

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY ZEKA VE MİMARLIK ETKİLEŞİMİ ÜZERİNE BİR ÇALIŞMA:
ÜRETKEN ÇEKİŞMELİ AĞ ALGORİTMASI İLE OTONOM MİMARİ PLAN
ÜRETİMİ VE DEĞERLENDİRMESİ**

DOKTORA TEZİ

Can UZUN

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

KASIM 2020

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

**YAPAY ZEKA VE MİMARLIK ETKİLEŞİMİ ÜZERİNE BİR ÇALIŞMA:
ÜRETKEN ÇEKİŞMELİ AĞ ALGORİTMASI İLE OTONOM MİMARİ PLAN
ÜRETİMİ VE DEĞERLENDİRMESİ**

DOKTORA TEZİ

**Can UZUN
(523152001)**

Bilişim Anabilim Dalı

Mimari Tasarımda Bilişim Programı

Tez Danışmanı: Prof. Dr. Meryem Birgül ÇOLAKOĞLU

KASIM 2020

İTÜ, Fen Bilimleri Enstitüsü'nün 523152001 numaralı Doktora Öğrencisi Can UZUN, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı “YAPAY ZEKA VE MİMARLIK ETKİLEŞİMİ ÜZERİNE BİR ÇALIŞMA:ÜRETKEN ÇEKİŞMELİ AĞ ALGORİTMASI İLE OTONOM MİMARİ PLAN ÜRETİMİ VE DEĞERLENDİRMESİ” başlıklı tezini aşağıda imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı : **Prof. Dr. M. Birgül ÇOLAKOĞLU**
İstanbul Teknik Üniversitesi

Jüri Üyeleri : **Prof. Dr. Sinan Mert ŞENER**
İstanbul Teknik Üniversitesi

Prof. Dr. Arzu Gönenç SORGUÇ
Orta Doğu Teknik Üniversitesi

Prof. Dr. Leman Figen GÜL
İstanbul Teknik Üniversitesi

Doç. Dr. Abdurrahman Tuğrul YAZAR
İstanbul Bilgi Üniversitesi

Teslim Tarihi : 19.10.2020
Savunma Tarihi : 13.11.2020





Sağlık çalışanlarına,



ÖNSÖZ

Doktora sürecinin uzun soluklu ve yoğun çalışmaları boyunca tezin oluşmasındaki en önemli aktörlerden danışmanım Prof. Dr. Meryem Birgül Çolakoğlu'na, beni cesaretlendirmeleri, büyük destekleri ve değerli yorumlarıyla tezin gelişmesinde büyük katkıları olan değerli jüri üyelerim Prof. Dr. Arzu Gönenç Sorguç ve Prof. Dr. Sinan Mert Şener'e teşekkür ederim.

Doktora eğitimimin başlangıcından sonuna kadar benim yanımda en önemli ve büyük bir destek olarak bulunan ve motivasyonlarını her fırsatta dile getiren aileme çok teşekkür ederim. Bu süreçteki ortak deneyimlerimiz ve problemlerimizle dertleşebildiğimiz birbirimizi en iyi anlayabilen bir diğer önemli motivasyon kaynağı İTÜ'den mezun olmuş ve İTÜ'de hâlâ eğitimlerine devam eden tüm arkadaşlarıma da teşekkür ederim. Nihat'a ve Berin'e de çok teşekkürler. Bu tez çalışmasının önemli bir parçası olan verisetinin oluşturulmasında büyük katkısı olan arkadaşım Halil Durmuş'a ve yapay zekanın teknik konularındaki kafa karmaşıklıklarımı konuşabildiğim İTÜ'de doktorasına devam eden bilgisayar mühendisi arkadaşım Arda İncaoğlu'na çok teşekkürler.

Doktora sürecinde aldığım TÜBİTAK Bideb2211-A Genel Yurt içi Doktora Burs Programı desteği, YÖK 100/2000 Doktora Burs desteği ve 2015-2017 yılları arasında TÜBİTAK 1001 projesi 115K515 numaralı proje kapsamında aldığım bursiyer desteği nedeniyle Türkiye Bilimsel ve Teknolojik Araştırma Kurumu ve Yükseköğretim Kurumu'na teşekkür ederim.

Bugün tezimi teslim ederken tüm dünya ortak bir problemle mücadele etmekte. Neredeyse dünyadaki yaşamı durdurmanın eşiğine getiren COVID-19 pandemisinin neden olduğu ortak bir dünya problemi. Bugün bu problem ile en önde savaşan sağlık çalışanlarına da teşekkürü bir borç bilirim.

Kasım 2020

Can Uzun
(Yüksek Mimar)



İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	vii
İÇİNDEKİLER	ix
KISALTMALAR	xi
SEMBOLLER	xiii
ÇİZELGE LİSTESİ.....	xv
ÖZET.....	xxi
SUMMARY	xxv
1. GİRİŞ.....	1
1.1 Problem Tanımı	2
1.2 Tezin Amacı	4
1.3 Tezin Yöntemi	5
1.3.1 Verisetinin Oluşturulması ve Değerlendirilmesi	6
1.3.2 Deneylerin Kurgusu.....	7
1.3.3 Üretim Çıktılarının Değerlendirilmesi.....	7
1.3.4 Çalışmanın Akış Diyagramı	8
2. LİTERATÜR ARAŞTIRMASI.....	11
2.1 Derin Öğrenme Algoritmalarındaki Kronolojik Gelişmeler.....	11
2.2 Mimari Plan Şeması Üretim Problemi ile İlgili Literatür.....	15
3. ÇEKİŞMELİ ÜRETKEN AĞ VE MİMARİSİ	21
3.1 Çekişmeli Üretken Ağ Tanımı.....	21
3.2 ÇÜA Algoritmasında Kullanılan Temel Kavramalar	22
3.3 Çekişmeli Üretken Ağ Mimarisi.....	29
3.4 DCGAN Mimarisi	31
4. ÇEKİŞMELİ ÜRETKEN AĞ EĞİTİM DENEYLERİ.....	33
4.1 ÇÜA ile Veri Setinin Test edilmesi: 1. Deney	34
4.2 ÇÜA ile Veri Setinin Test edilmesi: 2. Deney	36
4.3 ÇÜA ile Veri Setinin Test edilmesi: 3. Deney	38
4.4 ÇÜA ile Veri Setinin Test edilmesi: 4. Deney	41
4.5 ÇÜA ile Veri Setinin Test edilmesi: 5. Deney	44
4.6 ÇÜA ile Veri Setinin Test edilmesi: 6. Deney	46
4.7 ÇÜA ile Veri Setinin Test edilmesi: 7. Deney	48
4.8 ÇÜA ile Veri Setinin Test edilmesi:8. Deney	49
5. VERİ SETLERİNİN DEĞERLENDİRİLMESİ	53
5.1 1. Deney Veriseti Değerlendirilmesi	54
5.2 2. ve 3. Deney Veriseti Değerlendirilmesi	55
5.3 4. Deney Veriseti Değerlendirilmesi	56
5.4 5. Deney Veriseti Değerlendirilmesi	57
5.5 6. ve 7. Deney Veriseti Değerlendirilmesi	58
5.6 8. Deney Veriseti Değerlendirilmesi	59
6. ÇÜA ÇIKTILARININ DEĞERLENDİRİLMESİ.....	61

6.1 ÇÜA Algoritmasının Verimliliğinin Değerlendirilmesi	62
6.2 Mimari Plan Şeması Üretimlerinin Değerlendirilmesi	68
6.2.1 Hızlı Sahne Sınıflandırması (Rapid Scene Categorization).....	68
6.2.2 Palladyan Gramer Kurallarına göre Sınıflandırma	72
6.2.3 Mekân Sentaksı Yöntemi ile Sınıflandırma.....	76
7. SONUÇLAR.....	81
7.1 Bulgular	82
7.2 Tartışma	84
KAYNAKLAR.....	87
ÖZGEÇMİŞ	93



KISALTMALAR

CAD	: Computer Aided Design
CNN	: Convolutional Neural Network
DCGAN	: Deep Convolutional Generative Adversarial Network
FID	: Frechet Inception Distance
ELU	: Exponential Linear Unit
GAN	: Generative Adversarial Network
GPU	: Graphical Proccesing Unit
GRAPE-SGI	: GRAPE Shape Grammer Interpreter
MIT	: Massachusetts Institute of Technology
MNIST	: Modified National Institute of Standards and Technology
MOGA	: Multi-Objective Genetic Algorithm
RGB	: Red Green Blue
ReLU	: Rectifier linear unit
SVM	: Support Vector Machine
YZ	: Yapay Zeka



SEMBOLLER

$\mathbf{z}$	: latent space
$\mathbf{G}_{(z)}$	: üretici (generator) çıktısı
$\mathbf{J}^{(G)}$	: üretici fonksiyonun kayıp fonksiyon
$\mathbf{J}^{(D)}$	: ayırıştırıcı fonksiyonun kayıp fonksiyon
$\mathbf{D}$	: ayırıştırıcı (discriminator) fonksiyon
$\mathbf{G}$	: üretici (generator) fonksiyon
$\mathbf{x}$	: girdi (input)
$\boldsymbol{\theta}^{(D)}$	: ayırıştırıcı fonksiyonun kayıp fonksiyon parametresi
$\boldsymbol{\theta}^{(G)}$	: üretici fonksiyonun kayıp fonksiyon parametresi
$\mathbb{E}_x$	: hata değeri (Error)
∇_{θ_d}	: Gradient Descent
d^2	: FID değerin karesi
$\mathbf{C}_1$	: veri setinin kovaryans matrisi
$\mathbf{mu}_1$	: veriseti içerisindeki özniteliklerin dağılımı



ÇİZELGE LİSTESİ

Sayfa

Çizelge 7.1: DCGAN ile yapılan deneylerde kullanılan değişkenler.	81
Çizelge 7.1 (devam) : DCGAN ile yapılan deneylerde kullanılan değişkenler.	82





ŞEKİL LİSTESİ

Sayfa

Şekil 1.1 : Durum çalışmasının akış diyagramı	9
Şekil 2.1 : Yapay zeka kavramı ve alt kümeleri.....	11
Şekil 2.2 :Tümdengelim ile programlama.....	11
Şekil 2.3 : Tümevarım programlamada çıkarım	12
Şekil 2.4 : Tümevarım programlamada tahmin.....	12
Şekil 2.5 : Mimari plan şeması üretim problemi ile ilişkilenen literatür	16
Şekil 2.6 : Kronolojik olarak otonom mimari plan üretimi çalışmaları çıktıları.....	20
Şekil 3.1 : ÇÜA algoritmasında beklenen değerinin en iyilenme fonksiyonu.....	21
Şekil 3.2 : Dekonvolüsyonel katmandaki filtrenin ilerleyişi.....	25
Şekil 3.3 : MNIST veriseti sınıflandırısındaki filtrelerin görselleştirilmiş hali	25
Şekil 3.4 : Konvolüsyonel katmandaki stride hareketi; solda stride=1, sağda stride=2	26
Şekil 3.5 : Konvolüsyonel katmandaki padding oluşumu, solda padding=0, sağda padding =1	26
Şekil 3.6 : Sinir ağlarında kullanılan aktivasyon fonksiyonları	27
Şekil 3.7 : ÇÜA mimarisi şeması	29
Şekil 3.8 : DCGAN mimarisi şeması	31
Şekil 3.9 : Deneylerde kullanılan DCGAN mimarisi, üretici ve ayrıştırıcı katmanlarıyla birlikte.....	32
Şekil 4.1 : DCGAN mimarisinin kontrolü amacıyla MNIST veri seti ile eğitilmiş algoritmanın kayıp fonksiyon grafiği	33
Şekil 4.2: 1. deneyde kullanılan, veri arttırıma yöntemi kullanılarak çoğaltılmış orjinal Palladyan planları veriseti	34
Şekil 4.3 : 1. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.....	35
Şekil 4.4 : 1. deneyin DCGAN üretim çıktıları.....	35
Şekil 4.5 : 2. deneyde kullanılan, veri arttırıma yöntemi kullanılarak çoğaltılmış orjinal Palladyan planları veriseti (renklerin ters çevrildiği veri arttırımı yöntemi ile elde edilen veriler çıkarılmıştır)	36
Şekil 4.6 : 2. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.....	37
Şekil 4.7 : 2. deneyin DCGAN üretim çıktıları.....	38
Şekil 4.8 : 3. deneyde kullanılan, veri arttırıma yöntemi kullanılarak çoğaltılmış orjinal Palladyan planları veriseti (2. deneydeki veriseti kullanılmıştır)... ..	39
Şekil 4.9 : 3. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.....	40
Şekil 4.10 : 3. deneyin DCGAN üretim çıktıları.....	40
Şekil 4.11 : 4. deneyde kullanılan, veri arttırımı yöntemi ile sayısı artırılmış orijinal Palladyan planları ve GRAPE-SGI ile türetilen Palladyan plan şemalarından oluşan veriseti	42

Şekil 4.12 : 4. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği	43
Şekil 4.13 : 4. deneyin DCGAN üretim çıktıları.....	43
Şekil 4.14 : 5. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği	45
Şekil 4.15 : 5. deneyin DCGAN üretim çıktıları.....	45
Şekil 4.16 : 6. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği	47
Şekil 4.17 : 6. deneyin DCGAN üretim çıktıları.....	47
Şekil 4.18 : 7. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği	48
Şekil 4.19 : 7. deneyin DCGAN üretim çıktıları.....	49
Şekil 4.20 : 8. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği	50
Şekil 4.21 : 8. deneyin DCGAN üretim çıktıları.....	50
Şekil 5.1 : 1. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği	54
Şekil 5.2 : 2.ve 3. deneylerde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.....	55
Şekil 5.3 : 4. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği	56
Şekil 5.4 : 5. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği	57
Şekil 5.5 : 6. ve 7. deneylerde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.....	59
Şekil 5.6 : 8. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği	60
Şekil 6.1 : 1. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	63
Şekil 6.2 : 2. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	64
Şekil 6.3 : 3. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	64
Şekil 6.4 : 4. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	65
Şekil 6.5 : 5. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	65
Şekil 6.6 : 6. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	66
Şekil 6.7 : 7. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	67
Şekil 6.8 : 8. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik	67
Şekil 6.9: GRAPE-SGI üretimi plan şemaları.....	69
Şekil 6.10 : DCGAN üretimi plan şemaları	70
Şekil 6.11 : Hızlı sahne sınıflandırması için anket arayüzü	70
Şekil 6.12 : Hızlı sahne sınıflandırması anket yanıt sayıları	71
Şekil 6.13 : Hızlı sahne sınıflandırmasında yanıtların yüzdeleri	71
Şekil 6.14 : 7. deneydeki GRAPE-SGI ile üretilen veri setinden 25, DCGAN ile üretilen veriden 25 DCGAN (doğru), 25 DCGAN (yanlış) veri.....	73

Şekil 6.15 : 7. deney çıktılarının Palladyan gramer kurallarına ile sol altta GRAPE-SGI kullanılarak üretilen verinin (5,a), alt sıra ortada DCGAN (5, a) Palladyan kuralına göre doğru üretilmiş alt sıra sağda(5,a) yanlış üretilmiş veri gramer kuralları.....	74
Şekil 6.16 : 8. deneydeki GRAPE-SGI ile üretilen veri setinden 25, DCGAN ile üretilen veriden 25 DCGAN (doğru), 25 DCGAN (yanlış) veri.....	75
Şekil 6.17 : 8. deney çıktılarının Palladyan gramer kurallarına ile sol altta GRAPE-SGI kullanılarak üretilen verinin (5,a), alt sıra ortada DCGAN (5, a) Palladyan kuralına göre doğru üretilmiş alt sıra sağda(5,a) yanlış üretilmiş veri gramer kuralları.....	76
Şekil 6.18 : 7. deneylerin DCGAN üretimlerinin mekan sentaksı yöntemi ile değerlendirilmesi.....	78
Şekil 6.19 : 8. deneyin DCGAN üretimlerinin mekan sentaksı yöntemi ile değerlendirilmesi.....	79
Şekil 7.1 : Literatürde mimari plan şeması üretim çalışmalarında kullanılan yöntemler	84



YAPAY ZEKA VE MİMARLIK ETKİLEŞİMİ ÜZERİNE BİR ÇALIŞMA: ÜRETKEN ÇEKİŞMELİ AĞ ALGORİTMASI İLE OTONOM MİMARİ PLAN ÜRETİMİ VE DEĞERLENDİRMESİ

ÖZET

Bu tez kapsamında “Makineler tasarlayabilir mi?” sorusunun bir alt sorusu olan “Makineler mimari plan şeması üretebilir mi?” sorusu tezde araştırılması hedeflenen problem tanımını oluşturmaktadır. Tez mimari plan şemasının, günümüz yapay zeka algoritmaları ile, otonom olarak üretiminin kapsamını tartışmaya açmaktadır. Bu problem tanımı ile hedeflenen, günümüzde görsel işleme verimliliği ve gücü oldukça yüksek olan yapay sinir ağları içerisindeki algoritmaların üretken modelleri sınıfına dahil olan üretken çekişmeli ağlar ile mimari plan şeması üretim sürecinin ve üretim çıktılarının değerlendirilmesi bulunmaktadır.

Tez öncelikle yapay sinir ağlarının üretken çekişmeli ağlara kadar olan gelişimini kronolojik olarak, önemli dönüm noktaları ile birlikte özetler. Yapay siniri ağları tarihçesi sonrasında mimari plan şeması üretimine dair literatür araştırması sunulmaktadır. Mimari plan şeması üretimine dair literatür geçmişte bu problem çözümü için kullanılan yöntemleri göstermekle birlikte bugün mimari plan şeması üretim problemi için hangi yöntemlerin kullanımının verimli sonuçlar verdiğini de gösterebilmektedir. Bir diğer yandan literatür araştırmasına bakıldığında, mimarlık disiplininin yapay zeka ile etkileşime geçtiği problemin çoğunlukla mimari plan şeması üretimi olduğu görülmektedir. Bunun yanında literatürde mimari plan şeması üretim probleminin yapay zeka problemi sınıfına dahil olduğu da söylenmektedir. Literatür araştırmasına göre yapay zeka alanındaki yeni algoritmalar ile mimari plan şeması üretim problemi potansiyel bir araştırma alanı tanımlamaktadır. Bu nedenle tez çekişmeli üretken ağlar ile mimari plan şeması üretim süreci ve çıktılarını değerlendirmeyi hedeflemiştir.

Bu çalışma güncel bir fenomen olarak çekişmeli üretken ağları ele almaktadır. İçinde bulunduğu bağlam olarak da mimari plan şeması üretimi problemini hedef alır. Bu hedef, hem çekişmeli üretken ağların mimari plan şeması üretimi sürecinin keşfi hem de üretim çıktılarının değerlendirilmesini kapsamaktadır. Bu yönüyle tez durum çalışması yöntemi ile ele alınacaktır. Bu çalışmada veriseti ve ÇÜA mimarisi girdilerinin (bağımsız değişkenlerin), çıktılar / mimari plan şeması üretimleri (bağımlı değişken) üzerindeki etkileri de araştırılmaktadır. Bu yönüyle de tezin bir deney çalışması yöntemi kullandığı söylenebilir. Ancak tezdeki deneylerde kontrol grubu kullanılmamış, değerlendirme tüm deneylerin karşılaştırılması ile elde edilmiştir. Bir diğer yandan tez içinde kurgulanan deneylerin sayısı, algoritmanın başarılı olması ile deneylerin tamamlanması nedeniyle, rassaldır. Bu durumlar nedeniyle tez gerçek deneysel yöntem yerine yarı-deneysel yöntemle başvurmuştur. Bu nedenlerle tezin yönteminde yarı-deneysel durum çalışmasına başvurulmuştur.

Çekişmeli üretken ağ kullanılarak mimari plan şeması üretimleri sekiz deney sonrasında başarıya ulaşmış ve deneyler tamamlanmıştır. Eğitimlerde çekişmeli

retken aęlar ierisinde yer alan derin evriřimsel retken ekiřmeli aęlar (deep convolutional generative adversarial network / DCGAN) kullanılmıřtır. DCGAN mimarisi iindeki hiper parametreler DCGAN ile iliřkili literatrde belirtilen parametrelere uygun olarak kullanılmıřtır. Deneylerdeki eęitimler “python” dili ile Google’ ın bulut ortamında sunduęu “COLAB” ortamında yapılmıřtır. “COLAB” ortamının seilmesinin nedeni bulut ortamında saęladıęı GPU donanım servisidir. Deneylerdeki eęitim sırasında iřlem gcnn yksek olması nedeniyle bulut servislerinde sunulan GPU hizmeti algoritma eęitim srelerini kısaltabilmektedir. Bu nedenle algoritmanın eęitimi iin bulut ortamındaki “COLAB” kullanılmıřtır.

Deneylerdeki algoritma eęitimleri, piksel tabanlı orijinal Andrea Palladio planlarından oluřan piksel tabanlı veriseti ve Palladio gramer kuralları ile retilen sentetik, piksel tabanlı Palladio plan řemalarından oluřan veriseti ile gerekleřtirilmiřtir. Palladio orijinal planları ve plan řemalarının veriseti iin kullanılmasının sebebi; Palladio plan sisteminin kurallar ile aıklanabilen ve oęunlukla simetrik plan kurgusuna sahip olması nedeniyle, algoritma retim ıktıları ile veriseti arasında hızlı ve kolay bir deęerlendirmenin yapılabileceęi dřncesidir. Bu dřnce ile yalnızca Andrea Palladio planları ve plan řemaları deneylerdeki eęitimlerde kullanılmıřtır. Her bir deneyde kullanılan veriseti ierisindeki znitelikler (siyah ve beyaz pikseller) deęerlendirilerek raporlanmıřtır.

DCGAN retimlerinin veriseti ile iliřkili olması nedeniyle her bir deneyde kullanılan planlar ve plan řeması grsellerinden oluřan verisetlerindeki siyah ve beyaz piksellerin olasılık daęılım grafikleri izdirilmiřtir. Bu grafikler zerinden deneylerde kullanılan verisetlerinin deęerlendirilmesi yapılmıřtır. Deęerlendirme sonucunda son  deneyin piksel daęılım grafiklerinin dięer deneylerden daha dzgn olduęu ve bu nedenle deney retimlerinin de son  deneyde kalitesinin arttırdıęı sonucuna varılmıřtır. Eęitimlerde kullanılan veri artırımı yntemi ile elde edilen Palladio plan řeması verisetlerinin ise ekiřmeli retken aęların retim ıktılarını negatif etkiledięi gzlemlenmiřtir.

DCGAN zerinden yapılan son alıřma hem algoritma hem de mimari plan retimlerinin verimlilięinin deęerlendirildięi alıřmadır. DCGAN verimlilięi literatrde nerilen “Frechet Inception Mesafesi” ile deęerlendirilmiřtir. Plan řeması retimleri ise literatrde nerilen deęerlendirme yntemi “Hızlı Sahne Sınıflandırması” yanında “Palladian Grameri ile Sınıflandırma” ve “Mekan Sentaksı ile Sınıflandırma” yntemleri ile gerekleřtirilmiřtir. Literatr dıřında nerilen son iki deęerlendirme ynteminin bu tezde kullanılmasının sebebi, literatrde nerilen deęerlendirme yntemlerinin, retimleri deęerlendirmekten daha ok, algoritmanın retim verimlilięini deęerlendirmek iin kullanılıyor olmasıdır. Ancak mimari plan řeması retimi probleminde, bu problemin zm ynteminin arařtırılmasının yanında, retim ıktılarının deęerlendirme yntemlerinin de arařtırılması gereklidir. Herhangi bir grsel retiminde literatrdeki deęerlendirme yntemi kullanılabilse de mimari plan řeması retimlerinin deęerlendirilmesi iin mimarlık disiplini ierisinden yntemler kullanılabilir. Bu nedenle son iki deęerlendirme, Palladian gramer kuralları ile sınıflandırma ve mekan sentaksı ile sınıflandırma deęerlendirme yntemleri bu tezde yer almaktadır.

Bu tez alıřmasında, bugnn derin ęrenme algoritmalarından retken ekiřmeli aęlarla otonom plan řeması retimlerinin iřlevsel olarak deęil, sadece biimsel olarak gerekleřebildięi sonucuna varılmıřtır. Bir dięer yandan biimsel olarak verisetine benzer retimler gerekleřiyor olsa da DCGAN retimlerinin kontrol dıřı retildeęi

gözelemlenmiştir. Kontrol dışı üretimler mimari plan şeması üretimi probleminde, birçok varyasyondan oluşan, verisetindeki özniteliklere uygun dağılımdaki çıktılardan oluşan bir keşif alanı sunabilir. Ancak DCGAN algoritması ile doğrudan plan şeması üretimine güvenilerek kullanılması doğru sonuçlar vermeyebilir. Bu nedenle çekişmeli üretken ağlar içindeki DCGAN algoritmasının mimari plan şeması üretiminde üretken bir sistem olarak sadece bir keşif aracı niteliğinde olduğu söylenebilir. Ancak çekişmeli üretken ağların plan şeması üretimi problemi için büyük bir potansiyeli bulunmaktadır. Plan şeması üretim probleminin çözümü için gelecekte karşılaşılabileceğimiz, kontrolsüz üretimlerin önüne geçen, yeni üretken çekişmeli ağ mimarilerinin plan şeması üretim probleminde verimli bir araç olarak kullanılabilme potansiyeli bulunmaktadır.





A STUDY ON INTERACTION OF ARTIFICIAL INTELLIGENCE AND ARCHITECTURE: PRODUCTION AND EVALUATION OF ARCHITECTURAL PLANS WITH GENERATIVE ADVERSARIAL NETWORKS

SUMMARY

In the scope of this thesis, The question of "Can machine generate architectural plan schemes?" constitutes the definition of the problem aimed to be investigated in this thesis. The aim is to evaluate the architectural plan scheme generation process and the outputs of generative adversarial networks, which have high visual processing efficiency and power today.

This thesis summarizes the development of artificial neural network, and generative adversarial network chronologically, with important milestones. After the history of artificial neural networks, a literature review on the problem of architectural plan scheme production is presented. The literature on the production of architectural plan schemes shows the methods used for solving this problem in the past, besides, it also shows which recent methods to use for the architectural plan scheme production problem. On the other hand, looking at the literature research, it is seen that the problem in which the discipline of architecture interacts with artificial intelligence is mostly the production problem of architectural plan schemes. In addition, it is stated in the literature that the problem of architectural plan scheme production is included in the artificial intelligence problem class. According to literature research, the problem of architectural plan scheme production with new algorithms in the field of artificial intelligence defines a potential research area. Therefore, it is aimed to explore and evaluate the architectural plan scheme production process and outputs of generative adversarial networks.

This study deals with generative adversarial networks as a current phenomenon in the technology field. It targets the problem of architectural plan scheme production. This objective includes both the exploration of the architectural plan scheme production process of adversarial generative networks and the evaluation of production outputs. For this reason, the thesis will utilize the case study method. In this study, the effects of dataset and GAN architecture (inputs / independent variables) on outputs (architectural plan scheme productions / dependent variable) are investigated too. In this respect, the thesis utilizes an experimental study method. However, there is no control group in the experiments, and the evaluation was obtained by comparing all the experiments. Moreover, the number of experiments set up in the thesis is random. When the algorithm is successful, the experiments are completed. Because of these reasons, the study has applied to the quasi-experimental method instead of the real- experimental method. For these reasons, a quasi-experimental case study is utilized for the method of the thesis.

The architectural plan scheme production using the adversarial generative network is successful after eight experiments. During the experiments, for training the

algorithm, Deep convolutional generative adversarial network (DCGAN) is utilized. Hyperparameters following the parameters specified in the literature related to DCGAN are utilized for DCGAN architecture in the experiments. The training in the experiments is carried out with "python" in the "COLAB" environment offered by Google cloud services. The reason for choosing the "COLAB" environment is the GPU hardware service that Google provides in the cloud. Due to the need for high processing power during the training in the experiments, the GPU service offered in cloud services can shorten the algorithm training time. Therefore, "COLAB" is used for the training process of the algorithm.

In the experiments, the algorithm training process was carried out with a pixel-based dataset consisting of original Andrea Palladio plans and a dataset consisting of synthetic, pixel-based Palladio plan schemes generated with Palladian grammar rules. The reason why Palladio's original plans and plan schemes were used for the dataset is that the evaluation between the algorithm production outputs and the dataset that the algorithm was trained with could be made easier. The plan system can be explained with rules (Palladian grammar) and mostly has a symmetrical plan setup. For this reason, only Andrea Palladio plans and plan schemes were used in the algorithm training experiments. At the end of each experiment, the features (black and white pixels) in the datasets were evaluated and reported.

Since the DCGAN production efficiency is related to the dataset, the probability distribution graphs of the black and white pixels in the datasets consisting of the plans and schemes used in each experiment are drawn. The datasets used in the experiments were evaluated through these graphics. As a result of the evaluation, the pixel distribution graphics of the last three experiments show a more homogenous distribution than the other experiments, therefore the quality of the experimental productions increased in the last three experiments. It has been observed that the Palladio plan scheme datasets obtained by the data augmentation method affect the production output of adversarial productive networks negatively.

The last study conducted on DCGAN is a study evaluating the efficiency of both algorithm and architectural plan production outputs. DCGAN efficiency was evaluated with the "Frechet Inception Distance" suggested in the literature. Frechet inception distance is a measure that calculates the distribution differences between the algorithms outputs and the dataset in the training process. With frechet inception distance, we can say that DCGAN was successful in the last three experiments. This success comes from directly the organization of the datasets. As the datasets in the first five experiments were not clean enough, the algorithm failed and so frechet inception scores showed bad result. But the last three experiments have clean and tidy datasets. So the algorithm could generalize a meaningful result, readable architectural plan scheme outputs. The algorithm could learn the probability distribution of the Palladian plan scheme geometries. But still frechet inception distance is not saying anything about the efficiency of the architectural plan schemes, this method only shows the efficiency of DCGAN training process. To evaluate the efficiency of the architectural plan scheme production outputs, frechet inception distance score was insufficient.

The evaluation of the Plan scheme production was carried out with "Rapid Scene Categorization", "Classification with Palladian Grammar" and "Classification with Space Syntax" methods in addition to the evaluation methods of GAN, suggested in the literature. The reason why the last two evaluation methods is that the evaluation

methods suggested in the literature are used to evaluate the production efficiency of the algorithm rather than evaluating the production outputs. In the problem of producing architectural plan schemes, besides investigating the solution method of this problem, it is necessary to investigate the evaluation methods of production outputs. Although the evaluation method in the literature can be used in any visual production, when it comes to the evaluation of the architectural plan scheme production outputs, methods within the architectural discipline can be utilized to evaluate the production outputs of GAN. For this reason, the last two evaluation methods (classification with Palladian grammatical rules and classification evaluation methods with space syntax) are proposed for evaluation methods.

In this thesis, it was concluded that the production of autonomous plan schemes with generative adversarial networks, which is one of the most recent deep learning algorithms, is only based on shapes, not function or spatial order. On the other hand, it has been observed that DCGAN productions are produced out of control, even though the outputs of DCGAN are similar to the dataset. Out-of-control productions can present an area of exploration in the problem of architectural plan scheme production, consisting of many variations, and outputs in a distribution suitable for the attributes in the dataset. However, using DCGAN algorithm relying on direct plan scheme production may not give correct results. For this reason, it can be said that the DCGAN algorithm in adversarial generative networks is only a discovery tool as a productive system in architectural plan scheme production problems. However, adversarial generative networks have great potential for the problem of plan scheme generation as we can see its power in mimicking the geometrical entities within the dataset. The new generative adversarial network architectures, which prevent uncontrolled productions that we may encounter in the future, have the potential to be used as an efficient tool in the architectural plan scheme production problem.

,



1. GİRİŞ

Bugünün yapay zeka algoritmalarından derin öğrenme algoritmaları, donanım gücü ile birlikte, görüntü işleme problemlerinde güçlü bir araç olarak kullanılmaktadır. Krizhevsky, ve diğ., (2012) “*ImageNet Classification with Deep Convolutional Neural Networks*” adlı çalışmasında yüksek çözünürlükteki 1.3 milyon görseli, derin evrimsel sinir ağları (Deep Convolutional Neural Networks) kullanarak, 1000 farklı kategori için görsel sınıflandırma problemini çözebilmişlerdir. Bu çalışma derin öğrenme algoritmalarının görsel işleme gücünü gösterebilmektedir. Derin öğrenme algoritmaları, görsel işlemedeki bu gücü ile birlikte; görsel sınıflandırma, nesne tanıma, nesne segmentasyonu, görsel stil transferi, görsel renklendirme, görsel çözünürlük artırma, görsel sentezi ve görsel üretimi problemlerinde kullanılabilmektedir (Brownlee, 2019).

Görsel işleme problem tanımları içerisinde, görsel üretim, yapay zeka problemleri arasında önemli bir yerde güncel bir problem tanımı olarak bulunmaktadır. Görsel üretim probleminde derin öğrenme algoritmaları alt sınıfında bulunan *Çekişmeli Üretken Ağlar* (ÇÜA/ *Generative Adversarial Networks*) görsel üretimi probleminde verimli sonuçlar vermektedir. (Goodfellow, 2014). Karras ve diğ.’nin, (2020) ÇÜA kullanarak elde ettikleri sonuçlar, ÇÜA ile sentetik ancak gerçeğinden ayıramayacak netlikte görsellerin üretilebilmesinin mümkün olduğunu göstermektedir.

Görsel işleme gücü nedeniyle ÇÜA, tasarım pratiği içerisinde de potansiyel bir araştırma alanı tanımlamaktadır. Tasarım pratiği mimarlık disiplini özelinde incelendiğinde, mimarlık disiplininin, işlevsel bir görsel veri ürettiği söylenebilir. Mimarlık disiplini de ÇÜA’yı, görsel veri üretimlerinde, yeni bir araç olarak kullanabilir. Ancak mimarlık disiplini işlevsel bir görsel üretmesi nedeniyle salt görsel üretim, mimarlık için yeterli olamayacaktır. Bu nedenle mimarlık disiplini içerisinde, görsel üretimde çok güçlü bir algoritma olarak bulunan ÇÜA’nın, mimarlığın hangi pratiği içerisinde, nasıl kullanılacağı ve mimarlık pratiğine ne düzeyde etkisinin olduğu, kesin bir yanıtı bulunmayan bir soru olarak bulunmaktadır.

1.1 Problem Tanımı

“Bu metnin hedefinde, mimaride plan üretim problemini yapay zeka alanına tanıtmak bulunmaktadır. Plan şeması üretimi sadece ilginç bir problem tanımı olarak kalmamış, bu problemin çözümüne yönelik birçok uygulama da geliştirilmiştir. Metin boyunca plan üretimi probleminin yapay zeka problemleri ile benzerlikleri ve farklılıkları tartışılacaktır” (Eastmen, 1973).

Mimari plan üretimi mimarlık disiplini içerisindeki pratiklerden biridir. MÖ 2000 yılında dahi görebileceğimiz (Donald, 1962) bu pratik günümüzde devam etmektedir.

Plan şeması tasarımı (metin boyunca mimari plan şeması üretimi olarak da ifade edilecektir) ve plan çizimi (drafting) iki ayrı pratik olarak ele alınabilir. Günümüzde plan çiziminde geleneksel yöntemler (el ile çizim) yerine, gelişmiş ve karmaşık yazılımlar (CAD) kullanılarak tekrar eden iş yükü azaltılabilmekte ve çizimdeki metrik hassasiyet ile daha güvenilir çizimler üretilebilmektedir. Ancak plan şeması üretim pratiğinde gelişmiş yeni yazılımlar kullanılabilse de hala tasarımın erken evresinde geleneksel yöntemlere başvurulabilmektedir. Bunun nedeni plan şeması üretim probleminin, plan çiziminden farklı bir problem tanımına sahip olmasıdır. Plan çizimi problemi tasarım kararları alınmış bir problemin çizim sürecini içermektedir. Bu nedenle iyi tanımlı (well-defined) bir problem sınıfına dahil edilebilir. Ancak plan şeması üretimi hasta tanımlı (ill-defined) problem sınıfı içerisinde (Yoon, 1992).

Plan şeması üretimi probleminin hasta tanımlı yapısı bu alanı potansiyel bir araştırma alanı haline getirmektedir. Shaviv (1986), CAAD alanına giren herhangi bir mimarın bu temel mimari soruna bir çözüm bulma sorumluluğunu hissettiğini belirtmiştir. Shaviv temel mimari sorun olarak plan şeması üretimini ifade etmiştir. Lobos ve Donath (2010), plan şeması üretimi tekniği için verimli bir sistem elde etmek amacıyla birçok araştırmanın olduğunu; araştırmalarda, üretken sistemler, kısıt tabanlı sistemler, biçim gramerleri ve uzman sistemler gibi çeşitli yöntemler kullanıldığını belirtmiş ve birçok araştırma olmasına rağmen mimarların kullanımına uygun bir aracın üretilmediğini ifade etmişlerdir.

Eastmen’in (1975) mimari plan üretim probleminin yapay zekanın problem sınıfına ait olduğunu bildirmesi, bu problemin önemini ortaya koymaktadır. Hem henüz

problemin net bir çözümünün bulunamaması, hem de problemin yapay zeka içerisinde de önemli bir yer bulması nedeniyle bu alan önemini korumaktadır.

Mimari plan üretimi problemi eski bir problem olsa da, hala güncel bir problemidir. Teknolojinin yeni araçları, bu problem çözümü için yeni yöntemler sunmaktadır. Yeni araçlardan biri de teknolojinin güncel meselesi haline gelmiş olan yapay zeka algoritmalarıdır. Mimari plan şeması üretimi probleminin yapay zeka algoritmalarındaki yeni yöntemler ile çözümlenebilmesi yeni bir araştırma alanı olarak karşımıza çıkmaktadır.

Giriş bölümünde ÇÜA'nın mimarlık disiplini içerisinde hangi pratikte kullanılacağına belirlenmesinin gerekli olduğu söylenmiştir. Yapay zekanın ilksel araştırmalarında odaklanılan problemler, dama oyunu veya satranç oyunu olmuştur. Satranç ve dama oyunlarında kurallar; mekansal, görelî konumlar üzerinden oluşturulması bakımından mimarideki mekansal düzenlemeler arasında bir analogi kurulabilir. Bu düşünce sistemi içerisinde *mimari plan şeması üretim probleminin* Eastmen'in (1975) da açıklaması ile birlikte, *mimarlık disiplini dışında, yapay zeka disiplini içerisinde de bir problem olarak bulunduğu* söylenebilir. ÇÜA her ne kadar görsel üretim odağında bir algoritma olsa da bu aracın mimari plan şeması üretim yöntemi ve çıktıları değerlendirilmelidir.

Turing (1950) "Computing Machinery & Intelligence" adlı metninde "*Makineler düşünebilir mi? (Can Machines Think?)*" sorusunu sormuştur. Bugün elimizdeki işlem gücü yüksek donanımlar ve görsel veri üretiminde güçlü olan ÇÜA vb. algoritmalar ile, *Makineler tasarlayabilir mi? (Can machines design?)* sorusu sorulabilir. "*Makineler tasarlayabilir mi?*" sorusunun bir alt sorusu olarak da "*Makineler mimari plan şeması tasarlayabilir mi?*" sorusunun yanıtı ÇÜA algoritması ile aranacaktır. Tezin başlığı da makinelerin mimari plan üretip üretemeyeceğinin tartışmasının açılması amacıyla *otonom plan üretimi* olarak adlandırılmıştır. Bu nedenlerle tezin problem tanımında *ÇÜA ile mimari plan şeması üretiminin gerçekleştirilmesi ve üretimlerinin değerlendirilmesi* bulunmaktadır.

1.2 Tezin Amacı

Bu tezde çekişmeli üretken ağlar ile mimari plan şeması üretimleri gerçekleştirilecek ve üretim çıktıları değerlendirilecektir. Yapılacak olan bu çalışmada genel ve özel olmak üzere iki amaç belirlenmiştir.

Genel amaç güncel teknolojik gelişmelerin mimarlar tarafından tartışılabilmesi ve kullanılabilmesini hedefler. Özel amaç ise bu tezin büyük bir parçası olan teknolojik gelişmesinin çalışma süreci ve çıktılarının değerlendirilmesi ile ilgilidir.

Genel amaç ile hedeflenen Norman'ın (2013), belirttiği kavramsal model düzeyinde, mimarların yapay zeka süreçlerinden haberdar olmasıdır. Norman (2013), *kavramsal model* ile birlikte, bir sistemin nasıl kullanılacağını bilgisini ifade etmektedir. Başka bir deyişle tezin genel hedefinde mimarların yapay zekanın güncel algoritmalarının kullanılma süreçleri ve sonuçlarından haberdar olması hedeflenmektedir.

Mimarların güncel teknolojik gelişmelere adapte olmasının önemi Charles Eames'in dönemin *Design Journal* editörü Ian McCallum'a yazdığı mektup ile desteklenebilir;

“Elbette, yaratıcısına oldukça fazla sorumluluk yükleyen kavramın (teknoloji) kışkırtılması nedeniyle bireysel ve yaratıcı süreçlerin kaybedilmesi korkusu vardır. Ancak şundan emin olabiliriz ki; yakın geleceğin binaları ve toplumları bu yeni kavramlardaki gelişmeler ile planlanacaktır. Bu planlamaların mimarlar tarafından yapılıp yapılmadığı, mimarın bu tür teknolojik araçları kullanmaya hazır olup olmaması ile ilgili olacaktır” (Eames, 1954).

Eames'in bu açıklaması ile birlikte mimarların güncel teknolojik gelişmeleri takip etmesinin mimarlık disiplini için bir sorumluluk oluşturduğu söylenebilir. Bu nedenle tezin hedefinde olan ve üzerinde durulan kavram bugünün en önemli teknolojik gelişmelerinin olduğu ve araştırmaların geliştirildiği disiplin olan yapay zeka kavramı olmuştur. Yapay zeka algoritmaları içerisinde *Çekişmeli Üretken Ağlar* kullanılarak mimari plan şeması üretimi ve değerlendirilmesi süreçlerinin mimarlık disiplinine tanıtılması bulunmaktadır.

Tezin özel amacı ÇÜA algoritmasının, mimarlık pratiklerinden olan, *mimari plan şeması üretimi* süreci ve çıktılarının değerlendirilmesi bulunmaktadır.

Önerilen bu çalışma ile;

- ÇÜA mimarisinin, plan şeması üretimi için nasıl geliştirilmesi gerektiği sorusunun yanıtlanması,
- ÇÜA ile plan şeması üretiminde verisetinin seçimi ve veriseti düzeninin nasıl geliştirilmesi gerektiği,
- ÇÜA üretimlerinin çıktılarının mimari plan organizasyonu bağlamında işlerliği ve ÇÜA'nın üretim verimliliğinin değerlendirilmesi,
- Tüm bu sorular sonrasında ÇÜA'nın mimari plan şeması üretiminde üretken bir sistem olarak kullanımının değerlendirilmesi,

hedeflenmektedir.

1.3 Tezin Yöntemi

Tez literatür araştırmasıyla başlamaktadır. Literatür araştırması kısmında derin öğrenme algoritmalarının çekışmeli üretken ağlara kadar olan gelişimi kronolojik olarak önemli noktaları açıklanarak aktarılmıştır. Bu bölüm derin öğrenmenin tarihçesini özetleyen bir bölüm olarak nitelendirilebilir. Literatürün ikinci kısmında ise mimari plan şeması üretim problemi ile ilişkilenen literatür sunulmuştur. Daha ÇÜA ile mimari plan şeması üretimi deneyleri ve değerlendirmeleri bölümleri aktarılmıştır.

Tezde mimari plan şeması üretimi hedefiyle ÇÜA algoritmalarının mimarisinin, ÇÜA'nın eğitileceği verisetinin özelliklerinin ve eğitime olan etkilerinin anlaşılabilmesi bunun yanında eğitim çıktılarının değerlendirilebilmesi için *yarı-deneysel durum çalışmasına (quasi-experimental case study)* başvurulmuştur.

Yin'e (1994) göre durum çalışması, güncel bir fenomeninin, içinde bulunduğu bağlam ile arasındaki ilişki belirsiz olduğunda, fenomeni birden fazla kaynaktan topladığı kanıtlarla ampirik olarak açıklama yöntemidir. Aytaçlı'ya (2012) göre değerlendirmelerin yapıldığı araştırmalarda durum çalışması yöntemine başvurulur. Bunun yanında, Yin (2003) açıklayıcı (explanatory), tanımlayıcı (descriptive), veya keşif (exploratory) hedefleriyle durum çalışmasına başvurulabileceğini söylemiştir.

Deneysel yöntem ise bağımsız değişkenlerin bağımlı değişkenlerin üzerindeki etkisinin gözlemlendiği çalışmalarda kullanılır (Rogers & Revesz, 2020). Aynı tanım deneysel yöntem içerisinde tanımlanan gerçek deneysel (true experimental) ve yarı-

deneysel (quasi-ezperimental) yöntemler için geçerlidir. Gerçek deneysel çalışmada deney grupları, kontrol deney grubu ile karşılaştırılır. Bir diğer gerçek deneyde, deney sayısı ve deneylerde kullanılacak örneklem sayısı önceden belirlenir, rassal olamaz. Yarı-deneysel yöntemlerde ise kontrol grubu yerine deney çıktıları birbirleri arasında karşılaştırılır ve deney sayısı ile deneylerde kullanılan örneklem sayısı rassal olabilir (Rogers & Revesz, 2020).

Bu çalışma güncel bir fenomen olarak çekişmeli üretken ağları ele almaktadır. İçinde bulunduğu bağlam olarak da mimari plan şeması üretimi problemini hedef alır. *Tezde amaç çekişmeli üretken ağların mimari plan şeması üretimi problemi ile arasındaki ilişkinin değerlendirilmesidir.* Bu yönüyle tez durum çalışması yöntemi ile ele alınacaktır. Bu çalışmada veriseti ve ÇÜA mimarsî girdilerinin (bağımsız değişkenlerin), çıktılar / mimari plan şeması üretimleri (bağımlı değişken) üzerindeki etkileri de araştırılmaktadır. Bu yönüyle de tezin bir deney çalışması yöntemi kullandığı söylenebilir. Ancak tezdeki deneylerde kontrol grubu kullanılmamış, değerlendirme tüm deneylerin karşılaştırılması ile elde edilmiştir. Yapılan deneylerdeki örneklem sayısı rassal bir şekilde belirlenmiş, yapılan deneylerde elde edilen çıktıların beklentiyi karşılaması sonucunda deneyler tamamlanmıştır. Bu durumlar da bu tez içerisinde yarı-deneysel yöntem kullanıldığını göstermektedir.

Deneylerde kullanılan verinin değiştirilmesi ile deney sonuçlarının değişimi gözlemlenmiştir. Bu uygulama durum çalışmalarındaki veri çeşitleme yöntemi (Aytaçlı, 2012) ile açıklanabilir. Deneylerden elde edilen verinin yeterli örneklem sayısını oluşturmaması nedeniyle *istatistiksel bir genelleme yerine analitik bir genellemeye* (Aytaçlı, 2012) başvurulmuştur. Bu durum da yarı-deneysel yöntem dışında durum çalışmasının da bu çalışmada bulunduğunu göstermektedir.

Yukarıdaki tüm sebepler nedeniyle; tez, deneylerin oluşturulma şekli ile yarı-deneysel yöntem, deneyin yapılma hedefi ve çıktılarının değerlendirmesi yönüyle durum çalışması yöntemine yakın olarak değerlendirilebilir. Bu nedenle tezin yönteminde hibrit bir yöntem olarak *yarı-deneysel durum çalışması (quasi-experimental case study)* yönteminin kullanıldığı söylenebilir.

1.3.1 Verisetinin Oluşturulması ve Değerlendirilmesi

Mimari plan şeması üretimi hedefi ile deneylerde algoritmanın eğitimi için kullanılan verisetleri *plan şemalarından* oluşturulmuştur. Plan şemaları algoritmanın doğru

genellemeler yapabilmesi için tek bir örneklem uzayı (tek bir mimarın plan şemaları) içerisinde seçilmesi kararlaştırılmıştır. Bu örneklem uzayının Andrea Palladio planları veya plan şemaları olmasına karar verilmiştir. Verisetinde Andrea Palladio planlarının kullanılmasının sebebi ise plan düzeninde simetrik ve kural tabanlı bir kurgunun olmasıdır. Bu sistematik kurgunun kuralları, Stiny ve Mitchell (1978) tarafından Palladian grameri ismi ile açıklanmıştır. Kural tabanlı durum hem eğitimdeki verimi denetlemek ve verisetindeki veri sayısını artırmak için kullanılabilir, hem de üretim çıktılarının veriseti ile karşılaştırılarak değerlendirilmesini kolaylaştırabilecektir. Bu nedenlerle Andrea Palladio planlarının veya plan şemalarının eğitim veriseti olmasına karar verilmiştir. Andrea Palladio planları orijinal plan çizimlerinden elde edilmiş ve toplamda birbirini tekrar etmeyen 125 plana ulaşılmıştır. Andrea Palladio plan şemaları ise Palladian grameri kuralları ile üretilen birbirini tekrar etmeyen 320 plan şemasından oluşan ve 150 plan şemasından oluşan iki verisetinden oluşmaktadır. 320 Palladio plan şeması bulunan verisetinde 3x3, 3x4, 3x5, 5x3, 3x7 ızgara düzeninde plan şemaları farklı sayılarda, 150 plan şemalı verisetinde ise her bir grid düzeninden 30 adet olmak üzere eşit sayıda grid düzenlerinden oluşan Palladio plan şemaları üretilmiştir. Bu verisetleri dışında aynı veriler kullanılarak veri arttırımı yöntemleri (döndürme, yansıtma, deform etme, renk değiştirme) ile de farklı sayıda verisetleri elde edilmiştir. Farklı verisetlerinin oluşturulması doğrudan eğitim veriminin artması için denenen veri çeşitlemesidir.

Her bir veriseti kendi içinde değerlendirilmiştir. Bu değerlendirmeler ile eğitimlerdeki başarımın ya da başarısızlığın sebepleri tahmin edilmeye çalışılmıştır.

1.3.2 Deneylerin Kurgusu

Tüm deneyler öncesinde algoritmanın işlerliğini denemek üzere bir pilot çalışma yapılmıştır. İşlerliğinden emin olunan algoritma ile eğitimler yapılmıştır. Eğitim deneyleri algoritmanın doğru üretimler yapmasıyla birlikte sonlandırılmıştır. Algoritma ile çeşitli verisetleri ve farklı algoritma mimarileri kullanılarak 8 deney gerçekleştirilmiştir.

1.3.3 Üretim Çıktılarının Değerlendirilmesi

Üretim çıktıları değerlendirilirken ÇÜA literatüründe bulunan ÇÜA nitel ve nicel değerlendirme yöntemlerinden, hızlı sahne sınıflandırması (rapid scene

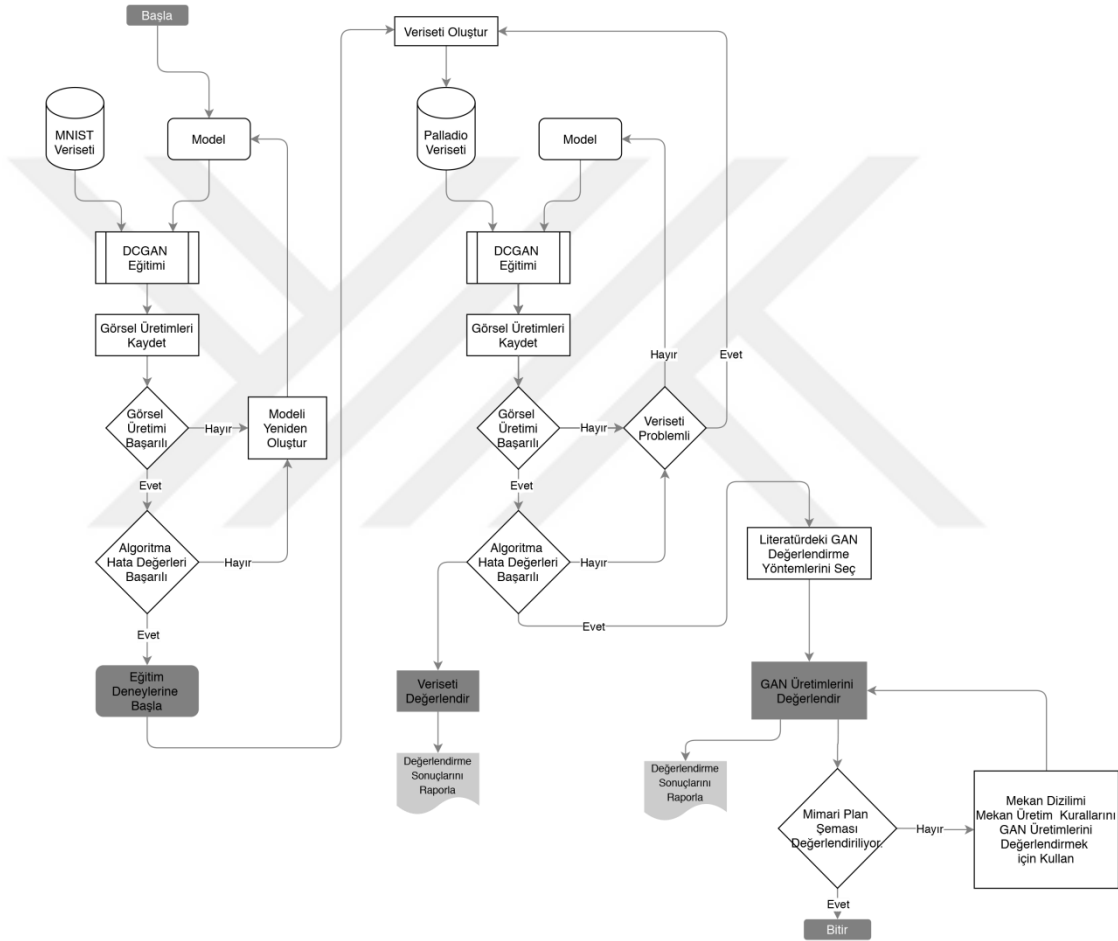
categorization) ve FID skoru değerlendirme yöntemlerine başvurulmuştur. FID skoru değerlendirme yöntemi sadece görsel benzetim ve çözünürlük kalitesi ile ilgili bir değerlendirme sonucu vermektedir. FID skoru, ÇÜA'nın çalışma verimini değerlendirebilmekte ancak üretilen plan şemalarının değerlendirmesini yapmamaktadır. Değerlendirme bölümünde, ÇÜA algoritmasının değerlendirilmesi dışında, üretilen mimari plan şemalarının değerlendirilmesi amacıyla, hızlı sahne sınıflandırması, Palladian gramer kuralları ve mekan sentaksı yöntemleri ile de değerlendirmeler yapılmıştır.

1.3.4 Çalışmanın Akış Diyagramı

Şekil 1.1 uygulanan deneyler ve sonuçlarının değerlendirme süreci için kurgulanan akış diyagramını göstermektedir.

1. ÇÜA ile plan şeması üretimi deneylerine başlamadan önce algoritmanın çalışmasının denenmesi için bir pilot eğitim çalışması gerçekleştirilir. Bu pilot çalışmada MNIST veriseti kullanılmıştır. (Derin sinir ağlarının eğitim verimi, verisetinin kalitesi ve sinir ağının mimarisi ile değişebilmektedir. Kullanılacak ÇÜA mimarisinin denetlenebilmesi için veriseti kalitesi literatürde doğru olarak kabul görmüş, MNIST veriseti ile pilot bir eğitim yapılmıştır.)
2. Palladio planları veya plan şemalarından oluşan veriseti oluşturulur. (Yeni veriseti üretimi var olan verisetinin veri arttırımı yöntemleri ile çoğaltılması ya da sentetik olarak Palladian gramer kuralları kullanılarak oluşturulan verisetlerini içerir.)
3. Model (algoritmadaki değişkenler) düzenlenir ve eğitime uygun hale getirilir.
4. Algoritma eğitimi gerçekleştirilir.
5. Eğitimde kullanılan veriseti değerlendirilir ve raporlanır.
6. Eğitim sonucunda elde edilen mimari plan şeması görselleri kaydedilir.
7. Görsel üretimler başarısız ise (plan şeması okunabilir durumda değil ise) 2. adıma veya 3. adıma geçilebilir.
8. Görsel üretimler başarılı ise algoritmanın hata değerleri kontrol edilir.
9. Algoritma hata değerleri problemlidir ise 2. adıma veya 3. adıma geçilebilir.
10. Algoritma hata değerlerinde bir sorun yoksa algoritmanın plan şeması üretimleri değerlendirilir.

11. Değerlendirme sürecinde literatürdeki ÇÜA değerlendirme yöntemleri kullanılır, sonuçları raporlanır.
12. Değerlendirme yöntemleri görsel kaliteye odaklanıyorsa (mimari plan şeması değerlendirilmiyorsa, çözünürlük ve biçimsel benzerlik konularına odaklanılıyorsa), hızlı sahne sınıflandırması, mekan üretimi kuralları (Palladian grameri) ve mekan dizilimi (Space Syntax) ile üretim çıktıları değerlendirilir, sonuçları raporlanır.



Şekil 1.1: Durum çalışmasının akış diyagramı.

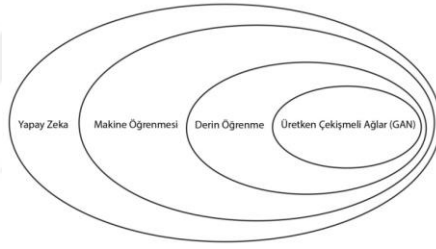


2. LİTERATÜR ARAŞTIRMASI

2.1 Derin Öğrenme Algoritmalarındaki Kronolojik Gelişmeler

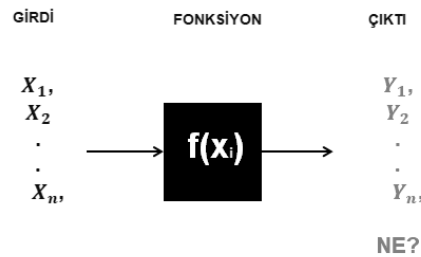
Bu bölümde yapay zekanın problem alanı kısaca açıklandıktan sonra, bu problem alanına yönelik çözüm yöntemlerinden olan ilişkiseli yapay zeka içerisinde gelişen, çekişmeli üretken ağların da içerisinde yer aldığı, derin öğrenmenin özet bir tarihçesi aktarılacaktır.

Yapay zeka olarak tanımlanan kavram makine öğrenmesi yöntemlerini kapsamaktadır. Derin öğrenme ise makine öğrenme yöntemleri içerisinde bulunmaktadır. Çekişmeli üretken ağlar da derin öğrenmenin üretken algoritmaları arasında sınıflandırılabilir (Şekil 2.1).



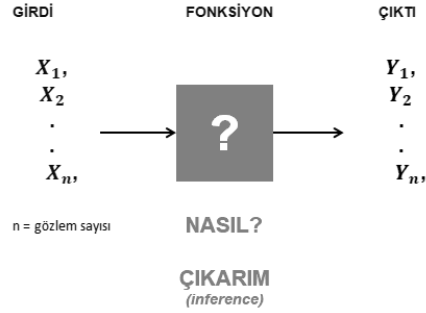
Şekil 2.1: Yapay zeka kavramı ve alt kümeleri.

Yapay zeka algoritmaları tümdengelim yöntemi ile *açık (explicit)* bir programlama yerine tümevarım yöntemi ile *örtük (implicit)* bir programlama süreci içermektedir. Girdi, girdinin işlendiği fonksiyon ve çıktı algoritmik sistemlerin yapısını oluşturmaktadır. Tümdengelim yönteminin kullanıldığı algoritmalarda girdinin işleneceği fonksiyon *apaçık* şekilde açıklanır (Şekil 2.2). Girdiyi işleyen fonksiyonun çıktısı, fonksiyonda bir hata yok ise, beklenmedik bir durum ile sonuçlanmaz.



Şekil 2.2 : Tümdengelim ile programlama.

Örtük programlamada ise girdinin işleneceği fonksiyon tanımlanmaz, fonksiyonun algoritma tarafından bulunması beklenir. Bu tür programlamada girdi tanımlanır, çıktı ya tanımlanır ya da çıktının da algoritma tarafından belirlenmesi beklenir. Makine öğrenmesi algoritmalarının fonksiyonu bulması işlemi *çıkarım (inference)* olarak adlandırılır (Şekil 2.3).



Şekil 2.3: Tümevarım programlamada çıkarım.

Örtük programlamada algoritmanın bulduğu fonksiyon ile çıktının belirlenmesi *tahmin (prediction)* olarak isimlendirilir (Şekil 2.4).



Şekil 2.4: Tümevarım programlamada tahmin.

Yapay zeka algoritmaları çıkarım ve tahmin problemleri çözümünde kullanılmaktadır. Çıkarım ve tahmin yapma sürecinde yapay zeka algoritmalarının içinde sembolik (symbolic) ve ilişkiseli (connectionist) olarak isimlendirilen iki temel yaklaşım bulunmaktadır (Minsky, 1991). Bu temel iki yaklaşım ilk kez, 1955 yılında gerçekleştirilen *Öğrenen Makineler Oturumu'nda (Session on Learning Machines)* tartışılmıştır. Oturum katılımcılarından, Oliver Selfridge, Willis Ware ve Gerald Dinneen donanım gücü ile insan bilişinin taklit edilmesinin mümkün olduğunu savunarak insan sinir sistemine odaklanılmasını önermiş, Allen Newell ve Herbert Simon ise sebepler hiyerarşisinin taklit edilmesinin doğru olduğunu savunmuştur (Nilsson, 2010). İnsanın sinir sistemine odaklanan yaklaşımı ilişkiseli,

sebepler hiyerarşisini mantık önermeleri ile açıklama yöntemi ise sembolik yapay zeka olarak tanımlanmaktadır (Minsky, 1991).

Bugünün derin öğrenme algoritmaları da insan sinir sisteminin çalışma prensipleri odağında gelişen ilişkiseli yapay zeka yöntemleri içerisinde bulunmaktadır. İlişkiseli yapay zeka çalışmalarının ilk adımını Warren McCulloch & Walter Pitts (1943) “A Logical Calculus of the Ideas Immanent in the Nervous Activity” adlı çalışma oluşturmaktadır. Warren McCulloch & Walter Pitts (1943), ilk kez yapay bir sinir ağı modelini mantık kapıları (logic gate) ile açıklamışlardır. McCulloch & Pitts (1943), nöronun ateşlemesini *ya hep ya hiç (all or none)* kuralına bağlayarak *boolean* kuralları ile nöronun ateşleme sisteminin modelini oluşturmuşlardır.

McCulloch & Pitts’in (1943) ilk yapay sinir ağı modeli sonrasında, psikolog Donald O. Hebb (1949), “The Organization of Behaviour” isimli kitabında beyindeki öğrenme sürecine dair çalışmalar yapmıştır. Hebb’in tanımladığı öğrenme sistemi *Hebbian Öğrenme (Hebbian Learning)* olarak bilinmektedir. Hebbian öğrenme sisteminde, aynı anda ateşleyen nöronlar kümelenme (*cells fire together wire together*) eğilimindedir. Aynı uyaran başka bir sefer daha geldiğinde aynı nöronlar daha kuvvetli bağlar oluşturmaktadır. Hebbian öğrenme sistemindeki bu açıklamalar, yapay sinir ağları ile oluşturulan derin öğrenme sisteminin gelişmesinde önemli rol oynamıştır.

1955 yılında öğrenen makinelerle ilgili ilk oturum gerçekleştirilmiş hemen bir yıl sonrasında 1956’da Dartmouth yaz atölyesinde yapay zekanın babaları olarak anılan isimler yapay zeka ile ilgili düşüncelerini paylaşmışlardır (Nilsson, 2010).

Psikolog Frank Rosenblatt (1959), ilk yapay sinir ağını geliştirmiştir. Perceptron tahminleri ile sınıflandırma problemi çözülebilmektedir. Ancak perceptron doğrusal sınıflandırma problemleri dışında kalan problemleri çözememektedir. Bunun yanında perceptron tek katmanlı bir yapay sinir ağı önerisi olmuştur. Perceptron sonrasında 1960 yılında Widrow & Hoff Uyarlanımsal Doğrusal Ağ (Adaptive Linear Network / ADALINE) ile yapay sinir ağlarındaki ağırlık değerlerini ayarlanabilir hale getirmişlerdir (Nilsson, 2010).

Minsky & Papert (1970) Perceptron’un doğrusal olmayan problemleri çözemeyeceğini açıklamalarıyla birlikte 1960’ların sonuna kadar başarılı giden ilişkiseli yapay zeka çalışmaları yavaşlamış ve sembolik yapay zekanın başarılı bir

alışma alanı olan uzman sistemler (expert systems) gelişmeye başlamıştır (Domingos, 2015).

Minsky (1991), sembolik ve ilişkilselci yapay zekayı karşılaştırdığı metninde, her iki yapay zeka yaklaşımının da avantaj ve dezavantajları olduğunu belirtmiştir. Yine de ilişkilcelilerin yönteminin Minsky'nin bulunduğu dönem içerisindeki donanım yetersizlikleri olduğunu belirtmiş ve sembolik yapay zekanın kullanılabilir bir yöntem olduğunu söylemiştir. Minsky (1991), donanım yetersizliğinin sebebini, ilişkilcelilerin yönteminin çalışma prensibinde yüklü bir veri kümesinin işlenmesi sürecinin bulunması ile açıklamıştır.

Donanım yetersizliği ile de başarıyı düşen ilişkilselci yapay zeka çalışmaları, 1986 yılında Rumelhart, Hinton ve Williams'ın algoritma hatalarından öğrenen sistem önerisi ile yeniden önemli hale gelmiştir. Rumelhart ve diğ. (1986) "Learning representations by back-propagating errors" isimli yayınlarında bugünün derin öğrenme sisteminin "öğrenme" kısmı için kullanılan "back-propagation" yöntemini önermişlerdir. Bu yöntemde yapay sinir ağındaki nöronların değerleri, algoritmanın yaptığı hatalara bakılarak eniylenir.

Hinton ve diğ. (2006) çok katmanlı sinir ağını önerdikleri "A fast learning algorithm for deep belief nets" adlı çalışmalarında derin sinir ağıları ile hiyerarşik olarak verinin özelliklerini öğrenebilen algoritmalarını sunmuşlardır. Bu çalışma ile birlikte, makine öğrenmesi içerisindeki çıkarım yapma ve tahmin etme problemlerinin çözümünde kullanılmak üzere, yapay zekanın yeni odağı, ilişkilcelilerin yeni yöntemi, derin öğrenme algoritmaları olmuştur.

Khrizhevsky ve diğ. (2012) "ImageNet Classification with Deep Convolutional Neural Networks" adlı çalışmasında derin evrimsel sinir ağıları ile birlikte 650000 nörondan oluşturulmuş sinir ağının 1.3 milyon yüksek çözünürlükteki görsel verinin 1000 farklı kategoriye sınıflandırılması problemini yüksek verimlilikle çözebilmişlerdir.

Goodfellow ve diğ. (2014), "Generative Adversarial Nets" (Çekişmeli Üretken Ağlar) adlı yayınlarında derin öğrenme algoritmalarını kullanarak üretken bir sistem önerisinde bulunmuşlardır. Derin öğrenme algoritmalarının görsel tanımadaki gücü aynı şekilde görsel üretme problemlerinde de verimli sonuçlar vermiştir. Çekişmeli

retken aęlar eęitimde kullanılan verisetini taklit ederek yeni grsel veri retebilmektedir.

Warren McCulloch & Walter Pitts ile 1943'te bařlayan ve gnmzde gl iřlem aralarına dnřmř yapay sinir aęları modeli, bugn A ile birlikte gereęinden ayırt edilemeyecek taklit grseller retim gcne eriřmiřtir.

Yapay zeka algoritmalarının geliřim hızı, grsel iřlemedeki bařarımları ve A'nın grsel retim gc, yakın gelecekte grsel verinin ierisinde bulunduęu tm disiplinleri etkileyebilme ihtimalini gstermektedir ve mimarlık da grsel verinin iinde bulunduęu bir disiplindir. Bu nedenle derin ęrenme algoritmalarında A'nın mimarlık disiplini ierisindeki yeri arařtırılması gereken bir soru olarak karřımıza çıkmaktadır. Bir sonraki blmde mimarlık pratięi iinde olan mimari plan řeması retimi ile ilgili literatr alıřması aktarılacaktır. Bu blmde A'nın da dahil olduęu mimari plan řeması retim alıřmaları da dahil edilmiřtir.

2.2 Mimari Plan řeması retim Problemi ile İlgili Literatr

Bu blmde mimari plan řeması retim problemine ynelik geliřtirilen uygulamalara yer verilecek ve uygulamalar iin kullanılan yntemler aıklanacaktır. "*Makineler mimari plan řeması tasarlayabilir mi?*" sorusunun yanıtı literatrde aranacaktır. Eastmen (1973) mimarideki plan řeması dzeni problemlerine, mekan planlama (space planning) problemi adını vermiřtir ve bu problemin otomatik mekan dzeni retimi (automated generation) iin zmler bulma hedefinde olduęunu belirtmiřtir. Literatr arařtırması sırasında 1955-2020 yılları arasında mekan planlama problemi ierisindeki otomatik plan dzeni retimi yapan uygulamalarla ilgili 50 alıřma zerinden bir arařtırma yapılmıřtır. Bylece literatrdeki otomatik plan řeması retim yntemlerinin eřitlenme řekli ve gnmzdeki yntemsel ynelimi aıklanabilir. řekil 2.5'te literatrde taranan otomatik plan řeması retimi ile ilgili kaynaklar gsterilmiřtir.

tekniki tasarımıda kullanılacak, geometrilerin ve bu geometrilerin ilişkilerinin belirlenmesi ile gerçekleştirilmektedir. 1970'lerde bir diğery yöntem olarak ortaya çıkan biçim grameri teorisi kullanılarak plan şemalarının türetilmesi sağlanabilmiştir. Bunun yanında plan şemasının üretilme sisteminde genetik algoritmalar da denenmiştir. Günümüzde plan şemasının üretimine dair çalışmalarda bugünün yapay zeka tekniğı derin öğrenme yönteminin kullanılmaya başlandığı gözlemlenmektedir.

Krejcirik (1969) bilgisayar destekli tasarım ile plan düzleminin optimum yerleşimi ile ilgili çalışmasını geliştirmiştir. Weinzapfel ve diğ. (1971) 3 boyutta mekansal düzenlemeleri bilgisayar destekli tasarım sürecinde üretmek amacıyla, mekanları geometrik olarak tanımlayarak, bu geometriler arasındaki ilişkileri tasarım kısıtları olarak kullanmışlardır. Levin (1964) ve Grason (1971) plan şemasında mekansal dizilim için çizge teorisini kullanmışlardır. Eastmen (1973) araştırmasında iki boyutta mekansal planlamanın otomatikleşmesine dair bir sistem önerisinde bulunmuştur. Otomatik plan şeması tasarlama sistemi için Eastmen (1973), üç değişken belirlemiştir. Bu değişkenler mekan, tasarım birimleri, tasarım birimleri arasındaki ilişkiler ve bu ilişkileri manipüle edecek operatörlerdir. Bu değişkenlerden tasarımı manipüle eden operatörler *iyi tanımlanmış (well-defined)* olmaması tasarımıdaki yaratıcı nokta için potansiyel sağladığı söylenmiştir (Eastmen, 1973). Bunun yanında tasarım birimleri arasındaki ilişkiler de *boolean* fonksiyonları olarak tanımlanmıştır. Eastmen mekansal planlamadaki zorluğu, tasarım birimlerinin sonsuz sayıdaki oryantasyon ve konumlanma olasılıklarının üretilmesi olduğunu söylemiştir (Eastmen, 1973).

Stiny & Mitchell (1978) Andrea Palladio'nun plan gramerlerini yeniden modern ve üretken bir sistem olarak düzenlemeyi hedeflemişlerdir. Palladio'nun plan sistemindeki gramer kullanılarak *Villa Malcontenta* plan şeması üretilmiştir (Stiny & Mitchell, 1978). Bu çalışmada Stiny ve Mitchell Palladio'nun plan şemasını otomatik şekilde üretmemişlerdir. Ancak Palladio Plan şemalarının otomatik üretilmesi için gerekli algoritmayı oluşturabilmişlerdir. Bu çalışma, biçim gramerleri ile yapı plan şemasının tasarım kurallarının oluşturulduğu bir çalışma olmuştur. Koning & Eizenberg (1981) Frank Lloyd Wright'ın kır evleri gramer kurallarını tanımlamış, Çolakoglu (2005), Hayat evleri gramerleri ile tasarımıda üretken bir model olarak biçim gramerlerini kullanmıştır. Duarte (2005) yaptığı çalışmada mimari tasarım için *üretim sistemi (production system)* kullanmayı hedeflemiştir. Çalışmasında

programlama grameri ile tasarım kurallarının oluşturulmasını sağlamış, tasarım grameri ile tasarım olasılıklarının üretken şekilde türetilmesini hedeflemiştir (Duarte, 2005). Easment'ın (1973) *S-İlişkileri (S-Relation)* olarak, tasarım birimleri arasında tanımladığı değişkenler Duarte'nin çalışmasında biçim grameri kuralları olarak karşımıza çıkmaktadır. Biçimlerin dizilimleri *S-ilişkilerine* göre değil biçim grameri kurallarına göre dizilmektedir. Duarte (2005) biçim gramerlerini kullanarak Alvora Siza'nın *Malagueira konutlarını* yeniden türetmiştir.

Ahmad ve diğ. (2004) plan şemasında mekanın verimli olarak kullanılabilmesine yönelik çalışmalarında genetik algoritmaları kullanmışlardır. Ahmad ve diğ. (2004), tasarımda kullanılacak biçimleri plan şemasında optimum düzenleyerek toplamda oluşan planın sınır çizgisini minimize etmişlerdir. Rojas ve Torres (2006) genetik algoritmaları kullanarak, banka ofisinin plan şemasının tasarlanması problemine çözüm aramışlardır. Dalgıç ve Diğ (2017) market raflarının plan düzleminde yerleşme probleminin, ürün satışı için önemli bir noktada olduğunu söylemişlerdir. Bu nedenle marketteki rafların görünübilirlik problemini de göz önünde bulundurarak genetik algoritmalar ile çalışan plan şeması üretici sistemi tasarlamışlardır. Çalışmaların sonucunda, Dalgıç ve diğ. (2017), genetik algoritmaların hem birçok olası tasarım önerisini hızlı bir şekilde üretebilmesi hem de tasarım problemine optimum çözümü bulabilmesi bakımından plan şeması oluşturmada yararlı bir teknik olduğunu söylemişlerdir.

Nagy ve diğ., (2017), 15 komşuluk ilişkisi ve 3 parametreden oluşan 45 boyutlu bir uzaydaki tasarım problemine üretken sistemler ile tasarım çözümü bulabilecek bir sistem geliştirmişlerdir. Bu çalışma Eastmen'in S-ilişkileri ve *tasarım birimlerine (design unit)* benzer bir mantık üzerinden geliştiği gözlemlenmektedir. Nagy ve diğ., (2017), tasarım problemi çözümü için optimum problem uzayı boyutu ölçüsünün belirlenmesinin mümkün olmadığını söylemişlerdir. Ürettikleri sistem yarı otonom olarak çalışmakta, tasarımcının algoritmayı yönlendirmesi ve seçimleri ile tasarım geliştirilmektedir. Yarı-otonom tasarım için geliştirilen sistemde genetik algoritma kullanılmıştır. Nagy ve diğ ., (2017), MOGA sayesinde tasarımcının tasarım problem uzayındaki tek bir problemle uğraşmasının yerine, büyük bir problem uzayında çok boyutlu problemlerle uğraşabileceği ve optimum sonuçları alabileceğini söylemiştir. Bu durum tasarlama eylemini radikal biçimde değiştirebilecek bir güce sahiptir. Tasarımda göz ardı edilen problemler çözülebilir hale gelebilir.

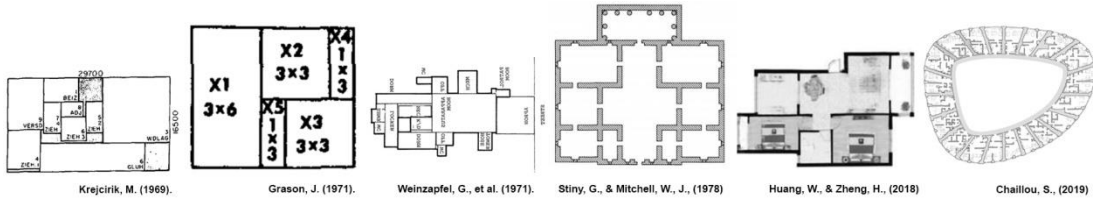
Plan şeması üretim problemi içerisinde günümüzde parametrik tasarım aracı olarak kullanılan grasshopper ile geliştirilen Finch yazılımı, plan şemasındaki mekansal dizilim, metrekaireler ve tefriş ile ilgili kararları otomatik olarak alabilmektedir. Mekansal büyüklükler, tefriş ve mekansal dizilim için bir parametre olarak ele alınmış olan bu yazılımda, plan şemasının toplam metrekaresinin artıp azalması, hem tefrişi hem de mekansal dizilimi yeniden üretmektedir (URL1).

Günümüzde en önemli teknolojik gelişme olarak karşılaştığımız derin öğrenme teknolojisi, otonom plan üretimi sistemi içerisinde de yerini almaya başlamıştır. Huang ve Zheng (2018), çalışmalarında derin öğrenme ağları içerisinde bulunan üretken sistemlerden ÇÜA ağlarını kullanarak plan tasarımı gerçekleştirebilmişlerdir. Kullandıkları ÇÜA algoritması Pix2pixHD adındaki özelleşmiş bir ÇÜA ağıdır. Bu ağ etiketlerle üretilen lekeler üzerinde etiketlere göre mekansal görseller türetebilmektedir. Etiket mutfak olduğunda, mutfak planına dair çizimler, oturma odasına ait olduğunda da plan şemasında oturma odasına dair çizimler yapmaktadır. Chaillou (2019)'nın çalışması ÇÜA ağları ile plan üretimi konusundaki bir diğer çalışmadır. Bu çalışmada ÇÜA ağları 3 basamakta plan üretimini gerçekleştirmektedir (Chaillou, 2019). Birinci basamakta planın toplamda kapladığı leke üretilmekte, daha sonra iç bölmelendirme ve mekansal dizilimi üretmekte, son olarak tefriş üretimi ile plan üretimini sonlandırmaktadır. Uzun, Çolakoğlu ve İnceoğlu (2020), DCGAN algoritması kullanarak Palladyan plan şemalarının üretimini gerçekleştirmişlerdir.

Literatürdeki çalışmalar incelendiğinde, çalışmaların çoğunun mimari plan şeması üretimi problemini çözebilmek amacıyla çizge teorisine başvurduğu görülmektedir (Ruizhen Hu ve diğ., 2020; Del-Rio, 2007; Medjoub and Yannou, 2001; Ruch, 1978; Fortin, 1978; Al Banna, 1972; Grason, 1971; Miller 1970). Mimari plan şeması üretiminde güncel bir üretim yöntemi olarak kullanılan yapay sinir ağı algoritmalarının görsel işleme gücü nedeniyle, bu alandaki araştırmalarda yapay sinir ağlarına odaklı çalışmaların (Uzun ve diğ., 2020; Chaillou, 2019; Eisenstadt ve diğ., 2019; Huang & Zheng, 2018; Bayer ve diğ., 2007). sayısının arttığı gözlemlenmektedir.

Şekil 2.6'te kronolojik olarak kullanılan yöntemlerin mimari plan şeması üretim çıktılarındaki çözünürlükleri karşılaştırılabilir. Bu karşılaştırmaya göre derin öğrenme yöntemlerinin kullanıldığı sistemlerin, plan şeması üretiminde yüksek

çözünürlükte (mimari plan şeması ifadesinin mimari bir dille oluşturulması) sonuçlar verdiği görülebilir. Bu durum derin öğrenme algoritmaları içerisindeki ÇÜA ağı ile mimari plan şeması üretiminin hem sürecinin hem de çıktılarının değerlendirilmesi için potansiyel bir çalışma alanı işaret eder. Bu doğrultuda bir sonraki bölümde ÇÜA mimarisi tanıtılacak ve ileri ki bölümlerde ÇÜA ile mimari plan üretim süreci deneylerle açıklanarak, çıktıları değerlendirilecektir.



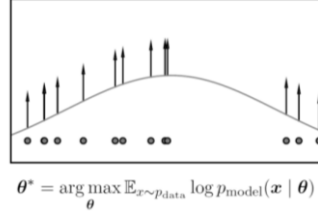
Şekil 2.6: Kronolojik olarak otonom mimari plan üretimi çalışmaları çıktıları.

3. ÇEKİŞMELİ ÜRETKEN AĞ VE MİMARİSİ

3.1 Çekişmeli Üretken Ağ Tanımı

Çekişmeli Üretken Ağlar derin öğrenme algoritmalarından üretken modeller içerisinde yer almaktadır. Ian Goodfellow (2016) çekişmeli üretken ağların (Generative Adversarial Networks / ÇÜA), üretken modeller içerisinde bir örnek olduğunu söylemektedir. ÇÜA modelleri herhangi bir eğitim veri seti ile eğitilerek, eğitim sonrasında öğrendiği veriye göre bir temsil üretmektedir. Sonuçta üretilen çıktı olasılıksal bir veri modelidir. ÇÜA ağlarının başlıca odağında model üretimi bulunmaktadır (Goodfellow, 2016).

ÇÜA ağları, maksimum benzerlik (maximum likelihood) yöntemini kullanarak modeller arasındaki benzerliği en iyilemeye çalışmaktadır.



Şekil 3.1: ÇÜA algoritmasında beklenen değerinin en iyilenme fonksiyonu (Goodfellow, 2016).

Ian Goodfellow (2016), Şekil 3.1 ile girdi verilerindeki her bir nokta için sonsuz benzerlik tanımlanamayacağını söylemiş ve bu noktalarındaki yoğunluk fonksiyonlarının değerlerinin toplamının 1'e eşit olması gerektiğini söylemiştir. Bunun nedeni bir noktanın benzerliğinin çok fazla arttırılması ile eğri noktaya yaklaşırken diğer noktalarda eğri uzaklaşmaktadır. Yani tüm noktalar kendi arasında dengede hareket etmektedir. Bu nedenle tüm noktaların aynı anda veri setine en çok benzeyebileceği şekilde yoğunluk fonksiyonu hesaplanmaktadır.

GAN 'ın en büyük avantajlarından birinin yüksek boyutlu uzaydaki olasılık dağılımını verimli ve hızlı olarak hesaplayabilmesidir (Goodfellow, 2016). Görsel verinin işlendiği sistemlerdeki piksel değerleri ile birlikte oldukça yüksek veri ile

karşılaşılmaktadır. Piksel değerlerinden gelen matris parametreleri arasındaki ilişkiyi ÇÜA ağlarının hesaplama süreci oldukça verimlidir.

Mimaride karşılaştığımız üretken sistemler; biçim gramerleri, L sistemler, fraktaller, hücresel özdevinim, sürü zekası ve genetik algoritmalar olarak sınıflandırılır (Dinçer ve diğ, 2014). Bu üretken sistemler içerisinde benzetim yaparak en uygun modele ulaşmayı hedefleme mantığıyla genetik algoritmalar ÇÜA ağlarına çok yakın bir sistemdir. Ancak genetik algoritmalar ile ÇÜA ağlarının çalışma mantığı oldukça farklıdır. Genetik algoritmalar evrimselci (evolutionarist) öğrenen algoritmalar içerisinde sınıflandırılır, çok katmanlı yapay sinir ağlarında geri yayılım (back propogation) ile çalışan öğrenme sistemleri ilişkiselselci (connectionist) öğrenen algoritmalar içinde sınıflandırılır (Domingos, 2015). ÇÜA ağları her ne kadar evrimselcilerin mantığındaki süreçler gibi ulaşmak istenen bilgiyle benzer bir model üretimi hedeflese de çalışma prensibi bakımından, geri yayılım kullanılması nedeniyle farklıdır.

GAN ağları genetik algoritmalarından çok daha verimli şekilde çalışmaktadır. Genetik algoritmalarından uygunluk (fitness) değeri sabit olarak tutulup birçok örneklem (sample) üretilmesi gerekmektedir. Genetik algoritmalar bu üretilen örneklem arasında uygunluk değerine en yakın değerde olanı seçerek sonuçlar üzerinden yeni bir örneklem üretir. Bu süreç uygunluk fonksiyonundaki değere yaklaşıncaya kadar devam eder. Ancak ÇÜA ağlarında sistemde örneklem üretim süreci yoktur, bu sistem süreci oldukça hızlandıracaktır. Bunun yanında sonuçta üretilen çıktının en verimli üretim olması için ÇÜA ağlarında uygunluk değerleri (loss value) dereceli azalma (gradient descent) yönteminde kısmi türevlerle hesaplanarak en verimli uygunluk değeri hesaplanır ve bu uygunluk değeri ile yeni model üretilir.

3.2 ÇÜA Algoritmasında Kullanılan Temel Kavramalar

Veri Seti: Makine öğrenmesi tekniklerinden derin öğrenme sistemi genellemeler yapabilmesi amacıyla veriye başvurmaktadır. Eğitimi yapılacak kapsam ile ilgili verilerden oluşan küme verisetini oluşturmaktadır. Veri seti gözetimli öğrenmede yapılan gözlemlere (observation) etiketler (label) atamaktadır. Gözetimsiz öğrenmede ise gözlemlere atanan etiketler yoktur. Burada gözlem sayısı veri setinin içindeki veri miktarını (n) gösterir. Etiketler gözlemlerdeki bağımsız değişkenlerden (predictor / p) etkilendiği için bağımlı değişken olarak adlandırılır. Gözetimli

öğrenmede problem etiket tahmini olabilir ve sınıflandırma problemi içerisinde yer alır. Gözetimsiz öğrenmede ise bağlantım (regression) ve kümeleme (clustering) problemleri çözümlenebilir. ÇÜA algoritmaları gözetimsiz öğrenme içerisinde yer alması nedeniyle veri seti ÇÜA ağlarında çoğunlukla etiketsiz olarak verilmektedir. Veri setinin eğitim için yetersiz olması durumunda n 'in küçük bir değer alması nedeniyle esnek bir öğrenen algoritma modeli oluşturmak zor olacaktır bu nedenle algoritmanın eğitimi de zorlaşacaktır. Bu nedenle sentetik olarak veri üretilerek veri setine eklenir. Bu işlem veri arttırımı olarak isimlendirilir. Birçok öklidyen dönüşüm veri seti arttırımı için kullanılacağı gibi ÇÜA ağları veri arttırımı için kullanılan yöntemler arasındadır. ÇÜA eğitiminde kullanılan verinin yetersiz olması durumunda öklidyen veri arttırımı yöntemlerinin etkisini tez içerisinde ÇÜA deneyleri bölümünde değerlendirilmiştir.

Ayrıştırıcı (Discriminator) : ÇÜA modelinde ayrıştırıcı (**D**) olarak tanımlanan fonksiyondur. Ayrıştırıcı fonksiyon hem gerçek veri seti ile eğitilir hem de üretici fonksiyonunun üretimlerini kontrol eder. İki farklı *maliyet fonksiyon (cost funtion)* değeri çıktısı vardır. Gerçek veri seti kullanılan eğitim sürecindeki maliyet değerini kendi eğitimini *en iyilemek (optimize)* için kullanmaktadır. Üretici fonksiyondan gelen verinin kontrolü sonucunda oluşan kayıp fonksiyon çıktısı ise üretici fonksiyonun girdisini oluşturur. ÇÜA modelinin ayrıştırıcı fonksiyonunda *evrişimsel katman (convolutional layer)* kullanılmaktadır. Girdi verisi evrişimsel katmandan geçerken bir *filtre (weight)* ile veri okunmaktadır ve sonucunda bir çıktı verisi üretilir. Bu çıktı verisinin büyüklüğü *filtre, stride* ve *padding* değerleri ile belirlenebilir. Ayrıştırıcı fonksiyon evrişimsel katmanlarla birlikte gerçek veriyi öğrenirken üretici fonksiyondan gelen sahte verileri yanlış olarak ayıklayabilmektedir. Bu eğitim sürecinde fonksiyon tahminlerindeki hatalarının büyüklüğünü anlayabilmesi amacıyla maliyet fonksiyonlarına başvurur. Ve maliyet fonksiyonu değerini azaltabilmek için en iyileyiciler kullanılır. Bu açıklama içerisinde bulunan evrişimsel katman, filtre, stride, padding, maliyet fonksiyonu ve eniyileyici kavramları aşağıda açıklanmaktadır.

Üretici (Generator) : Üretici fonksiyon (**G**) belirsiz, gürültülü(noisy) bir veriyi (z / latent space) daha temiz ve özellikleri anlaşılabilir bir veriye ($G(z)$) dönüştürmektedir. Üretici fonksiyon ya upsampling ve evrişimsel katmanı birlikte kullanmaktadır ya da sadece dekonvolüsyonel katmanı kullanmaktadır. Upsampling

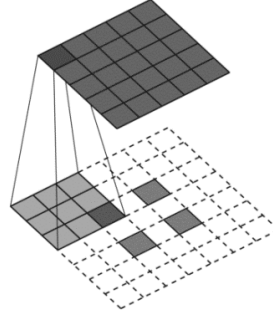
ve evrişimsel katman kullanımında veya sadece dekonvolüsyonel katman kullanımında imaj verisi üzerinde gezinecek olan *algılayıcı alan (receptive field, filter, kernel)* ile elde edilecek çıktı imajın boyutunu *strides* ve *padding* değerleri ile belirlemektedir. Evrişim işlemi sonrasında tahmin edilen değer ile birlikte ayrıştırıcı fonksiyon maliyet değerini hesaplar. Ayrıştırıcıdan gelen maliyet fonksiyonuna göre üretimlerini güncellemektedir (Brownlee, 2019).

Evrişimsel Katman (Convolutional Layer) : Bir imaj üzerinde filtrenin gezmesi ile birlikte, filtre ve girdi imajındaki piksel değerleri ile oluşan matris çarpımı işlemi evrişim / konvolüsyon (convolution) olarak adlandırılır (Goodfellow, I., Deep Learning). Girdi verisi x , piksel değerlerinden oluşan $m \times n$ boyutlu bir matris, ağırlık w ise yine piksel değerlerinden oluşan $t \times s$ boyutlu bir matrisi tanımlar. $y = x \times w$ konvolüsyonel işlem olan matrisler çarpımını tanımlamaktadır. Eğitim sürecinde hedef y değerini doğru üreten w değerini yani filtreyi bulabilmektir. Konvolüsyonel katmanlarda yapılan işlemlerde padding değeri 0'dan farklı değilse girdi imaj boyutu küçülecektir. Doğru bir *padding* değeri ile birlikte konvolüsyonel katmandan geçen imajın boyutunda değişim olmayacaktır. Konvolüsyonel katmandaki çıktı boyutunu etkileyen bir diğer kavram da *stride*'dir. Filtre, stride ve padding kavramları aşağıda detaylı olarak açıklanmıştır. Bunun yanında tezde eğitimi gerçekleştirilen ÇÜA algoritmasının ayrıştırıcı (discriminator) fonksiyonunda konvolüsyonel katmanlar kullanılmaktadır (Sharma, 2018).

Upsampling : Upsampling üretken sinir ağlarında üretici fonksiyon içinde kullanılmaktadır. Upsampling eğitimdeki girdi boyutunu iki katına çıkarmaktadır. Upsampling filter kullanmamaktadır, upsampling katmanı sonrasında konvolüsyonel katman kullanılmaktadır. Konvolüsyonel katmanda padding değeri girdinin şeklini değiştirmeyecek şekilde ayarlandığında upsampling nedeniyle girdi boyutu çıktıda iki katına çıkmış olur (URL2).

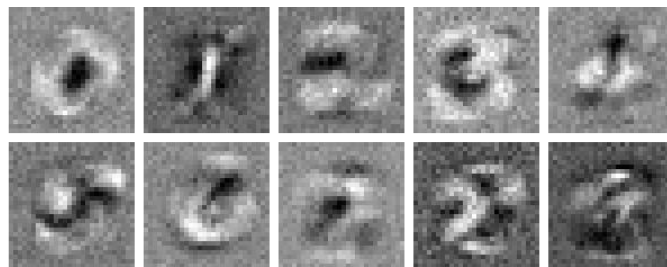
Dekonvolüsyonel Katman (Deconvolutional Layer) : Dekonvolüsyonel katmanlar semantik segmentasyon işlemleri ve üretken modellerde kullanılan bir konvolüsyon yöntemidir (Shi ve diğ., 2016). Dekonvolüsyonel katman, kademeli konvolüsyonel katman (fractional convolutional layer), tranpoz konvolüsyonel katman (transposed convolutional layer), ters, yukarı veya geriye konvolüsyonel katman (inverse, up, backward convolutional layer) olarak da adlandırılmaktadır (Shi ve diğ., 2016). Dekonvolüsyonel katman upsampling ve konvolüsyonel katmanın bir arada yaptığı

yapmaktadır. Upsampling işlemine göre daha kompleks bir yöntemdir. Dekonvolüsyonel katman upsampling yaparken aynı zamanda konvolüsyon işlemi yapmaktadır. Konvolüsyonel katmanda olduğu gibi bir filtre imaj üzerinde hareket etmekte ve çıktı verinin boyutu artmaktadır. Dekonvolüsyonel katmanda da padding ve stride değerleri ile çıktı boyutuna karar verilebilmektedir. (URL 4)



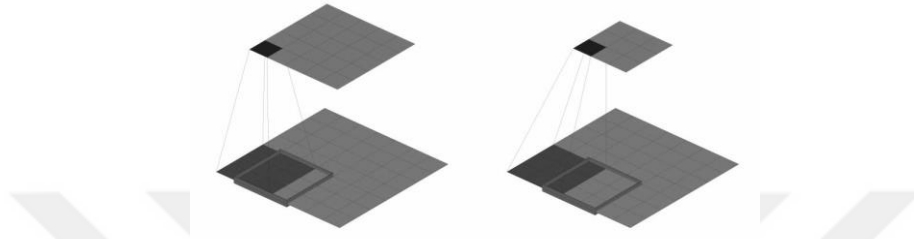
Şekil 3.2: Dekonvolüsyonel katmandaki filtrenin ilerleyişi (URL6).

Filtre (Filter) : Makine öğrenmesi erken çalışmalarından itibaren filtre, imaj üzerinden özellik çıkarımı (feature extraction) için kullanılan bir yöntem olmuştur. Filtre sayesinde veri üzerindeki gürültü (noise) ortadan kaldırılabilir. ÇÜA modelinde veri imaj olarak tanımlanabilir, gürültü ise imaj üzerindeki piksel değerlerinden kaynaklanmaktadır. Erken makine öğrenimi çalışmalarında filtre değeri imaj boyutu ile aynı olmakta ve her katmanda yeni bir filtre kullanılmaktadır. Bu durum oldukça fazla işlem gücü gerektirmesi nedeniyle işlem yapan donanım için verimsiz bir süreç oluşturmaktadır. Günümüzde sinir ağlarında filtre boyutları, yapay sinir ağının girdi verisi olan imajın boyutundan küçük olarak kullanılmakta ve bir filtre imajın her yerinde adım adım (stride) kullanılabilir. Bu durum işlemsel süreci eski filtre yöntemleri ile karşılaştırıldığında oldukça verimli bir noktaya taşıyabilmiştir. Filtreler eğitim sürecinde farklı özelliklere (feature) göre güncellenmektedir. Eğitim sonucunda en verimli sonucu veren filtre bulunur. Filtreler fonksiyonlarda w / ağırlık(weight) olarak da adlandırılır (Deshpande, t.y.).



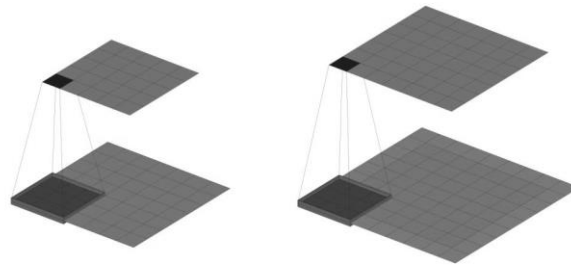
Şekil 3.3: MNIST veriseti sınıflandırısındaki filtrelerin görselleştirilmiş hali (URL3).

Stride : Stride filtrenin imajda bulunan pikseller üzerinden hareket ettirilme büyüklüğünün belirlendiği parametredir. Bir başka deyişle algısal alanın (receptive field) kaydırılma büyüklüğü ile ilgilidir. Strides değeri çıktıdaki verinin boyutunu etkilemektedir. 7×7 px büyüklüğündeki bir imaj üzerinde 3×3 px boyutlu bir filtre strides 1 olarak dolaştığında, çıktıdaki imajın büyüklüğü 5×5 px olmaktadır. Strides boyutu 2 olduğunda 7×7 px boyutundaki bir imajın çıktı büyüklüğü 3×3 px boyutuna düşmektedir (Deshpande, t.y.).



Şekil 3.4: Konvolüsyonel katmandaki stride hareketi; solda stride=1, sağda stride=2.

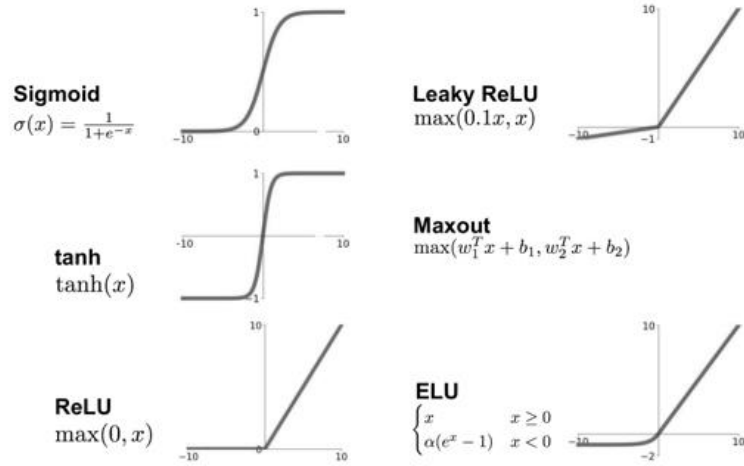
Padding : Padding konvolüsyonel veya dekonvolüsyonel katmandaki filtrenin hareketi sonrasındaki girdinin imaj biçiminin çıktıdaki boyutunun belirlenmesini sağlamaktadır. Verilen padding değeri ile imajın çevresine padding değerine göre pikseller eklenmektedir. 7×7 px boyutundaki bir imajın üzerinde konvolüsyonel katmanda 3×3 boyutundaki bir filtre gezindiğinde padding değeri 0 olarak belirlenirse çıktıda imajın boyutu 5×5 olmaktadır. 7×7 px boyutundaki bir imajın üzerinde konvolüsyonel katmanda 3×3 boyutundaki bir filtre padding değeri 1 olarak gezindiğinde çıktı imajın değeri 7×7 olarak aynı kalmaktadır. Bu nedenle konvolüsyonel veya dekonvolüsyonel katmanlarda işlenen imajın boyutunun değişmemesi isteniyorsa uygun padding değerleri tanımlanması gerekmektedir.



Şekil 3.5: Konvolüsyonel katmandaki padding oluşumu, solda padding=0, sağda padding =1.

Aktivasyon Fonksiyonu (Activation Function): Bir yapay sinir ağında aktivasyon fonksiyonları nöronun çıktısını belirler. Bunun yanında sinir ağının yakınsama

(yapay sinir ağıının doğru tahminleri için doğru kernel değerlerine karar verilmesi) hızını da belirlemektedir. Yapay sinir ağıındaki nörona bağlı olan aktivasyon fonksiyonu çıktı değerini belirleyerek o nöronun aktifleşip aktifleşmeyeceğini belirlemektedir. Sigmoid, LeakyReLU, tanh, Maxout, ReLU, ELU, Softmax çeşitli aktivasyon fonksiyonlarıdır. Softmax ikili (binary) sınıflandırma yapabilmekteyken, sigmoid çoklu sınıflandırma yapabilmektedir. ReLU'nun kullanımı işlemsel gücü bakımından hızlı olmasına rağmen gradient vanishing (bazı nöronların ölmesi) probleminde neden olabilir. Bundan kurtulabilmek için LeakyReLU fonksiyonu kullanılabilir. LeakyReLU fonksiyonu ise işlem gücü daha yoğun olması nedeniyle eğitim süresini uzatmaktadır. ÇÜA algoritmalarından DCGAN modelinde çoğunlukla üreticinin evrişimsel katmanlarındaki aktivasyon fonksiyonu ReLU olarak tercih edilmekte en son katmanında ise tanh aktivasyon fonksiyonu tercih edilmektedir. DCGAN algoritmasının ayrıştırıcı fonksiyonunun evrişimsel katmanlarında yine ReLU ve en son katmanında da sigmoid fonksiyonu kullanılmaktadır (URL4).



Şekil 3.6: Sinir ağlarında kullanılan aktivasyon fonksiyonları (URL5).

Maliyet Fonksiyonu (Cost Function): Maliyet fonksiyonunun çıktısı sinir ağıının tahminindeki hata büyüklüğünü göstermektedir. Hata büyüklüğü weight (W), bias(B), girdi (Sr) ve beklenen değer (Er) değişkenlerine bağlıdır. Maliyet fonksiyonu C(W, B, Sr, Er) şeklinde gösterilmektedir. Sinir ağı içerisindeki en iyileyici ile birlikte maliyet fonksiyonu düşürülmeye çalışılır bu süreçte kerneller (weight) sürekli güncellenir. Binary cross entropy, Exponential, Hellinger Distance, Kullback–Leibler divergence, Itakura–Saito distance farklı maliyet fonksiyonlarıdır.

Tez çalışmasında kullanılacak maliyet fonksiyonu Binary Cross-Entropy olarak seçilmiştir (URL6).

Eniyileyici (Optimizer): Eniyileyici fonksiyon maliyet fonksiyonunu, maliyet fonksiyonuna bağlı parametreyi optimize ederek minimize etmektedir. $J(\theta)$ fonksiyonunda θ parametresinin en iyilenmesi ile birlikte $J(\theta)$ fonksiyonu minimize edilebilir. 3.1’de görülen fonksiyonda maliyet fonksiyonuna karşılık olarak p^{data} deneysel veri üretim dağılımıyla oluşturulan öngörü fonksiyonu (expectation function ‘E’) görülmektedir (Goodfellow, 2016).

$$J(\theta) = \mathbb{E}_{(x,y) \sim p^{data}} L(f(x; \theta), y) \quad (3.1)$$

Optimize etme sürecinde çoğunlukla kullanılan yöntem gradyan yöntemidir. Gradyan ile birlikte maliyet fonksiyonu kısmi türevlerle birlikte en iyilenmektedir. Aşağıdaki fonksiyonda Goodfellow, I., (2016) tanımladığı maliyet fonksiyonunun gradyan fonksiyonu 3.2’da gösterilmiştir.

$$\nabla_{\theta} J(\theta) = \mathbb{E}_{(x,y) \sim p^{data}} \nabla_{\theta} L(f(x; \theta), y) \quad (3.2)$$

En iyileyici fonksiyonda veri yığınının (batch) boyutunun doğru belirlenmesi fonksiyonun verimli çalışabilmesi bakımından önemlidir. Veri yığının büyük olması gradyan hesabının doğruluğunu arttırabilmektedir ancak doğru orantılı olarak arttıramamaktadır. Bunun yanında küçük veri yığınları ile eğitim gerçekleştirildiğinde en iyileyici fonksiyondaki öğrenme oranı da küçültülmelidir. Yığın büyüklükleri donanım hafızasını da en verimli kullanılabilecek şekilde seçilmesi gerekmektedir.

En iyileyici fonksiyonlara örnek olarak AdaGrad, RMSProp, Adadelta, Adam en iyileyicileri verilebilir. Tez içerisinde kullanılan en iyileyici fonksiyon Adam (Adaptive Moments) optimizer’dir.

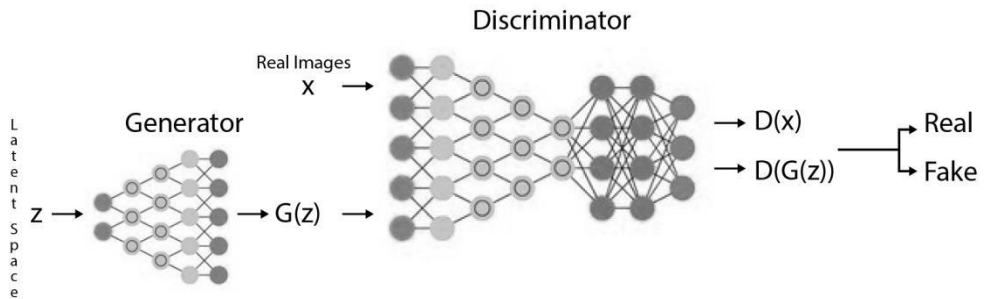
Dropout: Bir yapay sinir ağındaki nöronlar eğitim sırasında bağımlı olma eğilimindedir. Eğitim sırasında sürekli aynı nöronların güncellenerek eğitilmesi aşırı uyum (overfitting) problemine sebep olabilmektedir. Aşırı uyumun önüne geçebilmek için her iterasyonda rastgele olarak bazı nöronlar devre dışı bırakılmaktadır. Bu durum daha verimli bir eğitimin gerçekleşmesini sağlamaktadır.

Eğitimdeki iterasyon sayısı dropout ile birlikte iki katına kadar çıkabilmekte ancak her iterasyondaki eğitim süresi de düşmektedir (Budhiraja, 2018).

Yığın Normalleştirme (Batch normalization): Yığın normalleştirme eğitimin verimini ve hızını artırmak için veri içindeki özellikleri aynı aralığa sıkıştırılmaktadır. Bu işlem normalleştirme olarak adlandırılmıştır. Verisetindeki bir özellik 0-1 arasında diğer özellik de 0-1000 arasında olduğu düşünüldüğünde bu iki farklı özelliğin de 0-1 aralığına düşecek şekilde yeniden tanımlanması işlemi yığın normalleştirmedir (URL 7).

3.3 Çekişmeli Üretken Ağ Mimarisi

Goodfellow (2016), ÇÜA ağlarını en basit şekilde, iki oyuncu arasındaki çekişme olarak tanımlamaktadır. Oyunculardan biri *üretici (generator)*, diğeri *ayrıştırıcı (discriminator)* olarak adlandırılmıştır. Üretici yeni görseller üretirken, ayrıştırıcı hem gerçek veriyi öğrenmekte hem de üretici tarafından üretilen görsellerin gerçek olup olmadığını tespit etmektedir. ÇÜA ağlarında üreticinin hedefi ürettiği görsellerle ayrıştırıcıyı kandırabilmektir. Şekil 3.7’de ÇÜA mimarisi gösterilmektedir.



Şekil 3.7: ÇÜA mimarisi şeması (URL8).

Sistemde iki çeşit girdi verisi bulunmaktadır bunlardan birincisi üreticinin ürettiği veri, ikincisi gerçek veridir. Üreticinin ilk üretimi gerçekleştirebilmesi amacıyla rastgele sayılardan oluşan bir vektör ile beslenmektedir. Sistemin başlatılabilmesi için rastgele bir vektör verisi ilk olarak üreticiye gönderilmektedir. Bu ilk vektör z ile gösterilmektedir. Üreticinin bu z vektörü ile ürettiği çıktı $G(z)$ olarak adlandırılmaktadır. Ayrıştırıcı hem $G(z)$ verisini hem de gerçek verilerden oluşan x verisini girdi olarak kullanmaktadır. Üretici ve ayrıştırıcı ağların ikisi de kendi parametrelerine bağlı birer kayıp fonksiyonuna (cost function) sahiplerdir. Üretici

için kayıp fonksiyonu $J^{(G)}$ ile gösterilirken ayrıştırıcı $J^{(D)}$ ile gösterilmektedir. Ayrıştırıcı fonksiyonun (D) girdi değeri x iken kayıp fonksiyonun parametresi $\theta^{(D)}$ ile ifade edilmektedir. Üretici fonksiyonun (G) girdisi z , parametresi de $\theta^{(G)}$ olarak ifade edilmektedir. Ayrıştırıcı fonksiyon sadece ayrıştırıcı parametresini $(\theta^{(D)})$ kontrol ederek $J^{(D)}(\theta^{(D)}, \theta^{(G)})$ değerini minimize etmeye çalışmaktadır. Üretici fonksiyon sadece üretici parametresini $(\theta^{(G)})$ kontrol ederek $J^{(G)}(\theta^{(G)}, \theta^{(D)})$ değerini maksimize etmeye çalışmaktadır (Goodfellow, 2016) Bir başka deyişle üretici ve ayrıştırıcı ağ arasında bir minimum maksimum oyunu oynanmaktadır. Ayrıştırıcı sürekli optimize edilmektedir. Bunu gerçek verilere yüksek olasılık, üreticinin oluşturduğu verilere düşük olasılık vererek sağlamaktadır.

Ayrıştırıcı fonksiyonun kayıp değerini hesaplayan fonksiyon 3.3'te gösterilmiştir (Goodfellow, 2016).

$$J^{(D)}(\theta^{(D)}, \theta^{(G)}) = -\frac{1}{2} \mathbb{E}_x \sim p_{data} \log D(x) - \frac{1}{2} \mathbb{E}_z \log(1 - D(G(z))) \quad (3.3)$$

Üretici fonksiyonun kayıp değerini hesaplayan fonksiyon 3.4'te $J^{(G)}$ ile gösterilmiştir (Goodfellow, 2016).

$$J^{(G)} = -\frac{1}{2} \mathbb{E}_z \log(1 - D(G(z))) \quad (3.4)$$

3.5'te üretici fonksiyonu gradient ascent ile birlikte kayıp fonksiyonu değerini maksimize etmeye çalışırken, 3.6'da ayrıştırıcı da kayıp fonksiyon değerini gradient descent ile minimize etmeye çalışmaktadır (Goodfellow, 2016).

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m [\log D(x^{(i)}) + \log(1 - D(G(z^{(i)})))] \quad (3.5)$$

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^m \log(1 - D(G(z^{(i)}))) \quad (3.6)$$

Üretici ve ayrıştırıcı paralel olarak eğitilmeleri gerekmektedir. Eğer ayrıştırıcı, veri seti ile önceden eğitilmiş olursa, üreticinin her ürettiğini yanlışlayarak, üreticinin üretim yapamamasına neden olacaktır. Başka bir durum ise ayrıştırıcının çok kararlı olması durumudur. Bu durumda üreticinin gradyan (gradient) alanı oldukça geniş

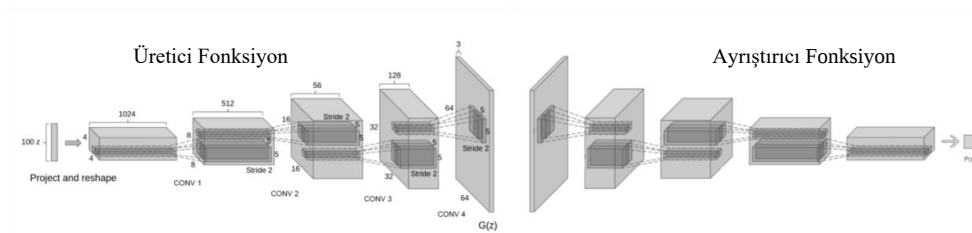
olacaktır ve bu nedenle üreticinin eğitimi oldukça zor hale gelecektir (Goodfellow, 2016).

ÇÜA algoritmasının eğitimi oldukça yoğun işlem gücü gerektirmektedir. Hem ayrıştırıcı hem de üreticinin eğitimi binlerce iterasyonlar mümkün olacaktır. Her bir iterasyonda *geri yayılım (back propogation)* ile her bir nöronun *ağırlık (weight)* değerleri *kısmi türevler (partial derivation)* ile yeniden hesaplanarak, hem üreticinin hem de ayrıştırıcının kayıp fonksiyonu değerleri en iyilenecektir. Eğitim sonunda hem ayrıştırıcı gerçek bilgiyi tanımada başarılı olacaktır hem de üretici gerçeğe yakın veri üretiminde usta olabilecektir. Böylece üretici ayrıştırıcıyı kandırmayı başarabilecek ve ÇÜA ağı veri üretimini gerçekleştirmiş olacaktır.

3.4 DCGAN Mimarisi

Tezin amacında bulunan ÇÜA ile mimari plan şeması üretim probleminde ÇÜA içinde yer alan DCGAN (Deep Convolutional Generative Adversarial Network /Derin Evrişimsel Çekişmeli Üretken Ağlar) algoritması kullanılmıştır. Bu nedenle bu bölümde DCGAN mimarisi açıklanmıştır.

Radford, Metz ve Chintala (2015), CNN algoritmalarının güdümlü öğrenme içerisinde sıklıkla kullanıldığını ancak güdümsüz öğrenmedeki çalışmaların sayısının az olduğunu söylemişlerdir. CNN algoritmalarının bir altı sınıfı olarak, güdümlü ve güdümsüz öğrenme arasındaki köprüyü kurabilen derin evrişimsel üretken çekişmeli ağları (deep convolutional generative adversarial network) Şekil 3.8’de 2015 yılında geliştirilen DCGAN mimarisi görünmektedir. Sol tarafta rastgele bir vektör ile başlatılan üretici ağ öncelikle 4 farklı deconvolutional ağından geçirilmektedir en sonunda $4 \times 4 \times 1024$ boyutundaki bir görselden $64 \times 64 \times 3$ boyutunda bir görsel üretilmektedir. Ayrıştırıcı üreticinin ürettiği boyutta veri ile besletilerek 4 farklı convolutional ağıdan geçirilerek kayıp fonksiyonu değerini hesaplamaktadır.



Şekil 3.8:DCGAN mimarisi şeması (URL 9).

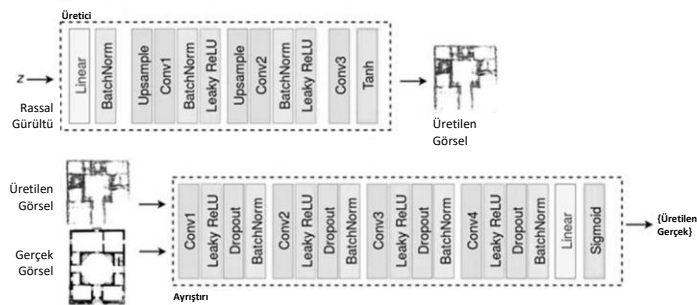
GAN ile ilgili deneyler bölümünde Radford ve diğ. (2015) önerdiği DCGAN mimarisine benzer bir mimari kullanılmıştır. Deneylerde kullanılan DCGAN algoritması da ayrıştırıcı ve üretici fonksiyonlar içindeki evrişimsel katmanlardan oluşmaktadır (Şekil 3.8).

Üretici fonksiyondaki evrişimsel katman sayısı deneylere göre farklılık göstermektedir. Üretici algoritma sırasıyla her biri 128,128 ve 64 adet filtreye sahip katmanlardan oluşmaktadır. Son evrişimsel katman haricinde batch normalizasyonu kullanılmıştır. Sonuncu evrişimsel katman dışında LeakyReLU ($\alpha=0.2$) aktivasyon fonksiyonu kullanılmıştır. Son evrişimsel katmanda ise tanh aktivasyon fonksiyonu kullanılmıştır.

Ayrıştırıcı algoritma da deneylerde kullanılan evrişimsel katman sayısı farklılık göstermektedir. Her evrişimsel katmanda aktivasyon fonksiyonu LeakyReLU ($\alpha=0.2$) kullanılmıştır. Her evrişimsel katmanda da aşırı uyma (*over-fitting*) probleminin önüne geçmek için eğitimin her iterasyonunda, dropout ($p=0.25$) yöntemi kullanılmıştır. Bunun yanında her bir evrişimsel katmanda yığın normalleştirilmesi de kullanılmıştır. Konvolüsyonel katmanlar sırasıyla 16, 32, 64 ve 128 filtreden oluşmaktadır.

Eğitim deneylerinde epok sayısı değişiklik göstermiştir ancak eğitimlerde kullanılan yığın (batch) büyüklüğü 32 ile sabit bırakılmıştır. Tüm deneylerde kullanılan en iyileyici Adam optimizer'dır. Kullanılan hiperparametreler sırasıyla; öğrenme oranı:0.0002, beta1:0,5 ve beta2:0.999'dır.

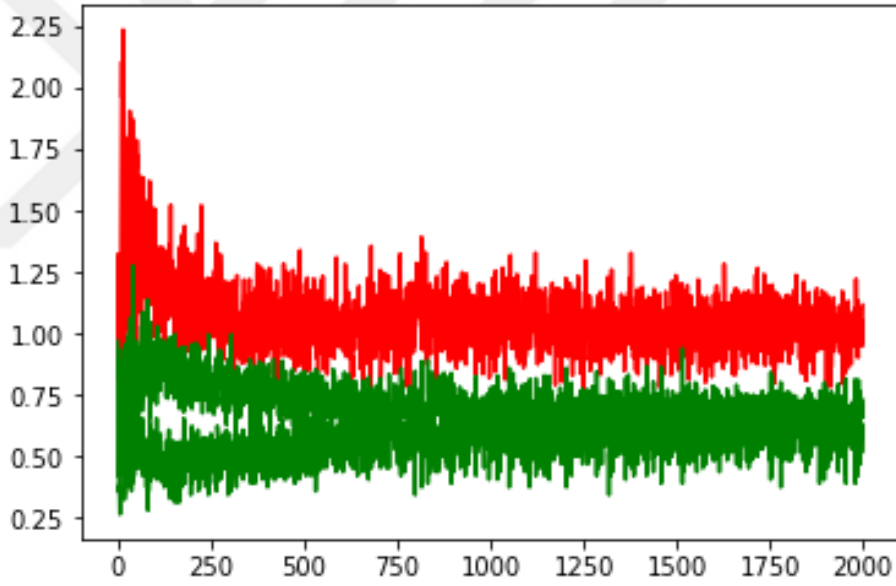
Tüm eğitimlerde imajlar tek renk kanallı (grayscale) imaj verisine dönüştürülmüştür. Son iki deneyde grayscale dışında verinin okunuşunu verimli hale getirmek amacıyla label smoothing (gerçek imaj etiketleri için 0.1 ve üreticinin ürettikleri için 0.9) kullanılmıştır (Uzun, Çolakoğlu ve İnceoğlu, 2020).



Şekil 3.9: Deneylerde kullanılan DCGAN mimarisi, üretici ve ayrıştırıcı katmanlarıyla birlikte.

4. ÇEKİŞMELİ ÜRETKEN AĞ EĞİTİM DENEYLERİ

Bu bölümde ÇÜA ağının mimari plan üretiminin gerçekleştirilmesi sürecinde karşılaşılan problemlerin nedenleri araştırılmıştır. ÇÜA ağı ile ilk denemelerde sonuç alınamamıştır. Problemin ÇÜA ağının yapısı ile ilgili olup olmadığının anlaşılması için ilk olarak düzgün bir veri seti ile ÇÜA ağı çalıştırılmıştır. Kullanılan ÇÜA mimarisi DCGAN'dır. DCGAN algoritmasının mimarisinde bir problem olup olmadığının anlaşılması için MNIST veri seti kullanılmıştır. MNIST veri seti ile eğitilen DCGAN üretimleri ve loss değerleri kontrol edilmiştir (Şekil 4.1).



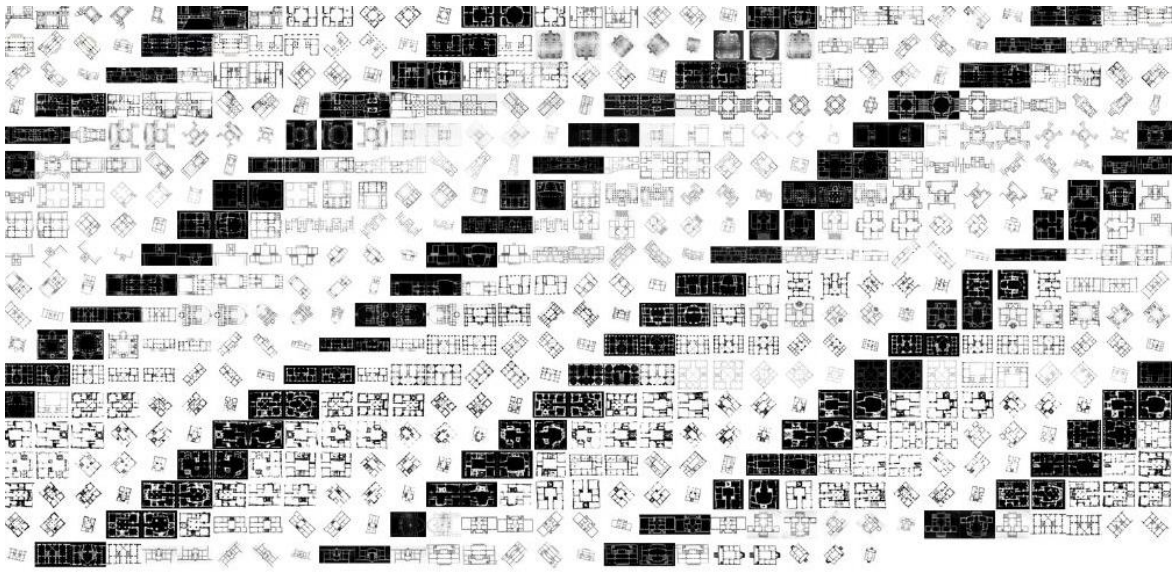
Şekil 4.1: DCGAN mimarisinin kontrolü amacıyla MNIST veri seti ile eğitilmiş algoritmanın kayıp fonksiyon grafiği.

MNIST ile elde edilen üretim çıktılarındaki görsel kalitesi yüksek olması nedeniyle ÇÜA mimarisinde sorun olmadığı kararlaştırılmış ve eğitim sırasındaki üretici ve ayrıştırıcı kayıp fonksiyon değerlerinin değişimi doğru kabul edilmiştir. Bu değerlendirme sonucunda mimari plan görselleri ile eğitilen ÇÜA ağındaki problemin veri seti olduğu kararlaştırılmış ve birçok farklı veri seti büyüklüğü ve farklı görsellerden oluşan veri setleri ile denemeler yapılmıştır. Bu bölümde, farklı

veri setleri ile gerçekleştirilen eğitim çıktıları, loss değerleri, veri setleri ile birlikte değerlendirilmiştir.

4.1 ÇÜA ile Veri Setinin Test edilmesi: 1. Deney

GAN ağının eğitimi için denenen ilk veri seti, mimar Andrea Palladio'nun 125 plan çiziminin arttırılması ile geliştirilen veri seti olmuştur. Bu veri setinde veri artırımı (data augmentation) tekniği olarak farklı açılarda döndürme, ayna simetriğini alma ve renk değerlerini tersine çevirme (invert) kullanılmıştır. Veri setindeki tüm imajların boyutları 128x128x3 olacak şekilde düzenlenmiştir.

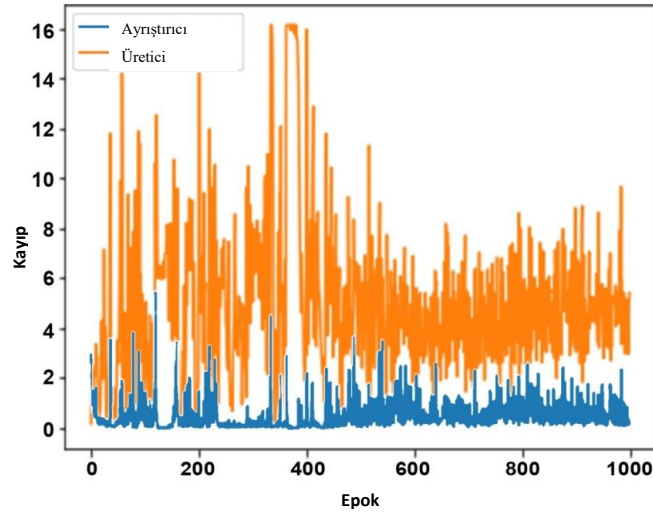


Şekil 4.2: 1. deneyde kullanılan, veri arttırıma yöntemi kullanılarak çoğaltılmış orjinal Palladyan planları veriseti.

Kullanılan DCGAN Mimarisi: ÇÜA eğitiminde kullanılan ilk algoritmada üreticinin ayrıştırıcıyı ikna edebilmesi için üretici katmanları ayrıştırıcıya göre daha güçlü olacak şekilde biçimlendirilmiştir. Üretici 6 konvolüsyonel ve 1 *tam bağlantılı* (fully connected) katmandan oluşmaktadır. Ayrıştırıcı ise 4 konvolüsyonel ve 1 tam bağlantılı katmandan oluşmaktadır.

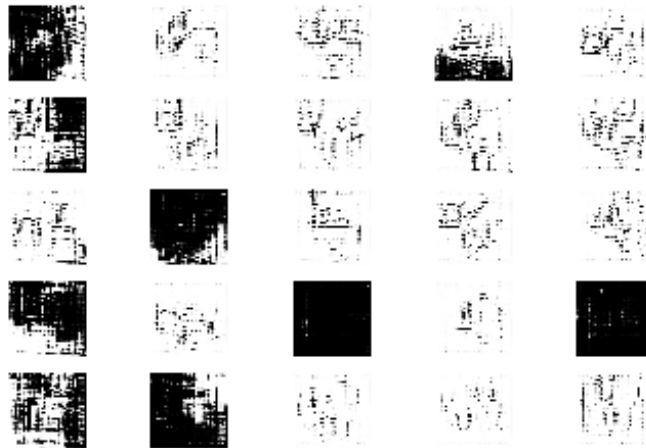
Kullanılan epok ve yığın büyüklükleri: Tüm eğitim 1000 epok kadar sürmüştür. Eğitimin 1000 epokta durdurulmasının sebebi, eğitim esnasında takip edilen loss değerlerindeki değişimin 400-1000 epok arasında anlamlı bir şekilde farklılık göstermemiş olmasıdır. Veri seti 32 imajdan oluşan yığınlar ile veri setine beslenmiştir. Üretim çıktıları her 50 epokta bir kayıt edilmiştir.

Kayıp Fonksiyon değerleri: Eğitim sonrasında kayıp fonksiyon değerlerinden oluşan grafik çıktısı alınmıştır. Kayıp fonksiyon değerlerine bakıldığında ilk 400 epokta sürekli değişen kayıp değerleri anlamlı bir grafik çizememiştir. 400 epok sonrasında kayıp değerleri hala stabil görünmese de 1000 epoka kadar ki değerlerde benzer kayıp değeri değişimi ile daha tutarlı bir grafik oluşturmuştur. ÇÜA ağı çıktısı olan görseller incelendiğinde bu loss değerlerindeki değişim ile paralellik içinde olduğu gözlemlenmiştir.



Şekil 4.3: 1. deneydeki eğitim sürecindeki üretici ve ayırıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Üretim çıktılarındaki görseller değerlendirildiğinde 400 epoka kadar anlamlı bir üretim gerçekleştiremediği gözlemlenmiştir. 400-1000 epok arasında üretimlerin gerçekleşmeye başladığı gözlemlenmiştir. Ancak üretimler hala net olduğu söylenemez.



Şekil 4.4: 1. deneyin DCGAN üretim çıktıları.

Değerlendirme: Kayıp değerlerindeki ilk 400 epoktaki stabil olmayan durum değerlendirildiğinde siyah üstüne beyaz çizimlerin ÇÜA eğitimi esnasında kafa karıştırıcı veri olabileceği yorumlanmıştır. Üretim çıktılarındaki görsellerde de bir siyah üstüne beyaz ve beyaz üstüne siyah üretimler gözlemlenmiş, ancak hiçbir net bir plan görseli oluşturamamıştır. Yine de üretimde plana dair bazı çizgiler gözlemlenmeye başlamıştır. Loss değerleri incelendiğinde ayrıştırıcının veri setini çok hızlı öğrendiği gözlemlenmiştir.

4.2 ÇÜA ile Veri Setinin Test edilmesi: 2. Deney

Bir önceki eğitimde veri seti içerisinde veri seti ile uyumsuz (outliner) veri olduğu değerlendirildiği yapılmıştır. Bu uyumsuz verilerin siyah üzerine beyaz çizgilerle çizilmiş olan plan görselleri olduğu kararlaştırılmıştır. Bu nedenle yeni eğitim siyah üzerine beyaz çizgili plan görsellerinin veri setinden silindiği veriler ile gerçekleştirilmiştir. Veri setinde döndürme ve aynalama ile veri artırımı yapılmış görseller ile birlikte Palladio'nun orijinal plan çizimleri de yer almaktadır. Veri setinde toplamda 128x128x3 şeklindeki imajlardan 744 adet bulunmaktadır.

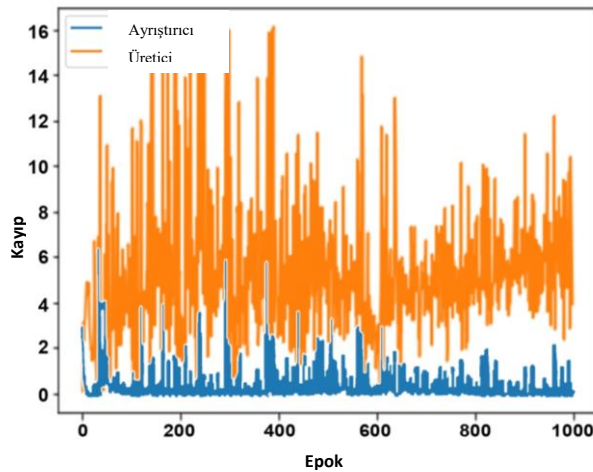


Şekil 4.5: 2. deneyde kullanılan, veri artırıma yöntemi kullanılarak çoğaltılmış orijinal Palladyan planları veriseti (renklerin ters çevrildiği veri arttırımı yöntemi ile elde edilen veriler çıkarılmıştır).

Kullanılan DCGAN mimarisi:GAN çıktılarındaki problemin veri seti kaynaklı olması durumuna odaklanıldığı için bu eğitim bölümünde bir önceki eğitimde kullanılan algoritma s tr kt r  kullanılmıştır. Bu eğitimde de  retici 6 konvol syonel ve 1 tam baėlantılı katmandan oluřmaktadır. Ayrıştırıcı ise 4 konvol syonel ve 1 tam baėlantılı katmandan oluřmaktadır.

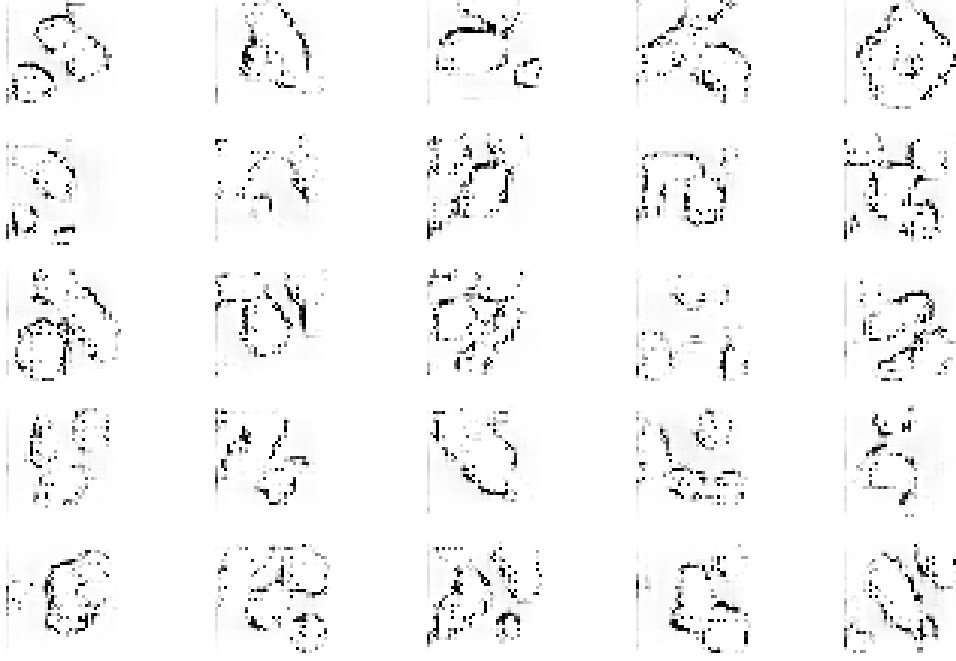
Kullanılan epok ve yığın b y kl kleri:Eėitim sırasında epok sayısı 1000 ile sınırlı tutulmuřtur. Eėer eğitim sonucunda hem kayıp deėerleri hem de  retim  ıktısı g rsellerde potansiyel bulunuyorsa, epok sayısı artırılarak eğitim yenilenmesi kararlařtırılmıştır. B ylece eğitim s recinde zamandan kaybetmemek planlanmıştır. Bu nedenle bu eğitim de 1000 epok ile sonlandırılmıştır. Veri seti 32 imajdan oluřan yığınlar ile veri setine beslenmiştir.  retim  ıktıları her 50 epokta bir kayıt edilmiştir.

Kayıp fonksiyon deėerleri:Kayıp deėerleri bir  nceki eėitimin kayıp deėerleri ile karřılařtırıldığında daha stabil bir eėitimin ger ekleřtiėi s ylenebilir. B ylece eğitimde veri seti i erisindeki siyah  zerine beyaz  izgilerle oluřturulan verilerin uyumsuz veri olduėu sonucuna varılmıştır. Veri artırımı y ntemi kullanılırken renklerin ters  evrilmesi eğitimde kafa karıştıracı veri olarak (outlier) karřılık bulmuřtur. Kayıp deėeri bir  ncekine g re daha stabil olsa da hala daha d zensizliėini korumakta ve ayrıştırıcının kayıp deėeri  ok hızlı d řerken  reticinin kayıp deėeri s rekli olarak y ksek deėerlerde dalgalanmaktadır.



řekil 4.6: 2. deneydeki eğitim s recindeki  retici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiėi.

Çıktılar: Üretim çıktılarından plan görseline dair bir bilgi bulunamamıştır. Belli bir alan oluşturma eğiliminde siyah piksellerin kümелendiğı gözlemlense de herhangi bir plan şeması görünmemektedir.



Şekil 4.7: 2. deneyin DCGAN üretim çıktıları.

Değerlendirme: Çıktılara bakıldığında her ne kadar kayıp değerleri daha düzgün değerlere ulaşsa da üretimin düzgün olmadığı değerlendirilebilir. Kayıp değerlerindeki düzelme ve siyah piksellerin alan oluşturmak amacıyla kümelenmesi veri setinin bir önceki deneye göre daha düzgün bir veri seti olması ile ilişkili olabilir. Bu durum her bir deneyde kullanılan veri setlerinin içindeki piksel dağılımlarının değerlendirildiğı bölümde detaylı bir şekilde açıklanmıştır. Bunun yanında 1000 epok eğitimle elde edilen sonuçların potansiyel barındırması nedeniyle aynı veri setinin daha yüksek epokta eğitilerek çıktıların değerlendirilmesi kararı alınmıştır.

4.3 ÇÜA ile Veri Setinin Test edilmesi: 3. Deney

Bu eğitim sürecinde 2. deney sırasında kullanılan veri seti kullanılmıştır. 128x128x3 boyutundaki 744 Palladio plan görseli eğitimde kullanılmıştır.

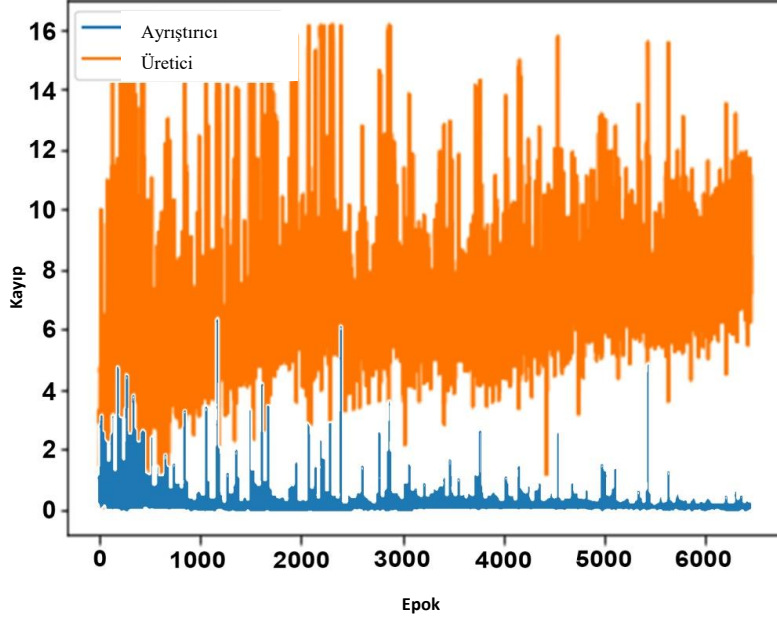


Şekil 4.8: 3. deneyde kullanılan, veri arttırma yöntemi kullanılarak çoğaltılmış orijinal Palladyan planları veriseti (2. deneydeki veriseti kullanılmıştır).

Kullanılan DCGAN Mimarisi: Bu eğitimde kullanılan süruktür de 2. deneyde kullanılan süruktür ile aynı tutulmuştur. Üretici 6 konvolüsyonel ve 1 tam bağlantılı katmandan oluşmaktadır. Ayırıştırıcı ise 4 konvolüsyonel ve 1 tam bağlantılı katmandan oluşmaktadır.

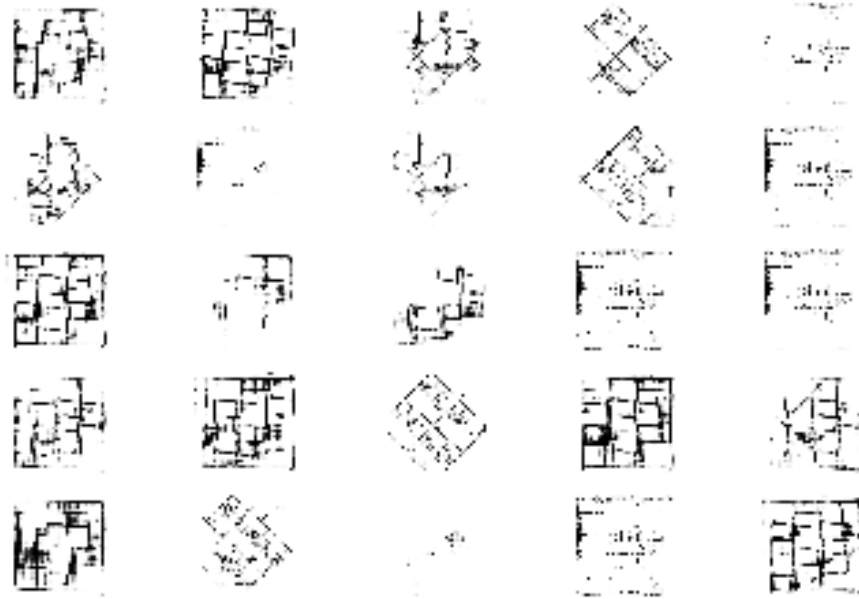
Kullanılan epok ve yığın büyüklükleri:2. deneyde kayıp değerleri ve üretim çıktılarının nitelikli olabilme ihtimali nedeniyle 1000 epokta eğitilen aynı algoritma 7000 epokta eğitilmiştir.

Kayıp fonksiyon değerleri:7000 epok sonrasında loss değerleri değerlendirildiğinde 1000 epok eğitimde elde edilen değerlerden daha stabil bir grafik ortaya çıktığı okunabilir. Diğer deneylerdeki aynı sorun burada da devam etmektedir. Üretici kayıp değeri sürekli farklı değerler alarak dalgalanmakta discriminator ise veriyi çok hızlı öğrenmektedir.



Şekil 4.9: 3. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Üretim çıktıları hem kendi içinde hem de 2. deney üretim çıktıları ile karşılaştırılarak değerlendirildiğinde, bu eğitimin sonucunda artık plan şemasına uygun çizimlerin üretilebildiği gözlemlenmiştir. Ancak bu üretimde görsellerin netlik durumları oldukça azdır. Üretilen plan görselleri çok fazla gürültü (noise) barındırmaktadır. Bu nedenle plan düzlemindeki mekansal organizasyonlar okunamamaktadır.



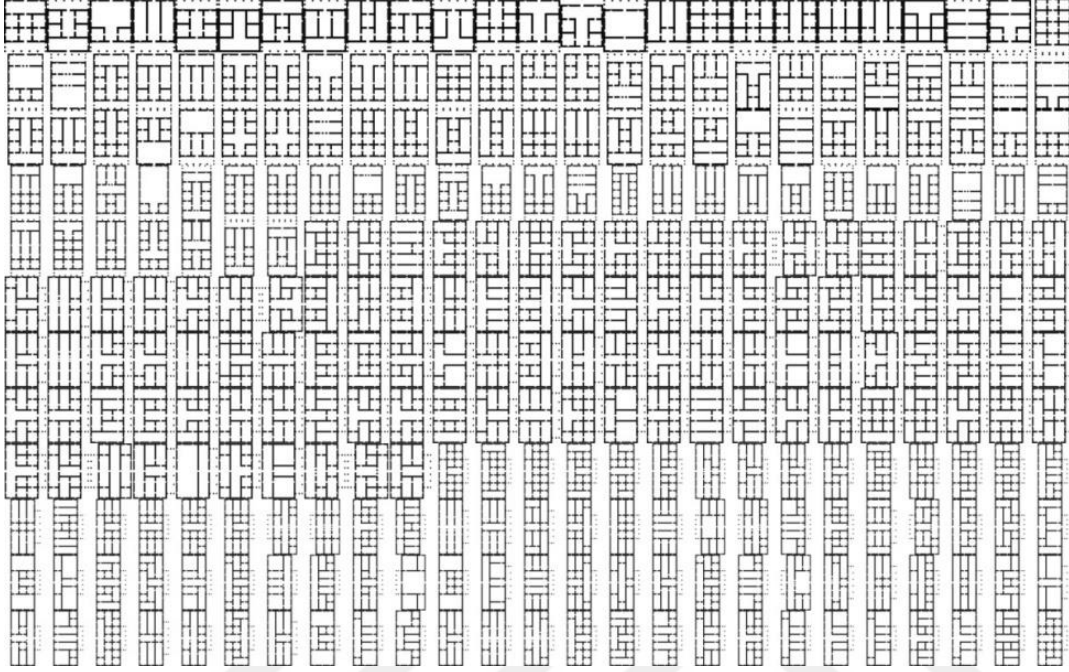
Şekil 4.10: 3. deneyin DCGAN üretim çıktıları.

Değerlendirme: Bu eğitim sonucunda elde edilen çıktılarla birlikte, veri setinin kendi içinde tutarlı olması (outlier verinin veri setinden temizlenmiş olması) ve eğitimdeki iterasyon miktarının yüksek tutulmasının gerekliliği sonucuna varılmıştır. Üreticinin üretim çıktılarının kalitesinin düşük olma sebebinin yine veri seti ile ilgili olduğu düşünülmektedir. Tüm bu denemeler sırasında kullanılan veri setinin görsel olarak çok fazla çeşitlilik bulundurması nedeniyle heterojen bir veri seti olmasının problemli olabileceği düşünülmüştür. Bu nedenle bir sonraki deneyde Palladio planlarını temsil eden temiz verilerin üretilmesi ile, orijinal Palladio çizimleri veri setinin veri artırımı yapılması kararlaştırılmıştır.

4.4 ÇÜA ile Veri Setinin Test edilmesi: 4. Deney

Bu eğitim öncesinde veri setinin problemli olduğu düşüncesiyle, veri setinin yeni bir veri seti artırımı yöntemi ile çoğaltılması kararlaştırılmıştır. Bu nedenle bu eğitim sürecinin ilk bölümünde veri seti artırımına yönelik araştırma yapılmıştır. Diğer eğitimlerde kullanılan veri seti Andrea Palladio'nun plan görsellerinden oluşmaktadır. Bu nedenle Palladio'nun plan görsellerinin yeniden üretilmesi kararlaştırılmıştır. Palladio planları ile ilgili biçim gramerleri çalışması ile Palladio plan kurallarına dair literatürde birçok çalışma ile karşılaşılmaktadır. Bu nedenle veri artırımı yöntemi olarak biçim gramerlerinin kullanılması fikri önerilmiştir. Bu nedenle biçim gramerleri kullanarak Palladio plan şeması oluşturabilen yazılımlar incelenmiştir. İnternet üzerinden üretim yapılabilen Palladian Grammar yazılımı kullanım için uygun bulunmuştur (URL 10). Bu yazılımda farklı gridler ile Palladio biçim gramerlerine uygun planlar üretilmiştir. 3x3 gridlerden 24 plan şeması, 3x4 gridlerden 51, 3x5 plan şemasından 32, 5x3 plan şemasından 103 ve 7x3 plan şemasından 100 adet olmak üzere toplamda 311 adet Palladio biçim gramerine uygun plan üretilmiştir. Bu planlar .dxf uzantılı dosyalar olarak kayıt edilebilmektedir. Bu nedenle her bir .dxf dosyası .jpeg formatına dönüştürülmüştür. 311 adet görsel, 4 farklı dönme açısında döndürülerek ve 2 yönde ayna simetrisi alınarak toplamda 6 dönüşüm ile veri artırımı yapılarak 1653 görsel elde edilmiştir. 311 imaj ile 6 dönüşüm 1866 görsel üretmesi gerekirken veri kayıpları ile birlikte 1653 görsel üretilmiştir. Görsel veride bu dönüşüm eylemlerinin yapıldığı yazılım ImBatch programıdır. Verideki kayıp nedeni bazı dosyaların bu programda, tüm görseller .jpeg de olsa, bazı veriler bu veri artırımı yazılımı tarafından okunamamıştır. Sonuç

olarak 1653 görselden oluşan yeni bir veri seti oluşturulmuştur. Bu veri seti Palladio'nun orijinal plan çizimlerinden oluşan 744 görsellik veri seti ile birleştirilerek toplamda 2397 görselden oluşan bir veri seti elde edilmiştir. Verideki her bir görselin şekli 128x128x3 olarak tanımlanmıştır.

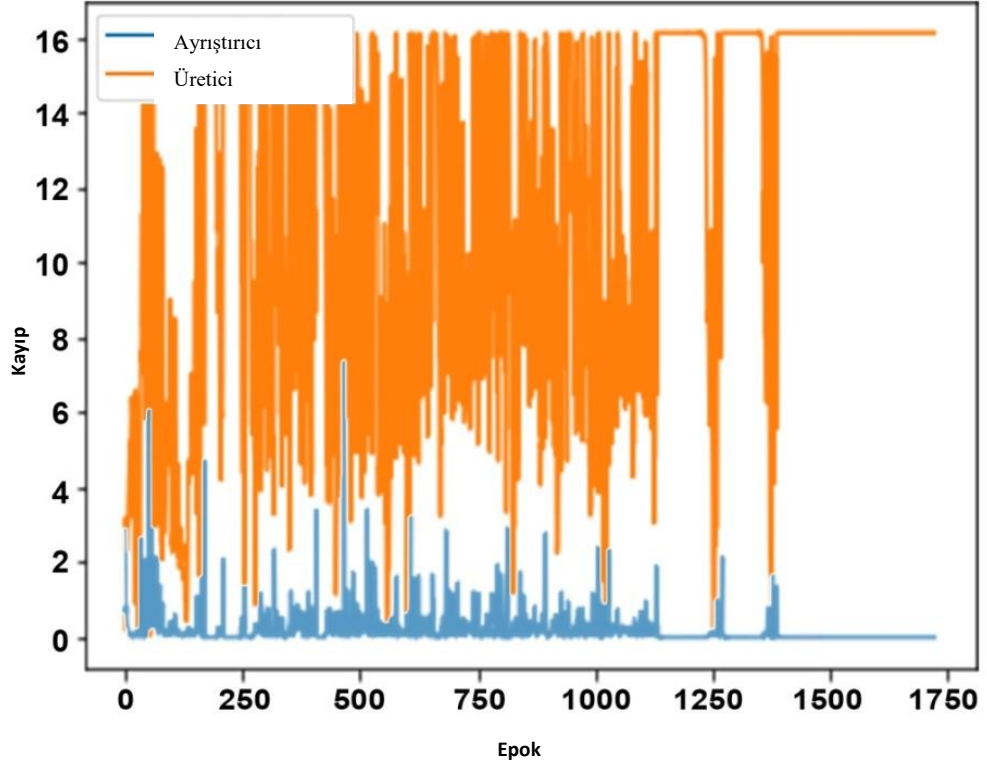


Şekil 4.11: 4. deneyde kullanılan, veri arttırımı yöntemi ile sayısı artırılmış orijinal Palladyan planları ve GRAPE-SGI ile türetilen Palladyan plan şemalarından oluşan veriseti.

Kullanılan DCGAN Mimarisi:Bu eğitimde kullanılan mimari de 2. ve 3. deneyde kullanılan sütrüktür ile aynı tutulmuştur. Üretici 6 konvolüsyonel ve 1 tam bağlantılı katmandan oluşmaktadır. Ayrıştırıcı ise 4 konvolüsyonel ve 1 tam bağlantılı katmandan oluşmaktadır.

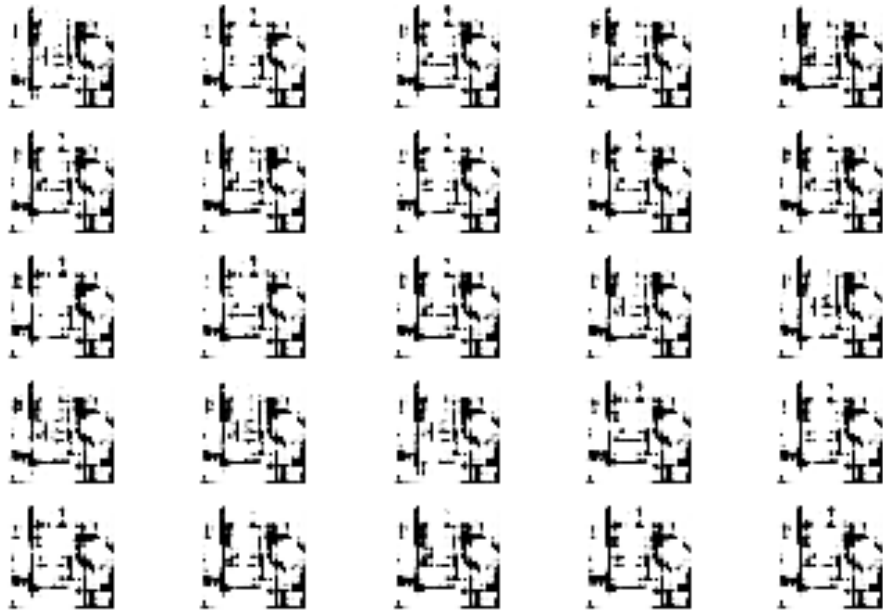
Kullanılan epok ve yığın büyüklükleri:Eğitim toplamda 7000 epok sürdürülmüştür. Batch büyüklüğü 32 olarak ayarlanmış ve 50 epokta bir eğitim çıktısı kayıt edilmiştir.

Kayıp fonksiyon değerleri: Kayıp değerleri diğer eğitimlerdeki kayıp grafiklerinden oldukça farklı bir sonuç vermiştir. Bazı epoklarda üretici ve ayrıştırıcı kayıp değerleri hiç değişmemektedir. Üreticinin kayıp değeri çok büyük aralıklarda sürekli oynamış ve stabil bir eğitim elde edilememiştir.



Şekil 4.12: 4. deneydeki eğitim sürecindeki üretici ve ayırıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Eğitim çıktıları değerlendirildiğinde, eğitimin aynı üretimleri tekrar üretme problemi (*mode collapse*) ile sonuçlandığı gözlemlenmiştir. Eğitim sırasındaki epoklarda sürekli aynı geometrik temsiller üretilmiştir. Bunun yanında bu geometrik temsiller herhangi bir plan şemasına karşılık gelmemektedir.



Şekil 4.13: 4. deneyin DCGAN üretim çıktıları.

Değerlendirme: Elimizdeki Palladio plan şemalarının veri artırımı için biçim gramerlerinin kullanılması olumsuz sonuçlanmıştır. Biçim gramerleri ile üretilmiş görsellerle, orijinal planların karıştırılarak oluşturulan veri seti eğitim sırasında kafa karıştırıcı olmuştur. Veri seti içindeki veriler birbiri ile uyumsuz veri olduğu sonucuna varılmıştır. Bu nedenle biçim gramerlerinin veri artırımı yöntemi olarak kullanılması, sonuçta üretilen verinin aykırı olmaması ile mümkündür. Biçim gramerleri kuralları ile veri artırımı kullanılarak veri seti boyutunun büyütülmesi olumsuz sonuçlanmıştır. Veri seti değerlendirilmesi ve algoritmaya olumsuz etkisinin nedeni ileri ki bölümlerde detaylı bir şekilde açıklanmıştır.

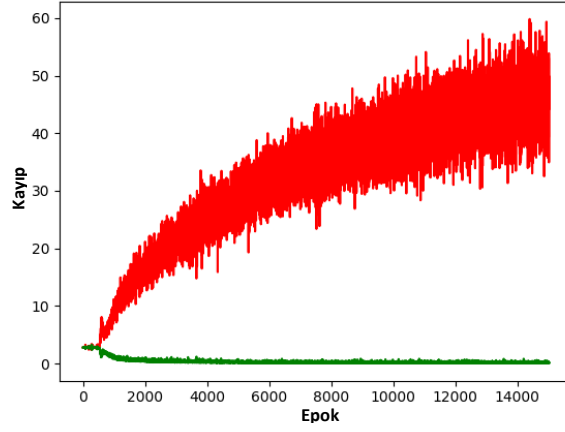
4.5 ÇÜA ile Veri Setinin Test edilmesi: 5. Deney

Veri artırımı ile ilgili 1. ve 4. deney sonucundaki olumsuz sonuçlar nedeniyle veri artırımı yapılmasından vazgeçilmiştir. Tüm öklidyen dönüşümler veri setinden temizlenerek 125 orijinal Palladio plan görselinden oluşan veri seti bu eğitimde kullanılmıştır. Bunun yanına veri setindeki görsel verinin biçimi 128x128x3 olmasının da eğitimde problem çıkarma ihtimali üzerinde durulmuştur. Veri seti 3 kanal yerine tek kanallı olabilmesi için, eğitime verilmeden önce grayscale renklere dönüştürülmüştür. Böylece 128x128x1 boyutundaki imajlar ile eğitim gerçekleştirilmiştir.

Kullanılan DCGAN Mimarisi: Bu eğitimde kullanılan mimaride üretici 3 konvolüsyonel katmandan oluşurken ayrıştırıcı 4 konvolüsyonel katmandan oluşmaktadır.

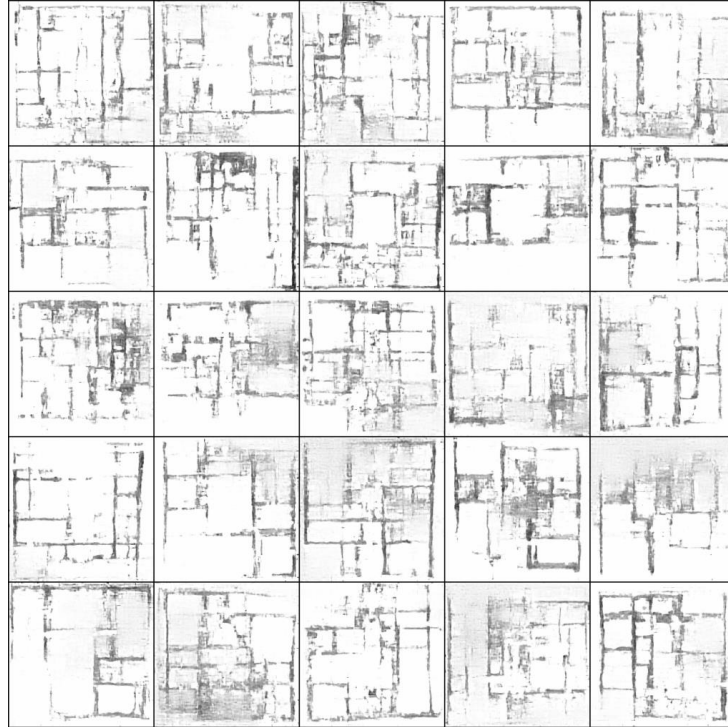
Kullanılan epok ve yığın büyüklükleri: Bu eğitim 15000 epokta yaklaşık 15 saat sürede tamamlanmıştır.

Kayıp fonksiyon değerleri: Kayıp değerleri eğitim süresince takip edilmiştir. Kayıp değerinin sürekli olarak artış eğiliminde olması problemli olduğu düşünülmüştür. Ayrıştırıcı kayıp değeri oldukça hızlı azalmakta ancak üretici fonksiyonun kayıp değerleri sürekli değişken şekilde artmaktadır.



Şekil 4.14: 5. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: ÇÜA üretim çıktıları değerlendirildiğinde olumlu sonuçlarla karşılaşılmıştır. İlk kez ÇÜA ağı okunaklı planlar üretebilmiştir. Bunun yanında planlar detaylı incelendiğinde Palladio plan şemasına uygun planlar olmasının yanında, yeni tanımlanan plan şemaları da görülebilmektedir. Plan şemasında 2 merdiven kovalı plan şemaları planların farklılaştığı üretimlerdendir. Bazı noktalarda gri bölgeler görünmektedir. Bunun sebebi orijinal planlardan oluşan veri setinin oluşturulurken photoshop yazılımında yapılan veri temizleme sürecindeki kalan bazı gri renklerden kaynaklı olduğu düşünülmektedir.



Şekil 4.15: 5. deneyin DCGAN üretim çıktıları.

Değerlendirme: Üretim çıktıları bu ÇÜA üretiminde doğru sonuçlanmıştır. Tüm deneylerle birlikte üretimler karşılaştırıldığında veri setinin büyüklüğünden çok veri seti içerisindeki verinin algoritma tarafından okunurluğunun önemli olduğu gözlemlenmiştir. 3 kanallı RGB değerlerinin eğitim sırasında kafa karıştırıcı olduğu, ve bu nedenle tek kanallı grayscale ile eğitimin doğru sonuçlandığı gözlemlenmiştir. Bunun yanında epok sayısının yüksek tutulması ile birlikte üretici daha iyi öğrenebilmiş ve üretimler netleşebilmiştir. Üretimlerdeki gri bölgelerin oluşmaması için veri seti oluşturulurken verinin temizlenmesi sürecine daha fazla dikkat edilmesi gerektiği de ortaya çıkmıştır. Plan üretimindeki dizilimlere bakıldığında Palladio plan şemasına benzer üretimlerin yanında, yeni şemaların da üretildiği okunabilmektedir. Bu durum algoritmanın keyfi olarak tasarlama eylemine katılmaya başladığı anlar olarak okunabilir. Bir diğer önemli nokta algoritmanın aşırı uyum (*overfitting*) yapmamasıdır. Üretim çıktılarında veri setindeki verilerle aynı görsellerle karşılaşılmamıştır.

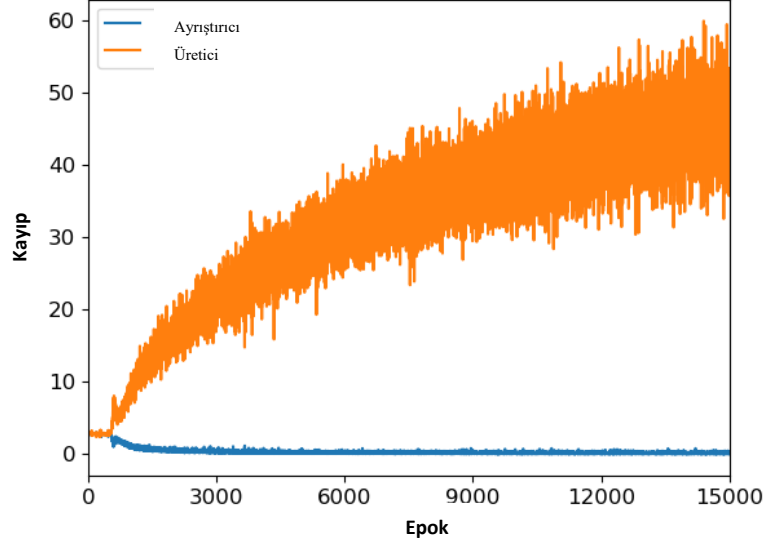
4.6 ÇÜA ile Veri Setinin Test edilmesi: 6. Deney

5. deneydeki olumlu sonuçlar sonrasında biçim gramerleri ile üretilen Palladio plan şemalarından oluşan veri setinin de yeniden eğitilmiştir. Biçim gramerlerindeki veri seti hiçbir veri artırımını uygulanmamış hali ile 320 plan görseli ile eğitim gerçekleştirilmiştir. Görselle 128x128x3 boyutundadır. Görseller 128x128x1 boyutuna dönüştürülerek eğitim gerçekleştirilmiştir.

Kullanılan DCGAN Mimarisi: Bu eğitimde de 5. deneydeki gibi kullanılan mimaride üretici için 3 konvolüsyonel katman ayrıştırıcı için 4 konvolüsyonel katman kullanılmıştır.

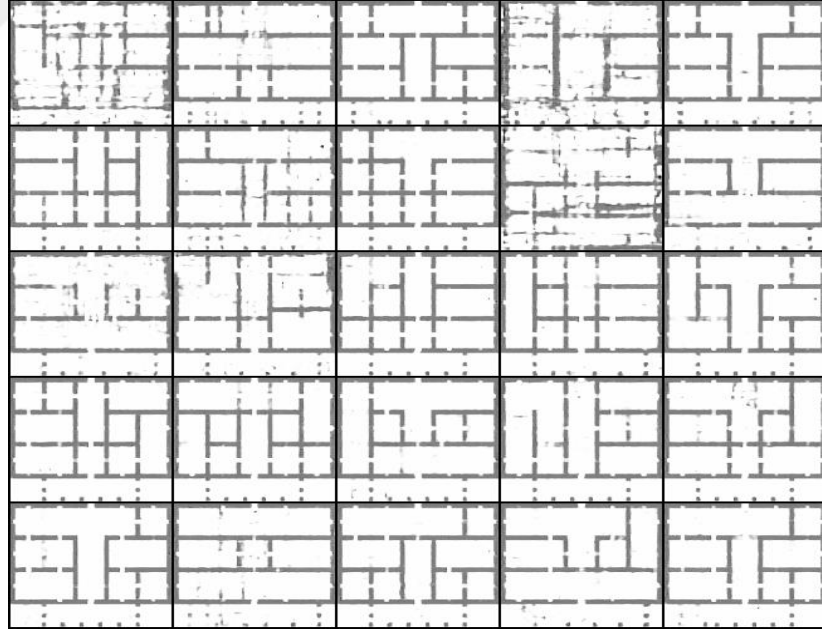
Kullanılan epok ve yığın büyüklükleri: Tüm eğitim 15000 epok ile 13 saat sürede tamamlanmıştır.

Kayıp fonksiyon değerleri: Kayıp değerleri eğitim süresince takip edilmiştir. Kayıp değerinin sürekli olarak artış eğiliminde olması problemli olduğu düşünülmüştür. Ayrıştırıcı kayıp değeri oldukça hızlı azalmakta ancak üretici fonksiyonun kayıp değerleri sürekli değişken şekilde artmaktadır.



Şekil 4.16: 6. deneydeki eğitim sürecindeki üretici ve ayırıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Çıktılar incelendiğinde üretimin doğru gerçekleştiği gözlemlenmiştir. Bir önceki deney için söylenenler bu deney için de geçerlidir. Plan şemalarından bazıları Palladio planlarına benzer kurgudayken, bazıları yeni plan şemaları olarak karşımıza çıkmaktadır.



Şekil 4.17: 6. deneyin DCGAN üretim çıktıları.

Değerlendirme: ÇÜA ağının Palladio biçim gramerine uygun üretimler gerçekleştirebildiği görüşmüştür. Bunun yanında biçim gramerine uymayan planlar da önerdiği gözlemlenmiştir. Genel bir bakışla algoritmanın aşırı uyum yapmadığı düşünülmektedir. Dikkat çeken bir diğer nokta da ÇÜA ağının üretiminin sürekli 5x3

grid üzerinden üretilmesidir. Veri setinde farklı boyutlardaki grid sistemi bulunsa da, üretimlerin bu grid dışına nadiren çıktığı gözlemlenmiştir. Veri setinin değerlendirilmesi ileriki bölümlerde detaylı bir şekilde yapılarak eğitime olumlu etkisinin nedeni açıklanmıştır.

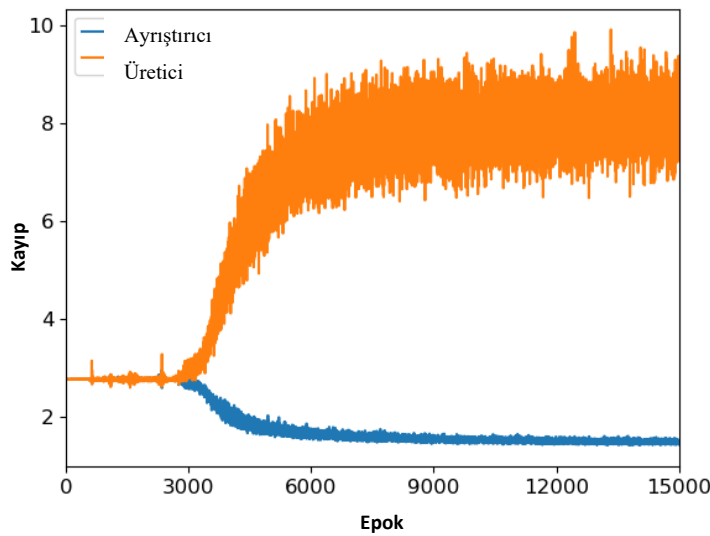
4.7 ÇÜA ile Veri Setinin Test edilmesi: 7. Deney

6. deneydeki olumlu sonuçlar sonrasında biçim gramerleri ile üretilen Palladio plan şemalarından oluşan veri setinin de yeniden eğitilmiştir. Biçim gramerlerindeki veri seti hiçbir veri artırımı uygulanmamış hali ile 320 plan görseli ile eğitim gerçekleştirilmiştir. Görseller 128x128x3 boyutundadır. Görseller 128x128x1 boyutuna dönüştürülerek eğitim gerçekleştirilmiştir.

Kullanılan DCGAN Mimarisi: Bu eğitimde de 6. deneydeki gibi, kullanılan mimaride üretici için 3 konvolüsyonel katman ayrıştırıcı için 4 konvolüsyonel katman kullanılmıştır.

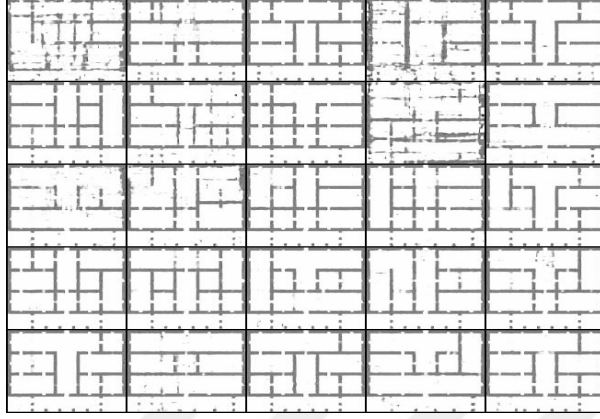
Kullanılan epok ve yığın büyüklükleri: Tüm eğitim 15000 epok ile 13 saat sürede tamamlanmıştır.

Kayıp fonksiyonu değerleri: Kayıp değerleri eğitim süresince takip edilmiştir. Kayıp değerinin sürekli olarak artış eğiliminde olması problemlili olduğu düşünülmüştür. Ayrıştırıcı kayıp değeri oldukça hızlı azalmakta ancak üretici fonksiyonun kayıp değerleri sürekli değişken şekilde artmaktadır.



Şekil 4.18: 7. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Çıktılar incelendiğinde üretimin doğru gerçekleştiği gözlemlenmiştir. Bir önceki deney için söylenenler bu deney için de geçerlidir. Plan şemalarından bazıları Palladio planlarına benzer kurgudayken, bazıları yeni plan şemaları olarak karşımıza çıkmaktadır.



Şekil 4.19: 7. deneyin DCGAN üretim çıktıları.

Değerlendirme: 7. deney çıktıları da 6. deneydeki gibi çoğunlukla çözünürlüğü düzgün plan şemaları okunaklı görsel üretimlerdir. 6. deneyden farklı olarak etiket yumuşatma (label smoothing) tekniği ile imaj verisindeki piksel değerleri manipüle edilerek eğitimin veriminin artırılması hedeflenmiştir. Kayıp fonksiyon değerlerine bakıldığında label smoothing'in olumlu etkileri gözlemlenmiştir.

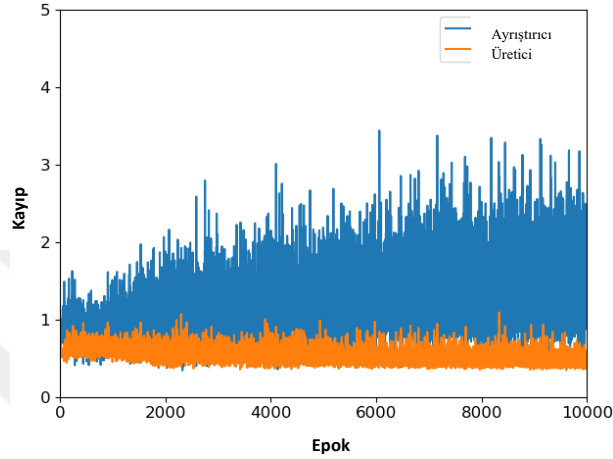
4.8 ÇÜA ile Veri Setinin Test edilmesi:8. Deney

Bu deney 7. deneyde kullanılan veri setinin düzenlenmiş hali ile gerçekleştirilmiştir. 7. deneyde kullanılan veri setindeki farklı grid düzenleri veri seti içerisinde farklı sayılarda bulunmaktadır. Veri seti içerisindeki bu dengesiz dağılımın üretim çıktılarına etkisi olabileceği düşünülerek 8. deneyde farklı grid düzenlerinden oluşan plan şemalarından eşit sayıda planlar ile veri seti oluşturulmuştur. 3x5, 5x3, 3x3, 7x3, 3x7 grid düzenlerindeki her bir plan şemasından 30'ar adet plan ile toplamda 150 plan üretilmiş ve DCGAN eğitiminde kullanılmıştır.

Kullanılan DCGAN Mimarisi: 7. Eğitimdeki veri setinin eğitime etkisi araştırılması nedeniyle bu eğitimde DCGAN mimarisi değiştirilmemiştir. Bu eğitimde de 3 konvolüsyonel katman üretici fonksiyon için kullanılırken 4 konvolüsyonel katman da ayrıştırıcı için kullanılmıştır.

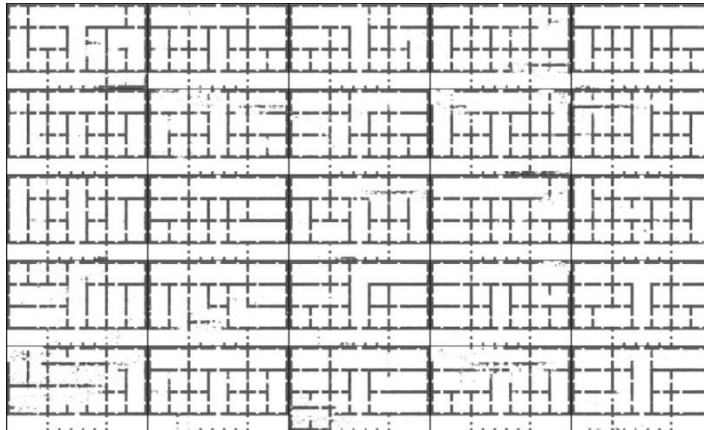
Kullanılan epok ve yığın büyüklükleri: Bu deneyde de epok sayısı ve yığın büyüklüğü 7. deneyle aynı olup yığın büyüklüğü 32 olarak seçilerek 15000 epok eğitim gerçekleştirilmiştir.

Kayıp fonksiyonu değerleri: Kayıp fonksiyonu 7. deneyin kayıp fonksiyon değerleri ile karşılaştırıldığında üretici fonksiyonun kayıp fonksiyon değerinin oldukça küçük değerler aldığı görülmektedir.



Şekil 4.20: 8. deneydeki eğitim sürecindeki üretici ve ayrıştırıcı fonksiyonlarının kayıp fonksiyon grafiği.

Çıktılar: Üretimlerde 7. deneyden farklı olarak üretimin 7x3 düzenindeki gridlerin baskın olduğu gözlemlenmiştir. 7. deneyde 5x3 grid düzeni üretimlerde baskın gelmiştir. 7. deneydeki veri setinde 5x3 düzenindeki verinin veri seti içerisinde en fazla sayıda bulunan grid düzenini oluşturmuştur. Bu deneydeki eşit sayıdaki grid düzenlerine rağmen üretimde 7x3 grid düzeni görülmektedir. Bunun başlıca sebebi piksel dağılım yoğunluğunun 7x3 grid düzenlerinde yoğunlaşması olabilir.



Şekil 4.21: 8. deneyin DCGAN üretim çıktıları.

Değerlendirme: 8. deneyde en önemli sonucu üretici kayıp fonksiyon değerinin oldukça küçük değerler almasıdır. Bu durum üretici fonksiyonun planları öğrenebildiğini göstermektedir. Bu deneyde de label smoothing tekniği kullanılmış ve düzenli bir veri seti ile de deney olumlu sonuçlar vermiştir.





5. VERİ SETLERİNİN DEĞERLENDİRİLMESİ

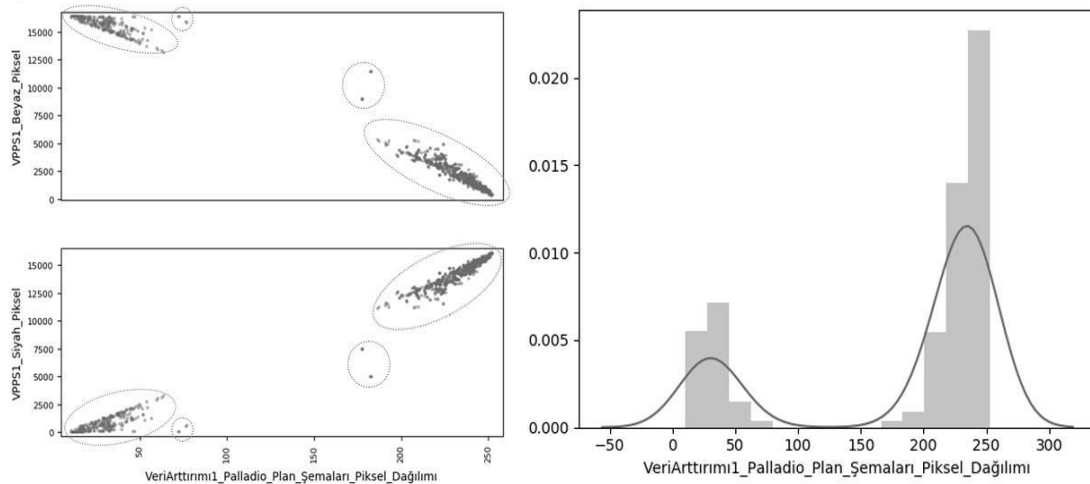
Makine öğrenmesi yöntemlerinde çözülmesi gereken problemlerden biri de veri problemidir. Algoritmanın verimli eğitimi için modelin doğru kurgulanmasının yanında verinin büyüklüğü ve temizliği de eğitimin verimini etkilemektedir. Geleneksel makine öğrenimi yöntemlerinde ve derin öğrenme yöntemlerinde de veri seti oluşturulurken gerçek dünyayı temsil edebilen bir veri seti oluşturulmalıdır. Veri bilimcileri gerçek dünyayı temsil eden her tür büyük verinin *normal dağılım* (*Gaussian / Normal Distribution*) içerisinde temsil edilebileceğini savunmaktadır. Bu sonuç *merkezi limit teoreminin* (*central limit theory*) sonucudur. 1812 yılında Laplace'ın çalışmalarını 1901 yılında geliştiren Rus matematikçi Aleksandr Lyapunov'un metamatiksel olarak net bir şekilde büyük bir verinin değişkenlerinin aritmetik ortalamalarının sıfır olacağı yani normal bir dağılım oluşturacağını ispat etmiştir (URL11). Bu nedenle büyük verinin kullanıldığı makine öğrenmesi tekniklerinde bu teoreme güvenilmektedir. Normal dağılıma sahip bir veri gerçek dünyayı temsil etme bakımından güvenilir olduğu söylenebilir. Normal dağılıma sahip olan bir veriden makine öğrenmesi yöntemleri ile doğru çıkarımlar yapılabilir. Yani veri setinin dağılımı ile algoritmanın eğitim verimi açıklanabilir. Bu nedenle bu bölümde her bir deney için kullanılan farklı veri setlerinin dağılımlarının değerlendirilmesi yapılmıştır.

Her veri setinde veri özniteliklerle temsil edilmektedir. Palladyan planları ve plan şemalarından oluşturduğumuz eğitim veri setinde öznitelikler oldukça azdır. Siyah beyaz imajlardan oluşan veri setindeki öznitelikler sadece siyah ve beyaz piksellerdir. Makine öğrenmesinden her bir öznitelik bir boyut olarak tanımlanır. Beyaz ve siyah pikseller ise grayscale değerleri ile sadece bir kanaldan 0 ile 255 arasında bir değer almaktadır. Bu nedenle aslında öznitelik sadece bir tane grayscale değeri olarak karşımıza çıkmaktadır. Yani algoritma sadece bir öznitelik üzerinden çıkarım yapmaya çalışacaktır. Makine öğrenmesi algoritmalarının eğitiminde boyutluluk bir problemidir. Çok özniteliğe sahip olan veri setlerinden çok boyutluluk problemi varken az özniteliğe sahip veri setlerinde de az boyutluluk (low feature)

problemi oluşmaktadır. Derin öğrenme algoritmalarından ÇÜA algoritmasının eğitimi gerçekleştirilen eğitim deneylerinin her birinden Palladyan plan şemalarından oluşan az boyutlu veriler kullanılmıştır. Bu nedenle veri seti değerlendirilken veri setindeki her bir görseldeki siyah ve beyaz piksel miktarlarına bakılmıştır. Bunun yanında veri setindeki dağılım grafiği her bir görselin tüm piksellerin aldığı renk değerlerinin aritmetik ortalamaları ile oluşturulmuştur.

5.1 1. Deney Veriseti Değerlendirilmesi

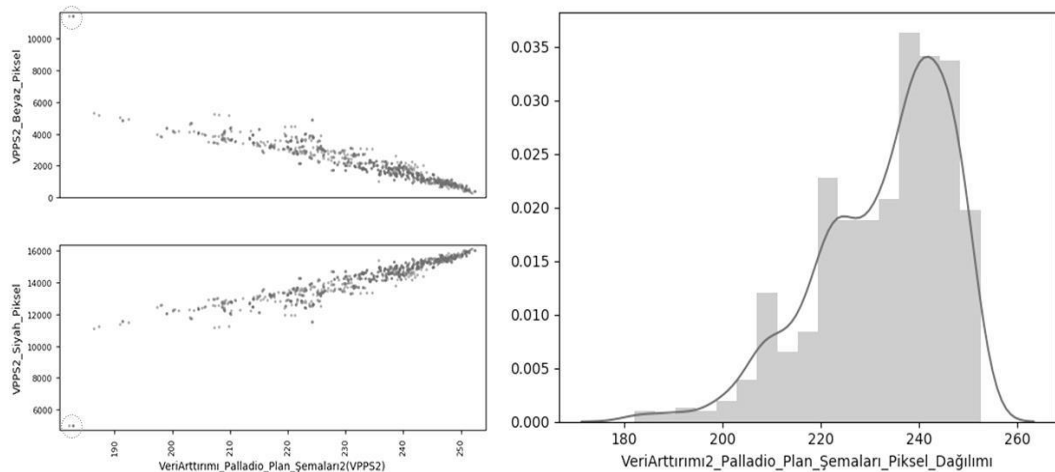
1. deneyde kullanılan veri seti orijinal 125 adet orijinal Palladio villa plan şemalarının döndürme, aynalama, siyah arka fonda beyaz çizim, beyaz arka fonda siyah çizim ve görselin biçimsel değişimi (distortion) ile veri seti arttırımı yapılmıştır. Sonuçta elde edilen veri seti 1000 imajdan olmuştur. Veri setindeki her bir imajdaki siyah ve beyaz piksel dağılımına bakıldığında veri setinde iki farklı küme olduğu gözlemlenmiştir. Bunun yanında distortion kaynaklı olabileceği düşünülen *ayrıksı (outlier)* görünen başka kümelenmelerin de olduğu gözlemlenmiştir. Piksel renk değerleri ortalamasına bakıldığında da bu iki farklı grubun iki farklı tepe oluşturduğu gözlemlenebilmektedir. Veri setindeki bu ikiliğe neden olan veri arttırımının siyah fon üstüne beyaz çizim ve beyaz fon üstüne siyah çizim olduğu açık bir şekilde söylenebilir. Bu nedenle veri arttırımı için kullanılan renk değiştirme yöntemi veri setinin dağılımını olumsuz yönde etkilemiştir. DCGAN algoritmasının üretim çıktılarının düzgün olamaması doğrudan veri setindeki düzensizlik nedeniyle olduğu söylenebilir.



Şekil 5.1: 1. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

5.2 2. ve 3. Deney Veriseti Değerlendirilmesi

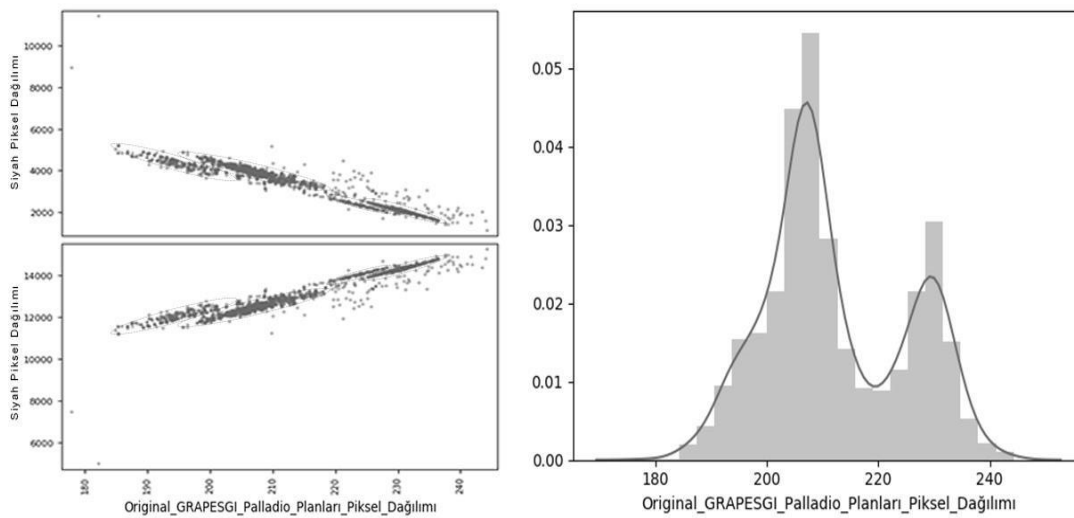
2. ve 3. deneyde kullanılan veri seti orijinal 125 adet orijinal Palladio villa plan şemalarının döndürme, aynalama ve görselin biçimsel değişimi (distortion) ile veri seti artırımı yapılmıştır. Bu veri artırımı yöntemleri içinde siyah ve beyaz renklerin yer değiştirdiği veri artırımı kullanılmamıştır. Bu veri setindeki siyah ve beyaz piksel dağılımlarına bakıldığında tek bir kümelenmenin gerçekleştiği görülmektedir. Bunun yanında distortion kaynaklı ayırksı verilerden oluşan kümelenmelerin de olduğu gözlemlenmektedir. Piksel renk dağılım ortalamalarına bakıldığında her ne kadar normal dağılımı yakalayamasa da tek tepe oluşturabilen bir dağılımın olduğunu görmekteyiz. Dağılımda eğri tam bir çan eğrisi oluşturamamaktadır. Bu da eğitim verisinde ayırksı verilerin olduğunu işaret etmektedir. 2. ve 3. deneydeki DCGAN çıktılarına bakıldığında her ne kadar kayıp fonksiyon değerleri stabil olmasa da, eğitimin görsel çıktıları 1. deneydeki eğitime kıyasla daha doğru sonuçları göstermektedir. Böylece 2. ve 3. deneyde kullanılmış olan veri setinin 1. deneyde kullanılan veri setinden daha doğru bir veri seti olduğu söylenebilir. Bu nedenle piksel renk değiştirme veri artırımı yönteminin doğru olmayan bir veri artırımı yöntemi olduğu gözlemlenmiştir. Piksel renk değiştirme tekniğinin eğitimde verimi arttırabilmesi için veri seti içindeki verinin miktarının daha da arttırılması gereklidir. Piksel rengi veri artırımı tekniği veri setindeki varyansı arttırmıştır. Varyansı yüksek olan veri setinin de eğitimde verimli çalışabilmesi için daha fazla veri kullanılarak varyansının dengelenmesi gerekmektedir. Varyans ile veri setindeki çeşitliliği işaret edilmektedir. Sayıca kısıtlı bir veri setinin varyansının yüksek olması eğitilen algoritmanın doğru genelleme ve çıkarım yapamamasına neden olmaktadır.



Şekil 5.2: 2. ve 3. deneylerde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

5.3 4. Deney Veriseti Değerlendirilmesi

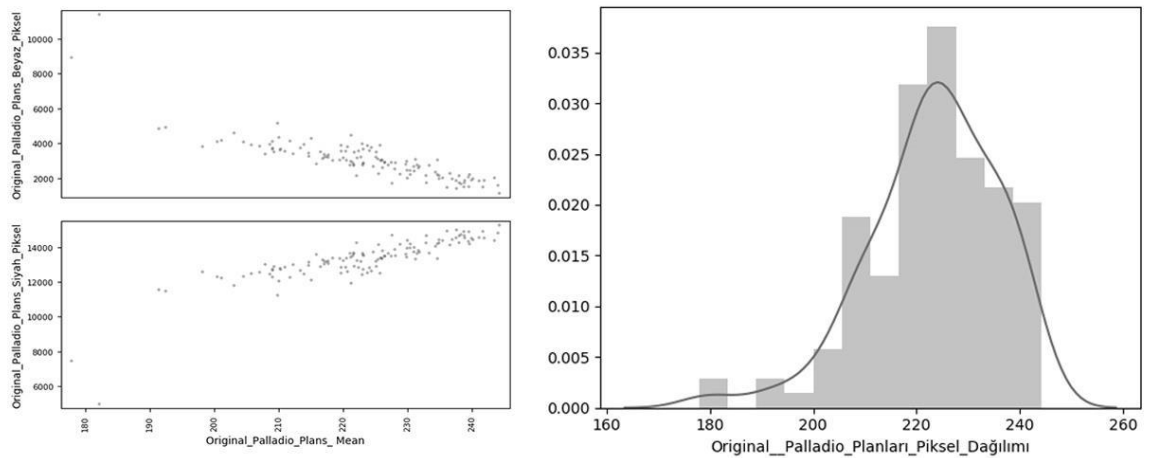
4. deneyde kullanılan veri setinde 3.deneydeki veri arttırımı yöntemlerinin yanında sentetik olarak Palladyan grameri ile Palladyan plan şemaları türetilerek veri setine eklenmiştir. Verinin arttırılmasının istenmesinin nedeni 1., 2. ve 3. deneydeki üretim çıktıları ve kayıp fonksiyon değerlerinin verimli bulunmamasının veri setindeki veri sayısı ile ilgili olduğu düşüncesidir. Ancak 4. deneyde kullanılan veri arttırımı yöntemi tamamen başka bir veri seti üreterek veri setinde iki farklı verinin oluşmasına neden olmuştur. Veri setindeki siyah ve beyaz piksel miktarlarının gösterildiği grafiğe bakıldığında 4 farklı kümelenmenin olduğu gözlemlenmiştir. Bunun yanında piksel renk değerlerinin ortalamalarının dağılımına bakıldığında iki tepe olduğu görülmektedir. Bu da veri setinde iki farklı tipten verinin bulunduğunu açık bir şekilde göstermektedir. Bu durum eğitimin verimini oldukça düşürmüş ve eğitim anında *mode collapse* problemi gözlemlenmiştir. Mode collapse probleminde üretimler birbirinin aynı olmaktadır. Diğer bir taraftan ise üretimler hem birbirinin aynısı hem de doğru olmayan üretimler olarak görülmektedir. Bu durumun en büyük nedeni yine veri setinde kullanılan veri arttırımı yönteminin hatalı oluşudur. Veri arttırımı yöntemlerinden piksel renklerinin değiştirilmesi ve biçim gramerleri kullanılarak yeni Palladyan plan şemalarının oluşturulması veri setinde ayrık veri gruplarının oluşmasına sebep olmuştur. Bu yapılan deneyle birlikte veri setinin oluşturulması sırasında kullanılan tüm veri arttırımı yöntemlerinden vazgeçilerek sadece orijinal verilerin kullanıldığı veri seti ile eğitimlerin gerçekleştirilmesi kararlaştırılmıştır.



Şekil 5.3: 4. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

5.4 5. Deney Veriseti Değerlendirilmesi

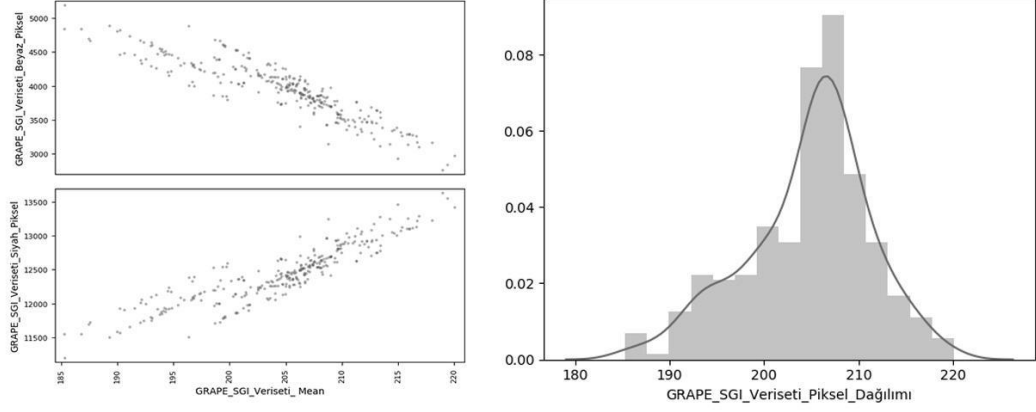
5. deneyde kullanılan veri seti 125 orijinal Palladio plan şemasından oluşan veri setidir. Bu veri setinde hiçbir veri arttırımı yöntemi kullanılmamış sadece veri setinin orijinal görselleri ile DCGAN eğitimi gerçekleştirilmiştir. Bu deneyde kullanılan veri seti 1., 2., 3. ve 4. deneyde kullanılan veri setleri ile karşılaştırıldığında her imaj için piksel renk değerlerinin aritmetik ortalamaları dağılımının daha düzenli olduğu gözlemlenmiştir. Düzenli olmasına rağmen dağılımın tam bir çan eğrisi de oluşturamadığı gözlemlenebilmektedir. Veri seti her ne kadar sadece orijinal Palladio plan şemalarından oluşması nedeniyle homojen bir veri seti gibi algılanabilse de planlar biçimsel olarak çeşitlendiği için veri setindeki dağılım tam bir çan eğrisi oluşturamamaktadır. Orijinal Palladio plan şemalarından oluşan veri setindeki dağılımın daha düzenlileştirilebilmesi için veri setindeki veri miktarının arttırılması gerekmektedir. Ancak daha önceki deneylerde gerçekleştirilen öklidyen dönüşümler ve renk değiştirme ile gerçekleştirilen veri arttırımı yöntemlerinin veri setini daha fazla karmaşıktırdığı gözlemlenmiştir. Diğer deneylerle DCGAN üretim çıktıları ve kayıp fonksiyon değerleri karşılaştırıldığında daha iyi sonuçlar alındığı gözlemlenmiştir. Bu nedenle plan şemalarının türetilmesi durumunda herhangi bir veri arttırımı yönteminin sağlıklı olamayacağı bunun nedeninin plan verisinin biçimsel bir veri olması herhangi bir veri arttırımı için kullanılan yöntemin plan biçimini doğrudan etkileyebilmesi nedeniyledir. Amacımız plan tanıma eğitimi olsaydı veri arttırımı yöntemlerinin işe yarama ihtimali vardı ancak tezin hedefinde plan şeması üretimi olması nedeniyle DCGAN algoritmasında veri arttırımı yönteminin plan şeması üretiminde verimli çalışmadığı gözlenmiştir.



Şekil 5.4: 5. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

5.5 6. ve 7. Deney Veriseti Değerlendirilmesi

6. ve 7. deneylerde kullanılan veri seti GRAPE-SGI (biçim grameri yorumlayıcı) ile türetilen sentetik Palladyan plan şemalarını içeren veri setidir. Veri seti GRAPE-SGI ile türetildikten sonra herhangi bir veri artırımı yöntemi kullanılmamıştır. Bu veri seti üretiminde Palladyan gramer kurallarına göre üretilen 320 plan şeması kullanılmıştır. GRAPE-SGI ile Palladyan plan şeması üretiminde öncelikle bir ızgara sistemi üretilmektedir. Izgara sistemi Palladyan plan şemalarında daha çok 5x3'lük ızgara sisteminde gözlemlenmektedir (Stiny & Mitchell, 1978). Izgara sistemi üretildikten sonra plan şemasındaki her bir ızgara hücresi doğu batı ve kuzey güney doğrultularında, doğu batı yönünde simetriyi bozmayacak şekilde birleştirilebilir. Grid üzerindeki orta aksta ise +, T veya I şeklinde ızgara hücreleri birleşebilir. Tüm plan şeması tamamlandıktan sonra kapı ve pencere açıklıkları simetriyi bozmayacak şekilde konumlandırılarak giriş aksına portiko eklenir. GRAPE-SGI ile türetilen tüm planlar yukarı anlatılan düzende türetilir. 6. ve 7. deneyde kullandığımız plan şemaları 3x3, 3x4, 3x5, 5x3, 3x7 ızgara düzenlerinden sırasıyla 30, 52, 35, 103 and 100 plan şemasından toplam 320 plan şeması türetilmiştir. Tüm bu türetilen sentetik plan şemalarının siyah ve beyaz piksel miktarları dağılımına bakıldığında dağınık bir düzenin olduğu gözlemlenmiştir. Bunun nedeni 5 farklı grid düzeninden gelen siyah ve beyaz piksel sayılarının değişmesidir. Veri setindeki her bir görselin piksel renk değerlerinin aritmetik ortalaması dağılımına bakıldığında görsellerin tüm diğer deneylere göre daha düzgün bir dağılım oluşturduğu gözlemlenmiştir. Bunun nedeni sentetik olarak üretilen tüm plan verilerinin birbirlerine çok yakın bir dilde üretilmesi ve yine sentetik olarak türetilmesi nedeniyle arka plan ve çizimin net bir şekilde siyah ve beyaz piksellerle ayrışabilmesi olabilir. Palladio'nun gerçek planlarından oluşan veri setinde planların taranması nedeniyle, photoshopta temizlenmesi nedeniyle arka plan ve çizim arasında net bir ayrım %100 yakalanamamaktadır. GRAPE-SGI ile türetilen veri seti gerçek planlardan oluşan veri setinden daha temiz bir sonuç vermiştir. Veri setindeki bu düzen DCGAN'ın üretimlerinin kalitesini arttırdığı gibi kayıp fonksiyon değerlerinin de daha düzgün değişmesini sağlamıştır. Düzgün bir veri seti ile DCGAN eğitimi daha stabil hale gelebilmiştir.



Şekil 5.5: 6. ve 7. deneylerde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

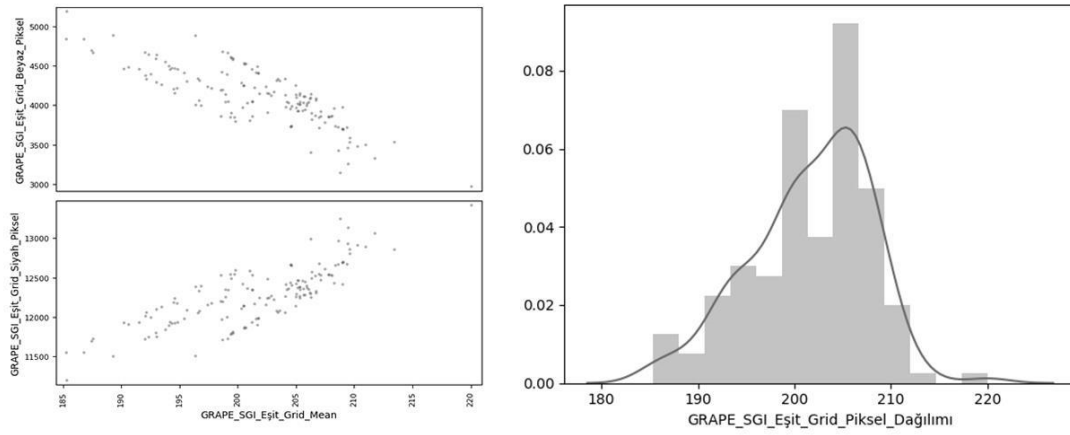
5.6 8. Deney Veriseti Değerlendirilmesi

8. deneyde kullanılan veri seti için 6. ve 7. deneyde kullanılan veri setinin üretilme yöntemi kullanılmıştır. GRAPE-SGI ile Palladyan plan şemaları 6. ve 7. deneyden farklı olarak, her bir ızgara düzeninden eşit sayıda plan şeması kullanılarak veri seti üretilmiştir. 3x3, 3x4, 3x5, 5x3, 3x7'lik her bir ızgara düzeninden 30'ar plan şeması olmak üzere toplamda 150 plan şeması üretilmiştir. 6. ve 7. deneyden farklı olarak bu deney, ızgara düzenlerinin eşit dağılımı ile birlikte DCGAN'nin eğitimi sonrasında farklı ızgara düzenlerine göre üretim çıktıların değişimini takip edebilmek amacıyla kontrollü bir deney olarak tasarlanmıştır.

Bu deneyde de 6. ve 7. deneyde olduğu gibi sentetik olarak üretilen plan şemalarına veri arttırımı yöntemi kullanılmamıştır.

Bu deneyde kullanılan veri setindeki siyah ve beyaz piksel sayısı dağılımlarına bakıldığında, veri setindeki verilerin üretim tekniklerinin aynılığı bakımından homojen olduğu düşünülse de, grafik oldukça heterojen bir yapıya sahiptir. Bu durum doğrudan ızgara düzenlerindeki gridin artması ile siyah piksel ve beyaz piksel sayılarının da değişmesi ile ilişkilidir. Bu nedenle her ne kadar homojen bir veri seti ile üretimin gerçekleştirilmesi istense de bu veri setinde de heterojen bir yapı bulunmaktadır. Bu heterojenliğin önüne geçebilmek veri setindeki her bir ızgara çeşidindeki verinin sayısının daha da fazla artırılması ile mümkün olduğu söylenebilir. Her bir imajın piksel renk değerleri ortalamasına bakıldığında bir çan eğrisi oluşturma da bir tepeli bir dağılımın olduğu gözlemlenebilir. Çubuk grafiklerine bakıldığında veri setinin iki farklı noktada çok yüksek değerler aldığı ve bu durumun veri seti dağılımını bozduğu söylenebilir. 6. ve 7. deneyde kullanılan

veri setleri dağılımları 8. deneydeki veri seti dağılımı ile karşılaştırıldığında 6. ve 7. deney verisetlerinin dağılımının daha düzgün olduğu gözlemlenmiştir. Bunun nedeni 8. deneyde örneklem (her bir veri noktası) sayısının 6. ve 7. deneydeki örneklem sayısından az olması ile ilgili olabilir ve veri setinin sayıca artırılması ile bu problem çözülebilir. 6. ve 7. deneyde de çeşitlilik fazla olmasına rağmen veri sayısının fazla olması nedeniyle varyansının daha düşük olduğu piksel renkleri ortalama değer grafiğinden görülebilir. Üretim çıktılarına bakıldığında 6. ve 7. deney üretimlerinden görsel olarak çok farklı kalitede bir üretim gözlemlenmemiş ve diğer deneylere göre üretim kalitesinin de oldukça yüksek olduğu gözlemlenmiştir.



Şekil 5.6: 8. deneyde kullanılan verisetinin beyaz ve siyah pikselleri dağılım grafiği.

6. ÇÜA ÇIKTILARININ DEĞERLENDİRİLMESİ

Tezin bu bölümünde DCGAN ile üretilen plan şemaları üzerinden DCGAN'ın plan üretimindeki verimliliği değerlendirilecektir. Bu değerlendirme yapılırken DCGAN eğitimi için kullanılan verisetleri ile DCGAN üretim çıktıları karşılaştırılacaktır. Bu karşılaştırmayı Borji (2019) ÇÜA değerlendirme yöntemleri üzerine yaptığı çalışmada iki yöntem ile yapılabileceğini açıklar. Bu yöntemlerden biri nitel (kalitatif) değerlendirme yöntemi diğeri ise nicel (kantitatif) değerlendirme yöntemidir.

ÇÜA değerlendirme yöntemleri ÇÜA algoritmalarının çalışma sisteminin verimini ölçmektedir. Değerlendirme sırasında odak ÇÜA algoritması veriminin ölçümü olmuştur. Bu nedenle ÇÜA ile üretilen verinin veriseti ile benzerlik oranına ya sezgisel yöntemlerle (kalitatif) ya da istatistiksel ve matematiksel yöntemlerle (kantitatif) bakılmaktadır. Bu bölümde kantitatif yöntemlerden *Frechet Inception Distance* DCGAN algoritmasının çalışma verimliliği ölçmek amacıyla kullanılacaktır. Kalitatif yöntemlerden hızlı sahne sınıflandırması, Palladyan grameri ve mekan sentaksı değerlendirmesinde hedeflenen sadece DCGAN algoritmasının çalışma verimliliğini ölçmek değil çıktıların mimari plan konfigürasyonlarının da değerlendirmesidir. ÇÜA ağlarının değerlendirmesinin yapıldığı kantitatif yöntemler istatistiksel ve matematiksel benzerlik sonuçları vermesi bakımından mimari plan değerlendirmesi yapılmamaktadır. Bu durum Christopher Alexander'ın düşüncesi ile paralellik göstermektedir. Alexander'a göre (1964) tasarımdaki formu sembolik olarak üretilebilme potansiyeli olmasına rağmen mimari üretim çıktılarını sembolik yöntemlerle değerlendirmek mümkün değildir. Bu nedenle istatistiksel yöntemlerden yararlanarak ÇÜA verimliliğini ölçen kantitatif yöntemler mimari plan kalitesini değil ÇÜA algoritması çalışma verimliliğini ölçebilmektedir. Bu nedenle ÇÜA üretimlerinin mimari plan şeması değerlendirilmesinin yapılması amacıyla hızlı sahne sınıflandırması, Palladyan gramer kuralları ve mekan sentaksı kullanılmıştır.

6.1 ÇÜA Algoritmasının Verimliliğinin Değerlendirilmesi

Bu bölümde ÇÜA ağlarının kantitatif değerlendirme yöntemi açıklanacaktır. Kantitatif ÇÜA değerlendirme yöntemleri ÇÜA algoritmasının çalışma verimliliğini göstermektedir. Ancak üretilen çıktının bağlamı içerisindeki verimliliği değerlendirilmemektedir. Bu nedenle bu bölümdeki değerlendirme üretilen plan şemalarını değil ÇÜA algoritmasının verimliliğini ölçmektedir.

Borji (2019) *Frechet Inception Distance*'ı (FID) ÇÜA algoritmalarının kantitatif değerlendirme yöntemi içerisinde sınıflandırmıştır. Borji (2019) FID tanımını sentetik ve gerçek verinin dağılımının karşılaştırması olarak yapmıştır, ve FID'in veri üzerindeki gürültüye karşı sağlam bir değerlendirme yöntemi olduğunu söylemiştir. Yine de FID değerlendirme yönteminin problemi verideki özelliklerin dağılımının normal dağılımlı olduğunun kabulüdür (Borji, 2019). Bu kabul her veri seti için geçerli olmadığı gibi bu tez içinde kullanılan veri için de geçerli değildir. Ancak yine de kantitatif ÇÜA değerlendirme yöntemleri arasında ayrıştırabilme, güvenilirlik ve hesaplama verimi ile kullanılan bir yöntemdir (Borji, 2019).

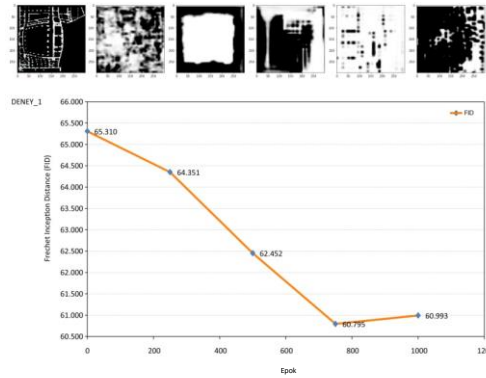
FID değerlendirmesiyle birlikte imaj veriseti ve ÇÜA üretimi imajlardaki özniteliklerin dağılımları hesaplanır (Koehrsen, 2018). FID hem veri seti hem de ÇÜA üretimindeki verilerin dağılımları arasındaki mesafeyi hesaplamaktadır. Öznitelikler verinin içerisindeki değişkenlerin vektörel temsilleridir. Hem veri seti hem de ÇÜA üretimi imaj verilerinin özniteliklerinin vektörel temsilleri arasındaki mesafe FID ile hesaplanabilir. 6.1'de FID hesabı fonksiyonu gösterilmiştir (Borji, 2019).

$$d^2 = ||\mu_1 - \mu_2||^2 + Tr(C_1 + C_2 - 2 * sqrt(C_1 * C_2)) \quad (6.1)$$

d^2 ile mesafenin karesi ifade edilmiştir. C_1 ve C_2 ile kovaryans matrislerini göstermektedir. μ_1 ve μ_2 veri seti ve DCGAN üretimlerinin özniteliklerinin dağılımını göstermektedir. μ_1 ve μ_2 arasındaki fark negatif çıkma ihtimali nedeniyle karesi alınmıştır. Böylece DCGAN üretimi veri ile veriseti içerisindeki verinin öznitelikleri arasındaki vektörel mesafe hesaplanmaktadır. Bu mesafe ne kadar küçük çıkarsa DCGAN'ın de o kadar verimli çalıştığı söylenbilir. Ancak FID skorunun 0 çıkması verisetindeki veri ile DCGAN üretimindeki verinin bire bir aynı olmasıdır. Bu durum aşırı uyum (overfitting) problemini göstermektedir. DCGAN

verisetinin birebir aynısını öğrenmiş ve üretmiştir. Ancak DCGAN'ın kopya değil verisetine benzer üretimler gerçekleştirmesi beklenmektedir. Bu nedenle FID skoru düşük ancak sıfırdan farklı olması gerekir. FID için optimum bir değer verilmemiştir. Ancak eğitim sonucu ile başlangıcı arasında FID skorunda düşme beklenmektedir. Aşağıda her bir deney için ayrı ayrı FID skoru hesaplanmıştır. İlk 5 deney'in çıktısı kalitatif değerlendirme yöntemlerinden anlaşılabileceği gibi değerlendirmek için yeterli çözünürlükte ve kalitede değildir. Ancak yine de 6., 7. ve 8. deneylerdeki FID değişimini gözlemleyebilmek için ilk beş deneyi de FID değerlendirmesi içerisinde incelenmiştir.

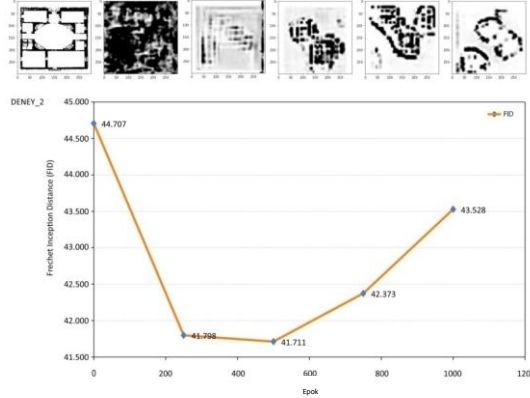
Deney 1'de FID skorunun epoklara göre değişimi şekil 6.1'de görünmektedir. FID değişimi her 200 epokta bir hesaplanmıştır. Grafiğin üstünde öncelikle veri setinden gerçek bir veri ve diğer imajlar ilişkili epoklarda üretilen veriler olarak gösterilmiştir. Eğitim süresince 800. epoka kadar FID skorunun dramatik düşüşlerinin olduğu gözlemlenebilir ancak 800. epoktan sonra FID artma eğilimindedir. FID başlangıç ve bitiş arasındaki değişimi yaklaşık 4.7 birim kadardır. Grafik her ne kadar dramatik düşüşler var gibi görünse de değişim sadece 4.7 birimlik küçük bir değişimdir. DCGAN verimli olmasa da doğru çalışmış ancak üretim çıktıları ve FID skorundan da anlaşılacağı üzere üretimde başarılı olunamamıştır.



Şekil 6.1: 1. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik.

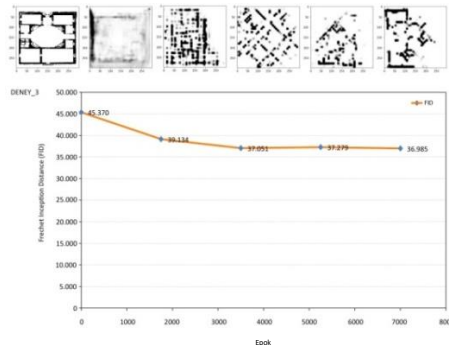
Deney 2 için de FID grafiği 1. deneydeki gibi oluşturulmuştur (Şekil 6.2). 2. deneyde 200. epokta dramatik bir düşüş olmuş ancak sonrasında FID skorunda artış gözlemlenmiştir. Başlangıç ve bitiş arasındaki FID skoru farkı 1.2 birim kadardır. Bu durum 1. deneyden daha da kötü bir sonuç olarak karşımıza çıkmıştır. Ancak yine de

retilen imaj verileri 1. deneyle karřılařtırıldığında daha anlamlı sonular elde edilmiřtir. Bu nedenle 2. deney epok sayısı artırılarak tekrarlanmıřtır.



řekil 6.2: 2. deneyin DCGAN retim ıktılarının Frechet Inception Distance deęerlerinden oluřan grafik.

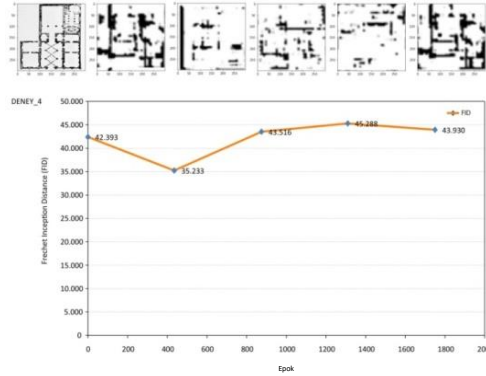
2. deneyde 1000 epok eęitilen algoritma 3. deneyde aynı veriseti ile 7000 epok eęitilmiřtir. Epoklardaki FID skoru deęiřimi dikkatli bir řekilde incelendiğinde FID skorunun 1. ve 2. deneylere gre daha iyi olduęu gzlemlenebilir. FID deęiřimi yaklaşık 9 birim kadardır. retilen grsellere de bakıldığında epok sayısının hem retilen grsellerin kalitesinin arttıęı hem de FID skorunun dřrlebildięi grlmřtir. Yine de hem FID skorundaki dřř deęiřimi hem de retilen grseller DCGAN retimi iin yeterli deęildir. 1., 2. ve 3. deneylerdeki retimlerin ve FID skorlarının nedeni doęrudan 5. blmde aıklanan verisetindeki zniteliklerin daęılımı ile ilgilidir.



řekil 6.3: 3. deneyin DCGAN retim ıktılarının Frechet Inception Distance deęerlerinden oluřan grafik.

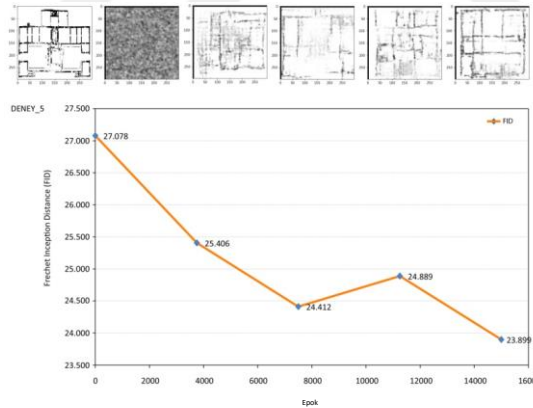
4. deneyde FID skoru deęerlendirmesine bakıldığında eęitim sonucunda FID skorunun bařlangıca gre artmıřtır. Bu nedenle eęitimin doęru gerekleřemedięi sylenebilir. retilen grsellere bakıldığında mode collapse denilen hatanın olduęu 4. Blmde aıklanmıřtır. Hem grsellerdeki hem de FID skorundaki problemin

sebebi ise 5. bölümde açıklanan 4. deneyde kullanılan verisetindeki özneliklerin dağılımı ile ilgili olduğu açıklanmıştır.



Şekil 6.4: 4. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik.

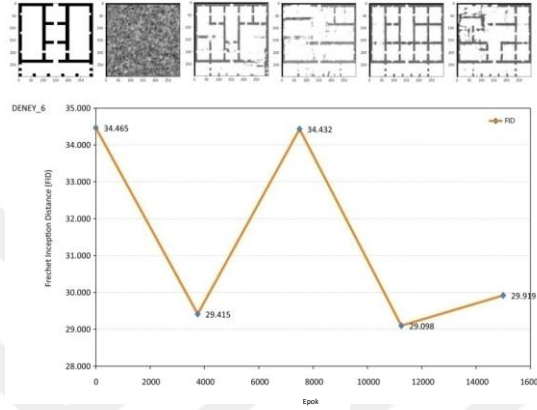
5. deneyde kullanılan verisetinin öznelik dağılımının bundan önceki deneylerden daha düzenli olması nedeniyle DCGAN'ın bu deneydeki üretim çıktılarının daha kaliteli olmuştur. Ancak FID skoruna bakıldığında değişimin yaklaşık 3 birim kadar azaldığı görülmüştür. Toplamda 15000 epok eğitim için 3 birimlik azalma oldukça az bir değişimdir. Bu nedenle DCGAN'ın veriminin bu deneyde de oldukça düşük olduğu söylenebilir.



Şekil 6.5: 5. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik.

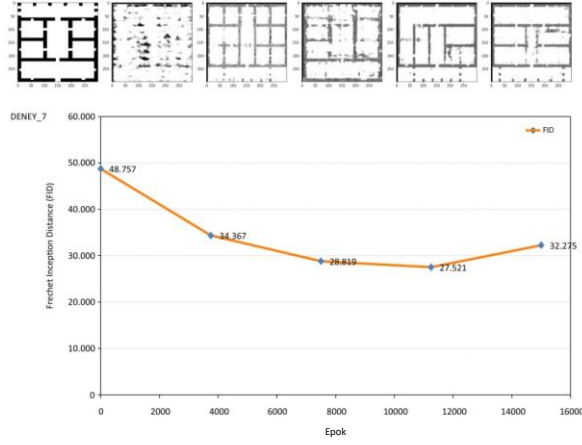
6. deneyde GRAPE-SGI ile üretilen Palladyan plan şemalarından oluşan veriseti eğitim için kullanılmıştır. Diğer deneylerde Palladio'nun orijinal plan şemalarından oluşan verisetleri kullanılmıştır. 6. deneydeki DCGAN üretim çıktıları diğer deneylerle karşılaştırıldığında oldukça kaliteli ve okunabilir sonuçların elde edildiği görülmüştür. Yaklaşık 15000 epok eğitilen algoritmanın üretim sonucunda FID skoru değişiminin yaklaşık 4,5 birimlik olduğu aşağıdaki grafikte görülebilir. Bu durum 15000 epok eğitilen bir algoritma için oldukça düşüktür. Bunun yanında 4.

deneyin kayıp fonksiyon değerlerinde de problem olduğu 4. bölümde açıklanmıştır. Bunun bir nedeni veriseti ile ilgili problem olabilir. Verisetinde 320 Palladyan plan şeması farklı sayıdaki farklı grid düzenlerindeki verilerden oluşmaktadır. Her ne kadar 5. bölümdeki verisetleri değerlendirmesinde diğer deneylerde kullanılan verisetlerinden daha düzgün bir dağılıma sahip olsa da hem kayıp fonksiyon değeri hem de FID skorlarından anlaşılabileceği üzere veriseti ile ilgili bir problemin olabileceği düşünülmüştür.



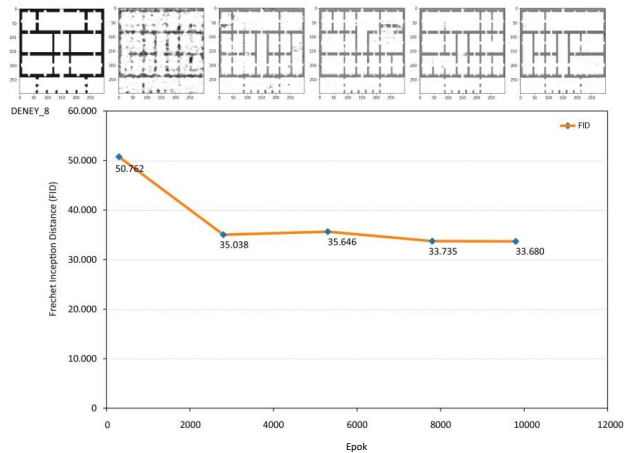
Şekil 6.6: 6. deneyin DCGAN üretim çıktılarından Frechet Inception Distance değerlerinden oluşan grafik.

7. deneyde 6. deney için kullanılan verisetinin aynısı eşit sayıda epok ile eğitilmiştir. Ancak 6. deney sonuçlarına göre verisetinin problemlili olması ihtimaline odaklanılmıştır. Bu tezde kullanılan her verisetinde olduğu gibi 7. deneydeki veriseti de siyah ve beyaz piksel özniteliklerine sahiptir. Yani veriseti oldukça düşük öznitelik sayısına sahiptir. Bu durum algoritmanın pikseller üzerinde siyah ve beyaz olarak kesin ayrımlar yapmasına arada kalan gri bölgelerin ise doğru yorumlanamamasına böylelikle üretim çıktılarında sorunların oluşmasına yol açmaktadır. Bu durumu engellemek için özniteliklere kesin siyah ve beyaz değerler atamak yerine bu değerlere yakın ancak eşit olmayan değerlerin atandığı label smoothing tekniği kullanılmıştır. Etiket yumuşatma tekniği kullanılan 7. deneyin FID değişimine bakıldığında başlangıç ve bitiş epok FID skoru farkının yaklaşık olarak 16,5 birimlik bir fark olduğu okunabilir. Bu değer DCGAN algoritmasının etiket yumuşatma ile birlikte siyah ve beyaz imajlardan oluşan verisetinde verimi oldukça arttırdığını göstermektedir. Bunun yanında 7. deneyde DCGAN ile verimli sonuçlar alındığı söylenebilir. Ancak verisetindeki grid düzenlerinin farklı sayılarda olması verisetinin dağılımını etkileyebileceği düşüncesi ile verisetindeki grid düzenlerinde eş sayıda plan şemalarından oluşan 8. deney kurgulanmıştır.



Şekil 6.7: 7. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik.

8. deneyde kullanılan GRAPE-SGI ile üretilen verisetinde her bir grid düzeninden 30'ar plan şeması olmak üzere toplamda 150 plan şemasından oluşan bir veri seti kullanılmıştır. Kayıp fonksiyonu grafiğine bakıldığında 10000 epok sonrasında çok fazla bir değişim görünmediği için algoritmanın eğitimi durdurulmuştur. Bu eğitim sırasındaki FID skorlarına bakıldığında başlangıç ve bitiş arasındaki farkın tüm deneylerden daha büyük ve yaklaşık 17,1 birimlik bir düşüş farkı olduğu aşağıdaki grafikte görülebilir. Ancak aşağıdaki grafik daha detaylı okunduğunda FID skorunun en dramatik düşüşünün 2000. epokta olduğu görülmüştür. Diğer epoklar plan şemasındaki mekansal düzenlemeden daha çok imajın çözünürlüğünü arttırdığı DCGAN üretimlerinin karşılaştırılması ile okunabilir. Bu nedenle bu deneyde 10000 epok eğitiminin de fazla olduğu söylenebilir.



Şekil 6.8: 8. deneyin DCGAN üretim çıktılarının Frechet Inception Distance değerlerinden oluşan grafik.

DCGAN algoritmasının çıktıları ile birlikte FID skoru değerlendirmesi yapıldığında DCGAN deneylerinden en yüksek verimin 8. deneyde olduğu söylenebilir. Kayıp

fonksiyon değeri, üretim çıktıları, FID skor değerleri diğer deneylerle karşılaştırıldığında 8. deneyde başarılı bir üretim gerçekleştiği görülebilir. 8. deneydeki bu başarı veriseti dağılımının düzgün oluşturulması, 7. deneyde de kullanılan etiket yumuşatma tekniğinin bu deneyde de kullanılması ile ilgilidir.

FID analizinde 6., 7. ve 8. deneyler arasında büyük farkların olduğu FID skoru değerlendirmesi ile ortaya çıkmıştır. Bu nedenle ÇÜA üretimlerinin değerlendirilmesi sırasında hem kalitatif hem kantitatif yöntemlerin bir arada kullanılmasının gerekli olduğu söylenebilir.

Plan şeması üretimi bakımından FID skoru değerlendirmesi ile mimari bir değerlendirme yapmak mümkün değildir. Ancak özellikle 7. ve 8. deneyle birlikte DCGAN’ın verisetindeki öznitelikler dağılımını matematiksel olarak doğru okuduğu ve bu öznitelik dağılımlarına göre üretimlerinin verimli olduğu söylenebilir. Kantitatif değerlendirme yöntemine göre mimari mekansal dizilim açısından olmasa bile DCGAN verisetindeki öznitelik dağılımları üzerinden verisetine olasılıksal olarak yakın plan şemaları üretmiştir. Bu da matematiksel olarak plan şemalarının doğruya yakın üretimler olduğunu gösterebilir.

6.2 Mimari Plan Şeması Üretimlerinin Değerlendirilmesi

Bu bölümde ÇÜA algoritmasının çalışma verimliliği değil, DCGAN’ın üretim çıktısının plan şeması düzeninin değerlendirilmesi yapılması hedeflenmektedir. Değerlendirme sürecinde ÇÜA nitel değerlendirme yöntemi olan “hızlı sahne sınıflandırması”, Palladyan gramer kuralları ve mekan sentaksı kullanılmıştır.

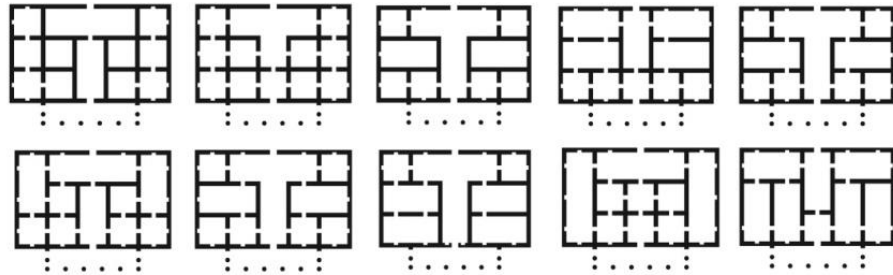
6.2.1 Hızlı Sahne Sınıflandırması (Rapid Scene Categorization)

Nitel yöntemlerde ÇÜA değerlendirmesi insan tarafından yapılmaktadır. Borji (2019) kalitatif yöntemlerin dezavantajını üç başlıkta sıralamıştır. İlki insan görüşü ile ÇÜA üretimlerinin değerlendirilmesi maliyetli ve yanlış olabilmektedir, ikinci olarak gözlemcinin insan olması nedeniyle varyansı yüksek olabileceğinden karşılaştırması gereken örneklem uzayında örneklem sayısının çok fazla olması gerekebilir, üçüncü ve son olarak da ÇÜA’nın eğitimindeki problemler insan gözlemci tarafından doğru gözlemlenemeyebilir. Buna örnek ÇÜA modelinin *aşırı uyum (overfitting)* yapması halinde üretilen veri kalitesi yüksek olsa bile ÇÜA’nın veriyi ezberlemiş olduğu gözden kaçırılabilir. *Overfitting*, *mode collapse* ve *mode drop* problemlerinin insan

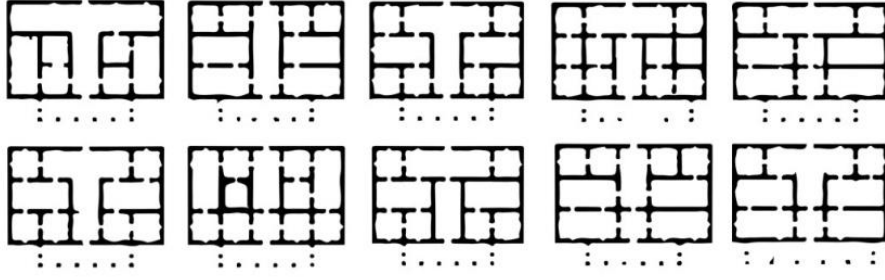
gözlemci tarafından saptanması sırasında hatalar oluşabilir (Borji,2019). Yine de ÇÜA çıktıların değerlendirilmesinde çokça kullanılan sezgisel bir yöntem olarak kalitatif yöntemler hala kullanılmaktadır.

Hızlı sahne sınıflandırması ÇÜA üretimlerinin kalitatif değerlendirilmesinde kullanılan yöntemlerdendir. Bu yöntem ile ÇÜA üretim çıktıların gerçekliği sorgulanmaktadır. Bu sorgulama anket yöntemi ile katılımcılar tarafından yapılır. Hızlı sahne sınıflandırmasında gerçeklik değerlendirmesi biri veri setinden biri de ÇÜA üretiminden olmak üzere iki verinin sorgulanması ile gerçekleşir. Anket katılımcıları iki şıktan birini gerçek olarak değerlendirir. En sonunda gerçek olarak değerlendirilen verilerin kaçının ÇÜA üretimi kaçının veri setinden olduğuna bakılmaktadır. Bu yöntemde doğrudan insanın öznel bakış açısı süreçte bulunduğu için bazı yanılgılar ve ön yargılar nedeniyle tam doğru sonuçlara ulaşılamaz. Ancak yine de hızlı bir şekilde değerlendirmeye değebilecek sonuçları nedeniyle ÇÜA üretimlerinin değerlendirilmesinde kullanılmaktadır.

Hızlı sahne sınıflandırmasını DCGAN ile üretilen Palladyan plan şemaları için bir değerlendirme yöntemi olarak kullanılmıştır. Ancak burada farklı olarak anket katılımcılarına sorulan soru hangisinin gerçek üretim olduğu değil hangi üretimin plan organizasyonunun daha iyi olduğu sorulmuştur. Soru oldukça genel bir soru olmasına rağmen veri seti ile ÇÜA üretimleri arasındaki plan düzenlerinin karşılaştırılarak değerlendirilmesi bakımından önemlidir. Böylece plan düzlemindeki mekansal yerleşimin veri seti ile karşılaştırıldığında nitelikli olup olmadığı gözlemlenebilir. Bu nedenle kalitatif yöntemlerden biri olarak hızlı sahne sınıflandırması kullanılmıştır. Bu değerlendirme yönteminde hem GRAPE-SGI veri setinden hem de DCGAN üretimlerinden 10'ar plan şeması olmak üzere 20 plan şeması kullanılarak anket oluşturulmuştur (Şekil 6.9, Şekil6.10).



Şekil 6.9: GRAPE-SGI üretimi plan şemaları.



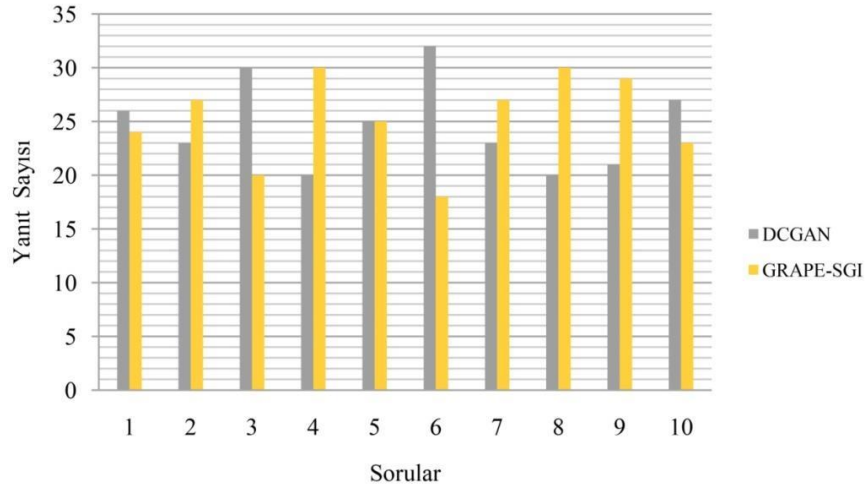
Şekil 6.10 DCGAN üretimi plan şemaları.

Her bir soruda bir tane DCGAN plan şeması, bir tane de GRAPE-SGI plan şeması bulunmaktadır. Anket katılımcısı iki plan şeması arasında, hangisinin GRAPE-SGI veya DCGAN'e ait olduğunu bilmeden, plan düzeni daha iyi olanı seçmektedir. Aşağıda *surveymonkey* ile oluşturulmuş anketin arayüzü görülmektedir.

Şekil 6.11 Hızlı sahne sınıflandırması için anket arayüzü.

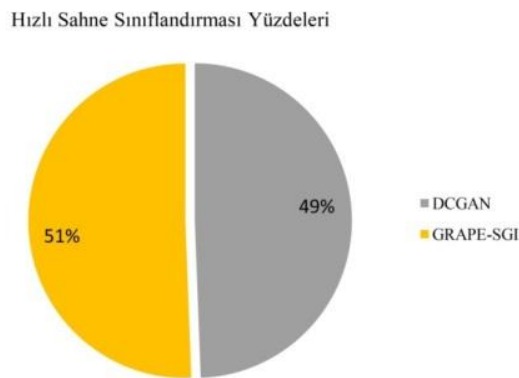
Surveymonkey üzerinden gerçekleştirilen bu anket 50 katılımcı ile gerçekleştirilmiştir. Sonuçta toplamda 500 farklı yanıt ile hızlı sahne sınıflandırması yapılmıştır. Aşağıdaki grafikte her bir soru için hem DCGAN üretimleri hem de GRAPE-SGI veri seti için seçilen yanıtlar ve sayıları gösterilmiştir. Grafik değerlendirildiğinde 10 soru içinde 4 tanesinin DCGAN yanıtları GRAPE-SGI'dan daha fazladır. Geri kalan 5 yanıtta GRAPE-SGI öne çıkmaktadır. 1 soruda da eşit olarak yanıtlandığı gözlemlenmiştir. Her ne kadar GRAPE-SGI DCGAN'den daha fazla yanıt almış gibi değerlendirilebilse de yanıt sayılarına bakıldığında, çok büyük farklar olmadığı gözlemlenmiştir. GRAPE-SGI'nın DCGAN'den en fazla farkla öne

çıkıtığı soruda GRAPE-SGI 30 yanıt DCGAN ise 20 yanıt almıştır. Diğerlerinde ise yanıt farkı düşmektedir. Bu durum aşağıdaki şekil 6.12’de açıkça okunabilmektedir. Gri ile belirtilen DCGAN ile sarı renkte belirtilen GRAPE-SGI arasında yanıt dağılımları bakımında çok büyük açıklıklar bulunmamaktadır. En büyük fark DCGAN’ın önde olduğu 6.soruda ve fark 14 yanıttır.



Şekil 6.12 Hızlı sahne sınıflandırması anket yanıt sayıları.

Hızlı sahne sınıflandırmasında toplam yanıtların yüzdesine bakıldığında %51 ile GRAPE-SGI %49 ile DCGAN yanıtlarının verildiği okunabilmektedir. 500 adet yanıt toplanarak yapılan bu çalışmada GRAPE-SGI ve DCGAN arasında plan düzenlemesi bakımından, hızlı sahne sınıflandırması analizine göre, sadece %2’lik bir fark olduğu söylenebilir.



Şekil 6.13 Hızlı sahne sınıflandırmasında yanıtların yüzdeleri.

DCGAN’ın Paladyan plan şeması üretimleri, kalitatif yöntemlerden hızlı sahne sınıflandırması ile değerlendirildiğinde veri setine yakın üretimler gerçekleştirebildiği söylenebilir. Ancak yine de kalitatif yöntem olması ve

değerlendiricinin insan olması nedeniyle oluşabilecek ön yargı ve yanlıgılar bu sonuçların içerisinde bulunduđu unutulmamalıdır.

6.2.2 Palladyan Gramer Kurallarına göre Sınıflandırma

Hızlı sahne sınıflandırma yönteminde olduđu gibi bu bölümde de ilk 5 deney için değerlendirme yapılabilecek kalitede bir çıktı olmaması nedeniyle son üç deney (6., 7., ve 8.) deneylerdeki çıktıları odaklanılmıştır. 6. ve 7. deneylerin veriseti ve üretim çıktı performansları benzer olması nedeniyle bu bölümde 7. ve 8. deneyler değerlendirmeye dahil edilecektir. 7. ve 8. deney çıktıları incelendiğinde çözünürlük ve üretim kaliteleri bakımından çok fazla fark olmadığı görülmüştür. 7. deneydeki plan şemaları 5x3 ızgara düzenine sahipken; 8. deneydeki üretim çıktıları 7x3'lük ızgara düzenine sahiptir. 7.deneydeki çıktının 5x3 ızgara düzenine 8. deneyin de 7x3'lük ızgara düzenine odaklanmasının bir nedeni piksel dağılımında bu iki verinin bulundukları veri setindeki dağılımında bulunma olasılıklarının en yüksek veri tipi olması ile ilgili olabilir. Sadece siyah ve beyaz piksellere bakılarak üretimin gerçekleşmesi nedeniyle siyah ve beyaz piksel sayısı dağılımı üretim çıktıları doğrudan yönlendirebilmektedir. Bu konu veri seti değerlendirmesinin yapıldığı 5. bölümde detaylı olarak anlatılmıştır.

Palladyan gramer kurallarına göre değerlendirme yöntemi ÇÜA ağırları değerlendirme yöntemleri içerisinde bulunmamaktadır. ÇÜA ağırları değerlendirme yöntemleri doğrudan ÇÜA üretim verimliliğini üretim çıktıları değerlendirerek gerçekleştirir. Ancak bu tezde amaçlanan ÇÜA üretim verimliliği değerlendirmenin yanında ÇÜA algoritmasının mimari plan üretim çıktıları mimari plan düzeni bakımından değerlendirmesini de içermektedir. Bu nedenle veri seti ile üretim çıktıları değerlendirilmesinde, veri setinin üretilme kuralları olan Palladyan gramer kurallarının üretim çıktıları olup olmadığı araştırılmıştır.

7. ve 8. deneylerde kullanılan veri setleri Stiny & Gips'in (1978) açıkladıkları Palladyan gramer kuralları kullanılarak türetilen Palladyan plan şemalarından oluşan bir veri setidir. Palladyan gramer kurallarına göre plan şemaları GRAPE-SGI adındaki çevrim içi biçim grameri yorumlayıcısı (Shape Grammar Interpreter / SGI) yazılım ile üretilmiştir.

Biçim grameri yorumlayıcısındaki Palladyan plan şeması üretimindeki kurallar sırasıyla aşağıda belirtilmiştir;

-3x3, 3x4, 3x5, 5x3, 3x7 düzeninde gridlerin üretilmesi

-Üretilen grid üzerinden duvarların üretilmesi

-Grid üzerinde birleştirilecek birimlere karar verilmesi (Grid üzerindeki birimler Palladyan gramer kuralına göre T, I ve + biçimlerinde birleşebilmektedirler.)

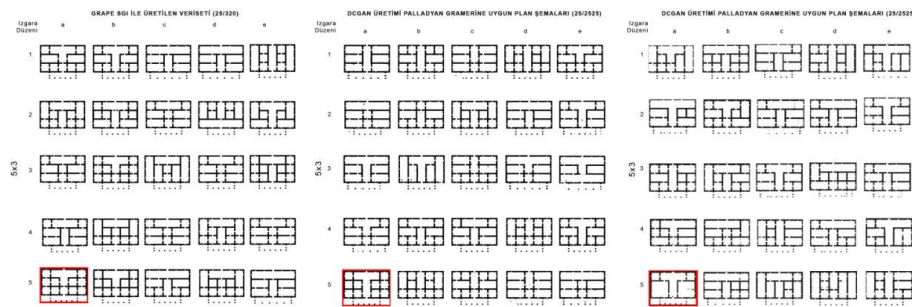
-Sundurmanın eklenmesi

-Portikonun eklenmesi

-Doğu-Batı, Kuzey-Güney akslarındaki kapı ve pencere açıklıklarının Kuzey- Güney aksına bilateral simetri kuralına uygun şekilde oluşturulması.

Tüm bu kurallara uygun olarak 7. ve 8. deneydeki veri setleri oluşturulmuştur. 7. deneylerde 320 plan şeması grid düzenlerinde oluşturulan plan şemaları eşit sayıda olmadan, 9. deneyde ise her bir grid düzeninden 30 adet olmak üzere 150 plan şeması üretilmiştir.

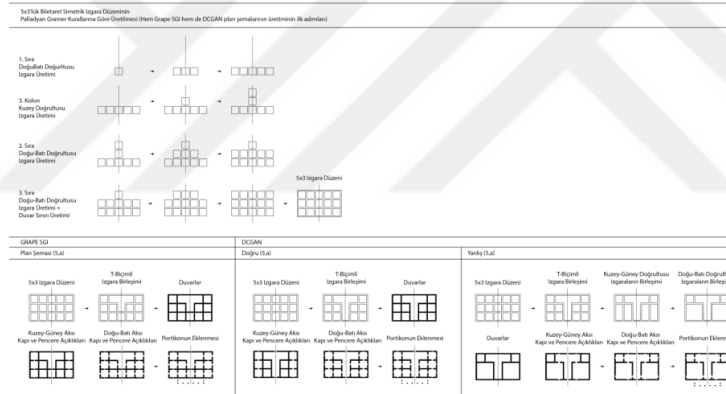
Şekil 6.14'da 7. deney üretimlerinden solda GRAPE-SGI ile üretilen veri setinden 25 plan şeması, ortada DCGAN'ın Palladyan grameri ile uyumlu olan 25 plan şeması, sağda ise DCGAN'üretimlerinden Palladyan grameri ile uyumlu olmayan 25 plan şeması gösterilmiştir. 7. deneyde DCGAN'ın 2525 plan şeması üretimi içinde 115 tanesinin Palladyan grameri kurallarına uygun olduğu bulunmuştur. Bu durum DCGAN'ın 7. deneydeki üretimlerinin %4,5'lik kısmının veri setindeki gramer kurallarına uygun üretim gerçekleştirdiğini göstermektedir.



Şekil 6.14 7. deneydeki GRAPE-SGI ile üretilen veri setinden 25, DCGAN ile üretilen veriden 25 DCGAN (doğru), 25 DCGAN (yanlış) veri.

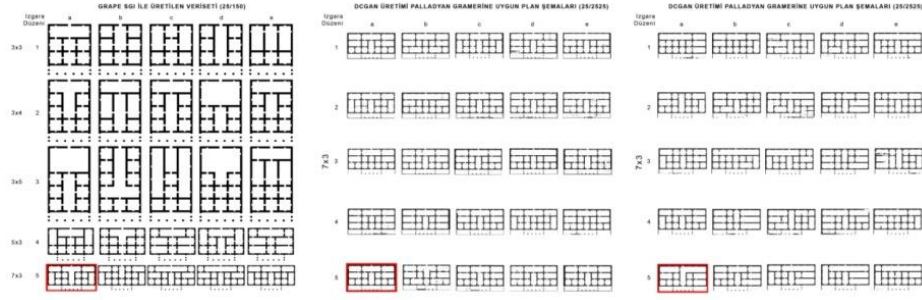
Şekil 6.15'te Şekil 6.14'teki plan şemalarından kırmızı ile çerçevelenmiş plan şemalarının üretilme kuralları deşifre edilmiştir. DCGAN(doğru) ve DCGAN(yanlış) olarak adlandırılan plan şemalarının gramer kuralları deşifre edilerek GRAPE-SGI'nın Palladyan gramer kurallarına uygun olup olmadığı araştırılmıştır. Şekil

6.15'te 5x3'lük gridin oluşturulmasında kullanılan ilk 4 kuralın GRAPE-SGI, DCGAN(doğru) ve DCGAN(yanlış) plan şemaları için ortak olduğu gözlemlenmiştir. GRAPE-SGI plan şemasının gramer kuralları deşifre edildiğinde yukarıdaki adımları verilen Palladyan gramer kurallarına uygun olduğu okunabilir. DCGAN(doğru) olarak tanımlanan plan şemasının gramer kuralları deşifre edildiğinde GRAPE-SGI ile üretilen plan şemasın gramer kurallarının aynı şekilde kullanıldığı, sadece T biçimli grid birimlerinin birleştirilmesinde bir birim daha fazla birleşim yapıldığı gözlemlenmiştir. DCGAN'ın bu üretiminin Palladyan gramer kuralına uygun olması nedeniyle DCGAN(doğru) olarak etiketlenmiştir. DCGAN(yanlış) üretiminin gramer kuralları deşifre edildiğinde Palladyan gramerinde farklı olarak en son adımda kuzey güney doğrultusunda birleştirilen sol ve sağ taraftaki birimler yeniden birleştirilerek kare biçimli mekanlara ulaşılmıştır. Bu kural Palladyan gramer kuralları içinde bulunmadığı için DCGAN'ın bu üretimi DCGAN(yanlış) olarak etiketlenmiştir.



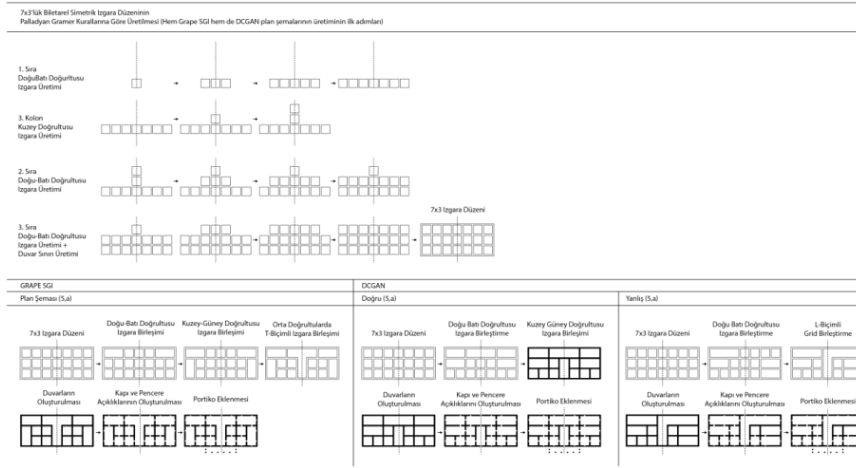
Şekil 6.15 7. deney çıktılarının Palladyan gramer kurallarına ile sol altta GRAPE-SGI kullanılarak üretilen verinin (5,a), alt sıra ortada DCGAN (5, a) Palladyan kuralına göre doğru üretilmiş alt sıra sağda(5,a) yanlış üretilmiş veri gramer kuralları.

Şekil 6.16'de 8. deney üretimlerinden solda GRAPE-SGI ile üretilen veri setinden 25 plan şeması, ortada DCGAN'ın Palladyan grameri ile uyumlu olan 25 plan şeması, sağda ise DCGAN'üretimlerinden Palladyan grameri ile uyumlu olmayan 25 plan şeması gösterilmiştir. 8. deneyde DCGAN'ın 2525 plan şeması üretimi içinde 63 tanesinin Palladyan grameri kurallarına uygun olduğu bulunmuştur. Bu durum DCGAN'ın 8. deneydeki üretimlerinin %2,5'lik kısmının veri setindeki gramer kurallarına uygun üretim gerçekleştirdiğini göstermektedir.



Şekil 6.16 8. deneydeki GRAPE-SGI ile üretilen veri setinden 25, DCGAN ile üretilen veriden 25 DCGAN (doğru), 25 DCGAN (yanlış) veri.

Şekil 6.17’de Şekil 6.16’daki plan şemalarından kırmızı ile çerçevelenmiş plan şemalarının üretilme kuralları deşifre edilmiştir. DCGAN(doğru) ve DCGAN(yanlış) olarak adlandırılan plan şemalarının gramer kuralları deşifre edilerek GRAPE-SGI’nın Palladyan gramer kurallarına uygun olup olmadığı araştırılmıştır. Şekil 6.17’de 7x3’lük gridin oluşturulmasında kullanılan ilk 4 kuralın her GRAPE-SGI, DCGAN(doğru) ve DCGAN(yanlış) plan şemaları için ortak olduğu gözlemlenmiştir. GRAPE-SGI plan şemasının gramer kuralları deşifre edildiğinde yukarıdaki adımları verilen Palladyan gramer kurallarına uygun olduğu okunabilir. 7. deneyde yapıldığı gibi 8.deneyin değerlendirilmesinde Palladyan gramerine uygun olan üretimler DCGAN(doğru) ve yanlış üretimler DCGAN(yanlış) olarak etiketlenmiştir. Şekil 6.17’de GRAPE-SGI üretimi plan şemasının öncelikle doğu batı doğrultusunda, sonrasında kuzey-güney doğrultusunda ve sonrasında T biçimli grid birimlerinin birleştirilmesi gerçekleştirilmiştir. DCGAN(doğru) plan şemasında ise kurallar sırasıyla doğu batı doğrultusunda kuzey güney doğrultusu bilateral simetriyi bozmayacak şekilde birimlerin birleştirilmesi sonrasında kuzey güney doğrultusunda orta aksta I biçimli birleşimin gerçekleştiği gözlemlenmiştir. DCGAN’ın bu plan şemasının deşifresi ile elde edilen kuralın Palladyan gramer kuralı ile çelişmemesi nedeniyle DCGAN(doğru) olarak etiketlenmiştir. DCGAN(yanlış) üretiminin kural deşifresinde sırasıyla; doğu batı doğrultusunda grid kuzey güney doğrultusu bilateral simetri kuralına uymadan birleştirildiği gözlemlenmiştir. Sonrasında ise yine Palladyan gramer kuralında bulunmayan ters L biçimli ve yine kuzey güney doğrultusu bilateral simetriyi bozan birim birleşimi gerçekleşmiştir. DCGAN’ın bu üretimi Palladyan gramer kurallarına uygun olmadığı için DCGAN(yanlış) olarak etiketlenmiştir. 7. deneyde olduğu gibi 8. deneyde de DCGAN(doğru) ve DCGAN(yanlış) plan şemaları kural deşifreleri ile elde edilmiş ve toplam 63 adet DCGAN(doğru) plan şeması üretildiği gözlemlenmiştir.



Şekil 6.17 8. deney çıktılarının Palladyan gramer kurallarına ile sol altta GRAPE-SGI kullanılarak üretilen verinin (5,a), alt sıra ortada DCGAN (5, a) Palladyan kuralına göre doğru üretilmiş alt sıra sağda(5,a) yanlış üretilmiş veri gramer kuralları.

Palladyan gramer kurallarına göre veri seti ile DCGAN üretimlerinin kural deşifresi kullanılarak yapılan bu ÇÜA değerlendirme yönteminde 7. deneyde 2525 plan şeması içinde 115 doğru plan şeması ile %4,5, 8. deneyde yine 2525 plan şeması içinde 63 doğru plan şeması ile %2,5 Palladyan grameri doğruluk oranı ile üretimlerin gerçekleştiği gözlemlenmiştir. Bu durum ÇÜA algoritmalarından DCGAN'ın biçimlerin mekansal düzen kurallarını öğrenmeye yönelmediğini göstermektedir. Ancak doğru üretimlerin olmasının nedeni üretimlerindeki siyah ve beyaz piksellerin plan şemalarında bulunma olasılıklarının dağılımı içerisinde biçim kurallarına uygun olan dağılımların da olması ile açıklanabilir. Olasılıktaki normal dağılım düşünüldüğünde üretimlerin çoğunlukla kurala uymayan mekansal düzenlemelerden oluştuğu azınlığın ise kurala uyduğu durumlardan oluştuğu söylenebilir.

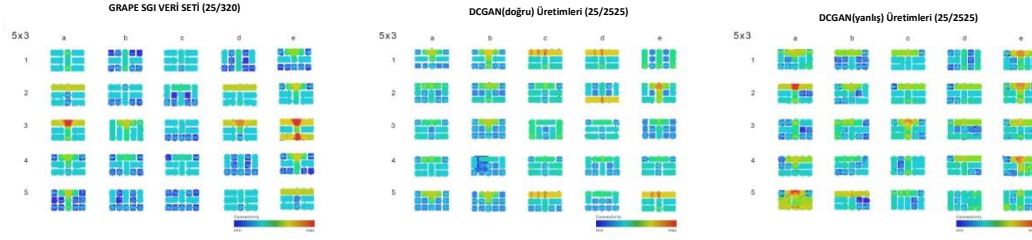
6.2.3 Mekân Sentaksı Yöntemi ile Sınıflandırma

ÇÜA mimari plan şeması üretimlerinin değerlendirilmesi amacıyla, veri seti ile ÇÜA çıktıların mekansal dizilimlerine bakılarak bir karşılaştırma yapılabilir. Böylece mekansal dizilime uygun üretim yapılmıyorsa da değerlendirilebilir. FID ile algoritmanın çalışma verimliliğinin değerlendirilmesinin yanında mimari bir değerlendirme yapmak amacıyla, hızlı sahne sınıflandırması ve palladyan gramer kuralları ile değerlendirmeler yapılmıştır. Mekân sentaksı yöntemi de üretimlerin mimari olarak değerlendirilmesinde üçüncü yöntem olarak önerilmiştir.

Bu çalışmada mekan sentaksı yazılımı *DepthMap* kullanılarak plan şemasını oluşturan mekanların entegrasyon (*connectivity*) değerleri analiz edilmiştir. Bu analiz ile birlikte hangi mekanın plan düzenine daha fazla entegre hangi mekanın daha derinde ve mahrem alanlarda bulunduğu analizi yapılabilir. Bunun yanında entegrasyon değerleri incelendiğinde mekanlara erişilebilirlik analizi de yapılmaktadır. Bu analiz sonucunda mekanlar içerisinde sıcaklık haritaları oluşturulmaktadır. Böylece plan şemalarındaki mekanlar entegrasyon değerlerine göre değerlendirilebilir.

Bu bölümde 7. ve 8. deneyler için veri seti, DCGAN(doğru) ve DCGAN(yanlış) çıktılarından 25'er plan şeması kendi içlerinde entegrasyon değerlerine göre karşılaştırılmıştır.

Şekil 6.18'de 7. deneydeki entegrasyon analiz sonuçları görülmektedir. Sıcaklık haritasına göre kırmızı alanlar plan düzeninde en fazla entegre olan alanları göstermektedir. Renk soğukluğu arttıkça mekan entegrasyonu düşük mekanlara ulaşmaktayız. Bu analiz ile GRAPE-SGI ile üretilen veri setinde mekanda en fazla entegre alanların kuzey güney doğrultusundaki bilateral simetri aksında yerleştiği gözlemlenmektedir. Bu durumun en büyük sebebi ise mekanları bağlayan sirkülasyon açıklıklarının bu alanlarda birleşmesidir. Entegrasyonu en düşük mekanlarınsa plan düzleminin dış çeperine yakın olduğu gözlemlenmiştir. Palladyan gramer kurallarına göre sınıflandırılan DCGAN çıktılarından DCGAN(doğru) olarak etiketlenmiş plan şemalarına bakıldığında mekandaki entegrasyon değerlerinin veri setine benzer bir hiyerarşi oluşturduğu gözlemlenebilir. Orta akstaki T ya da I biçimli mekanların entegrasyon değerlerinin yüksek olduğunu yapının çeperlerine ve köşelerine doğru bu entegrasyon değeri düştüğü görülmüştür. Mekan sentaksı bakımından veri setindeki değerler ile benzer bir hiyerarşi içindedir. DCGAN(yanlış) olarak etiketlenen çıktılar değerlendirildiğinde Palladyan gramer kuralına uymasa dahi bu plan şemalarında da mekansal entegrasyon değerlerinin orta akslarda ve bu aksa birleşen doğu-batı yönelimindeki uzunlamasına mekanlarda yüksek çıkmıştır. Bu durum Palladyan gramer kuralına uygun bir mekan üretilmese bile mekanlar arası sirkülasyon açıklıklarının veri setine benzer şekilde oluşturulduğu ve orta akstaki mekanlara bağlantının DCGAN(yanlış) üretimlerinde de diğer mekanlara göre fazla olduğu söylenebilir.



Şekil 6.18: 7. deneylerin DCGAN üretimlerinin mekan sentaksı yöntemi ile değerlendirilmesi.

8. deneydeki GRAPE-SGI, DCGAN(doğru), DCGAN(yanlış) plan şemalarının her birinden 25'er planın mekan sentaks değerlerinin karşılaştırıldığı görsel şekil 6.19'de verilmiştir. 7. deneydeki 5x3'lük grid düzenlerinden üretilen plan şemalarında olduğu gibi 8. deneyde de 7x3'lük grid düzenlerinin mekan sentaksı değerleri veri seti ile paralellik göstermektedir. Mekan sentaksı değerlerine göre veri setindeki plan düzlemine en fazla entegrasyonu olan mekanın çoğunlukla plan şemasının en üstünde doğu batı aksında uzandığı daha sonra da bilateral simetri aksı olan kuzey güney doğrultusundaki orta aksın entegrasyon değerinin yüksek olduğu gözlemlenmiştir. Entegrasyon değeri düşük olan mekanlar plan düzleminin dış çeperine yakın ve köşelerde yerleşmiştir. 7. ve 8. deneylerde kullanılan plan düzenlerinin mekan sentaksı değerleri çok fazla değişiklik göstermemiştir. Bunun nedeni gridal bir düzenden yola çıkarak aynı kurallarla üretilen benzer plan şemalarının üretilmesidir. Grid sayısı artsa dahi portiko yönelimindeki kuzey güney doğrultusundaki orta aksın entegrasyon değeri diğer mekanlarla kurduğu sirkülasyonun fazla olması sebebiyle her zaman fazla olmuştur. Tüm mekansal sirkülasyon orta akstan çepere doğru bir hiyerarşik düzen oluşturur. Veri seti 7. ve 8. deneylerde mekan sentaksı değerleri bakımından değişiklik göstermese de grid düzeninin değişiminin DCGAN üretimine etkisi analiz edilmek istenmiştir. DCGAN(doğru) olarak etiketlenen plan şemalarında da veri setinde olduğu gibi plan şemasının üst kısmında en yüksek entegrasyon değeri daha sonra da orta aksın en yüksek entegrasyon değerini karşıladığı gözlemlenebilir. Bunun yanında DCGAN(yanlış) olarak etiketlenen Palladyan gramer kurallarına uygun olmayan DCGAN üretimlerinin de içerisinde veri setinin mekan sentaksı değerleri ile paralellik gösteren üretimler olduğu okunabilir.



Şekil 6.19: 8. deneyin DCGAN üretimlerinin mekan sentaksı yöntemi ile değerlendirilmesi.

Her iki deneyde de en önemli değerlendirme sonucu DCGAN(yanlış) olarak etiketlenen plan şemalarının çoğunlukla veri seti ile benzer mekan sentaks entegrasyon değerlerine sahip olduğu durumu olmuştur. Palladyan gramer kuralına göre yapılan sınıflandırmada yanlış olarak etiketlenen plan şemalarının çoğunlukla mekan sentaks değerlerinin yakınlık göstermesi DCGAN(doğru) ve DCGAN(yanlış) etiketlendirmesinin plan şemaları düzenlerinin değerlendirilmesinde yeterli olamayacağını göstermiştir. DCGAN(doğru) etiketli plan şemalarının hepsi veri setindeki mekan dizilimine uymaktadır. Ancak DCGAN(yanlış) etiketli plan şemaları ise mekan dizilimi bakımından veri setinden tamamen farklı üretimler içermemekte hatta çoğunlukla benzer dizilimi gösteren plan şemalarının üretildiği söylenebilir. Grid sayısının fazla veya az olması da DCGAN üretiminde, veri setindeki mekanlar arası sirkülasyon kuralı değişmediğinden, mekan sentaks değerleri de hem 7. deneyde hem de 8. deneyde birbirine paralel sonuçlar vermiştir.

Palladyan grameri ile değerlendirme sonucunda elde edilen DCGAN(doğru) ve DCGAN(yanlış) etiketleri siyah ve beyaz piksel dağılımlarının alt kümelerinin DCGAN(doğru) ve DCGAN(yanlış) etiketlerini içermesi ile ilişkilidir. Ancak bu siyah ve beyaz piksel dağılımlarının algoritma tarafından doğru genellenmesi sonucunda DCGAN mekan sentaksı değerlendirmesinde daha yüksek bir doğruluk ile üretim yaptığı söylenebilir. Bunun nedeni de ÇÜA algoritmasının mimari mekanın düzenini değil piksellerin dağılımını öğreniyor olmasıdır.

ÇÜA değerlendirmelerine bakıldığında bu alandaki problemin, ÇÜA'nın mimari plan şeması üretmesi halinde üretimlerin değerlendirileceği uygun bir değerlendirme yönteminin bulunmuyor olmasıdır. FID skoru ile değerlendirme ÇÜA çalışma verimliliğini ölçmektedir. Hızlı sahne sınıflandırması, Palladyan gramer kuralları ile değerlendirme ve mekan sentaksına göre değerlendirme her ne kadar plan şeması ile ilgili bir değerlendirme sunuyor olsa bile değerlendirme sonuçlarının güvenilirliği tartışmalı olabilir. Bu nedenle literatürde ÇÜA ile üretilen plan şemalarının değerlendirme yöntemlerine dair bir çalışmanın bulunmadığı gözlemlenmiş ve bu

alanın potansiyel bir araştırma alanı olduđu görölmüştür. Tez kapsamında kullanılan ve mimari plan şeması değerdendirmeye yönelik yöntemler geliştirilerek güvenilirlikleri test edilmeli ya da yeni yöntemler geliştirilmelidir.



7. SONUÇLAR

Bu tezde derin öğrenme algoritmaları içerisinde sınıflandırılan üretken çekişmeli algoritmalarının plan üretimindeki uygulanış yönteminin açıklanması, ve sonuç üretimlerinin değerlendirmesi hedeflenmiştir.

Mimari plan şeması üretimi probleminin ÇÜA ile araştırılmasında Palladyan planlarından ve plan şemalarından oluşturulan verisetleri ile 8 algoritma eğitim deneyi yapılmış ve sonuçları değerlendirilmiştir.

Aşağıdaki tabloda deneyler yapılış sırasında göre sıralanmıştır. Her bir deneyde kullanılan veriseti tipi, ÇÜA modeli, veriseti boyutu, en iyileyici seçimi, etiket yumuşatma (label smoothing) tekniğinin kullanılıp kullanılmadığı, kullanılan öğrenme oranı, kullanılan yığın (batch) büyüklüğü, veri arttırımı için kullanılan yöntemler ve girdi ile çıktı boyutları verilmiştir. Bu değişkenler içerisinde deneyler arasında değişiklik gösteren değişkenler; veriseti, veriseti boyutu, etiket yumuşatma tekniği kullanımı, epok sayısı ve veri arttırımı yöntemleri olmuştur. Böylece bu değişkenlerin DCGAN üretimine etkisi gözlemlenebilmiştir.

Çizelge 7.1: DCGAN ile yapılan deneylerde kullanılan değişkenler.

Veriseti	GAN modeli	Veri Seti Boyutu	En İyileyici	Etiket Yumuşatma	Öğrenme Oranı	Epok	Küme Boyutu	Veri Arttırımı	Girdi /Çıktı Boyutu
<i>Orijinal Palladyan Plan Şeması / 2D</i>	DCGAN	1000	ADAM	Yok	0.0002	1000	32	3xDöndürme, 2xSimetri, 2xSiyah&Beyaz	128X128/128X128
<i>Orijinal Palladyan Plan Şeması / 2D</i>	DCGAN	750	ADAM	Yok	0.0002	1000	32	3xDöndürme, 2xSimetri,	128X128/128X128
<i>Orijinal Palladyan Plan Şeması / 2D</i>	DCGAN	750	ADAM	Yok	0.0002	7000	32	3xDöndürme, 2xSimetri,	128X128/128X128

Çizelge 7.1 (devam) : DCGAN ile yapılan deneylerde kullanılan değişkenler.

<i>SGI Üretimi Palladyan Plan Şeması</i>	DCGAN	2397	ADAM	Yok	0.0002	1750	32	Orijinal Palladyan Plan Şeması, 2xSimetri	128X128/ 128X128
<i>Orijinal Palladyan Plan Şeması 2D</i>	DCGAN	125	ADAM	Yok	0.0002	15000	32	Yok	128X128/ 128X128
<i>SGI Üretimi Palladyan Plan Şeması 2D</i>	DCGAN	325	ADAM	Yok	0.0002	15000	32	Yok	128X128/ 128X128
<i>SGI Üretimi Palladyan Plan Şeması 2D</i>	DCGAN	325	ADAM	gerçek imaj etiketi, 0.1; gerçek olmayan imaj etiketi, 0.9	0.0002	10000	32	Yok	128X128/ 128X128
<i>SGI Üretimi Palladyan Plan Şeması 2D</i>	DCGAN	150	ADAM	gerçek imaj etiketi, 0.1; gerçek olmayan imaj etiketi, 0.9	0.0002	15000	32	Yok	128X128/ 128X128

7.1 Bulgular

- İlk 4 deneyde elde edilen sonuçlara göre veri arttırımında kullanılan farklı yöntemler ÇÜA çıktılarının kalitesini değiştirememiştir.
- 5. deney hiçbir veri arttırımı kullanılmadan gerçekleşmiş ve okunaklı üretimler gerçekleştirilmiştir.
- İlk 5 deney sonucunda ÇÜA ile mimari plan şeması üretiminde veri arttırımı yönteminin, algoritma eğitimi için dezavantajlı bir yöntem olduğu söylenebilir.
- İlk 5 deneye göre, verisetinin boyutunun küçük olması eğitim verimini negatif etkilememiş aksine algoritma üretim çıktıları iyileşmiştir.
- 6. deneyde kullanılan sentetik Palladyan plan şemaları ile eğitim çıktılarındaki çözünürlük düzelmiştir. 6. ve ilk 5 deneyde kullanılan verisetleri karşılaştırıldığında, ilk 5 deneyin verisetinin okunaklılığında problemler olduğu, arka plan ve biçim arasında belirsizliklerin olduğu, görülmüştür. 6. deneyde kullanılan verisetinde ise çizim ile arka plan arasında

herhangi bir belirsizlik yoktur ve veriseti okunaklılık bakımından oldukça temizdir. Bu da eğitim sonuçlarını olumlu etkilemiştir.

- İlk 6 deneyin verisetleri karşılaştırıldığında, verisetleri içerisinde bulunan siyah ve beyaz piksel öznitelik dağılımlarının 5. ve 6. deneylerde düzenli bir dağılım oluşturduğu görülmüştür. Bu durum da eğitimi olumlu etkilemiştir.
- İlk 6 deneyden 5. ve 6. deneylerde görsel olarak başarımlar elde edilse de algoritma hata değerleri yüksek çıkmıştır. Bu da algoritmanın çalışma sürecinde bir hataya işaret etmektedir.
- 7. deneyde ilk 6 deneyden farklı olarak etiket yumuşatma tekniği kullanılmış ve etiket yumuşatma tekniği hem görsel üretim kalitesini arttırmış hem de algoritma hata değerlerini makul değerlere indirebilmiştir.
- 8. deneyde 7. deneyden farklı olarak veriseti içerisindeki farklı grid düzenlerinden oluşan plan şemaları eşit sayıda oluşturularak algoritma eğitilmiştir. Bu durum da algoritma eğitimi olumlu yönde etkilemiştir.

Algoritmanın eğitimlerde başarılı olduğu 5., 6., 7. ve 8. deneyler değerlendirildiğinde, ÇÜA'nın mimari plan şemasını görsel olarak üretebildiği söylenebilir. Bunu FID skoru ile yapılan değerlendirme de kanıtlamaktadır. Ancak FID skoru plan şeması düzenindeki mekansal organizasyon ile ilgili bir bilgi vermemiştir. Bu nedenle hızlı sahne sınıflandırması, palladyan gramer kuralları ve mekan sentaks değerlerine göre bir değerlendirme daha yapılmıştır. Buradaki önemli sonuçlardan biri ÇÜA ile üretilen mimari plan şeması değerlendirme yöntemlerindeki yetersizlik olabilir. ÇÜA üretimlerinin değerlendirilebilmesi amacıyla yeni yöntemler önerilmesi açık bir araştırma alanı olarak bulunmaktadır.

ÇÜA eğitimi çıktıların hızlı sahne sınıflandırması, palladyan gramer kuralları ve mekan sentaks değerlerine göre değerlendirilmesi sonucunda, algoritmanın Palladyan gramer şemasını öğrenmediği ancak üretimlerinde Palladyan şemasına uygun üretimleri dahil ettiği sonucuna ulaşılmıştır. Bunun nedeninin ÇÜA üretimlerinin dağılımında Palladyan gramer kuralına uygun şemaların da bulunması ve olasılıksal bir üretimin gerçekleştiğini göstermektedir. ÇÜA biçimsel ve mekansal kurguyu değil, piksellerin olasılıksal dağılımını öğrenmiştir. Ancak bu olasılıksal dağılım doğru olan Palladyan gramer kuralları ve mekan sentaks değerlerini de kapsayabildiği değerlendirme sonuçlarında görülmüştür. Bu nedenle otonom mimari

plan şeması üretiminde olasılıksal yöntemlerin hem doğru sonuçlar verebildiği, hem de olasılıklar üzerinden üretim yapması nedeniyle beklenmedik ancak yine olasılıksal dağılıma yakın, ayrık olmayan, üretimleri gerçekleştirebildiği sonucuna varılabilir. Ancak bu durum da ÇÜA algoritmasının üretim yaparken özgür davrandığı ve bu nedenle üretim çıktılarının mimarın kontrolü dışında gerçekleşebileceğini göstermektedir.

7.2 Tartışma

Şekil 8.1’de tez kapsamında mimari plan şeması üretim problemi ile ilgili araştırılan literatür kronolojik olarak ilişkili olduğu plan şeması üretim yöntemi ile verilmiştir. Grafikte kırmızı çizgilerle belirtilmiş dikdörtgen alanlar, plan şeması üreten sistem önerisinde bulunan çalışmalardan, yapay zeka literatürüne referans veren çalışmaları göstermektedir. Grafik bu yönüyle okunduğunda 2000 yılından itibaren mimari plan şeması üreten sistem çalışmalarının, yapay zeka disiplinine yoğunlukla referans verdiği ve yapay zeka ile ilişkisinin kuvvetinin arttığı söylenebilir. Şekil 8.1 yapay zekanın güncel alanı olan üretken yapay sinir ağlarından ÇÜA algoritmalarının, mimari plan şeması üretim probleminde kullanılan en güncel yöntem olduğunu gösterebilmektedir. Bir diğer yandan mimarlık disiplininin mimari plan şeması üretiminde yapay zeka algoritmalarına yönelimi de görülebilmektedir. Yapay zeka ve mimarlık etkileşiminin olduğu araştırmalar literatürde artmaya başladıkça; “Yapay zeka ve mimarlık disiplinlerinin etkileşimleri gerekli midir?”, “Yapay zeka algoritmaları tasarlama eylemine katılabilir mi?”, “Yapay zeka tasarlama eylemine katılıyorsa üretim çıktıları nasıl değerlendirilmeli?”, “Yapay zeka ile mesleğin otonomlaşması mümkün müdür?”, “Yapay zeka ile birlikte mimarın gelecekteki rolü ne olmalıdır.” soruları önemli hale gelmektedir.



Şekil 7.1: Literatürde mimari plan şeması üretim çalışmalarında kullanılan yöntemler.

Gordon Pask (1969) Sibernetiğin Mimarlık İlişkisi (The Architectural Relevance of Cybernetics) metninde yapay zekanın mimari tasarımın belirli yönlerini taklit edebileceğini söylemiştir.

“Yapay zeka bilgisayar programları mimarlığın belirli yönlerini taklit edebilir (Program, mimarlardan, mimarların dilinden, mimari planları keşfetmesi, malzeme kısıtlarının belirlenmesi, müşterinin yorumları vb. ile mimarlık ile ilgili bilgiyi öğrenebilir.). Tasarım için potansiyel yardımcılarıdır; araç benzeri programların akıllı uzantıları olarak hareket edebilirler” (Pask, G., 1969).

Yapay zekanın mimarlık disiplini içerisinde olma sorusu güncel bir soru olmakla birlikte geçmişte de tartışılan bir konu olduğu görülebilir. Yapay zekanın mimarlık disiplini içerisinde yer almasından öte, mimarlık problemi yapay zeka problemi içerisinde yer aldığı söylenmesi daha doğru olabilir. Akıllı davranışlar, muhakeme (reasoning), karar verme, çıkarım yapma, tahmin etme yapay zekanın problem alanını oluşturmaktadır. Goldschmidt & Weil (1998), tasarım eyleminin beceri kazanmak, pratik, ve deneyimin yanında, düşünce sürecinin bir çıktısı olduğunu söylemektedir. Bu özellikleri ile tasarım, yaratıcı düşünce ve problem çözme ile ilgili bir eylemdir (Goldschmidt & Weil, 1998) tasarım eyleminin muhakeme, karar verme, değerlendirme, öneri üretme eylemlerini içinde barındırdığını söylemektedir. Newell (1993), akıl yürütme, problem çözme ve karar verme süreçlerinin yapay zeka araştırma alanının temel sorularından olduğunu söylemiştir. Yapay zeka ile tasarım eylemi arasındaki kesişim noktası olan problem çözme eylemi, tasarım problemlerini yapay zekanın bir problem sınıfı haline getirmektedir. Mimarlığın bir tasarım disiplini olduğu düşünüldüğünde, mimarlık eyleminin de yapay zekanın problem tanımı içerisinde yer aldığı söylenebilir. Bu durumu kanıtlayan bir başka açıklama Eastmen (1973) tarafından yapılmıştır. Eastmen (1973), mimarlık problemi içerisinde mimari plan şeması üretiminin yapay zekanın bir alt problemi olduğunu ve çalışmasının hedefinin mimari plan şeması üretim problemini yapay zeka disiplinine tanıtmak olduğunu söylemiştir. Bu nedenle yapay zeka disiplini mimarlık ile geçmişte etkileşimde olduğu gibi günümüzde teknolojinin gelişmiş imkanları ile daha fazla etkileşimde olabilir.

Bu etkileşim mimarlara yeni bir sorumluluk yüklemektedir. Yapay zeka içinde bulundurduğu veri işleme potansiyeli yüksek araçları ile mimarlık yapma süreçlerinde paradigma kaymaları yaratabilir. Mimarlık disiplininde bu durumu işaret eden ilk denemelerin plan şeması üretim problemine yoğunlaştığını görebilmekteyiz. Mimarlar olarak algoritmaların çalışma sisteminin farkında olmak üretimlerin kontrolünün sağlanması için gerekli olacaktır. Bu tezin çıktısı olan Palladio plan şemalarına bakıldığında üretimlerin sadece piksel dağılım olasılıkları içinde oluşan altkümelerle oluşturulduğu görülmektedir. Mimarlık salt görsel üretme eylemi değil, işlevsel bir görsel üretme eylemi olarak karşımıza çıkmaktadır. Eames (1954) yeni teknolojinin mimarlara sorumluluk yüklediğini, gelecekteki planların mimarlar tarafından yapılıp yapılmadığının, mimarın yeni teknolojiye hakimiyet durumuna bağlı olacağına vurgu yapmıştır.

Eames'in açıklaması önemli bir öngörü olmakla birlikte güncel olarak kullanılan yapay zeka algoritmalarında mesleki olarak büyük bir tehlikenin olmadığı da söylenmelidir. Bir diğer yandan gerçek tehlike algoritmaların kontrolsüz üretimlerinin gerçek hayatta mimari karşılığını bulması olabilir. Elimizdeki algoritmalar rassal araştırmalarla görsel benzerlik yakalayabilen üretimler gerçekleştirebilse de sonuçları henüz tam kontrollü değildir. Bu rassallığın kontrolü yine mimarların sorumluluğunda olacaktır.

Bu tezin de çıktılarından biri olan mimari plan şeması üretimlerine bakıldığında yapay zeka algoritmalarının henüz emekleme döneminde olduğu, biçimsel üretimler gerçekleşebilse de tam bir tasarlama eyleminden söz edilemeyeceği görülebilir. Yine de yapay zekanın gelecekteki yeni algoritmaları ve yeni donanımlarıyla dahi, tasarlama eylemini öğrenebilen bir algoritmanın mümkün olamayacağını söylemek çok kesin bir çıkarım olacaktır. Mimarlık disiplini, teknoloji nedeniyle üstlendiği sorumluluk ile, teknolojiden gelen gelişmelere karşı bağımsızlık kazanmış olarak, doğru reflekslerle yanıt verebilir hale gelecektir.

Teknolojinin mimarlık disiplinine eklediği sorumlulukla birlikte, mimarlık mesleğinde yeni uzmanlaşma alanları öngörülebilir. Mimari veriseti tasarlama, algoritma mimarisi tasarlama, algoritma çıktılarının değerlendirilmesi gibi uzmanlıklar gelecekte mimarlık mesleği içerisinde gelişebilir.

KAYNAKLAR

- Ahmad, A. R., Basir, O. A., Hassanein, K., & Imam, M. H.** (2004). *Improved placement algorithm for layout optimization*. In Proc. of the 2nd Int'l Industrial Engineering Conf.(IIEC'04).
- Alexander, C.** (1967). *Notes on the Synthesis of Form*. Harvard University Press.
- Aytaçlı, B.** (2012). Durum çalışmasına ayrıntılı bir bakış. *Adnan Menderes Üniversitesi Eğitim Fakültesi Eğitim Bilimleri Dergisi*, 3(1), 1-9.
- Banna, S. A., & Spillers, W. R.** (1972). An interactive computer graphics space allocation system. In *Proceedings of the 9th Design Automation Workshop*, 229-237.
- Bayer, J., Bukhari, S. S., & Dengel, A.** (2017). Floor plan generation and auto completion based on recurrent neural networks. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, 2, 49-50. IEEE.
- Brownlee, J.**, (2019). Applications of Deep Learning for Computer Vision. Alındığı Tarih: Kasım 24, 2020, 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. Alındığı Tarih: Eylül 23, 2019. Erişim Adresi : <https://machinelearningmastery.com/upsampling-and-transpose-convolution-layers-for-generative-adversarial-networks/>
- Borji, A.** (2019). Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding*, 179, 41-65.
- Budhiraja, A.**, (2018). Learning Less to Learn Better-Dropout in (Deep) Machine learning. Erişim Adresi : <https://medium.com/@amarbudhiraja/https-medium-com-amarbudhiraja-learning-less-to-learn-better-dropout-in-deep-machine-learning-74334da4bfc5>
- Buffa, E. S.** (1955). Sequence analysis for functional layouts. *Journal of Industrial Engineering*, 6(2), 12-13.
- Chaillou, S.** (2019). *AI+ Architecture: Towards a New Approach*. Harvard University.
- Colakoglu, B.** (2005). Design by grammar: an interpretation and generation of vernacular hayat houses in contemporary context. *Environment and Planning B: Planning and Design*, 32(1), 141-149.
- Dalgic, H. O., Bostanci, E., & Guzel, M. S.** (2017). *Genetic Algorithm Based Floor Planning System*. arXiv preprint arXiv:1704.06016.

- Del Rio-Cidoncha, G., Martínez-Palacios, J., & Iglesias, J. E.** (2007). A multidisciplinary model for floorplan design. *International journal of production research*, 45(15), 3457-3476
- Deshpande, A.**, (t.y.). A Beginner's Guide To Understanding Convolutional Neural Networks. Erişim Adresi : <https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/>
- Domingos, P.** (2015). *The master algorithm: how the quest for the ultimate learning machine will remake our world*. New York: Basic Books, a member of the Perseus Books Group.
- Donald, T.** (1962). A Sumerian Plan In The John Rylands Library1. *Journal of Semitic Studies*, 7(2), 184-190.
- Duarte, J. P.** (2005). A discursive grammar for customizing mass housing: the case of Siza's houses at Malagueira. *Automation in Construction*, 14(2), 265-275.
- Eames, C.** (1954). "Letter to Ian McCallum, Esq. of the Architectural Review (London)," The Work of Charles and Ray Eames Collections of the Manuscript Division, Library of Congress, Box 218.
- Eastman, C. M.** (1975). Automated space planning. *Artificial intelligence*, 4(1), 41-64.
- Eisenstadt, V., Langenhan, C., & Althoff, K. D.** (2019). Generation of Floor Plan Variations with Convolutional Neural Networks and Case-based Reasoning-An approach for transformative adaptation of room configurations within a framework for support of early conceptual design phases. 37. *Education and Research in Computer Aided Architectural Design in Europe Conference*, (79-84 ss.)
- Fortin, G.** (1978). BUBBLE: Relationship diagrams using iterative vector approximation. 15. *Design Automation Conference*, (145-151 ss.).
- Goldschmidt, G., & Weil, M.** (1998). Contents and structure in design reasoning. *Design issues*, 14(3), 85-100.
- Do, E. Y. L., & Gross, M. D.** (1996). Drawing as a means to design reasoning. In *AI and Design*.
- 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.
- Grason, J.** (1971). *An approach to computerized space planning using graph theory*. In Proceedings of the 8th Design automation workshop, 170-178. ACM.
- Hebb, D. O.** (1949). *The organization of behavior: a neuropsychological theory*. J. Wiley; Chapman & Hall.
- Hinton, G. E., Osindero, S., & Teh, Y. W.** (2006). A fast learning algorithm for deep belief nets. *Neural computation*, 18(7), 1527-1554.

- Hu, R., Huang, Z., Tang, Y., van Kaick, O., Zhang, H., & Huang, H.** (2020). *Graph2Plan: Learning Floorplan Generation from Layout Graphs*. arXiv preprint arXiv:2004.13204.
- 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.
- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., & Aila, T.** (2020). *Analyzing and improving the image quality of stylegan*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, (8110-8119 ss.).
- Krizhevsky, A., Sutskever, I., & Hinton, G. E.** (2012). *Imagenet classification with deep convolutional neural networks*. In Advances in neural information processing systems, (1097-1105 ss.).
- Koning, H., & Eizenberg, J.** (1981). The language of the prairie: Frank Lloyd Wright's prairie houses. *Environment and planning B: planning and design*, 8(3), 295-323.
- Koopmans, I. C. ve diğ.** (1957). Assignment Problems and the Location of Economic Activities. *Econometrica*, 25(1), 53-76
- Krejcirik, M.** (1969). Computer-aided plant layout. *Computer-Aided Design*, 2(1), 7-19.
- Levin, P. H.** (1964). Use of graphs to decide the optimum layout of buildings. *The Architects' Journal*, 7, 809-815
- Lobos, D., & Donath, D.** (2010). The problem of space layout in architecture: A survey and reflections. *Arquiteturarevista*, 6(2), 136-161.
- Mcculloch, W. S., & Pitts, W.** (1943). *A logical calculus of the ideas immanent in nervous activity*. Chicago, IL: Univ. Press Medjoub and Yannou, 2001;
- Medjdoub, B., & Yannou, B.** (2001). Dynamic space ordering at a topological level in space planning. *Artificial Intelligence in engineering*, 15(1), 47-60.
- Miller, W. R.** (1970). Computer-aided space planning. In *Proceedings of the 7th Design Automation Workshop*, 28-34.
- Minsky, M. L., & Papert, S.** (1970). *Perceptrons: an introduction to computational geometry*. Cambridge Mass.: The MIT Press.
- Minsky, M.,** (1991). Logical vs. Analogical or Symbolic vs. Connectionist or Neat vs. Scruffy. *AI Magazine*, 12(2), 34-51.
- Nagy, D., Lau, D., Locke, J., Stoddart, J., Villaggi, L., Wang, R., ... & Benjamin, D.** (2017). *Project Discover: An application of generative design for architectural space planning*. In Proceedings of the Symposium on Simulation for Architecture and Urban Design. Society for Computer Simulation International.
- Newell, A.** (1993). Reasoning, problem solving, and decision processes: The problem space as a fundamental category. *Research on integrated intelligence*, 55-80.

- Nilsson, N. J.** (2010). *The quest for artificial intelligence: a history of ideas and achievements*. Cambridge: Cambridge University Press.
- Norman, D.** (2013). *The design of everyday things: Revised and expanded edition*. Basic books.
- Pask, G.** (1969). The architectural relevance of cybernetics. *Architectural Design*, 39(9), 494-496.
- Radford, A., Metz, L., & Chintala, S.** (2015). *Unsupervised representation learning with deep convolutional generative adversarial networks*. arXiv preprint arXiv:1511.06434.
- Rosenblatt, F.** (1959). Perceptron. *SpringerReference*. doi: 10.1007/springerreference_57806
- Rogers, J., & Révész, A.** (2020). Experimental and quasi-experimental designs. *The Routledge handbook of research methods in applied linguistics*. New York: Routledge.
- Rojas, G. S., & Torres, J. F.** (2006). *Genetic algorithms for designing bank offices layouts*. In *Prosiding Third International Conference on Production Research–Americas’Region*
- Ruch, J.** (1978). Interactive space layout: A graph theoretical approach. In *15th Design Automation Conference*, (152-157 ss.).
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J.** (1986). Learning representations by back-propagating errors. *Nature*, 323(6088), 533–536. doi: 10.1038/323533a0
- 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>
- Shaviv, E.** (1986). Layout design problems: systematic approaches. In *Computer-Aided Architectural Design Futures*, (28-52 ss). Butterworth-Heinemann.
- Shi, W., Caballero, J., Theis, L., Huszar, F., Aitken, A., Ledig, C., & Wang, Z.** (2016). Is the deconvolution layer the same as a convolutional layer?. *arXiv preprint arXiv:1609.07009*.
- Stiny, G., & Mitchell, W. J.** (1978). The palladian grammar. *Environment and planning B: Planning and design*, 5(1), 5-18
- Turing, A. M.** (1950). Computing Machinery And Intelligence. *Mind*, LIX(236), 433–460. doi: 10.1093/mind/lix.236.433
- URL1** (t.y.). Erişim Adresi <https://finch3d.com/>
- URL2** (2019, Nisan 12). Erişim Adresi: https://github.com/vdumoulin/conv_arithmetic
- URL3** (t.y.). Erişim Adresi: https://ml4a.github.io/ml4a/looking_inside_neural_nets/
- URL4** (t.y.). Erişim Adresi: <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/>
- URL5** (2018, Mayıs 11). Erişim Adresi: <https://medium.com/@krishnakalyan3/introduction-to-exponential-linear-unit-d3e2904b366c>

- URL6** (1965, Şubat 1). Erişim Adresi: <https://stats.stackexchange.com/questions/154879/a-list-of-co st-functions-used-in-neural-networks-alongside-applications> =
- URL7** (2017, Ekim 25). Erişim Adresi <https://towardsdata science.com/batch-normalization-in-neural-networks-1ac91516821c>
- URL8** (2019, Haziran 4).Erişim Adresi : <https://www.groundai.com/project/generative-adversarial-networks-a-survey-and-taxonomy/1>
- URL9** (t.y.). Erişim Adresi : https://gluon.mxnet.io/chapter14_generative-adversarial-networks/dcgan.html
- URL10** (t.y.).Erişim Adresi : <http://image-net.org/about-over view>
- URL11** (2020, Mayıs 2). Erişim Adresi : https://en.wikipedia.org/wiki/Central_limit_theorem
- Uzun, C., Çolakoğlu, M. B., & İnceoğlu, A.** (2020). GAN as a generative architectural plan layout tool: A case study for training DCGAN with Palladian Plans, and evaluation of DCGAN outputs. *AZ Itu Journal of the Faculty of Architecture*, 17(2), 185-198
- Weinzapfel, G., Johnson, T. E., & Perkins, J.** (1971). *IMAGE: an interactive computer system for multi-constrained spatial synthesis*. In Proceedings of the 8th Design Automation WorkshoP, (101-108 ss.). ACM.
- Yin, R.** (1994). *Case study research: Design and methods*. Thousand Oaks, CA: Sage.
- Yoon K. B.,** (1992). *A Constraint Model of Space Planning*. Southampton, UK: Computational Mechanics Publications.



ÖZGEÇMİŞ



Ad-Soyad : Can Uzun
Doğum Tarihi ve Yeri : 1989 Adapazarı
E-posta : canuzun8@gmail.com

ÖĞRENİM DURUMU:

- **Lisans** : 2012, İTÜ, Mimarlık Fakültesi, Mimarlık Bölümü
- **Yükseklisans** : 2014, İTÜ, Mimarlık, Mimari Tasarım

DOKTORA TEZİNDEN TÜRETİLEN YAYINLAR, SUNUMLAR VE PATENTLER:

- **Uzun, C.**, (2020). The Network of Interactions for an Artificial Architectural Intelligence. In As, İ., & Basu, P.(Ed.), *The Routledge companion to artificial intelligence in architecture*. Abington, Oxon: Routledge. ISBN 9780367424589
- **Uzun, C.**, Çolakoğlu, M. B., & İnceoğlu, A. (2020). GAN as a Generative Architectural Plan Layout Tool: A Case Study for Training DCGAN With Palladian Plans, and Evaluation of DCGAN Outputs. ITU A|Z 17 (2), 185- 198. doi: 10. 5505/itujfa.2020.54037.
- **Uzun, C.**, Colakoglu, M., B., (2019), “Architectural Drawing Recognition : A Case Study for Training The Learning Algorithm with Architectural Plan and Section Drawings”. eCAADe

