

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YENİ FENOMEN ALGORİTMALAR:
ÇEKİŞMELİ ÜRETKEN AĞLARIN MİMARLIKTAKİ POTANSİYELLERİ
ÜZERİNE BİR ARAŞTIRMA**

YÜKSEK LİSANS TEZİ

Ruşen EROĞLU

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

TEMMUZ 2021

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ

**YENİ FENOMEN ALGORİTMALAR:
ÇEKİŞMELİ ÜRETKEN AĞLARIN MİMARLIKTAKİ POTANSİYELLERİ
ÜZERİNE BİR ARAŞTIRMA**

YÜKSEK LİSANS TEZİ

**Ruşen EROĞLU
(523171010)**

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Prof. Dr. Leman Figen GÜL

TEMMUZ 2021

İTÜ, Lisansüstü Eğitim Enstitüsü'nün 523171010 numaralı Yüksek Lisans Öğrencisi Ruşen EROĞLU, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “YENİ FENOMEN ALGORİTMALAR: ÇEKİŞMELİ ÜRETKEN AĞLARIN MİMARLIKTAKİ POTANSİYELLERİ ÜZERİNE BİR ARAŞTIRMA” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. Leman Figen GÜL**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Doç. Dr. Sevil YAZICI**
İstanbul Teknik Üniversitesi

Doç. Dr. Güzden VARİNLİOĞLU
İzmir Ekonomi Üniversitesi

Teslim Tarihi : 09 Haziran 2021
Savunma Tarihi : 13 Temmuz 2021





Yeğenlerime,



ÖNSÖZ

Yüksek lisans sürecimin başından sonuna rehberliği, gösterdiği anlayışı; yürüttüğümüz araştırmanın gelişmesini sağlayan desteği ve yorumları için saygıdeğer danışmanım Leman Figen Gül'e, çalışmanın devamlılığına önemli katkılar sunan değerli jüri üyelerim Sevil Yazıcı ve Güzden Varinlioğlu'na, beni bu algoritmalarla tanıştıran, bildiği her şeyi sorgusuzca aktaran Özgün Balaban'a çok teşekkür ederim.

Bu yolculukta bahsettiğim algoritmalarla olan çalışmalarımda ilk oyun arkadaşım Dilan'a, teze başlamamdaki etkin rolü ile Müge'ye, sunum becerilerimi geliştirerek beni sabırla dinleyen Gizem'e ve yolculuğumun çeşitli bölümlerinde yanımda olan tüm sevdiklerime teşekkürü borç bilirim.

Beni bu günlere taşıyan canım aileme, hayatım boyunca arkamda oldukları ve karşılaştığım zorlu yolları aşmamı sağlayacak gücü sevgiyle verdikleri için sonsuz teşekkür ederim.

Temmuz 2021

Ruşen Eroğlu



İÇİNDEKİLER

Sayfa

ÖNSÖZ	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ	xv
ŞEKİL LİSTESİ	xvii
ÖZET	xix
SUMMARY	xxi
1. GİRİŞ	1
1.1 Araştırmanın Motivasyonu	2
1.2 Araştırmanın Amacı	2
1.3 Yöntem & Kapsam	3
1.4 Araştırmanın Akış Diyagramı	4
2. VERİ BİLİMİ VE VERİ BİLİMİNİN MİMARLIK PERSPEKTİFİ	5
2.1 Veri Biliminin Kronolojisi	5
2.2 Veri Biliminin Mimarlıktaki Yansıması	8
3. ÇEKİŞMELİ ÜRETKEN AĞLAR	13
3.1 Üretici Model (Generator-G)	14
3.2 Ayırt Edici Model (Discriminator-D)	14
3.3 Aktivasyon Fonksiyonları ve Optimizasyonlar	15
3.4 Evrişimli Katmanlar ve Kullanılan Temel Kavramlar	17
4. ÇEKİŞMELİ ÜRETKEN AĞLAR İLE DENEYLER	21
4.1 Veri Seti ve Düzenlenmesi	22
4.2 Deney Ortamı	23
4.3 Seçilen Modeller	23
4.3.1 DCGAN	23
4.3.2 Pix2Pix	24
4.3.3 CycleGAN	26
4.3.4 StyleGAN	26
4.4 Deneyler	28
4.4.1 DCGAN deneyleri	28
4.4.2 Pix2Pix deneyi	33
4.4.3 CycleGAN deneyleri	33
4.4.4 StyleGAN deneyleri	35
5. BULGULAR VE TARTIŞMA	39
5.1 Bulgular; Keşifler	39
5.2 Tartışma; Potansiyeller	42
6. SONUÇ VE ÖNERİLER	45
KAYNAKLAR	49
ÖZGEÇMİŞ	53



KISALTMALAR

CAD	: Computer Aided Design
CAM	: Computer Aided Manufacturing
CNN	: Convolutional Neural Network
CPU	: Central Processing Unit
DCNN	: Deep Convolutional Neural Network
DCGAN	: Deep Convolutional Generative Adversarial Network
GAN	: Generative Adversarial Network
GPU	: Graphical Processing Unit
MIT	: Massachusetts Institute of Technology
ReLU	: Rectifier Linear Unit
TanH	: Hiperbolik Tanjant



SEMBOLLER

b	: bias değeri
D	: ayırt edici (discriminator) fonksiyonu
G	: üretici (generator) fonksiyonu
G(z)	: üretici (generator) çıktısı
w	: ağırlık (weight)
x	: girdi (input)
y, F(x)	: x fonksiyonu çıktısı
z	: rastlantısal gürültü (random noise)



ÇİZELGE LİSTESİ

Sayfa

Çizelge 5.1 : Deneylerin detay çizelgesi.....	40
Çizelge 5.2 : Deneylerin keşif çizelgesi.	42
Çizelge 5.3 : Deneylerin potansiyel çizelgesi.....	43





ŞEKİL LİSTESİ

Sayfa

Şekil 1.3 : Durum çalışması akış diyagramı.	4
Şekil 2.1 : Yapay zeka ve alt kümeleri (referanstan alınarak yazar tarafından çizilmiştir; Nilsson, 2010).	5
Şekil 2.2 : Yaşam ve ölüm kurallı bir hücre özdevinimi (Kirke & Miranda, 2010). ...	8
Şekil 2.3 : ÇÜA modellerinin kullanımıyla oluşturulan plan şeması çalışmaları (a) (Huang & Zheng ,2018) (b) (Chaillou, 2019) (c) (Uzun, 2020).	10
Şekil 3.1 : ÇÜA algoritma çalışma prensibi (referanstan alınarak yazar tarafinfan çizilmiştir; Silva, 2018).	13
Şekil 3.2 : (a) Sigmoid (b) TanH (c) ReLU (d) Leaky ReLU fonksiyonları grafikleri (URL2).	16
Şekil 3.3 : Evrimsel katman kernel filtre ile aktarım diyagramı (referanstan alınarak yazar tarafından çizilmiştir; URL4).	18
Şekil 3.4 : (a) stride değeri 1 (b) stride değeri 2 tarama aşaması ve çıkış matrisleri.	19
Şekil 3.5 : (a) padding değeri 1 olan katmanda filtre (b) nitelik haritası.	19
Şekil 3.6 : (a) girdi imajı (b) nitelik haritası (c) imaja uygun ağırlıklı kernel filtreler (URL4).	20
Şekil 4.1 : DCGAN algoritma çalışma yapısı (URL5).	24
Şekil 4.2 : Pix2Pix, cGAN algoritma çalışma prensibi.	25
Şekil 4.3 : Pix2Pix, U-Net çalışma yapısı (Isola ve diğ., 2017).	25
Şekil 4.4 : (a) CycleGAN çalışma prensibi (b) CycleGAN çift yönlü iş akışı (Zhu ve diğ., 2017).	26
Şekil 4.5 : (a) geleneksel ağ (b) StyleGAN çalışma prensibi (Karras ve diğ., 2019).	27
Şekil 4.6 : DCGAN ilk deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.	28
Şekil 4.7 : DCGAN ilk deney veri seti ve epok 50000 çıktıları.	29
Şekil 4.8 : DCGAN ikinci deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.	30
Şekil 4.9 : DCGAN ikinci deney veri seti ve epok 100000 çıktıları.	30
Şekil 4.10 : DCGAN üçüncü deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.	31
Şekil 4.11 : DCGAN üçüncü deney veri seti ve epok 100000 çıktıları.	32
Şekil 4.12 : DCGAN üç deneyi, epok değerlerine göre karşılaştırması.	32
Şekil 4.13 : Pix2Pix deneyi (a) kısmi etiketli eşlenmiş veri seti (b) epok aşamaları.	33
Şekil 4.14 : CycleGAN ilk deney (a) birinci küme (b) ikinci küme (c) epok aşamaları.	34
Şekil 4.15 : Pix2Pix, CycleGAN karşılaştırması.	34
Şekil 4.16 : CycleGAN ikinci deney (a) birinci küme (b) ikinci küme (c) epok aşamaları.	35
Şekil 4.17 : StyleGAN ilk deney (a) veri seti örnekleri (b) çıktılar, son döngü 1185 epok (c) epok aşamaları (soldan sağa düzenli artmaktadır).	36

Şekil 4.18 : StyleGAN ikinci deney (a) başlangıç imajları; ahameniş ilk deney çıktıları (b) geçiş dönemi (c) çıktılar, son döngü 890 epok (d) epok aşamaları soldan sağa; Ahameniş altlığına Bauhaus stiline uygulanması (e) epok aşamaları (soldan sağa; Bauhaus geçiş stiline Paladyan stiline uygulanması).....	37
Şekil 4.19 : StyleGAN üçüncü deney (a) veri seti örnekleri (b) çıktılar, son döngü 1180 epok (c) epok aşamaları (soldan sağa düzenli artmaktadır).....	38
Şekil 5.1 : DCGAN, StyleGAN karşılaştırması.....	41
Şekil 5.2 : StyleGAN üçüncü deney (epok değerleri imajların altındadır).....	44



YENİ FENOMEN ALGORİTMALAR: ÇEKİŞMELİ ÜRETKEN AĞLARIN MİMARLIKTAKİ POTANSİYELLERİ ÜZERİNE BİR ARAŞTIRMA

ÖZET

Gün geçtikçe gelişen yapay zeka, hızla geniş bir araştırma alanına dönüşmektedir. Yapılan tez çalışması, işte tam bu noktada; son yıllarda oldukça gündemde olan yapay zeka yöntemlerini kullanarak mimari görsel üretim özelinde potansiyelleri araştıracaktır. Bununla birlikte, çalışmanın bir diğer vurgusu üretken modelleri eğitme sürecinde bilgisayarın görüntüleri nasıl anladığı ve oluşturduğunu anlamamızı sağlayacaktır. Bilgisayarın nasıl gördüğünü anlamak, bunun potansiyellerini keşfetmek; bizi imaj üretiminin bir adım ötesine taşıyacaktır. Bu şekilde bu sistemlerin gelişmesinde rol alma olanağı verecektir.

Tez, sırasıyla; veri biliminin Çekişmeli Üretken Ağlara kadar olan gelişimini açıklamakla birlikte bu gelişmelerin mimarlık disiplinindeki etkilerini anlatmaktadır. Çekişmeli Üretken Ağlar ile mimarlık alanında yapılmış çalışmaların açıklandığı literatürde kullanılan modellerden çeşitli Çekişmeli Üretken Ağların deney için seçilmesine karar verilmiştir. Farklı derecede özniteliklere sahip veri setlerinin, yapısal ve üretim döngüleri ile seçilen dört Çekişmeli Üretken Ağ, tezin ana bölümünü oluşturmaktadır. Bu ağlar ile yapılacak deneyler için kullanılan araçlar; Google Colab bulut ortamı ve Anaconda uygulamasındaki Jupyter Notebook arayüzünde Python programlama dili olarak seçilmiştir. Seçimde kullanılan modellerin bu programlama diline uygunluğu göz önüne alınmıştır. Sonuç olarak; deneyler sonucu üretilen imajların görsel potansiyelleri mimari perspektifte irdelenmiş, bulunan keşiflerden potansiyeller çıkarılmıştır. Bu bakımdan tezin yürütülmesi, denetimsiz bir derin öğrenme modelini andırmaktadır.

Çalışma; açık kaynak paylaşımlı olan DCGAN, Pix2Pix, CycleGAN ve StyleGAN algoritmaları ile yine açık kaynak alınmış üç veri kümesi olan Ahameniş, Bauhaus ve Paladyan tarzı veri setleri çalıştırılarak başlamıştır. Deneylerde ortaya çıkan geri dönüşler ile yeni veri setleri oluşturulmuştur. Böylece deneylerde çeşitlilik sağlanmış, kontrollü değişkenler ile toplam 9 deney organize edilmiştir. Bu dokuz deneyde veri setleri öznitelik farklılıklarına ya da modelin ihtiyaçlarına göre değiştirilmiştir. Çıkan sonuçlar sezgisel olarak dolaylı nitel yöntemle değerlendirilmiştir. Mimarlık disiplinindeki tasarım ve üretim potansiyellerini arayan çalışma, bu gözlemler sonucu mimari görsel üretim anlamında hem stil transferi hem de form üretimi konularında keşifler ve bu keşiflerden potansiyeller açığa çıkarmaktadır.

Deneyler sonucu oluşan çıktılar mimari çizim ve form bulma problemlerine uygun kullanımlar oluşturmuştur. Çekişmeli Üretken Ağların bu alanda kullanılabilecek bir üretken model olmasını doğrulamaktadır. Sonuç olarak; araç olarak seçilen yapay zeka algoritmaları, mimarlık için yeni ilkeler, kurallar ve yollar oluşturma fırsatı vermektedir. Yapay zeka alanında bu algoritmalar gelişirken; buna açık olmak ve kullanım alanlarını ölçmek yeni bir düşünsel bakış açısı getirebilir.



NEW PHENOMENON ALGORITHMS: A RESEARCH BASED ON POTENTIALS OF GENERATIVE ADVERSARIAL NETWORKS IN ARCHITECTURE

SUMMARY

Generative models are rapidly transforming into a broad field of research. At this point, the study investigates generative models in architectural visual production by using artificial intelligence methods, which have been on the agenda in recent years. Also, the deeper promise of this work is to enable us to understand how the computer works and creates images in the process of training generative models. It can be said that understanding how the computer sees and exploring its potential will take us one step beyond generating images. This research has become inevitable for a world where the machine-human relationship is and will become stronger. Because of that reason, Generative Adversarial Networks (GANs) with high production potential, one of the deep learning algorithms, were studied.

In the study, the training principles of these algorithms were explained by measuring the visual generation power of the GANs. The study also aims to create an “information set” by evaluating the resulting images. With this aspect, the embedded theory proceeds with the case study method.

Experiments were set up with four types of GANs selected from the diversity of structural and production cycles and these experiments comprise the core of the thesis. Experiments were implemented in the Google Colab cloud environment, and in the Anaconda application with the Python programming language in the Jupyter Notebook interface. As a result, the visual potentials of the generations were examined from an architectural perspective, and the discovered qualities were extracted. The execution of the thesis also resembles an unsupervised deep learning model. It was seen that the results contain potentials as were conveyed in the literature review on this subject. On the other hand, a rather surprising process was described in terms of the method followed.

First of all, the thesis examines the development of data science methods and presents them in chronological order. According to the literature study investigating the effects of this development on the discipline of architecture; It explains the necessity of studying the deep learning algorithms offered by artificial intelligence by focusing on the studies that have developed in the practice of architecture in the last five years. The study which focuses on the GAN structure works on specialized structures created in different flows of structure and designs suitable experiments for these specialized structures. In the experiments, various visual data were produced by using GANs that models were trained. Thus, diversity was provided in the experiments, and a total of 9 experiments were organized with their variables. In these 9 experiments, datasets were modified according to qualitative differences or the needs of the model. The results were evaluated intuitively with the indirect qualitative method. The study, which seeks

the design and production potentials in the discipline of architecture, reveals certain potentials from these discoveries through the aforementioned observations.

Environmental images of buildings categorized with various architectural styles shared as an open source were chosen as the data set. In this data set, 3 clusters were taken, then since they are pixel-based and suitable for the models' method, the file formats and sizes were changed to be suitable for GANs. In the selection of these 3 data sets, it was taken into account having different details and qualities in terms of content. The Achaemenid style was chosen because it has similar features that are easy to learn and have few semantic differences in appearance. Unlike the Achaemenid style, the Bauhaus style was chosen because of the different quality structures in the data set. The Palladian style was chosen because it has more similar qualities than the Bauhaus style, but has more diverse semantic structures than the Achaemenid. In addition to these three datasets, additional datasets were created with the feedback from the experiments. Three of these experiments were done with DCGAN, one with Pix2Pix, two with CycleGAN, and three with StyleGAN.

It has been observed that DCGAN produces outputs of different qualities. However, it needs a big data source and long training time. As the variety of qualifications increases, education becomes shorter; With the increase in the variety of qualifications, education is prolonged. It is seen that the training time is longer to get good results compared to other models.

Pix2Pix needs training with an extensive dataset labelled in detail. At this point, it can make detailed productions because it makes matched production. However, preparing and training a large labelled data set complicates the process compared to other models studied.

In CycleGAN, quick training can be done in two-cluster transformation problems. It has been observed that even if the amount of data to be used is small, it can give clear results. It has been seen in the second experiment that he responded well to the example creation problems over style.

StyleGAN can produce clear images without style-related noise. When DCGAN and its experiments with the Achaemenid dataset were compared, it was observed that StyleGAN produces clearer images. Although StyleGAN's epoch numbers were low, the training time reaches the same hours as DCGAN. Simulation ability was observed in these epoch stages.

Among the tested models, StyleGAN has a strong capability for image learning and reconstruction simulation. It is good at absorbing the pixel characteristics of images in the input dataset, producing somewhat similar imitation outputs, and even summarizing all the input data into one combined "average output". In addition, it can realize the cross-effect between input data through style mixing or style conversion. As a result of the use of data sets produced with various architectural styles and 3D model images, it is seen that product transformation creates the most stable and ideal product when compared to other output types. On the other hand, it did not produce the expected result by causing a quality loss in the style mixing experiment.

GAN models make many other phenomena found in architecture. Artificial intelligence allows us to reveal the complexities hidden in architectural design. At this stage, it becomes essential to adapt to the developing models and go beyond and solidify the basis of the architectural discourses formed with these models. To create an approach specific to the architectural discipline, one must go beyond accepting

these systems. The development of the systems that will encompass relevant pieces of expertise, in which designers can manipulate each model and thus move back and forth between humans and machines, shows that incredible possibilities will open up for the field of architecture. Data science also offers the opportunity to adopt a new set of research methodologies of the architectural discipline.

As a result, artificial intelligence algorithms are chosen as tools that provide an opportunity to create new principles, rules and paths for architecture. Being open to these algorithms as these algorithms develop in artificial intelligence and measuring their usage areas can offer a new ethos, a valid intellectual anchor to architectural research.





1. GİRİŞ

21. yüzyılda gelişen yapay zeka yöntemleri ve donanımsal iş gücüyle birlikte görüntü işleme problemlerinde önemli bir adım atılmıştır. Bu gelişmelerin bir diğer adımı ise işlenecek büyük veri; sosyal medya ve araçlarının artması takip etmektedir. Özellikle 1950'lerin başlarında Turing ve McCarthy, yapay zeka kavramını ilk defa ortaya atarak bu gelişmeleri hızlandırıcı bir teorik ortam oluşturmuşlardır (Kelleher & Tierney, 2020). Gelişen ileri matematik problemlerini çözmek için kullanılmaya başlanan bilgisayarlar, ilerleyen yıllarda pek çok alana (mühendislik ve mimarlık) hesaplama bilimleri olarak katkı sağlamış; herhangi bir problemin çözümünde, verilerle elde edilen bilginin önemli hale geldiğini göstermiştir. Şimdilerde sıkça duyduğumuz makine öğrenmesi, dijital mecralar, internetin yaygınlaşması, 'büyük veri' nin sınıflandırılması, yorumlanması, işlenmesinin kolaylaşması ve bu uygulamaların bulut ortamlarında erişilebilir olması pek çok disiplinde olduğu gibi mimari tasarım alanında da büyük değişimlere sebep olmaktadır.

Bu değişimler mimari tasarımın temelini oluşturan görsel üretim yöntemlerinin artmasını ciddi ölçüde etkilemektedir. Böylece tasarım alanında önemli bir yer tutan görsel üretim; verinin anlamlı hale ulaşması için kullanılabilecek donanımsal birimlerin ve algoritmaların geliştirilmesi ile önemli potansiyeller barındırmaktadır. Son yaşanan gelişmelerle derin öğrenme algoritmaları; görsel veriyi sınıflandırma, tanıma, renklendirme ve stil transferi gibi birçok görsel veri üretim probleminde çözüm olarak kullanılabilmektedir (Brownlee, 2019).

Özellikle makine öğreniminin dikkat çekici ilerlemesinden yararlanan araştırmacılar, bilgisayar destekli bu teknikler aracılığı ile mimari stilleri analiz etmeye başlamıştır. Zang ve diğ. (2014) çok sınıflı DVM (Destek Vektör Makinesi) ile çoklu mimari yerel öğelerin tanımlanmasını gerçekleştirmiştir. Yoshimura ve diğ. (2019), derin evrişimli sinir ağı (DCNN) modelini eğiterek farklı mimarlara ait mimari tasarımlar için bir sınıflandırma yapmıştır. Goodfellow (2014) derin evrişimli sinir ağlarındaki katmanları kullanarak üretken bir algoritma önermiştir. Bu üretken ağ modeli daha önce kullanılan üretken modeller arasında en başarılı sonucu vermektedir. Araştırma

da çözüm olarak yayınlanan görsel işleme üzerine çalışan algoritmalarından Çekişmeli Üretken Ağları ele almaktadır.

1.1 Araştırmanın Motivasyonu

Yapay zeka araştırmalarının hız kazandığı bu günlerde, bu araştırmaların çeşitli alanlara çoktan nüfuz ettiği ve endüstri genelinde çeşitli zorluklara karşı yeni araçlar ve yöntemler getirme potansiyeli olduğu düşünülmektedir. Yapay zekanın mimarlıkta kullanımı hala ilk günlerinde olmasına rağmen, şimdiden umut verici sonuçlar elde edilmektedir (Chaillou, 2019). Mimarlık pratiğinde ise üretimin büyük çoğunluğunu görsel üretim oluşturmakta ve gelişen bu modellerdeki potansiyeller mimarlıktaki üretim problemlerine cevap olabilmektedir. Bu yüzden bu araştırma yapay zeka yöntemlerinden üretken modeller olarak öne çıkmayı başaran Çekişmeli Üretken Ağlar (ÇÜA/Generative Adversarial Networks: GANs) ile mimari görsel üretim problemlerine odaklanmaktadır. Araştırmanın asıl odağını, motivasyonunu bu algoritmaların mimarlıkta kullanım olanaklarını keşfetmek oluşturmaktadır.

1.2 Araştırmanın Amacı

Mimarlıkta görsel üretim, hesaplamalı tasarım alanında matematiksel veya farklı yöntemler kullanılarak çeşitli kurallandırmalar ile yapılmaktadır. Mevcut veri biliminden faydalanarak oluşturulan tez, yapay zeka yöntemlerinin mimari üretim ile ilgili oluşabilecek potansiyellerini anlamayı hedef almıştır. Özetle, bir zamanlar kalemimiz olan makine, bugün mimari üretimin örüntülerini çıkarmak için kullanılabilir ve uygulanabilir tasarım seçenekleri oluşturmamıza yardımcı olacak şekilde eğitilebilir hale gelmektedir. Bu noktada araştırmada, bu üretimlerdeki örüntünün ortaya çıkmasını sağlayan algoritmaların mimari üretim çeşitliliğindeki olanakların ortaya çıkarılması amaçlanmaktadır.

Araştırmada, açık kullanıma sunulmuş belli veri setleri ve buna ek olarak hazırlanacak yeni veri setleri ile makine öğrenimi algoritmalarından Çekişmeli Üretken Ağ modellerini deneyerek, bu modellerin çıktılarının değerlendirilmesi ile bir durum çalışması yapılacaktır. Bahsedilen derin öğrenme modelleri ile tartışılacak her adım, süreç sonunda insanlar ve makineler arasındaki olası ileri ve geri bildirimli ilişkinin tanımlanmasına altlık oluşturacaktır. Araştırmada, ÇÜA ile yapılan deneyler

düzenlenerek, sonuçlar değerlendirilecek, kullanıcıya çeşitli tasarım seçenekleri arasından yeni imkanlar sunulacaktır. Bu amaçla, veri bilimi tarafından sunulan araçların miktarı ve açık kaynak bulunması, araştırmamız için değerli olmaktadır. Araştırma, tezin yaklaşım potansiyelini kanıtlamak için birçok deney ve araç önermektedir. Bu noktada izlenecek yol; seçilen veri seti ile, makinenin doğru şekilde "öğrenmesini" sağlamak olacaktır.

1.3 Yöntem & Kapsam

Tez; ilk olarak yapay zeka alanı ve yöntemlerinin gelişimini inceleyerek, bu gelişmeleri kronolojik bir şekilde sunacaktır. Bu gelişimin mimarlık disiplinine etkilerini de araştıran literatür çalışması; yapay zekanın sunduğu derin öğrenme algoritmalarının son 5 yılda mimarlık pratiğinde gelişen çalışmalara odaklanarak çalışmanın gerekliliğini açıklamaktadır. ÇÜA yapısına odaklanan çalışma, ÇÜA'nın farklı akışlarda oluşturulmuş özelleşmiş yapılarını kullanarak, bunlara uygun deneyler tasarlanmaktadır. Deneylerde, ÇÜA kullanılarak çeşitli görsel veri üretimi sağlanacak, modellere eğitim yapılacaktır. Deneyler sonucunda, bu algoritmaların mimarlık disiplindeki kullanım olanaklarının genişletebileceği öngörülmektedir. Bu alanın, deneysel olduğu kadar yeni olduğu ve bugüne kadar şaşırtıcı sonuçlar verdiği açıklanmaktadır.

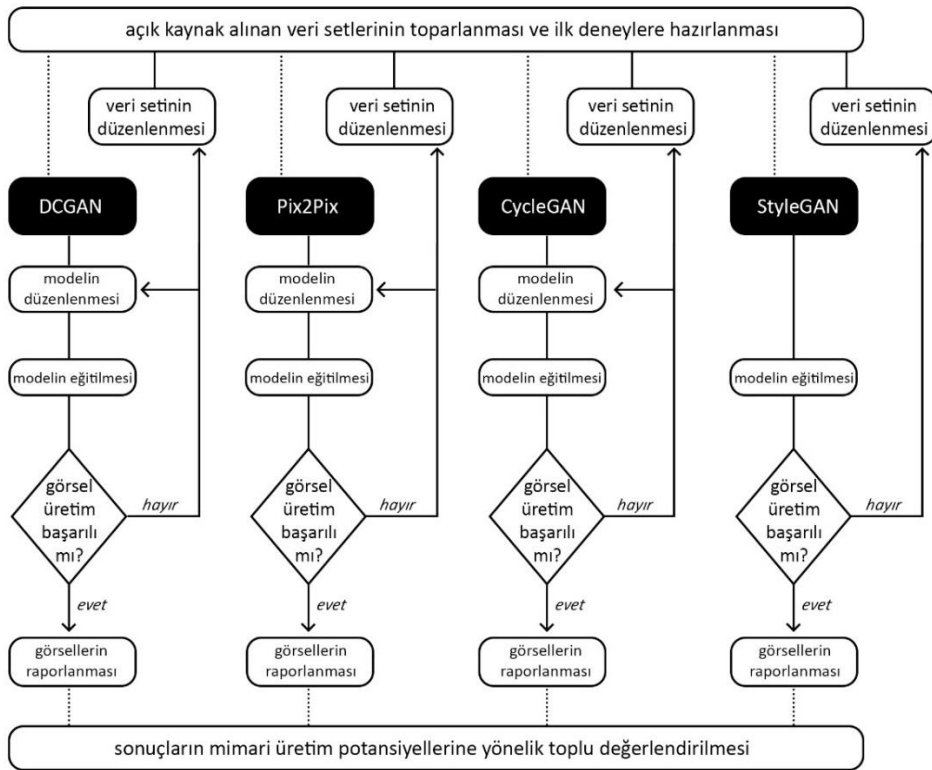
Yürütülen araştırma; yöntem olarak deneysel durum çalışması şeklinde ilerlemektedir. Gömülü teori yöntemi durum çalışmalarının bir alt başlığı olarak değerlendirilip gizlenmiş bilgi setini ortaya çıkarmak, keşfetmek için uygulanmaktadır (Scholz & Tietje, 2002). Araştırma, süreci bu şekilde ele alarak ÇÜA ile ele alınmamış mimari üretim olanaklarını araştırarak ve çalışma prensibini irdeleyecektir. Çalışma, açık kaynak sunulmuş veri ve ÇÜA modellerinin toplanması düzenlenmesi ve gerekli durumlarda yeni veri setlerinin oluşturulması ile başlamaktadır. Düzenlenen ve oluşturulan modellerle deneyler kurgulanmaktadır. Deneylerin sonucu çıkan görseller kalitatif yöntemle değerlendirilecek; bu değerlendirme ile keşfedilmemiş bilgi setleri ortaya çıkarılacak ve bunların potansiyelleri sunulacaktır.

Araştırma, temel anlamıyla mimari tasarım özelinde üretken model çalışmalarında mimari cephe üretimi ve yapay zekanın bir alanı olan makine öğrenmesi

algoritmalarının kesişiminde yer almaktadır. Mimarlık; konu alanı, yapay zeka da yöntem olmak üzere bu iki disiplinin aralarındaki ilişki ve denenerek örneklendirilecek bir dizi deney üzerinden ele alınacaktır.

1.4 Araştırmanın Akış Diyagramı

(1) Seçilen ÇÜA modelleri çalışma prensiplerine göre ilk küme veri seti ile çalıştırılacaktır. (Açık kaynak paylaşılan veri setleri ile ilk deneyler yapılacak ve yeniden çalıştırmadan önce veri setine uygulanması gereken ihtiyaçlar çıkarılacaktır). (2) Bir önceki adımda çıkan ihtiyaçlar ile veri setinde eksiltme, artırma gibi yöntemlerle, çıkacak ürünü iyileştirilmesi için çalışılacaktır. (3) Modeller düzenlenecektir. (Seçilen ÇÜA arasından StyleGAN özelinde açık kaynak paylaşılmış hazır model eğitilecektir.) (4) Modeller eğitilecektir. (5) Eğitim sonuçları değerlendirilecektir. İhtiyaç halinde yeni deneyler oluşturulacaktır (Burada 2. ya da 3. adıma geçilebilir). (6) Üretim sonuçları raporlanacaktır. (7) Sonuçların mimari üretim potansiyellerine yönelik toplu değerlendirilmesi yapılacaktır.



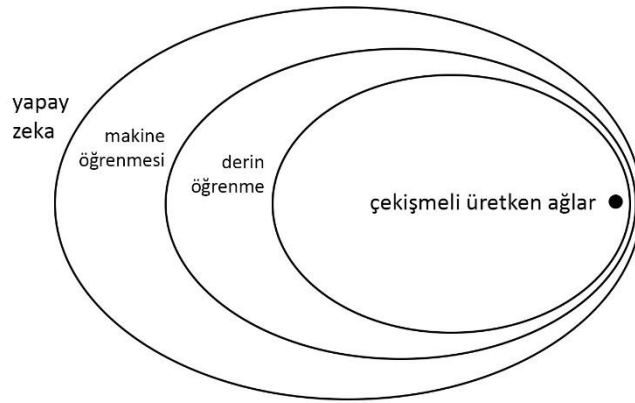
Şekil 1.3 : Durum çalışması akış diyagramı.

2. VERİ BİLİMİ VE VERİ BİLİMİNİN MİMARLIK PERSPEKTİFİ

Çalışmanın literatür incelemesi, veri biliminin gelişmesiyle günümüz algoritmalarına nasıl ulaşıldığını açıklamaktadır. 2.1 bölümünde incelenen modellerin gelişmesiyle etkilenen mimari disiplindeki yeni fikirlerin ortaya çıkması 2.2. bölümünde açıklanmaktadır.

2.1 Veri Biliminin Kronolojisi

Yapay zeka ve öğrenme modelleri, veri biliminin problemlere çözüm olarak oluşturulan sinir ağlarını kapsamaktadır. Bu noktada, ilk çıkışı 1805'te Gauss'un kuyruklu yıldızların yörüngesini bulmak için oluşturduğu en küçük kareler yöntemi olan matematiksel ifade, bugün halen yapay zeka algoritmalarının temelini oluşturmaktadır (Nilsson, 2010). Günümüz derin öğrenme algoritmaları, insan sinir sisteminin çalışma şeklinin odağında, ilişkisel makine öğrenmesi yöntemleri içerisinde değerlendirilmektedir. Çekişmeli üretken ağlar ise bu derin öğrenme yöntemleri kümesinde yer almaktadır (Şekil 2.1).



Şekil 2.1 : Yapay zeka ve alt kümeleri (referanstan alınarak yazar tarafından çizilmiştir; Nilsson, 2010).

McCulloch & Pitts (1943), ilk yapay sinir ağı modelini önermişlerdir. Günümüze uzanan bu ilişkisel yapay zeka yaklaşımında, ilksel olan sinir ağı çalışmasını “A Logical Calculus of the Ideas Immanent in the Nervous Activity” başlığıyla

yayınlamıştır. Bu çalışmada; McCulloch & Pitts, yapay sinir ağı modelini mantıksal adımlarla açıklamaktadır.

Turing (1950) “Computing Machinery and Intelligence” adlı çalışmasında, makineler düşünebilir mi gibi sorular sorulmakta ve yanıtlar aramaktadır. Bu sorular, her ne kadar insan zekasını taklit eden bir zekayı tanımlasa da o dönem işlem gücü zayıf makinelerle yeterli bir çıktı elde edilmesini sağlayamamıştır. Turing’in bu soruları, sorulduğu dönem neticesiyle önemli sayılmaktadır.

Bunların ardından, 1956 yılında, yapay zekanın temellerinin atıldığı Dartmouth Konferansı gerçekleşmiştir. Burada öne çıkan John McCarthy, Marvin Minsky, Claude Shannon, Ray Solomonoff, Alan Newell, Herbert Simon, Arthur Samuel, Oliver Selfridge, Nathaniel Rochester ve Trenchard More isimleri, yapay zekanın kurucuları olarak kabul edilmektedir (Kantayya, 2020). Bu etkinlik ile yapay zekanın gelişmelerine yön verecek önemli adımlar atılmış, pek çok problemin ortaya atılan zeki makineler tarafından çözülebileceğine inanılmıştır (Say, 2018). Solomonoff (1957), konferansın çalışma raporunda da geçen “An Inductive Inference Machine” adlı çalışmasıyla tümevarımsal çıkarım için öneride bulunmuştur. Amerikalı psikolog olan Rosenblatt (1957), aynı yıl ilk yapay sinir ağı birimi olan “perceptron” (algılayıcı) geliştirmiştir. Bu algılayıcı; bir ya da daha fazla girdi, bir işlemci ve bir çıktıdan oluşan basit bir yapı modeli olarak karşımıza çıkmaktadır. Bu yapay zekanın en küçük yapı modelinin formülü aşağıdaki gibidir. Formül 2.1 de görüleceği üzere; x : girdi, w : ağırlık (kuvvet), b : bias değeri ve y : çıktı yani $f(x)$ fonksiyonunu ifade etmektedir. Bu formül ile sinir ağlarının ağırlık değerleri değiştirilebilir niteliğe ulaşmıştır (Nilsson, 2010).

$$y=(w*x+b) \quad (2.1)$$

Minsky & Popert, 1959 yılında perceptronun zeki makineler için tanımlanan doğrusal olmayan problemleri çözmede yeterli olmadığını açıklayan bir çalışma yayınlamışlardır. Donanımsal yetersizlikler, o dönem yapay zekaya ilişkisel yaklaşan bu grup için bir duraksama yaşanmasına sebep olmuştur. Bu noktada, sembolik ve evrimselci yapay zeka çalışmaları hız kazanmış ve uzman sistemler gelişmeye başlamıştır.

1980’lerde nesne tabanlı programlamanın çıkışı, Merkez İşlemci Birimi (Central Processing Unit-CPU) kullanımının artmasıyla, bazı yazılımsal ve donanımsal

problemler çözüme kavuşmuştur. Bu gelişmelerle Rumelhart, Hinton & Williams (1986), makalelerinde geri yayılımlı (back-propagation), hatalarından öğrenen yapay sinir ağı modelini önermişlerdir. Algoritmanın çalışması; nöronların iletiminde yapılan hatalara bakılarak iyileme optimizasyonu yapmaktadır. Bu gelişmeyle, öğrenen sistemler için önemli bir adım atılmıştır.

Vapnik & Cortes (1995) “Support-vector networks” (destek vektör ağları) önerisi ile büyük veriyi sınıflandırmaya yönelik bir çalışma gerçekleştirmişlerdir. Önerdikleri bu ağ, uzamsal boşlukta atadığı özniteliklerle veri kümesini iki kümeye ayırmaktadır. Bu noktada büyük verinin sınıflandırma problemlerine çözüm olacak ağların gelişmesini sağlamaktadır.

LeCun ve diğ. (1998) Evrişimsel Sinir Ağlarını (Convolutional Neural Networks-CNN) kullanarak LeNet adını verdiği çok katmanlı sinir ağı ile desen tanıma çalışması yapmıştır. Kullanılan bu evrişimsel katmanlar sayesinde görsel işleme ve sınıflandırma hızlanmaya başlamıştır. Fakat yeniden donanımsal yetersizlikler, büyük veri ile görüntü tanıma ve işleme problemlerine engel olmaktadır.

1990'larda, Grafik İşlemci Birimi (Graphics Processing Unit-GPU) kullanımı, bilgisayarla görü çalışmalarında yapay sinir ağlarının daha karmaşık veri yapıları üzerinde çalışmasını sağlamaya başlamıştır.

Bu süreçte, internetteki görüntü verilerinin hızla artışı ile görüntüler ve multimedya verileri almak, düzenlemek ve etkileşim kurmak için daha karmaşık ve sağlam modellere ihtiyaç duyulmuştur. Bu tür verilerin tam olarak nasıl kullanılacağı ve organize edilebileceği kritik bir sorun olmaya başlamıştır. Bu noktada; Deng ve diğ. (2009), “ImageNet: A large-scale hierarchical image database” adlı çalışmalarıyla büyük ölçekli bir görüntü ontolojisi olan ImageNet veritabanını sunmuşlardır. Khrizhevsky ve diğ. (2012), “ImageNet Classification with Deep Convolutional Neural Networks” adlı çalışmayla ImageNet’in açtığı yarışma çağrısına yanıt vermiş ve bu veri setinden 1,2 milyon yüksek çözünürlüklü görüntüyü 1000 farklı kümeye sınıflandırma yaparak eğittikleri derin evrişimsel sinir ağlarını yayınlamışlardır. Bu çalışma bu zamana kadar bu veriyi öğrenen algoritmalar arasında en az hata oranını veren araştırma olmuştur.

Goodfellow ve diğ. (2014) “Generative Adversarial Nets” ismiyle yayınladıkları çalışmada, üretken modelleri aynı anda iki model ile eğiten yöntem olarak yeni bir ağ

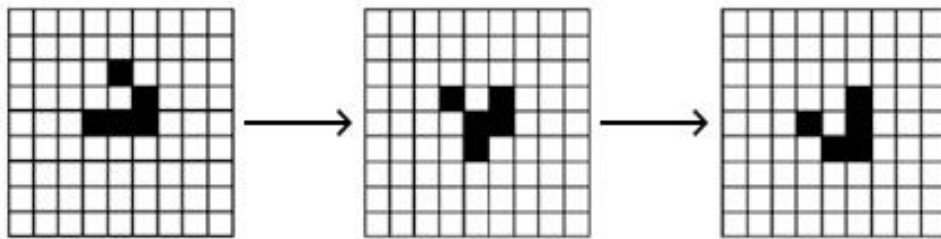
modeli önermişlerdir. Çekişmeli bir süreç ile öğrenen bu iki ağ yapısı; bir öğrenme modeli tanımlaması ile yeni, kullandığı ağ yapısı ile evrimsel sinir ağlarına benzer bir yapıdadır. Modelin yapısı; verilerdeki dağılımını yakalayan üretici bir model (G) ve bir örneğin eğitim verilerinden gelme olasılığını tahmin eden ayırt edici bir model (D) den oluşmaktadır.

Veri biliminin; literatürde yapay zeka alanındaki gelişmeler ve görsel üretim anlamındaki bu başarılar, görsel veri üretimi yapan tüm disiplinleri etkilemektedir. ImageNET çalışmasının başındaki araştırmacılardan Fei-Fei Li de “Görsel zeka, yapay zekanın temel taşıdır.” sözüyle bu noktadaki ilerlemenin temel bir ihtiyaç olduğunu göstermektedir. Bir sonraki başlıkta; bu gelişmelerin mimarlık disiplinine olan etkileri, bilgisayarla görü, görüntü işleme ve sınıflandırma konularında yükselen fenomen ağların kullanımı ve potansiyellerinin mimarlık alanında kullanıma yatkınlığı açıklanacaktır.

2.2 Veri Biliminin Mimarlıktaki Yansıması

Veri biliminin gelişmesiyle mimarlıkta gelişen yeni yaklaşımlar, tezin sorduğu “Makine mimari görüntü, görsel üretim yapabilir mi?” sorusuna odaklı gelişmeler bağlamında incelenecektir.

Yapay zekanın temel konusunun mimarlık konularına paralel ilerlediği düşünülebilir. Veri biliminde gelişen; insan zekası taklit etme, canlılığı taklit etme düşüncesi o yıllarda mimarlıkta da yankı bulmuştur. Neumann (1950) Hücresel Özdevinim teorisini ilk ortaya attığında aslında canlılıktan esinlenen bir ölüm ve yaşam biçimindeki kurallı üretimin temellerini atmıştır (Şekil 2.2). Diğer yandan mimarlık teorilerinde gelişen modülerite yaklaşımı Le Corbusier ve Gropius tarafından ortaya atılmakta, bu gridal büyüme ve oluşum teorik olarak desteklenmektedir.



Şekil 2.2 : Yaşam ve ölüm kurallı bir hücre özdevinimi (Kirke & Miranda, 2010).

Bu noktada, bilgisayar teknolojilerinin donanımsal olarak da gelişmesiyle bilgisayarlar artık mimari üretim için araç haline gelmiştir. 1959'da ilk CAD/CAM yazılımı geliştirilmiş; ardından etkileşimli bir arayüze sahip Sketchpad, Sutherland (1963) tarafından sunulmuştur.

Yine veri bilimindeki gelişmelerle mimarlık teorisyenleri mimari üretimde yeni yaklaşımlar ortaya atmış; Alexander, "Notes on the Synthesis of Form" (1964) ve sonrasında "a Pattern Language" (1968) adlı çalışmalarında, şekil çalışmalarının bilgisayarla kullanım potansiyellerini değerlendirip, bunları açıklamaktadır. Yayınladığı çalışmalarda kurduğu ilkeler bugün hala görsel yazılım programlamanın temel taşı olarak görülebilir. Özyinelemeler, nesne yönelimli programlama ve bunların tasarıma uygulanması gibi kavramlar, ileriye doğru yenilikçi bir hareketi temsil etmektedir.

Diğer yandan Eastman (1969) "Automated Space Planning" çalışmasında yapay zeka problemleri ile plan üretimi problemlerinin benzerliklerini tartışarak plan üretim problemlerini yapay zeka problemlerinin içinde olduğunu savunmuştur.

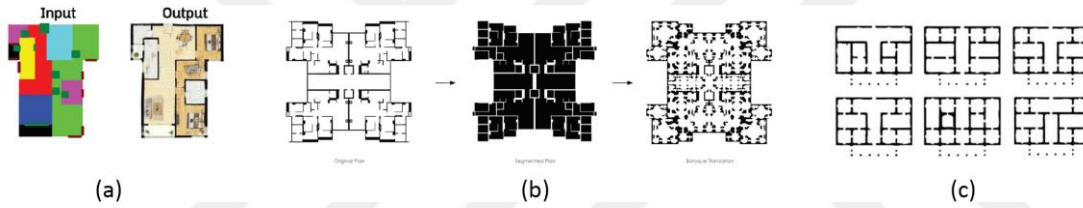
Aynı dönemlerde MIT'deki Mimari Makine Grubu'nun da (The Architecture Machine Group) başında olan Negroponte (1970) "The Architecture Machine" kitabı ile makinelerin yaratıcı süreci ve daha özel anlamda bütün olarak mimari üretimin nasıl geliştirileceğini araştırmıştır.

Veri biliminin donanımsal durgunluk yaşadığı dönemde, mimari üretim pratiğinin yapay zekadan etkileşimi sınırlı kalmaya devam etse de yaşamdan esinlenen, yaşamı taklit etmeye yönelik yaklaşımlar ortaya atılmıştır. Örneğin, Frazer (1995) "an Evolutionary Architecture" çalışmasında evrimsel mimari yaklaşımın yükselişine neden olacak öneriler sunmuştur. Parametrik mimari yaklaşımın da atağa geçtiği bu dönemde Lynn (1999) "Animated Form" çalışmasıyla bunu destekleyecek manifestolara yer vermiştir. Bilişsel süreçlerin bilgi tabanlı bir akıl yürütme ile gerçekleşmesi mimari üretimde çözülmeye çalışılan bir evre olmaya ve bu sürece yardımcı araçlar üretilmeye devam etmiştir. Bu noktada 2007 yılında mimarların kullanımına Grasshopper görsel kodlama programı sunulmuştur.

Diğer yandan görsel okuma ve anlamlandırma konusunda Stiny (2006) görsel olarak algılanan şekiller ve onlara uygulanan çeşitli kural dizileri ile formüle etmiştir. Geometrik imajları biçim grameri ile okuma ve oluşturma yöntemini açıklayarak

bilgisayarla görme noktasında şekilleri tanıma yolculuğu, mimarlık alanında çalışılmaya başlanmıştır. Bu teorilerin yanında veri biliminde yapay zeka yöntemlerindeki hızlı gelişmeler, artık makinenin görmesine ve gördüğünü tanımasına olanak sağlamıştır.

Literatürün ilk kısmında bahsedildiği üzere ardı ardına görsel veri işleme ve sınıflandırma buluşları ile mimarlar tarafından görsel üretim sorularına ve mimari problemlere yanıt aranmaya başlanmıştır. Hesaplama gücündeki yükseliş ve yapay zekâya harcanan fonun hızlı bir şekilde artmasıyla beraber yapay zekânın tasarım gibi karmaşık problemlerin çözümünde kullanılmasıyla oluşan çalışmalar oldukça artmaktadır. Özellikle Google Brain’de araştırmacı olan Ian Goodfellow tarafından 2014 yılında oluşturulan Yapay Sinir Ağlarından Çekişmeli Üretken Ağlar daha önce geliştirilen makine öğrenmesi modellerine kıyasla daha yüksek karmaşıklıkta üretimler yapabilmektedir.



Şekil 2.3 : ÇÜA modellerinin kullanımıyla oluşturulan plan şeması çalışmaları (a) (Huang & Zheng ,2018) (b) (Chaillou, 2019) (c) (Uzun, 2020).

Huang & Zheng (2018) plan çözümünü bir mimari problem olarak ele alıp ÇÜA’nın mimarlık disiplininde kullanımı konusunda ilksel bir çalışma yapmışlardır. Kullandıkları ÇÜA algoritması Pix2Pix adındaki özelleşmiş bir üretken ağ modelidir. Çalışmalarını mutfak planı geliştirmek için kullanıp, deneylerinde veri setlerini Pix2Pix algoritmasına özel etiketlenmiş yapay ve gerçek olmak üzere iki küme ile gerçekleştirmişlerdir.

Campo ve diğ. (2019), mimariye göre yapay zeka ve otomasyonun dönüşüm potansiyelini keşfetmek için mimari çizimlere sanatsal tarzın sinirsel algoritması ile stil aktarımı yapmıştır.

Chaillou (2019) plan şeması problemini ele alıp ÇÜA’yı mimari plan üretim aracı olarak değerlendirdiği çalışması ile ÇÜA’nın bu işlevde kabul edilebileceğini sunmuştur. Burada ÇÜA modellerinin mimariye uygulanması ile stil öğrenimini ve diğer birçok fenomenin mimarlık perspektifinde incelenebileceğini öne sürmüştür.

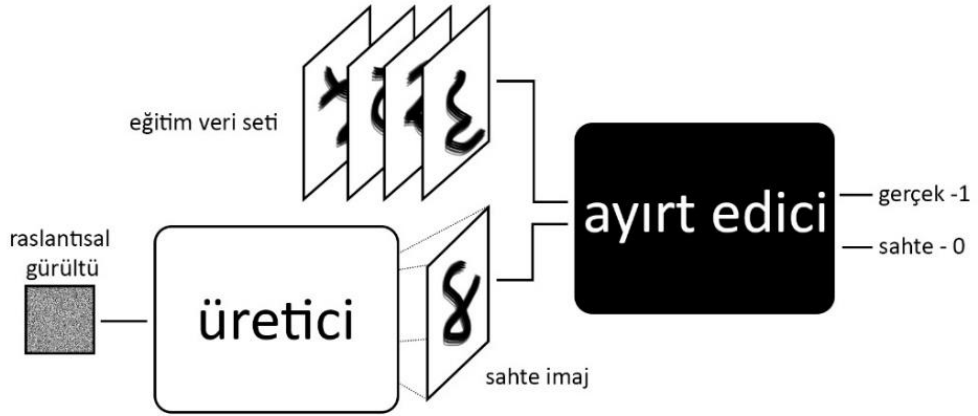
Uzun (2020) ise özelleşmiş bir ÇÜA modeli olan DCGAN ile plan üretimi gerçekleştirmiştir. Tasarım çalışmalarında algoritmanın öğrenme sürecinin ve veri setinin hazırlanmasının öneminden bahsederken çıktıların da değerlendirilmesinin bu süreç kadar önemli olduğunu savunmaktadır. Aslında bu da bize her evresinin ölçülmesi, değerlendirilmesi gereken yeni algoritmaların tasarımı kullanılması ile mimarın yeni alanlarla multidisipliner bir çalışmaya gireceğini söylemektedir. Geleneksel yöntemlerde işlenen ritüelde veri toplamayı, işlemeyi ve sonucu bireysel olarak ele alan mimarlar için yeni bir dönemin başlaması kaçınılmazdır.

Literatürde görüldüğü üzere bu algoritmalar çoğunlukla plan şeması üretimi üzerinde kullanılmış ve farklı üretim yaklaşımlarının denenmeye ihtiyacı olduğu belirtilmiştir. Bu araştırmalar, makine öğreniminin mimari çizimleri öğrenmek ve üretmek için büyük bir potansiyele sahip olduğunu göstermektedir. Ancak hepsi, üç boyutlu mimari mekansal tasarımı keşfetmeden, yalnızca makine öğreniminin mimari tasarıma iki boyutlu düzeyde uygulanmasına odaklanmaktadır. Bununla birlikte, makalelerinin belirttiği gibi, görüntü çözünürlüğü ve hesaplama süresi ile ilgili bazı pratik kısıtlamalarla ilgili hala bir sınırlama vardır ve daha yüksek çözünürlüklü makine öğrenimi yoluyla daha ayrıntılı üç boyutlu form çalışmasının keşfi hala yapılmamıştır. Bu probleme yönelik olumlu değerlendirmelere rağmen halen keşfedilmemiş üretim potansiyellerinin olduğu belirtilmektedir. Mimari üretim anlamında çok çeşitli problemlere yanıt aradığımız bu noktada araştırma, mimari imaj üretimi döngülerine odaklanmıştır. Literatürde yer alan bu modellerden özgün imaj üretimi sağlaması açısından konu özelinde DCGAN, Pix2Pix modellerinin tez kapsamında denenmesine karar verilmiştir. Ek olarak, stil transferinde yüksek çözünürlükte kaliteli sonuçlar oluşturan CycleGAN ve StyleGAN modelleri de deney algoritmalarına eklenmiş, bu modellerin mimari üretim özelinde potansiyelleri irdelenmiştir.



3. ÇEKİŞMELİ ÜRETKEK AĞLAR

ÇÜA, yapay zeka alanında önemli bir araştırma alanı olarak ortaya çıkmıştır. 2014 yılında Goodfellow ve arkadaşları tarafından atılan bu model üretken ağlar içerisinde en başarılı örnek olarak gösterilmektedir. Buradaki fikir, basit bir dağılımdan (rastgele gürültü, Gauss dağılımı gibi) gerçeğe yakın görseller üretecek bir dağılıma dönüşümü (model parametreleri) olarak açıklanabilir. Bu tür modellerin yaratıcı yeteneği, yakın zamanda yapılan çalışmalarla kanıtlanır niteliktedir. Herhangi bir makine öğrenim modeli gibi, ÇÜA da kendilerine sunulan veriler arasından istatistiksel olarak önemli örüntüleri öğrenir. Bununla birlikte, yapıları yeni bir strüktür ile oluşturulmuş olup; iki anahtar modelden, üretici ve ayırt ediciden oluşmaktadır. Bu birbirine zıt çalışan iki sinir ağının birbiri ile rekabetinden faydalanmaktadır (Şekil 3.1).



Şekil 3.1 : ÇÜA algoritma çalışma prensibi (referanstan alınarak yazar tarafınan çizilmiştir; Silva, 2018).

ÇÜA, ilgili görüntüleri üretme yeteneklerini geliştirmek için her iki model arasında bir geri bildirim döngüsünden yararlanır. Ayırt edici, bir dizi veriden gelen görüntüleri tanımak üzere eğitilir. Düzgün bir şekilde eğitilmiş olan bu model, veri kümesinden çıkarılan gerçek bir örneği veri kümesine yabancı bir "sahte" görüntüden ayırt edebilir. Üretici, aynı veri kümesindeki görüntülere benzeyen görüntüler oluşturmak için eğitilir. Üretici görüntüler oluştururken, ayırt edici ona çıktısının kalitesi hakkında bazı geri bildirimler sağlamaktadır. Buna karşılık, üretici daha gerçekçi görüntüler üretmek

için uyum sağlamaya başlamakta, bu geri bildirim döngüsü ile gözlemlenen veriler arasında bulunan özellikleri hesaba katarak, ilgili sentetik görüntüler oluşturmaya geliştirmektedir. Goodfellow (2016) bu geri bildirimli döngüde ÇÜA'nın maksimum benzerlik yöntemi kullanarak üretimler arasındaki benzerliği en yüksek seviyede tutmaya çalıştığını açıklamıştır. Bununla birlikte üretilen çıktının eniyilenmiş hali olması için kayıp fonksiyonu (loss function) değerleri kullanmaktadır. Üretici eğitim görüntüleri üzerinde ayırt edici kaybını hesaplayarak sahte görüntüleri oluşturur. Ayırt edici kaybını sahte oluşturulan görüntülerde hesaplar. Toplam kaybı elde etmek için gerçek ve sahte kayıplar toplanır, ayırt edicinin kayıpları güncellemek için bir optimizasyon adımını geri yayılımla gerçekleştirir ve döngü bu şekilde (maximum likelihood) tamamlanır (URL1).

ÇÜA, yüksek boyutlu olasılık dağılımını verimli ve hızlı şekilde hesaplayabilmesi ile diğer üretken ağlardan kendini ayırarak görsel verideki piksel değerlerinden oluşan matris değişkenleri arasındaki hesaplamalarda oldukça verimli sonuçlar çıkarmaktadır (Goodfellow, 2016).

3.1 Üretici Model (Generator-G)

Üretici model (G) rastlantısal gürültü (z) halindeki veriyi eğitimle sağladığı filtreler (kernel filters) ile gürültüsüz, okunur bir veriye dönüştürmektedir. Oluşturacağı görselin boyutu evrişimli katmanlarda filtre ile kullanılan “stride” ve “padding” değerleri ile belirlenmektedir. Bu veri kümesi $G(z)$ şeklinde ifade edilmektedir. Ayırt ediciye giden bu çıktı, ayırt edici tarafından değerlendirilir. Üretici, bu çıktının değerlendirilmesi ile ayırt edici model aracılığıyla güncellenir. Bu, oluşturulan görüntülerin ayırt ediciye gerçekmiş gibi sunulduğu ve hatanın üretici aracılığıyla geri yayıldığı anlamına gelir. Böylece, üretici, ayırt ediciyi kandırma olasılığı daha yüksek olan görüntüler oluşturmaya doğru gelişme etkisine sahiptir (Brownlee, 2019).

3.2 Ayırt Edici Model (Discriminator-D)

Ayırt edici model (D) hem gerçek veri seti ile eğitilir hem de üreticinin oluşturduğu görselleri analiz eder. Bu nedenle hem gerçek veri setinde bir kayıp oluşturur, hem de üreticinin gönderdiği sahte imaj üzerinden kayıp oluşturur. Bunları değerlendirerek kaybı üreticiye iletir. Böylece gerçek veri seti eğitiminde maksimum benzerlik uyumu

için bu değerler kullanılmaktadır. Ayırt edici gerçek veri seti ile eğitilirken, evrişimli katmanlarda hazırlanmış filtreler ile “stride” ve “padding” değerleri kullanılarak taranır. Bu işlemler sonucunda gerçek görüntüler için ayırt edicinin (D) gerçek görüntüleri, bunların gerçek olduğunu belirten “=1” etiketiyle sınıflandırmasını istenmektedir. Sahte görüntülerin üretici çıktısı olduğunu, “=0” etiketiyle belirterek, sahte imaj “= G(z)” kümesine göndermesi beklenmektedir (Brownlee, 2019).

3.3 Aktivasyon Fonksiyonları ve Optimizasyonlar

Literatürde bahsedildiği üzere bir sinir ağındaki nöron, bir ya da birkaç sayıda girdi ve bir çıktıdan oluşmaktadır. Formül 3.4’te görüldüğü üzere aktivasyon fonksiyonları burada y değerini kontrol edebilmek için kullanılır. Bu şekilde bir nöronun aktif olup olmayacağına karar verilebilir. Bu yönüyle fonksiyonlar derin öğrenme algoritmaları için oldukça önemli hale gelmektedir.

$$y=\text{Aktivasyon}(\sum(w*x+b)) \quad (3.3)$$

Hem üretici hem ayırt edici içinde bulunan katmanlarda yeterli bir y değeri çıkması için çeşitli derin öğrenme aktivasyon fonksiyonları yazılmaktadır. Bu aktivasyon fonksiyonlarındaki kritik nokta türevlenebilir olmaları ve türevi alındığında nasıl bir değer aldıkları ile ilgilidir. Çünkü nöron içinde gerçekleşen karmaşık matematiksel ifadeler türevlenen problemlerdir. Eğer türevlenmiyorsa işlem yapılamaz veya türevi sabit bir değer alıyorsa öğrenme gerçekleşmez. Bu yüzden aktivasyon fonksiyonlarında türevin alacağı değerler kritik hale gelmektedir (URL2). Aşağıda çekişmeli üretken ağlarda kullanılan popüler aktivasyon fonksiyonları açıklanmıştır.

Sigmoid Fonksiyonu: Sürekli ve türevi alınabilir bir fonksiyondur. Doğrusal olmamasından dolayı yapay sinir ağı uygulamalarında sık tercih edilir. Girdi değerlerinin her biri için 0 ile 1 arasında bir değer üretir. Bu yüzden sinir ağlarının çıkış katmanlarında sonuç almak için kullanılır.

$$f(x) = 1 / (1 + e^{-x}) \quad (3.4)$$

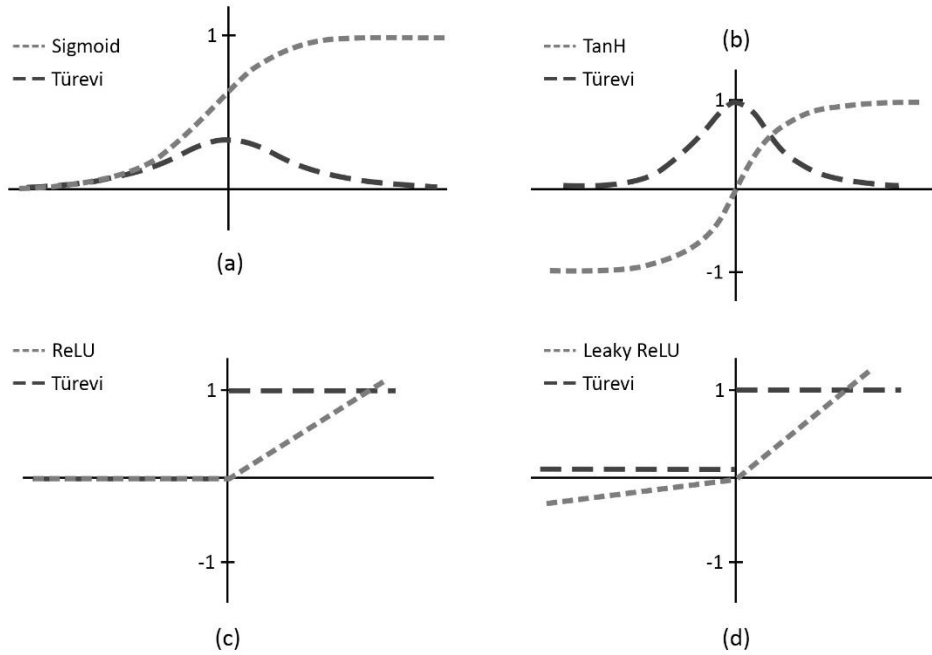
TanH Fonksiyonu: Hiperbolik tanjant fonksiyonu -1 ile 1 aralığında çıktı üretir. Doğrusal olmayan bir fonksiyondur. Sigmoid ile karşılaştırılırsa sıfır merkezli olma sorununu çözer, ondan daha fazla değer alabilmektedir.

$$\tanh(x) = 2 * \text{sigmoid}(2x) - 1 \quad (3.5)$$

ReLU Fonksiyonu: ReLU (rectified linear unit- RELU), özellikle Evrişimli Sinir ağlarında yaygın olarak kullanılan bir etkinleştirme işlevidir. Hesaplaması kolaydır. Tüm negatif girdiler için çıktı sıfır olduğundan bazı düğümlerin tamamen yok olmasına ve hiçbir şey öğrenmemesine neden olur. Bu yüzden fonksiyonu pozitif değerler için tanımlanır.

$$f(x) = x, x > 0; f(x) = 0, x \leq 0 \quad (3.6)$$

Leaky(sızıntı) ReLU Fonksiyonu: ReLU'nun bir çeşididir. Negatif değerler için de değişken bir aralığı vardır, fakat işlem yükü daha fazla olduğu için gerekli yerlerde tercih edilir.



Şekil 3.2 : (a) Sigmoid (b) TanH (c) ReLU (d) Leaky ReLU fonksiyonları grafikleri (URL2).

Swish Fonksiyonu: Google araştırmacıları tarafından önerilen bu fonksiyon girdinin sigmoid fonksiyonu ile çarpımını verir.

$$f(x) = x * \text{sigmoid}(x) \quad (3.7)$$

Maliyet Fonksiyonu: Hata fonksiyonu, eğitim döngüsünün her yinelemesinde hesaplanır. ÇÜA'da iki maliyet hesabı yapılır; hem üreticiden hem ayırt ediciden. Bu ikisi toplam kaybı oluşturur. Maliyet fonksiyonu sinir ağı içerisindeki eniyileyici ile birlikte düşürülmeye çalışılır.

Eniyileyici: Eniyileyici fonksiyon maliyet fonksiyonuna bağlı değişkenleri optimize ederek düşürmeye çalışır. Bu noktada fonksiyonda belirlenen yığın (batch) değeri önemli hale gelmektedir. Kullanılan ağlarda çoğunlukla Adam optimizier seçilmiştir.

Dropout: Eğitim sırasında aşırı öğrenmeyi (overfitting) engellemek kullanılır. Bazı nöronların unutulmasını sağlar. Eğer ağ çok büyükse, çok uzun süre eğitim yapılacaktır ve/veya veri miktarı çok az ise modelin aşırı öğrenme riski vardır. Bundan kaçınmak için kullanılmaktadır.

Yığın normalleştirme (Batch Normalization): Çıktının özellik değerinde normalleştirme yapmaktadır. Bu kullanılan aktivasyon fonksiyonuna yardımcı olarak kullanılabilen bir optimizasyon işlemidir. Katmanlar arasında nitelik farklılıkları normalize ederek özellikleri genellikle yararlı olan kompakt bir alanda toplar. Çıktıların farklı nitelikler sunmasını sağlar.

Örnek normalleştirme (Instance Normalization): Çıktıların yalnızca özelliklerin uzamsal boyutlarında, ancak her kanal (ve her örnek) için bağımsız olarak normalleştirme yapmaktadır. Yığın normalleştirmenin aksine, örnek normalleştirme çıktıların stilini bir hedef stile çevrilmesini sağlar.

Uyarlanabilir örnek normalleştirme (Adaptive Instance Normalization –AdaIN): AdaIN ise bir içerik görüntüsü ve bir stil görüntüsü alır ve bunları birebir eşleştirecek şekilde hizalayarak bir stil transferi sağlar (URL3).

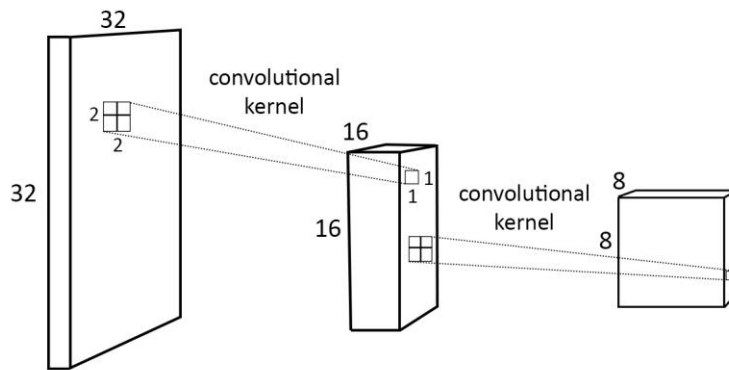
3.4 Evrişimli Katmanlar ve Kullanılan Temel Kavramlar

Evrişimli sinir ağları bahsedilen işlemleri elde etmek için çok katmanlı olarak inşa edilir. Bu katmanlar sinir ağlarının daha iyi çalışmasını sağlayacak şekilde yapılandırılmıştır. ÇÜA'lar da bu katman yapılarından oluşmaktadır.

Evrişimsel Katman: Konvolüsyonel katman olarak da bilinir. Görselin üzerinde gezinen filtre (kernel) ile görsel piksel değerleri bir matris çarpımı oluşturur. Giriş matrisimiz üzerinde filtre sol üst köşeden başlayarak bütün matrisi gezer. Bu işlem sonucunda çıkış matrisini elde ederiz. Bu matrise nitelik haritası (feature map) denir. Bu işlem evrişimsel katmanda gerçekleşir. İşlem sırasında nitelik haritası için y'yi elde edecek en doğru w değeri aranmaktadır. ÇÜA algoritmasında ayırt edicide kullanılan bir katmandır (Sharma, 2018). Oluşacak görselin boyutunu kullanan “strides” ve “padding” değerleri belirlemektedir.

Upsampling : Üretici fonksiyon içinde evrişimsel katmanla birlikte kullanılır. Upsampling eğitimdeki girdi boyutunu iki katına çıkmasını sağlar. Filtre kullanmaz; filtreyi upsampling katmanı sonrasında konvolüsyonel katman kullanılır.

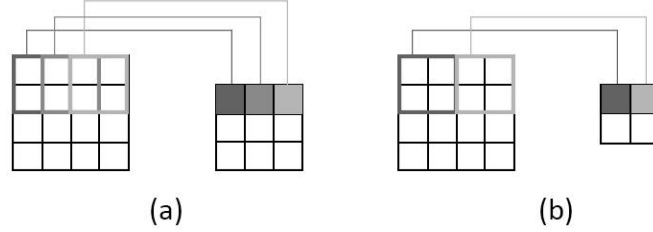
Devrik Evrişimsel Katman: Dekonvolüsyonel katman olarak da bilinir. Üretici modelde kullanılır. Evrişimsel katmanda olduğu gibi filtre ile rastlantısal gürültünün üzerinden hareket etmekte ve çıktı verinin boyutu artmaktadır. Upsampling ve evrişimsel katmanın birlikte yaptığını tek başına yapar. Burada yine stride ve padding değerleri çıktının boyutunu belirlemektedir. Filtre sayesinde verideki rastlantısal gürültü ortadan kaldırılabilir. Bu aşamada eğitim sırasında en doğru ağırlık (w) değerini veren filtreler kullanılır.



Şekil 3.3 : Evrişimsel katman kernel filtre ile aktarım diyagramı (referanstan alınarak yazar tarafından çizilmiştir; URL4).

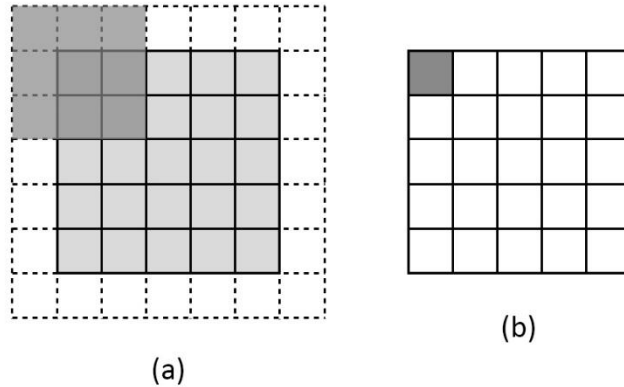
Şekil 3.3’te görülen diyagramda eğitim seti veya üreticiden gelen imajın ayırt edicide evrişimsel katmana kernel filtre ile aktarımı anlatılmaktadır. Bu aktarım “stride” ve

“padding” değerleri verilerek yapılır ve çeşitli filtrelerle nitelik haritalaması yapılarak gerçekleşir. İmajlarda doğru ağırlığı bulan ayırt edici model üreticiden gelen imajı da bu şekilde kontrol eder.



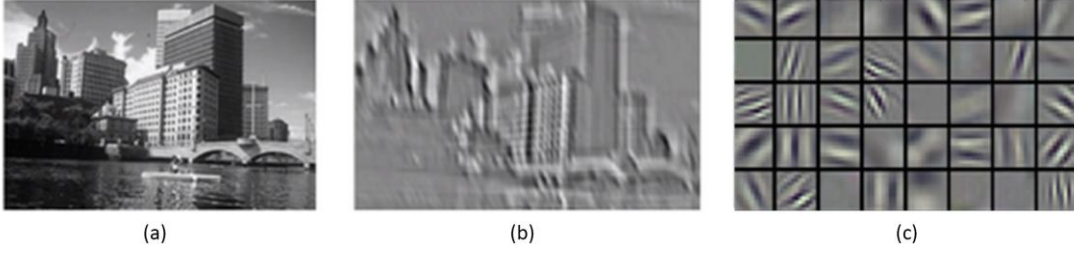
Şekil 3.4 : (a) stride değeri 1 (b) stride değeri 2 tarama aşaması ve çıkış matrisleri.

Stride ile verilen değer evrişim işlemi için ağırlık matrisi olan filtreyi görüntü üzerinde birer piksellik ya da daha büyük adımlarla kaydıracağının bilgisini verir. Bu da doğrudan çıkış boyutunu etkileyen bir parametredir. Giriş matrisi, filtre ve stride boyutuna göre çıkış matrisimizde farklılık göstermektedir (Şekil 3.4).



Şekil 3.5 : (a) padding değeri 1 olan katmanda filtre (b) nitelik haritası.

Şekil 3.5'te gösterildiği üzere 5*5 piksellik imajın dışında çizgili olarak ifade edilen her kenarından piksel olarak genişletilmesine padding denmektedir. Burada 1'er piksel uygulanan padding nitelik kaybını önlemek için yapılmaktadır. Diyagramda 3*3 piksellik bir filtre imaj üzerinde aldığı piksel değerlerini matris ile diğer katmana aktarmakta ve nitelik haritasını çıkarmaktadır (Şekil 3.6).



Şekil 3.6 : (a) girdi imajı (b) nitelik haritası (c) imaja uygun ağırlıklı kernel filtreler (URL4).

Stride, padding ve filtre değerleri girdiyi tarayarak çıktının boyutunu belirlemekle birlikte, diğer her değişken gibi modelin eğitim süresini de etkilemektedir. Bu yüzden uygulanacak modellerde optimum sayıda tutulmuştur.

Nitelik haritalama ve doğru ağırlık tespiti için oluşturulmuş piksel okuma ve üretme prensibine sahip evrişimli katmanlar tüm ÇÜA modellerinde kullanılmaktadır. ÇÜA modellerinin birbirinden ayrılma noktaları; katmanlarında kullanılan fonksiyonlar, optimizasyonlar, normalizasyonlar, stride değeri, padding değeri ve filtre gibi değişkenlerdir. Seçilen ÇÜA modelleri yapısı gereği bu değişkenlerin belirli hale gelmesi ile kurallandırılmıştır.

4. ÇEKİŞMELİ ÜRETKEN AĞLAR İLE DENEYLER

Derin öğrenme algoritmaları üç ana kategoriye ayrılmaktadır; denetimli öğrenme (supervised learning), denetimsiz öğrenme (unsupervised learning) ve pekiştirmeli öğrenme (reinforcement learning). Denetimli öğrenmede, x verisine ve (etiket) y verisi vardır ve amaç, x 'i y 'ye eşlemek için bir fonksiyon öğrenmektir. Oluşturulacak ilişki kullanılarak gelecekte verilen x' için y' çıktıları tahmin etmesi istenir. Regresyon, sınıflandırma, nesne algılama yaklaşımlarında kullanılır. Denetimsiz öğrenmede ise hiçbir etiket yoktur ve amaç, verilerin altında yatan bazı gizli yapıları bulmaktır. Denetimli öğrenmeden farklı olarak veri içerisindeki ilişkileri ve yapıları öğrenmektedir. Kümeleme, boyutluluk azaltma, özellik öğrenmesi yaklaşımlarında kullanılır (Alpaydın, 2010). Çekişmeli üretken ağlar ise derin öğrenme algoritmaları içinde denetimsiz öğrenme kategorisine girmektedir. Üretken modellerin amacı, bir dağıtımdan yeni veri örnekleri oluşturmaktır. Bu modeller, denetimsiz öğrenme problemi olan yoğunluk tahmini gibi problemlerde kullanılır (Brownlee, 2019).

ÇÜA tarafından üretilen sonuçlar dikkat çekici olsa da, kararlı bir model eğitmek zor olabilir. Bunun nedeni, eğitim sürecinin doğası gereği denetimsiz olması ve bunun sonucunda iki rakip modelin eş zamanlı geri yayımlı eğitimi ile sonuçlanmasıdır. Bununla birlikte, büyük miktarda ampirik deneme yapılma, ancak birçok uygulayıcı ve araştırmacı göz önüne alındığında, kararlı bir ÇÜA modelinin güvenilir eğitimi ile sonuçlanan az sayıda model mimarisi ve eğitim konfigürasyonu bulunmuş ve yayınlanmıştır (Saxena & Cao, 2020).

Araştırmada, ÇÜA ile yapılan deneylerden bir bilgi seti keşfetmek amaçlanmaktadır. Bu bakış açısıyla değerlendirildiğinde denetimsiz öğrenme modeli ile görüntü üretimindeki keşif sürecine en uygun farklı ağ yapısı ve içeriklerine sahip ÇÜA modelleri seçilmiştir. Rapor edilen bu modellerden DCGAN, Pix2Pix, CycleGAN ve StyleGAN modellerinin eğitimi için deneysel bir keşif süreci izlenmesi uygun görülmüştür.

ÇÜA algoritmalarının eğitimi için yoğun bir işlem gücü gerekmektedir. Her bir model eğitimi için binlerce iterasyon gerçekleştirilmektedir. Bu iterasyonlarda geri yayılım (back propagation) ile her bir nöronun ağırlık (weight) değerleri kısmi türevler (partial derivation) ile sürekli olarak hesaplanıp, hem üreticinin hem de ayırt edici kayıp fonksiyonu değerleri en iyilenecektir. Böylece eğitim sonunda ayırt edici gerçek bilgiyi tanımada, üretici gerçeğe yakın veri üretiminde usta olabilecektir. Sonuç olarak gerçeğe yakın yapay imajlar elde edilebilir olmaktadır.

4.1 Veri Seti ve Düzenlenmesi

Zu ve diğ. (2014) “Architectural Style Classification Using Multinomial Latent Logistic Regression” adlı çalışmalarında mimari stil sınıflandırması için bu veri setini oluşturmuş ve paylaşmışlardır. Veri seti olarak açık kaynak paylaşılmış çeşitli mimari tarz ile kategorilendirilmiş yapıların çevre görüntüleri seçilmiştir. Bu veri setinde 3 küme alınıp, piksel tabanlı olmasından dolayı kullanılacak ÇÜA’ya uygun olacak şekilde dosya formatları ve boyutları değiştirilmiştir. Bu üç veri setinin seçilmesinde içerik bakımından farklı ayrıntı ve özniteliklere sahip olması dikkate alınmıştır. Bu seçimlerin farklı öznitelik ve yapısal farklılıklar olmasının sebebi, kullanılan modellerin çalışma yöntemine uygunluğu olmuştur. Seçilen tarzların dönem ve mimari arka planından ziyade görüntü olarak anlattığı çevre ve yapı dikkate alınmıştır. Ahameniş tarzı, görüntü olarak az anlamsal ayrımlara sahip, kolay öğrenilebilir benzer öznitelikleri olması ile seçilmiştir. Bauhaus tarzı, Ahameniş tarzından farklı olarak veri seti içerisinde daha farklı öznitelik yapılarının olması ile seçilmiştir. Paladyan tarzı ise Bauhaus tarzına nazaran daha benzer öznitelikler olması fakat Ahameniş göre daha çeşitli anlamsal yapıların olması ile seçilmiştir. Bu üç veri seti adet ve içerik bakımından uygunlukları kontrol edildikten sonra azaltılıp, çoğaltılmış; modellere uygun olarak düzenlenmiştir. Bu noktada diğer değişkenler için veri seti üzerinde çok durulmamıştır. Asıl amaca hizmet etmesi için modellerin çalışmasından çıkan keşiflerle sorgulanacaktır.

Kullanılan modeller genel olarak denetimsiz öğrenme modeline girse de Pix2Pix koşullu yapısı ile denetimli öğrenme mantığında çalışır. Bu yüzden yapay

kategorili/etiketli bir veri kümesine ihtiyaç duymaktadır. Aynı şekilde iki küme ile çalışan CycleGAN için de ikinci bir küme oluşturulacaktır.

Çözünürlük, görüntü kalitesini etkileyeceğinden ve karşılaştırılmalı bir değerlendirme için veri boyutu 256*256 piksel olarak sabit tutulmuştur.

4.2 Deney Ortamı

Deneyler ilk olarak uygun görülen şekilde veriler eklenerek ve her deneyden öğrenilen bilgiler eşliğinde bir sonraki deney kurgulanmıştır. Deney ortamı olarak Python programlama dili ile bulut ortamında çalışan Google Colab ve bilgisayar içinde Anaconda Jupyter Notebook arayüzü seçilmiştir. Kullanım kolaylığı bakımında Colab ortamında sunulan GPU hizmeti kullanılmıştır. Ancak kullanımın hızlı olmasına rağmen Google tarafından zaman kısıtına tabidir. Bu yüzden bazı deneyler modelin eğitim süresinin hayli uzun olmasından kaynaklı lokal bir bilgisayarda Anaconda programı ile sürdürülmüştür.

4.3 Seçilen Modeller

Deneyler için katmanlarında kullandıkları farklı değişkenler ve yapılar göz önüne alınarak DCGAN, Pix2Pix, CycleGAN ve StyleGAN seçilmiştir. Bu modeller aynı zamanda ÇÜA modelleri arasından imaj üretim potansiyeli en yüksek modellerdir. Modeller kendi kullanım alanlarına göre düzenlenmiştir. ÇÜA modelleri kendine ait algoritma yapıları ile özelleşmiş isimler almışlardır. Bazıları yapısı gereği kurallandırıp isimlendirilmiş, bazıları ismini kullandığı veri setinden almıştır.

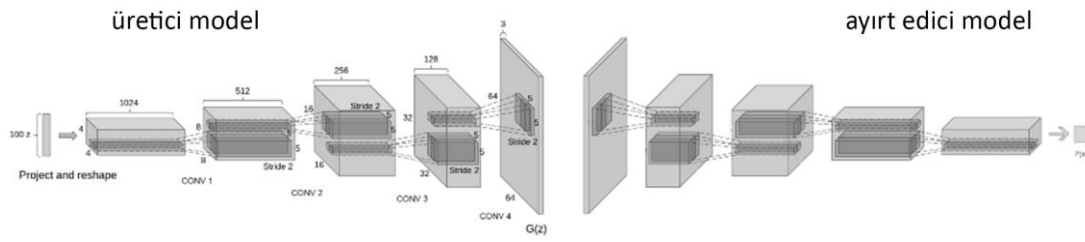
Eğitimler modellerin bitiş şartlarına göre değişiklik gösterir. Bu noktada derin öğrenme algoritmalarının çoğunda epok (döngü-iterasyon) sayısı ayarlanarak eğitilmektedir. Fakat seçilen algoritmalar arasından StyleGAN hata oranı istenilen düzeye geldiğinde durmaktadır. Bu noktada döngü sayısı ayarlanmadan çalışmaktadır.

4.3.1 DCGAN

Derin evrimsel çekişmeli üretken ağ olarak isimlendirilen DCGAN belli kriterler ile tanımlanmaktadır. Bu kriterlerin olmasının sebebi evrimsel katmanlarının belirli fonksiyonlar ve katmanlar ile çalışmasıdır. ÇÜA'nın DCGAN olarak tanımlanması için öncelikle ayırt edici kısmında evrimsel katmanlar arasında sırasıyla Leaky ReLu,

Batch normalizasyonu ve Sigmoid fonksiyonlarını kullanması gereklidir. Üretici kısmında ise Batch normalizasyonu, ReLu ve TanH fonksiyonları kullanılması gereklidir (Şekil 4.2) (Radford ve diğ.,2016).

Gauss ağırlık başlatma değerleri ortalama 0.0 ve standart 0.02 olarak uygulanmakta, en iyileyci fonksiyon Adam (Adaptive Moments) optimizör 0,0002 öğrenme oranı ve 0,5 beta 1 oranı ile uygulanmaktadır. Görüntüleri $[-1,1]$ aralığına ölçeklendirmek için üreticinin çıkışında tanH kullanılır. DCGAN için belirtilen bu düzen, bir dizi problem üzerinde yüzlerce veya binlerce konfigürasyon işlemi kombinasyonunu test eden ve değerlendiren uygulayıcılar tarafından zor kazanılmıştır (Radford ve diğ., 2016).



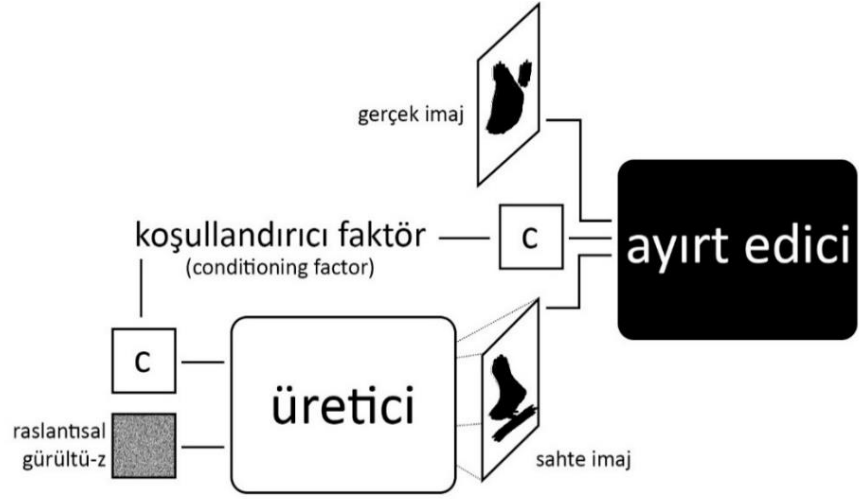
Şekil 4.1 : DCGAN algoritma çalışma yapısı (URL5).

Bu noktada çalışmada kullanılan model DCGAN kurallarına uygun şekilde 256*256*3 pikseli veri setini çalıştıracak şekilde düzenlenmiştir. Ayırt edicide, katmanları evrişim-BatchNorm-LeakyReLu-dropout oluştururken; üreticide, upSampling-evrişim- BatchNorm-ReLu oluşturmaktadır.

4.3.2 Pix2Pix

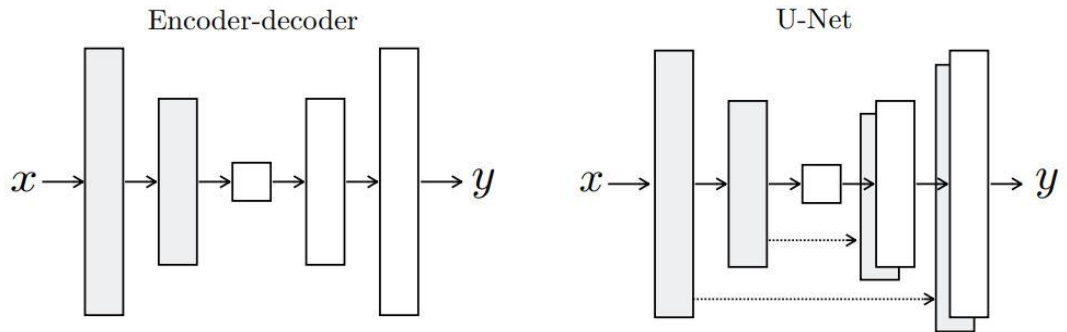
Pix2Pix bir giriş görüntüsünden bir çıkış görüntüsüne eşleme öğrenmek için koşullu bir üretken çekişmeli ağ (cGAN) kullanır, bu görüntüden görüntüye çevirme görevleri için derin bir evrişimsel sinir ağını eğitmeye yönelik bir yaklaşımdır.

Bir görüntü koşullu ÇÜA türü, hem önceki ÇÜA modellerine kıyasla büyük görüntülerin oluşturulmasını hem de çeşitli görüntüden farklı görüntülere iyi performans gösterme yeteneğine izin vermektedir (Isola ve diğ., 2017).



Şekil 4.2 : Pix2Pix, cGAN algoritma çalışma prensibi.

Hem oluşturucu hem de ayırıcı, evrişim-BatchNorm-ReLu biçimindeki modülleri kullanır. Bunlara “encoder” ve “decoder” denmektedir. Ayırt edici model, çıktı ile girdi görüntüsündeki piksel sayısı arasındaki ilişkiyi tanımlayan PatchGAN kullanır ve modelin her bir çıktı tahmininin giriş görüntüsünün 70×70 karesine veya yamasına eşlenmesini sağlar. Üretici model, ayırt edici modelden daha karmaşıktır. Üretici, U-Net mimarisi kullanan bir kodlayıcı-kod çözücü modelidir. Model, bir kaynak görüntüyü alır ve bir hedef görüntü oluşturur. U-Net mimarisi, kodlama katmanları ve karşılık gelen kod çözme katmanları arasına atlamalı bağlantıların eklenerek bir U şekli oluşturması anlamına gelir. Şekil 4.3’te kodlayıcının ilk katmanının kod çözücünün son katmanına nasıl bağlandığını, atlamalı bağlantıları göstermektedir. Üreticinin kilit noktası, ağ üzerinden giriş ve çıkış görüntüleri arasında paylaşılan düşük seviyeli bilgileri aktaran atlama bağlantıları ekleyen U-net mimarisidir.



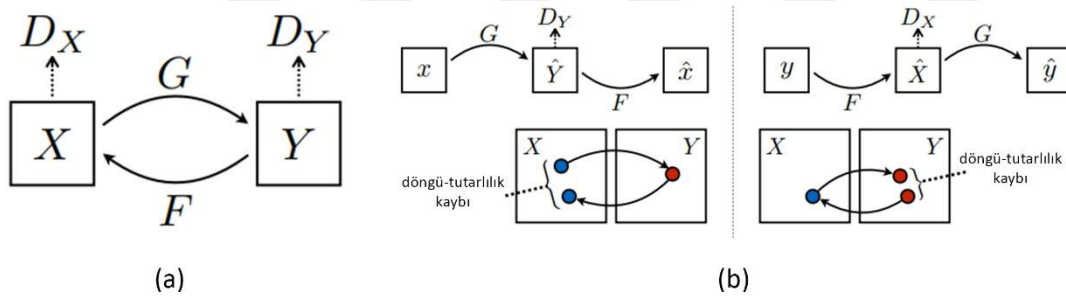
Şekil 4.3 : Pix2Pix, U-Net çalışma yapısı (Isola ve diğ., 2017).

Pix2Pix ile ilgili sorun eğitimidir, çünkü iki görüntü kümesinin sıkı ilişkili içinde görüntüyü tutan tek bir X/Y görüntüsüne önceden biçimlendirilmesi gerekir.

Bu noktada mimari yapıyı çevreyi sağlayan veri kümesinde etiketleme gerçekleştirilip ikinci veri kümesi oluşturulmuştur. Veri seti iki küme eşleşecek şekilde .npz dosya formatına dönüştürülmüştür. Pix2Pix modeli de anlatılan kurallandırmalara uygun olarak $256*256*3$ piksel veriye göre düzenlenmiştir.

4.3.3 CycleGAN

CycleGAN, görüntüden görüntüye çeviri için kullanılır. CycleGAN'ın amacı, bir görüntüyü etki alanından Y etki alanına ait gibi görünen bir görüntüye dönüştürmeyi öğrenen üreticiyi eğitmektir. CycleGAN, bir görüntü etki alanından diğerine eşlemeyi öğrenmek için denetimsiz bir yaklaşım kullanır, yani eğitim görüntülerinin etiketleri yoktur. Fakat model ihtiyacı iki benzer niteliklere sahip veri kümesidir (Zhu ve diğ., 2017). Bu yönüyle Pix2Pix'e benzeyen bu model, yığın normalleştirme yerine örnek normalleştirme kullanır. Bu değişiklik gerçek zamanlı görüntü üretimi için yüksek performans sağlamaktadır.



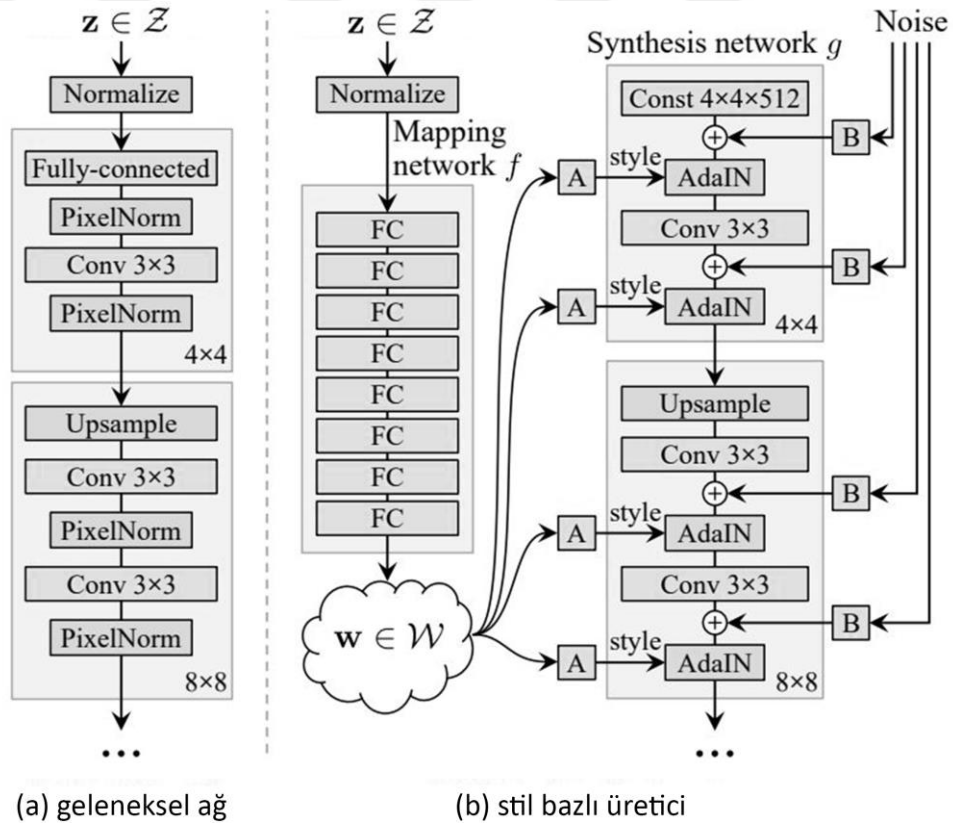
Şekil 4.4 : (a) CycleGAN çalışma prensibi (b) CycleGAN çift yönlü iş akışı (Zhu ve diğ., 2017).

Çalışmada kullanılan CycleGAN modeli aynı şekilde $256*256*3$ piksel veri seti için düzenlenmiştir. Pix2Pix için hazırlanan veri seti eşlenmeden bu model için de hazırlanmıştır. Ek olarak çizim stil transferi uygulaması için deney tasarlanmış ve bu modele özel yeni bir veri seti oluşturulmuştur.

4.3.4 StyleGAN

StyleGAN, tarz transferi için çekışmeli üretken ağlara alternatif bir üretici yapısı önermektedir. “Derin Evrişimli Sinir Ağı” (DCNN)’in ilerlemesinden yararlanan Gatys ve diğ. (2016), hedef görüntünün anlamsal içeriğini korurken görüntü içeriğini

ve doğal görüntülerin stilini ayırabilen ve yeniden birleştirebilen “A Neural Algorithm of Artistic Style” (Sanatsal Tarzın Sinirsel Algoritması) tanıtmıştır. Bu geleneksel şekilde çalışan bir sinir ağıdır. Daha sonra 1024 çözünürlüğe kadar 2 veya daha fazla stil arasında seri sürekli dönüşüm görüntüleri oluşturmada son derece iyi olan StyleGAN (çekişmeli üretken ağlar için stil tabanlı bir oluşturucu mimarisi) sunulmuştur. Bu yeni yapı otomatik olarak öğrenilen, üst düzey niteliklerin ve oluşturulan görüntülerdeki değişken çeşitliliklerinin denetimsiz bir şekilde ayrılmasını sağlar. Yeni üretici, geleneksel dağıtım kalitesi metrikleri açısından yeni geliştirilmiştir. Yapı, gözle görülür şekilde daha iyi yaklaşım özelliklerine yol almakta, gizli çeşitleme faktörlerini iyi çözmektedir (Karras ve diğ., 2019). Burada, ayrı bir evrişim katmanının girişi, gizli vektör "stil" yerleştirmeleri kullanılarak uyarlamalı örnek normalleştirme (AdaIN) işlemiyle normalleştirilir.



Şekil 4.5 : (a) geleneksel ağ (b) StyleGAN çalışma prensibi (Karras ve diğ., 2019).

Bu kullanım ağın sürekli olarak stil ile beslenmesini ve stile bağlı örnekler çıkmasını sağlar. Ayrıca başlangıç verisi seçilebilir alt yapısı ile farklı stiller arasında geçişler sağlamaktadır. Haritalama ağının ve AdaIN yapısının eklenmesi, Üreticiye direkt gürültü girişinin kaldırılması, her bloğa gürültü eklenmesi ile çok yeni bir yapı

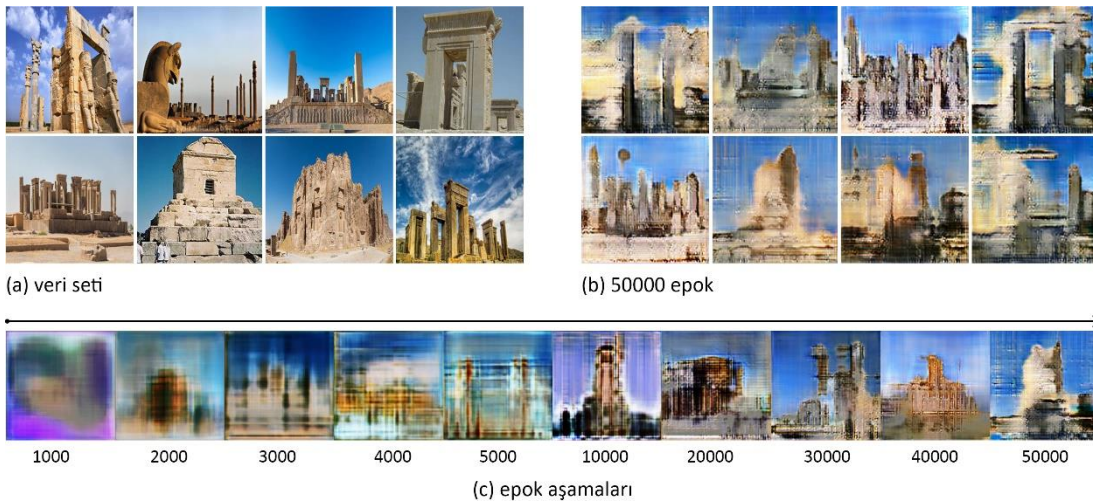
tanımlanmaktadır. Çalışmada bu model, daha önce 256*256*3 piksele uygun modelin GitHub (açık kaynak yazılım paylaşım servisi)'dan kopyalanarak eğitimi yapılmaktadır. Bu yüzden yapıda herhangi bir müdahale yapılmamıştır.

4.4 Deneyler

Deneyler, hazırlanan ve ayarlanan veri setleri ile oluşacak potansiyelleri keşfetmeye yönelik kurgulanmıştır. Veri seti olarak farklı semantik ayrımları olan ve birbirinden görece nitelik sayısı değişken olan veri setleri seçilmiştir. Bu deneyler ile modellerin veri setlerindeki kriterlere nasıl cevap verdiği ve modellerin genel çalışma akışlarında oluşabilecek keşifleri gözlemlenecektir. Çıkan görüntülerin oluş akışları eğitim epok çıktılarına bakılarak dolaylı kalitatif bir yöntemle değerlendirme yapılacaktır.

4.4.1 DCGAN deneyleri

İlk deney için veri setindeki ilk küme Ahameniş mimari stil alınıp 256*256 piksel düzenlenerek 392 imaj DCGAN modeline eklenmiştir. İlk eğitimin oluşturulacak deneylere altlık oluşturması amaçlandığından ekstra bir işlem yapılmamış, epoh sayısı fazla tutularak eğitime Anaconda Jupyter ortamında başlanmıştır. Aşağıda, eğitim dizisinin girdisi ve çıktısı gösterilmekte; yaklaşık bir günlük süresince gerçekleştirilen bir DCGAN modelinin veri setindeki özniteliklerle görüntü ayrımını yavaş yavaş oluşturduğu görülmektedir.



Şekil 4.6 : DCGAN ilk deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.

DCGAN için seçilen veri seti tek bir kümede benzer ölçekte aynı tip veri bilgisine sahip olmalıdır. Bu yüzden veri seti içinden bir mimari tarz alınıp modele eklenmiştir. DCGAN burada veri setindeki imajların özniteliklerini bularak, bu özniteliklerle özgün bir üretim yapmayı deneyecektir. Şekil 4.6'da görüldüğü üzere epok 1000 çıktılarında şekilleri ve renkleri henüz algılayamamıştır. Epok 10000'e kadar renk kararsızlıkları süren öğrenme sürecinde, 10000 ile birlikte bir renk şeması çıkardığı gözükmemektedir. Giderek şekilleri tanımaya çalışan eğitimde oluşan bazı görüntüler veri setine uygun görüntüleri terk etmektedir. Şaşırtıcı sonucu epok 50000'de vererek imajlarda özgünlüğe yaklaşmaya başlamıştır. Şekil 4.6'te veri seti ile yaklaşan benzerlikleri görülmektedir. Çıktılar az gürültülü olsa da veri setine uygun net görüntüler elde edilmemiştir.



Şekil 4.7 : DCGAN ilk deney veri seti ve epok 50000 çıktıları.

İlk deneyde çıktılarında epok sayısının yüksek tutulmuş olmasına rağmen algoritma net görüntülere ulaşamamıştır. Bir sonraki deney için epok sayısının iki katına çıkarılmasına karar verilmiştir. Veri setinin ise daha ayrıntılı görüntüde veri setinin denenmesi düşünülerek, algoritmanın niteliği tanımadaki farklılıkları gözlemlenecektir. Ahameniş mimari stil veri setinden daha çeşitli özniteliklere sahip olan Bauhaus veri seti uygulanmıştır. Bu sette yer alan 315 Bauhaus imajı modele uygun ölçeklendirip, 100000 epok ayarlanarak Anaconda Jupyter ortamında deneye başlanmıştır. Deneyin 100000 epokta veri setine benzer gerçeklikte görüntüler oluşturulması beklenmektedir.

Burada görüldüğü üzere modelin öğrenmesinde gecikmeler yaşanmıştır. Ahameniş deneyinde 10000 epokta renkleri tanımaya başlamış olmasına rağmen veri setindeki

daha özgün çeşitlilikler nedeni ile renkleri tanımayı 20000-30000 epok arasında oturtmuş şekiller belirlemeye başlamıştır. Sonuçlarda verdiği çeşitlilik Şekil 4.8’de gösterilmiştir. Bu deneyle DCGAN yapısının daha büyük veriler ve uzun eğitimlerle özgün içerik üretiminde başarılı olacağı görülmektedir.

Sonuçlar malzeme ve yapıdaki ayrıntı çeşitliliği konusunda çıktı görsellerde oluşan çeşitliliği beslemiştir. Çıktılar yine stili okunabilecek bina görselleri verse de halen gürültülü noktalar bulunmaktadır.



Şekil 4.8 : DCGAN ikinci deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.



Şekil 4.9 : DCGAN ikinci deney veri seti ve epok 100000 çıktıları.

Bauhaus deneyinin verdiği çok çeşitli çıktılarla gürültünün hala gitmemiş olması veri seti öz niteliklerinin fazlalığı ile alakalı olduğu düşünülmektedir. Bu noktada modele stil tarzı daha benzer ama bina seviyesinde ayrıntı miktarı aynı olan Paladyan veri seti eklenmiştir. Burada genellikle ön cephe görünüşünde bulunan imajların model tarafından daha erken öğrenildiği gözlemlenmiştir.



Şekil 4.10 : DCGAN üçüncü deney (a) veri setinden örnekler (b) 50000 epok çıktıları (c) epok aşamaları.

İlk deneyde olduğu gibi 10000 epokta bina detayının oluşmuş olması ve gürültü seviyesinin azalması, veri setindeki her imajın öz nitelik benzerliğinden kaynaklandığı düşünülmektedir.

DCGAN ile yapılan bu üç deneyin kısmen başarılı otonom üretim aracı olması mimarlıkta kullanımı için oldukça geniş üretim bandı oluşturmaktadır. Fakat eğitimlerin öğrenmesini tamamlaması için daha iyi sonuçlar verecek epok sayısı ve deneylerde kullanılan daha büyük veri ile gerçekleşmesi gerekmektedir. Bu da son ürün odaklı kararlı modellerde kullanımı uygun görülmektedir.



Şekil 4.11 : DCGAN üçüncü deney veri seti ve epok 100000 çıktıları.

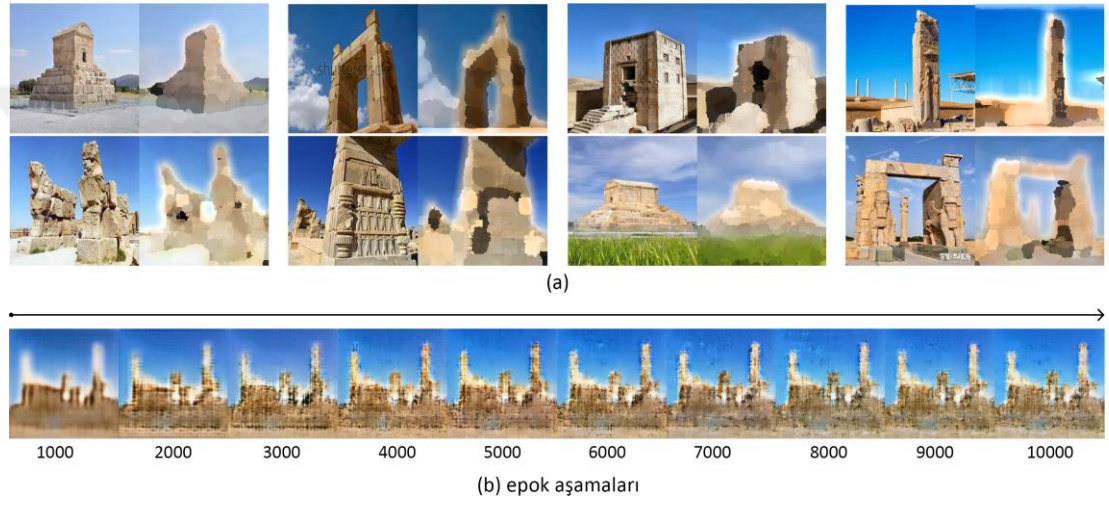
DCGAN ile yapılan bu üç deneyde, değişen öğrenme seviyeleri tespit edilmiştir. İlk deney ve üçüncü deneyde 10000 epok itibari ile şekiller netleşirken, ikinci deneyde 30000 epokta bina şekilleri netleşmeye başlamıştır. DCGAN deneylerinde amaç, veri setine benzer, gerçeğe yakın ve özgün görüntüler oluşmasını sağlamak olmuştur. Bu noktada değerlendirildiğinde veri seti yetersizliklerine rağmen veri setinde görülmemiş benzer görüntüler oluşturabilmiştir.



Şekil 4.12 : DCGAN üç deneyi, epok değerlerine göre karşılaştırması.

4.4.2 Pix2Pix deneyi

Pix2Pix deneyi için DCGAN’da kullanılan Ahameniş mimari stil veri seti anlamsal bir yapı ayrımıyla etiketlenerek; piksel piksel eşleme ve binary sınıflandırma sağlayan “npz” formatında ikinci bir veri seti hazırlanmıştır. Deney Google Colab ortamında GPU çalışma türünde gerçekleştirilmiştir. Veri sınıflandırmasını karşılıklı okuyan Pix2Pix etiketlenmiş veri için imaj transferinde 1000 epok deney yapılmıştır. Bu deney için 100 etiketli görüntü eşli olarak kullanılmıştır. Pix2Pix içinde kullanılan koşullu sınıflandırma yapısının veriyi hızlı bir şekilde kullanma noktasında dezavantajlı olduğu çıktılarıyla okunmaktadır.

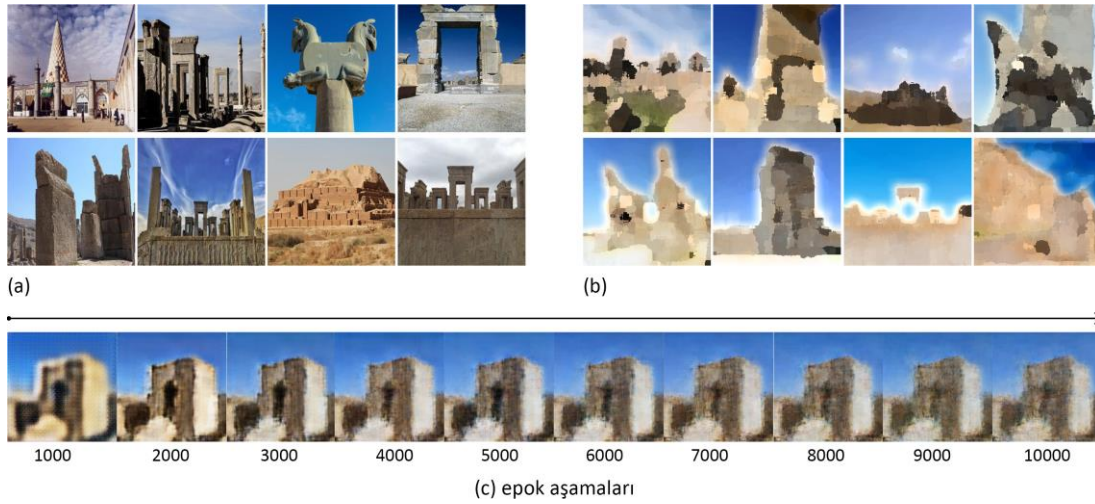


Şekil 4.13 : Pix2Pix deneyi (a) kısmi etiketli eşlenmiş veri seti (b) epok aşamaları.

Veri kümesinin oluşturulması ve deneyde kullanılan veri miktarından daha fazla veriye ihtiyaç duyuyor olmasından dolayı Pix2Pix için deneylerin devam etmesi söz konusu olmaktan çıkmıştır. Bu yapıdaki üretken ağ için daha kararlı ve oturmuş tasarım kararlarında kullanım öngörülmektedir.

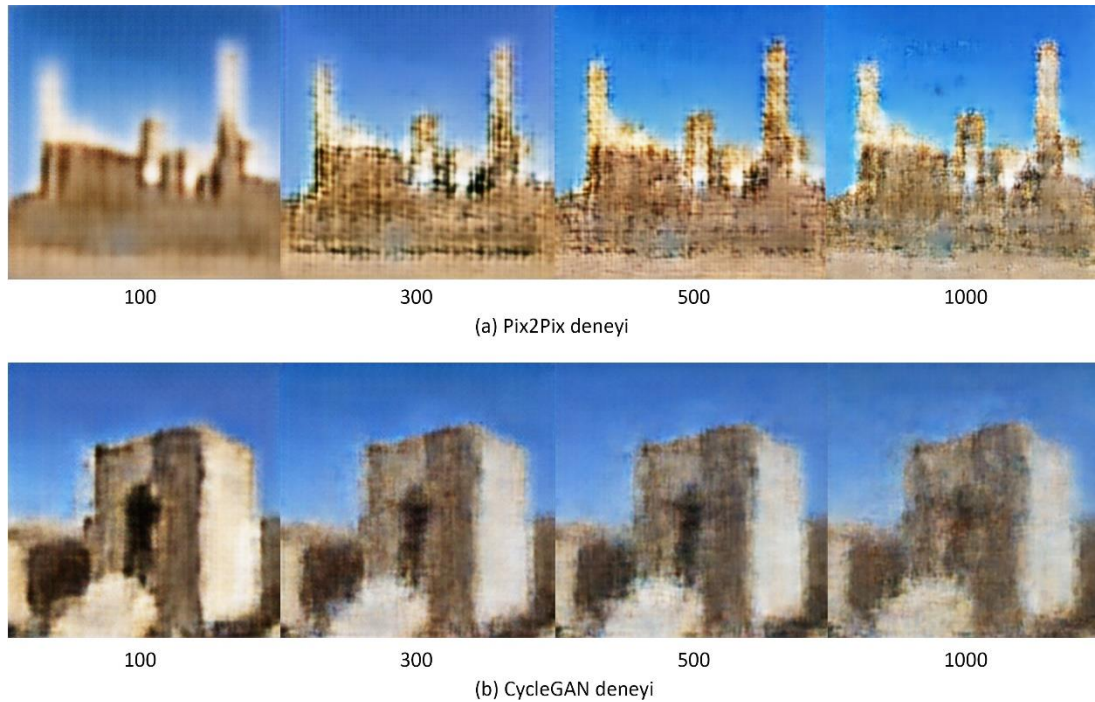
4.4.3 CycleGAN deneyleri

Pix2Pix için hazırlanan veri birleştirilmeden iki kategori şeklinde CycleGAN’a aktarılmıştır. Deney için yine Google Colab ortamında, çalışma türü GPU olarak gerçekleştirilmiştir. Bu deney, iki kümeli çalışan farklı iki yapıdaki ÇÜA modelinin kıyaslanabilmesini sağlayacaktır.



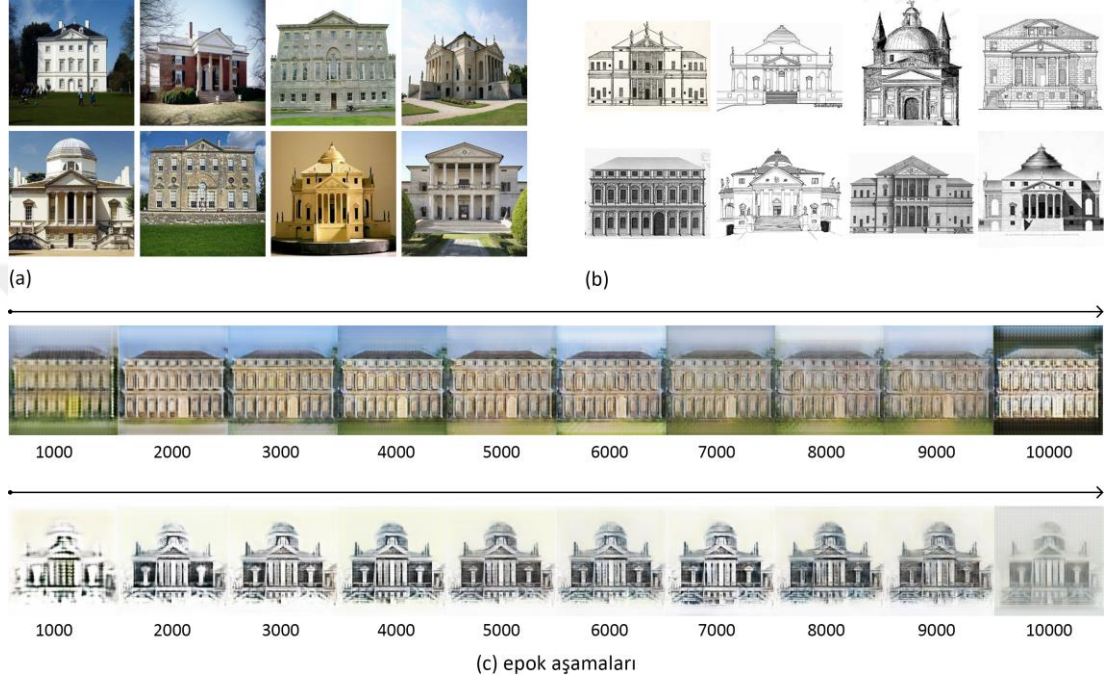
Şekil 4.14 : CycleGAN ilk deney (a) birinci küme (b) ikinci küme (c) epok aşamaları.

Şekil 4.14 ve Şekil 4.15'te görüldüğü üzere, Pix2Pix'de eşlenilip kullanılan veri seti CycleGAN'a karışık şekilde iki küme halinde eklenmiştir. Pix2Pix eşlenmiş veri kullanmasına rağmen görüntüyü netleştirmede CycleGAN bu epok sayısında daha başarılı olmuştur. Bu kıyaslama ile görülen Pix2Pix'in daha fazla veri ve uzun bir eğitime ihtiyaç duyması onu DCGAN gibi son ürün odaklı otonom araç uygulamalarında kullanıma uygunluğunu gösterirken; CycleGAN'ın Pix2Pix'e göre hızlı eğitim ve az veri ile belirgin sonuçlar vermesi, kullanım alanını gerçek zamanlı uygulamalar için uygun olduğunu öne çıkarmaktadır.



Şekil 4.15 : Pix2Pix, CycleGAN karşılaştırması.

CycleGAN'ın bu hızlı dönüştürücü uygulaması benzer iki küme arasında stil dönüştürücü olarak kullanılması Paladyan tarzı için uygulanması ile devam etmiştir. Bu deney için veri setine ikinci küme olarak Paladyan binalarının cephe görünüşlerinden oluşan çizim seti oluşturulmuştur. Bu veri seti ile iki küme arasında dönüşüm sağlayacak ikinci deney eğitime alınmıştır. Yine Google Colab ortamında başlanan eğitim veri ve çıktıları Şekil 4.16'da görülmektedir.



Şekil 4.16 : CycleGAN ikinci deney (a) birinci küme (b) ikinci küme (c) epok aşamaları.

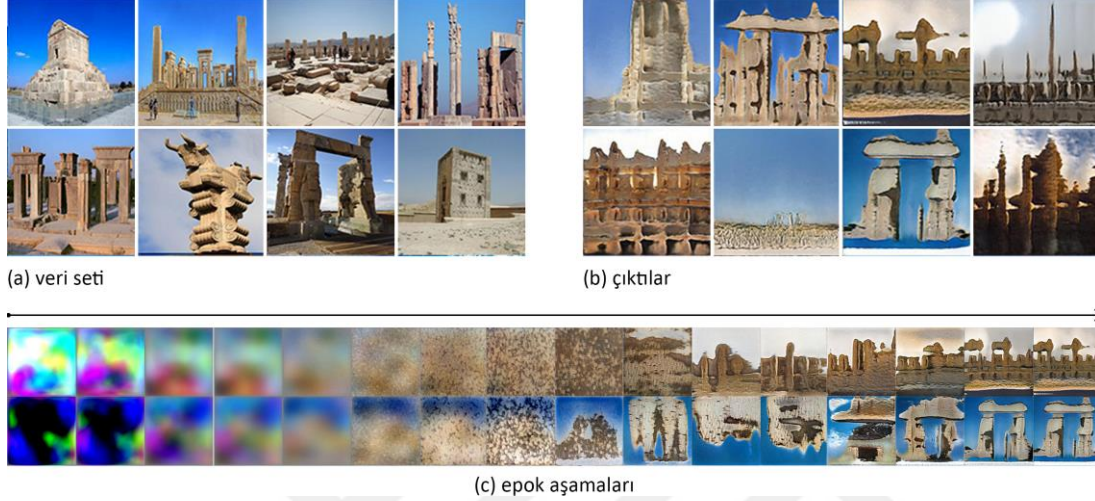
Bu deney, ilk deneyle ortaya çıkan hızlı eğitim başarısını kanıtlamaktadır. Bu başarının sebebi aynı zamanda benzer öznelikte iki stil ile kümelenmiş veri setidir. Deneyin ilk 1000 epoktan itibaren iyi sonuç vermesi çok hızlı bir dönüşüm sağladığını göstermektedir. Öyle ki 10000 epokta ilk öğrendiği stilden farklılaşmaya başlayarak dönüşmektedir.

4.4.4 StyleGAN deneyleri

StyleGAN ile yapılan ilk deneyde yine Ahemeniş veri seti modele eklenmiştir. Model orjinal makalesde kullanılan kodun GitHub'tan drive klasörüne kopyalanması ile kullanıldığı için Google Colab ile eğitime başlanmıştır. StyleGAN'ın yapısı girdi ve stil verisi ile kullanılabilir. Bu deneyde ilk girdi tanımlanmamış sadece çıktı için stil veri seti eklenmiştir. Böylece sıfırdan bir görüntü oluşumu hedeflenmiştir. Bu

noktada DCGAN ile kıyaslanabilecek bir eğitim tasarlanmıştır. Şekil 4.17’te görüldüğü üzere görüntü piksel piksel renklerin tanınması ile oluşmaya başlamıştır.

Ardından şekil geçişlerindeki belirmelerin ayrıntısız ama gürültüsüz olması açısından deney sonuçları net olmaktadır. Bu deneyle StyleGAN diğer modellerden daha net görüntüler oluşturmayı başarmıştır.

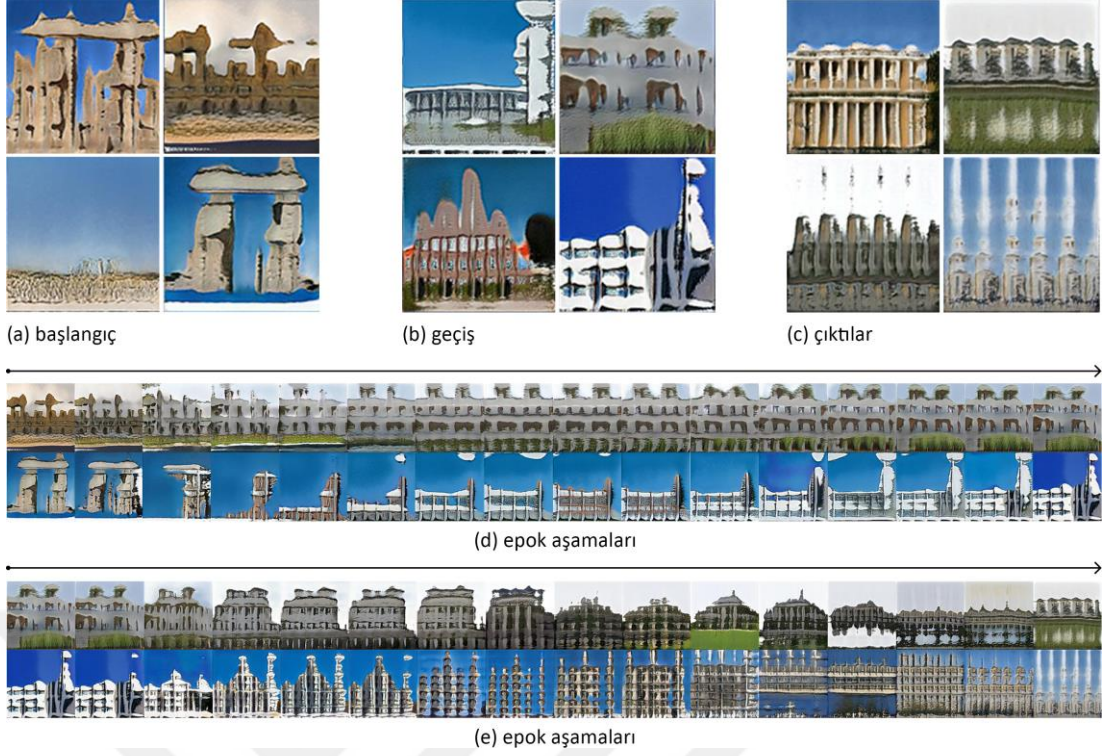


Şekil 4.17 : StyleGAN ilk deney¹ (a) veri seti örnekleri (b) çıktılar, son döngü 1185 epok (c) epok aşamaları (soldan sağa düzenli artmaktadır).

StyleGAN ile yapılan ikinci deneyde ilk deney sonuçları altlık olarak kullanılarak stil transferi için oluşum potansiyelleri denenmiştir. İlk olarak Bauhaus stili eklenmiştir. Bu geçişte kayıplar görülmüş, stil transferini tamamlayamamıştır. Ardından Paladyan stili eklenmiş çıktılar Şekil 4.18’de görülmektedir. Bu deneyle farklı stillerin karışımı ile yeni bir yapı görseli oluşturulması amaçlanmış fakat veri çeşitliliğinde nitelik benzerliği ve veri sayısındaki azlık nedeni ile istenen etki görülmemiştir.

¹ StyleGAN ilk deney çıktılarının simülasyon videosu karekod ile ulaşılabilir.



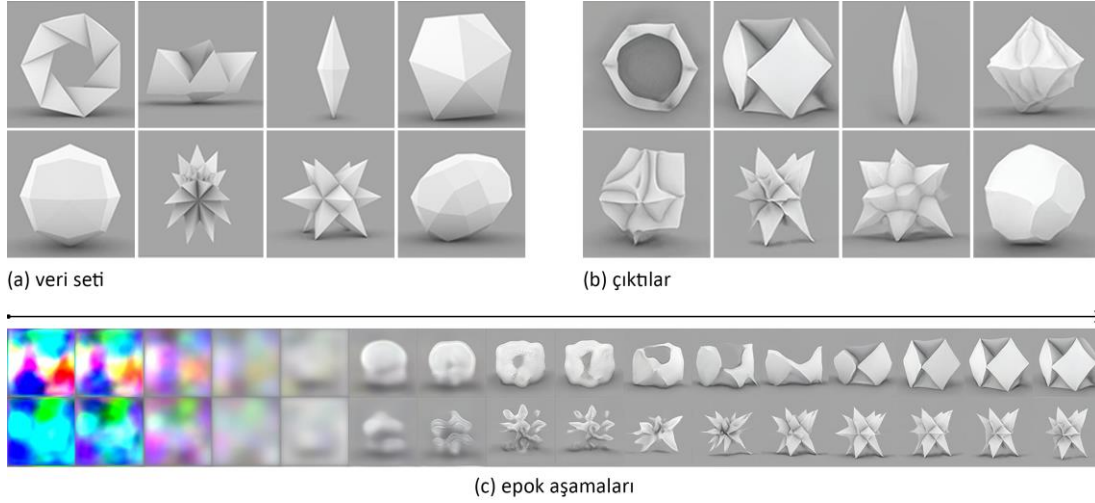


Şekil 4.18 : StyleGAN ikinci deney² (a) başlangıç imajları; ahameniş ilk deney çıktıları (b) geçiş dönemi (c) çıktılar, son döngü 890 epok (d) epok aşamaları soldan sağa; Ahameniş altlığına Bauhaus stiline uygulanması (e) epok aşamaları (soldan sağa; Bauhaus geçiş stiline Paladyan stiline uygulanması).

StyleGAN ile keşfedilen uyumlu epok geçişleri iyi bir simülasyon yeteneği göstermektedir. Oluşan geçişlerde mimari topografi ve form bulma denemeleri gözlenmiştir. Böylece modelin 3 boyutlu model görselleri ile çalışılması uygun görülmüştür. Üçüncü deney için sıfırdan basit geometrik modellerden başlamaya karar verilerek 200 adet çok yüzlü 3 boyutlu geometrik imajlardan oluşan bir veri oluşturulmuştur. Daha önceki deneylerde keşfedilen simülatif form bulma yeteneğini daha iyi görmek için eğitime başlanmıştır.

² StyleGAN ikinci deney çıktılarının simülasyon videosu karekod ile ulaşılabilir.





Şekil 4.19 : StyleGAN üçüncü deney³ (a) veri seti örnekleri (b) çıktılar, son döngü 1180 epok (c) epok aşamaları (soldan sağa düzenli artmaktadır).

Epok aşamalarında görüldüğü üzere üçüncü deneyde oldukça net imajlar ortaya çıkmaktadır. Deney verisinin basit 3 boyutlu modeller olması eğitim çıktılarının belirgin şekillerde oluşmasını sağlamaktadır. Sonuç olarak, form bulma problemleri için StyleGAN'ın uygulamaya açık olduğu görülmektedir.

³ StyleGAN üçüncü deney çıktılarının simülasyon videosu karekod ile ulaşılabilir.



5. BULGULAR VE TARTIŞMA

Araştırma, ÇÜA ile mimari üretim potansiyelleri üzerine odaklanarak çeşitli deneyler kurgulanmıştır. Bu araştırma, gömülü teori yöntemi ile durum çalışması şeklinde ilerlemiş, yapılan deneylerle algoritmaların mimari tasarım alanındaki kullanımına yönelik olasılıklar irdelenmeye çalışılmıştır. Farklı veri setleri ve ÇÜA'lar kullanılarak elde edilen görsel çıktılar değerlendirilerek olası kullanım potansiyelleri açıklanmaya çalışılmıştır. ÇÜA'ların çalıştırılmasında yaşanan GPU yetersizlikleri nedeni ile deney ortamı değiştirilmiş ve görsel boyutları 256*256*3 pikselde tutulmuştur. Deneylerde karşılaştırma değişkenlerinin sabit tutulma amacıyla girdi görsellerinin boyutu her algoritma için aynı tutulmuştur. Deney çıktılarının değerlendirilmesi için ise Borji (2019) tarafından önerilen nitel ve nicel değerlendirme yöntemleri uyarlanmıştır. Bu yöntemler arasında benzerlik oranı en yakın olan görsel ile sezgisel bir değerlendirme olan nitel yöntem seçilmiştir. Bu noktada deneylerin çıktıları, epok aşamaları, bu aşamalardaki modelin davranışı üzerinden gözleme dayalı dolaylı nitel yöntemle değerlendirmeler yapılmıştır.

5.1 Bulgular; Keşifler

ÇÜA, 4. bölümde bahsedildiği üzere görüntü sentezi konusunda oldukça potansiyelli ve gelişmekte olan bir denetimsiz öğrenme modelidir. Bu modelde, belirli bir fotoğraf koleksiyonu ile eğitilerek makul görüntüler üretebilir, sanat yaratabilir. Yeni ve sanatsal görüntüleri, eskizleri, resimleri ve daha fazlasını mükemmel hale getirmeye yardımcı olabilir. Daha bir çok çalışmayı gerçekleştirebilecek bu algoritmalar; görüntü tercümesi, fotoğraflarla konsept değiştirme gibi pekçok işlemi yapabilir ve geliştirebilir. Araştırmada, bu çalışmalara altlık oluşturabilecek keşifler sağlayan toplamda 9 deney tasarlanmıştır. Bu deneyler, Çizelge 5.1'de görüldüğü gibi üçü DCGAN, biri Pix2Pix, ikisi CycleGAN ve üçü StyleGAN ile yapılmıştır. Çizelgede belirtilen veri setleri bölüm 4.1'de açıklanan veri setleri ve deneylerin ihtiyacına göre hazırlanan veri setlerini ifade etmektedir. Adet bilgisi kullanılan veri setlerindeki imaj

miktarını, epok sayısı eğitim sırasında öğrenme için belirlenen döngü sayısını belirtmektedir. Burada DCGAN, Pix2Pix ve CycleGAN eğitime belli bir sayı hedefi konarak başlanırken, StyleGAN’da epok hedefi belirlenmemektedir. Hata oranı en düşük döngüde algoritma çalışmayı durdurduğu için döngü sayıları gerçekleşmiş son epok sayısını göstermektedir. Deney ortamı için kısıtlamalar ve GPU kullanımında en iyi ortam seçilerek yapılmıştır. Son olarak; çizelgedeki zamanı bilgisi eğitimlerin gerçekleştiği süreyi saat cinsinden vermektedir.

Çizelge 5.1 : Deneylerin detay çizelgesi.

GAN modeli	Veri Seti	Adedi	Epok	Deney Ortamı	Zaman (saat)
DCGAN	Ahameniş (A)	392	50000	Anconda Jupyter	26
DCGAN	Bauhaus (B)	315	100000	Anconda Jupyter	46
DCGAN	Paladyan (P)	343	100000	Anconda Jupyter	47
Pix2Pix	A (etiketli-eşli)	100'er	1000	Google Colab	3
CycleGAN	A (etiketli-kümeli)	100'er	1000	Google Colab	1
CycleGAN	P (etiketli-kümeli)	100'er	10000	Google Colab	5
StyleGAN	A	392	1185	Google Colab	27
StyleGAN	A-->B-->P	392-100-100	890	Google Colab	24
StyleGAN	Çok yüzlü model	200	1180	Google Colab	26

Yapılan deneyler ile araştırmanın mimari alandaki keşifleri Çizelge 5.2’de özetlenerek şu şekilde sıralanabilir;

- DCGAN’ın farklı özniteliklerde çıktılar oluşturduğu gözlenmiştir. Fakat büyük veri ve uzun bir eğitim ihtiyacı görülmüştür. Öznitelik çeşitliliğinin azalması ile eğitim kısalmakta; öznitelik çeşitliliğinin artması ile eğitim uzamaktadır.

Diğer modellere göre iyi sonuçlar almak için eğitim süresinin fazla olduğu görülmektedir.

- Pix2Pix'in detaylı etiketlenmiş ve büyük bir veri seti ile eğitilmesi gerekmektedir. Bu noktada eşlemeli veri üretim yaptığı için detaylı sonuçlar verebilmektedir. Fakat etiketli büyük bir veri seti hazırlamak ve bunu eğitmek çalışılan diğer modellere göre süreci oldukça zorlaştırmaktadır.
- CycleGAN, iki kümeli dönüşüm problemlerinde hızlı bir eğitim yapmaktadır. Kullanılan veri miktarının 100 imaj kadar az olmasında dahi net sonuç verebildiği gözlenmiştir. Stil üzerinden örnek oluşturma problemlerini iyi yanıtladığı yapılan ikinci deney ile ortaya çıkmaktadır.



(a) DCGAN 50000 epok

(b) StyleGAN 1185 epok

Şekil 5.1 : DCGAN, StyleGAN karşılaştırması.

- StyleGAN, stile bağlı gürültüsüz net imajlar üretebilmektedir. Ahameniş veri seti ile yapılan deneylerde DCGAN ile kıyaslandığında StyleGAN'ın daha net imajlar oluşturduğu gözlenmiştir (Şekil 5.1). Bu noktada StyleGAN, hedef veriye benzemeye çalışırken; DCGAN, hedef verinin özneliliklerini öğrenerek daha süpriz sonuçlar çıkmasını sağlamaktadır. DCGAN için sunulan bu çıktı Bauhaus veri seti ile yapılan deneyde görülmektedir.
- StyleGAN'ın epok sayıları az olsa da eğitim süresi DCGAN ile aynı saatlere ulaşmaktadır. Aynı zamanda epok aşamalarında diğer modellerden ayrı olarak simülasyon yeteneği gözlenmiştir.

Çizelge 5.2 : Deneylerin keşif çizelgesi.

GAN modeli	Veri Seti	Keşifler
DCGAN	Ahameniş (A)	• Çıktılarda özgün şekiller • 10000 epokta öğrenme • 50000 epokta veri setine yakın az gürültülü çıktılar oluşturma
	Bauhaus (B)	• Değişken öznitelikleri koruma ve çıktılarda gösterme • Öznitelik artması ile eğitimin uzaması
	Paladyan (P)	• Verideki özniteliklerin benzer olması ile eğitimin kısalması
Pix2Pix	A (etiketli-eşli)	• Uzun eğitim ve büyük etiketlenmiş veri gerekliliği
CycleGAN	A (etiketli-kümeli)	• Az veri ve kısa eğitimle pix2pix'e göre daha iyi sonuç vermesi
	P (etiketli-kümeli)	• Daha zahmetsiz veri seti hazırlama olanağı ve kısa eğitim süresi ile stil transferi
StyleGAN	A	• Az ayrıntılı ve gürültüsüz çıktılar • Mimari topoloji araması ve stil transferi aktarımındaki uyum süreci
	A-->B-->P	• Karışık stillerde bazı özniteliklerin tamamen kaybolması, bazı özniteliklerin tekrarlanması
	Polyhedron	• 3 boyutlu form bulma aşamalarında eksiltmeli-artırmalı süreç tarifi, simülasyonu

5.2 Tartışma; Potansiyeller

Gözlemlenen bu bulgulara (keşiflere) karşılık gelen potansiyeller Çizelge 5.3'te özetlendiği üzere şu şekilde sıralanabilir:

DCGAN için keşfedilen uzun eğitim süreci, modelin uygulanabilirliğini düşürmektedir. Buna rağmen veri setine özgü özniteliklerin çıktılarda oluşturulmaya başlanmış olması ve bu özniteliklerde çeşitliliği koruyabilmesi modelin son ürün odaklı otonom bir üretim uygulaması oluşturma potansiyelini arttırmaktadır. Uzun (2020), DCGAN ile paladyan tipi mimari plan şeması üretim problemini çözmeye çalışmış ve bu konuda olasılıksal yöntemlerin doğru sonuçlar verebildiğini savunmuştur. Aynı zamanda olasılıksal üretim prensibi nedeniyle beklenmedik ancak olasılıksal dağılıma yakın sonuçlar verebildiğini de kaydetmiştir. Sonuç olarak,

DCGAN için beklenmedik eğitim çıktılarının söz konusu olması otonom bir araç için yaratıcı bir durum tarif etmektedir.

Pix2Pix için kurgulanan deneyde, etiketli veri ile çalışması nedeniyle deney hazırlığı uzun sürmektedir. Veri seti hazırlığı modelin uygulanabilirliğini DCGAN’da olduğu gibi düşürmektedir. Koşullu öğrenme sayesinde semantik sınıflandırma eşlemesi yapan Pix2Pix için büyük veri ve uzun eğitimle otonom bir üretim aracı olma potansiyeli vardır. Bu noktada, DCGAN ile aynı şekilde son ürün odaklı üretken modeller için uygulama oluşturabilir.

CycleGAN, yapılan deneyler sonucu hızlı bir eğitim ve az veri kullanarak çıkardığı imajlarla hem gerçek zamanlı uygulamalarda farklı eğitilmiş modeller oluşturabilir hem de çeşitli stil değişimleri için erken tasarım aşamalarında uygulanabilir.

Çizelge 5.3 : Deneylerin potansiyel çizelgesi.

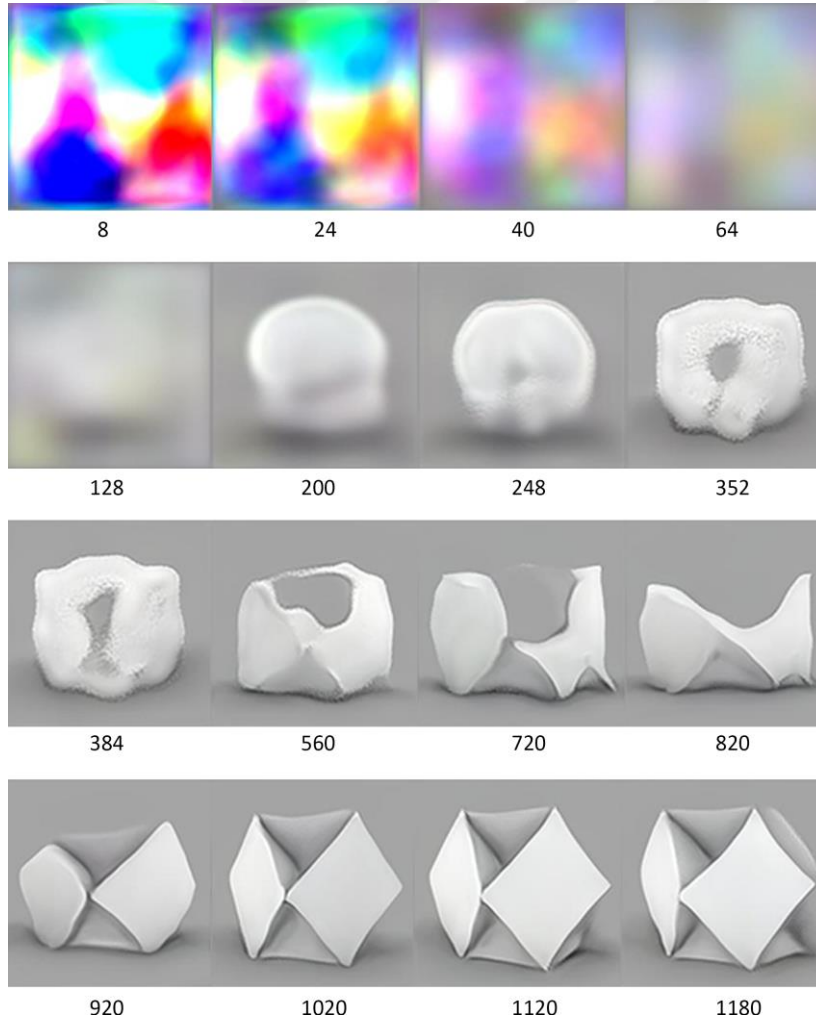
GAN modeli	Potansiyeller
DCGAN	Özniteliklerin farklılığını koruyabilmesi ve büyük bir eğitim seti ile son ürün odaklı otonom bir üretim uygulaması oluşturabilir.
Pix2Pix	Uzun hazırlık aşaması gerektirdiği için son ürün odaklı kalıcı uygulamalar oluşturabilir.
CycleGAN	- Erken tasarım aşamalarında hızlı stil transferi için kullanılabilir. - Hızlı eğitim edilebilir olması ile gerçek zamanlı çalışan uygulamalara aktarılabilir.
StyleGAN	- Mimari topoloji karar aşamalarında (tasarımın ilk aşamalarında) önerici model olarak kullanılabilir. - Öznitelik kaybının olmayacağı daha baskın veri setleri ile tarzlar arası geçiş yaparak yeni stil ve üretim biçimleri bulmayı sağlayabilir. - Form bulma problemlerinde tariflediği simülatif süreç, dijital fabrikasyonda çalışacak bir aracın otonomlaşmasını sağlayabilir.

Denenen modeller arasından StyleGAN, görüntü öğrenme ve yeniden yapılandırma simülasyonu için güçlü bir yeteneğe sahiptir. Girdi veri setindeki görüntülerin piksel özelliklerini absorbe etmede, oldukça benzer taklit çıktılar üretmede ve hatta tüm girdi verilerini tek bir birleşik "ortalama çıktı" olarak özetlemede iyidir. Ek olarak, stil

karıştırma veya stil dönüştürme yoluyla girdi verileri arasındaki çapraz etkiyi gerçekleştirebilir. Çeşitli mimari stil ve 3 boyutlu model görseli ile üretilen veri setlerinin kullanımı sonucu ürün dönüşümü diğer çıktı türleriyle karşılaştırıldığında en istikrarlı ve ideal ürünü oluşturduğu görülmektedir. Buna karşılık, stil karıştırma deneyinde öznelite kaybına sebep olarak beklenen sonucu üretememiştir.

Bütün bunlara ek olarak, StyleGAN epok aşamalarında tariflenen görsel üretim aşamaları, güçlü simülasyon yeteneği ile eksiltmeli-artırmalı üretim yöntemlerine kılavuz oluşturabilir (Şekil 5.2). Bu durum StyleGAN ile karşılaştırılan DCGAN için geçerli olmamaktadır. DCGAN her epok aşamasında farklı bir öznelite öğrenerek, bir önceki çıktıdan çok farklı bir görüntü ortaya koyabilmektedir.

StyleGAN’da bulunan bu simülasyon yeteneği fiziksel bir üretim aracının bilgisayarla görü modelleri ile nesne tanımanın önüne geçerek otonom bir şekilde üretim yapmasını sağlayacak bir yol oluşturabilme potansiyeline sahip bulunmuştur.



Şekil 5.2 : StyleGAN üçüncü deney (epok değerleri imajların altındadır).

6. SONUÇ VE ÖNERİLER

Araştırma, derin öğrenme algoritmalarından Çekişmeli Üretken Ağların mimarlıktaki potansiyellerini gömülü teori durum çalışması yöntemi ile irdelemiştir. Bu yöntem, durum çalışmalarında gizlenmiş bilgi setini ortaya çıkarmak üzerine ilerlemektedir (Scholz & Tietje, 2002). Bu şekilde oluşturulan deney süreçlerinden çıkan sonuçlarda sezgisel yöntemler ile öznitelik benzerlikleri ve epok aşamaları gözlemlenmiştir. ImageNET'in yaratıcılarından Frei-Frei Li'nin 'Yapay zekanın temeli görsel zekadır' sözü ile görsel üretim çalışmaları yapay zeka alanında önemli bir yere sahip olmaktadır. Yapay zeka yöntemi olan ÇÜA da veri setinin niteliklerine uygun görseller üretme becerisi sayesinde mimari kullanımda üretim problemlerine çözüm olabilmektedir.

Yapay zeka; mimari araştırmaya yeni araçlar getirerek, mimari tasarımın gizinde bulunan karmaşıklıkları ortaya çıkarmamıza olanak tanımaktadır. ÇÜA modelleri de mimarlıkta bulunan birçok olguyu bu şekilde incelenebilir kılmaktadır. Bu aşamada, gelişen modellere adapte olmak ve bir ötesine geçerek bu modellerle oluşacak mimari söylemlerin temelini sağlamlaştırmak önemli hale gelmektedir. Mimarlık disiplinine özgü bir yaklaşım oluşturmak için, bu sistemleri kabul etmekten öteye geçilmelidir. Önerilen potansiyeller de bunun için bir adım niteliğindedir. Bu modellerin tasarım ve üretim süreçlerine eklenebilir uygulamaları deneylerle keşfedilmiştir. Bununla birlikte halen bazı zorlukları vardır. 3 boyutlu veri ile çalışılmaması ve bazı donanım gereksinimleri henüz bahsedilen potansiyellerin tam olarak denenmesine ortam sağlayamamaktadır. Aynı zamanda 5. bölümde bahsedilen tüm bu potansiyellere rağmen çıktılara bakıldığında halen tamamlanmamış, öznitelik kayıpları olan veya gürültülü imajlar görülmektedir. Saxena & Cao (2020) bu durum için geliştirici adımların daha çeşitli optimizasyonlar ve yapılar oluşturularak geliştirilebileceğini önermişlerdir.

ÇÜA üretimlerinin değerlendirmesinde kalitatif yöntemlerin kullanıldığı söylenebilir. Bu yüzden yapılan deney çıktılarının iyi bir ayırt edici olan bizler tarafından bir anket ile değerlendirilmesi önerilmekte, bu algoritmaların ürettiği yapay imajların insan algısında ne kadar geçerli olduğu merak konusu olmaktadır.

Diğer modellerden kendini ayıran StyleGAN'ın, keşfedilen potansiyellerini daha iyi gösterebilmesi için form bulma problemlerinde daha karmaşık 3 boyutlu model veri setleri ile uygulanması önerilmektedir. Epok aşamalarında tariflediği simülasyon yeteneğinin otonom üretim araçlarına adaptasyonunu sağlayacak bir adım olması düşünülmektedir.

Ek olarak bu tür araştırmalarda, çalışmanın yöntemle dayalı başlaması ile çalışılan alandan uzaklaşmakta, veri bilimi odaklı geliştiği görülmektedir. Yürütülen araştırmada veri setlerinin yöntemle dayalı seçilmesi, durumun bu çalışma için de geçerli olduğunu göstermektedir. Yöntemle dayalı oluşturulan deneylerde alan bilgisinin eksik kalması, bu çalışmaların disiplinler arası olması gerektiğini söylemektedir. Bu noktada, ileri çalışmalarda konularla ilgili alanların bir arada çalışması önerilmektedir.

Yapılan deneylerle modellerin kullanımı için oluşturulmuş iş akışı bize aslında mimarların bu alanda daha detaylı uzmanlıklara evrileceğini göstermektedir. Bu algoritmaların çalışması için ciddi ölçüde veri gerekmektedir. Mimari özniteliklere sahip verinin olmaması, bir uzmanlığın burada oluşacağını bize söylemektedir. Burada mimarlık için geçerli özniteliklerin ne olacağını belirlenmesi önemli hale gelmektedir. Tez kapsamında, semantik ayrımlar ve yapısal ayrıntıların öznitelikleri oluşturduğu düşünülmüş ve deneylerde bu yönde ilerlenmiştir. Oluşacak bir diğer uzmanlık alanı ise üretim başarısının artması için gerekli yeni model yapılarının hazırlanması olacaktır. Şu an veri biliminin ihtiyaçlarına yönelik hazırlanan modeller kullanılmaktadır. Bu noktada, uzmanlık alanlarına özel problemlere yönelik modellerin oluşturulması ihtiyaç haline gelmektedir. Son evrilecek uzmanlık alanı olarak çıktıların değerlendirilmesi önemli bir adım olarak karşımıza çıkmaktadır. Bu yüzden değerlendirme yöntemlerinin çeşitlenmesi gerekmektedir.

Sonuç olarak yapay zeka; yakın gelecekte mimarların, günlük uygulamalarının kullanımında büyük ölçüde güçlenecektir. Böyle bir potansiyel gerçekleşmek üzere olduğundan, çalışma; kavramsal bir altyapının varlığını ortaya çıkarırken, tartışma için bir çerçeve sunmayı amaçlamıştır. Mimarları, yapay zeka ile etkileşime girmeye davet eden bu ağlar, aynı zamanda veri bilimcilerini bu alanı mimari bir araştırma alanı olarak görmeye davet etmektedir. Bu noktada umut vadeden çıktılardan ilki; tasarım anlayışına yönelik istatistiksel bir yaklaşımın, yapay zekanın mimari veri üretiminde potansiyel oluşturduğunu göstermektir. Bir dizi değişkeni optimize etmek için

kullanılan makineler yerine tasarım süreci boyunca tasarımcıya yardımcı ve öğrenen makineler kullanmak üretimleri daha ileriye taşıyacaktır.

Bu yolculuğun daha ilerisinde; ilgili uzmanlık alanlarını kapsayacak, tasarımcılar için her modele müdahale edilebilecek ve böylece insanlar ve makineler arasında gidip gelebilecek sistemlerin gelişmesi, mimarlık alanı için inanılmaz olanakların açılacağını göstermektedir. Böylece veri bilimi, mimarlık disiplinin yeni bir dizi araştırma metodolojisini benimseme olanağı sunabilir.





KAYNAKLAR

- Aish, R.** (2013). First build your tools. *Inside Smartgeometry: Expanding the Architectural Possibilities of Computational Design*, 36-49 .
- Alexander, C.** (1967). Notes on the Synthesis of Form. Harvard University Press.
- Alexander, C.** (1977). A pattern language: towns, buildings, construction. Oxford university press.
- Alpaydın, E.** (2010). Introduction to Machine Learning. Londra: The MIT Press.
- Borji, A.** (2019). Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*.
- Brownlee, J.** (2019). Applications of Deep Learning for Computer Vision. Erişim Tarihi: 13.05.2021, Erişim Adresi: <https://machinelearningmastery.com/applications-of-deep-learning-for-computer-vision/>.
- Brownlee, J.** (2019). How to use the UpSampling2D and Conv2DTranspose Layers in Keras? Erişim Tarihi: 13.05.2021, Erişim Adresi: <https://machinelearningmastery.com/upsampling-and-transpose-convolution-layers-for-generative-adversarial-networks/>.
- Brownlee, J.** (2019). How to Develop a Pix2Pix GAN for Image-to-Image Translation Erişim Tarihi: 15.05.2021, Erişim Adresi: <https://machinelearningmastery.com/how-to-develop-a-pix2pix-gan-for-image-to-image-translation/>.
- Chaillou, S.** (2019). AI+ Architecture: Towards a New Approach. Harvard University.
- Cortes, C., Vapnik, V.** (1995). Support-vector networks. Mach Learn 20, 273–297. <https://doi.org/10.1007/BF00994018>.
- Del Campo, M., Manninger, S., Sanche, M., Wang, L.** (2019). The Church of AI—An examination of architecture in a posthuman design ecology. Proceedings of the 24th CAADRIA Conference.
- Eastman, C. M.** (1975). Automated space planning. Artificial intelligence, 4(1), 41-64.
- Gatys, L. A., Ecker, A. S., Bethge, M.** (2014). Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.
- Goodfellow, I., Bengio, Y., & Courville, A.** (2016). Deep learning. MIT press.
- 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.
- 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, Mexico City, Mexico.

- 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, 1125-1134.
- Kantayya, S.** (Producer & Director). (2020). Coded Bias [Motion picture]. CA: Netflix.
- Karras, T., Laine, S., Aila, T.** (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 4401-4410.
- Kelleher, J. D. & Tierney, B.** (2020). Veri Bilimi. Tellekt Yayınları.
- Kirke, A. & Miranda, E. R.** (2010). Evaluating Mappings for Cellular Automata Music. Erişim Adresi: https://www.researchgate.net/publication/228363808_Evaluating_Mappings_for_Cellular_Automata_Music.
- Krizhevsky, A., Sutskever, I., and Hinton, G.** (2012). ImageNet classification with deep convolutional neural networks. In NIPS 2012.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P.** (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.
- Lynn, G.** (1999). Animate Form, Princeton Architectural Press, New York.
- Nilsson, N. J.** (2010). The quest for artificial intelligence: a history of ideas and achievements. Cambridge: Cambridge University Press.
- Radford, A., Metz, L., & Chintala, S.** (2016). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434.
- Rosenblatt, F.** (1959). Perceptron. SpringerReference. doi: 10.1007/springerreference_57806.
- Saxena, D., Cao, J.** (2020). Generative Adversarial Networks (GANs): Challenges, Solutions, and Future Directions.
- Say, C.** (2018). 50 Soruda Yapay Zeka, İstanbul, Bilim ve Gelecek.
- Scholz, R.W., Tietje, O.** (2002). Embedded Case Study Methods, Integrating Quantitative and Qualitative Knowledge. Thousand Oaks, CA: Sage.
- Sharma, S.** (2018). Beginner to Advance: Convolutional Neural Networks Made Easy. Erişim Adresi: <https://medium.com/deep-ai/beginner-to-advance-convolutional-neural-networks-made-easy-part-1-9a6e003dbaed>.
- Silva, T.** (2018). An Intuitive Introduction to Generative Adversarial Networks (Gans). FreeCodeCamp. Erişim tarihi: 27.03.2020, Erişim Adresi: <https://www.freecodecamp.org/news/an-intuitive-introduction-to-generative-adversarial-networks-gans-7a2264a81394/>.
- Solomonoff, R.** (1957). Endüktif Çıkarım Makinesi. IRE Sözleşme Kaydı. Bilgi Teorisi Bölümü, 2. bölüm s. 56–62.

- Stiny, G.** (2006). Shape grammar. MIT press.
- Turing, A. M.** (1950). Computing Machinery and Intelligence. Mind, LIX (236), 433–460. doi: 10.1093/mind/lix.236.433.
- URL1** (2016, Haziran 16). Erişim Adresi: <https://openai.com/blog/generative-models/>.
- URL2** (2018, Nisan 17). Erişim Adresi: <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>.
- URL3** (2020, Mayıs 10). Erişim Adresi: <https://theaisummer.com/gan-computer-vision-style-gan/>.
- URL4** (2018, Şubat 26). Erişim Adresi: <https://rubikscore.net/2018/02/26/introduction-to-convolutional-neural-networks/>.
- URL5** (t.y.). Erişim Adresi: https://gluon.mxnet.io/chapter14_generative-adversarial-networks/dcgan.html.
- Uzun, C.** (2020). Yapay Zeka ve Mimarlık Etkileşimi Üzerine Bir Çalışma; Üretken Çekişmeli Ağ Algoritması ile Otonom Mimari Plan Üretimi ve Değerlendirilmesi. ITU.
- Yoshimura, Y., Cai, B., Wang, Z., Ratti, C.** (2019). Deep Learning Architect: Classification for Architectural Design Through the Eye of Artificial Intelligence. In International Conference on Computers in Urban Planning and Urban Management.
- Zhang, L., Song, M., Liu, X., Sun, L., Chen, C., Bu, J.** (2014). Recognizing architecture styles by hierarchical sparse coding of blocklets. In Information Sciences.
- Zhu, J., Park, T., Isola, P., Efros, A. A.** (2017) Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. ICCV.



ÖZGEÇMİŞ

Ad-Soyad : Ruşen Eroğlu

ÖĞRENİM DURUMU:

- **Lisans** : 2017, İstanbul Teknik Üniversitesi, Mimarlık Fakültesi, Mimarlık

MESLEKİ DENEYİM VE ÖDÜLLER:

- 2016-2017, Norm Mimarlık, Mimar

YAYINLAR, SUNUMLAR VE PATENTLER:

- **Eroğlu, R.** 2019. Jerusalem Kulesi Üzerinden Yeniden Üretim, *13. Mimarlıkta Sayısal Tasarım Sempozyumu*, Kocaeli Üniversitesi, Türkiye.
- **Eroğlu, R., Karşı, M.** 2019. Form Generator As an Alternative Solution For Architectural Layout, *In Proceedings 22. Generative Art Conference, Exhibition, Live Performances*, Rome, Italy.
- **Bor, H.O., Eroğlu, R.** 2020. Allometri Üzerinden Kavramsal Araştırma ve Sistem Tasarımı, *14. Mimarlıkta Sayısal Tasarım Sempozyumu*, Karadeniz Teknik Üniversitesi, Türkiye.