suggested a computer database capable of representing buildings, at a scale of construction detail comprising a set of operations, to be developed. This database was envisioned as a probable solution to inefficiencies of building drawings as the principal medium for communication among all parties of building constructions. Consequently, a computer-based description of architectural design was developed. However, there were some problems to be resolved such as hardware configuration, capability to describe geometric complexity of spaces and incorporation of data structures into the generalized database along with many issues unresolved. Looking back, we see that many of the issues that put an end to the computer aided design research is no longer valid. Thanks to the advances in digital technologies both in terms of software and hardware the whole AEC industry earned competitive skills suitable for each step of workflows.

## 2.3. Artificial Intelligence Aided Architect (AIAA)

Computers and digital technologies provided architecture with immense capabilities implemented in phases from conceptual design to 3D visualization, from design optimization to project management etc. providing architectural knowledge management at unforeseen levels of efficiency. However, those contributions can still be classified as automation of routine tasks for sharing the workload. AI applications however are expected to demonstrate learning capabilities in order to be differentiated from other computational design practices in architecture. *Computers cannot be explained to on how to do specific tasks but if once given the learning algorithm, and the examples to train on they can learn new skills* (Domingos, 2015).

Therefore, AI in architecture has mainly been yielding learning outcomes in the 1980's when models of artificial neural networks appeared in the research field. Rather than a symbolist approach it acquired the processes of human nervous system and the brain as a model. This approach called *artificial neural networks (ANN)* gave the computers the ability to learn even in the absence of explicit instructions. Here, it is important to note that this new paradigm also

implied that causality might be replaced by pattern recognition and correlations when *machine learning (ML)* is the case.

Applications of AI or ML in architecture currently suffer from several limitations: 1. *they require well balanced training sets- not yet easily available in architecture* 2. *They require large amount of data – datasets do not exist, or they are immature* 3. *They lack explainability* (Belém, 2019). However, through the explosion of research in this yet to be explored domain architects are receiving a new awareness of design that is data driven.

This explosion therefore is happening in both academia and the industry covering both theory and practice. Classification/Prediction Applications use ANNs and cover measuring similarities between architectural designs by different architects (Yoshimura, et al, 2018), analyses of changing styles in centuries (Lee, et al, 2015) or age prediction of buildings from photographs (Zeppelzauer, et al, 2018), and architectural style recognition and prediction (Mathias, 2012; Shalunts G., et al. 2012) Therefore Generative approaches include mainly Generative Adversarial Networks (GANs), a more recent model of ANNs that train and learn on large datasets and generate output. GANs basically resemble a game with two players: the generator and the discriminator. The generator creates samples, and the discriminator determines the sample to guess whether they are real or fake. Success comes from generating samples that are drawn from the same distribution as the training data. The samples however do not exist in the training data but are created by the algorithm (Goodfellow, 2017) Architectural research using GANs (Chaillou, 2019; Huang, 2018; As, 2018; Isola, 2016) explore the generation of completely new design solutions that do not exist in the training data containing floor plans and or relational graphs. Thus, advanced design systems can be trained on style and manner based on previous projects. As the researcher concludes, function does not merely represent the topological and geometric knowledge but holds the *latent utility of spaces waiting to be deciphered* (As, et al, 2018). These research studies reveal interesting and productive results for architects helping them search through and find seemingly endless number of design options. However, the architects also witness computers do what they have been trained on for years before they were licensed to practice.

So, it is time if not late to focus on the core value that architects hold which cannot be replicated by AI, if there is any. Cudzik and Radziszewski (2018) tackling this shift of knowledge have optimistic views in the sense that architectural expertise will continue to belong



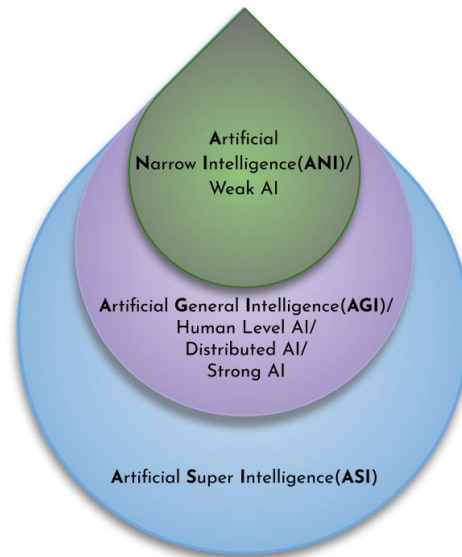
to themselves. However, they claim that architect's role will not change as the final decision maker, basing the argument on ever-changing needs of the society to be solved as the design problem. Another platform based on AI applications is XKool that might refute this optimism. XKool is an AI design cloud platform designed to handle routine tasks to help the architect focus on human innovation. Implementing the implicit knowledge into the computational intelligence. The platform is set to provide non-experts with the opportunity to create architectural designs (Leach, 2018). This resembles the democratization of design capabilities that have been witnessed recently in several other domains of creativity.

As a result, living in the knowledge age **Table1**, there is a possibility the AI aided architect may take back the responsibility and the power of expertise previously scattered around the industry and become the only licensee of the architectural expertise. However, creativity should also be handled in a new manner since *if it still exists in the design process, it should be found in constraint definition that generate the range of possible solutions to a problem, and secondly, in developing an effective method of filtering or evaluating them* (Leach, 2018).

As the discussion unfolds, there is another possibility that the AI aided architect may face an existential threat after s/he hands over all the implicit knowledge that had accumulated throughout. That scene may take place when and if AGI is achieved.

### 3. ARTIFICIAL GENERAL INTELLIGENCE

Levels of artificial intelligence has been hypothesized in **Figure 1**, referring to three main goals of AI. The experiences of the current AI research and implementations still belong to artificial narrow intelligence (ANI), also called weak AI. Therefore, it is a definition of AI models helping solve limited number of tasks (Girasa, 2020). Winning against the human chess champion is an example of weak AI as well as the superiority of AlphaGo against the human counterpart. Although the two winning models have different nature, they are both classified as narrow AI, the latter allowing unforeseen moves in an intuitive game that only humans could master until that milestone. The success was attributed to the deep reinforcement learning method of ANNs.



**Figure 1:** Three Levels of AI. Developed by the author based on Girasa's (2020) definitions.

The field has opponents of the concept of AGI, that artificial intelligence may reach human level also called the *strong AI* (Dreyfus 1972; Penrose 1989). The main reason for contradiction is that the experience of the body is an important generator of human knowledge and that *expertise* is not suitable for explicit definition. Neither can it be explained by causality. However, upon the advent of artificial neural networks, the field witnessed that this objection went obsolete since the AI models had an option of learning rather than being explicitly programmed therefore were not bound by explicit expert knowledge. Nevertheless, after twenty years Dreyfus still argued that meaning depends on context and since contexts are indeterminate, context dependence cannot be formalized. Context therefore needs a background thus a physical presence in the world to build a frame for meaning (Dreyfus, 1992). Goertzel on the other hand argues that for a mind to understand the world it should frame a context based on simplicity building up a *hierarchical* and *heterarchical* structural interpretation of the world. (Goertzel, 2020) Eventually, this discussion of whether AI will reach the level of human intelligence will find its answer, however this is not only a philosophical or scientific discussion. This is indeed a professional discussion for architects.

### Handing Over the Knowledge Base

Can AI learn better than humans? Is design being handed over to a new intelligence and/or a non-biological one? This should bring us back to the core question of who the occupant is and who the architectural designer should be. Architects had always had the mere task of creating the most appropriate built environment for the occupant. Through time, challenges changed but the core responsibility stayed the same.

Even during times when the architect had the godlike impression creating the world around us, the goal of the service was the same. The real lesson of AI according to Neil Leach *might not be how “artificial” AI is, but rather how “artificial” — and fundamentally misguided — was our previous perception of the “genius” of human intelligence* (Leach, 2018).

It is a matter of time whether or not all the applications of narrow AI will start connecting at several different levels of interoperability and merge themselves into a strong AI. But for the time being let us imagine an AI called Archiye, being a very hard-working, curious, quick, and diligent apprentice in architecture. Archiye is the next generation of *ArchiRobie* that is the version 1.0 of an AI architect who has started his design journey as an apprentice and achieved his license to design and build urban spaces just recently. *ArchiRobie* has followed the model suggested by John McCarthy (1955) who used the human brain for machine logic. Archiye has just learned that machinery in the industrial age has achieved efficiency through mass-production based on repetition, and economies of scale and that in the information age, digitally controlled machines have allowed mass-customization. She therefore doesn't need to produce any construction documents since she already designed (?) and produced the *digital twins* of 200,000,000 buildings and the robot counterparts have already started the excavation while parts and raw materials are being transferred to the sites from the most optimal locations to the sites. By this time Archiye already trained itself on modelling how each *micro- drone brick* should transfer itself and connect with which others at what instance during post-occupancy. The following day she will finish training on retrofitting and will evaluate herself to give feedback on what new knowledge she should be learning if there is any left.

Why do we look at AI while we already have the capability to design? Because computers deliver reliable and precise outcomes and most importantly, they are objectively controllable except for the case of black boxes. While AI is augmenting architects, it has the capacity to build up a knowledge base for architecture. This knowledge may scale up thanks to the hardware capacity to run non-stop at huge speeds. Consequently, the AI that has reached and learned all the explicit and implicit knowledge that is present acquires a wisdom of creating the best architectural solutions for occupants whoever they will be in the wisdom age.

## 4. CONCLUSION

This study sets out to draw a vision of architects being augmented by other disciplines and specializations. The approach of architects to their own area of expertise throughout the ages of nomadic, neolithic, industrial, information, knowledge and the hypothetical wisdom age are discussed within the context of architectural identity. The conceptual levels of AI as the narrow AI, general AI and superintelligent AI are introduced. Design within the context of AI augmented architecture is discussed. A hypothetical AI apprentice of architecture, Archiye, is depicted in order to encourage the reader to tackle the important design and knowledge issues of the profession.

Several questions are raised for today's architects as a modest call to start a discussion by introducing a hypothetical AI counterpart called Archiye that is the V2.0 of the again hypothetical and previously licenced AI architect called ArchiRobie. The resulting implication is that architects might be bound to revise the core reason for their existence within the coming decade. Nonetheless, architects need to redefine their own role and identity either to maintain or transform their profession even after architecture's knowledge base is handed over to an AI.

## References

- Aish, R. (2003). Extensible Computational Design Tools for Exploratory Architecture, In B. Kolarevic (ed), *Architecture in the Digital Age* (pp. 338-347). New York: Spon Press.
- As, I., Pal, S., and Basu, P. (2018). Artificial Intelligence in Architecture: Generating Conceptual Design via Deep Learning. *International Journal of Architectural Computing*, 16(4), 306–327.
- Belém, C., Santos, L., & Leitão, A. (2019). On the Impact of Machine Learning. *International Conference on Computer-Aided Architectural Design Futures 2019 (CAAD Futures 19)*, Porto.
- Carrara, G., Kalay, Y.E. and Novembri, G. (1994). Knowledge-based Computational Support for Architectural Design, *Automation in Construction*, 3(2–3), 123-142.
- Chaillou, S. (2019). *AI+ Architecture: Towards a New Approach*. Cambridge: Harvard University.

Cross, N. (1984). *Developments in Design Methodology*. Chichester, UK: John Wiley and Sons.

Cudzik, J., and Radziszewski, K. (2018). "Artificial Intelligence Aided Architectural Design". In *Proceedings of European Computer Aided Architecture and Design (eCAADe) 36(1)*, AI for Design and Built Environment (pp. 77-84).

Dreyfus, H. L. (1972). *What Computers Can't Do: A Critique of Artificial Reason*. New York: Harper and Row.

Dreyfus, H. L. (1992). *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge: MIT Press.

Domingos, P. (2015). *The Master Algorithm: How the Quest for The Ultimate Learning Machine Will Remake Our World*. New York: Basic Books.

Eastman, C.; Fisher, D, Lafue G., Lividini, J., Douglas, S., Yessios, C. (1974). *An Outline of the Building Description System Research* (Report No. 50). Carnegie-Mellon University, Pittsburgh, PA. Institute of Physical Planning (ED113833).

Eastman, C., Teicholz, P., Sacks, R., and Liston, K. (2008). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. John Wiley & Sons, New Jersey.

Fjelland, R. (2020). Why general artificial intelligence will not be realized. *Humanities & Social Sciences Communications* 7(10).

Girasa R. (2020). AI as a Disruptive Technology. In *Artificial Intelligence as a Disruptive Technology*. Cham: Palgrave Macmillan.

Goertzel, B. (2020). *Grounding Occam's Razor in a Formal Theory of Simplicity*. ArXiv: 2004.05269. Retrieved on 07.08.2020, from <https://arxiv.org/abs/2004.05269v2>

Goodfellow, I. (2017). NIPS 2016 Tutorial: Generative Adversarial Networks. ArXiv: 1701.00160. Retrieved on 07.11.2019, from <https://arxiv.org/abs/1701.00160>

Huang, W., Zheng, H. (2018). Architectural Drawings Recognition and Generation through Machine Learning. *The Association for Computer Aided Design in Architecture (ACADIA)*, Computational Infidelities (156-165).

Isola, P., Zhu, JY., Zhou, T., and Efros, A. A. (2018). Image-to-Image Translation with Conditional Adversarial Networks. ArXiv:

1611.07004v3. Retrieved on 01.01.2019, from <https://arxiv.org/abs/1611.07004>

Kurzweil, R. (2006). The Coming Merger of Biological and Non biological intelligence. In *Proceedings of the 2006 ACM/IEEE conference on Supercomputing (SC '06)*. New York.

Leach, N. (2018) Design in The Age of Artificial Intelligence. *Landscape Architecture Frontiers*, 6(2), 9-19.

Lee, S., Maisonneuve, N., Crandall, D.J., Efros, A.A., and Sivic, J. (2015). Linking Past to Present: Discovering Style in Two Centuries of Architecture. *2015 IEEE International Conference on Computational Photography (ICCP)*, (pp. 1-10).

Mathias, M., Martinovic, A., Weissenberg, J., Haegler, S., and Gool, L.V. (2012). Automatic Architectural Style Recognition. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 171-176.

McCarthy, J., Minsky, M., Shannon, C. E. and Rochester, N., (1955). *A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence*. Dartmouth College.

Mitchell, W.J. (1990). *The Logic of Architecture: Design Computation and Cognition*. Cambridge: MIT Press.

Mitchell, W.J. (2005). Constructing Complexity. In *B. Martens and A. Brown (eds.), Computer Aided Architectural Design Futures 2005*, (pp. 41-50). The Netherlands: Springer.

Oxman, R. (2006). Theory and Design in the First Digital Age. *Design Studies*, 27, 229-265.

Penrose, R. (1989). *The Emperor's New Mind. Concerning computers, Minds and the Law of Physics*.

Shalunts G., Haxhimusa Y. and Sablatnig R. (2012) Architectural Style Classification of Domes. In: *Bebis G. et al. (eds), Advances in Visual Computing (ISVC 2012)*. Lecture Notes in Computer Science, 7432. Berlin, Heidelberg: Springer.

Vallor S. (2017). "AI and the Automation of Wisdom", (ed) Powers T., *Philosophy and Computing*. Philosophical Studies Series, 128. (pp: 161-178). Cham: Springer.

Williams, K. and Ostwald, M.J. (eds.). (2015). *Architecture and Mathematics from Antiquity to the Future*. Switzerland: Springer International Publishing.

Yoshimura, Y., Cai, B.Y., Wang, Z., and Ratti, C. (2018). *Deep Learning Architect: Classification for Architectural Design through the Eye of Artificial Intelligence*. ArXiv: 1812.01714. Retrieved on 01.01.2020, from <https://arxiv.org/abs/1812.01714>

Zeppelzauer, M., Despotovic, M., Sakeena, M., Koch, D., and Döller, M. (2018). *Automatic Prediction of Building Age from Photographs*. ArXiv: 1804.02205v2. Retrieved on 18.11.2019, from <https://arxiv.org/abs/1804.02205v2>

# A Case Study on Evaluation of Architectural Plan Production with GAN

Can Uzun<sup>1</sup>

0000-0002-4373-9732<sup>1</sup>

<sup>1</sup> Istanbul Technical University, Graduate School of Science, Engineering, and Technology, Department of Informatics, Architectural Design Computing, Istanbul, Turkey

This study is a study on the evaluation of the methods by which the GAN algorithm is evaluated. Although GAN algorithm evaluation methods are accepted in the literature, it is still a question to be answered whether GAN evaluation methods can be used for evaluating the outputs of GAN training with a data set consisting of architectural plan schemes. Throughout this study, the architectural plan scheme outputs of GAN algorithm are evaluated. While making this evaluation, Frechet Inception Distance and Rapid Scene Classification were used, respectively, among the quantitative and qualitative evaluation methods accepted in the literature of the GAN algorithm. As a result of the evaluation, the suitability of these methods for the evaluation of autonomous architectural plan generation was discussed. At the end of the text, it is concluded that quantitative and qualitative GAN evaluation methods need new specialized methods to evaluate architectural plan scheme productions with GAN algorithm.

**Received:** 31.08.2020

**Accepted:** 04.10.2020

**Corresponding Author:**

uzunc@itu.edu.tr

Uzun, C. (2020). A Case Study on Evaluation of Architectural Plan Production with GAN. JCoDe: Journal of Computational Design, 1(3), 167-182.

**Keywords:** GAN Evaluation, Frechet Inception Distance, Rapid Scene Categorization, Autonomous Architectural Plan Scheme Generation



# GAN ile Mimari Plan Üretimlerinin Değerlendirilmesi Üzerine Bir Durum Çalışması

Can Uzun<sup>1</sup>

0000-0002-4373-9732<sup>1</sup>

<sup>1</sup>İstanbul Teknik Üniversitesi, Fen Bilimleri Enstitüsü, Bilişim Anabilim Dalı, Mimari Tasarımda Bilişim, İstanbul, Türkiye

Bu çalışma GAN algoritması değerlendirme yöntemlerin değerlendirilmesi üzerine bir çalışma niteliğindedir. GAN algoritması değerlendirme yöntemleri her ne kadar literatürde kabul görmüş olsa da mimari plan şemalarından oluşan bir veri seti eğitim çıktılarında da GAN verimliliğinin aynı değerlendirme yöntemleri ile kullanılıp kullanılmaması cevaplanması gereken bir soru halindedir. Bu çalışma boyunca GAN algoritmasının alt sınıfında bulunan DCGAN algoritması ile üretilmiş Palladyan plan şemalarının ve GAN algoritmasının verimliliği değerlendirilmiştir. Bu değerlendirme yapılırken GAN algoritmasının literatürde kabul görmüş nicel ve nitel değerlendirme yöntemlerinden sırasıyla Frechet Inception Distance ve Hızlı Sahne Sınıflandırması kullanılmıştır. Değerlendirme sonucunda bu yöntemlerin mimari plan üretimi için uygunluğu tartışılmıştır. Metnin sonunda nicel ve nitel GAN değerlendirme yöntemlerinin mimari plan şeması üretimlerini değerlendirmek üzere özelleşmiş yeni yöntemlere ihtiyacı olduğu sonucuna varılmıştır.

**Teslim Tarihi:** 31.08.2020

**Kabul Tarihi:** 04.10.2020

**Sorumlu Yazar:**

uzunc@itu.edu.tr

Uzun, C. (2020). GAN ile Mimari Plan Üretimlerinin Değerlendirilmesi Üzerine Bir Durum Çalışması. JCoDe: Journal of Computational Design, 1(3), 167-182.

**Anahtar Kelimeler:** GAN Değerlendirme, Frechet Inception Distance, Hızlı Sahne Sınıflandırması, Otonom Mimari Plan Şeması Üretimi.

## 1. GİRİŞ

Yapay zeka disiplinde yapay sinir ağlarına yönelik çalışmalar mimarlık disiplini ile etkileşime geçmeye başlamıştır. Bu etkileşimlerden biri GAN algoritmaları ile kurulmaktadır. 2014 yılında Ian Goodfellow tarafından önerilen GAN algoritmaları veriyi (görsel, işitsel) işleyerek veriye yüksek benzerlikte yeni üretimler gerçekleştirebilmektedir. (Goodfellow ve diğ., 2014). GAN algoritması, derin öğrenme algoritmalarının alt sınıfında yer alan üretken bir algoritmadır. GAN algoritması, ayrıştırıcı (discriminator) ve üretici (generator) iki fonksiyonun birbiri ile yarışması ile çalışmaktadır. Üretici fonksiyon ayrıştırıcı fonksiyonu ürettiği veri ile kandırabilirse üretimler algoritma tarafından gerçek olarak değerlendirilecek ve üretici fonksiyonun hata değeri düşmeye başlayacaktır. Bu şekilde üretici fonksiyon öğrendiği veriyi üretmeyi başarabilmektedir GAN algoritması veri üretim başarımını, görsel veriler üzerinde de oldukça verimli ve gerçekçi sonuçlar ile göstermiştir. GAN algoritmalarının görsel veriyi üretme gücü nedeniyle otonom mimari plan üretim çalışmalarında doğrudan yararlanılabilecek bir algoritma olmuştur (Huang & Zheng, 2018; Chaillou, 2019; Uzun ve diğ., 2020). GAN algoritmaları ile yapılan çalışmalar çoğunlukla piksel tabanlı imaj türünde mimari plan üretim çalışmalarını içermektedir (Huang & Zheng, 2018; Chaillou, 2019; Uzun ve diğ. 2020). Ancak otonom üretimlerin gerçekleştirilmesinin yanında diğer önemli nokta otonom üretimlerin değerlendirilmesidir. GAN ile üretilen mimari plan şemalarının değerlendirilmesi ile ilgili literatürde az sayıda çalışma bulunmaktadır (Chaillou, 2019; Uzun ve diğ. 2020). Bu durumun başlıca nedeni yapay sinir ağları ile mimarlık disiplini etkileşimi çalışmalarının oldukça yeni bir çalışma alanı olmasıdır. Güzelci ve diğ. (2019) yapay sinir ağları ile mimarlık disiplini arasındaki çalışmaların 2017 yılından itibaren artmaya başladığını göstermiştir. Aynı şekilde 2017 yılından itibaren GAN algoritmaları ile mimarlık disiplini arakesitinde çalışmalar görülmüştür, ancak literatürde hala gelişmekte olan bir alandır (Huang & Zheng, 2018; Chaillou, 2019; Uzun ve diğ. 2020).

GAN algoritma eğitimi başlı başına bir alanken, algoritmanın değerlendirilmesi GAN değerlendirme yöntemleri olarak farklı bir alan açmaktadır (Borji, 2019). GAN üretimlerinin değerlendirilmesi algoritmanın değerlendirilmesinden farklı bir ifadedir. GAN üretimleri çoğunlukla GAN değerlendirme yöntemleri ile değerlendirilmektedir (Borji, 2019). GAN değerlendirme yöntemleri literatüründen

anlaşılacağı üzere, eğer GAN algoritması değerlendirmesinde GAN algoritmasının performansı verimli ise GAN üretimleri de verimlidir şeklinde bir sonuca ulaşılmaktadır. Ancak mimari plan üretiminde GAN algoritmasının verimli çalışması her zaman doğru mimari plan şeması mı üretir sorusu yanıtlanmamış bir soru olarak bulunmaktadır. Mimari plan, GAN algoritma eğitimlerinde sıkça kullanılan fotoğraflarda piksel değerlerinden gelen salt görsel veri dışında, mekansal dizilim kararları verisini de içermektedir. GAN algoritması üretimleri değerlendirilmek üzere bir çok nitel ve nicel yöntemle sahiptir. Ancak bu değerlendirme yöntemleri iyi bir mimari plan üretiminin gerçekleştiğini gösterip göstermediği en önemli sorudur. Bu nedenle bu metnin amacı GAN algoritmasının nicel ve nitel değerlendirme yöntemlerinin, mimari üretim çıktılarının mimari anlamda değerlendirilmesinde yeterli olup olmadığının araştırılmasıdır. Bu bağlamda bu çalışma GAN ile mimari plan şeması üretimlerinin değerlendirilme yöntemlerinin bir değerlendirilmesi çalışmasıdır.

Bu değerlendirmenin gerçekleştirilmesi amacıyla GAN algoritmasının eğitiminin Palladyan plan şemaları ile gerçekleştirilmesi kararlaştırılmıştır. Bu seçimin nedeni Palladyan plan şemalarında sürekli tekrar eden plan kurgusu nedeniyle üretim çıktıları ile veri setinin karşılaştırılmasının daha kolay olabilmesidir. Böylece değerlendirme sürecinde karmaşıklığı az okunaklı plan şemaları karşılaştırılarak değerlendirilebilecektir.

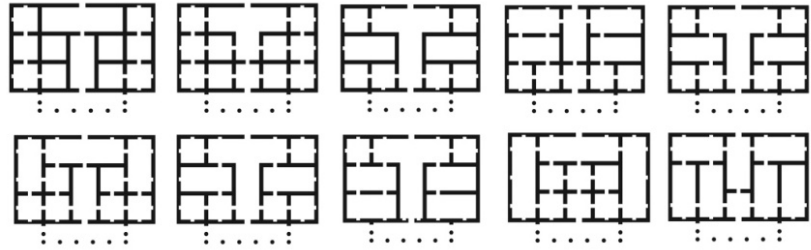
GAN eğitimi sürecinde ilk denenen veri seti gerçek Palladyan plan şemalarından oluşan bir veri seti olmuştur. Ancak gerçek Palladyan plan şemaları eğitim için yeterli sayıda veriyi oluşturamamıştır. Bunun yanında veri setinde bulunan gerçek Palladyan plan şemalarının çözünürlük kalitesi düşüktür. Bu nedenlerden dolayı GAN algoritması doğru bir genelleme yapamayarak çözünürlüğü ve okunurluğu düşük üretimler gerçekleştirmiştir. Bu üretimler GAN değerlendirme yöntemlerinde kullanılmak için yeterli kalitede olamamıştır. Bu nedenle gerçek Palladyan plan şemalarından oluşan veri seti ile gerçekleştirilen eğitim değerlendirilmemiştir. Veri setinin daha temiz ve yeterli büyüklükte olabilmesi için algoritma üretimi bir Palladyan plan şeması veri setinin üretilmesi kararlaştırılmıştır. Bu nedenle veri setinin üretiminde GRAPE-SGI (Grasl, T. (t.y.)) adında Palladyan plan şeması üreten bir biçim grameri yorumlayıcısı kullanılmıştır. GRAPE-SGI ile 320

adet Palladyan plan şeması üretilmiş ve GAN eğitimi bu veri seti ile gerçekleştirilmiştir.

Algoritma eğitim sürecinde GAN alt sınıfında bulunan DCGAN algoritması GRAPE-SGI ile üretilmiş 320 adet Palladyan plan şemalarından oluşan veri seti ile 15000 epok eğitilmiştir. Epok algoritma eğitimi sürecinde algoritmanın veri setinin tümünü kaç kere tekrar okuduğunun sayısal değeridir. GRAPE-SGI çevrimiçi

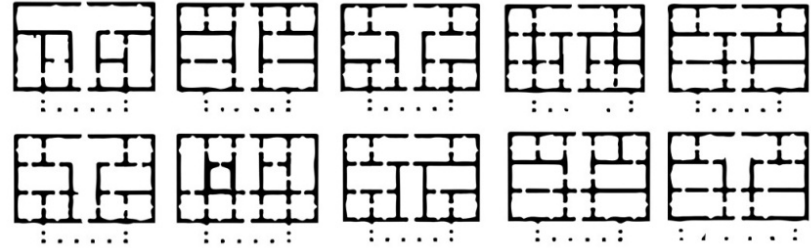
**Şekil 1'**de GRAPE-SGI ile oluşturulan Palladyan plan şeması veri setinden 10 plan şeması gösterilmiştir.

**Şekil 1:** GRAPE-SGI üretimi plan şemaları



15000 epok eğitilen DCGAN algoritmasının üretim çıktılarından 10 tanesi **Şekil 2'**de gösterilmiştir.

**Şekil 2:** DCGAN üretimi plan şemaları



İlk bakışta görseller birbirine benzese de, DCGAN'ın gerçekten verimli çalışıp çalışmadığı GAN değerlendirme yöntemleri ile test edilmelidir. Takip eden bölümlerde nicel ve nitel olarak eğitilen GAN algoritması değerlendirilmiş ve bu değerlendirme yöntemlerinin, üretilen mimari plan şemalarının değerlendirilmesindeki rolü tartışılmıştır.

## 2. GAN ALGORİTMASI DEĞERLENDİRME YÖNTEMLERİ

GAN algoritmaları ile mimari plan üretiminin otonom olabilmesi mümkün hale gelmiştir (Huang & Zheng, 2018; Chaillou, 2019; Uzun ve diğ., 2020). Üretimin gerçekleşmesi ile birlikte GAN algoritmasının üretim çıktılarının da değerlendirilmesi üretim verimliliğinin kararının

verilmesi için gerekli hale gelmektedir. Değerlendirme sonucunda hem GAN algoritması mimarisinin işlerliği ve verimliliği hem de üretim çıktılarının kalitesi değerlendirilmektedir. Üretimler değerlendirilmeden kullanılması halinde rastgele ve gerçek olmayan bir üretimin gerçekleşmiş olma ihtimali olacaktır. Algoritma üretim çıktılarının değerlendirilmesiyle algoritmanın hangi yönde üretimler yaptığı incelenebilir ve üretimlerdeki hatalı çıktılar temizlenebilir ya da hata payı büyük bir değerdeyse algoritma revize edilerek doğru yönelimde üretimlerin yapılması sağlanabilir. Bu nedenle algoritmanın üretim çıktılarının değerlendirilmesi, algoritma eğitim süreci kadar önemlidir.

GAN algoritmalarının üretim çıktılarının değerlendirilmesi yöntemleri ikiye ayrılmaktadır. Borji (2019), bu yöntemleri nicel ve nitel yöntemler olmak üzere ikiye ayırmaktadır. Nicel ve nitel yöntemlerin her biri kendi içinde avantaj ve dezavantajları da içermektedir.

Nicel yöntemlerden bazıları; Average Log-Likelihood, Coverage Metric, Inception Score, Modified Inception Score, Mode Score, Frechet Inception Distance, The Wasserstrain Critic, Geometry Score, Image Quality Measure, Precision Score ve F1 Score olarak sıralanabilir (Borji, 2019). Nicel değerlendirme yöntemlerinden veri seti ve üretim çıktısı arasındaki benzerliğin araştırıldığı yöntemlerde, logaritmik fonksiyonlar üzerinden tanımlanan olasılıksal yöntemler kullanılmaktadır. Böylece veri seti ve üretim çıktısı arasındaki benzerlik olasılıksal bir ifade ile tarif edilebilmektedir.

Nitel yöntemlerde GAN algoritması çıktıları insan gözlemci tarafından değerlendirilmektedir. Borji (2019), bu durumu insan yargılarının öznel olması nedeniyle dezavantajlı olduğunu söylemiştir ve nitel değerlendirmedeki üç problemi sıralamıştır. Borji'ye (2019) göre insan değerlendirmesi öznel yargılar nedeniyle yanlı olabilir bunun yanında yeteri kadar insan gözlemci bulma problemi ile karşılaşmaktadır. Bir diğer problem ise yine insanın öznel yargılarından kaynaklı insan gözleminin varyansının oldukça büyük olmasıdır. Başka bir deyişle herhangi bir veriden bir genelleme yaparak doğruya ulaşması ancak büyük bir veri seti ile karşılaşması sonucunda mümkün olabilir. Bu nedenle insan gözlemcinin değerlendirmesi sırasında karşılaştığı örneklem sayısı yüksek olmalıdır. Sonucu problem ise GAN algoritmasının mimarisi nedeniyle oluşan problemleri insan gözlemcinin tespit edebilecek algılarının bulunmamasıdır. Yüksek düzeyde işlem ve

hesaplama gerektiren bu sürecin insan tarafından yapılabilmesi mümkün olmayacaktır (Borji, 2019). Bu nedenle de GAN algoritmasının teknik problemleri nitel değerlendirme yöntemlerinde gözden kaçabilecek problemlerdir. Bu teknik problemler, aşırı uyum (overfitting), mod yığılımı (mode collapse) ve mod düşüşü (mod drop) şeklinde sıralanabilir. Aşırı uyum algoritmanın eğitim sırasında veri setini ezberlemesi ve ezberlediği veri dışında üretim yapmamış olmasıdır. Çözünürlük anlamında çok iyi olan bir GAN algoritma çıktısının veri setinden kopyalanmış, ezberlenmiş bir veri olduğunu insan gözlemci kaçırabilir. Mod yığılımı ise aynı görselin tekrar tekrar üretilmesi şeklindedir. Mod düşüşü ise üretimlerde hiç bir gerçek üretimin gözlemlenmemiş olması durumudur. Tüm bu problemler insan gözlemcinin gözünden kaçabilecek özellikleri barındırmaktadır (Borji, 2019). Bu nedenle nitel yöntemlerde değerlendirme sürecinde hatalar ile karşılaşılabilir. Nitel yöntemlerdeki bu dezavantaja rağmen algoritma üretim çıktılarının değerlendirilmesinde sıkça kullanılan bir yöntemdir. Hızlı sahne sınıflandırması (rapid scene categorization), tercih ile değerlendirme (preference judgement), en yakın komşu (nearest neighbor) nitel değerlendirme yöntemlerinde sınıflandırılmışlardır (Borji, 2019).

### **3. GAN ALGORİTMASI MİMARİ ÜRETİMLERİNİN DEĞERLENDİRMESİ**

Bu bölümde GAN değerlendirme yöntemlerinden nicel ve nitel yöntemler kullanılarak elde edilen sonuçlar paylaşılacaktır. GAN ağlarını nicel değerlendirme yöntemlerinden Frechet Inception Distance (FID), nitel değerlendirme yöntemlerinden Hızlı Sahne Sınıflandırması (Rapid Scene Categorization) kullanılarak değerlendirme süreci ve sonuçları paylaşılacaktır.

#### **3.1 FID ile Değerlendirme**

Frechet Inception Distance GAN algoritması nicel değerlendirme yöntemleri arasında sınıflandırılmıştır. FID sentetik verinin dağılımı ile gerçek veri dağılımını karşılaştırarak sayısal bir sonuç vermektedir (Borji, 2019). FID yönteminin en önemli avantajı verinin gürültülü olması halinde de verimli çalışabiliyor olmasıdır. Ancak problemler olan nokta FID yöntemi çalıştırılırken algoritma, karşılaştırılması yapılacak olan veriyi normal dağılımda (Gaussian / Normal distribution) kabul etmektedir (Borji, 2019).

Veri bilimcilerine göre gerçek dünyayı temsil eden her bir veri seti normal dağılıma (Gaussian / Normal distribution) sahiptir (URL1). Normal dağılım veriyi temsil eden değişkenlerin aritmetik ortalamasının sıfır olduğu dağılımdır. Görsel verilerden oluşan bir veri setinde özellikle siyah ve beyaz piksellerden oluşan mimari plan veri setinden veriyi temsil eden değişkenler (öznitelikler) siyah ve beyaz piksellerdir. Mimari planlardan oluşan bir veri setinin normal dağılıma sahip olabilmesi ise mimarlık disiplini içinden gelen bir diğer problem olacaktır. Mimari plan şemalarının tasarım eylemi nedeniyle çeşitlilik göstermesi normal dağılımı etkileme ihtimali olan bir özelliktir. Bunun yanında mimari planlardan oluşan büyük bir veri setinin henüz üretilmemiş olması da mimari plan verisini temsil edecek bir gerçek dünya modeli oluşturan veri setinin GAN algoritması eğitimlerinde kullanılamayacak olmasıdır. Tüm bu nedenlerden dolayı nicel olarak GAN algoritması eğitim çıktılarının değerlendirilmesi güvenilirliğinin kritik olduğu unutulmamalıdır. Ancak Borji (2019), normal dağılımla ilgili bu problemin her tür veri seti için değerlendirme verimini düşürebilecek bir özellik olduğunu belirtmiştir. Mimari plandaki özel durum ise normal dağılıma sahip bir veri setinin henüz üretilmemiş olmasıdır.

FID yönteminin çalışma prensibinde iki farklı veri kümesinin özniteliklerinin olasılıksal olarak dağılımlarının hesaplanması bulunmaktadır (Brownlee, 2019). FID veri setindeki gerçek veri ve DCGAN üretimindeki sentetik verinin her biri için öznitelik dağılımı hesaplamakta ve bu dağılımlar arasındaki mesafeyi bir skor olarak sonuçlandırmaktadır. Öznitelikleri veri içinde vektörlerle temsil edilmektedir. Bu vektörler veri içerisindeki değişkenlerdir. Gerçek görsel veri ve sentetik görsel verinin özniteliklerini temsil eden vektör değerleri arasındaki fark FID skorunu vermektedir. FID ismindeki mesafe (distance) bu iki veri arasındaki vektörel mesafe ile açıklanmaktadır. **Denklem 1**'de FID fonksiyonu gösterilmiştir (Brownlee, 2019).

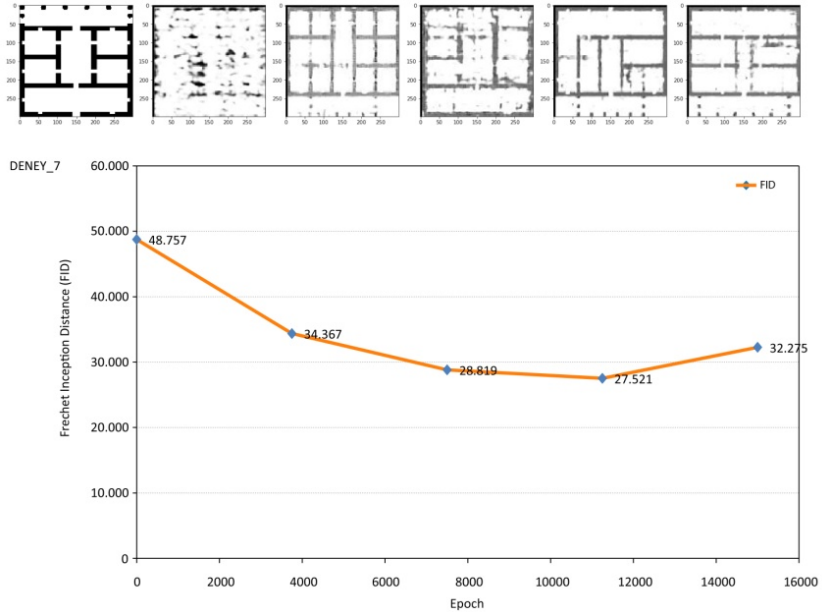
$$d^2 = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2 * \text{sqrt}(C_1 * C_2)) \quad (1)$$

**Denklem 1**'de mesafenin karesi  $d^2$  ile ifade edilmektedir.  $C_1$  ve  $C_2$  ile kovaryans matrislerini göstermektedir. Veri seti ve DCGAN öznitelik dağılımları da  $\mu_1$  ve  $\mu_2$  ile temsil edilmiştir. Tüm bu fonksiyon ile sentetik ve gerçek veri arasındaki mesafenin sayısal değeri öznitelik

dağılımları kullanılarak elde edilebilir. GAN algoritmasının çalışmasının verimli olduğunu ifade etmek amacıyla her ne kadar FID değerinin olabildiğince küçük olması beklense de FID değerinin 0'a eşit çıkması algoritmanın veri seti ile birebir aynı üretim yaptığını göstermektedir. Başka bir deyişle yukarıda değindiğimiz algoritma eğitim problemlerinden aşırı uyum probleminin yaşandığı söylenebilir. FID skorunun 0 olması halinde GAN algoritmasının mimarisi ya da veri seti kaynaklı bir problem olduğu okunabilir. GAN algoritması değerlendirmesinde FID skoru kullanılırken beklenen değer ise her bir epok sonrasında FID skorunun sürekli düşen bir değer almasıdır. Böylece algoritmanın sürekli öğrenme yöneliminde ve kendini sürekli geliştirerek daha iyi temsiller ürettiği sonucuna varılabilir.

**Şekil 3**'te DCGAN algoritması kullanılarak üretilen Palladyan plan şemalarının GRAPE-SGI ile üretilen Palladyan plan şemalarından oluşan veri seti ile karşılaştırılması sonucu elde edilen FID değerlerini gösteren grafik verilmiştir.

Grafikte yatay eksen de epok düşey eksen de de FID değerleri okunabilir.



**Şekil 3:** DCGAN çıktısının Frechet Inception Score ile değerlendirilmesi.

Epok her bir eğitim basamağını ifade etmektedir. Tüm veri seti belli küme büyüklükleri ile belirli bir iterasyonda eğitilerek bir epok eğitimi tamamlar. Yani tüm veri setinin algoritmaya eğitilmesi 1 epok karşılığına



denk gelir. DCGAN' in eğitildiği bu otonom mimari plan üretimi deneyinde 15000 epok kadar eğitim sürdürülmüştür. Grafiğin yukarısında da DCGAN üretimi görsellerin epoklar sürecinde gelişimini göstermektedir. En baştaki görsel, veri setindeki gerçek veriyi ifade etmektedir. FID skoru grafiğine bakıldığında başlangıç epok ve sonuç epok arasında 11000. ve 15000. Epok arası dışında, sürekli bir azalma gözlenebilir. Ayrıca ilk epok ile son epok arasındaki değişim yaklaşık 16,5 birim kadardır. Bir diğer yandan epok artışı ile birlikte hem mekansal karşılıklarının olduğu hem de çözünürlük olarak daha kaliteli sonuçların son epokta elde edilebildiği gözlemlenebilir. 11000. epok sonrasındaki FID skorundaki artış, eğitimde veri setinin öznitelik dağılımındaki problemler nedeniyle olma ihtimali oldukça fazladır. Bir diğer yandan eğitimin 11000. epokta en verimli halini aldığı gözlenebilir. Bu da eğitimi 11000. epokta durdurmanın eğitim için yeterli olduğunu gösterir. Bu sonuçtan yola çıkarak FID skoru ile de eğitimin verimliliğinin her zaman daha fazla epok eğitiminin olması ile doğru bir orantısının olmadığı gösterilebilir. Ancak tüm bu değerlendirme sonrasındaki genel sonucumuz DCGAN'ın çoğunlukla doğru yönde eğitiminin devam ettiği ve veri setini doğru şekilde genelleyerek veri setine benzer üretimleri neredeyse iyi bir çözünürlük kalitesinde üretebildiği sonucuna ulaşabiliriz. Daha iyi bir özniteliksiz dağılıma sahip veri seti ile DCGAN eğitiminde daha iyi bir başarımla alınabilse de DCGAN'ın mimari plan üretiminde verimli bir öğrenme süreci gerçekleştirdiğini bu verimlilik nedeniyle de dolaylı olarak mimari plan şemalarının uygun üretilmeye başlandığı söylenebilir.

Burada problemler olarak görülen, değerlendirme sonucu FID skoru üzerinden dolaylı olarak mimari plan üretimlerinin değerlendirilmiş olmasıdır. Sonuçta FID skoru GAN algoritmasının çalışma verimini matematiksel olarak doğru olduğunu çözümlenmiş olsa bile üretilen mimari planların mekansal organizasyonları ile ilgili bir bilgi vermemektedir. FID değeri mekansal dizilim ve mekanların fonksiyonları üzerine bir değerlendirme sonucu veremeyeceği için mimari plan organizasyonu değerlendirilmesinde işlevsiz bir değerlendirme yöntemi olacaktır.

Nicel yöntemlerle GAN algoritması değerlendirmesinin özellikle mimari plan görseli üretiminde kritik olarak ele alınması gereken bir yöntem olduğu düşünülmektedir. Bu nedenle bir sonraki bölümde nicel yöntem

değerlendirmesinin yanında nitel bir değerlendirme yöntemi de denenmiş ve çıktıları değerlendirilmiştir.

### **3.2 Hızlı Sahne Sınıflandırması Yöntemi ile Değerlendirme**

GAN algoritması nitel değerlendirme yöntemleri arasında Hızlı Sahne Sınıflandırması (Rapid Scene Categorization) bulunmaktadır. Bu yöntem anket tabanlı bir yöntemdir. Hızlı sahne sınıflandırması insan gözlemcinin hızlı şekilde karar alarak sınıflandırma yapıyor olmasıdır. Hızlı sahne sınıflandırmasında gerçeklik değerlendirmesi yapılmaktadır. Gerçeklik değerlendirmesi, biri veri setinden diğeri ise GAN üretiminden iki görsel çıktının hangisinin gerçek olduğunun sorulması ve anket katılımcısı tarafından yanıtlanması ile gerçekleşir. Eğer katılımcı çoğunlukla GAN üretimlerini gerçek olarak sınıflandırırsa GAN algoritmasının verimli olduğu söylenebilir.

DCGAN ile üretilen Palladyan plan şeması üretimlerinin değerlendirilmesinde kullanılan nitel yöntem hızlı sahne sınıflandırması olmuştur. Hızlı sahne sınıflandırmasında, gerçek olan verinin bulunması hedeflenir. Ancak mimari plan şeması verisinde hangisinin gerçek hangisinin gerçek olmayan veri olduğunu belirleme durumu mimari planın bir özneliğini belirleyen bir seçim olmayacaktır. Bu nedenle seçilen soru gerçeklik özelliği üzerinden değil hangi plan organizasyonunun daha iyi olduğu sorusu olmuştur. Böylece anket katılımcıları GAN üretimlerinde plan organizasyonunu değerlendirmeye başlayacaklardır. Bu GAN algoritmasının verimliliğinden üretilen mimari plan şemalarının durumunun değerlendirilmesini sağlayacaktır.

Hızlı Sahne Sınıflandırması yapılırken **Şekil 1** ve **Şekil 2**'deki hem veri setinden hem de DCGAN üretimi 10'ar adet plan şeması görselinden oluşan on soruluk bir anket çalışması hazırlanmıştır. Anket SurveyMonkey üzerinden online gerçekleştirilmiştir. Her soruda bir DCGAN bir de GRAPE-SGI üretimi Palladyan plan şeması bulunmaktadır. Anket katılımcısı bu iki plan şeması üzerinden hangisinin daha iyi bir mekansal organizasyon sunduğunu seçmektedir. Katılımcı anket boyunca hangi verinin DCGAN, hangi verinin GRAPE-SGI üretimi olduğunu bilmemektedir.

**Şekil 4** SurveyMonkey'de oluşturulan anket arayüzü ve bir örnek soruyu göstermektedir.

## Plan Şeması Düzeni Değerlendirmesi / Plan Scheme Layout Evaluation

### Plan Şeması Düzeni Değerlendirmesi / Plan Scheme Layout Evaluation

TR

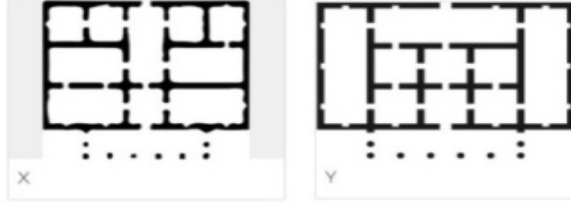
Bu ankette katılımcılardan aşağıdaki plan şemalarından hangilerinin daha doğru bir mimari plan düzenine sahip olduğunu işaretlemeleri beklenmektedir.

EN

In this survey, participants are expected to mark which of the following plan schemes have a more accurate architectural plan configuration.

Tamam

\* 6. Hangi plan düzeni daha iyidir? / Which plan configuration is better?

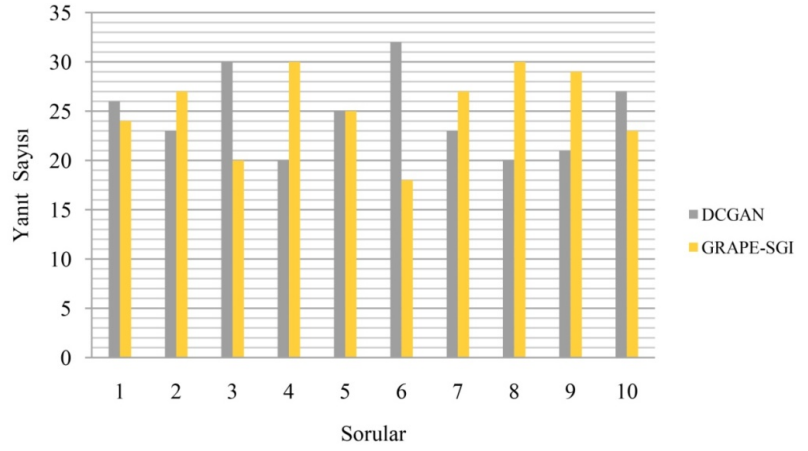


50 mimar katılımcı ile gerçekleştirilen bu ankette 10’ar plan şeması ile oluşturulan sorular üzerinden toplamda 500 farklı yanıt elde edilmiştir. Şekil 5’ te sorulara verilen yanıtların sayısal olarak dağılımı görülebilir.

Hızlı sahne değerlendirme sonucuna göre, Şekil 5’ teki grafikten de anlaşılacağı üzere sorulan 10 soru içinde 4 tanesinin DCGAN yanıtları GRAPE-SGI üretimi veri seti yanıtlarından daha fazla olmuştur. 1 soruda eşit sayıda yanıt alınmış ve diğer 5 soruda da GRAPE-SGI üretimleri daha yüksek sayıda yanıt almıştır. Ancak grafik üzerinde DCGAN ve GRAPE-SGI seçimleri her bir soru üzerinden incelendiğinde hiç bir soru için büyük bir fark olmadığı da okunabilir. En büyük sayısal farkın ise DCGAN üretimlerinin 14 yanıt fazla olduğu 6. sorudur.

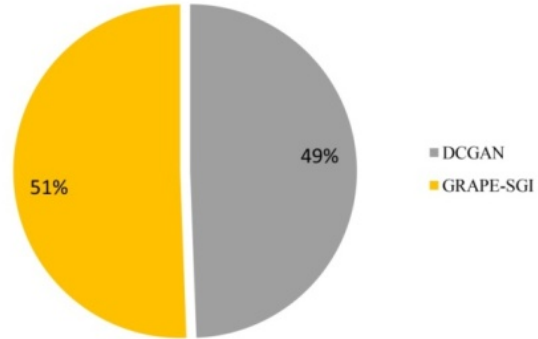
Şekil 4: Hızlı Sahne Sınıflandırması Anket Arayüzü

Şekil 5: Anket Yanıt Sayıları Grafiği



Şekil 6'da hızlı sahne sınıflandırması yanıtlarının yüzde dağılımı gösterilmiştir. GRAPE-SGI yanıtların %51'ini, DCGAN ise % 49'unu almıştır. Böylece katılımcılar plan düzenlemesi bakımından veri seti ve DCGAN üretim çıktılarının plan şeması organizasyonunu benzer düzeyde bulduğu söylenebilir.

Hızlı Sahne Sınıflandırması Yüzdeleri



Şekil 6: Anket Yanıt Yüzdeleri

Kalitatif yöntemlerden hızlı sahne sınıflandırmasına göre DCGAN üretimleri veri seti ile benzer bir mimari plan şeması organizasyonuna sahiptir. Ancak bu çalışmanın net bir sonucunun olabilmesi katılımcı sayısının ve soru sayısının daha fazla olması ile mümkün olabilir. Nitel yöntemlerin dezavantajlarında bildirilen katılımcı maaliyeti bu durumla ilgilidir. Bir diğer yandan katılımcıların seçimlerinde öznel yargılarının bulunması nedeniyle seçimlerinin değişiklik gösterebileceği de unutulmamalıdır. Hızlı sahne sınıflandırmasının diğer bir dezavantajı ise bu değerlendirme yönteminin verinin gerçekliği üzerine kurgulanıyor olmasıdır. Ancak mimari plan şemasında bir gerçeklik sorusunun doğru sonuca götüreceği bir soru olmaması, başka bir deyişle gerçeklik kavramının mimari plan şeması özneteliği olmaması nedeniyle soru

değiştirilmiş ve hangi plan organizasyonunun daha iyi bir mekansal kurguya sahip olduğu sorulmuştur. Bu sorunun da kendi içinde veri setinin ve üretimlerin grid tabanlı üretimler olması nedeniyle problemler yanları olabilir. Ancak yine de bu soru DCGAN üretim verimliliğinden üretilen plan şemalarının karşılaştırılarak değerlendirilmesi bakımından nicel yöntemle göre daha faydalı sonuçlar verdiği söylenebilir.

#### 4. SONUÇ

Bu metinde, GAN algoritmasının değerlendirme yöntemlerinin, GAN algoritması ile üretilen mimari plan şemalarının değerlendirilmesinde kullanımının verimliliği tartışılmıştır. GAN algoritması değerlendirme yöntemleri her ne kadar literatürde kabul görmüş olsa da mimari plan şemalarından oluşan bir veri seti ile eğitim çıktılarında da aynı değerlendirme yöntemlerinin kullanılıp kullanılmaması cevaplanması gereken bir sorudur.

Borji'nin (2019) sınıflandırmasını yaptığı GAN algoritması nitel ve nicel değerlendirmesi yöntemleri GAN algoritmasının çalışma verimliliğini ölçerek özellikle nitel yöntemlerde çoğunlukla dolaylı yoldan üretim çıktıları değerlendirilmektedir.

GAN algoritması değerlendirme yöntemi genel olarak üretim çıktılarının çözünürlük, netlik, gerçeklik ve hedef görsele benzerlik değerleri üzerinden bir analiz yöntemi listesi sunmaktadır. Ancak mimari plan üretimlerinde çözünürlük, netlik gibi özellikler mimari planın bir özelliğini göstermemekte, yalnızca mimari görselin temsil kalitesini açıklayabilmektedir.

Mimari planın *gerçeklik* niteliği bulunmamaktadır. Bu açıklama GAN algoritması tarafından üretilen bir kedi görseli için yapılabilmeye mümkündür. Herhangi bir GAN algoritmasının doğru bir şekilde üretmediği kedi görseli gerçek bir kediyi temsil edemeyecektir. Ancak bir mimari plan için gerçek bir üretim ya da gerçek olmayan bir üretim denmesi mümkün olmayacaktır. Bu nedenle Hızlı Sahne Sınıflandırmasında hangi plan şeması gerçektir sorusu yerine hangi plan şeması organizasyonu daha iyi çözülmüştür sorusu katılımcılara sorulmuştur. Bu durum GAN algoritması nitel değerlendirme yöntemlerinin mimari plan üretimi değerlendirilmesinde yeni tür değerlendirme yöntemlerine ihtiyaç duyulduğunu göstermektedir.

GAN algoritması değerlendirmesindeki diğer bir nitelik ise *benzerliktir*. Benzerlik GAN algoritmasının çalışma verimliliği ile ilgili olup GAN algoritmasının veri setine birebir benzer bir görsel üretmesi problemli olarak görülmektedir. Bu nedenle veri setindeki mimari planın birebir aynı şekilde üretimi sorunlu olacağı gibi, bir mimari planın veri setindeki mimari plan ile benzer olup olmadığının belirlenmesi de diğer bir problemdir. Üretimi yapılan mimari planın benzerliği ürettiği stil üzerinden mi, mekan kurgusu mu ya da sadece morfolojik bir benzerlik ile mi değerlendirilmesi gerektiği önemli bir soru haline gelir. GAN ağının bir kedi görseli üretimi yapması halinde o görselin veri setindeki kedi görsellerine ne kadar benzediği kolayca anlaşılabilir. Mimari bir plan için benzerliğin hangi kavramlar üzerinden açıklanacağı bir değerlendirme sorunu olarak görülmektedir.

Frechet Inception Distance yöntemi ile nicel olarak GAN algoritması verimliliği değerlendirmesi veri seti ile çıktılar arasındaki benzerlik yakınlığına bakmaktadır. Ancak FID tamamen matematiksel bir işlem ile mimari plan şeması üretimlerindeki özniteliklerin vektörel olarak mesafelerine bakmaktadır. Bir görselin özneliği olarak piksellerin renk değerlerinden oluşan bu vektörler FID skorunda önemli bir girdi olmasına rağmen, mimari plan şeması öznitelikleri içinde değildir. Bu nedenle aslında mimari plan şeması sadece piksel dağılımı benzerliği üzerinden değerlendirilmiş, mekansal organizasyon üzerinden bir değerlendirme yapılmamıştır. FID skoru, GAN algoritması ile mimari plan şeması üretimi çıktılarının mekansal organizasyonunun değerlendirilmesinde işlevsiz kalmıştır.

Hem nitel hem nicel değerlendirmeler sonucunda GAN ile mimari plan şeması üretimlerinin değerlendirilmesinde, FID skorunun ve hızlı sahne sınıflandırmasının da kesin bir sonuç alabilecek bir yöntem olmadığı gözlemlenmiştir. Bu değerlendirmenin doğru yapılması halinde algoritmanın mimarisi de doğru şekilde revize edilebilir ve otonom mimari plan üretimi çalışmaları da daha verimli hale gelebilir. Sonuç olarak GAN değerlendirilmesi çalışmalarında, mimari plan şeması üretimi değerlendirilmesi özelinde yeni yöntemlerin geliştirilmesinin oldukça önemli bir araştırma alanı olduğu söylenebilir.

## Referanslar

Borji, A. (2019). Pros and Cons of Gan Evaluation Measures. *Computer Vision and Image Understanding* 179, 41-65. <https://doi.org/10.1016/j.cviu.2018.10.009>.

Brownlee, J. (August 30, 2019). *How to Implement the Frechet Inception Distance (FID) for Evaluating GANs* [Web blog]. Eriřim tarihi: 20-11-2019, eriřim adresi: <https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/>

Chaillou, S. (2019). *AI+ Architecture: Towards a New Approach* [This Project]. Harvard University Graduate School of Design.

Grasl, T. (t.y.). GRAPE For Web - Shape Grammar Interpreter. Eriřim tarihi: 20-11-2019, eriřim adresi: <http://grape.swapzt.com/App/PalladianGrammar>.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., & Bengio, Y. (2014). Generative Adversarial Nets. *In Advances in Neural Information Processing Systems* (pp. 2672-2680). Arxiv: 1406.2661. Eriřim tarihi 20-11-2019, eriřim adresi: <https://arxiv.org/abs/1406.2661>.

Güzelci, O. Z., Alaçam, S., & Güzelci, H. (2019). Trend Topics and Changing Concepts of Computational Design in the Last 16 Years-A Content Analysis. *In Proceedings of the 37th Annual Conference of Education and research in Computer Aided Architectural Design in Europe (eCAADe) 1* (pp. 423-430). Porto, Portugal.

Huang, W., & Zheng, H. (2018). Architectural Drawings Recognition and Generation Through Machine Learning. *In Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA)*. Mexico City, Mexico.

Uzun, C., Çolakođlu, M. B., & İnceođlu, A. (2020). GAN as a Generative Architectural Plan Layout Tool: A Case Study for Training DCGAN With Palladian Plans, and Evaluation of DCGAN Outputs. *ITU A/Z 17 (2)*, 185-198. doi: 10.5505/itujfa.2020.54037.

Url-1: <[https://en.wikipedia.org/wiki/Central\\_limit\\_theorem](https://en.wikipedia.org/wiki/Central_limit_theorem)>, eriřim tarihi: 4-12-2019.

