



T.C.

**BATMAN ÜNİVERSİTESİ
LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ
BİLGİ TEKNOLOJİLERİ ANABİLİM DALI**

**DERİN EVRİŞİMLİ ÇEKİŞMELİ ÜRETİCİ
AĞLAR İLE YÜZ GÖRÜNTÜLERİNİN ÜRETİMİ**

YÜKSEK LİSANS TEZİ

Nizamettin ÇİÇEKLİ

**Danışman
Prof Dr. Ömer Faruk ERTUĞRUL**

**Ocak-2024
BATMAN
Her Hakkı Saklıdır**

TEZ KABUL VE ONAYI

Nizamettin ÇiÇEKLİ tarafından hazırlanan “**Derin Evrişimli Çekişmeli Üretici Ağlar ile Yüz Görüntülerinin Üretimi**” adlı tez çalışması 12/01/2024 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Batman Üniversitesi Fen Bilimleri Enstitüsü Bilgi Teknolojileri /Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

Jüri Üyeleri

İmza

Başkan

Doç. Dr. Yılmaz KAYA

.....

Danışman

Prof. Dr. Ömer Faruk ERTUĞRUL

.....

Üye

Doç. Dr. Cafer BUDAK

.....

Yukarıdaki sonucu onaylıyorum.

Dr. Öğr. Üyesi Ömer Murat ÖTER
Lisansüstü Eğitim Enstitüsü Müdürü V.

TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

DECLARATION PAGE

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

İmza

Nizamettin ÇİÇEKLİ

Tarih: 12/01/2024

ÖZET

YÜKSEK LİSANS TEZİ

DERİN EVRİŞİMLİ ÇEKİŞMELİ ÜRETİCİ AĞLAR İLE YÜZ GÖRÜNTÜLERİNİN ÜRETİMİ

Nizamettin ÇİÇEKLİ

**Batman Üniversitesi Lisansüstü Eğitim Enstitüsü
Bilgi Teknolojileri Anabilim Dalı**

**Danışman: Prof. Dr. Ömer Faruk ERTUĞRUL
2024, 90 Sayfa**

Jüri

**Başkan : Doç. Dr. Yılmaz KAYA
Danışman : Prof. Dr. Ömer Faruk ERTUĞRUL
Üye : Doç. Dr. Cafer BUDAK**

Bu çalışmada ilkel insandan günümüze kadar bilginin artışı, nesiller arası aktarımı ve bu sürecin büyük veriye dönüşümü incelenmiştir. Aynı zamanda yapay zekanın tarihsel gelişimi, yapay sinir ağlarının yapısı, derin öğrenme, derin öğrenme katmanları ve derin öğrenme modelleri gibi konular ayrıntılı bir şekilde ele alınmıştır. Bunun yanı sıra görüntü oluşturma modellerinden biri olan ve derin öğrenmenin bir alt dalı olan çekişmeli üretici ağlar hakkında detaylı bilgiler sunulmuştur.

Çekişmeli üretici ağların tarihsel süreci, yapılan akademik çalışmalar, gelişimi ve akademik literatüre katkıları titizlikle araştırılmıştır. Özellikle çekişmeli üretici ağların görüntü oluşturma alanında en iyi modellerinden biri olan DEÇÜA modeli ayrıntılı bir şekilde incelenmiştir. DEÇÜA modelini oluşturan üretici ağın yapısı, matematiksel yöntemleri ve deneylerde kullanılan fonksiyonları ile birlikte kayıp fonksiyonu değerleri üzerinde yoğun bir çalışma yapılmıştır. Ayrıca çekişmeli üretici ağların ikinci bileşeni olan ayırıcı ağın yapısı, işlevleri, matematiksel yöntemleri ve deneylerde kullanılan fonksiyonları detaylı bir şekilde ele alınmıştır.

Deney ortamında kullanılan kütüphaneler, GPU, işlemci, dil işleme editörleri ve Google Colab ortamı dikkatlice incelenmiştir.

Sentetik görüntüler üretmek için en yaygın olarak kullanılan veri setleri özenle seçilmiştir. Bu çalışmada iki farklı türde dört veri seti kullanılmıştır. İlk olarak vektör tabanlı görüntüler içeren Cartoonset10k ve Anime Face veri setleri dikkatlice seçilmiş ve detaylı bir şekilde analiz edilmiştir. İkinci olarak piksel tabanlı görüntüler içeren Animal Face ve CelebaFace veri setleri ayrıntılı bir şekilde incelenmiştir. Her bir veri seti üzerinde gerçekleştirilen deneylerde belirli aralıklarla tekrarlanan eğitim süreçleri sonucunda elde edilen 8x8 boyutundaki görüntüler aşama aşama kaydedilmiştir. Bu elde edilen en iyi görüntüler, veri setlerinde rastgele seçilen resimlerle karşılaştırılarak detaylı bir şekilde değerlendirilmiştir. Üretici ağ ile ayırıcı ağın Nash dengesine göre kayıp-kazanç değerleri, matematiksel ve grafiksel yöntemler kullanılarak elde edilmiştir.

Anahtar Kelimeler: Ayırt Edici Ağ, Çekişmeli Üretici Ağlar, Derin Evrişimli Çekişmeli Ağlar, Derin Öğrenme, Üretici Ağ, Yapay Sinir Ağları, Yapay Zekâ.

ABSTRACT

MS/Ph.D THESIS

GENERATION OF FACE IMAGES USING DEEP CONVOLUTIONARY ADVERTISING GENERATIVE NETWORKS

Nizamettin ÇİÇEKLİ

**Batman University Graduate Education Institute
Department of Information Technologies**

Advisor: Prof. Dr. Ömer Faruk ERTUĞRUL

2024, 90 Pages

Jury

President	: Assoc. Dr. Yılmaz KAYA
Advisor	: Prof. Dr. Ömer Faruk ERTUĞRUL
Member	: Assoc. Dr. Cafer BUDAK

In this study, the increase of knowledge from primitive man to the present day, its intergenerational transfer and the transformation of this process into big data are examined. At the same time, topics such as the historical development of artificial intelligence, the structure of artificial neural networks, deep learning, deep learning layers and deep learning models are discussed in detail. In addition, detailed information is presented about adversarial generative networks, which is one of the image generation models and a sub-branch of deep learning.

The historical process of contentious productive networks, academic studies, their development and contributions to academic literature have been meticulously researched. In particular, the DCGAN model, which is one of the best models in the field of image generation of adversarial generative networks, has been examined in detail. An intensive study has been carried out on the structure of the generative network that forms the DCGAN model, its mathematical methods and the functions used in the experiments, as well as the loss function values. In addition, the structure, functions, mathematical methods and functions of the discriminator network, which is the second component of adversarial generator networks, and the functions used in the experiments are discussed in detail.

The libraries, GPU, processor, language processing editors and Google Colab environment used in the experimental environment were carefully examined.

The most commonly used datasets to generate synthetic images have been carefully selected. Four data sets of two different types were used in this study. First, Cartoonset10k and Anime Face datasets containing vector-based images were carefully selected and analyzed in detail. Secondly, Animal Face and CelebaFace datasets containing pixel-based images were examined in detail. In the experiments carried out on each data set, the 8x8 images obtained as a result of the training processes repeated at regular intervals were recorded step by step. These best images were evaluated in detail by comparing them with randomly selected images in the data sets. Loss-gain values of the generating network and the discriminator network according to Nash equilibrium were obtained using mathematical and graphical methods.

Keywords: Discriminative Network, Adversarial Generative Networks, Deep Convolutional Adversarial Networks, Deep Learning, Generative Network, Artificial Neural Networks, Artificial Intelligence.

ÖNSÖZ

Çalışmalarım süresince farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, ufuklarımı açan ve her zaman yakın ilgi ve yardımlarını esirgemeyen danışmanım ve değerli hocam Prof. Dr. Ömer Faruk ERTUĞRUL başta olmak üzere,

Hayatımın her döneminde bana yol gösteren ve dualarıyla, sevgileriyle desteklerini her zaman hissettiğim canım annem ve babama, büyük zenginliklerim olan ablamlara,

Çalışmalarım süresince sevgisini, hoşgörüsünü, desteğini eksik etmeyen ve bana her konuda destek olan sevgili eşim Aynur ÇİÇEKLİ'ye, neşe kaynağım olan çocuklarım Dila Narin, Muhammed Azad ve Azra Berfin'e,

Sonsuz teşekkür ediyorum.

Nizamettin ÇİÇEKLİ
BATMAN-2024

İÇİNDEKİLER

ÖZET	iv
ABSTRACT.....	iv
ÖNSÖZ	iv
İÇİNDEKİLER	iv
ŞEKİLLER DİZİNİ.....	iv
KISALTMALAR	vi
1. GİRİŞ.....	1
2. KAYNAK ARAŞTIRMASI	5
3. MATERYAL VE YÖNTEM.....	25
3.1 Veri Seti	25
3.1.1 Cartoonset10k Veri Seti.....	25
3.1.2 Anime Face Veri Seti.....	26
3.1.3 Animal Face Veri Seti.....	26
3.1.4 CelebFaces Nitelikleri Veri Kümesi (CelebA)	26
3.2. Yapay Zekâ (Artificial İntelligence).....	26
3.3. Makine Öğrenmesi (Machine Learning).....	30
3.4.Yapay Sinir Ağları (Artificial Neural Networks)	32
3.4.1 Yapay Sinir Hücresinde Yer Alan Elemanlar	34
3.3.2 Yapay Sinir Ağı Modeli.....	36
3.5. Derin Öğrenme (Deep Learning).....	36
3.5.1 Giriş Katmanı.....	37
3.5.2 Konvolüsyon Katmanı	37
3.5.3 Aktivasyon Katmanı	39
3.5.4 Havuzlama (Pooling) Katmanı	41
3.5.5 Ezberleme (Dropout) Katmanı.....	42
3.5.6 Tam Bağlantı (Fully Connected, Fc) Katmanı.....	43
3.5.7 Sınıflandırma (Classification) Katmanı	43
3.6 Çekişmeli Üretici Ağlar (ÇÜA)	44
3.6.1 ÇÜA Yapısı.....	44
3.6.2 Matematiksel Alt yapısı	44
3.6.3 ÇÜA Modelleri,	46
3.7. Derin Evrişimli Çekişmeli Üretici Ağlar (DEÇÜA).....	47
3.8 Uygulama (Deney Ortamı)	47
3.8.1 Kütüphaneler.....	49
3.8.2 Üretici (Generator) Ağ.....	51
3.8.3 Ayırt Edici (Discriminator).....	52
3.8.4 Loss Fonksiyonları ve Optimize Ediciler	52
4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....	53

4.1 Eğitim Aşaması.....	53
4.1.1 Ayrımcıyı Eğitimi.....	53
4.1.2 Üretici Eğitimi	54
4.2 Sonuçlar (Görüntüler).....	55
4.2.1 Cartoonset10k Veri Seti Görüntüleri	55
4.2.2 Anime Face Veri Seti ile Oluşturulan Görüntüler	59
4.2.3 Animal Faces veriseti ile oluşturulan görüntüler.....	62
4.2.4 CelebAFaces veriseti ile oluşturulan görüntüler.....	66
5. SONUÇLAR VE ÖNERİLER.....	70
5.1 Sonuçlar	70
5.2 Öneriler	71
6. KAYNAKLAR	72
ÖZGEÇMİŞ	79

ŞEKİLLER DİZİNİ

Şekil 2.1. Çekişmeli üretici ağ modelinin çıktıları (Goodfellow, vd. 2014)	6
Şekil 2.2. DEÇÜA ile oluşturulmuş yatak odası resimleri (Radford, vd. 2015)	7
Şekil 2.3. Ağ genellikle ön planı arka plandan ayırmayı öğrenir (Vondrick, vd. 2016) ..	7
Şekil 2.4. ÇÜA'larla videolardaki eksik kareleri tamamlama (Vondrick, vd. 2016)	8
Şekil 2.5. Fotoğraflardaki eksik kısımların tamamlanması (Pathak, vd. 2016).....	9
Şekil 2.6. Referans görüntüsü/nesnesi olmadan 3D-GAN tarafından vektörlerden oluşturulan nesnelere (Wu, vd. 2016).....	10
Şekil 2.7. ÇÜA'lar ile yüz tamamlama sonuçları. CelabA test veri kümesinden (Li, vd. 2017)	11
Şekil 2.8. IcGAN'ın karmaşık varyasyonları gerçek bir görüntü üzerinde nasıl yeniden oluşturduğuna ve uyguladığına dair örnek (Perarnau, vd. 2016).....	12
Şekil 2.9. CoGAN kullanılarak farklı niteliklerde yüz görüntülerinin üretilmesi (Liu ve Tuzel, 2016)	12
Şekil 2.10. Facescrub veri kümesinden alınan resimlere karşılık oluşturulan emoji (Taigman, vd. 2017).....	13
Şekil 2.11. PrGAN tarafından oluşturulan çeşitli 3 boyutlu şekiller (Gadelha vd. 2017)	14
Şekil 2.12. Çözünürlük yükseltme çalışmasından birkaç örnek (Ledig vd. 2017).....	15
Şekil 2.13. CGAN kullanılarak kalitesi 4 kat artırılmış görüntüler (Bin, vd. 2017)	16
Şekil 2.14. Modelin çeşitli görevlere ilişkin sonuçları (Isola, vd. 2017).....	17
Şekil 2.15. Farklı pozlar altında TP-GAN ile sentez sonuçları (Huang, vd. 2017).....	18
Şekil 2.16. Age-cGAN ile yüzün yeniden yapılandırılması ve yaşlanma örnekleri (Antipov, vd. 2017).....	18
Şekil 2.17. Yüz yaşlanmasına örnekleri (Zhang, vd. 2017)	19
Şekil 2.18. Model ile 4x4 piksellik bir görüntüden 1024x1024 piksellik bir oluşturma (Karras, vd. 2017)	20
Şekil 2.19. Sabit görüntü kısmı ve rastgele niteliklerle oluşturulan görüntüler (Jin, vd. 2017)	21
Şekil 2.20. Modelin diğer modellerle karşılaştırılması (Vasu, vd. 2018).....	23
Şekil 2.21. Yağmurlu bir görüntünün belirtilen yöntemle temizlenmiş bir örneği (Zhang, vd. 2019)	24
Şekil 2.22. BigGAN ile üretilmiş örnekler (Brock, vd. 2018).....	24
Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi	28
Şekil 3.2. Yapay zekâ alanları	30
Şekil 3.3. Bir Sinir hücresinin biyolojik gösterimi	32
Şekil 3.4. Bir sinir ağının biyolojik ve matematiksel gösterimi	33
Şekil 3.5. Bir sinir ağının temel mantıksal önermeleri.	34
Şekil 3.6. Yapay sinir hücresi (Yavuz ve Deveci, 2012).....	35
Şekil 3.7. Klasik derin öğrenme mimarisi (Architecture of Deep Learning)	37
Şekil 3.8. Konvolüsyon katmanı ve filtre uygulaması.....	38
Şekil 3.9. Havuzlama katmanı (Metlek ve Kayaalp 2020).....	41
Şekil 3.10. (a) Yapay sinir ağı, (b) Dropout uygulanmış sinir ağı (Çarpı işareti atılmış nöronlar ağdan çıkarılmış)	42
Şekil 3.11. ÇÜA Eğitiminin İlerlemesi (Goodfellow, vd. 2014).....	45
Şekil 3.12. Deney ortamında Colab'ın sundu GPU özellikleri.....	48
Şekil 3.13. Deney ortamında Colab'ın sunduğu RAM özellikleri	49
Şekil 4.1. Üretici ağın ilk görüntüsü ile 500 iterasyondaki görüntüler.....	55
Şekil 4.2. Üretici ağın 800 ile 1000 iterasyondaki görüntüler.....	56

Şekil 4.3. Üretici ağın 1200 ile 1400 iterasyondaki görüntüler.....	56
Şekil 4.4. Üretici ağın ürettiği en iyi görüntü ile veri setinden seçilen rasgele görüntüler	57
Şekil 4.5. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik.....	57
Şekil 4.7. Üretici ağın ilk görüntü ile 5000 iterasyondaki görüntüler	59
Şekil 4.8. Üretici ağın 10000 ve 15000 iterasyondaki görüntüler	60
Şekil 4.9. Üretici ağın 20000 ile 25000 iterasyondaki görüntüler.....	60
Şekil 4.10. Üretici ağın üretmiş olduğu en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler.....	61
Şekil 4.11. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik.....	61
Şekil 4.13. Üretici Ağın ilk görüntü ile 2000 iterasyondaki görüntüler	62
Şekil 4.14. Üretici ağın 3000 ile 4000 iterasyondaki görüntüleri.....	63
Şekil 4.15. Üretici ağın 5000 ile 6000 iterasyondaki görüntüleri.....	64
Şekil 4.16. Üretici ağın üretmiş olduğu en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler.....	64
Şekil 4.17. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik.....	65
Şekil 4.19. Üretici ağın ürettiği ilk görüntü ve 500 iterasyondaki görüntüler.....	66
Şekil 4.20. Üretici ağın 2000 ile 3000 iterasyondaki görüntüler.....	67
Şekil 4.22. Üretici ağın üretmiş olduğu en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler.....	68
Şekil 4.23. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik.....	68

KISALTMALAR

3D-GAN	: Üç Boyutlu Çekişmeli Üretici Ağlar
cGAN	: Koşullu Çekişmeli Üretici Ağlar
CoGAN	: Birleştirilmiş Çekişmeli Üretici Ağlar
ÇÜA	: Çekişmeli Üretici Ağlar
D	: Ayırt Edici Ağ (Discriminator)
DEÇÜA	: Derin Evrişimli Çekişmeli Üretici Ağlar
DESA	: Derin Evrişimli Sinir Ağı
DL	: Derin öğrenme (Deep Learning)
EDSR	: Geliştirilmiş Algısal Süper Çözünürlük Ağı
ESA	: Evrişimli Sinir Ağı
EPSR	: Gelişmiş Algısal Süper Çözünürlük Ağı
G	: Üretici Ağ (Generator)
GAN	: Generative Adversarial Networks
IcGAN	: Tersine Çevrilebilir Koşullu Çekişmeli Üretici Ağlar
ML	: Makine öğrenimi (Machine Learning)
IoT	: Nesnelerin İnterneti
SRGAN	: Süper Çözünürlüklü Çekişmeli Üretici Ağlar
YZ	: Yapay Zekâ (Artificial Intelligence)

1. GİRİŞ

İnsanođlu, varoluşundan bu yana sürekli olarak daha güvenli bir çevre ve sürdürülebilir bir yaşam için gelişme göstermektedir. Hayatta kalmak, beslenmek, neslini sürdürmek, salgın hastalıklarla mücadele etmek, doğal afetlere karşı korunmak gibi birçok zorlu durum ile mücadele etmek zorunda kalmıştır (Ünal, 2009).

İnsan, alet yapma yeteneğine sahip tek canlıdır. Zekasını kullanarak taşlardan, hayvan kemiklerinden, ağaçlardan ve daha sonra işlenen metallere ilkel aletler yapmıştır. Edinilen bu bilgiler ilk insanlardan itibaren babadan ođula ve nesilden nesile aktararak, geçmişten bugüne kadar büyük bir birikim oluşturmuştur. Yazının icadıyla birlikte bilgi kalıcı hale gelmiş ve farklı bölgelere taşınabilir hale gelmiştir. Bu durum, bilginin kolayca aktarılmasını sağlamış ve birçok kültürde yayılmıştır. Sümerler, Mısırlılar, Yunanlılar, Romalılar, Hintliler ve Araplar gibi toplumlar da bilgi birikimi sayesinde teknolojik gelişmeler ortaya çıkmıştır (Düvenci, 2018). Sümerlerde yazı, Mısırlılarda tarım, Hintlilerde matematik, Yunanlılarda mantık ve felsefe, Araplarda astronomi ve tıpta gelişmeler görülmüştür.

Yapay zekâ kavramı, Yunan mitolojisinde ve felsefesinde de yer bulmuştur. Antik Mısır doğumlu Yunan matematikçi ve mühendis Heron (Heron of Alexandria veya Heron Alexandrinus olarak da bilinir), M.Ö. 1. yüzyılda yaşamış ve bazı otomatik mekanizmalar tasarlamıştır. Bu otomatik mekanizmalar, yapay zekânın erken örnekleri olarak kabul edilir (Yakıt ve Külcü, 2016). Bunu yanında 1900-1901 yılları arasında süngerciler tarafından batık bir gemide bulunan Antikitera düzeneđi bilinen ilk analog bilgisayar olarak kabul edilir. Ancak modern anlamda Yapay Zekâ (Artificial Intelligence) terimi ve kavramı daha sonraki dönemlerde gelişmiştir. Daha yakın zamanda, Alan Turing'in 1950 yılında "Turing Testi"ni önermesi, yapay zekâ araştırmalarının hız kazanmasına yol açmıştır (Pehlivan, 2019). Alan Turing'in "Makineler düşünebilir mi?" sorusu, bilgisayarın sorgucunun sorularına verdiği cevaplarla onu kendisinin insan olduğuna inandırıp inandıramayacağı temelinde ele alınmaktadır.

John McCarthy, vd. 1955 yılında Amerika Birleşik Devletleri'ndeki Rockefeller Vakfı'na bir teklif sunarak yapay zekâ kavramını önerdi (Sucu, 2019). Bu teklif, "Yapay Zekâ" teriminin ilk kez kullanılmasını sağladı.

Yapay zekâ, 1960-1970'li yıllarda satranç ve dama gibi oyunları oynayabilen, temel hesaplamalar yapabilen ve matematiksel problemleri çözebilen bir bilgisayar yeteneđi olarak kabul edildi (Coşkun ve Güllerođlu, 2021). 1980-1990'lı yıllarda ise risk

değerlendirmesi ve karar verme yetenekleri gibi alanlarda kullanılmaya başlandı (Pirim, 2006). 2000'li yılların başında, bilgisayarların hesaplama gücünün artmasıyla birlikte öğrenen sistemlerin mümkün olduğu anlaşıldı. Bu dönemde yapay zekâ, özellikle görsel ve ses tanıma gibi insanlara özgü olan görevlere odaklandı.

Günümüzde yapay zekâ, birçok farklı alanda çeşitli uygulamalar ve işlevler sunmaktadır. Otomasyon ve robotik, sağlık hizmetleri, finansal hizmetler, Nesnelerin İnterneti (IoT), oyun ve eğlence, doğal dil işleme, otonom araçlar, güvenlik ve izleme, enerji ve kaynak yönetimi, eğitim gibi birçok alanda gelişme kaydetmektedir.

Yapay zekâ, geniş bir kapsama alanına sahiptir ve alt disiplinler içermektedir. Bu alanlardan biri olan makine öğrenimi, veriyi değerlendirip modeller oluşturan yapay zekanın bir parçasıdır (Atalay ve Çelik, 2017). Makine öğrenimi, Donald Hebb'in 1949 yılında yayınladığı "The Organization of Behavior" isimli kitapta ele alınmış ve McCulloch ve Pitts tarafından önerilen sinir ağındaki öğrenme sürecine çözüm getirmiştir (Elmas, 2021). Arthur Lee Samuel, bilgisayar oyunları ve yapay zekâ alanında öncü bir isim olarak kabul edilir (Ersöz ve Çınar, 2021). Samuel, bilgisayarların kendi deneyimlerinden öğrenme yeteneğine sahip olabileceğini savunmuştur. Makine öğrenimi araştırmalarının temsili bir kitabı olan Nilsson'un "Öğrenme Makineleri" adlı kitabı, 1960'larda yayımlanmış ve genellikle örüntü sınıflandırması üzerine odaklanmıştır (Samuel, 1959).

Makine Öğreniminin bir diğer alt dalı Yapay sinir ağları modelidir. Warren McCulloch ve Walter Pitts tarafından 1943 yılında geliştirilmiştir (Elmas, 2021). McCulloch ve Pitts, insan beyninin hesaplama yeteneğinden ilham alarak elektrik devrelerini kullanarak bir yapay sinir ağı modeli oluşturmuşlardır.

Makine öğrenmesi, veri kullanarak modeller oluşturur ve bu modellerle yeni verilerin tahminini yapar. Makine öğrenmesi genellikle düzenlenmiş veri setlerini (sayılar, resimler, harfler, karakterler vb.) girdi olarak alır ve basit çıktılar üretir, örneğin sınıflandırma, kümeleme ve regresyon gibi. Makine öğrenmesi, veriden bilgi çıkarma bilimidir (Diri, 2014). Bu alan istatistik, matematik, bilgisayar bilimi ve alan bilgisinin kesişim noktasında bulunur.

Yapay sinir ağlarının daha karmaşık problemlere çözüm olması için Derin öğrenme modelleri geliştirildi. Derin öğrenme, yapay sinir ağlarının bir türüdür ve genellikle daha karmaşık ve derin yapıya sahiptir. Derin sinir ağları, karmaşık veri görevlerini öğrenmek ve temsil etmek için kullanılır. Terim olarak "Derin Öğrenme" 1986 yılında Rina Dechter tarafından yayınlanan bir makalede kullanılmıştır (Elmas,

2021). Günümüzde veri miktarı sürekli olarak artmaktadır. Büyük veri (Big Data), geleneksel veri işleme araçlarıyla yönetilemeyen ve analiz edilemeyen büyük miktardaki veri setlerini ifade eder (Atalay ve Çelik, 2017). Geleneksel yöntemler ve araçlar bu veriyi işlemek ve bilgiye dönüştürmek için yetersiz kalmaktadır. Bu durum, yeni fırsatlar sunmasının yanı sıra zorluklar da içermektedir. Grafik işlemcilerin ve bulut bilişim sistemlerinin gelişmesi, daha fazla yapay sinir ağı katmanının işlenmesine olanak sağlamıştır. Bu yaklaşım, derin öğrenme alanında farklı alt modellerin geliştirilmesine yol açmıştır.

Derin öğrenme birçok alt modellerden oluşmaktadır. Bu modellerden bir olan üretici modeller önemli bir araştırma alanıdır. Bu modeller, gerçek dünyadaki örneklerle benzerlik gösteren yeni görüntüler veya veriler üretmeyi hedeflemektedir. Bu alandaki dikkat çekici bir yaklaşım Çekişmeli Üretici Ağlar (ÇÜA'lar) dır. ÇÜA'lar biri ile rekabet eden iki yapay sinir ağdan oluşur. Bu ağlar Üretici ağ ve Ayırıcı ağdır. Üretici, gerçek görüntülerle ayırt edilemez görüntüler üretmeye çalışırken, Ayırıcı ağ, görüntünün gerçek mi yoksa üretilmiş mi olduğunu belirlemeye çalışır. Bu rekabetçi eğitim süreci, giderek daha gerçekçi görüntülerin üretilmesine yol açar.

Üretici modeller, sanat, eğlence ve veri artırma gibi çeşitli alanlarda uygulanmıştır. Araştırmacılar, yüksek kaliteli görüntüler oluşturmak için sürekli olarak bu modelleri geliştirmekte ve gerçekçi insan avaturları oluşturmadan sanat eserleri üretmeye kadar çeşitli uygulamalarla ilgilenmektedirler.

Bu tez çalışmasında, Çekişmeli üretici Ağların bir modeli olan DEÇÜA (DCGAN) modeli kullanılmıştır. ÇÜA'lar, Üretici ve Ayırt Edici ağlarından oluşur. Üretici ağ, gerçek verilere benzeyen sahte veriler üretmeye çalışırken, Ayırt Edici ağ, Üretici ağ tarafından üretilen verilerin gerçeklik derecesini belirlemeye çalışır. Bu süreç, Üretici ağın daha inandırıcı veriler üretmesini ve Ayırt Edici ağın daha doğru tahminler yapmasını sağlar.

ÇÜA'lar, son yıllarda gerçekçi ve yüksek kaliteli veriler üretmede etkili olduğu gösterilen popüler bir derin öğrenme modelidir. Bu tez çalışmasının amacı, DEÇÜA modelinin farklı veri setleri ile kullanıldığında görüntü oluşturma yeteneğini araştırmak ve değerlendirmektir. Ayrıca, farklı veri setleri kullanarak DEÇÜA'ların performansını karşılaştırarak, hangi veri setlerinin daha başarılı sonuçlar verdiğini belirlemeyi amaçlamaktadır. Bu bilgiler, DEÇÜA tekniklerinin iyileştirilmesi ve daha kaliteli, gerçekçi ve çeşitli görüntülerin üretilmesi için kullanılacaktır.

Bu alıřma, DEÜA'ların grnt sentezi alanında bilgi ve anlayıřı artırmayı hedeflemektedir. Metot olarak ÜA'ların DEÜA modelini kullanarak, farklı veri setleri zerinde yapılan deneylerin sonuları karřılařtırılacaktır. Bu sayede, DEÜA 'ların hangi veri setlerinde daha iyi performans gsterdiĐi belirlenecektir.

2. KAYNAK ARAŞTIRMASI

ÇÜA, yapay zekâ algoritmalarıdır ve üretken modelleme sorununu çözmek için kullanılırlar. Bu modeller, bir eğitim veri setinin olasılık dağılımını öğrenmeyi ve daha fazla örnek üretmek için bu öğrenmeyi kullanmayı hedefler. ÇÜA'lar özellikle gerçekçi ve yüksek çözünürlüklü görüntüler oluşturma konusunda başarılıdırlar. Bu algoritmalar, çeşitli görevler için araştırma ortamlarında uygulanmış ve oyun teorisi temeline dayanan benzersiz araştırma fırsatları sunmuşlardır (Goodfellow vd. 2014).

Günümüzde yapay zekâ alanındaki mevcut yaklaşımlar, genellikle denetimli öğrenmeye dayanmaktadır. Denetimli öğrenme, makine öğrenimi algoritmalarının, girdi-çıkı çiftleri veri kümesiyle sunulurken her girdiyi ilgili çıktısıyla ilişkilendirmeyi öğrenmesini içerir. Ancak yüksek doğruluk elde etmek için denetimli öğrenme yöntemi geniş miktarda eğitim verisi ve insan gözetimi gerektirir. Bu nedenle, araştırmacılar denetimsiz öğrenme yöntemlerine yönelerek üretici modelleri araştırmışlardır. ÇÜA'lar, denetimsiz öğrenme alanında popüler bir yaklaşım olup verinin temel dağılımını öğrenmeyi ve yeni örnekler üretmeyi hedefler. Bu şekilde, ÇÜA'lar insan gözetimine ve eğitim verisi miktarına bağımlılığı azaltarak önemli bir rol oynarlar. Bu nedenle, gelecekte yapay zekâ alanındaki gelişmelerde daha etkili ve verimli bir rol oynamaları beklenmektedir (Goodfellow vd. 2014).

Çekişmeli ağlar içeren bir fikir, ilk olarak 2010 yılında Olli Niemitalo'nun bir blog yazısında ortaya atılmıştır. Ne yazık ki bu fikir hiçbir zaman uygulanmamıştır çünkü jeneratörde stokastisite (rastgelelik veya belirsizlik) unsurları bulunmamıştır (Anonymous, 2023). 1991 yılında Juergen Schmidhuber, bir ağın kazancının diğer ağın kaybını oluşturduğu, rekabetçi ve üretken sinir ağlarını içeren zero-sum game (sıfır toplamlı bir oyun) yapısını ortaya koymuştur. İlk ağ, rastgelelik (stochasticity) özelliğine sahip üretken bir yapıyla olasılık dağılımını modellemektedir. İkinci ağ ise, bu modele uygun tepkileri tahmin etmek için gradyan inişini öğrenir. Bu yaklaşım Yapay Merak (Artificial Curiosity) olarak adlandırılmıştır (Anonymous, 2023). 2013 yılında Li, Gauci ve Gross tarafından hayvan davranışını modellemek için benzer bir fikir kullanıldı (Anonymous, 2023).

ÇÜA, ilk olarak 2014 yılında Ian Goodfellow ve arkadaşları tarafından yayınlanan "Generative Adversarial Nets" başlıklı makalede tanıtıldı. Bu makale, yapay zekâ alanında büyük bir etki yarattı. MNIST el yazısı rakam veri kümesi, CIFAR-10 obje

resimleri veri kümesi ve Toronto yüz veri kümesi gibi farklı veri kaynakları kullanılarak ÇÜA ile elde edilen sonuçlar ayrıntılı bir şekilde sunuldu (Goodfellow, vd. 2014).



Şekil 2.1. Çekişmeli üretici ağ modelinin çıktıları (Goodfellow, vd. 2014)

İlk ÇÜA modelinin tanıtımı, derin öğrenmenin yapay zekâ alanında popülerliğini artırmıştır. ÇÜA'lar, gerçekçi görüntü, ses, metin ve diğer verilerin üretiminde büyük bir başarı elde etmiştir. Aynı zamanda, ÇÜA'lar; transfer öğrenme, veri artırma, görüntü tamamlama ve diğer birçok uygulama için de kullanılmaktadır.

ÇÜA kavramının derin evrişim ağları ile birleştirilmesiyle Derin evrişimli çekişmeli Üretici ağlar ortaya çıkmıştır. DEÇÜA'lar, özellikle görüntü ve benzeri veri türlerinin üretimi için kullanılan bir ÇÜA modelidir.

DEÇÜA'lar, önceki ÇÜA modellerine kıyasla daha kararlı ve daha yüksek kaliteli sonuçlar üretme yeteneğine sahiptir. Bu tür ağlar, girdi olarak rastgele gürültü vektörlerini alır ve bu vektörleri gerçekçi görüntüler üreten yapay görüntülere dönüştürmek için evrişim tabanlı bir yapı kullanır. Bu şekilde, gerçekçi ve ayrıntılı görüntülerin oluşturulması hedeflenir.

DEÇÜA'lar, 2015 yılında Radford ve arkadaşları tarafından yayınlanan bir makale ile tanıtılmıştır. Bu makale, DEÇÜA'ların temel prensiplerini ve mimarisini açıklar ve görüntü sentezi alanında önemli bir ilerleme kaydedilmesini sağlar (Radford, vd. 2015).



Şekil 2.2. DEÇÜA ile oluşturulmuş yatak odası resimleri (Radford, vd. 2015)

Vondrick ve arkadaşları tarafından yayınlanan makalede, etiketlenmemiş video verileri kullanılarak sahne dinamiklerinin bir modeli öğrenilmeye çalışılmıştır. Bu modelin, video tanıma görevleri gibi eylem sınıflandırması ve gelecek tahmini gibi video oluşturma görevleri için faydalı olması hedeflenmektedir.

Önerilen yaklaşım, özellikle videolar için tasarlanmış bir modeldir. Bu model uzaysal-zamansal evrişimsel mimariyi içermektedir. Bu mimari, bir sahnenin ön plan ve arka plan bileşenlerini birbirinden ayırmayı amaçlamaktadır. Şekil 2.3'te, makaleden alınan çalışma örnekleri görülmektedir (Vondrick, vd. 2016).



Şekil 2.3. Ağ genellikle ön planı arka plandan ayırmayı öğrenir (Vondrick, vd. 2016)

Yazarlar, yaptıkları deneylerle bu modelin tam kare hızında ve bir saniyeye kadar uzunluğa sahip kısa videolar oluşturabildiğini göstermişlerdir. Özellikle modelin video oluşturma kalitesi açısından basit temel yöntemlere göre daha iyi performans sergilediği belirtilmiştir (Vondrick vd. 2016).

Araştırmacılar, modellerin gelecekteki olası sahneleri tahmin etmedeki faydasını göstermektedir. Model, etiketlenmemiş video verilerinden öğrenerek, bir sahnedeki karmaşık dinamikleri ve ilişkileri yakalama yeteneği kazanır ve tutarlı video dizileri oluşturabilir. Uzay-zaman evrişimlerinin ÇÜA mimarisine entegrasyonu, bu yeteneğe katkıda bulunmaktadır (Vondrick, vd. 2016).



Şekil 2.4. ÇÜA'larla videolardaki eksik kareleri tamamlama (Vondrick, vd. 2016)

Pathak ve arkadaşları tarafından yayınlanan makalede, denetimsiz bir görsel özellik öğrenme algoritmasına odaklanan bağlam tabanlı piksel tahmini üzerinde durmaktadır. Çalışmada, Bağlam Kodlayıcılar (Context Encoders) adı verilen bir kavram önerilmekte olup, belirli bir görüntü bölgesinin içeriğini çevre bağlamına dayalı olarak oluşturmak için Evrişimli Sinir Ağının (ESA) eğitilmesi üzerine odaklanılmaktadır (Pathak, vd. 2016).

Bu görevde başarılı olabilmek için bağlam kodlayıcıların, tam görüntü içeriğini anlaması ve eksik kısımlar için makul tahminler oluşturması gerekmektedir. Bağlam kodlayıcıların eğitim süreci, standart piksel bazında yeniden yapılandırma kaybı ve yeniden yapılandırma ile rekabetçi kaybın bir kombinasyonunun denenmesini içermektedir. İkinci kombinasyon, çıktıdaki birden fazla modu etkili bir şekilde işleyebildiği için daha keskin sonuçlar sağlamaktadır (Pathak, vd. 2016).

Bu çalışma, bağlam kodlayıcıların sadece nesnelere görsel görünümünü değil, aynı zamanda görsel yapıların altında yatan semantiği de yakalayabilen bir temsili öğrendiğini göstermektedir. Yazarlar; sınıflandırma, nesne algılama ve görüntü bölümlendirme gibi görevler için ön eğitim evrişimli sinir ağları aracılığıyla öğrenilen özelliklerin etkinliğini nicel olarak sergilemektedir (Pathak, vd. 2016).

Ek olarak, makale anlamsal iç boyama görevlerinde bağlam kodlayıcıların uygulanmasını vurgulamaktadır. Bu kodlayıcılar bağımsız olarak veya görüntülerin eksik

veya kapalı kısımlarının doldurulması ile ilgili görevlerde şekil 2.5'te olduğu görülmektedir.



Şekil 2.5. Fotoğraflardaki eksik kısımların tamamlanması (Pathak, vd. 2016)

Wu ve arkadaşları tarafından yayınlanan makale, 3D nesne üretme problemine odaklanmakta ve bu amaçla 3D Çekişmeli Üretici Ağ (3D-GAN) adlı yeni bir çerçeve sunmaktadır.

3D-GAN çerçevesi, son dönemdeki hacimsel konvolüsyon ağları ve çekişmeli üretici ağlar alanındaki ilerlemeleri kullanarak 3D nesnelere olasılıksal bir uzaydan üretmektedir.

DeneySEL sonuçları; önerilen yöntemin, yüksek kaliteli 3D nesnelere üretme yeteneğini sergilemektedir. Bu çerçeve üzerinden denetimsiz bir şekilde öğrenilen özellikler, 3D nesne tanıma görevlerinde etkileyici performans elde eder ve denetimli öğrenme yöntemleriyle karşılaştırılabilir başarılar elde eder. Bu yaklaşım, farklı ve gerçekçi 3D nesnelere üretme konusunda umut vaat ederken aynı zamanda nesne tanıma görevleri için kullanışlı özellikler sağlar (Wu, vd. 2016).



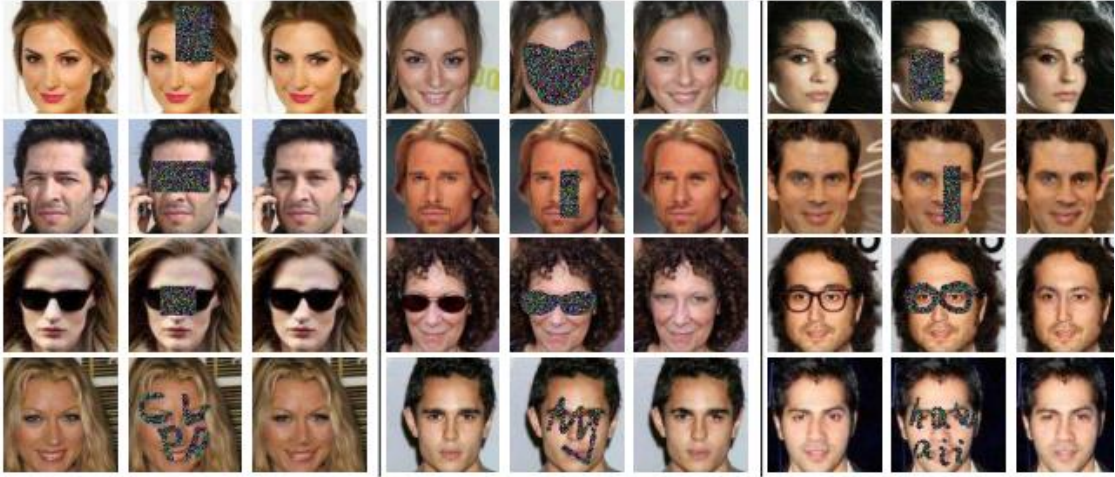
Şekil 2.6. Referans görüntüsü/nesnesi olmadan 3D-GAN tarafından vektörlerden oluşturulan nesnelere (Wu, vd. 2016).

Li ve arkadaşları tarafından yayınlanan makalede, eksik yüz görüntülerinin etkili bir şekilde tamamlanması için derin bir üretken model sunmaktadır.

Görüntülerde arka plan tamamlama konusu geniş ölçüde incelenmiş olsa da yüz tamamlama daha zorlayıcı bir görev sunar çünkü genellikle eksik pikselleri olan anahtar bileşenlerin (örneğin, gözler ve ağızlar) anlamsal olarak yeni pikseller üretilmesini gerektirir.

Bu algoritma, geleneksel olmayan parametrik yaklaşımların aksine, eksik bölgeler için içeriği doğrudan bir sinir ağı kullanarak üretir. Önerilen model, farklı kayıp fonksiyonlarının bir kombinasyonu ile eğitilir.

Araştırma, algoritmanın eksik piksellerin büyük bir alanını rastgele şekillerde işleyebilme yeteneğini ve gerçekçi yüz tamamlama sonuçları üretme kabiliyetini göstermektedir. Şekil 2.6, modelin yapılan deneylerde görsel olarak tutarlı yüz özellikleri üretme konusundaki etkinliğini sergilemiştir (Li, vd. 2017).



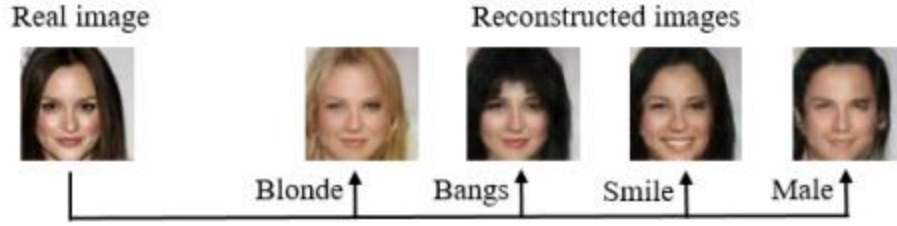
Şekil 2.7. ÇÜA'lar ile yüz tamamlama sonuçları. CelabA test veri kümesinden (Li, vd. 2017)

Perarnau ve arkadaşları tarafından 2016 yılında yayınlanan makalede, ÇÜA modelini temel alarak koşullu ÇÜA'lar (cGAN'lar) üzerine odaklanmaktadır ve bu modeli ters çevrilebilir hale getirmeyi amaçlamaktadır.

cGAN'lar, dış bilginin belirli görsel temsillerini üretmeye olanak tanıyan bir ÇÜA türüdür. Bu çalışmada, cGAN'ların işlem sırasını tersine çevirmeye yönelik bir yaklaşım ele alınmıştır. Yani, gerçek bir görüntüyü latent bir uzaya ve koşullu bir temsile dönüştürmek (encoder) ve bu sayede gerçek görüntüleri istenen niteliklere göre yeniden oluşturmak ve değiştirmek (görüntü düzenleme) mümkün hale gelmiştir.

Ayrıca çalışmada cGAN'ların tasarımı da değerlendirilmiştir. Bir encoder'ın bir cGAN ile birleştirilmesi, Invertible cGAN (IcGAN) olarak adlandırılan bir modeli ortaya çıkarmıştır. IcGAN, gerçek görüntüleri belirli karmaşık değişikliklerle yeniden oluşturmayı sağlayan bir yapı sunar (Perarnau, vd. 2016).

Bu çalışma, cGAN tabanlı yaklaşımların, görüntü düzenleme ve manipülasyonunda kullanılabilirliğini geliştirmek ve gerçek görüntüler üzerinde değişiklik yapma yeteneğini artırmak için önemli bir adım olarak kabul edilmektedir (Perarnau, vd. 2016).



Şekil 2.8. IcGAN'ın karmaşık varyasyonları gerçek bir görüntü üzerinde nasıl yeniden oluşturduğuna ve uyguladığına dair örnek (Perarnau, vd. 2016).

Liu ve Tuzel tarafından 2016 yılında yayınlanan makalede, CoGAN (Birleştirilmiş Çekişmeli Üretici Ağlar) kavramını sunarak iki farklı alana ait karşılıklı görüntü çiftleri üretmeye odaklanmaktadır.

CoGAN, iki alana ait görüntüleri üretmekten sorumlu olan iki karşılıklı üretici ağdan oluşur. İlginç olan şudur: CoGAN, eğitim veri setinde iki alan arasında karşılıklı görüntü çiftlerinin bulunmadığı durumlarda bile bu karşılıklı görüntü çiftlerini üretmeyi öğrenir. Başka bir deyişle CoGAN, iki ayrı alanın marjinalerinden çekilen görüntülerden, iki alanın birlikte dağılımını öğrenir. Bu, mevcut çoklu modlu üretici modellerin eğitimi için karşılıklı görüntü çiftlerine ihtiyaç duyanlardan farklı bir yaklaşımdır.

Makalede, CoGAN'ın çeşitli karşılıklı görüntü eşleme görevlerine nasıl uygulandığı gösterilmektedir. Her görev için CoGAN, ikna edici karşılıklı görüntü çiftleri üretmeyi öğrenir. Ayrıca CoGAN, çerçevesinin alan uyarlama ve farklı alanlarda görüntü üretme görevleri için nasıl kullanıldığı da açıklanır.

Bu çalışma, birçok uygulama için karşılıklı görüntü eşleştirmesinin gerektiği durumlarda önemli bir katkı sunmaktadır ve CoGAN, iki farklı alan arasında uygun çiftlerin olmadığı durumlarda bile bu tür karşılıklı eşlemeyi gerçekleştirebilen bir çerçeve sunmaktadır.



Şekil 2.9. CoGAN kullanılarak farklı niteliklerde yüz görüntülerinin üretilmesi (Liu ve Tuzel, 2016)

Taigman ve arkadaşları tarafından yapılan makalede, etiketlenmiş veri kullanmadan bir alandan diğerine görüntü transferi yapma zorluğunu ele almaktadır. Yazarlar, her iki alandan da girdileri kabul eden bir fonksiyonun davranışını korurken örnekleri bir alandan diğerine eşlemeyi hedeflemektedir.

Önerilen Alan Transfer Ağı (ATA), eğitim için bileşik bir kayıp fonksiyonu kullanmaktadır. Bu kayıp fonksiyonu, çok sınıflı ÇÜA kaybı, verilen fonksiyonun çıktısının sürekliliğini sağlayan bir bileşen (f-süreklilik) ve Üretici fonksiyonun hedef alanından gelen örnekleri kendi üzerine eşlemesini teşvik eden bir düzenleyici bileşen içermektedir. Bu kayıp terimlerinin birleşimi, ağın karmaşık ilişkileri yakalayabilmesini ve transfer edilen varlıkların görsel olarak ikna edici görüntülerini üretebilmesini sağlamaktadır, aynı zamanda varlıkların kimliklerini korumaktadır.

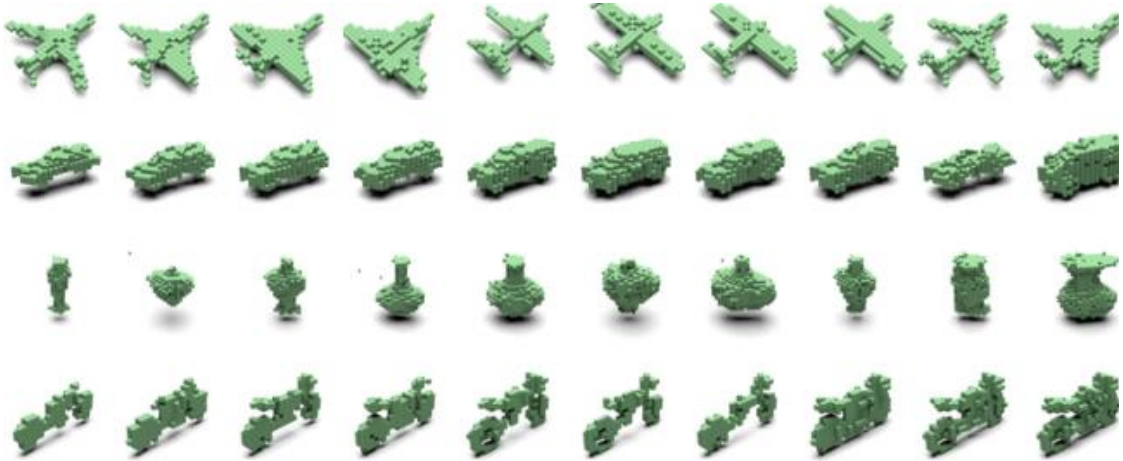
Araştırma, önerilen yöntemin farklı görsel alanlarda, örneğin rakam ve yüz görüntülerindeki etkinliğini göstermektedir. Model, eğitim sırasında daha önce görülmemiş varlıkların hem kimliklerini koruyan hem de görsel olarak gerçekçi görüntülerini üretebilme yeteneğini sergilemektedir. Etiketsiz bir eğitim kurulumunun kullanımı ve alanlar arası görüntü üretme yeteneği, tarz transferi, alan adaptasyonu ve veri artırma gibi görevler için önemli sonuçlar doğurmaktadır.



Şekil 2.10. Facescrub veri kümesinden alınan resimlere karşılık oluşturulan emoji (Taigman, vd. 2017)

Gadelha ve arkadaşları tarafından 2017 sunulan bir makalede, bilinmeyen görüş açılarından alınan çoklu nesnelerin 2D görüntülerinden 3D yapıların dağılımını çıkarmayı hedefleyen bir çalışmayı ele almaktadır.

Bu çalışmada, "Projective Generative Adversarial Networks" (PrGANs) adı verilen derin bir üretici model kullanılmaktadır. Bu model, 3D şekillerin 2D görüntülerle eşleşen projeksiyonlarını öğrenmek için eğitilmektedir. Eklenen projeksiyon modülü sayesinde, 3D şekil dağılımının temelini öğrenirken 3D verilerin bakış açısı bilgisine veya işaretlemeye ihtiyaç duymadan çalışabilme imkânı sağlar. Çalışma; sandalyeler, uçaklar ve arabalar gibi çeşitli şekil kategorileri için eğitilmiş ÇÜA'larla karşılaştırıldığında benzer kalitede 3D şekiller üretme yeteneğini göstermektedir. Deneyler ayrıca 2D şekillerin geometrisi ve bakış açısına ayrılmış bir temsilin başarılı bir üretici modeline yol açtığını göstermektedir. Temel avantajı, modelin bir giriş görüntüsünden 3D yapı ve bakış açısı tahmini yapabilmesi ve yeni görünüm oluşturabilmesidir (Gadelha, vd. 2017).



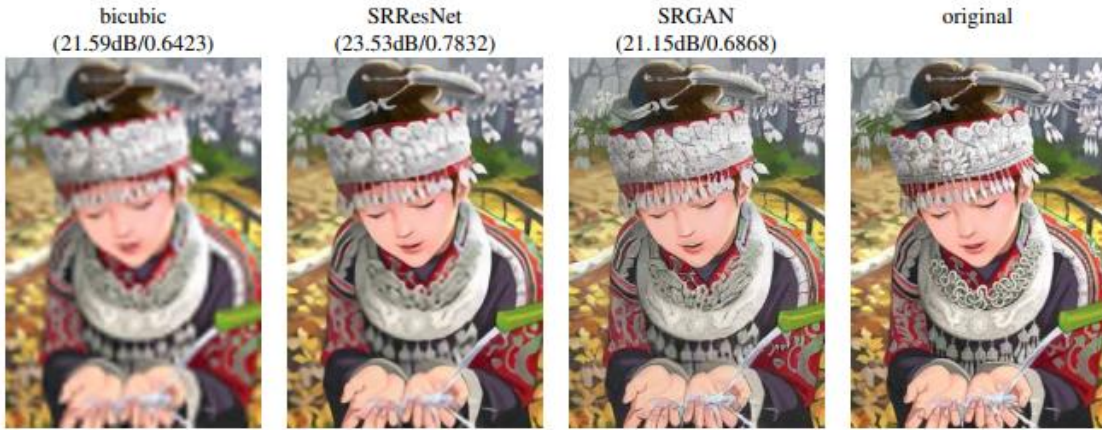
Şekil 2.11. PrGAN tarafından oluşturulan çeşitli 3 boyutlu şekiller (Gadelha vd. 2017)

Ledig ve arkadaşları tarafından 2017 sunulan bir makalede, tek bir 2D görüntüsünün çözünürlüğünü artırmak için kullanılan ÇÜA tabanlı bir yaklaşımı tanıtmaktadır.

Bu çalışma, tek bir görüntünün süper çözünürlüğünü artırmak için daha hızlı ve daha derin evrişimli sinir ağlarının doğruluk ve hızdaki ilerlemelerine rağmen büyük ölçekli büyütme faktörlerinde daha ince doku detaylarını nasıl kurtarabileceğimizi ele almaktadır. Geleneksel olarak bu tür işlemlerde kullanılan optimizasyon tabanlı yöntemler, amaç fonksiyonunun seçimine bağlı olarak farklı davranabilir. Son

zamanlarda yapılan arařtırmalar, ortalama kare rekonstrüksiyon hatasını minimize etmeye odaklanmıřtır. Ancak bu yaklařımla elde edilen tahminler, yüksek piksel sinyal-gürültü oranına sahip olabilir ancak yüksek frekansta detayları eksik olabilir ve beklenen sadakati daha yüksek çözünürlükte karşılayamayabilir.

Makale, SRGAN (Super-Resolution Generative Adversarial Network) adı verilen bir ÇÜA modelini tanıtmaktadır. Bu model, 4x büyütme faktörleri için doğal görüntülerin tahminini yapabilen ilk ‘‘framework’’tür. Bu hedefe ulaşmak için makale bir algısal kayıp fonksiyonu önermektedir. Bu fonksiyon bir düşmanca kayıp ve bir içerik kaybından oluşur. Düşmanca kayıp, bir çözünürlük artırma işlemi sonucunda elde edilen görüntüleri orijinal foto-realistik görüntülerden ayırt etmek için eğitilen bir ayırım ağı kullanır. Ayrıca piksel uzayında benzerlik yerine algısal benzerliğe dayanan bir içerik kaybı kullanılır. Derin geriye dönüş ağıları, kamu verileri üzerinde aşırı örnekleştirilmiş görüntülerden foto-gerçekçi dokuları geri kazanabilir. Kapsamlı bir ortalama görüş skoru (MOS) testi, SRGAN'ın algısal kalite açısından büyük kazançlar sağladığını göstermektedir. SRGAN ile elde edilen MOS skorları, orijinal yüksek çözünürlüklü görüntülerin skorlarına daha yakındır ve diğer en son teknoloji yöntemleri ile elde edilen skorlardan daha yüksektir (Ledig vd. 2017).



Şekil 2.12. Çözünürlük yükseltme çalışmasından birkaç örnek (Ledig vd. 2017)

Bin ve arkadaşları tarafından 2017 yılında yayınlanan bir makalede, yüksek kaliteli yüz görüntüsü süper çözünürlüğü için koşullu Çekişmeli Üretici Ağları (Conditional Generative Adversarial Networks - CGAN) kullanarak yeni bir yöntemi tanıtmaktadır (Bin, vd. 2017).

Bu çalışmada, "Face Conditional Generative Adversarial Network" (FCGAN) adlı bir yöntem önerilmektedir. Bu yöntem, Boundary Equilibrium Generative Adversarial Networks temel alınarak geliştirilmiştir. Yöntem, yüz öncesi bilgi kullanmadan düşük çözünürlüklü bir yüz görüntüsünden yüksek çözünürlüklü bir yüz görüntüsü üretebilmektedir. Eğitim ve test aşamaları, özellikle ön/son işlem gerektirmeyen uçtan uca bir iş akışı şeklinde tasarlanmıştır. Ayrıca üretici ve ayırt edici ağlarda özellik iletimini güçlendirmek ve yakınsama hızını artırmak için skip-layer bağlantıları kullanılmıştır. Yapılan kapsamlı deneyler, bu modelin en son teknoloji modelleriyle rekabet edebilecek performansı elde ettiğini göstermektedir (Bin, vd. 2017)



Şekil 2.13. CGAN kullanılarak kalitesi 4 kat artırılmış görüntüler (Bin, vd. 2017)

Isola ve diğerleri tarafından yayınlanan makalede, görüntüden görüntüye çeviri problemlerine genel bir çözüm olarak koşullu rakip ağları (Conditional Adversarial Networks) üzerinde bir araştırmayı ele almaktadır. Bu ağlar, sadece giriş görüntüsünden çıkış görüntüsüne olan eşlemeyi öğrenmekle kalmaz, aynı zamanda bu eşlemeyi eğitmek için bir kayıp fonksiyonu öğrenirler. Bu, geleneksel olarak çok farklı kayıp formülasyonları gerektiren sorunlara aynı genel yaklaşımın uygulanabilmesini mümkün kılar. Makale, bu yaklaşımın etiket haritalarından fotoğraflar sentezlemek, kenar haritalarından nesnelere rekonstrüksiyon yapmak, renklendirme gibi görevlerde etkili olduğunu göstermektedir. Ayrıca bu makale ile ilişkilendirilen pix2pix yazılımının

yayınlanmasından bu yana yüzlerce twitter kullanıcısının kendi sanatsal deneylerini paylaştığını belirtir. Bu çalışma, artık eşleme fonksiyonlarını el ile tasarlamadığımızı ve bu çalışmanın, kayıp fonksiyonlarını el ile tasarlamadan da makul sonuçlar elde edebileceğimizi önermektedir (Isola, vd. 2017).



Şekil 2.14. Modelin çeşitli görevlere ilişkin sonuçları (Isola, vd. 2017)

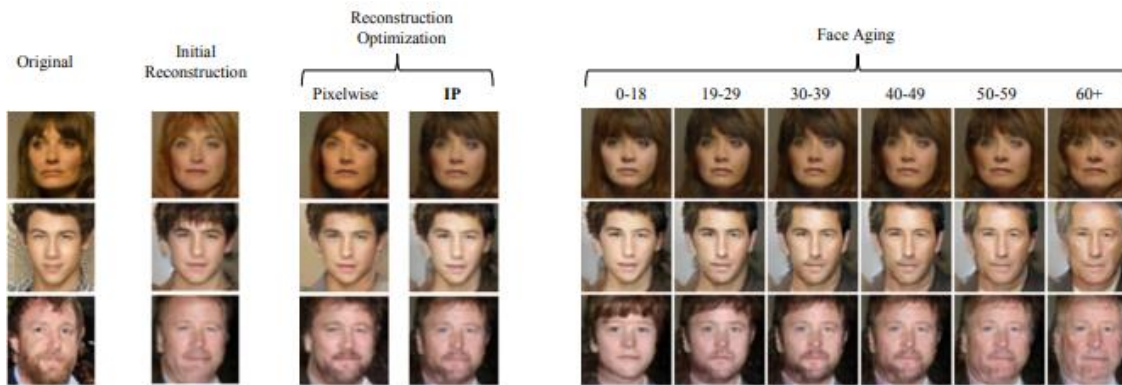
Huang ve arkadaşları tarafından yazılan makalede, tek bir yüz görüntüsünden gerçekçi ve kimlik koruyan ön görünüm sentezlemesi yapma konusundaki bir çalışmayı ele almaktadır (Huang, vd. 2017). Bu tür sentezlemeler, yüz tanıma alanında geniş bir uygulama yelpazesine sahiptir. Derin öğrenme yöntemleri, yüz verilerinden çözüm arayan veri tabanlı yöntemler önermiş olmasına rağmen bu sorun temelde yanıltıcıdır ve zorlu bir sorundur. Bu makale, fotorealistik ön görünüm sentezi için "Two-Pathway Generative Adversarial Network (TP-GAN)" adlı bir yaklaşım önermektedir. Aynı anda global yapıları ve yerel detayları algılayarak. Yerel dokulara odaklanmak için genellikle kullanılan global kodlayıcı-çözücü ağına ek olarak, dört farklı noktada bulunan yama ağları önerilmiştir. Yeni mimari, bu yanıltıcı sorunu rekabetçi kayıp, simetri kaybı ve kimlik koruma kaybının bir kombinasyonunu tanıtarak iyi kısıtlanmış hale getirmektedir. Bu birleşik kayıp fonksiyonu, ön yüz dağılımını ve önceden eğitilmiş ayırım yapıcı derin yüz modellerini kullanarak profillerden ön görünüm için kimlik koruyan sonuçlar elde etmek için kullanılır. Bu yöntem, önceki derin öğrenme yöntemlerinden farklı olarak, tanıma gibi yüz tanıma ve atıf tahmini gibi aşağı akış görevler için kimlik koruyan sentezlenmiş görüntüyü doğrudan kullanır. Deneysel sonuçlar, yöntemin hem etkileyici

algısal sonuçlar sunduğunu hem de büyük pozlar üzerinde yüz tanıma alanında en iyi sonuçları gösterdiğini göstermektedir (Huang, vd. 2017).



Şekil 2.15. Farklı pozlar altında TP-GAN ile sentez sonuçları (Huang, vd. 2017)

Antipov ve arkadaşları tarafından 2017 yılında yayınlanan makalede, Çekişmeli Üretici Ağlar (ÇÜA) kullanarak otomatik yüz yaşlandırma yöntemini sunan ilk çalışmayı ele almaktadır. Son zamanlarda ÇÜA'ların olağanüstü görsel sadakati olan sentetik görüntüler üretebildiği gösterilmiştir. Bu çalışmada, yüz özelliklerini değiştirmek için ÇÜA'ları kullanan önceki çalışmalardan farklı olarak, yaşlanmış bir kişinin yüzünün orijinal kişinin kimliğini korumasına özel bir vurgu yapılmaktadır. Bu amaçla, ÇÜA'ların latent vektörlerinin "kimlik koruma" optimizasyonu için yeni bir yaklaşım sunulmaktadır. Elde edilen yaşlı ve gençleştirilmiş yüz görüntülerinin, en iyi yüz tanıma ve yaş tahmini çözümleri tarafından objektif olarak değerlendirilmesi, önerilen yöntemin yüksek potansiyelini göstermektedir (Antipov, vd. 2017).



Şekil 2.16. Age-cGAN ile yüzün yeniden yapılandırılması ve yaşlanma örnekleri (Antipov, vd. 2017)

Zhang ve arkadaşları tarafından yayınlanan makalede, yüz yaşlandırma ve gençleştirme konularında bir yöntemi ele almaktadır.

Makale, geleneksel olarak yaşlandırma veya gençleştirme işlemi için gereken eşleştirilmiş örnekler ve etiketlenmiş sorgu görüntüsü olmaksızın, veri madenciliği yoluyla büyük bir yüz verisi kümesini kullanarak bir kişinin yüzünün nasıl görüneceğini

tahmin etme sorununu ele almaktadır. Bu çalışma, bir kişinin yaşlandırılmış veya gençleştirilmiş bir sürümünün, bu kişinin gerçek yaşını bilmeksizin nasıl oluşturulabileceğini araştırmaktadır.

Bu çalışmada, Koşullu Düşmanlı Otomatik Kodlayıcı (Conditional Adversarial Autoencoder) adlı bir derin öğrenme modeli kullanılarak bu tür dönüşümleri öğrenmek amaçlanmaktadır. Bu model, yüz yaşlandırma ve gençleştirme işlemlerinde kullanılmak üzere tasarlanmıştır. Çalışmanın sonuçları, yaşlandırma ve gençleştirme işlemlerinin yüksek potansiyele sahip olduğunu göstermektedir.

Bu makale, yüz yaşlandırma ve gençleştirme alanındaki derin öğrenme ve yapay zekâ çalışmalarına katkıda bulunmaktadır (Zhang, vd. 2017).

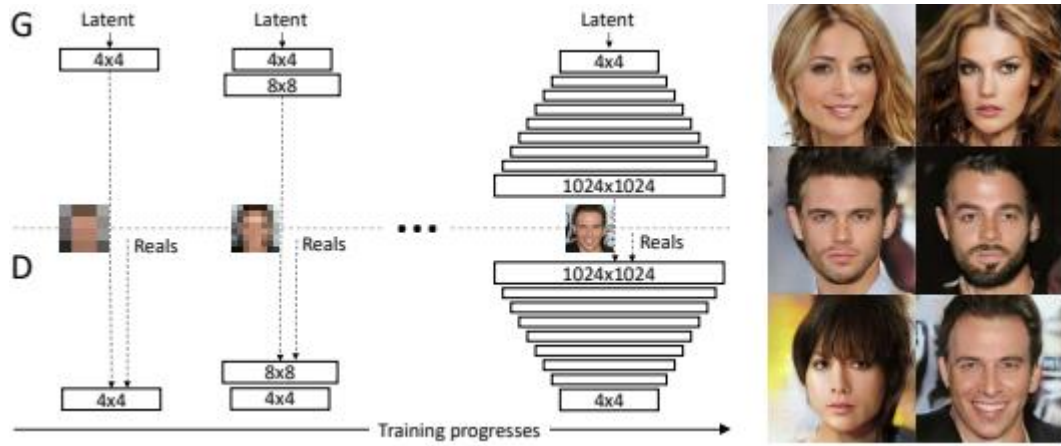


Şekil 2.17. Yüz yaşlanmasına örnekleri (Zhang, vd. 2017)

Karras ve arkadaşları tarafından yazılan makalede, ÇÜA'lar için yeni bir eğitim yöntemini açıklamaktadır.

Çalışmanın temel fikri, üreticiyi ve ayırt ediciyi aşamalı bir şekilde büyütmeektir. Düşük çözünürlükten başlayarak, eğitim ilerledikçe giderek daha ince detayları modelleyen yeni katmanlar eklenmektedir. Bu yöntem, eğitimi hızlandırır ve büyük ölçüde istikrarlı hale getirir. Örneğin 10242 çözünürlüğünde CelebA görüntülerinin üretilmesine olanak tanır. Ayrıca üretilen görüntülerin çeşitliliğini artırmak için basit bir yöntem önerilmiş ve denetimsiz CIFAR10 için kayda değer bir inception skoru olan 8.80 elde edilmiştir. Üstelik, üretici ve ayırt edici arasındaki sağlıksız rekabeti önlemek için birkaç uygulama ayrıntısı açıklanmaktadır. Son olarak, ÇÜA sonuçlarını hem görüntü kalitesi hem de çeşitlilik açısından değerlendirmek için yeni bir ölçüt önerilmektedir. Ayrıca CelebA veri kümesinin daha yüksek kaliteli bir sürümü de oluşturulmaktadır (Karras, vd. 2017).

Bu çalışma, ÇÜA'ların eğitimini geliştirmek ve yaratılan görüntülerin kalitesini, istikrarını ve çeşitliliğini artırmak için yeni bir yaklaşım sunmaktadır.



Şekil 2.18. Model ile 4x4 piksellik bir görüntüden 1024x1024 piksellik bir oluşturma (Karras, vd. 2017)

Jin ve arkadaşları tarafından yazılan makalede, ÇÜA kullanarak anime karakterlerinin otomatik olarak yaratılmasına odaklanmaktadır.

Çalışma, yüz görüntülerinin otomatik olarak oluşturulmasına ilişkin araştırmaların ÇÜA'nın ortaya çıkmasından sonra gerçekleştirildiğini belirtmektedir. Ancak anime karakterlerinin yüz görüntülerini üretme denemelerinin başarılı olmadığı ifade edilir. Bu nedenle araştırmacılar, anime yüz görüntülerine özgü ÇÜA modellerinin eğitimini incelemeye karar verirler. Bu inceleme veri ve model açısından gerçekleştirilir. Daha temiz ve uygun bir veri kümesi toplanır ve DRAGAN'ın uygun ve deneysel uygulanması sağlanır. Nicel analizler ve örnek olay incelemeleri, çalışmanın istikrarlı ve yüksek kaliteli bir model oluşturduğunu gösterir (Jin vd. 2017). Ayrıca anime karakter

tasarımına yardımcı olmak amacıyla, eğitilmiş modeli çevrimiçi olarak kullanılabilir hale getiren bir web sitesi (<http://make.girls.moe>) oluşturulur. Böylece modeli genel kullanıcılara kolayca erişilebilir kılarlar.

Bu çalışma, ÇÜA'ların anime karakterleri yaratma potansiyelini incelemekte ve anime karakter tasarımı alanında kullanılabilecek bir model geliştirmektedir. Ayrıca eğitilen modelin genel halk tarafından erişilebilir olduğu bir web sitesi oluşturarak bu alandaki çalışmalara katkı sağlamaktadır.



Şekil 2.19. Sabit gürültü kısmı ve rastgele niteliklerle oluşturulan görüntüler (Jin, vd. 2017)

Wang ve arkadaşları tarafından yazılan makalede, belirli Koşullu ÇÜA kullanarak çözünürlüklü foto-realistik görüntülerin sentezlenmesini ve semantik manipülasyonunu ele almaktadır (Wang, vd. 2018).

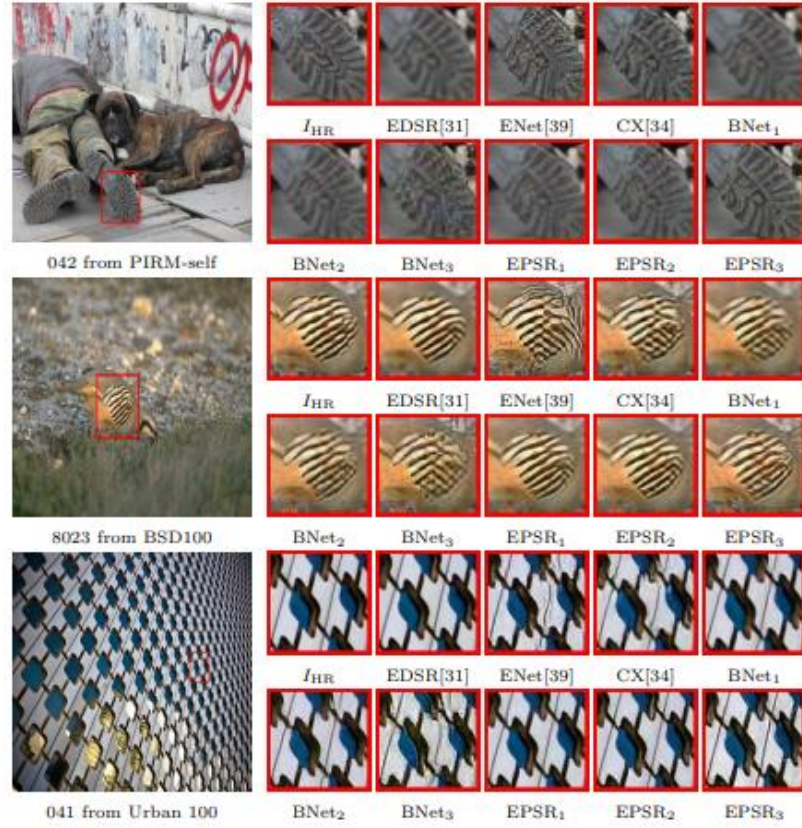
Makale, koşullu GAN türü kullanımıyla çeşitli uygulamaların gerçekleştirilebileceğini ancak elde edilen sonuçların genellikle düşük çözünürlükte ve gerçekçilikten uzak olduğunu belirtmektedir. Bu çalışma, yeni bir rekabetçi kayıp işlevi, çok ölçekli üreteç ve ayırt edici mimariler ile 2048x1024 çözünürlükte görsel olarak çekici sonuçlar üretmektedir. Ayrıca çerçeveyi iki ek özellikle etkileşimli görsel manipülasyona genişletmektedir. İlk olarak nesne örneği bölümlenme bilgisini içerir. Bu da nesnelere kaldırma/ekleme ve nesne kategorisini değiştirme gibi nesne manipülasyonlarını mümkün kılar. İkinci olarak aynı giriş verildiğinde farklı sonuçlar üretme yöntemini önerir. Bu da kullanıcıların nesne görünümünü etkileşimli olarak düzenlemelerine olanak tanır. İnsan görüşü çalışmaları, bu yöntemin derin görüntü

sentezi ve düzenleme kalitesini ve çözünürlüğünü önemli ölçüde geliştirdiğini göstermektedir (Wang, vd. 2018).

Vasu ve arkadaşları tarafından yazılan makalede, algısal süper çözünürlük ağı kullanarak görüntü süper çözünürlüğü (SR) için algısal kalite ve bozulma arasındaki dengeyi analiz etmeyi incelemektedir.

Makale, son dönemlerde Derin Evrişimli Sinir Ağı (DESA) tabanlı yöntemlerin görüntü süper çözünürlüğünde büyük başarılar elde ettiğini vurgulamaktadır. Ancak çoğu DESA tabanlı SR modelinin, bozulma ölçülerini (örneğin PSNR, SSIM, IFC, VIF) iyileştirmeye çalışırken algısal kaliteyi (örneğin insan görüş skoru, NIQE gibi referanssız kalite ölçüleri) düşürme eğiliminde olduğunu belirtir. Çalışmanın ana odak noktası, bozulma ve algısal kalite arasındaki dengeyi ve her zaman bir takas olduğunu vurgulamaktır. Genellikle algısal kalite açısından üstün olan restorasyon algoritmaları, bozulma ölçülerinden yetersiz olabilmektedir. Bu çalışma, tek bir görüntü SR problemi için bozulma ve algısal kalite arasındaki dengeyi analiz etmeye çalışmaktadır (Vasu, vd. 2018).

Araştırmacılar, geliştirilmiş algısal süper çözünürlük ağı (Enhanced Deep Super-Resolution - EDSR) mimarisini kullanarak bir çözüm önermektedir. Bu ağa "Enhanced Perceptual Super-Resolution Network" (EPSR) adı verilir ve belirli bir bozulma ölçüsü aralığında daha iyi algısal kalite elde etmek için uyarlanabilir. EDSR'nin orijinal ağı, yalnızca piksel başına doğruluk temel alınarak hatayı minimuma indirmek için eğitilmiştir. EPSR ise EDSR'nin bir üreteç modülü olarak kullanıldığı bir Çekişmeli Üretici Ağ (ÇÜA) çerçevesinde eğitilmiştir. EPSR, ortalama kare hata kaybı, algısal kayıp ve düşmanca kayıp kombinasyonunu kullanarak eğitilir. Deneyler, EPSR'nin bozulma ve algısal kalite arasında en iyi dengeyi sağladığını göstermektedir (Vasu vd. 2018).



Şekil 2.20. Modelin diğer modellerle karşılaştırılması (Vasu, vd. 2018)

Zhang ve diğerleri tarafından yazılan makale, yoğun hava koşullarında çekilen görüntülerin kalitesini iyileştirmek için kullanılan bir yöntemi inceler. Özellikle yağmur veya kar altında çekilen görüntülerin kalitesi olumsuz etkilenebilir. Bu gibi bozuk görüntüler görsel sistemlerin performansını büyük ölçüde etkileyebilir. Bu nedenle tek bir görüntü üzerindeki yağmur damlalarını giderme sorunu önemlidir. Ancak bu sorunun doğasında bulanıklık olması, birçok zorluğu beraberinde getirir (Zhang, vd. 2019).

Makale, bu sorunu ele almak için son zamanlarda tanıtılan cGAN güçlü üretici modelleme yeteneklerini kullanmayı amaçlar. Temel olarak bu yöntem, yağmur damlalarını giderilmiş bir görüntünün karşılık gelen gerçek temiz görüntüsünden ayırt edilemez olması gerektiği bir kısıtlama uygular. Bu, cGAN'dan gelen karşıtlık kaybının ek bir düzenleme sağlamasına ve üstün sonuçlar elde etmesine yardımcı olur (Zhang, vd. 2019).

Ayrıca makalede yeni bir rafine kayıp işlevi ve üretici-diskrimine çiftinde mimari yenilikler sunulur. Bu kayıp işlevi, cGAN'lar tarafından tanıtılan hataları azaltmayı ve daha iyi görsel kaliteyi sağlamayı amaçlar. Üreteç alt ağı, son zamanlarda tanıtılan yoğun bağlantılı ağlar kullanılarak oluşturulurken; Ayırıcı ağ, bir görüntünün gerçek veya makul

olduđuna karar vermek için global ve yerel bilgileri kullanacak şekilde tasarlanmıřtır (Zhang, vd. 2019).

Makale, "image de-raining conditional generative adversarial network (ID-cGAN)" adını verdikleri yeni bir tek görüntü de-rain yöntemi sunar. Bu yöntem nesnel, görsel ve ayrıca ayırt edici performansı objektif fonksiyona dahil eder. Sentezlenen ve gerçek görüntüler üzerinde yapılan deneysel sonuçlar, bu yöntemin yağmur nedeniyle bozulmuş görüntüler üzerinde objektif ve görsel performans açısından birçok son dönem tek görüntü de-rain yöntemini ařtıđını göstermektedir (Zhang, vd. 2019).



Şekil 2.21. Yađmurlu bir görüntünün belirtilen yöntemle temizlenmiş bir örneđi (Zhang, vd. 2019)

Brock ve diđerleri tarafından yazılan makale, yüksek kaliteli dođal görüntü sentezine odaklanan büyük ölçekli bir cGAN eğitimi üzerinde durmaktadır. Bu çalışma, ICLR 2019'da sunulan bir arařtırmadır ve ImageNet gibi büyük ve karmařık veri kümelerinden yüksek çözünürlüklü ve çeřitli örneklerin başarılı bir şekilde üretilmesini hedeflemektedir.

Makale, büyük ölçekli eğitimde karşılaşılan özgün zorlukları ele almaktadır. Arařtırmacılar, üreticiye dikey (orthogonal) düzenlemeler uygulayarak örnekleme sadakati ve çeřitliliđi arasındaki dengeyi basit bir "kırpma hilesi" ile yönetmeyi başarmıřlardır (Brock, vd. 2018). Bu, üreticinin girişinin varyansını azaltarak örnek sadakati ve çeřitliliđi arasında hassas bir denge oluşturur. Bu yapılan deđişiklikler, sınıf koşullu görüntü sentezi alanında yeni bir başarı seviyesi belirlemiřtir (Brock, vd. 2018).



Şekil 2.22. BigGAN ile üretilmiş örnekler (Brock, vd. 2018)

3. MATERYAL VE YÖNTEM

3.1 Veri Seti

Görüntü üretme algoritmaları gerçekçi ve yeni görüntüler oluşturmayı hedefleyen tekniklerdir. Bu algoritmaların eğitimi ve geliştirilmesi için genellikle büyük miktarda görüntü örneği kullanılır. Bu görüntü örnekleri veri seti adı verilen bir görüntü koleksiyonundan oluşur. Veri setleri gerçek dünyadan elde edilen veya sentetik olarak oluşturulan görüntülerden oluşabilir. Bu veri setleri, genellikle binlerce veya yüzbinlerce görüntü içerebilir ve farklı konuları veya nesnelere temsil edebilir. Örneğin orman manzaraları, plaj görüntüleri, hayvanlar veya insanlar gibi çeşitli doğal sahneler içerebilir.

Veri setleri genellikle etiketlenmiş veya etiketsiz olarak olabilir. Etiketlenmiş veri setlerinde görüntülerle ilişkili etiketler veya açıklamalar bulunur. Örneğin bir görüntünün içerdiği nesnelere türünü veya sahnede bulunan öğeleri belirten etiketler olabilir. Etiketsiz veri setlerinde ise görüntülerin kendisiyle ilgili herhangi bir ayrıntılı bilgi bulunmaz ve algoritmanın keşif yapmasına olanak tanır.

Görüntü üretme algoritmalarının başarısı kullanılan veri setinin boyutu, çeşitliliği ve kalitesine bağlıdır. Daha büyük ve çeşitli bir veri seti, algoritmanın genel ve gerçekçi sonuçlar üretme yeteneğini artırabilir.

ÇÜA'ların temel amacı gerçekçi görünen yeni veriler üretmektir. Özellikle görüntü, metin veya ses gibi doğal veri tipleriyle çalışıldığında kullanışlıdır. Bu tür algoritmalar ayrıca veri artırma, resim restorasyonu, görüntü sentezi, stil transferi, görüntü-toplama, metin yaratma, öneri sistemleri gibi alanlarda da kullanılabilir. Bu uygulamalar yaratıcı ve inovatif amaçlarla gerçekçi veri üretimini sağlar.

Bu çalışmada gerçekçi görüntü üretimi üzerine odaklanılmıştır ve dört farklı veri seti kullanılmıştır. Bu veri setleri aşağıda açıklanmıştır.

3.1.1 Cartoonset10k Veri Seti

Google LLC tarafından lisanslanan Cartoon Set 450MB boyutunda toplamada 10000 adet vektör sesimden oluşmaktadır. Veri setinde "etiketlenmiş" (saç rengi, göz rengi, ten rengi vs. açısından kategorize edilmiş) yüzler bulunmaktadır. Veri seti her bir yüz için bir ".png" uzantılı resim dosyası ve bir ".csv" dosyası bulunacak şekilde hazırlanmıştır. Resim dosyalarında yüzler yer alırken, .csv dosyalarında ise yüzlerle ilgili

etiketlenmiş veriler bulunmaktadır. Veri seti açık kaynak olarak <https://google.github.io/cartoonset/download.html> adresinde mevcuttur.

3.1.2 Anime Face Veri Seti

Anime Face veri seti 395MB boyutunda toplamda 63.565 adet yüksek kalitede .jpg uzantılı resimlerden oluşmaktadır. Kaggle platformunda elde edilen veri seti, “Mckinsey666 veri seti www.getchu.com” adresini kaynak göstermiştir.

Anime Face Veri Seti

<https://www.kaggle.com/datasets/splcher/animefacedataset> adresinde mevcuttur.

3.1.3 Animal Face Veri Seti

Animal Faces-HQ (AFHQ) olarak da bilinen bu veri seti, 512×512 çözünürlükte 16.130 adet yüksek kaliteli görüntüden oluşuyor. Toplam boyutu 750MB olan veri seti her biri yaklaşık 5000 görüntüden oluşan üç (3) sınıfa ayrılmıştır. Veri seti kedi, köpek ve yaban hayatı olmak üzere üç sınıfa ayrılmıştır.

3.1.4 CelebFaces Nitelikleri Veri Kümesi (CelebA)

CelebA veri kümesi yalnızca ticari olmayan araştırma amaçları için kullanılabilir. CelebA veri setinin tüm görselleri internetten elde edilmiştir.

CelebA Nitelikleri Veri Kümesi, 200.000'den fazla ünlünün resmini içeren büyük ölçekli bir yüz öznitelikleri veri kümesidir. Her bir resim, 40 özellik ek açıklamasına sahiptir. Bu veri kümesindeki görüntüler geniş poz çeşitliliği ve arka plandaki çeşitli unsurları içerir. CelebA büyük çeşitliliği, büyük miktardaki verileri ve zengin ek açıklamalarıyla dikkat çeken bir veri kümesidir. Adresten <https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html> ulaşılabilir.

3.2. Yapay Zekâ (Artificial Intelligence)

Bazı kaynaklarda yapay zekanın tarihini Harezmi'nin (780-850) “sıfır” (0) sayısını kullanması ile başladığını kabul eder. Bazı kaynaklar ise Cezeri (1136-1206) tarihleri arasında yaptığı otomatları kabul eder. Yapay zekâ modern anlamda ikinci dünya

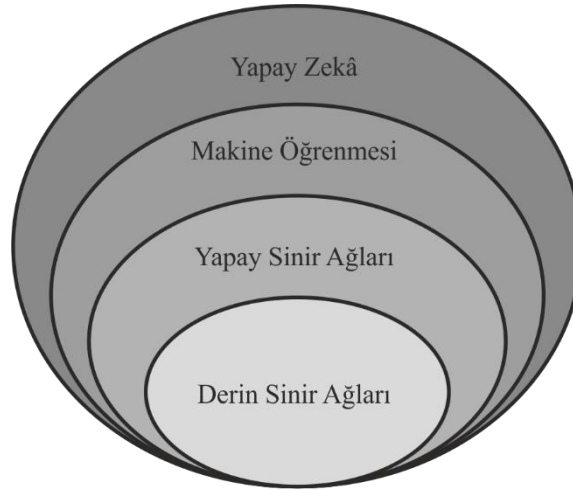
savaşından sonra önem kazanmaya başlamıştır (İrdem ve Çobanoğlu, 2021). Alan Mathison Turing, "Bombe" adını verdiği ilk otomatik kod kırma makinesini icat ederek savaşın kaderini etkilemiştir (Acar, 2020). 1947 yılında Alan Turing, ilk kez yapay zekâ ve bilgisayar programlarının birleştirilerek akıllı makinelerin geliştirilebileceğini açıklayan bir konferans vermiştir. Turing'e göre, bu iki teknolojinin birleştirilmesi akıllı makinelerin icat edilmesine olanak sağlayacaktı (McCarthy, 2007).

1950 yılında yayınlanan "Computing Machinery and Intelligence" makalesinde, Alan Turing, makine ve düşünmek kelimelerinin kombinasyonundan yola çıkarak "Makineler düşünebilir mi?" sorusunu tartışmıştır. Turing yapay zekanın düşünsel temellerini bu makalesinde oluşturmuştur (Turing, 1950).

Yapay zekâ terimi ise ilk kez 1956 yılında John McCarthy tarafından Dartmouth Koleji'nde düzenlenen bir çalıştayda kullanıldı. McCarthy çalıştayın amacını "yapay zekâ alanındaki araştırmaları başlatmak ve yönlendirmek" olarak tanımladı. Çalıştaya katılan bilim insanları yapay zekanın temellerini oluşturan teori ve uygulamaları tartıştılar (Arıcı, 2023).

Bu çalıştay yapay zekanın modern tarihinin başlangıcı olarak kabul edilir. Çalıştaydan sonra, yapay zekâ alanındaki araştırmalar hız kazandı ve yapay zekâ sistemleri günümüzde birçok alanda kullanılmaya başlandı.

Günümüzde yapay zekâ, insanlar tarafından zeki davranış olarak nitelendirilen faaliyetlerin makineler tarafından gerçekleştirilmesini inceleyen bir bilgisayar bilimleri dalıdır (Çeber, 2022). Yapay zekâyı tanımlayan literatürdeki tanımları incelersek, John McCarthy'ye göre yapay zekanın amacı "insan gibi davranan makineler geliştirmektir" (McCarthy, 2007). Nabyev'e göre ise yapay zekâ, "bir bilgisayarın ya da bilgisayar denetimli bir makinenin, genellikle insana özgü nitelikler olduğu varsayılan akıl yürütme, anlam çıkartma, genelleme ve geçmiş deneyimlerden öğrenme gibi yüksek zihinsel süreçleri gerçekleştirme yeteneği" olarak tanımlanmaktadır (Nabyev ve Zeka, 2016). Bu tanıma göre, yapay zekanın kullanım alanı, düşünsel yetenek gerektiren görevleri verilerle birlikte makinelere yaptırma amacını taşır. Yapay zekanın bazı alanları ve uygulamaları Şekil 3.1'de gösterilmiştir.



Şekil 3.1. Yapay zeka, makine öğrenmesi ve derin öğrenme ilişkisi

Russel ve Norvig'e göre ise yapay zekâ, akıllı varlıkları anlamak ve bunları taklit ederek karar verme sürecini basit, hızlı ve verimli hale getirmek için tasarlanmış bir mantık sistemidir (Russell ve Norvig, 2010).

Yapay zekâ, bilgisayar bilimi ve insan zekasını benzetmek ve geliştirmeyi amaçlayan bir teknolojidir (Bayuk ve Demir, 2019). Yapay zekânın temel amacı, bilgisayarların insan gibi düşünme ve problem çözme yeteneklerine sahip olmalarını sağlamaktır. Bu amaç doğrultusunda teoriler, yöntemler, algoritmalar ve uygulamalar geliştirilir (Gürlek vd. 2023).

Günümüzde modern yapay zekâ, insanların problemlerini çözmek için bilgisayara kurallar verdiği eski bir yaklaşımdan, insanların doğrudan bilgisayara problemleri verdiği ve makinelerin bunları bir dizi algoritma kullanarak kendi başlarına nasıl çözeceklerini öğrendikleri yeni bir yaklaşıma doğru ilerlemektedir. Algoritma, bir hesaplama makinesi tarafından gerçekleştirilecek bağımsız bir talimatlar ve eylemler dizisidir (Aslıhan ve Kılınç, 2020). Yapay zekânın nihai hedefi, hesaplama makinelerinin son derece akıllı bir şekilde çalışmasını sağlayacak teknolojiler geliştirmektir (Özdoğan, 2017).

Bu teknolojiler sayesinde yapay zekâ sistemleri insan zekasına benzer şekilde düşünme, öğrenme ve karar verme yetenekleri kazanabilir. Bu da insanların daha karmaşık problemlerin çözümünde ve daha verimli işlemlerin gerçekleştirilmesinde yapay zekâ sistemlerinden faydalanmalarını sağlar. Ancak yapay zekâ teknolojisi hala gelişmekte olan bir alandır ve insan zekasını tamamen taklit etmekten uzaktır (Sayar, 2021).

Yapay zekâ, bilgisayar bilimi ve insan zekasının birleşiminden doğan ve hesaplama makinelerinin insan benzeri zekâ seviyelerine ulaşmasını hedefleyen bir teknolojidir (Deng, 2018).

Günümüzde pek çok alanda kullanılan yapay zekâ, gelecekte hayatımızın birçok yönünü etkileyecek daha akıllı ve verimli sistemlerin geliştirilmesine yardımcı olacaktır. Yapay zekâ, insan zekasını taklit etmeyi amaçlayan bir teknoloji olduğu için birçok farklı yeteneği içerir. Bunlar arasında insan konuşmasını başarıyla anlama, stratejik oyun sistemlerinde üst düzeyde rekabet etme, kendi kendini süren arabalar, içerik dağıtım ağlarında akıllı yönlendirme, askeri benzetimler ve karmaşık verileri yorumlama gibi konular yer alır (Ongsulee, 2017).

Akıllı bir sistemin özellikleri:

- Uyarlanabilir hedefe yönelik davranışlar gösterme,
- Deneyimlerden öğrenme,
- Büyük miktarda bilginin kullanılması,
- Öz farkındalık gösterme,
- Dil ve konuşmayı kullanarak insanlarla etkileşim kurma,
- İletişimde hata ve belirsizliği giderme
- Gerçek zamanlı yanıt verme gibi özellikleri içerir (Reddy, 1996).

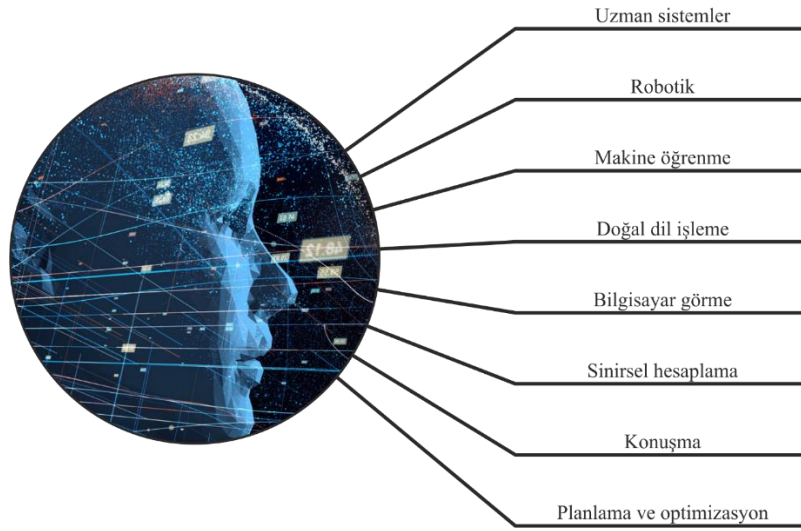
Yapay zekâ, uzman sistemler, robotik, makine öğrenme, doğal dil işleme, bilgisayar görme, sinirsel hesaplama, konuşma, planlama ve optimizasyon gibi birçok alanda ve uygulamada kullanılmaktadır. Akıllı sistemler, insanların daha verimli ve güvenli bir şekilde çalışmasını sağlamak için tasarlanmıştır. Otomasyon, veri analizi ve tahmin gibi alanlarda yapay zekâ teknolojilerinden yararlanır.

Yapay zekanın kullanıldığı bazı alanlar ve uygulamalar şunlardır:

- **Uzman sistemler:** Belirli bir konuda uzmanlık sahibi olan bir sistemdir.
- **Robotik:** Yapay zekâ teknolojileri, robotların insan benzeri davranışlar sergilemesine ve insanların yardımı olmadan karmaşık görevleri gerçekleştirmesine olanak sağlar.
- **Makine öğrenme:** Bilgisayarların verilerden öğrenmesi ve kendini geliştirmesi için kullanılan bir yöntemdir. Örneğin bir makinenin resimleri tanıması ve sınıflandırması için kullanılabilir.
- **Doğal dil işleme:** Bilgisayarların insan diliyle etkileşime girebilmesini sağlayan bir teknolojidir. Bu, dijital asistanlar ve dil çevirisi gibi uygulamalarda kullanılabilir.

- **Bilgisayar görme:** Bilgisayarların görüntüleri analiz etmesini ve nesnelere tanımlamasını sağlayan bir teknolojidir. Örneğin güvenlik kameralarında yüz tanıma gibi uygulamalarda kullanılabilir.
- **Sinirsel hesaplama:** Beyindeki sinir ağlarını taklit ederek hesaplama yapmayı amaçlayan bir teknolojidir. Özellikle büyük veri analizinde kullanılır.
- **Konuşma:** Bilgisayarların insan sesini algılayarak anlamasını ve insanlarla etkileşim kurmasını sağlar. Sesli komutlara cevap veren dijital asistanlar bunun bir örneğidir.
- **Planlama ve optimizasyon:** Yapay zekâ, bir problemi en iyi şekilde çözmek için en uygun planı oluşturabilir ve uygulayabilir.

Yapay zekanın kullanıldığı alanlar Şekil 3.2’de gösterilmiştir.



Şekil 3.2. Yapay zekâ alanları

3.3. Makine Öğrenmesi (Machine Learning)

Makine öğrenimi, bilgisayarın geçmiş deneyimlerinden elde ettiği bilgi ve tecrübeleri kullanarak bir model oluşturmasını ifade eden bir yapay zekâ alanıdır (Aslıhan ve Kılınç, 2020). Bu model gelecekte meydana gelebilecek benzer olaylar hakkında kararlar verebilme yeteneği kazandırarak problemlere çözüm üretebilir. Başka bir deyişle makine öğrenimi bilgisayarın bir olayla ilgili bilgileri ve tecrübeleri öğrenerek gelecekte benzer olaylar hakkında kararlar almasına ve problemlere çözümler üretmesine imkân tanır (Pulat ve Kocakoç, 2021).

Makine öğrenimi kavramı, bilgisayar oyunları ve yapay zekâ alanında öncü isimlerden biri olan Arthur Lee Samuel tarafından 1959 yılında, bilgisayarların yeniden

programlamaya ihtiyaç duymadan görev yapmasını sağlayan bir bilim olarak tanımlanmıştır (Şimşek, 2022). Tom Mitchell ise 1977 yılında daha teknik bir mühendislik tanımını şu şekilde yapmıştır: "Görevin T, performansın P ile ifade edildiği durumda, eğer P ile ölçülen T performansı E deneyimi ile iyileşiyorsa ilgili programın E deneyiminden öğrendiği söylenir" (Mitchell, 1997).

Makine öğrenimi, bilgisayarın meydana gelen bir olayla ilgili bilgi ve deneyim kazanarak gelecekte benzer olaylar hakkında kararlar alabilmesi ve ortaya çıkabilecek problemlere çözümler üretebilmesi sürecidir (Öztemel, 2006). Makine öğrenimi, belirli yöntemleri kullanarak geçmiş veri setlerinden faydalanır ve yeni veri için en uygun modeli tespit etmeye çabalar (Keleş, vd. 2017).

Büyük miktardaki verinin manuel olarak işlenmesi ve analiz edilmesi oldukça zorlu bir süreçtir (İçöz, 2021). Bu bağlamda, geçmiş verilerin kullanılarak gelecekteki durumlar için tahminlerde bulunma amaçlanmaktadır. Uygulama alanı ne olursa olsun çok sayıda verinin analiz edilerek geleceğe dair tahminlerde bulunması ve karar alma süreçlerimize katkıda bulunması makine öğrenimi yöntemlerinin giderek daha fazla önem kazanmasına sağlamaktadır (Çekerol ve Kesler, 2021).

Makine Öğrenimi birçok alanda kullanılmaktadır:

- Doğal Dil İşleme,
- Konuşma ve El Yazısı Tanıma,
- Nesne Tanıma,
- Bilgisayar Oyunları,
- Robot Hareketleri,
- Arama Motorları ve Tıbbi Teşhis vb. (Kutlugün ve vd. 2017).

Makine öğrenmesi sürecinde üç temel aşama bulunmaktadır:

1. Dokümanların Hazırlanması,
2. Öğrenme Yöntemlerinin Belirlenmesi ve Uygulanması,
3. Öğrenme Performansının Değerlendirilmesi.

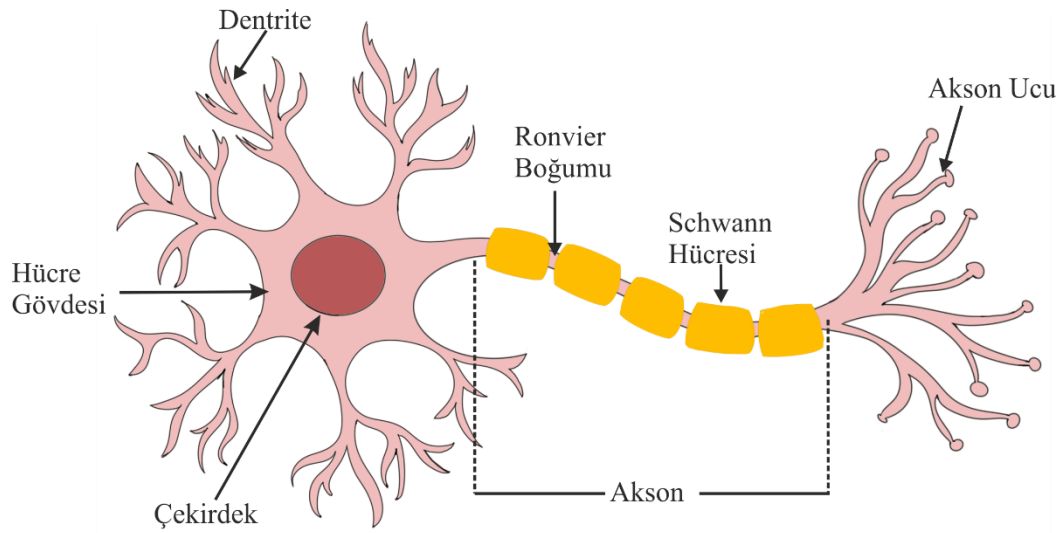
Makine öğrenmesinde, öncelikle öğrenme sürecinin gerçekleştirileceği veri kümesinin uygulanacak öğrenme yöntemine uygun bir şekilde hazırlanması gereklidir (Balt, 2018). Öğrenme metodolojisinde genellikle istatistiksel yöntemler kullanılmaktadır ve yeni geliştirilen metotlar da genellikle istatistiksel temellere dayanmaktadır. Yeni bir

yöntem geliştirildiğinde bu metodun performansı ölçülerek diğer metotlarla karşılaştırılması yapılmaktadır (Uzun, 2007).

Makine öğrenimi bilgisayar bilimi ve istatistik alanlarının birleşimiyle ortaya çıkan bir disiplindir. Makine öğrenimi, bir bilgisayarın verileri analiz edip örüntüleri tanımasına ve bu verilere dayalı olarak öğrenmesine izin veren bir yapay zekâ dalıdır. Makine öğrenimi, özellikle büyük veri setlerinde insanların manuel olarak yapamayacağı kadar karmaşık ve büyük ölçekte işlemler gerçekleştirmek için kullanılır. Temelde bir makine öğrenimi modeli verileri incelemek ve öğrenmek için algoritmalar ve istatistiksel teknikler kullanır (Kuzey, 2012).

3.4.Yapay Sinir Ağları (Artificial Neural Networks)

İnsan beyninin üstün özellikleri, bilim insanlarını çalışmaya teşvik etmiş ve beyin nörofiziksel yapısından ilham alarak matematiksel modeller çıkarmaya yönelik çabalar ortaya konmuştur. Beynin tüm davranışlarını tam olarak modelleyebilmek için fiziksel bileşenlerinin doğru bir şekilde modellenmesi gerektiği düşüncesiyle çeşitli yapay hücre ve ağ modelleri geliştirilmiştir. Şekil 3.3' te bir sinir hücresinin biyolojik görüntüsü verilmiştir.



Şekil 3.3. Bir Sinir hücresinin biyolojik gösterimi

Böylece Yapay Sinir Ağları denen yeni ve günümüz bilgisayarlarının algoritmik hesaplama yönteminden farklı bir bilim dalı ortaya çıkmıştır.

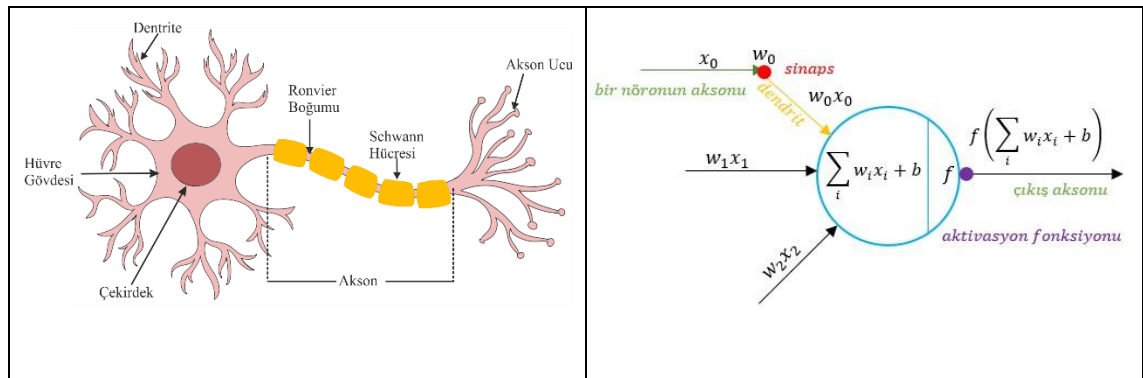
Yapay sinir ağı mimarisi alanında yapılan ilk çalışma, nörofizyolog Warren McCulloch ve matematikçi Walter Pitts tarafından 1943 yılında yayınlanan "Sinir

Etkinliğinde Bulunan Fikirlerin Mantıksal Bir Hesaplaması" isimli makalede sunulmuştur (Pitts ve McCulloch, 1943).

Pitts ve McCulloch, sinir bilimi ile matematik arasındaki etkileşimi temsil ederek yapay sinir ağlarının temel prensiplerini ilk kez ortaya koymuştur. Canlı beyinde yer alan biyolojik nöronların birlikte nasıl çalıştıklarının indirgenmiş bir modelini göstermişlerdir.

Pitts ve McCulloch'un çalışması, yapay sinir ağlarının gelişimi için bir dönüm noktası olmuş ve bu alandaki ilerlemelerin temelini atmıştır.

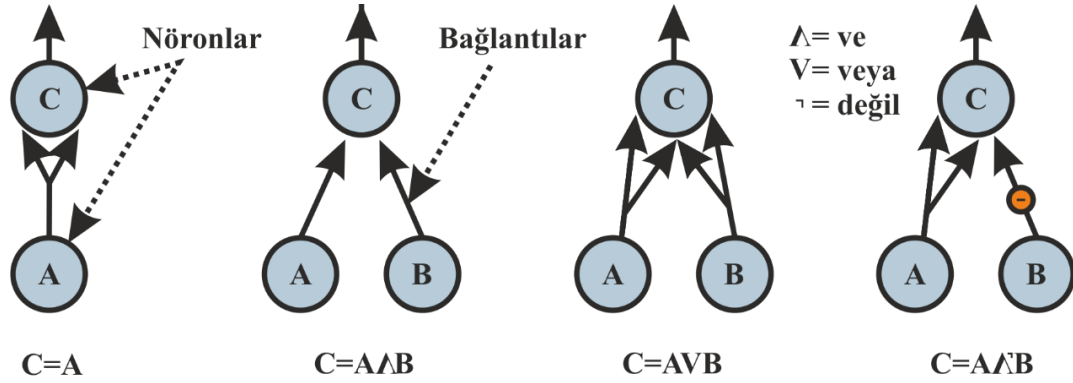
McCulloch, beyni modellemeye çalışırken Russell ve Whitehead'in "Principia Mathematica" adlı kitabından ilham almıştır (Sancakdar ve Kamer, 2021). Bu kitap bütün matematiğin temel mantık kullanılarak kurulabileceğini göstermeye çalışan bir çalışmadır. Principia Mathematica'da en temel parça önermedir ve mümkün olan en basit ifade doğru ya da yanlıştır. Russell ve Whitehead, bu basit önermelerden yola çıkarak modern matematiğin tüm karmaşıklığını türetmeye çalışmışlardır. Russell ve Whitehead'in bu temel mantık yaklaşımından ilham alarak sinir bilimi ve matematiği birleştirmeyi amaçlamıştır. Russell ve Whitehead'in temel mantıksal işlemleri, yani 've', 'veya' ve 'değil' McCulloch'un sinir ağı modelini oluşturmak için kullanılmıştır. McCulloch, bu modeli matematikçi olan Pitts'e aktarmış ve birlikte yaptıkları çalışmalarda, yapay sinir ağı modelini matematiksel bir ifadesini ortaya koymuşlardır (Pehlivan ve Güven, 2019). Şekil 3.4. de Bir sinir ağının biyolojik ve matematiksel karşılaştırması verilmiştir.



Şekil 3.4. Bir sinir ağının biyolojik ve matematiksel gösterimi

McCulloch ve Pitts, yapay sinir hücreleri üzerinden her türlü mantıksal ifadenin formülize edilebileceğini göstermişlerdir. McCulloch'un, Russell ve Whitehead'in mantık

temelli yaklaşımını sinir ağlarıyla birleştirmesi yapay sinir ağlarının gelişiminde kritik bir adımdır. Şekil 3.4. de Bir sinir ağının temel mantıksal işlemleri görsel olarak verilmiştir.

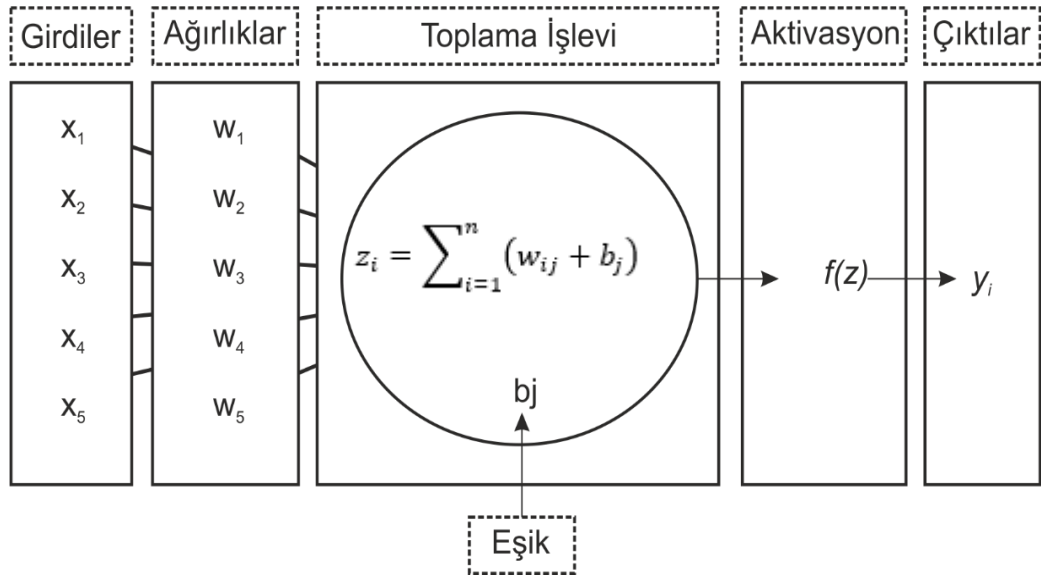


Şekil 3.5. Bir sinir ağının temel mantıksal önermeleri.

Bu sinir ağı mimarisini matematiksel bir temele oturtarak, bilgisayar bilimleri ve yapay zekâ alanındaki ilerlemelere önemli bir katkı sağlamıştır.

3.4.1 Yapay Sinir Hücresinde Yer Alan Elemanlar

Yapay sinir sistemleri, biyolojik sinir sisteminden ilham alarak tasarlanmıştır ve birbirleriyle bağlantılı bir yapıya sahiptirler (Peker, vd. 2018). Yapay sinir hücreleri -yani yapay nöronlar- yapay sinir ağlarının yapısını oluşturan temel birimlerdir. Bu yapay sinir hücreleri, işleme elemanları olarak kullanılır ve beş temel bileşene sahiptir (Uzun, 2007). Her bir yapay sinir hücresi, dışarıdan gelen bilgileri alan girdilere, bu bilgileri işlemek ve bağlantıları oluşturmak için kullanılan ağırlıklara, toplama fonksiyonuna, aktivasyon fonksiyonuna ve işlenen bilgileri dış dünyaya sunan çıktılara veya çıktı elemanlarına sahiptir (Yavuz, 2012). Bu bileşenler bir araya gelerek yapay sinir ağının işlevselliğini sağlarlar. Şekil 3.6. Yapay Sinir Hücresinin elemanları gösterilmiştir.



Şekil 3.6. Yapay sinir hücresi (Yavuz ve Deveci, 2012)

Yapay sinir hücrelerinin temel bileşenlerini açıklamak gerekirse:

Girdiler: Diğer hücrelerden veya dış dünyadan gelen bilgileri temsil eder (Tutcu, 2023). Bu girdiler yapay sinir hücresine çeşitli bilgileri iletebilir ve işlenmek üzere bekler.

Toplama Fonksiyonu: Toplama fonksiyonu, yapay sinir hücresine giren net girdiyi hesaplayan fonksiyondur (Terzi ve Mehmet, 2012). Bu fonksiyon genellikle gelen bilgilerin ağırlıklarla çarpılıp toplanması şeklinde gerçekleşir. Yapay sinir ağı modeline bağlı olarak farklı toplama fonksiyonları kullanılabilir.

Aktivasyon Fonksiyonu: Aktivasyon fonksiyonu girdi ve çıktı arasında bağlantı kurar. Toplama fonksiyonundan gelen bilgileri işleyerek çıktı bilgilerini oluşturur (Saplıoğlu ve Çimen, 2010). Aktivasyon fonksiyonu da yapay sinir ağı modeline göre değişiklik gösterebilir ve çeşitli fonksiyonlara sahip olabilir.

Çıktı: Aktivasyon fonksiyonu tarafından üretilen değerlerdir. Bu çıktılar yapay sinir hücresinin işlemler sonucunda elde ettiği sonuçları temsil eder (Çevik ve Dandil, 2012). Bu çıktılar diğer hücrelere veya dış dünyaya iletilerek bilginin aktarılmasını sağlar.

Bu bileşenler bir araya gelerek yapay sinir hücresi, yapay sinir ağının temel birimidir ve yapay sinir ağının işlevselliğini sağlar.

3.3.2 Yapay Sinir Ağı Modeli

Yapay Sinir Ağları (YSA) iki temel sinir ağı modeline sahiptir: tek katmanlı ve çok katmanlı sinir ağları.

a. Tek katmanlı sinir ağı;

Bu sinir ağı modelinde yapay sinir hücreleri tek bir katmanda yer alır. Girdiler, doğrudan bu katmandaki sinir hücrelerine iletilir ve çıktılar bu hücrelerden elde edilir. Tek katmanlı sinir ağları genellikle basit problemler için kullanılır ve daha az besleme gücü gerektirir (Öztemel, 2003).

b. Çok katmanlı sinir ağı;

Bu sinir ağı modelinde yapay sinir hücreleri birden fazla katmanda düzenlenmiştir. Girdiler ilk katmana iletilir ve ardışık katmanlardaki sinir hücrelerinden geçerek son katmandan çıktılar elde edilir. Her bir katmanda bulunan sinir hücreleri, gelen bilgileri işler ve ağırlıklar aracılığıyla bağlantılarını oluşturur. Çok katmanlı sinir ağları, karmaşık problemleri çözmek için daha fazla besleme gücü ve öğrenme kapasitesi sağlar (Ersoy ve Karal, 2012).

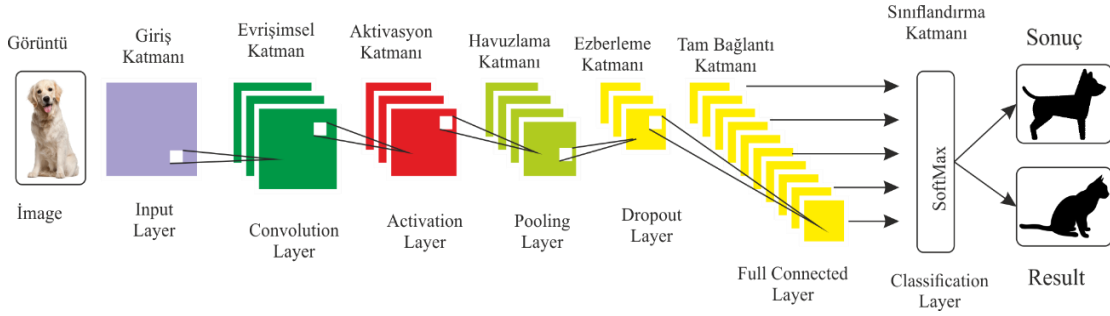
Bu iki sinir ağı modeli, farklı problem türleri ve karmaşıklık düzeyleri için kullanılabilir. Tek katmanlı sinir ağları daha basit problemlerde etkili olabilirken çok katmanlı sinir ağları daha karmaşık veri yapıları ve daha derin öğrenme için daha uygun olabilir (Ataseven, 2013).

3.5. Derin Öğrenme (Deep Learning)

Makine öğrenme algoritmalarının günümüzde ulaştığı en ileri nokta derin öğrenme algoritmalarıdır. Derin öğrenme, geleneksel öğrenme yöntemlerinden farklı olarak hiyerarşik bir yapıya sahiptir. Bu yöntemde, özneteliklerin otomatik olarak elde edilmesi için denetimli ve denetimsiz öğrenme yöntemleri kullanılabilir (Yu ve Deng, 2010). Derin öğrenme algoritmalarının genel mantığı konvolüsyon tabanlı bir yapay sinir ağı modeline dayanmaktadır.

Günümüzde donanım kapasitesinin artmasıyla birlikte, konvolüsyonel sinir ağının farklı kombinasyonlarıyla bir araya gelerek Şekil 3.7'de gösterilen derin öğrenme mimarisinin temeli oluşturur. Bu temel, katman ve bağlantı sayılarının değiştirilerek

farklı isimler altında yeni derin öğrenme mimarilerinin geliştirilmesine olanak sağlar. Derin öğrenme katmanları daha ayrıntılı olarak aşağıda verilmiştir.



Şekil 3.7. Klasik derin öğrenme mimarisi (Architecture of Deep Learning)

3.5.1 Giriş Katmanı

Giriş katmanı, birçok yapay zekâ algoritmasında olduğu gibi verilerin ağa sunulmadan önce standart bir formata dönüştürüldüğü katmandır. Bu katmanda genellikle kullanılan mimari ve veri setine bağlı olarak normalizasyon işlemi uygulanır. Normalizasyon işlemi Denklem 1 ile gerçekleştirilir:

$$x_{norm} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (1)$$

Denklem 1'deki x_{norm} normalize edilmiş veriyi, $\min(x)$ kullanılan veri seti içerisindeki en küçük değeri, $\max(x)$ veri seti içerisindeki en büyük değeri ifade eder. Burada x ise normalize edilecek veriyi temsil eder (Göçmen, 2022).

3.5.2 Konvolüsyon Katmanı

Konvolüsyon katmanı, derin öğrenme ve sinir ağı modellerinde sıkça kullanılan bir yapıdır. Bu katman girdi veri üzerinde filtreleme işlemi yaparak çıktı üretir. Genellikle görüntü işleme ve ses işleme gibi alanlarda kullanılır çünkü bu tür verilerde lokal özelliklerin tanınması önemlidir.

Konvolüsyon katmanı, bir filtre matrisi veya kernel kullanarak girdi veri üzerinde konvolüsyon işlemi uygular. Konvolüsyon, iki işaretin (girdi ve filtre) noktalarını çarparak toplamını almak ve sonucu çıktıya yerleştirmek anlamına gelir. Bu işlem, girdi

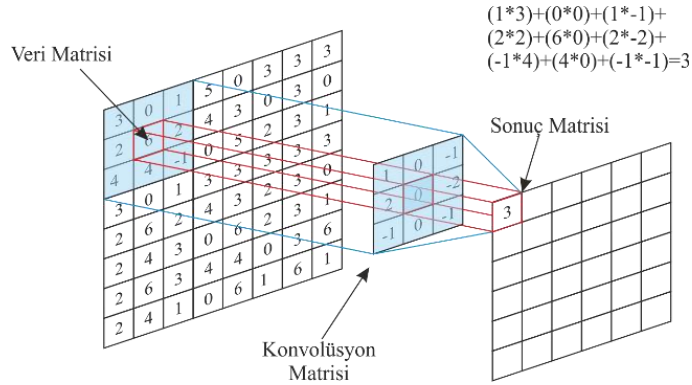
veri üzerinde filtreleyerek önemli özellikleri vurgular ve gereksiz ayrıntıları azaltır (Cihan, 2020).

Konvolüsyon katmanının önemi, verinin lokal bağımlılıklarını yakalamak ve özellik çıkarımını kolaylaştırmaktır. Örneğin bir görüntüdeki kenarları veya şekilleri tanımak için konvolüsyon filtreleri kullanılabilir. Bu sayede daha küçük ve daha basit özelliklerin bir araya gelerek daha karmaşık özelliklerin elde edilmesi sağlanır.

Ayrıca konvolüsyon katmanları ağırlıkların paylaşılması özelliğine sahiptir. Yani aynı filtre matrisi farklı bölümlerdeki veri üzerinde tekrar tekrar kullanılır. Bu modelin daha az parametre kullanarak daha etkili bir şekilde öğrenmesini sağlar ve aşırı uyum (overfitting) sorununu azaltır (Özkan ve Ülker, 2017).

Konvolüsyon katmanı veri üzerinde filtreleme işlemi yaparak önemli özelliklerin vurgulanmasını sağlar. Bu katman görüntü işleme, ses işleme ve diğer benzer alanlarda kullanılarak derin öğrenme modellerinin daha etkili ve verimli olmasını sağlar (Aktürk ve Serbest, 2022).

Şekil 3.8. de bir Konvolüsyon Katmanına filtre uygulanması gösterilmiştir.



Şekil 3.8. Konvolüsyon katmanı ve filtre uygulaması

Sonuç matrisini elde etmek için denklem 2’de yazılan denklem uygulanır.

$$(L * K)_{xy} = \sum_{i=1}^h \sum_{j=1}^w K_{ij} \cdot I_{X+i-1, Y+j-1} \quad (2)$$

Filtrenin görüntü üzerinde dolaştırılması ağırlık performansını doğrudan etkiler. Örneğin dolaştırılacak matris ağ üzerinde çok büyük boyutlarda ve büyük sayılabilecek

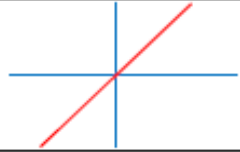
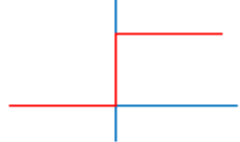


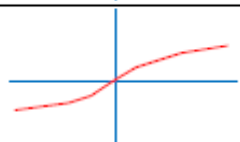


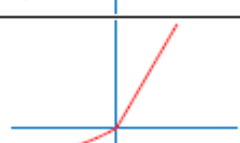
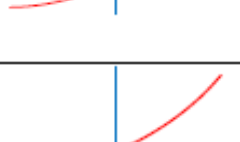
değerlerden oluşuyorsa ağı eğitimi çok uzun sürebilir ve hatta tercih edilen değerlere bağlı olarak hata riski ortaya çıkabilir (Pang vd. 2017).

3.5.3 Aktivasyon Katmanı

Aktivasyon katmanı, sinir ağı modellerinde kullanılan bir yapıdır ve genellikle konvolüsyon veya tamamen bağlı (fully connected) katmanlardan sonra kullanılır. Aktivasyon katmanı, bir fonksiyon aracılığıyla gelen sinyalleri değiştirerek ve sınırlayarak çıktı üretir. Bu katman, sinir ağının öğrenme kapasitesini arttırmak, modelin daha kompleks kararları almasını sağlamak ve doğrusal olmayan ilişkileri yakalamak için kullanılır (Doğan ve Türkoğlu, 2019).

Aktivasyon fonksiyonları aktivasyon katmanlarında kullanılan matematiksel fonksiyonlardır. Bu fonksiyonlar girdi sinyallerini dönüştürerek çıktı üretir. Yaygın olarak kullanılan bazı aktivasyon fonksiyonları ve denklemleri:

Tablo 3.2. Aktivasyon fonksiyonları

İsim	Grafik	Denklem $f(x)$	Türev $f'(x)$
Identity		$= x$	$= 1$
Binary Step		$= \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$= \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic		$= \frac{1}{1 + e^{-x}}$	$= f(x)(1 - f(x))$
TanH		$= \tanh(x)$ $= \frac{2}{1 + e^{-2x}} - 1$	$= 1 - f(x)^2$
ArcTan		$= \tan^{-1}(x)$	$= \frac{1}{x^2 + 1}$
ReLU		$= \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$= \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
PReLU		$= \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$= \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
ELU		$= \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$= \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$= \log_e(1 + e^x)$	$= \frac{1}{1 + e^{-x}}$

Aktivasyon fonksiyonları, sinir ağının çıktısını sınırlamak non-linearite (doğrusal olmayan ilişkiler) eklemek ve modelin daha kompleks örüntüleri öğrenmesini sağlamak için kullanılır. ReLU fonksiyonu negatif girdi değerlerini sıfır olarak sınırlarken pozitif girdi değerlerini doğrudan aktarır (Kuş, 2019). Bu şekilde ReLU aktivasyon fonksiyonu

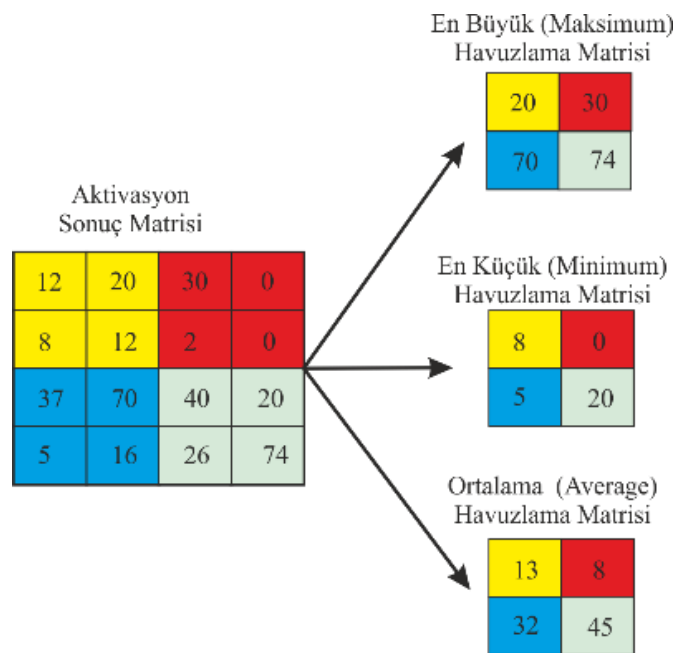
sinir ağının daha spesifik ve daha az hesaplama yoğun bir şekilde öğrenmesine yardımcı olur.

Aktivasyon katmanları sinir ağlarında büyük bir öneme sahiptir. Doğru aktivasyon fonksiyonunun seçimi modelin performansını etkileyebilir. Farklı aktivasyon fonksiyonları farklı senaryolarda tercih edilebilir ve modelin öğrenme sürecini optimize etmek için kullanılırlar (Saleh, 2021).

3.5.4 Havuzlama (Pooling) Katmanı

Literatürde birçok uygulamada genellikle aktivasyon katmanından sonra havuzlama katmanı kullanılmaktadır. Havuzlama katmanının temel amacı verileri daha küçük boyutlara indirmektir. Bu indirgeme işlemi sırasında kullanılan havuzlama fonksiyonlarına bağlı olarak veri kaybı yaşanabilir (Hinton, 2012). Bu nedenle havuzlama katmanı uygulandığında makul düzeyde veri kaybı beklenmelidir.

Literatürde birçok havuzlama fonksiyonu bulunmasına rağmen genellikle en büyük değer (max pooling), en küçük değer (min pooling) ve ortalama değer (average pooling) fonksiyonları kullanılmaktadır (Metlek ve Çetiner, 2021). Bu fonksiyonlar, aktivasyon katmanından elde edilen matris üzerinde dolaşarak yeni bir veri matrisi oluşturmak için kullanılır. Şekil 3.9'da gösterildiği gibi.



Şekil 3.9. Havuzlama katmanı (Metlek ve Kayaalp 2020)

3.5.5 Ezberleme (Dropout) Katmanı

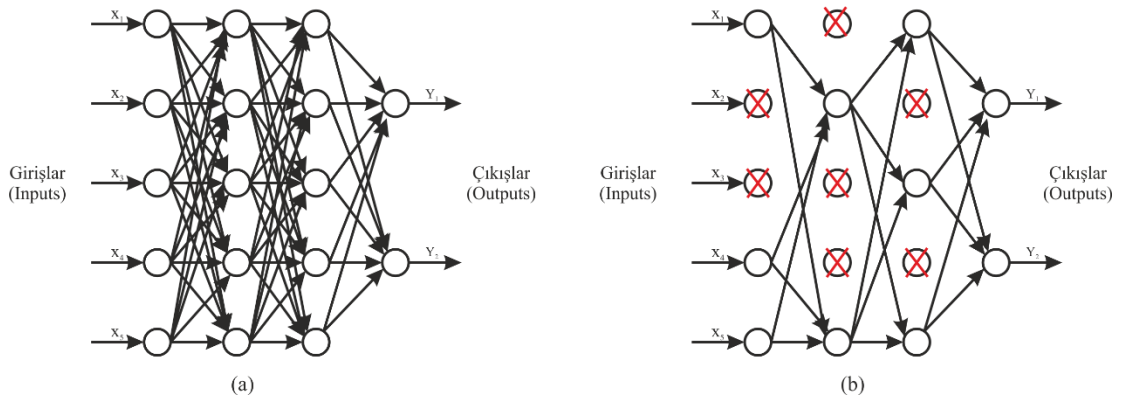
Ezberleme (Dropout) katmanı, sinir ağı eğitilirken aşırı öğrenme sorununu engellemek amacıyla kullanılır. Aşırı öğrenme, sinir ağının eğitim verilerine aşırı derecede uyum sağlaması ve yeni verileri genelleme yapamadan ezberlemesi anlamına gelir. Bu durum sinir ağının gerçek dünya verileri üzerinde iyi performans gösterememesine yol açabilir.

Dropout katmanı, eğitim sırasında rastgele olarak seçilen bazı düğümleri geçici olarak devre dışı bırakır veya "dropout" eder. Bu, ağın farklı düğüm alt kümeleriyle eğitilmesini sağlar. Dolayısıyla her bir öğrenme adımında farklı düğümler devre dışı bırakılır ve sinir ağı farklı varyasyonlarla eğitilir. Bu süreç sinir ağının ezberlemesini engeller ve genelleme yeteneğini artırır.

Ezberleme (Dropout) katmanı Hinton ve arkadaşları tarafından önerilmiştir. Yapılan test performanslarıyla da düzenleme yeteneğini kanıtlamıştır. Bu katman, sinir ağının düzenlenmesini ve genelleme yeteneğini artırarak aşırı uyum sorununu azaltır (Hinton vd. 2012).

Şekil 3.10. da gösterilen çok katmanlı sinir ağı ve dropout yapılmış sinir ağı yapısı, dropout işleminin nasıl uygulandığını göstermektedir. Dropout işlemi yapılmış olan sağdaki şekilde bazı düğümler rastgele olarak seçilip ortadan kaldırılmış olarak görülmektedir. Bu sayede sinir ağının farklı varyasyonlarla eğitildiği ve ezberlemenin önüne geçildiği gösterilmektedir.

Şekil 3.10. da çalışma prensibi kısaca açıklanmıştır. Şekil 3.10. (a) ezberleme yapmış bir ağ yapısı gösterilirken, Şekil 3.10. (b) ezberleme katmanını kullanılan bir ağ yapısı gösterilir.



Şekil 3.10. (a) Yapay sinir ağı, (b) Dropout uygulanmış sinir ağı (Çarpı işareti atılmış nöronlar ağdan çıkarılmış)

3.5.6 Tam Bağlantı (Fully Connected, Fc) Katmanı

Sinir ağı modellerinde kullanılan en temel katmanlardan biridir. Bu katman bir önceki katmandaki tüm nöronlar ile kendi nöronları arasında bağlantı kurar. Her bir nöron, girdi verisindeki tüm özellikler ile ilişkilendirilir ve çıktı üretir (Çalışır, 2022).

Tam bağlantı katmanı her bir girdi özelliği ile ağırlıkla çarpılır ve toplanır. Ardından bir aktivasyon fonksiyonu (genellikle ReLU, sigmoid veya tanh gibi) kullanılarak çıktı hesaplanır. Bu işlem girdi verisindeki tüm özelliklerin model tarafından kullanılmasını sağlar.

Tam bağlantı katmanlarının önemi, verinin farklı özelliklerinin bir araya getirilerek daha karmaşık örüntülerin öğrenilmesini sağlamasıdır. Örneğin bir görüntü sınıflandırma modelinde tam bağlantı katmanı, görüntünün piksellerinden çıkarılan özellikleri kullanarak görüntünün hangi sınıfa ait olduğunu tahmin edebilir.

Tam bağlantı katmanları sinir ağı modellerindeki son katmanlarda genellikle kullanılır. Özellikle sınıflandırma, regresyon veya dil işleme gibi görevler için uygundur. Ancak tam bağlantı katmanlarının kullanımı büyük veri setlerinde ve karmaşık model yapılarında aşırı öğrenmeye (overfitting) neden olabilir (Yiğit, 2017). Bu nedenle aşırı öğrenmeyi azaltmak için dropout veya başka düzenleme teknikleri gibi yöntemlerle birlikte kullanılabilir.

3.5.7 Sınıflandırma (Classification) Katmanı

Derin öğrenme algoritmalarında olduğu gibi genel olarak yapay zeka algoritmalarında da son katman sınıflandırma katmanıdır. Bu katman, kendisinden önceki tam bağlantılı katmandan gelen verileri değerlendirerek ağın çıkış değerini oluşturan katmandır (Ciresan, 2011).

Tasarlanan ağın çıkışında üretilen sonuç, sınıflandırma bilgisi olabileceği gibi, eğri uydurma yöntemlerinde kullanılan bir polinomun katsayıları da olabilir. Sınıflandırma katmanından elde edilecek değer sınıf bilgisi ise bu katmanda birçok farklı sınıflandırma fonksiyonuyla kullanılabilir. Derin öğrenme uygulamalarında genellikle literatürde yaygın olarak kullanılan bir olasılıksal hesaplama yöntemi olan SoftMax sınıflandırıcısı tercih edilmektedir. SoftMax sınıflandırıcısı her bir sınıf için 0-1 aralığında değerler üretir. Bu üretilen sınıf değerlerinden 1'e en yakın olan değer girişin sınıf bilgisini oluşturur.

Bazı durumlarda bağlantı katmanından gelen öznitelikler, kullanılan algoritmalara bağlı olarak farklı sınıflandırma algoritmalarıyla da sınıflandırılabilir.

3.6 Çekişmeli Üretici Ağlar (ÇÜA)

Çekişmeli Üretici Ağlar (ÇÜA'lar), 2014 yılında Ian Goodfellow ve arkadaşları tarafından geliştirilen bir derin öğrenme modelidir. Yayınlandığı günden beri derin öğrenme alanında popüler bir yapay zekâ modelidir (Goodfellow vd. 2014).

3.6.1 ÇÜA Yapısı

Çekişmeli Üretici Ağlar (ÇÜA) modeli, birbirleriyle rekabet eden iki farklı yapay sinir ağından oluşur: Üretici (Generative, G) ve Ayırıcı (Discriminator, D). Bu iki ağın çekişmesiyle öğrenme gerçekleştirilir. Ayırıcı yapay sinir ağı, sentetik olarak üretilen (sahte) görüntüleri gerçek veri tabanındaki (gerçek) görüntülerden ayırmaya çalışırken; üretici yapay sinir ağı, gürültü sinyalinde gerçek veri tabanındaki görüntülere benzer görüntüler üretmeye çalışır ve bu üretilen verileri ayırıcı ağı kandırmak için kullanır. Üretici ağ, sahte para basan ve kontrolsüz bir şekilde kullanmaya çalışan bir kalpazan şebekesi gibi davranır. Ayırıcı ağ ise sahte parayı tespit eden polis rolünü üstlenir. Bu şekilde her iki ağ rekabet ederek öğrenme sürecini gerçekleştirir (Goodfellow vd. 2014).

3.6.2 Matematiksel Alt yapısı

Çekişmeli Üretici Ağın matematiksel alt yapısı;

Üretici, P_g dağılımını x verisi üzerinde öğrenilmesi için giriş olarak $P_{z(z)}$ üzerindeki gürültü değişkenlerini kullanarak önsel olarak tanımlanır.

Ardından $G(z; \theta_g)$ olarak adlandırılan bir haritalama veri uzayında gerçekleştirilir.

G , θ_g parametresiyle temsil edilen türevlenebilir bir fonksiyon olan çok katmanlı bir algılayıcıdır.

$D(x; \theta_d)$ adı verilen ikinci birçok katmanlı algılayıcı tanımlanır. Bu da skaler bir çıktı sağlar. Burada $D(x)$, verilerden (x) gelen olasılığı temsil eder.

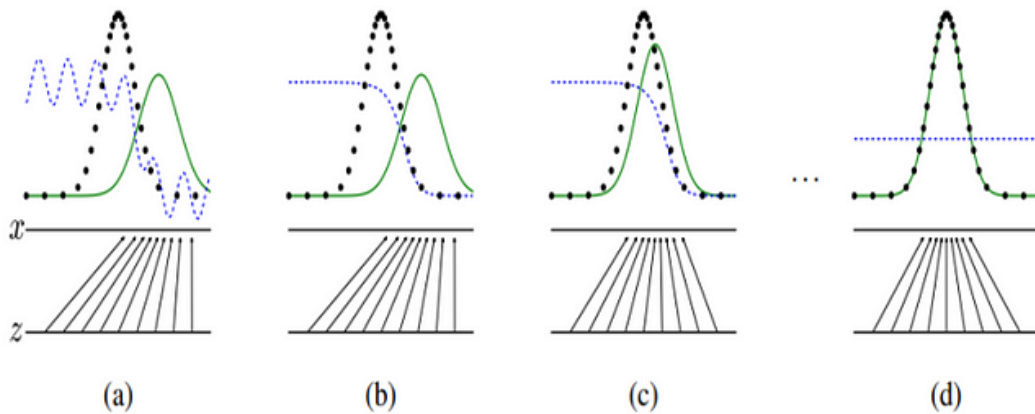
D , eğitim örneklerini ve G 'den gelen örnekleri doğru bir şekilde etiketlemek için eğitilir.

Aynı zamanda $\log(1 - D(G(z)))$ ifadesinin en aza indirmek için G eğitilir. Yani D ve G , $V(G, D)$ değer fonksiyonunu kullanarak aşağıdaki minimum ve maksimum işlevini gerçekleştirirler (Goodfellow, vd. 2014). Bu fonksiyona ayrıca "amaç" fonksiyonu da denir.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim P_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim P_z(z)} [\log (D(G(z)))] \quad (1)$$

Pratikte, denklem 1'de G 'nin daha iyi öğrenmesi için yeterli değişimi sağlamayabilir. Başlangıçta G zayıf olabilir ve eğitim verilerinden oldukça farklı olabilir. Bu nedenle D yüksek olasılıkla gelen örnekleri reddeder.

Bu durumda, $\log(1 - D(G(z)))$ ifadesi doyuma ulaşır. Bu nedenle G 'nin $\log(1 - D(G(z)))$ 'yi minimize etmek yerine $\log(1 - D(G(z)))$ 'yi maksimize etmek için eğitilmesi gerekmektedir. Bu şekilde G ve D aynı noktaya ulaşırlar. Çekişmeli üretici ağların eğitim aşamaları Şekil 3.11. de gösterilmiştir. Siyah noktalı çizgiler, hedeflenen veri dağılımını, mavi kesikli çizgi D 'nin ayırmacı dağılımını, yeşil düz çizgi G 'nin üretici dağılımını temsil eder. Yukarı oklar $x = G(z)$ eşlemesinin dönüştürülmüş örnekler üzerindeki P_g dağılımını nasıl uyguladığını gösterir (Goodfellow vd. 2014).



Şekil 3.11. ÇÜA Eğitiminin İlerlemesi (Goodfellow, vd. 2014)

3.6.3 ÇÜA Modelleri,

ÇÜA, birbirleriyle çatışan ve çatıştıkça öğrenen iki farklı sinir ağından oluşur. Tanıtıldığı günden itibaren büyük ilgi gören ve gün geçtikçe yeni modelleri ortaya çıkan ÇÜA modelleri, giderek daha da önem kazanan ve üzerinde çalışmaların yapıldığı gelişmekte olan bir alandır.

Güncel ÇÜA modelleri ve kısa açıklamaları:

Vanilla GAN (Standart GAN): İlk ÇÜA modelidir. Üretici ağ ile ayırıcı ağ arasında basit bir rekabeti içerir.

DCGAN (Deep Convolutional GAN): ÇÜA'yı daha iyi hale getirmek için konvolüsyonel katmanları ekleyerek özellikle görüntü verileri için daha etkili bir şekilde çalışan bir modeldir.

CGAN (Conditional GAN): Üretici ağ ve ayırıcı ağ belirli bir sınıfa ait veri üretmek veya değerlendirmek üzere etiket bilgilerini kullanabilir. Bu şekilde belirli sınıflara ait verilerin üretilmesi veya değerlendirilmesi mümkün olur.

InfoGAN (Information Maximizing GAN): Bu model üretici ağın öğrenmesini ve daha özgün veriler üretmesini sağlamak için ek bilgi ekler.

CycleGAN: ÇÜA'nın bir türüdür ve iki farklı veri kümesi arasında dönüşümler gerçekleştirmek için tasarlanmıştır. Örneğin bir resmin bir sanat eserine dönüştürülmesi gibi.

StyleGAN ve StyleGAN2: Özellikle yüzler gibi karmaşık yapıları üretmek için tasarlanmış olan bu modeller, yüksek çözünürlükte gerçekçi görüntüler üretebilir.

BigGAN: Daha büyük ve daha karmaşık modeller üretebilmek için tasarlanmış bir ÇÜA türüdür.

WGAN (Wasserstein GAN): Kaynak ve hedef veri dağılımları arasındaki mesafeyi azaltmak için tasarlanan bir ÇÜA türüdür.

Pix2Pix: ÇÜA'nın bir türüdür ve bir tür görüntüden başka bir tür görüntüye dönüşüm yapabilir.

Bunlar ÇÜA modellerinin sadece birkaç örneğidir ve sürekli olarak yeni modeller ve geliştirmeler eklenmektedir. ÇÜA'lar, özellikle görüntü işleme, sanat ve yaratıcı veri üretimi gibi birçok alanda kullanılmaktadır (Çelik ve Talu, 2020).

3.7. Derin Evrişimli Çekişmeli Üretici Ağlar (DEÇÜA)

2014 yılında Ian Goodfellow ve arkadaşları tarafından tanıtılan Generative Adversarial Networks (GAN'lar), bilim dünyasında büyük bir heyecan yaratmıştır. Bu gelişmenin ardından birçok model ortaya çıkmıştır. DEÇÜA bu modellerden biridir. DEÇÜA, 2015 yılında Radford ve arkadaşları tarafından yayımlanan "Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks" başlıklı makalede tanıtılmıştır (Radford, vd. 2015).

DEÇÜA, ÇÜA'ları görüntü sentezi alanında kullanmayı ve daha iyi sonuçlar elde etmeyi amaçlayan bir çalışma olarak ortaya çıkmıştır. Makale, ÇÜA'ların görüntü sentezi alanındaki potansiyelini incelemekte ve daha kaliteli ve gerçekçi görüntülerin üretilmesini sağlamak için derin sinir ağlarının kullanılmasını önermektedir (Radford, vd. 2015).

Makalede, DEÇÜA'nın bir dizi yenilikçi bileşeni ve metodolojisi sunulmuştur. Evrişimli katmanların ve ters evrişimli katmanların kullanımı, upsampling işleminin convolutional transpose katmanları ile gerçekleştirilmesi, toplu normalleştirme katmanlarının entegrasyonu ve LeakyReLU aktivasyon fonksiyonunun kullanılması gibi unsurlar DEÇÜA'nın temel özelliklerini oluşturur (Radford, vd. 2015).

DEÇÜA'nın ortaya çıkması görüntü sentezi alanında büyük bir dönüm noktası olmuştur. Derin sinir ağlarının kullanımı, daha kaliteli ve gerçekçi görüntülerin üretilmesini mümkün kılarak ÇÜA teknolojisinin potansiyelini artırmıştır. DEÇÜA, daha sonraki çalışmalara ilham vermiş ve ÇÜA tabanlı görüntü sentezi alanında birçok ilerlemeye yol açmıştır (Fang, vd. 2018).

Bu gelişmeler, ÇÜA'ların ve DEÇÜA'nın görüntü sentezi alanında önemli bir rol oynadığını ve araştırmacıları daha gerçekçi ve etkileyici görüntüler üretme konusunda ileriye taşıdığını göstermektedir.

3.8 Uygulama (Deney Ortamı)

Bu uygulama Google Colaboratory ortamında geliştirilmiştir. Colab, Google'ın geliştirdiği bir Jupyter Notebook hizmetidir. Jupyter Notebook, Python kodunu adım adım çalıştırmanızı ve sonuçları anlık olarak görmeyi sağlar. Colab ise bu Jupyter Notebook deneyimini bulut tabanlı hale getirerek kullanıcıların internet tarayıcıları üzerinden erişim sağlamasını sağlar.

Colab'ı kullanıcılara sağladığı avantajlar:

1. Ücretsiz Kullanım: Colab, ücretsiz bir sürüm sunar ve kullanıcıların bulut tabanlı hesaplama gücünden yararlanmasını sağlar. Bu şekilde kendi bilgisayarınızda yeterli kaynağa sahip olmasanız bile karmaşık hesaplamalar yapabilirsiniz.

2. Kolay Paylaşım: Colab projelerini paylaşmak oldukça kolaydır. Projelerinizi Google Drive üzerinde saklayabilir ve başkalarıyla paylaşabilirsiniz. Bu sayede iş birliği yapmak veya projelerinizi sunmak daha da kolaylaşır.

3. Donanım Kaynakları: Colab, kullanıcılara GPU ve TPU gibi yüksek performanslı donanım kaynakları sunar. Bu özellikle makine öğrenimi ve derin öğrenme gibi hesaplama yoğun işlemler için büyük bir avantaj sağlar.

4. Popüler Kütüphaneler ve Araçlar: Colab, popüler Python kütüphanelerini ve araçlarını önceden yüklenmiş olarak sunar. Bu sayede veri analizi, görüntü işleme, doğal dil işleme gibi farklı alanlarda hızlıca çalışmalar yapabilirsiniz.

5. Entegrasyon ve Uzantılar: Colab, Google Drive, GitHub ve diğer Google hizmetleriyle entegre çalışır. Ayrıca Jupyter Notebook uzantılarına da destek verir. Bu da kullanıcıların iş akışlarını özelleştirmelerine olanak tanır.

Colab'ın bazı sınırlamaları da vardır, örneğin oturum süresi sınırlaması ve sınırlı kaynak kullanımı gibi.

Bu çalışmada kullanılan; GPU özellikleri

```
[8] # GPU özelliklerini kontrol etme
!nvidia-smi
```

```
Fri Dec 8 08:41:04 2023
+-----+
| NVIDIA-SMI 525.105.17   Driver Version: 525.105.17   CUDA Version: 12.0   |
+-----+-----+-----+-----+-----+
| GPU  Name          Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf    Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|=====  
| 0   Tesla T4           Off       | 00000000:00:04.0 Off  |          0          |
| N/A   39C    P8      11W / 70W   |  3MiB / 15360MiB |    0%      Default  |
|=====  
+-----+-----+-----+-----+-----+
+-----+
| Processes:                                     GPU Memory |
|  GPU   GI    CI          PID    Type   Process name          Usage      |
|=====  
| No running processes found                    |
+-----+-----+-----+-----+-----+
+-----+

```

Şekil 3.12. Deney ortamında Colab'ın sundu GPU özellikleri

RAM özellikleri

```
# 2.2 RAM özelliklerini öğrenme
!cat /proc/meminfo

MemTotal:      13294208 kB
MemFree:       8849332 kB
MemAvailable:  11764996 kB
Buffers:       73844 kB
Cached:        3040676 kB
SwapCached:    0 kB
Active:        778020 kB
Inactive:      3353188 kB
Active(anon):  1032 kB
Inactive(anon): 1016980 kB
Active(file):  776988 kB
Inactive(file): 2336208 kB
Unevictable:   16 kB
Mlocked:       16 kB
SwapTotal:     0 kB
SwapFree:      0 kB
Dirty:         6884 kB
Writeback:     0 kB
AnonPages:     1016740 kB
Mapped:        497500 kB
Shmem:         1288 kB
KReclaimable:  86848 kB
Slab:          126964 kB
SReclaimable:  86848 kB
SUnreclaim:    40116 kB
KernelStack:   5832 kB
PageTables:    18756 kB
NFS_Unstable:  0 kB
Bounce:        0 kB
WritebackTmp:  0 kB
CommitLimit:  6647104 kB
Committed_AS: 3141956 kB
VmallocTotal: 34359738367 kB
VmallocUsed:   78592 kB
VmallocChunk:  0 kB
Percpu:        1320 kB
HardwareCorrupted: 0 kB
AnonHugePages: 0 kB
ShmemHugePages: 0 kB
ShmemPmdMapped: 0 kB
FileHugePages: 0 kB
FilePmdMapped: 0 kB
CmaTotal:      0 kB
CmaFree:       0 kB
HugePages_Total: 0
HugePages_Free: 0
HugePages_Rsvd: 0
HugePages_Surp: 0
Hugepagesize:  2048 kB
Hugetlb:       0 kB
DirectMap4k:   156472 kB
DirectMap2M:   4034560 kB
DirectMap1G:   11534336 kB
```

Şekil 3.13. Deney ortamında Colab'ın sunduğu RAM özellikleri

3.8.1 Kütüphaneler

Python kütüphaneleri, belirli bir işlevi yerine getirmek için bir araya getirilmiş modüller ve fonksiyonlar koleksiyonudur. Bu kütüphaneler, Python programlama dilinde yazılmış kodları yeniden kullanarak tekrar tekrar kullanılan işlevleri daha kolay ve hızlı hale getirmektedir. Colab birçok popüler Python kütüphanesini önceden yüklenmiş olarak sunar. Bu kütüphaneler, veri analitiği, makine öğrenimi, derin öğrenme ve daha pek çok alanda yapılacak çalışmaları kolaylaştırır ve verimliliği artırır. GPU ve TPU Desteği sağlar. Colab, grafik işlem birimleri (GPU) ve tensor işlem birimleri (TPU) gibi yüksek performanslı işlem birimlerini kullanmanıza olanak tanır. Derin öğrenme ve makine öğrenimi gibi hesaplama yoğun işlemleri hızlandırır. Bu çalışmada yoğun hesaplam

gerektiğinden Colab kullanılmıştır. Deney ortamında kullanılan Python kütüphaneleri Tablo 3.1. de verilmiştir.

Tablo 3.1. Deneyde ortamında kullanılan Python kütüphaneleri

Kütüphane Adları	Açıklama
%matplotlib inline	Matplotlib kütüphanesinin grafikleri Jupyter Notebook içinde gösterilmesini sağlar. “%matplotlib inline” komutu, grafikleri hücre içinde gösterir.
argparse	Komut satırı argümanlarını işlemek için kullanılan argparse kütüphanesini içeri aktarır.
os	İşletim sistemi ile ilgili işlevleri sağlayan os kütüphanesini içeri aktarır.
random	Rastgele sayılar üretmek için kullanılan random kütüphanesini içeri aktarır.
torch	PyTorch derin öğrenme kütüphanesini içeri aktarır.
torch.nn	PyTorch'ta sinir ağı modelleri oluşturmak için kullanılan nn (neural network) modülünü içeri aktarır.
torch.nn.parallel:	PyTorch'ta paralel hesaplamaları destekleyen modülü içeri aktarır.
torch.backends.cudnn	CUDA ile uyumlu derin sinir ağı uygulamaları için optimizasyonlar sağlayan cuDNN modülünü içeri aktarır.
torch.optim:	PyTorch'ta optimizasyon algoritmalarını içeren optim modülünü içeri aktarır.
torch.utils.data	PyTorch'ta veri yüklemesi ve işleme için yardımcı sınıfları içeren modülü içeri aktarır.
torchvision.datasets	PyTorch'ta popüler veri setlerini içeren torchvision.datasets modülünü içeri aktarır.
torchvision.transforms	PyTorch'ta görüntü dönüşümlerini içeren torchvision.transforms modülünü içeri aktarır.
torchvision.utils	PyTorch'ta yardımcı görüntü işleme işlevlerini içeren torchvision.utils modülünü içeri aktarır.
numpy	NumPy, bilimsel hesaplamalar ve matematiksel operasyonlar için kullanılan bir kütüphanedir.
matplotlib	Matplotlib, grafik ve görselleştirmeler oluşturmak için kullanılan bir çizim kütüphanesidir.
matplotlib.animation	Animasyon oluşturmak için Matplotlib kütüphanesinin animasyon modülünü içeri aktarır.
IPython.display	IPython ortamında HTML içeriğini görüntülemek için kullanılan IPython.display modülünden HTML sınıfını içeri aktarır. Bu genellikle animasyonları Jupyter Notebook içinde göstermek için kullanılır.

Tablo 3.2. Deneyde kullanılmak için atanan deęişkenler

Deęişken Adları	Açıklama
dataroot	Veri kümesi için kök dizin (Colab ortamına yüklendi)
workers = 2	Veri yükleyici için işçi sayısı
batch_size =128	Eđitim sırasında toplu boyut
image_size = 64	Eđitim görüntülerinin uzamsal boyutu. Tüm görüntüler bu boyuta yeniden boyutlandırılacak.
nc = 3	Eđitim görüntülerindeki kanal sayısı. Renkli görüntüler için bu deęer 3'tür.
nz = 100	Z gizli vektörünün boyutu (yani, üretcinin giriş boyutu)
ngf = 64	Üretcinin özellik haritalarının boyutu
ndf = 64	Ayırt edici özellik haritalarının boyutu
num_epochs =30	Eđitim epokları
lr = 0.0002	Optimizatörler için öğrenme oranı
beta1 = 0.5	Adam optimizatörü için beta1 hiper parametresi 0,5 seçildi.
ngpu = 1	Kullanılabilir GPU sayısı. CPU modu için 0 kullanın.

3.8.2 Üretici (Generator) Ağ

ÇÜA'ların bir bileşeni olan Üretici ağ rastgele gürültüden gerçekçi veri üretmeye çalışan bir yapay sinir ağıdır. Üretici ağ genellikle bir dizi katmandan oluşur ve girdi olarak rastgele bir vektör alır (Liu ve Hsieh, 2019). Bu girdi genellikle Gaussian dağılımına sahip rastgele gürültüdür. Üretici (G) gizli uzay vektörünü (z) veri uzayına dönüştürmek için tasarlanmıştır. Verilerimiz görüntüler olduğundan, z 'yi veri alanına dönüştürmek sonuç olarak eğitim görüntüleriyle aynı boyutta (yani $3 \times 64 \times 64$) bir RGB görüntü oluşturmak anlamına gelir. Pratikte bunu gerçekleştirmek için, her biri iki boyutlu toplu normalizasyon katmanı ve bir relu aktivasyon fonksiyonu ile eşleştirilmiş bir dizi adımli iki boyutlu evrişimli transpoze katman kullanılır.

Üretici çıkışı $[-1, 1]$ aralığında bir veri girişi için bir tanh fonksiyonuyla beslenir. DEÇÜA makalesinin önemli bir katkısı olduğundan dönüşüm-transpoze katmanlarından sonra toplu normalizasyon fonksiyonlarının varlığına dikkat etmek önemlidir (Radford, 2015). Bu katmanlar, eğitim sürecinde gradyan akışına yardımcı olur. Eğitim ilerledikçe, Üretici ağı daha gerçekçi veri örnekleri üretmesi beklenir. Üretici ağ, Ayırt edici ağla yarışma ve karşılıklı öğrenme süreci sayesinde orijinal görüntülere yakın nitelikte veri üretir ve bu şekilde ayırt edilemez hale gelir.

3.8.3 Ayırt Edici (Discriminator)

Ayırtıcı (D), bir görüntüyü girdi olarak alır ve bu girdi görüntüsünün gerçek (sahte olmayan) olduğuna dair bir olasılık değeri üreten ikili bir sınıflandırma ağıdır (Bahat ve Buluş, 2021). D , $3 \times 64 \times 64$ boyutunda bir giriş görüntüsünü alır bu görüntüyü bir dizi Conv2d, BatchNorm2d ve LeakyReLU katmanıyla işler ve bir Sigmoid aktivasyon fonksiyonu aracılığıyla nihai olasılık çıktısını verir. Bu mimari, gerektiğinde daha fazla katmanla genişletilebilir ancak stepwise evrişim, BatchNorm ve LeakyReLU kullanmanın önemli olduğu vurgulanmalıdır. DEÇÜA makalesi, havuzlama yerine adımli evrişim kullanmanın iyi bir uygulama olduğunu belirtir çünkü ağın kendi havuzlama işlevini öğrenmesine olanak tanır (Radford, vd. 2015). Ayrıca toplu normalizasyon ve LeakyRelu fonksiyonları hem G 'nin hem de D 'nin öğrenme süreci için önemli olan sağlıklı gradyan akışını destekler.

3.8.4 Loss Fonksiyonları ve Optimize Ediciler

Loss fonksiyonun amacı, fonksiyonun her iki log bileşenini (yani $\log(D(x))$ ve $\log(1 - D(G(z)))$) hesaplamasını sağlamaktır. Hangi kısmının kullanılacağını belirlemek için y girişini kullanabiliriz. Bu, eğitim döngüsünde gerçekleştirilir. Ancak sadece y 'yi değiştirerek hangi bileşeni hesaplamak istediğimizi seçebileceğimizi anlamak önemlidir (yani GT etiketleri).

Daha sonra gerçek etiketimizi 1 sahte etiketimizi ise 0 olarak tanımlıyoruz. D 'nin kayıplarını hesaplarken etiketleri kullanacağız. Bu aynı zamanda Ian Goodfellow ve arkadaşları tarafından yayınladıkları makalede kullanılan bir kuraldır. Son olarak biri D için ve diğeri G için olmak üzere iki ayrı optimize edici oluşturduk. Her ikisi de DEÇÜA belgesinde belirtildiği gibi Adam optimize edici kullanacak, öğrenme oranı 0,0002 ve Beta1 = 0,5 olacak. Üretici ağın öğrenme ilerlemesine bağlı olarak sabit bir grup oluşturacağız. Bu grup Gauss dağılımından çizilen gizli vektörlerin sayısını temsil edecek (yani sabit_noise). Eğitim döngüsünde periyodik olarak bu sabit_noise'u G 'ye iletilecek ve iterasyonlar boyunca gürültüden oluşan görüntüler göreceğiz (Colab ortamında).

4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

4.1 Eğitim Aşaması

Çekişmeli üretici ağlar, yapay zekâ ve makine öğrenmesi alanında oldukça etkileyici sonuçlar elde etmeyi başarmış bir derin öğrenme modelidir. ÇÜA'lar, üretici ağ ve ayırıcı ağ olmak üzere iki ana ağ bileşeninden oluşurlar. Bu iki bileşen birbirleriyle rekabet ederek eğitilirler. Sonucunda gerçekçi ve yeni verilerin üretilmesi amaçlanır. Bu süreç birbirini takip eden iterasyonlar halinde gerçekleştirilir. Her iterasyonda hata geri yayılımı (backpropagation) kullanılarak modellerin parametreleri güncellenir. Üretici ve ayırıcı model arasındaki bu rekabetin devam ettiği bir denge durumuna ulaşana kadar iterasyonlarla sürdürülür. Eğitimin sonunda, üretici model gerçekçi veriler üretebilir hale gelirken ayırıcı modelin de gerçek ve yapay verileri doğru bir şekilde ayırt edebilme yeteneği artar.

Bu çalışmada Goodfellow ve ard. tarafından yayınlamış oldukları makale ile Radford ve arkadaşları tarafından yayınladıkları makalelerde geçen algoritmalar baz alınarak deney ortamı hazırlanmıştır. Bu çalışmada “üretici ağ eğitim ve ayırıcı ağ eğitimi” olmak üzere iki ana bölümde incelenmiştir.

4.1.1 Ayırıcıyı Eğitimi

Ayırıcıyı eğitmenin amacı, belirli bir girdiyi gerçek veya sahte olarak doğru şekilde sınıflandırma olasılığını en üst düzeye çıkarmaktır. Goodfellow ve vd. göre, "ayırıcıyı güncellemek için stokastik gradyan kullanılmıştır.

$$\log(D(x)) + \log(1 - D(G(z))) \text{ ifadesini maksimize edildi.}$$

Bu işlemi iki adımda gerçekleştirildi.

1. Adım;

- Eğitim setinden bir grup gerçek örnek seçildi.
- Bu örnekleri ayırıcıya besleyip ileri geçiş yapıldı ve $(\log(D(x)))$ fonksiyonu ile kaybı hesaplandı.
- Gradyanları geriye doğru hesaplayarak güncelleme adımı gerçekleştirildi.

2. İkinci adım;

- Mevcut jeneratörle bir grup sahte örnek oluşturuldu.
- Sahte örneklerle ayırmacı beslendi ($\log(1 - D(G(z)))$) fonksiyonu ile kayıp hesaplandı.
- Gradyanlar geriye doğru biriktirildi.

Tamamen gerçek örneklerden biriken gradyanları ile tamamen sahte örneklerden biriken gradyanlar kullanarak ayırmacının optimize ederek fonksiyonu çağırılır.

4.1.2 Üretici Eğitimi

Goodfellow ve arkadaşlarının makalesinde, jeneratörün eğitimi için hedef, daha iyi sahte örnekler üretmek ve $\log(1-D(G(z)))$ fonksiyonunu mümkün olduğunca düşürmektir. Ancak, Goodfellow vd. bunun özellikle öğrenme sürecinin başlarında yeterli geçişler sağlamadığını göstermiştir. Bu çalışmada soruna çözüm olarak, $\log(D(G(z)))$ fonksiyonu maksimize edilmiştir. Ayrıca, jeneratörün kaybı, ayırmacı çıktısını kullanarak hesaplanmıştır. Gerçek etiketleri kullanarak kaybın hesaplanması sağlanmıştır. Gradyanlar geriye doğru hesaplanmış ve jeneratörün parametreleri güncellenmiştir. Gerçek etiketlerin kullanılması, kayıp fonksiyonunun $\log(x)$ kısmını ($\log(1-x)$) yerine kullanmamıza olanak sağlamıştır.

İstatistiksel raporlar yapılmıştır. Her epoch sonunda, jeneratörün eğitim ilerlemesini görsel olarak takip etmek için sabit_noise grubu jeneratöre beslenmiştir.

Kullanılan eğitim istatistikleri;

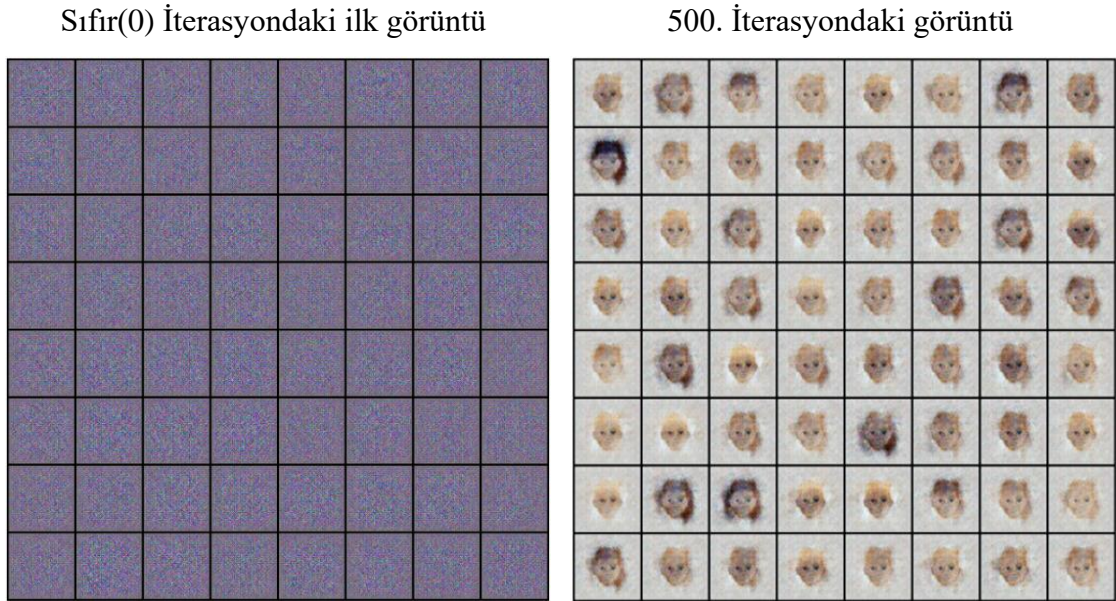
- **Loss_D:** Ayırmacı kaybı, tüm gerçek ve tüm sahte örnekler için kayıpların toplamı olarak fonksiyonu ile ($\log(D(x)) + \log(D(G(z)))$) hesaplandı.
- **Loss_G:** Jeneratör kaybı, $\log(D(G(z)))$ fonksiyonu olarak hesaplandı.
- **D(x):** Tüm gerçek örnekler için ayırmacının ortalama çıktısı (toplu olarak). Bu başlangıçta 1'e yakın olmalıdır ve G iyileştikçe teorik olarak 0.5'e yaklaşmalıdır.
- **D(G(z)):** Tüm sahte örnekler için ayırmacının ortalama çıktısı. İlk sayı, ayırmacı güncellenmeden önceki çıktıyı temsil ederken, ikinci sayı ayırmacı güncellendikten sonraki çıktıyı temsil eder. Bu sayılar başlangıçta 0'a yakın olmalıdır ve G iyileştikçe 0.5'e yaklaşmalıdır.

4.2 Sonuçlar (Görüntüler)

Deney ortamı 4 farklı veri seti üzerinde colab ortamında ve GPU kullanılarak gerçekleştirilmiştir. Deneylerden elde edilen görüntüler, grafikler ve ağların kayıpları verilmiştir. Elde edilen görüntüler tablo halinde aşama aşama verilmiştir. DEÇÜA ağlarını farklı veri setleri ile yapılan çalışmaları farklı epoch ve iterasyonda üretmiş olduğu resimleri ve kayıp fonksiyonları verilmiştir.

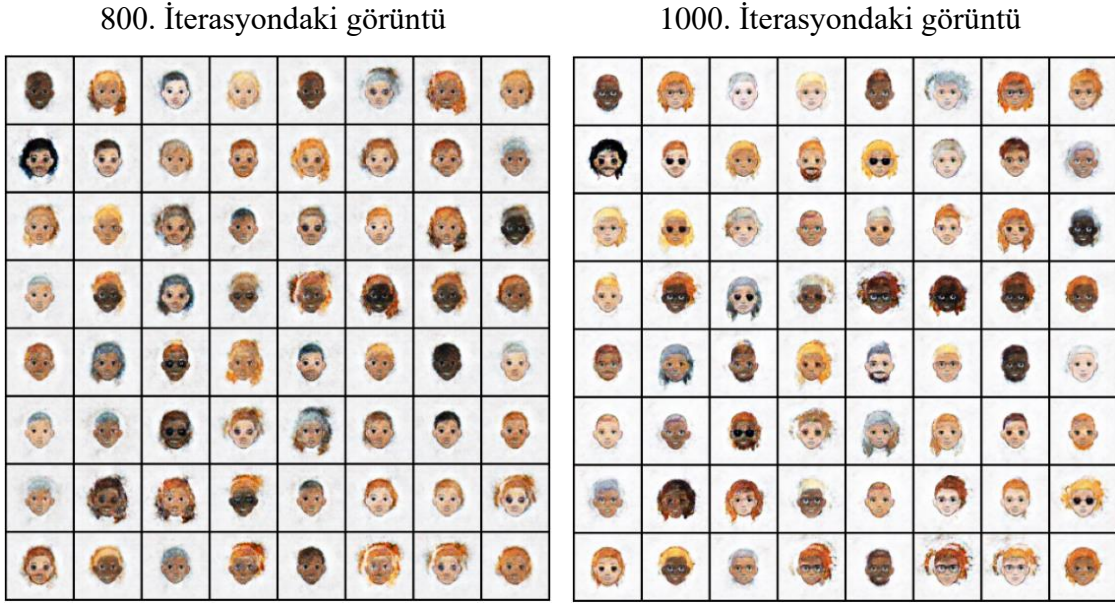
4.2.1 Cartoonset10k Veri Seti Görüntüleri

Colab ortamında Cartoonset10k veri seti ile çalışılmıştır. Veri seti 10.000 .png uzantılı resimden oluşmaktadır. Deneyde GPU kullanılmıştır. Deney ortamında üretici ağın (G) sentetik veri üretirken ilerlemesinin görselleştirilmesini gösteren görüntüler aşağıda gösterilmiştir.



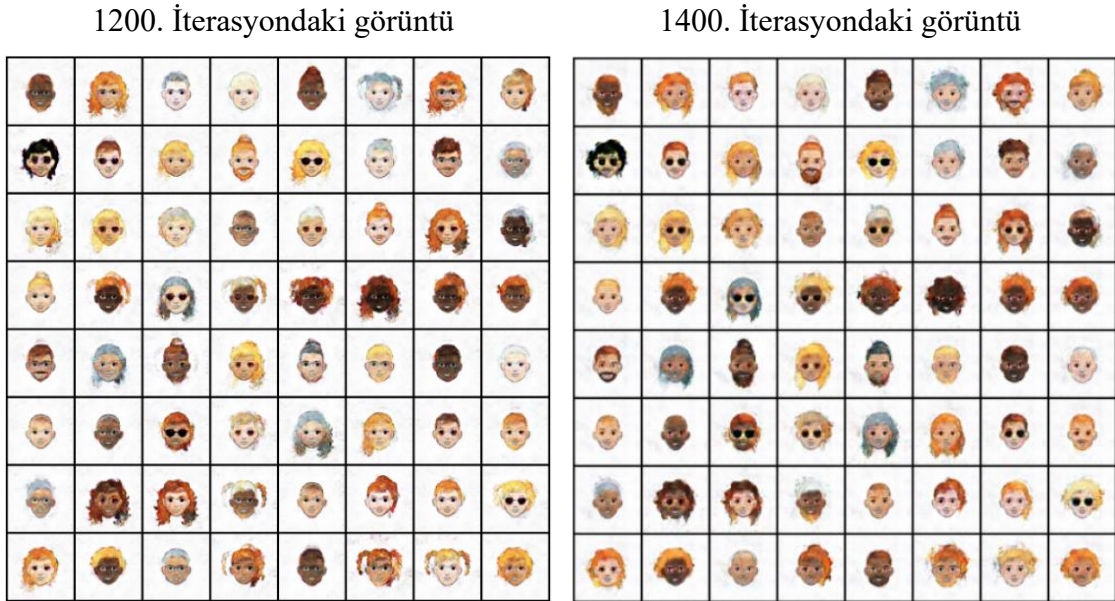
Şekil 4.1. Üretici ağın ilk görüntüsü ile 500 iterasyondaki görüntüler

Şekil 4.1. de Üretici ağın rasgele üretmiş olduğu ilk görüntü görülmektedir. Üretici ağının ilk ürettiği görüntüler genellikle rastgele ve düşük kaliteli olabilir. Üretici ağ eğitim ilerledikçe üreteceği görüntülerin gerçekçi resimlere benzeme beklenir. Üretici ağ 500. İterasyondan itibaren gerçekçi görüntülere benzetmeye başladığı görülmektedir.



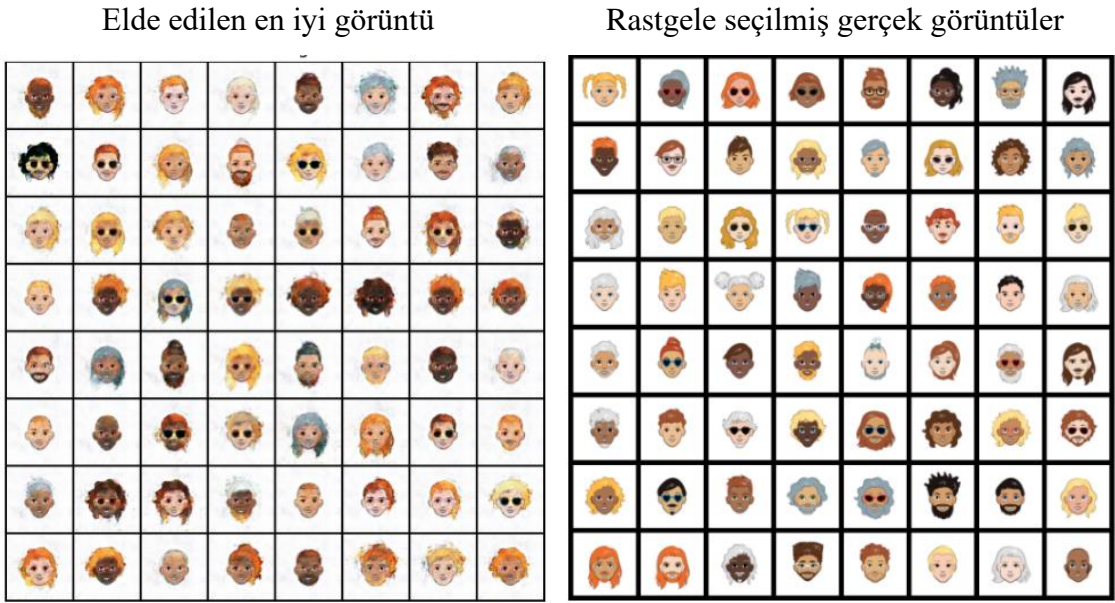
Şekil 4.2. Üretici ağın 800 ile 1000 iterasyondaki görüntüler

Şekil 4.2. Üretici ağın 800 ve 1000 iterasyonlarda üretmiş olduğu görüntüler görülmektedir. Üretici ağ eğitim ilerledikçe üretmiş olduğu görüntüler gerçekçi görüntülere daha da benzediği görülmektedir.



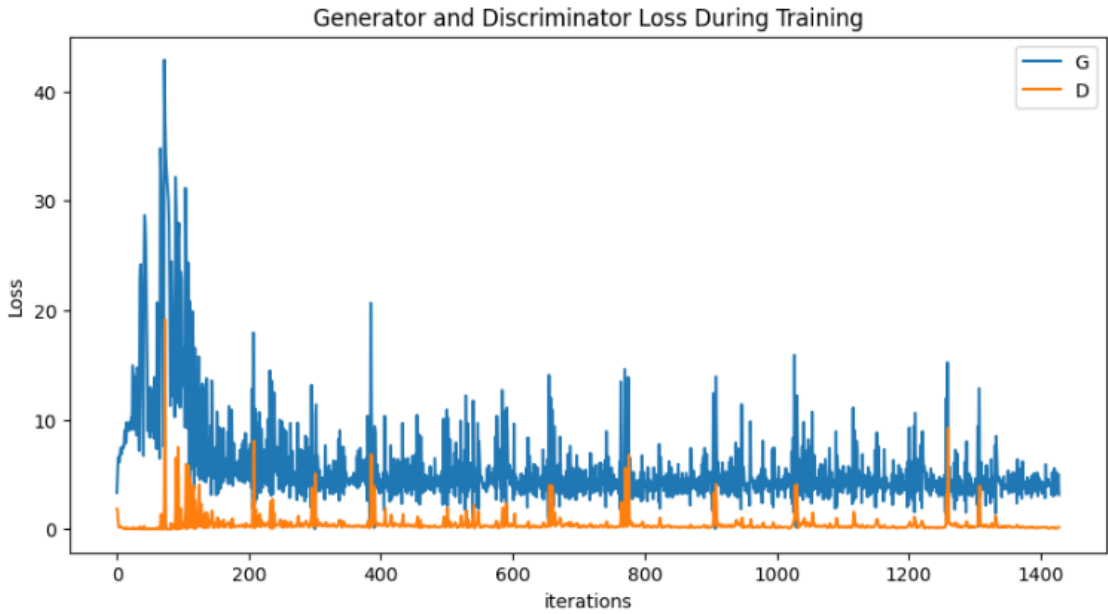
Şekil 4.3. Üretici ağın 1200 ile 1400 iterasyondaki görüntüler

Şekil 4.3. de Üretici ağın 1200 iterasyon ile 1400 iterasyondaki üretmiş olduğu sentetik görüntüler görülmektedir. Üretici ağın, eğitim süreci ilerledikçe gerçek görüntülere benzer ve daha gerçekçi görüntüler üretmeyi öğrendiği görülmektedir.



Şekil 4.4. Üretici ağın ürettiği en iyi görüntü ile veri setinden seçilen rasgele görüntüler

Şekil 4.4. de Üretici ağın ürettiği en gerçekçi görüntüsü görülmektedir. Sağda Cartoonset10k veri setinden rastgele(random) seçilmiş görüntüler bulunmaktadır. Üretici ağın eğitim boyunca gerçekçi görüntüler üretmekte kendini eğittiği, ürettiği kaliteli görüntülerle göstermiştir.



Şekil 4.5. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik

Şekil 4.5. Colab ortamında gerçekleştirilen deneyine ait Üretici ağ ile Ayırıcı ağın birbiri ile oyun dengesine dayalı eğitim sürecinin olduğu grafik gözükmemektedir. Üretici

ağ ve Ayırıcı ağ arasında bir oyuna dayalı bir eğitim süreciyle çalışır. Her iki ağın da eğitimi sırasında kayıp fonksiyonları kullanılır. Bu kayıp fonksiyonları, ağların performansını ölçer ve optimize etmek için kullanılır. Üretici ağındaki kayıp fonksiyonu ve Ayırıcı ağındaki kayıp fonksiyonu arasındaki ilişki, ÇÜA'ların nasıl çalıştığını anlamak için önemlidir.

Üretici ağındaki kayıp fonksiyonu genellikle "Üretici kaybı" olarak adlandırılır. Amacı, ürettiği sentetik görüntülerin Ayırıcı ağ tarafından gerçek olarak sınıflandırılmasını sağlamaktır. Üretici ağ, kayıp fonksiyonunu en aza indirerek Ayırıcı ağın yanıtacak ve ürettiği görüntülerin gerçek gibi algılanmasını sağlayacak şekilde öğrenir.

Ayırıcı ağdaki kayıp fonksiyonu ise "Ayırıcı kaybı" olarak adlandırılır. Amacı, gerçek görüntüleri doğru bir şekilde gerçek olarak sınıflandırmak ve Üretici tarafından üretilen sentetik görüntüleri yanıltıcı olarak sınıflandırmaktır. Ayırıcı ağ, kayıp fonksiyonunu en aza indirirken gerçek görüntüleri doğru bir şekilde sınıflandırmayı ve sentetik görüntüleri yanlışlıkla sınıflandırmayı öğrenir.

ÇÜA'ların eğitim süreci boyunca Üretici ağ ve Ayırıcı ağları arasında bir denge sağlanır. İdeal olarak, Üretici ve Ayırıcı ağlarının kayıp fonksiyonları birbirine yaklaşır ve yakınsar. Bu durumda, Üretici ağ daha gerçekçi görüntüler üretebilirken Ayırıcı ağ da daha doğru sınıflandırmalar yapabilir.

Kayıp fonksiyonu grafiği, eğitim sürecinde ağların performansının nasıl değiştiğini gösterir. Genellikle eğitim süreci boyunca Üretici kaybı azalırken Ayırıcı kaybı artar. Bu, Üreticinin daha başarılı hale geldiğini ve Ayırıcının daha zorlandığını gösterir. Eğitim süreci ilerledikçe ve ağlar arasındaki denge arttıkça hem Üretici kaybı hem de Ayırıcı kaybı azalır ve birbirine yaklaşır.

Şekil 4.5. Kayıp fonksiyonu grafiği incelendiğinde 0 ile 200 iterasyonları arasında üretici ağın kaybının en çok olduğu gözükmektedir. Eğitim ilerledikçe Üretici kaybın azaldığı gözükmektedir. Ayırıcı ağın da kaybının arttığı ve üretici ağa yaklaştığı görülmektedir.

Tablo 4.1. Ayırıcı (D) ağ ve üretici (G) ağın kayıplarının matematiksel gösterimi

[49/50][0/79]	Loss_D: 0.0665	Loss_G: 6.6825	D(x): 0.9641	D(G(z)): 0.0262 / 0.0055
[49/50][50/79]	Loss_D: 0.0500	Loss_G: 4.7653	D(x): 0.9689	D(G(z)): 0.0170 / 0.0168

Tablo 4.1. da Üretici ağ ile Ayırıcı ağın kayıp fonksiyonları değerleri grafikten ziyade matematiksel olarak gösterilmiştir. Colab ortamında gerçekleştirilen deneyde Loss 50 'ye ait Üretici(G) ağ ve Ayırıcı(D) ağına ait kayıp fonksiyonları değerleri görülmektedir.

4.2.2 Anime Face Veri Seti ile Oluşturulan Görüntüler

Colab ortamında Anime Face data seti ile çalışılmıştır. Data seti 63.565 adet .jpg uzantılı resimden oluşmaktadır. Deney Colab ortamında ve GPU kullanılarak gerçekleştirilmiştir. Deney ortamında Üretici ağın (G) sentetik veri üretirken ilerlemesinin görselleştirilmesini gösteren görüntüler aşağıda gösterilmiştir.



Şekil 4.7. Üretici ağın ilk görüntü ile 5000 iterasyondaki görüntüler

Şekil 4.7. de Üretici ağın rasgele üretmiş olduğu ilk görüntü görülmektedir. Solda Üretici ağının ilk ürettiği görüntüler görülmektedir. Sağdaki görüntü Üretici ağın 5000 iterasyondaki görüntüsüne ait. Üretici ağ eğitimi ilerledikçe üreteceği görüntülerin gerçekçi resimlere benzemesi beklenir. Üretici ağ 5000. İterasyondan itibaren gerçekçi görüntülere benzetmeye başladığı görülmektedir.

10000. İterasyondaki görüntü



15000. İterasyondaki görüntü



Şekil 4.8. Üretici ağıın 10000 ve 15000 iterasyondaki görüntüler

Şekil 4.8. Üretici ağıın 10000 ve 15000 iterasyonlarda üretmiş olduğu görüntüler görülmektedir. Üretici ağı eğitim ilerledikçe üretmiş olduğu görüntüler gerçekçi görüntülere daha da benzediği görülmektedir.

20000. İterasyondaki görüntü



25500. İterasyondaki görüntü



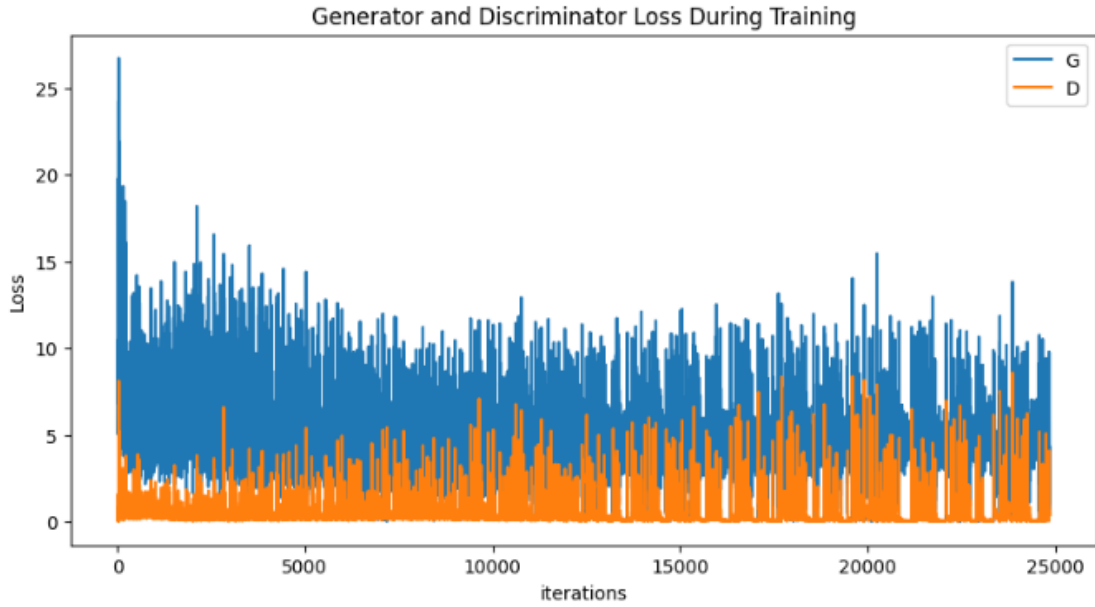
Şekil 4.9. Üretici ağıın 20000 ile 25000 iterasyondaki görüntüler

Şekil 4.9. de Üretici ağıın 20000 iterasyon ile 25000 iterasyondaki üretmiş olduğu sentetik görüntüler görülmektedir. Üretici ağıın, eğitim süreci ilerledikçe gerçek görüntülere benzer ve daha gerçekçi görüntüler üretmeyi öğrendiği görülmektedir.



Şekil 4.10. Üretici ağın ürettiği en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler

Şekil 4.10. da Üretici ağın ürettiği en gerçekçi görüntüsü görülmektedir. Sağda Anime Face (Mckinsey666) veri setinden rastgele(random) seçilmiş görüntüler bulunmaktadır. Üretici ağın eğitim boyunca gerçekçi görüntüler üretmekte kendini eğittiği, ürettiği kaliteli görüntülerle göstermiştir.



Şekil 4.11. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik

Şekil 4.11. Colab ortamında gerçekleştirilen deneyine ait Üretici ağ ile Ayırıcı ağın birbiri ile oyun dengesine dayalı eğitim sürecinin olduğu grafik gözükmemektedir. Üretici ağ kayıp fonksiyonları sıfırdan itibaren azalarak Ayırıcı ağa yaklaşılmaya

çalışıldığı görülmektedir. Aynı şekilde Ayırıcı ağda sıfırdan itibaren kayıp fonksiyonu artarak Üretici ağa yaklaştırmaya çalıştığı grafikte görülmektedir.

Tablo 4.2. Ayırıcı (D) ağ ve Üretici (G) ağ kayıplarının matematiksel gösterimi

[49/50][100/497]	Loss_D: 0.0598	Loss_G: 4.8049	D(x): 0.9808	D(G(z)): 0.0372 / 0.0152
[49/50][150/497]	Loss_D: 0.0417	Loss_G: 4.5690	D(x): 0.9804	D(G(z)): 0.0207 / 0.0190
[49/50][200/497]	Loss_D: 0.0716	Loss_G: 4.0662	D(x): 0.9654	D(G(z)): 0.0327 / 0.0312
[49/50][250/497]	Loss_D: 0.2278	Loss_G: 3.1814	D(x): 0.8634	D(G(z)): 0.0650 / 0.0705
[49/50][300/497]	Loss_D: 0.0993	Loss_G: 4.2401	D(x): 0.9514	D(G(z)): 0.0402 / 0.0285
[49/50][350/497]	Loss_D: 0.1595	Loss_G: 3.7233	D(x): 0.9357	D(G(z)): 0.0780 / 0.0424

Tablo 4.2. da Üretici ağ ile Ayırıcı ağın kayıp fonksiyonları değerleri matematiksel olarak gösterilmiştir. Colab ortamında gerçekleştirilen deneyde Loss 50 'ye ait Üretici(G) ağ ve Ayırıcı(D) ağa ait kayıp fonksiyonları değerleri görülmektedir.

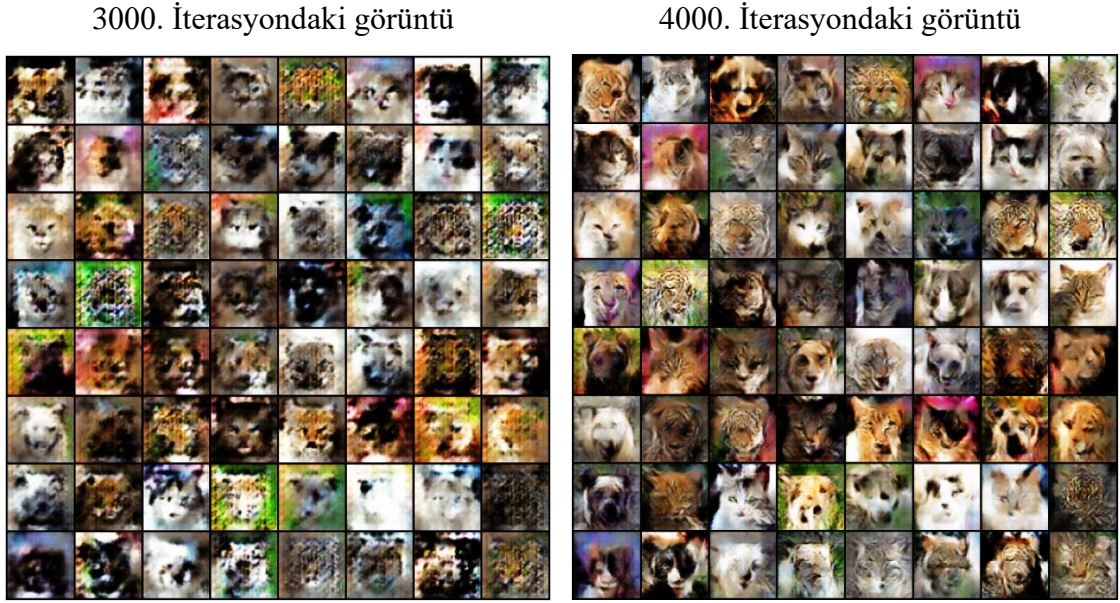
4.2.3 Animal Faces veriseti ile oluşturulan görüntüler

Colab ortamında Animal Faces veri seti ile çalışılmıştır. Veri seti 16.130 .jpg uzantılı resimden oluşmaktadır. Deneyde GPU kullanılmıştır. Deney ortamında Üretici ağın (G) sentetik veri üretirken ilerlemesinin görselleştirilmesini gösteren görüntüler aşağıda gösterilmiştir.



Şekil 4.13. Üretici Ağın ilk görüntü ile 2000 iterasyondaki görüntüler

Şekil 4.13'te Üretici ağın solda rasgele üretmiş olduğu ilk görüntü görülmektedir. Üretici ağın ilk ürettiği görüntü düşük kaliteli olabilir. Üretici ağın eğitim ilerledikçe üreteceği görüntülerin gerçekçi resimlere benzemesi beklenir. Sağda Üretici ağ 2000. İterasyondan üretmiş olduğu görüntü görülmektedir.



Şekil 4.14. Üretici ağın 3000 ile 4000 iterasyondaki görüntüleri

Şekil 4.14.'de Üretici ağın oluşturduğu görüntüler görülmektedir. Solda 3000. İterasyondaki görüntüler görülmektedir. Üretici ağ eğitim ilerledikçe üretmiş olduğu görüntüler gerçekçi görüntülere daha da benzediği görülmektedir. Solda Üretici ağın 4000. İterasyondaki görüntüsünün giderek gerçekçi resimlere yaklaştığı görüntüler görülmektedir.

5000. İterasyondaki görüntü



6000. İterasyondaki görüntü



Şekil 4.15. Üretici ağın 5000 ile 6000 iterasyondaki görüntüleri

Şekil 4.15’de Üretici ağın 5000 iterasyon ile 6000 iterasyonda üretmiş olduğu görüntüler gösterilmiştir. Solda Üretici ağın görüntülerde yüz ifadelerinin belirginleşmeye başladığı görülmektedir. Sağdaki görüntüde ise yüz ifadelerindeki belirginleşme gittikçe artmaya başlamıştır.

Elde edilen en iyi görüntü



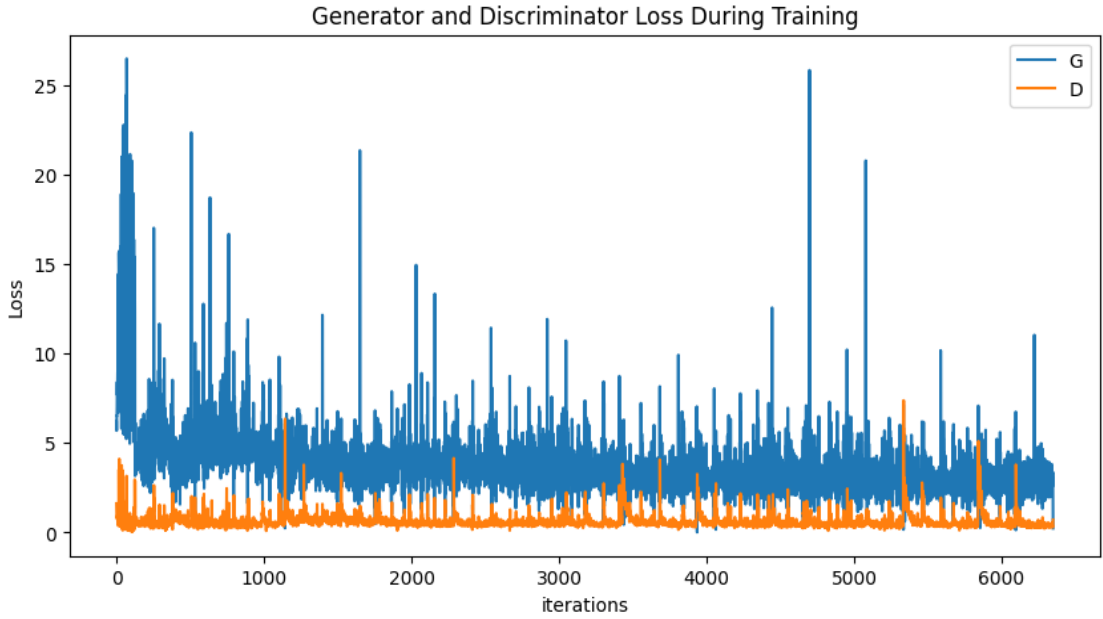
Rastgele seçilmiş gerçek görüntüler



Şekil 4.16. Üretici ağın üretmiş olduğu en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler.

Şekil 4.16’da Üretici ağın üretmiş olduğu en gerçekçi görüntüsü görülmektedir. Sağda Animal Face veri setinden rastgele (random) seçilmiş görüntüler bulunmaktadır.

Üretici ağı eğitim boyunca gerçekçi görüntüler üretmekte çalıştığı görülmektedir. Ancak kaliteli görüntüler oluşturmada başarılı olduğunu söyleyemez.



Şekil 4.17. Üretici(G) ve Ayırıcı(D) ağı eğitim boyunca kayıp ve kazançlarını gösteren grafik

Şekil 4.17.'de Colab ortamında gerçekleştirilen deneyine ait Üretici ağı ile Ayırıcı ağı eğitim sürecinin ait grafik görülmektedir. Grafikte Üretici ağı kayıp fonksiyonları azalıp artışı görülmektedir. Üretici ağı eğitim boyunca Ayırıcı ağı yaklaşıp uzaklaşmaktadır. Neticede başarılı görüntü üretmeme nedeni bundan kaynaklanmaktadır. Ayırıcı ağı ise genel itibari ile incelendiğinde 5000 ile 6000 iterasyonları arasında artmaya başladığı görülmektedir.

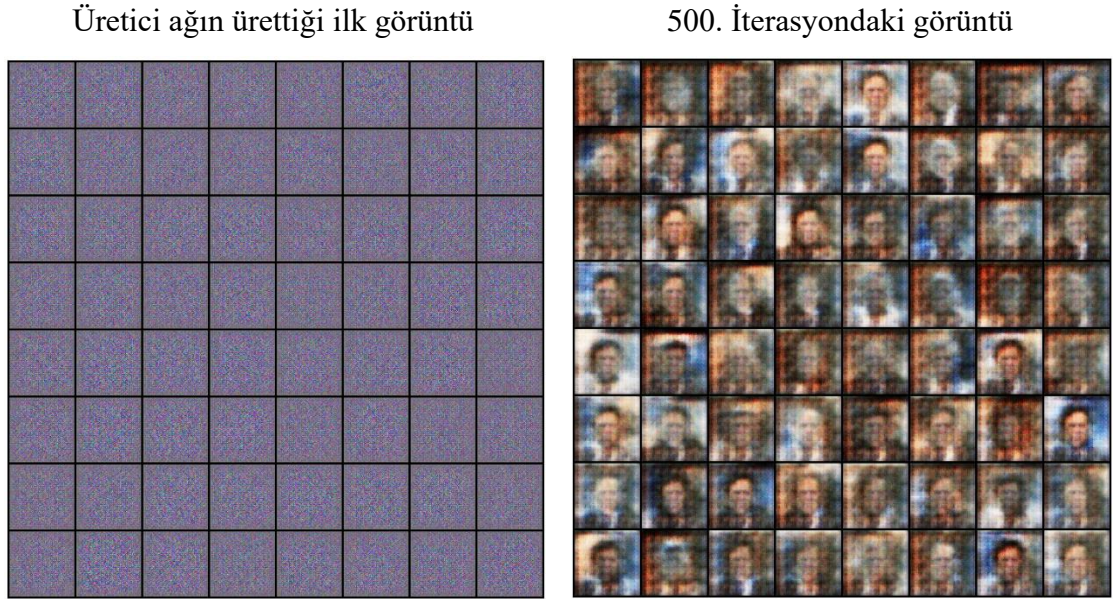
Tablo 4.3. Ayırıcı (D) ağı ve Üretici (G) ağı kayıplarının matematiksel gösterimi

[49/50][0/127]	Loss_D: 0.4600	Loss_G: 3.7186	D(x): 0.9030	D(G(z)): 0.2332 / 0.0490
[49/50][50/127]	Loss_D: 0.3804	Loss_G: 2.6900	D(x): 0.8234	D(G(z)): 0.1360 / 0.0968
[49/50][100/127]	Loss_D: 0.3369	Loss_G: 3.0280	D(x): 0.8300	D(G(z)): 0.1161 / 0.0751

Tablo 4.3'te Üretici ağı ile Ayırıcı ağı kayıp fonksiyonları değerleri matematiksel olarak gösterilmiştir. Colab ortamında gerçekleştirilen deneyde Loss 50 'ye ait Üretici(G) ağı ve Ayırıcı(D) ağı ait kayıp fonksiyonları değerleri görülmektedir.

4.2.4 CelebAFaces veriseti ile oluşturulan görüntüler

Colab ortamında CelebFaces veri seti ile çalışılmıştır. Veri seti 13.233 adet .jpg uzantılı resimden oluşmaktadır. Deneyde GPU kullanılmıştır. Deney ortamında Üretici ağın sentetik veri üretirken ilerlemesinin görselleştirilmesini gösteren görüntüler aşağıda gösterilmiştir.



Şekil 4.19. Üretici ağın ürettiği ilk görüntü ve 500 iterasyondaki görüntüler

Şekil 4.19.'da Üretici ağın solda rasgele üretmiş olduğu ilk görüntü görülmektedir. Üretici ağın eğitim ilerledikçe üreteceği görüntülerin gerçekçi resimlere benzemesi beklenir. Sağda Üretici ağ 5000. İterasyondan üretmiş olduğu görüntü görülmektedir.

2000. İterasyondaki görüntü



3000. İterasyondaki görüntü



Şekil 4.20. Üretici ağın 2000 ile 3000 iterasyondaki görüntüler

Şekil 4.20’de Üretici ağın oluşturduğu görüntüler görülmektedir. Solda 3000. İterasyondaki görüntüler görülmektedir. Solda Üretici ağın 4000. İterasyondaki görüntüsü görülmektedir.

4000. İterasyondaki görüntü



5000. İterasyondaki görüntü



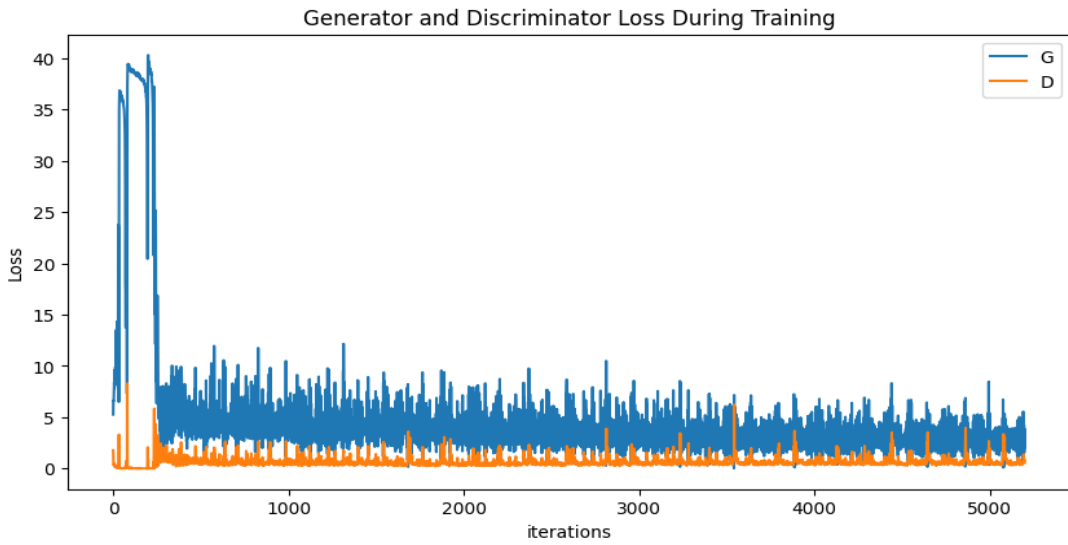
Şekil 4.21. Üretici ağın 4000 ile 5000 iterasyondaki görüntüler

Şekil 4.21.’de Üretici ağın CelebaFace veri setinden 4000 iterasyon ile 5000 iterasyonda üretmiş olduğu görüntüler gösterilmiştir. Solda Üretici ağın görüntülerde yüz ifadelerinin belirginleşmeye başladığı görülmektedir. Sağdaki görüntüde ise yüz ifadelerindeki belirginleşme oranında artışlar görülmektedir.



Şekil 4.22. Üretici ağın üretmiş olduğu en iyi görüntü ve veri setinden rasgele seçilmiş görüntüler

Şekil 4.22.'de Üretici ağın CelebAFace verisetinden üretmiş olduğu en gerçekçi görüntüsü görülmektedir. Sağda CelebAFace veri setinden rastgele(random) seçilmiş görüntüler bulunmaktadır. Üretici ağın eğitim boyunca gerçekçi görüntüler üretmekte çalıştığı görülmektedir. Yüzlerde belirgin bulanıklıklar mevcuttur. Kaliteli görüntüler oluşturmada başarılı olduğunu söylenemez.



Şekil 4.23. Üretici(G) ve Ayırıcı(D) ağın eğitim boyunca kayıp ve kazançlarını gösteren grafik

Şekil 4.23.'te Colab ortamında gerçekleştirilen deneyine ait Üretici ağ ile Ayırıcı ağın eğitim sürecine ait grafik görülmektedir. Grafikte Üretici ağın kayıp fonksiyonları 0 ile 1000 iterasyonları arasında azaldığı görülmektedir. Bu kayıp eğitim boyunca

azaldığı görülmektedir. Üretici ağ eğitim boyunca Ayırıcı ağa yaklaşmaktadır. Ayırıcı ağ ise eğitim boyunca incelendiğinde genel itibari ile sabit kaldığı, sona doğru ara ara arttığı görülmektedir. Grafikte Üretici ağın, Ayırıcı ağa yaklaşmaya çalışsa da istenilen ölçüde yaklaşma olmadığı, görüntülerde de istenilen başarı elde edilmediği görülmüştür.

Tablo 4.4. Ayırıcı ağ ve Üretici ağın kayıplarının matematiksel gösterimi

[49/50][0/104]	Loss_D: 0.7468	Loss_G: 4.1511	D(x): 0.8979	D(G(z)): 0.4080 / 0.0296
[49/50][50/104]	Loss_D: 0.5315	Loss_G: 3.3502	D(x): 0.8238	D(G(z)): 0.2554 / 0.0496
[49/50][100/104]	Loss_D: 0.7505	Loss_G: 1.5615	D(x): 0.5727	D(G(z)): 0.0812 / 0.2837

Tablo 4.4.'te Üretici ağ ile Ayırıcı ağın kayıp fonksiyonları değerleri matematiksel olarak gösterilmiştir. Colab ortamında gerçekleştirilen deneyde Loss 50 'ye ait Üretici ağ ve Ayırıcı ağa ait kayıp fonksiyonları değerleri görülmektedir.

5. SONUÇLAR VE ÖNERİLER

5.1 Sonuçlar

Bu çalışmada DEÇÜA'ların farklı veri setleri ile gerçek görüntü üretme performansı ölçülmüştür.

1. Cartoonset10k veri seti sonucunda gerçeğe çok yakın resimler elde edilmiştir. Bu set ile ilgili çalışmada 50 loss'da 5000 integrasyonla elde edilen resimlerde başlangıç noktasında herhangi bir görüntü elde edilememiştir. İlerleyen eğitimlerde görüntülerin gittikçe gerçeğe daha yakın olduğu gözlemlenmiştir.

2. İkinci veri setinde birinci veri setinde olduğu gibi 50 Loss'da çalışılmaya başlanmıştır. Bu veri setinde daha kaliteli resimler ortaya çıkmıştır. Bu durum, çalışmamız da giriş görüntülerinin vektör olması ve büyüklüğüne (Image size) bağlanmıştır.

3. Üçüncü veri seti çalışmasında Loss 50'den başlanılmıştır. Deneyler sonucu üretilen resimlerin piksel yoğunluğundan kaynaklı olarak gerçeğe benzer olmakla beraber beklenen oranda çıkmadığı görülmüştür.

4. Dördüncü veri setinde tekrar 50 Loss seviyesinden eğitime başlanmıştır. Diğer setlere benzer şekilde üretilen resimlerin bazılarında yüz ifadeleri görünürken bazılarında ise yüz ifadeleri net olarak çıkmamıştır.

Sonuç olarak

1. Giriş resimlerinden boyut olarak büyük olanlarının görüntü üretmede daha başarılı olduğu görülmüştür.

2. Boyut dışında vektör tabanlı resimlerin görüntü üretmede daha başarılı olduğu görülmüştür.

3. Verideki görüntülerin netliğinin resim üretme üzerinde etkili olduğu görülmüştür.

4. Veri setlerindeki resim adedine göre gerçekçi resim üretme yeteneğinin daha çok olduğu gözlemlenmiştir.

5. Vektör tabanlı resimlerin piksel tabanlı resimlere göre daha başarılı olduğu görülmüştür. Bu durum, resmin netliği ve gerçeğe yakın olması sonucunu ortaya çıkarmaktadır.

5.2 Öneriler

Bu alanda çalışma yapacak araştırmacıların;

1. GPU: COLAB'ın sunduğu GPU süresi ve internet kesintilerine karşı farklı alternatif GPU kaynaklarına ulaşım imkanlarını çözmeleri gerekir.

2. Veri setleri: Veri setlerindeki görüntü kalitesine ve boyutlarına çözüm getirmeleri önerilir.

3. Mod Çökmesi: Araştırmacıların ÇÜA'ların mod çökmelerine karşı çözümler üretmeleri gerekir.

4. Farklı ÇÜA modellerinin karşılaştırılması: Farklı ÇÜA modelleri (örneğin CGAN, WGAN, StyleGAN) kullanılarak görüntü üretme performansı karşılaştırılabilir. Bu, farklı modellerin avantajları, dezavantajları ve üretkenlik açısından nasıl farklılaştığını belirlemeye yardımcı olur.

5. Veri seti boyutlandırması ve etkisi: ÇÜA'larla görüntü üretme için kullanılan veri setinin boyutunun ve çeşitliliğinin sonuçlar üzerindeki etkisi incelenebilir. Farklı boyutlarda veri setleri kullanılarak üretilen görüntülerin kalitesi, çeşitliliği ve gerçekçiliği üzerindeki etkileri analiz edilebilir.

6. Öğrenme süreci ve hiperparametre ayarlaması: ÇÜA'larla görüntü üretme sürecinde kullanılan öğrenme süreci ve hiperparametrelerin (örneğin öğrenme oranı, toplam iterasyon sayısı, mini-batch boyutu) sonuçlar üzerindeki etkisi araştırılabilir. Bunun, daha istikrarlı ve kaliteli görüntü üretimi için en uygun ayarların belirlenmesine yardımcı olabileceği düşünülmektedir.

7. ÇÜA'lar için yeni özelliklerin keşfi: ÇÜA'ların üzerine yeni özellikler ekleyerek (örneğin dikkat mekanizmaları, çoklu ölçekli üretim, denetimli öğrenme) daha gelişmiş görüntü üretme sonuçları elde etmek mümkündür. Sonraki çalışmalarda farklı özelliklerin ÇÜA'lara nasıl entegre edilebileceği ve sonuçların nasıl iyileştirilebileceği araştırılabilir.

8. ÇÜA'lar ile eğitimsiz veriye dayalı görüntü üretimi: ÇÜA'lar genellikle büyük miktarda etiketli veri gerektiren denetimli öğrenme problemlerinde kullanılır. Ancak ÇÜA'ların eğitimsiz veriye dayalı görüntü üretimi için nasıl kullanılabileceği ve bu alanda potansiyel uygulamaların neler olduğu incelenebilir.

6. KAYNAKLAR

- Acar, O., 2020, Yapay zekâ fırsat mı yoksa tehdit mi?, *Kriter Yayınevi*, İstanbul.
- Aktürk, S., & Serbest, K., 2022, Nesne tespiti için derin öğrenme kütüphanelerinin incelenmesi, *Journal of Smart Systems Research*, 3(2), 97-119.
- Antipov, G., Baccouche, M., & Dugelay, J. L., 2017, Face aging with conditional generative adversarial networks, *In 2017 IEEE international conference on image processing (ICIP)* (pp. 2089-2093). IEEE.
- Arıcı, İ., 2023, Artificial intelligence and music education, *Sciences*, 13(3), 579-584.
- Atalay, M., & Çelik, E., 2017, Büyük veri analizinde yapay zekâ ve makine öğrenmesi uygulamaları-artificial intelligence and machine learning applications in big data analysis, *Mehmet Akif Ersoy Üniversitesi Sosyal Bilimler Enstitüsü Dergisi*, 9(22), 155-172.
- Ataseven, B., 2013, Yapay sinir ağları ile öngörü modellemesi, *Öneri Dergisi*, 10(39), 101-115.
- Balta, A., 2018, Makine öğrenmesi teknikleri ile ekolojik verilerin değerlendirilmesi, Yüksek lisans, *Fırat Üniversitesi fen bilimleri enstitüsü*, Elazığ.
- Bayuk, M. N., & Demir, B. N., 2019, Endüstri 4.0 kapsamında yapay zekâ ve pazarlamanın geleceği, *International Journal Of Social, Humanities And Administrative Sciences*, 5(19), 781-799.
- Bin, H., Weihai, C., Xingming, W., & Chun-Liang, L., 2017, High-quality face image sr using conditional generative adversarial networks, *arXiv preprint arXiv:1707.00737*.
- Birim, Ş. Ö., 2022, Moda görseli sınıflandırma: düzenleyici teknikler ile evrimsel sinir ağları uygulaması. *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 15(1), 66-76.
- Brock, A., Donahue, J., & Simonyan, K., 2018, Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.
- Canbolat Göçmen, Z. N., 2022, Sağlık hizmetlerinde yapay zekâ uygulamaları: Yoğun bakım örneği, Doktora Tezi, *İstanbul Medipol Üniversitesi Sosyal Bilimler enstitüsü*, İstanbul.
- Cihan, M., 2020, Hiperspektral görüntüleme yöntemi kullanılarak yenidoğan sağlık durumlarının derin öğrenme metotları ile sınıflandırılması, Yüksek lisans tezi, *Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü*, Konya.
- Ciresan, D. C., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J., 2011, Flexible, high performance convolutional neural networks for image classification.

- In *Twenty-second international joint conference on artificial intelligence*, Galleria 2, 6928 Manno-Lugano, Switzerland, 1237-1242.
- Coşkun, F., & Gülleroğlu, H. D., 2021, Yapay zekânın tarih içindeki gelişimi ve eğitimde kullanılması, *Ankara University Journal of Faculty of Educational Sciences (JFES)*, 54(3), 947-966
- Çalışır, B., 2022, Radyo haberleşmesinde evrişimli sinir ağı kullanılarak yapılan modülasyon sınıflandırması, *Fırat Üniversitesi Mühendislik Bilimleri Dergisi*, 34(2), 867-877.
- Çeber, B., 2022, Yapay zekâ uygulamalarının halkla ilişkiler aracı olarak kullanımı, Doktora tezi, *Marmara Üniversitesi Sosyal Bilimler Enstitüsü*, İstanbul.
- Çekerol, G. S., & Kesler, U., 2021, Artificial intelligence in marketing, *Journal of Current Researches on Social Sciences*, 11(4), 407-428.
- Çelik, G., & Talu, M. F., 2020, Çekişmeli üretken ağ modellerinin görüntü üretme performanslarının incelenmesi, *Balıkesir Üniversitesi Fen Bilimleri Enstitüsü Dergisi*, 22(1), 181-192.
- Çevik, K. K., & Dandil, E., 2012, Yapay sinir ağları için. net platformunda görsel bir eğitim yazılımının geliştirilmesi, *Bilişim Teknolojileri Dergisi*, 5(1), 19-28.
- Deng, L., Seltzer, M. L., Yu, D., Acero, A., Mohamed, A. R., & Hinton, G., 2010, Binary coding of speech spectrograms using a deep auto-encoder, In *Eleventh annual conference of the international speech communication association*. Makuhari, Chiba, Japan, 1692-1695.
- Doğan, F., & Türkoğlu, İ., 2019, Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme, *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 10(2), 409-445.
- Diri, B., 2014, Makine öğrenmesine giriş, *Retrieved from Lecture Notes Online Web site: <https://www.ce.yildiz.edu.tr/personal/banud/file/2634/Makine+Ogrenmesi-ML-10.Pdf>*, 10.10.2023.
- Elmas, Ç., 2021, Yapay Zekâ Uygulamaları, *Seçkin Yayıncılık*, (5. Baskı), Ankara, Syf. 99,151.
- Ersoy, E., & Karal, Ö., 2021, Yapay sinir ağları ve insan beyni, *İnsan ve Toplum Bilimleri Araştırmaları Dergisi*, 1(2), 188-205.
- Ersöz, F., & Çınar, Y., 2021, Veri madenciliği ve makine kullanımını yaklaşımlarının karşılaştırılması: Tekstil sektöründe bir uygulama, *Avrupa Bilim ve Teknoloji Dergisi*, (29), 397-414.
- Fang, W., Zhang, F., Sheng, V. S., & Ding, Y., 2018, A Method for Improving CNN-Based Image Recognition Using DCGAN, *Computers, Materials & Continua*, 57(1).

- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y., 2014, Generative adversarial nets, *Advances in neural information processing systems*, 27.
- Gürlek, Y., Bozkoyun, E., Ulutürk, M., & Zeyrekgündüz, F., 2023, Yapay Zekanın Eğitime Etkileri ve Uygulamaları, *International Journal of Original Educational Research*, 1(1), 25-132.
- Hinton, G. E., 2012, A practical guide to training restricted Boltzmann machines, In *Neural Networks: Tricks of the Trade: Second Edition* (pp. 599-619), Berlin, Heidelberg: Springer Berlin Heidelberg.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. R., 2012, Improving neural networks by preventing co-adaptation of feature detectors, *arXiv preprint arXiv:1207.0580*.
- Huang, R., Zhang, S., Li, T., & He, R., 2017, Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis, In *Proceedings of the IEEE international conference on computer vision* (pp. 2439-2448).
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A., 2017, Image-to-image translation with conditional adversarial networks, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
- İçöz, E., 2021, Covid-19 pandemi sürecinde Milli Eğitim Bakanı'nın Twitter mesajlarının metin madenciliği yöntemiyle incelenmesi, Yüksek lisans tezi, *Akdeniz Üniversitesi Eğitim Bilimleri Enstitüsü*, Antalya.
- İpek, S., 2019, Yapay zekanın toplum üzerindeki etkisi ve yapay zekâ (AI) filmi bağlamında yapay zekaya bakış, *Uluslararası Ders Kitapları ve Eğitim Materyalleri Dergisi*, 2(2), 203-215.
- İrdem, İ., & Çobanoğlu, S., 2021, Yapay zekânın iç güvenlik yönetimi üzerine yansımaları: Siber güvenlik, *Kamu Yönetimi ve Teknoloji Dergisi*, 3(2), 175-202.
- Jin, Y., Zhang, J., Li, M., Tian, Y., Zhu, H., & Fang, Z., 2017, Towards the automatic anime characters creation with generative adversarial networks, *arXiv preprint arXiv:1708.05509*.
- Karras, T., Aila, T., Laine, S., & Lehtinen, J., 2017, Progressive growing of gans for improved quality, stability, and variation, *arXiv preprint arXiv:1710.10196*.
- Keleş, A., Keleş, A., & Akçetin, E., 2017, Pazarlama alanında yapay zekâ kullanım potansiyeli ve akıllı karar destek sistemleri. *Electronic Turkish Studies*, 12(11).
- Kuş, Z., 2019, Mikrokanonikal optimizasyon algoritması ile konvolüsyonel sinir ağlarında hiper parametrelerin optimize edilmesi, Yüksek lisans tezi, *Fatih Sultan Mehmet Vakıf Üniversitesi Lisansüstü Eğitim Enstitüsü*, İstanbul.

- Kutlugün, M. A., Çakır, M. Y., & Kiani, F., 2017 Yapay sinir ağları ve K-En yakın komşu algoritmalarının birlikte çalışma tekniği (Ensemble) ile metin türü tanıma, 22. *Türkiye'de İnternet Konferansı*.
- Kuzey, C., 2012, Veri madenciliğinde destek vektör makinaları ve karar ağaçları yöntemlerini kullanarak bilgi çalışanlarının kurum performansı üzerine etkisinin ölçülmesi ve bir uygulama, Yayınlanmamış Doktora Tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İstanbul.
- Ledig, C., Theis, L., Huzár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W., 2017, Photo-realistic single image super-resolution using a generative adversarial network, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).
- Li, Y., Liu, S., Yang, J., & Yang, M. H., 2017, Generative face completion, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3911-3919).
- Liu, X., & Hsieh, C. J., 2019, Rob-gan: Generator, discriminator, and adversarial attacker, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11234-11243).
- McCarthy, J., 2007, From here to human-level AI, *Artificial Intelligence*, 171(18), 1174-1182.
- McCulloch, W. S., & Pitts, W., 1943, A logical calculus of the ideas immanent in nervous activity, *The bulletin of mathematical biophysics*, 5, 115-133.
- Metlek, S., & Çetiner, H., 2021, Matlap Ortamında Derin Öğrenme Uygulamaları, *İksad yayınevi*, Ankara.
- Metlek, S., & Kayaalp, K., 2020, Derin öğrenme ve destek vektör makineleri ile görüntüden cinsiyet tahmini, *Düzce Üniversitesi Bilim ve Teknoloji Dergisi*, 8(3), 2208-2228.
- Mitchell, T., 1997, *Machine Learning*, New York: McGraw Hill.
- Nabiyev, V. V., & Zekâ, Y., 2016, İnsan-bilgisayar etkileşimi, *Seçkin Yayıncılık, Sözkese Matbaacılık: Ankara*, 2-55.
- Ongsulee, P., 2017, Artificial intelligence, machine learning and deep learning, In *2017 15th international conference on ICT and knowledge engineering (ICT&KE)* (pp. 1-6). IEEE.
- Özdoğan, O., 2017, Endüstri 4.0: Dördüncü Sanayi Devrimi ve Endüstriyel Dönüşümün *Anahtarları*. *Pusula yayıncılık*, İstanbul
- Özkan, İ., & Ülker, E., 2017, Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6(3), 85-104.

- Öztemel, E., 2003, Yapay sinir ağları, *Papatya Yayıncılık*, İstanbul.
- Öztemel E., 2006, Yapay Sinir Ağları, *Papatya yayıncılık*, 2. Baskı. İstanbul
- Pang, Y., Sun, M., Jiang, X., & Li, X., 2017, Convolution in convolution for network in network, *IEEE transactions on neural networks and learning systems*, 29(5), 1587-1597.
- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A., 2016, Context encoders: Feature learning by inpainting, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2536-2544).
- Pehlivan, G., & Güven, Ö., 2019, Derin Öğrenmenin Felsefi Temelleri, Yüksek lisans, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İstanbul.
- Peker, O., Yılmaz, A., & Demır, A. U., 2018, Design of apnea detection interface including time delay neural networks for portable recording devices with three channels, In *2018 26th Signal Processing and Communications Applications Conference (SIU)* (pp. 1-4). IEEE.
- Perarnau, G., Van De Weijer, J., Raducanu, B., & Álvarez, J. M., 2016, Invertible conditional gans for image editing, *arXiv preprint arXiv:1611.06355*.
- Pirim, A. G. H., 2006, Yapay zekâ. *Yaşar Üniversitesi E-Dergisi*, 1(1), 81-93.
- Pulat, M., & Kocakoç, İ. D., 2021, Türkiye’de Makine Öğrenmesi ve Karar Ağaçları Alanında Yayınlanmış Tezlerin Bibliyometrik Analizi, *Yönetim ve Ekonomi Dergisi*, 28(2), 287-308.
- Radford, A., Metz, L., & Chintala, S., 2015, Unsupervised representation learning with deep convolutional generative adversarial networks, *arXiv preprint arXiv:1511.06434*.
- Reddy, R., 1996, The challenge of artificial intelligence. *Computer*, 29(10), 86-98.
- Russell, S. J., & Norvig, P., 2010, *Artificial intelligence a modern approach*. London.
- Rabie, A. A. A., 2021, Yarı-Nemli İklim Koşullarında Sürdürülebilir Su Yönetimi İçin Derin Öğrenme Kullanılarak Referans Bitki Su Tüketiminin Tahmin Edilmesi, Yüksek lisans, *Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü*, Bursa.
- Samuel, A. L., 1959, Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 3(3), 210-229.
- Sancakdar, A., & Kamer, V., 2021, Floridi’nin enformasyon felsefesi ve büyük veri, Yüksek lisans tezi, *İstanbul Üniversitesi Sosyal Bilimler Enstitüsü*, İstanbul.
- Saphioğlu, K., & Çimen, M., 2010, Yapay sinir ağlarını kullanarak günlük yağış miktarının tahmini, *Mühendislik Bilimleri ve Tasarım Dergisi*, 1(1), 14-21.

- Sayar, B., 2021, Yapay Zekâ ve Robot Teknolojisinin Gazetecilik Pratikleri Üzerindeki Etkileri, Doktora tezi, *Marmara Üniversitesi Sosyal Bilimler Enstitüsü*, İstanbul.
- Şimşek, M., 2022, Makine öğrenmesinin eğitimde kullanımı ile ilgili örnekler, *Eğitim & bilim 2022-iii*, 39.
- Taigman, Y., Polyak, A., & Wolf, L., 2016, Unsupervised cross-domain image generation, *arXiv preprint arXiv:1611.02200*.
- Terzi, Ö., & Mehmet, K., 2012, Yapay sinir ağları yöntemi ile Göksu nehri'nin akım tahmini, *Uluslararası Teknolojik Bilimler Dergisi*, 4(3), 1-7.
- Tutcu, B., 2023, Taşıt kredileri talep tahmininin yapay sinir ağları kullanılarak analiz edilmesi, *Dumlupınar University Journal of Social Science/Dumlupınar Üniversitesi Soysyal Bilimler Dergisi*, (78).
- Uzun, E., 2007, İnternet tabanlı bilgi erişimi destekli bir otomatik öğrenme sistemi, Doktora tezi, *Trakya Üniversitesi Fen Bilimleri Enstitüsü*, Edirne.
- Ünal, A., & Kılınc, İ., 2020, Yapay zekâ işletme yönetimi ilişkisi üzerine bir değerlendirme, *Yönetim Bilişim Sistemleri Dergisi*, 6(1), 51-78.
- Vasu, S., Madam N. T., & Rajagopalan A., 2018, Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network, *Computer Vision–ECCV*.
- Vondrick, C., Pirsivash, H., & Torralba, A., 2016, Generating videos with scene Dynamics, *Advances in neural information processing systems*, 29.
- Wang, T. C., Liu, M. Y., Zhu, J. Y., Tao, A., Kautz, J., & Catanzaro, B., 2018, High-resolution image synthesis and semantic manipulation with conditional gans, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8798-8807).
- Wu, J., Zhang, C., Xue, T., Freeman, B., & Tenenbaum, J., 2016, Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling, *Advances in neural information processing systems*, 29.
- Yakıt, İ., & Külcü, R., 2016, Ktesibios' un Su Saatinin Bilim Tarihi ve Mekanik Açısından Değerlendirilmesi. *International Multidisiplinary Congress Of Eurasia*, 6-11.
- Yavuz, S., & Deveci, M., 2012, İstatiksel normalizasyon tekniklerinin yapay sinir ağın performansına etkisi. *Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi*, (40), 167-187.
- Yenal, Ü., 2009, *Bilgi toplumunun tarihçesi*, Tarih Okulu Dergisi, 2009(V).

- Yiğit, A., 2017, İş süreçlerinde insan görüşünü derin öğrenme ile destekleme, Yüksek lisans tezi, Trakya Üniversitesi Fen Bilimleri Enstitüsü, Edirne.
- Yu, D., & Deng, L., 2010, Deep learning and its applications to signal and information processing [exploratory dsp], *IEEE Signal Processing Magazine*, 28(1), 145-154.
- Zhang, H., Sindagi, V., & Patel, V. M., 2019, Image de-raining using a conditional generative adversarial network, *IEEE transactions on circuits and systems for video technology*, 30(11), 3943-3956.
- Zhang, Z., Song, Y., & Qi, H., 2017, Age progression/regression by conditional adversarial autoencoder, In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 5810-5818).
- Zhu, Z., He, M., Dai, Y., Rao, Z., & Li, B., 2019, Multi-scale cross-form pyramid network for stereo matching. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)* (pp. 1789-1794). IEEE.