

**ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ**

DOKTORA TEZİ

**ÜRETKEN ÇEKİŞMELİ AĞLARDA GİZLİ UNSUR KODLAYICI İLE ÇIKTI
İMGESİ ARASINDAKİ İLİŞKİNİN HESAPLAMALI MODELLENMESİ**

Yahya DOĞAN

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

**ANKARA
2021**

Her hakkı saklıdır

ÖZET

Doktora Tezi

ÜRETKEN ÇEKİŞMELİ AĞLARDA GİZLİ UNSUR KODLAYICI İLE ÇIKTI İMGESİ ARASINDAKİ İLİŞKİNİN HESAPLAMALI MODELLENMESİ

Yahya DOĞAN

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Hacer YALIM KELEŞ

Üretken Çekişmeli Ağ (ÜÇA)'lar, başarılı foto-gerçekçi imgeler üretmeye olanak sağlamakta ve anlamlı bir gizli vektör içermektedir. ÜÇA mimarisi, gizli vektörden imge üretmeye olanak sağlasa da imgeden gizli vektöre kodlamayı sağlayacak bir yapı barındırmamaktadır. Tez kapsamında, denetimsiz bir ortamda bir Kodlayıcı ağ aracılığıyla, Üreticinin ters işlevini doğru bir şekilde öğrenmeyi sağlayan Döngüsel Ters Üretici (DTÜ) adlı bir model önerilmiştir. DTÜ modeli mevcut uçtan uca modellerle, gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılmış olup daha başarılı olduğu görülmüştür. DTÜ modeli kullanılarak birbirinden bağımsız iki problem alanı için çözümler sunulmuştur:

İmge öznitelik düzenleme: Amaç, herhangi bir imgede istenen bir özniteliği değiştirip imgedeki diğer ayrıntıları korumaktır. Literatürdeki hemen hemen tüm çalışmalar, bu amaçla denetimli bir ortamda etiketli veri setlerini kullanmaktadır. Yarı denetimli olarak bu hedefe ulaşmanın yöntemi, bir imgeyi temsil eden doğru gizli vektör temsilini ve bir özniteliğe karşılık gelen doğru yönü bulmaktır. Bu kapsamda, önerilen yöntem ile bir öznitelik için, ilgili özniteliği içeren ve içermeyen bir çift referans imge kullanılarak öznitelik düzenleme yapılmıştır. Ayrıca öznitelik yönleri ile ilgili çeşitli analizler ve deneysel çalışmalar yapılmıştır.

Yüz imgesi içboyama: Amaç, maskelenmiş bölgeleri algısal olarak makul içerikle doldurmaktır. Bu amaçla önerilen yaklaşımların farklı kısıtlamalara sahip oldukları gözlenmektedir. Bu bölümde, DTÜ modeli kullanılarak ilgili problem alanı için etkili bir yöntem önerilmiştir. Nihai olarak önerilen yöntem, en iyi çalışmalarla karşılaştırılmış ve özellikle büyük maskelerin kullanıldığı imgelerde daha iyi olduğu gözlenmiştir.

Şubat 2021, 135 sayfa

Anahtar Kelimeler: Üretken çekişmeli ağlar, imge öznitelik düzenleme, yüz imgesi içboyama, derin öğrenme, evrimsel sinir ağları

ABSTRACT

Ph.D. Thesis

COMPUTATIONAL MODELING OF THE RELATIONSHIP BETWEEN LATENT VECTOR ENCODING AND OUTPUT IMAGE IN GENERATIVE ADVERSARIAL NETWORKS

Yahya DOĞAN

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Doç. Dr. Hacer YALIM KELEŞ

Generative Adversarial Networks (GANs) enable successfully generating photo-realistic images and have a meaningful latent vector. GANs allow the generation of an image from a latent vector but do not provide an architecture to encode from image to latent vector. In the scope of the thesis, we proposed a model named Cyclic Reverse Generator (CRG) that enables to learn the inverse function of the generator via an encoder in an unsupervised setting. We show that CRG performs better in terms of image reconstruction compared to the existing end-to-end generative models. We also provide solutions for two problems using CRG:

Image attribute editing: The aim is in the manipulation of selected attributes of images while preserving the other details. Almost all the studies use labeled datasets in a supervised setting for this purpose. The method to achieve this goal by semi-supervised is to find an accurate latent vector representation of an image and a direction corresponding to the attribute. In this context, with the proposed method, we edit an attribute using a pair of reference images with and without the relevant attribute. In addition, we provide various analyses and experimental studies on attribute directions.

Facial image inpainting: The aim is to fill the masked regions with perceptually plausible content. We observe that the approaches proposed for this purpose have different limitations. In this section, we provide an efficient solution using CRG. Finally, we compared our method with state-of-the-art models and observed it is particularly better in images where larger masks are utilized.

February 2021, 135 pages

Key Words: Generative adversarial networks, image attribute editing, facial image inpainting, deep learning, convolutional neural networks

TEŞEKKÜR

Tez çalışmamın her aşamasında bilgi, öneri ve yardımlarını esirgemeyen, öğrencisi olmaktan onur duyduğum değerli hocam ve tez danışmanım Doç. Dr. Hacer YALIM KELEŞ'e,

Tez izleme aşamasında toplantılara katılarak fikir ve önerileriyle çalışmamın olgunlaşmasına yardımcı olan nezaketleriyle örnek aldığım tez izleme komitesi üyesi hocalarım Prof. Dr. Şahin EMRAH ve Doç.Dr. Sinan KALKAN'a,

Birlikte çalışmaktan zevk aldığım, desteklerini hep hissettiğim arkadaşlarım Arş. Gör. Özge MERCANOĞLU SİNCAN ve Arş. Gör. Zeynep YILDIRIM'a,

Doktora eğitimim süresince maddi olarak beni destekleyen TÜBİTAK-BİDEB'e (2211-C Öncelikli Alanlara Yönelik Yurt İçi Doktora Burs Programı), araştırmalarımnda kısmi nümerik hesaplamalara imkân sağlayan TÜBİTAK ULAKBİM'e ve tez çalışmamı destekleyen Ankara Üniversitesi BAP'a (Proje no: 18L0443010),

Bu günlere gelmemde büyük emek veren aileme,

Çalışmalarım boyunca bana güç veren, desteğini bir an bile eksik etmeyen, hayatımı özel yapan kıymetli eşime,

Son olarak zamanlarından çaldığım en değerli varlıklarım Taha'ya ve Kerem'e

sonsuz teşekkür ediyorum.

Yahya DOĞAN
Ankara, Şubat 2021

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK	i
ÖZET	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGELER VE KISALTMALAR DİZİNİ	vii
ŞEKİLLER DİZİNİ	ix
ÇİZELGELER DİZİNİ	xii
1. GİRİŞ	1
1.1 Genel Bilgi.....	1
1.2 Motivasyon ve Tez Kapsamı	3
1.3 Tezin Özgün Değeri.....	7
1.4 Tezin Yapısı	8
2. ÜRETKEN ÇEKİŞMELİ AĞLAR	9
2.1 Üretken Modeller	9
2.1.1 Otobağlayıcı ağlar	12
2.1.2 Değişimsel otokodlayıcılar	13
2.1.3 Üretken çekişmeli ağlar	15
2.2 Üretken Çekişmeli Ağ Mimarileri	17
2.2.1 Temel ÜÇA mimarisi	17
2.2.2 Koşullu ÜÇA mimarisi	18
2.2.3 Derin evrişimsel ÜÇA	20
2.2.4 Kademeli büyüyen ÜÇA	22
2.2.5 Stil tabanlı ÜÇA	23
2.3 Üretken Çekişmeli Ağlarda İstikrar ve Çeşitlilik	27
2.4 Modelleri Karşılaştırma Metrikleri.....	31
3. YARI DENETİMLİ İMGE ÖZNİTELİK DÜZENLEME	34
3.1 Literatür	35
3.2 Üretici Ağın Elde Edilmesi	38
3.3 Önerilen Yöntem	53

3.4 Modellerin Karşılaştırılması	58
3.5 Gizli Öznitelik Temsillerinin Elde Edilmesi	61
3.6 Öznitelik Yönlerinin Analizi	62
3.7 Öznitelik Kodlamada Rastgele Referans İmgelerin Etkisi.....	65
3.8 k Parametresi için Etkili Aralığın Belirlenmesi	67
3.9 Değerlendirme	71
4. YÜZ İMGESİ İÇBOYAMA	78
4.1 Lite atür.....	81
4.2 Önerilen Yöntem	84
4.3 Ayırıştırıcı Ağ	90
4.4 Görsel Bozuklukların Ortadan Kaldırılması.....	92
4.5 Önerilen Yöntemin Yetenekleri	95
4.6 Modellerin Karşılaştırılması	98
4.7 Modelin Başarısız Olduğu Durumlar	101
4.8 Değerlendirme	102
5. SONUÇ	104
5.1 Bulgular ve Öneriler	104
5.2 Gelecekte Yapılması Planlanan Çalışmalar.....	111
KAYNAKLAR	112
EK Öznitelik Düzenleme Sonuçları.....	122
ÖZGEÇMİŞ.....	133

SİMGELER VE KISALTMALAR DİZİNİ

α	Alpha
P	Dağılım
z	Gizli vektör
x	İmge
$\ddot{U}$	Üretici ağ
g_{ω}	Üretici ağ parametrik gösterim
K_j	j. katman aktivasyon haritası için normalizasyon katsayısı
P_z	z'nin olasılık dağılımı
z_a	Özniteliğin kodlandığı gizli vektör
z_f	f özniteliğini temsil eden gizli vektör yönü
z_p	p imgesini temsil eden gizli vektör
$\hat{\theta}$	Maksimum olabilirlik tahmincisi
μ_{z_i}	i. öznitelik vektör yönü kullanılarak elde edilen dağılımın ortalaması
σ_{z_i}	i. öznitelik vektör yönü kullanılarak elde edilen dağılımın standart sapması
ϕ_j^{I*}	I_* girdi imgesi için j. katmanın aktivasyon haritası
D	Değer fonksiyonu
ϕ	Haritalama fonksiyonu
L	Kayıp fonksiyonu
e_{ϵ}	Kodlayıcı ağ parametrik gösterim
z'	Tahmini gizli vektör
$\odot$	Piksel tabanlı çarpım
∇	Gradyan
$\in$	Eleman
A	Ayrıştırıcı ağ
$\exp$	Üstel Fonksiyon
n	Öğrenme oranı
Π	Çarpım sembolü
Σ	Toplam sembolü
C	Özellik haritası sayısı
G	Gram matrisi (ϕ_j^{I*} aktivasyon haritasının kovaryansları)
H	Yükseklik
I	İmge
M	Maske
W	Genişlik
k	Öznitelik empoze etme miktarı
$\mathbb{E}$	Beklenen değer
θ	Gerçek olasılık dağılımının parametreleri
μ	Dağılımın ortalaması
σ	Dağılımın standart sapması

Kısaltmalar

AEGAN	Otokodlayıcı tabanlı ÜÇA (Autoencoder based GAN)
-------	--

AEM	Ayırt Edici Model
AGE	Çekişmeli Üretici-Kodlayıcı (Adversarial Generator-Encoder)
ALI	Çekişmeli Öğrenilmiş Çıkarım (Adversarially Learned Inference)
BCE	İkili Çapraz Entropi (Binary Cross Entropy)
BiGAN	Çift yönlü ÜÇA (Bidirectional GAN)
CE	Bağlam Kodlayıcı (Context Encoder)
CRG	Döngüsel Ters Üretici (Cyclic Reverse Generator)
DCGAN	Derin Evrişimsel ÜÇA (Deep Convolutional GAN)
DenseNet	Yoğun Evrişimsel Ağ (Dense Convolutional Network)
DESA	Derin Evrişimsel Sinir Ağ
DFNet	Derin Füzyon Ağ (Deep Fusion Network)
dhash	Fark adresleme (Difference hash)
DTÜ	Döngüsel Ters Üretici
ESA	Evrişimsel Sinir Ağ
FID	Fréchet Başlangıç Mesafesi (Fréchet Inception Distance)
GAN	Üretken Çekişmeli Ağ (Generative Adversarial Network)
GIICA	Bağlamsal Dikkatle Üretken İmge İçboyama (Generative Image Inpainting with Contextual Attention)
GLCIC	Global ve Yerel Olarak Tutarlı İmge Tamamlama (Globally and Locally Consistent Image Completion)
GMCNN	Üretken Çok Sütunlu Evrişimsel Sinir Ağ (Generative Multi-column Convolutional Neural Networks)
GTT	Gradyan Tabanlı Teknik
ILSVRC	ImageNet Büyük Ölçekli Görsel Tanıma Yarışması (ImageNet Large Scale Visual Recognition Challenge)
IS	Başlangıç skoru (Inception score)
KF	Kayıp Fonksiyonu
KL	Kullback–Leibler
LR	Gizli Regresör (Latent Regressor)
LReLU	Sızdıran Doğrultulmuş Doğrusal Birim (Leaky Rectified Linear Unit)
MSS	Miniyığın Standart Sapma
OKH	Ortalama Karesel Hata
OMH	Ortalama Mutlak Hata
ÖD	Öz-Dikkat
Pconv	Kısmi Evrişimsel (Partial Convolution)
PGGAN	Kademeli Büyüyen ÜÇA (Progressive Growing GAN)
phash	Algısal adresleme (Perceptual hash)
PIC	Çoğulcu İmge Tamamlama (Pluralistic Image Completion)
ReLU	Doğrultulmuş Doğrusal Birim (Rectified Linear Unit)
ResNet	Artık Sinir Ağı (Residual Neural Network)
RMSProp	Kök Ortalama Kare Yayılımı (Root Mean Square Propagation)
SAGAN	Öz Dikkat ÜÇA (Self-Attention GAN)
SNGAN	Spektral Normalleştirme ÜÇA (Spectral Normalization GAN)
TG	Eğitilen Üretici (Trained Generator)
TTUR	İki Zaman Ölçekli Güncelleme Kuralı (Two Time-scale Update Rule)
ÜÇA	Üretken Çekişmeli Ağlar
ÜM	Üretken Model
VAE	Değişimsel Otokodlayıcı (Variational Autoencoder)
VAE-GAN	Değişimsel Otokodlayıcı ÜÇA (Variational Autoencoder GAN)
WGAN	Wasserstein ÜÇA (Wasserstein GAN)
WGAN-GP	WGAN - Gradyan Cezası (WGAN - Gradient Penalty)
whash	Dalgacık adresleme (Wavelet hash)

ŞEKİLLER DİZİNİ

Şekil 1.1 (a) ayırt edici model (b) üretken model	2
Şekil 1.2 ÜÇA mimarisi (Goodfellow vd. 2014)	3
Şekil 1.3 İlk kolon (1) gülme özniteliği taşıyan imgeleri, ikinci kolon (2) gülme özniteliği taşımayan imgeleri, üçüncü kolon (3) gülme özniteliği uygulanmak istenen imgeleri temsil etmektedir. (a) çerçevesi imge uzayı ve (b) çerçevesi gizli vektör uzayında yapılan cebirsel işlemler ve sonuçları göstermektedir (Radford vd. 2016).....	4
Şekil 2.1 Üretken modellerin sınıflandırılması (Goodfellow 2016a).....	11
Şekil 2.2 PixelRNN çalışması diyagonal koşullu olasılıkların belirlenmesi için temsili gösterim. Sol üst köşeden sol alt köşeye doğru ardışık olarak piksel üretimi yapılmaktadır. Renkli olanlar üretilen pikselleri temsil etmektedir.....	13
Şekil 2.3 Değişimsel otokodlayıcı mimarisi. Kodlayıcı ağ, imgeleri girdi olarak alıp ortalama ve standart sapma vektörüne dönüştürmektedir. Kodçözücü ağ ise örneklenmiş gizli vektörü girdi olarak almakta ve imgeyi yeniden oluşturmaktadır. Mimaride, gizli vektör uzayında KL kaybı, imge uzayında ise yeniden yapılandırma kaybı kullanılmaktadır.	14
Şekil 2.4 Tam bağlantılı ÜÇA mimarisi (Goodfellow vd. 2014). Mimari, çok katmanlı algılayıcı (multilayer perceptrons) yapısında kurulmakta olup imgeler piksel dizisine dönüştürüldükten sonra girdi olarak verilmektedir. Karmaşıklık önlemek amacıyla nöronlar arasındaki tüm bağlantılar çizilmemiştir	17
Şekil 2.5 Sınıf etiketinin bir-elemanı-bir olan vektöre dönüştürülmesi.....	19
Şekil 2.6 Koşullu ÜÇA mimarisi (Mirza ve Osindero 2014). Mimaride, girdi imgesinin dışında ilgili imge ile ilgili yardımcı bir bilgi hem Üretici hem de Ayırıştırıcı ağına girdi olarak verilmektedir	19
Şekil 2.7 DCGAN mimarisi (Radford vd. 2016). Mimaride, hem Üretici hem de Ayırıştırıcı ağında evrişimsel katmanlar kullanılmaktadır. Sol taraf: Üretici ağını, Sağ taraf: Ayırıştırıcı ağını temsil etmektedir	21
Şekil 2.8 PGGAN mimarisi (Karras vd. 2017). Şekil ilgili çalışmadan alınmıştır. Mimaride, eğitimin başlangıcında hem Üretici hem de Ayırıştırıcı ağ için 4x4 piksel çözünürlüğü kullanılmaktadır. Bu çözünürlükte eğitim kararlı duruma geldiğinde, her iki ağın çözünürlüğü iki katına çıkarılmaktadır. Bu işleme, hedef çözünürlüğe ulaşılan kadar devam edilmektedir	23
Şekil 2.9 StyleGAN mimarisi (Karras vd. 2018). Şekil ilgili çalışmadan alınmıştır. (a) PGGAN Üretici ağ mimarisini, (b) ise StyleGAN Üretici ağ mimarisini temsil etmektedir	24
Şekil 2.10 StyleGAN’da kullanılan farklı uzayların incelenmesi (Karras vd. 2018). Şekil ilgili çalışmadan alınmıştır. (a) veri setindeki imgelerde “erkek” ve “uzun saç” özniteliklerinin dağılımını, (b) bu imgeleri temsil eden z vektörlerinin özniteliklere haritalanma sonucunu ve (c) ise bu imgeleri temsil eden w vektörlerinin özniteliklere haritalanma sonucunu göstermektedir	25
Şekil 2.11 StyleGAN kullanılarak imgelerde stil değiştirme. 1. imge: düşük, orta veya yüksek seviyeli özniteliklerden birinin alındığı imgeyi, 2. imge: ilk imgede alınan öznitelikler dışındaki özniteliklerin alındığı imgeyi temsil etmektedir. Kaba; 1. imgeden kaba özniteliklerin ve 2. imgeden diğer özniteliklerin, Orta: 1. imgeden orta özniteliklerin ve 2. imgeden diğer özniteliklerin, İnce: 1.	

imgeden ince özniteliklerin ve 2. imgeden diğer özniteliklerin alınması sonucunu göstermektedir	26
Şekil 2.12 Gürültü verisinin imgeler üzerindeki etkisi. (a) sentezleme ağına gürültü verisinin verilmemesi, (b) ise sentezleme ağına rastgele gürültü verisinin girdi olarak verilmesi sonucunu göstermektedir	27
Şekil 2.13 <i>FID</i> ve <i>IS</i> metriklerinin karşılaştırılması (Heusel vd. 2017). Şekil ilgili çalışmadan alınmıştır. <i>FID</i> : <i>Fréchet inception distance</i> , <i>IS</i> : <i>Inception score</i> . (a) imgelere <i>Gauss</i> gürültüsü eklenmesi, (b) <i>Gauss</i> bulanıklığı uygulanması, (c) siyah dikdörtgenler eklenmesi, (d) spiral döndürme yapılması, (e) ise tuz biber gürültüsü eklenmesi sonucu metriklerin davranışlarını göstermektedir	33
Şekil 3.1 GTT yapısı (Lipton ve Tripathi 2017)	36
Şekil 3.2 Üretici ve Ayrıştırıcı ağları kayıp grafiği.....	40
Şekil 3.3 Rastgele üretilen imgeler	41
Şekil 3.4 Gradyan azalmasının imge kalitesi üzerindeki etkisi	42
Şekil 3.5 Gradyan patlamasının imge kalitesi üzerindeki etkisi	42
Şekil 3.6 SAGAN mimarisi kullanılarak rastgele üretilen imgeler	44
Şekil 3.7 Modelde kayıp fonksiyonu olarak <i>WGAN-GP</i> kullanılmasının etkisi.....	45
Şekil 3.8 Büyük öğrenme oranı yerine farklı sayıda eniyilemenin etkisi	46
Şekil 3.9 Miniyığın standart sapmanın imge çeşitliliği üzerindeki etkisi	47
Şekil 3.10 Öz-dikkat mekanizmasının imge kalitesi üzerindeki etkisi	48
Şekil 3.11 PGGAN çalışması kullanılarak elde edilen ilk imgeler.....	49
Şekil 3.12 4x4 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	49
Şekil 3.13 8x8 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	50
Şekil 3.14 16x16 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	50
Şekil 3.15 32x32 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	51
Şekil 3.16 64x64 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	51
Şekil 3.17 128x128 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler.....	52
Şekil 3.18 CelebA veri seti için oluşturulan ÜÇA mimarisi.....	52
Şekil 3.19 Üretilen iki imge arasında aradeğerleme sonuçları.....	53
Şekil 3.20 Önerilen CRG mimarisi	54
Şekil 3.21 Kodlayıcı ağ mimarisi.....	56
Şekil 3.22 Gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılması: (birinci sıra) gerçek imgeler, (ikinci sıra) önerilen CRG mimarisi (Üretici parametreleri sabit) ve (üçüncü sıra) CRG(TG) mimarisi (Üretici parametreleri de eğitilmektedir) sonuçları gösterilmiştir	57
Şekil 3.23 Modellerin gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılması. <i>Pioneer</i> , <i>LR</i> ve CRG mimarisi 128x128, diğer modeller ise 64x64 piksel çözünürlüğünde imgeler üretmektedir	58
Şekil 3.24 Üst sıra: rastgele üretilmiş imgeler, alt sıra: önerilen model kullanılarak yeniden yapılandırılan imgeler	60
Şekil 3.25 Gülme özniteliğine karşılık gelen gizli vektörün çıkarılması	61
Şekil 3.26 Gizli öznitelik vektörleri üzerine izdüşümler: sol-üst: gülme, sağ-üst: saç rengi, sol- alt: gözlük, sağ-alt: sakal özniteliği.....	64
Şekil 3.27 Dört farklı öznitelik için mavi ile boyanan dağılımın orta ve uç noktalarından alınan örnekler	65
Şekil 3.28 Saç rengi özniteliği için rastgele 5 farklı referans imge çifti kullanılarak elde edilen dağılımlar	66

Şekil 3.29 Dört öznitelik için farklı referans imge çiftleri kullanılarak öznitelik düzenleme. Sol üst çerçeve: gülme, sağ üst çerçeve: saç rengi, sol alt çerçeve: gözlük, sağ alt çerçeve: sakal	67
Şekil 3.30 k parametresi için değer aralığının belirlenmesi. Üst: farklı k değerleri için ten rengindeki değişimi, alt: sakal özniteliğindeki değişimi göstermektedir. Her bir özellik için kullanılan referans imgeler, her bir örneğin üstünde verilmiştir. $k = 0$ başlangıç imgesine karşılık gelmektedir	69
Şekil 3.31 Sol sütun: örnek girdi imgeleri, sağ sütun: dört öznitelik için, k 'nın farklı değerleri kullanılarak düzenlenmiş imgeleri göstermektedir	71
Şekil 3.32 Poz ayarlanması	73
Şekil 3.33 Gerçek imgelerde gülme özniteliği ve poz ayarlama.....	75
Şekil 3.34 Keyfi bir referans imge çifti kullanılarak eş zamanlı olarak gülme öznitelisinin kazandırılması ve poz ayarının yapılması. Üst sıra: referans imge çiftini, ikinci ve üçüncü sıra: referans öznitelik yönünde birden çok örneklemeyi göstermektedir	76
Şekil 4.1 Önerilen yaklaşım	86
Şekil 4.2 Maskelenen bölgeye ortalama piksel değeri atandığında elde edilen kaba imge.....	88
Şekil 4.3 Maskelenen bölgeye rastgele piksel değeri atandığında elde edilen kaba imge.....	89
Şekil 4.4 Rasgele düzensiz bir maske kullanılarak elde edilen kaba imge	89
Şekil 4.5 Ayırıştırıcı ağ mimarisi	90
Şekil 4.6 <i>Unet</i> modeli kullanılarak maskelenen bölge ile diğer bölgeler arasındaki geçiş sürekliliğinin sağlanması. Yeşil, mavi ve kırmızı kesik çizgilerle gösterilen bölgelere %300 yakınlaştırma uygulanmıştır. (1) - sol, (2) - sol ve (3) - yukarı imgeler CRG çıktıları, (1) - sağ, (2) - sağ, (3) - alt imgeler, <i>Unet</i> modeli kullanılarak işlem sonrası elde edilen imgelerdir	93
Şekil 4.7 Soldan-Sağa girdi imgesi, farklı maske türleri (düzenli, düzensiz veya elle çizilmiş), maskelenen imge (maskelenmiş bölgede farklı piksel değeri veya eskiz kullanılmıştır), kaba sonuç (n. CRG döngü sonucu), ve ince sonuç verilmiştir. Sağ alttaki blokta ise (k) örneği için tüm döngü sonuçları verilmiştir.....	96
Şekil 4.8 İstenmeyen bölgelerin düzeltilmesi	97
Şekil 4.9 Gerçek imgeler kullanılarak elde edilen sonuçlar.....	98
Şekil 4.10 Model sonuçlarının karşılaştırılması.....	100
Şekil 4.11 Önerilen yöntemin zayıflığı. (a) başarılı örnek (b) başarısız örnek. Her örnek için, üst sıra: her bir CRG döngüsünün sonucunu, alt sıra: ilgili döngü için Üretici uzayında üretilen imgeyi göstermektedir	102

ÇİZELGELER DİZİNİ

Çizelge 3.1 Üretici ağ katman ve detayları	39
Çizelge 3.2 Ayırıştırıcı ağ katman ve detayları.....	39
Çizelge 3.3 Modellerin kıyaslanması	43
Çizelge 3.4 Örnek sayısına bağlı olarak <i>FID</i> skorları.....	44
Çizelge 3.5 Gerçek imgelerle yeniden yapılandırılan imgelerin benzerlikleri	60
Çizelge 3.6 <i>CelebA</i> veri kümesinde tanımlanan tüm öznelilikler	72
Çizelge 4.1 <i>Unet</i> mimarisi	94



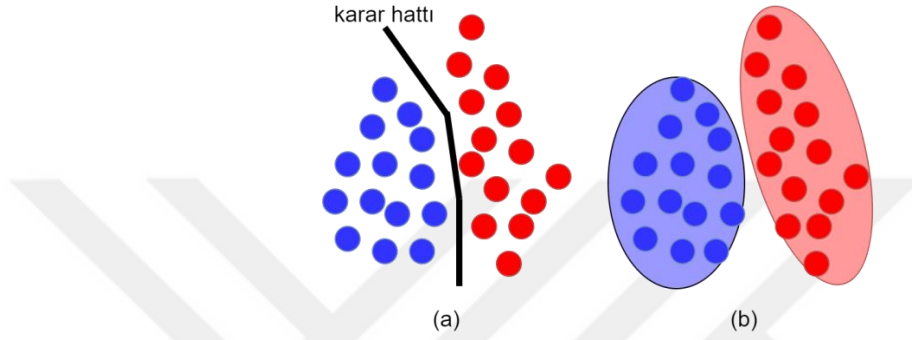
1. GİRİŞ

1.1 Genel Bilgi

Derin öğrenme; açık bir şekilde algoritması yazılamayan, verilerden öğrenmeyi sağlayan, yapay sinir ağları olarak adlandırılan beyin yapısından esinlenerek tasarlanmış, makine öğrenmesi biliminin bir alt dalıdır. Derin öğrenme algoritmaları, büyük boyutlu veri setleri ve işlem gücüne ihtiyaç duymaktadır. Günümüzde görüntü kaynaklarının artması, depolama birimlerinin büyümesi ve bilgi işlem gücündeki gelişmeler sayesinde çok büyük boyutlu veri setleri oluşturulmakta ve bu veri setleri kolay bir şekilde işlenebilmektedir. Bu gelişmeler bilgisayarla görme alanında geleneksel yöntemler ile çözüm aranan birçok problem (imge sınıflandırma, imge öznitelik düzenleme, imge içboyama, nesne tespiti, imge bölütleme vb.) için derin öğrenme modelleri kullanılarak umut verici sonuçların elde edilmesine olanak sağlamaktadır.

Derin öğrenme yöntemleri, Ayırt Edici Model (AEM)'ler (Yan vd. 2018, Tavakolian ve Hadid 2018) ve Üretken Modeller (ÜM)'ler (Frey vd. 1995, Kingma vd. 2013, Goodfellow vd. 2014) olmak üzere iki gruba ayrılmaktadır. AEM'lerde amaç veri setindeki farklı örnek gruplarını ayırmayı sağlayacak karar hatları belirlemektir. AEM'lerde eğitim işleminden sonra bir örneğin hangi sınıfta olduğuna karar vermek için bu hatlar kullanılmaktadır. Şekil 1.1 (a)'da iki farklı sınıf içeren bir sınıflandırma problemi için temsili bir örnek gösterilmektedir; model, iki sınıf arasında bir karar hattı oluşturmayı öğrenmekte ve sınıfı bilinmeyen örnekleri hattın hangi tarafına düştüğüne göre sınıflandırmaktadır. AEM'ler genellikle denetimli öğrenme görevlerinde kullanılmaktadır. Denetimli öğrenme, veri setinde her bir örneğin sınıfının; diğer bir deyişle etiketinin önceden bilinmesi anlamına gelmektedir. ÜM'ler ise veri setindeki her bir sınıfın olasılık dağılımını öğrenmektedir. Şekil 1.1 (b)'de etiketi bilinmeyen iki sınıfın, örneklerin benzerliği göz önünde bulundurularak gruplandırılması için örnek bir gösterim verilmektedir. ÜM'lerde eğitim işleminden sonra herhangi bir örneğin sınıfı, bu örneğin her bir örnek grubu için oluşturulmuş dağılımlardan üretilmiş olma olasılığına bakılarak belirlenmektedir. ÜM'ler veri seti için olasılık dağılımları

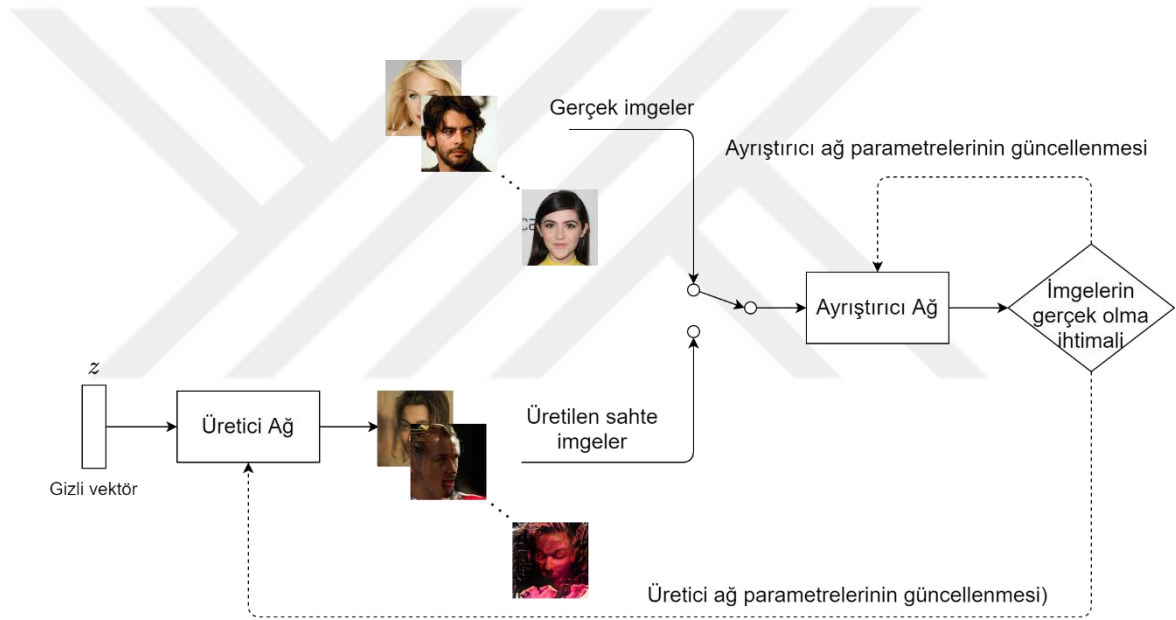
öğrendiğinden, veri setinden yeni örnekler oluşturmayı sağlamaktadır. ÜM'ler denetimsiz öğrenme görevlerinde kullanışlı olmaktadır. Denetimsiz öğrenme, veri setindeki örneklerin bilindiği; ancak örneklerin etiketinin bilinmediği durumu ifade etmektedir. Yakın zamana kadar derin öğrenme çalışmalarında ayırt edici modellerin daha fazla etkiye sahip olduğu ve ÜÇA mimarisinin önerilmesi ile araştırmacıların üretken modellere yöneldikleri gözlenmektedir.



Şekil 1.1 (a) ayırt edici model (b) üretken model

Üretken çekişmeli ağlar, mevcut veri setine benzer sentetik veriler oluşturmayı amaçlayan üretken bir modeldir. ÜÇA mimarisi, Üretici ve Ayırıştırıcı olmak üzere rekabet halindeki iki ağdan oluşmaktadır. Üretici ağ, Normal veya Düzgün dağılımdan P_z , rastgele örneklenen z gizli vektörünü $z \in P_z$ girdi olarak almakta ve bir x verisi üretmektedir. Ayırıştırıcı ağ ise, girdi olarak aldığı gerçek veya sentetik verinin gerçek olma ihtimalini vermektedir. Uygulamada Üretici ve Ayırıştırıcı ağları, genel olarak derin ağlar kullanılarak gerçekleştirilmektedir. Model eğitiminde ise Ayırıştırıcı ağın çıktısına göre Üretici ve Ayırıştırıcı ağların parametreleri güncellenmektedir. Nihai olarak iki ağ arasında bir denge kurulmakta ve Ayırıştırıcı ağ hem gerçek hem de üretilen örnekler için hemen hemen aynı olasılık değerini üretmektedir. Bu durum Üretici ağın gerçek veri setine benzer örnekler ürettiğini göstermektedir. ÜÇA mimarisi Şekil 1.2'de gösterilmiştir. Genel olarak, eğitim işlemi tamamlandıktan sonra Ayırıştırıcı ağ atılmakta ve Üretici ağ, veri setine benzer örnekler üretmek için kullanılmaktadır. Bu yaklaşım veri setine benzer örnek üretiminde başarılı sonuçlar elde edilmesine olanak sağladığı gibi, birçok probleme de çözüm sunmaktadır. Son zamanlarda ÜÇA'lar:

- İmge üretimi (Radford vd. 2016, Karras vd. 2019),
- İmge içboyama (Pathak vd. 2016, Lizuka vd. 2017),
- İmge öznitelik düzenleme (Choi vd. 2018, He vd. 2019),
- Kenar haritalarından imgelerin yapılandırılması (Isola vd. 2017, Zou vd. 2020),
- Metin tanımlamalarından imge sentezleme (Zhang vd. 2018, Yin vd. 2019),
- Düşük çözünürlüklü imgeden yüksek çözünürlüklü imge üretilmesi (Ledig vd. 2017, Shamsolmoali vd. 2019),
- İmge renklendirme (Zhang vd. 2016, Nazeri vd. 2018),
- İmgeden imgeye dönüşüm (Isola vd. 2017, Wang vd. 2018a) vb. gibi birçok problem alanında kullanılmaktadır.

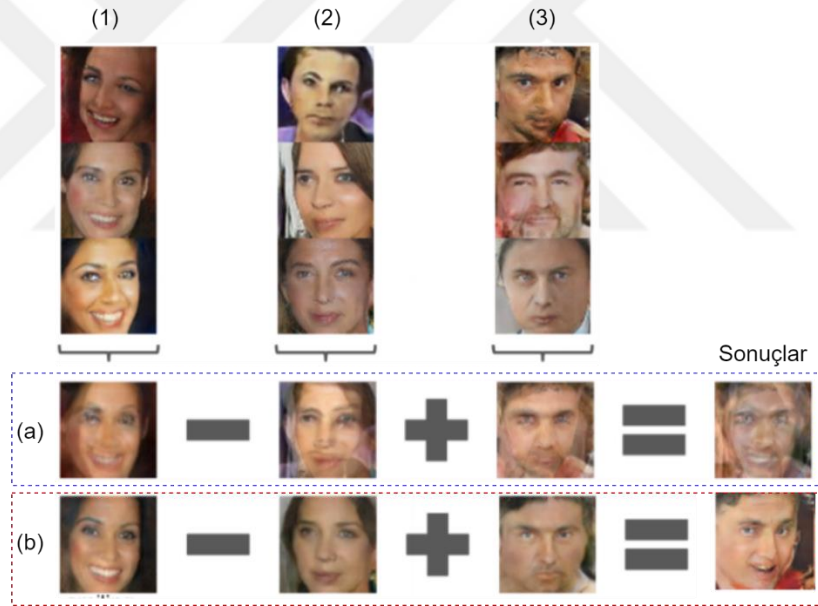


Şekil 1.2 ÜÇA mimarisi (Goodfellow vd. 2014)

1.2 Motivasyon ve Tez Kapsamı

Yakın zamanda Radford vd. (2016), ÜÇA'ların kullanımı ile imge uzayında anlamlı sonuçlar üretmeyen cebirsel işlemlerin, gizli vektör uzayında anlamlı sonuçlar ürettiğini belirtmiştir. Bu yapı bir imgedeki bir özniteliğin farklı bir imgeye uygulanmasına olanak sağlamaktadır. İlgili çalışmada yüz imgelerinden oluşan bir veri seti kullanılmaktadır. Bu veri seti için gülme, yaş, cinsiyet, sakal, makyaj vb. öznitelik

(attribute) olarak ifade edilmektedir. Şekil 1.3'te ilk kolon (1) gülme özniteliği taşıyan imgeleri, ikinci kolon (2) gülme özniteliği taşımayan imgeleri ve üçüncü kolon (3) ise öznitelik uygulanmak istenilen imgeleri göstermektedir. Şekil 1.3'te gülme özniteliği için (a)'da imge uzayı ve (b)'de gizli vektör uzayı için öznitelik değiştirme yöntemi ve sonuçları gösterilmektedir. İlgili çalışmada her bir kolondaki üç imgenin ortalaması alınmaktadır. (a)'da ilk kolondaki üç imgenin ortalaması sonucu elde edilen imgeden, ikinci kolondaki ortalama imge çıkarılmakta ve elde edilen sonuç üçüncü kolondaki ortalama imgeye uygulanmaktadır. (b)'de ise ilk kolondaki üç imgeyi temsil eden gizli vektörlerin ortalamasından, ikinci kolondaki ortalama gizli vektör çıkarılmakta ve üçüncü kolondaki ortalama gizli vektöre eklenmektedir. Şekil 1.3 incelendiğinde, imge uzayında bulanık ve anlamız bir sonuç elde edilmekte, gizli vektör uzayında ise gülmeyen bir yüz imgesinin gülen bir yüz imgesine dönüştüğü gözlenmektedir.



Şekil 1.3 İlk kolon (1) gülme özniteliği taşıyan imgeleri, ikinci kolon (2) gülme özniteliği taşımayan imgeleri, üçüncü kolon (3) gülme özniteliği uygulanmak istenen imgeleri temsil etmektedir. (a) çerçevesi imge uzayı ve (b) çerçevesi gizli vektör uzayında yapılan cebirsel işlemler ve sonuçları göstermektedir (Radford vd. 2016)

Tez çalışmasında Şekil 1.3'te gösterilen Radford vd. (2016)'nin yaptığı çalışmadan esinlenilmiştir. Bu çalışmada, bir özniteliği temsil eden vektörü bulmak için ilgili özniteliği içeren ve içermeyen üretilmiş imgeler kullanılmaktadır. Bu imgelerin önden bilinen gizli vektörleri arasında cebirsel işlem yapıldığında, imgeler arasındaki farklılık

sadece ilgili öznitelik ile sınırlı olmadığından, elde edilen öznitelik vektörü sadece bu özniteliği kodlamamaktadır; bu da özniteliğin uygulanmak istendiği imgede istenmeyen değişimlerin olmasına neden olmaktadır. Radford vd. (2016), bu farklılığı azaltmak için birkaç imgenin (Şekil 1.3'te üç tane imge kullanılmış) gizli vektörlerinin ortalamasını kullanmıştır; ancak bu durumda da istenmeyen değişimlerin olduğu gözlenmektedir. Bu problemin üstünden gelmenin yolu, ilgili öznitelik dışında diğer farklılıkların aynı olduğu bir imge çifti kullanmak ve bunların gizli vektörleri arasında cebirsel işlem uygulamaktır. Bunun için imgelerin gizli vektörlerini elde etmeyi sağlayacak bir yapıya ihtiyaç duyulmaktadır. Söz konusu çalışma Üretici ve Ayrıştırıcı ağlarından oluşmakta ve buna olanak sağlamamaktadır.

Bu doğrultuda tez çalışmasında, ÜÇA'larda imgeden gizli vektöre haritalamayı sağlayacak (Kodlayıcı ağ aracılığıyla) CRG olarak adlandırılan bir mimari önerilmiştir. Önerilen mimari kullanılarak imge öznitelik düzenleme ve yüz imgesi içboyama problemleri için etkili çözüm önerileri sunulmuştur.

İmge öznitelik düzenleme: İmge öznitelik düzenleme; bir imgede belirlenen bir öznitelik veya öznitelikleri istenilen şekilde değiştirip imgedeki diğer öznitelik ve detayları korumaktır. Tez kapsamında yüz imgelerinden oluşan bir veri seti kullanılmaktadır. Yüz imgesinde öznitelik düzenleme farklı amaçlar için kullanılmaktadır:

- İmgelerde istenmeyen durumların üstesinden gelmek için kullanılmaktadır. Örneğin; imgelerde kapalı çıkan gözlerin açılması, ağzın kapanması vb.
- Yüzde herhangi bir fiziksel değişim yapmadan önce muhtemel sonucu görmek için kullanılmaktadır. Örneğin; saç boyama, makyaj, saç stili vb.
- Herhangi bir suça karışmış birinin eşkâlinin belirlenmesinde kullanılabilir. Eşkâl, bir kişinin kimliğinin belirlenmesi için vücut veya yüz görüntüsünün tespit edilmesidir. Günümüzde kameralarının yoğun kullanımı ile birlikte suçluların görüntüleri rahatlıkla elde edilebilmektedir; ancak kişi insan yüzündeki değişken unsurlardan (saç, sakal, bıyık, şapka, gözlük vb.) yararlanarak kendisini

gizleyebilmektedir. Öznitelik düzenleme yöntemi ile tek bir yüz imgesinden yola çıkarak farklı varyasyonlara sahip imgeler elde edilebilmektedir.

Literatürde yüz imgesi düzenleme çalışmaları genel olarak tüm veri setinin etiketlenmesini (imgenin her bir özniteliği içerip içermediği hakkında bilgi) ve/veya her bir öznitelik için ayrı ayrı modellerin oluşturulmasını gerektirmektedir. Bu kısımda önerilen CRG mimarisi ile çok az sayıda etiket verisi (her bir öznitelik için ilgili özniteliği içeren ve içermeyen olmak üzere bir imge çifti) kullanılarak tek bir model vasıtasıyla çok sayıda özniteliği değiştirmeye olanak sağlayan bir yaklaşım önerilmektedir.

Yüz imgesi içboyama: İçboyamada amaç bir imgede eksik veya maskelenmiş bölgeleri anlamlı olarak doldurmaktır. Maske, imgede yeniden yapılandırılması istenen bölgeyi ifade etmektedir. İçboyama; hasar görmüş imgelerin onarılması, istenmeyen metin veya objelerin kaldırılması, maskelenen bir bölge için farklı içerik üretilmesi, kullanıcı yönlendirmelerine bağlı olarak içerik üretilmesi gibi farklı amaçlar için kullanılmaktadır. Bir imgede maskelenen bölge içeriğinde ne olması gerektiği ile ilgili standart bir yol olmaması ve maskelenen bölge için çok sayıda alternatif içerik olabilmesi nedeniyle zorlayıcı bir problem (ill-posed) olarak adlandırılmaktadır. Yüz imgesi içboyama ise, yüz içinde karmaşık ve birbiri ile ilişkili bölgeler bulunduğundan zorlayıcı bir problem olmaktadır. Örneğin, sadece bir gözün maskelendiği durumda, maskelenen bölgeye bir göz nesnesi oluşturmak yeterli olmamaktadır; maskelenen bölge için oluşturulan göz nesnesinin diğer gözle olan ilişkisinin de ele alınması gerekmektedir. Yakın zamanda bu problem uzayı için geliştirilen yöntemler maskelenen bölge için görsel olarak gerçekçi içerik doldurma görevinde dikkate değer bir başarı sağlamıştır. Fakat bu yöntemlerin her biri maskelenen bölgeyi doldururken farklı sınırlamalar veya kısıtlamalar içermektedir. Örneğin:

- Yüz imgesi içboyama alanında farklı maske türleri (düzenli, düzensiz ve elle çizilmiş) kullanılmaktadır. Genel olarak bu alanda önerilen modellerin belirli bir maske türünde başarılı olduğu, diğerlerinde ise istenen seviyede performans gösteremediği gözlenmektedir.
- Maskelenen bölge için tek ve sabit bir çıktıya izin verilmektedir.

- Maskelenen bölgede farklı içerik oluşturulması için kullanıcı tarafından çizilen eskizler (sketch) göz ardı edilebilmektedir.

Bu bağlamda, tez kapsamında CRG mimarisi kullanılarak yüz imgesi içboyama problemi için, belirtilen bu sınırlamalar ve kısıtlamaların üstesinden gelmeyi sağlayan bir yöntem önerilmiştir.

1.3 Tezin Özgün Değeri

Tez kapsamında önerilen CRG mimarisi ile hem imgeden gizli vektör uzayına, hem de gizli vektörden imge uzayına tek bir iterasyonda haritalama yapılabilir. Bu mimari, farklı problem alanları için çözüm sunmaktadır. Tez çalışmasında imge öznitelik düzenleme ve yüz imgesi içboyama problemleri için sağladığı avantajlar ve bu doğrultuda yapılan detaylı analizler ele alınmıştır. Bu kapsamda gerçekleştirilen katkılar aşağıdaki gibi listelenebilir.

- ÜÇA'larda yüz imgesi veri setinde (*CelebA*) istikrar ve çeşitliliği sağlamak için referans alınan model iyileştirilerek kullanılmıştır (Dogan ve Keles 2019).
- ÜÇA'larda imgeden gizli vektöre haritalamayı sağlayacak CRG mimarisi (Dogan ve Keles 2020) önerilmiştir.
- Öznitelik başına sadece tek bir gerçek referans imge çifti kullanılarak öznitelik vektörleri elde edilmiştir; bu öznitelik düzenleme problemi için, tüm veri setini etiketleme ihtiyacını ortadan kaldırmıştır.
- Farklı gerçek referans imgeler kullanılarak elde edilen öznitelik vektörlerinin tutarlılığı analiz edilmiştir.
- Herhangi bir imgeyi istenilen oranda değiştirmeyi sağlayan bir parametre tanımlanmış (k); bu parametre için güvenli bir aralık bulmayı sağlayacak bir yaklaşım önerilmiştir.
- Yüz imgesi içboyama problemi için yeni bir yöntem (Dogan ve Keles 2021) önerilmiştir.

1.4 Tezin Yapısı

Tez çalışmasının 2. Bölümünde, ÜÇA'lar ile ilgili genel bilgiler verilmiştir. Bu bölümde; ÜÇA'ların eğitim yöntemi, farklı ÜÇA mimarileri, eğitim aşamasında meydana gelen problemler, bu problemlerin üstesinden gelmek için son zamanlarda önerilen yaklaşımlar ve ÜÇA modelinin gelişiminin izlenmesi veya farklı modellerle karşılaştırılması için kullanılan metrikler ele alınmıştır. 3. Bölümde imgeden gizli vektöre haritalamayı sağlayan modeller ile ilgili literatür taraması, referans olarak alınan bir ÜÇA modelinin iyileştirilmesi için yapılan deneysel çalışmalar, önerilen CRG mimarisi ve CRG'nin benzer mimarilerle karşılaştırma sonuçları verilmiştir. Ayrıca bu bölümde, imge öznitelik düzenleme problemi için sunulan çözüm, bu kapsamda yapılan analizler, sonuçlar ve değerlendirmeler detaylı olarak verilmiştir. 4. Bölümde yüz imgesi içboyama problemi ile ilgili literatür taraması, önerilen yöntem ve benzer modellerle karşılaştırma sonuçları, önerilen modelin yetenekleri ve başarısız olduğu durumlar ele alınmış ve bu kapsamda değerlendirmelerde bulunulmuştur. 5. Bölümde ise tez çalışması boyunca elde edilen sonuçlar ile ilgili genel bir özet yapılmış, önerilerde bulunulmuş ve gelecekte yapılması planlanan çalışmalara değinilmiştir.

2. ÜRETKEN ÇEKİŞMELİ AĞLAR

Bu bölümde ÜÇA'lar ile ilgili genel bilgiler verilmiştir. Tez çalışmasında imgelerden oluşan veri setleri kullanıldığından, örneklemeler genel olarak imgeler üzerinden verilmektedir.

2.1 Üretken Modeller

Üretken modellerde hedef, mevcut veri setine yakın sentetik örnekler üretmektir; diğer bir deyişle, verilen bir veri setindeki örneklerin türetildiği dağılımı öğrenip bu dağılımdan yeni örnekler üretmektir. Genel olarak veri setleri sınırlı sayıda örnekler içerdiğinden, her zaman veri seti dağılımını tam olarak öğrenmek mümkün olmamaktadır. Bu nedenle gerçek veri dağılımına mümkün olduğunca benzer bir dağılım modellenmeye çalışılmaktadır. Uygulamada genel olarak modellenen dağılımı, gerçek dağılıma yaklaştırmak için sinir ağlarından yararlanılmaktadır.

Bu kısımda maksimum olabilirlik tahmini (maximum likelihood estimate) ilkesiyle çalışan üretken modellere odaklanılmaktadır; çünkü yakın zamanda en dikkat çekici yaklaşımlar bu ilkeyi referans almaktadır. Maksimum olabilirlik tahmini; veri setindeki örnekleri en iyi açıklayan, diğer bir deyişle veri setindeki örneklerin olabilirliğini en üst düzeye çıkaran, olasılık dağılımını tahmin eden model parametrelerini tanımlamayı hedeflemektedir. Gerçek olasılık dağılım parametreleri θ ve örneklerin olabilirliğini maksimize eden parametreler $\hat{\theta}$ ile temsil edilmesi durumunda, $\hat{\theta}$ Denklem 2.1 kullanılarak belirlenmektedir. $\hat{\theta}$ parametreleri, θ 'nın maksimum olabilirlik tahmincisi olarak adlandırılmaktadır.

$$\hat{\theta} = \operatorname{argmax}_{\theta} \prod_{i=1}^n p_{model}(x_{(i)}; \theta) \quad (2.1)$$

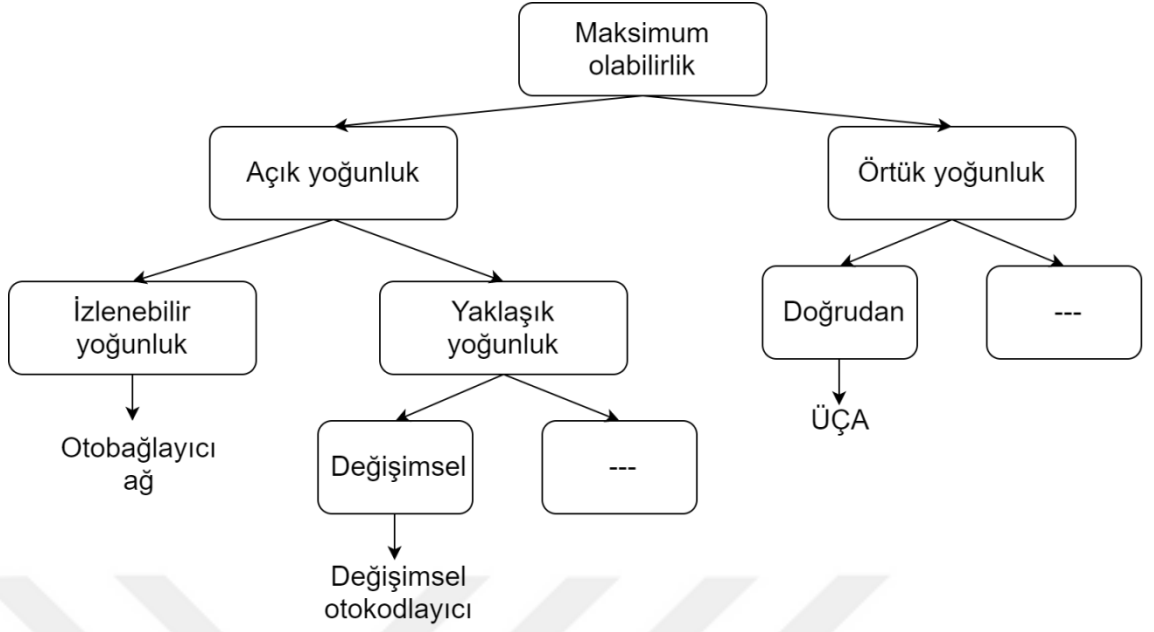
Denklem 2.1'de p_{model} olasılık dağılımını tahmin eden modeli, $x_{(i)}$ veri setindeki her bir örneği, n ise veri setindeki toplam örnek sayısını temsil etmektedir. Uygulamada bu işlemler genellikle iki sebepten dolayı logaritmik uzayda yapılmaktadır:

- Örneklerin olabilirliklerinin çarpımından ziyade, modelin logaritmasını alıp olabilirliklerin toplamına dönüştürmek işlem kolaylığı sağlamaktadır. Bu üstel fonksiyon içeren modellerin hesaplama yükünü oldukça azaltmaktadır. Ayrıca dijital ortamda, küçük olabilirlik değerlerinin çarpımından kaynaklanan alttan taşma (underflow) probleminden kaçınmayı sağlamaktadır.
- Çarpımların toplam işlevine dönüştürülmesi, olabilirliğin türevleri için cebirsel ifadeleri basitleştirmektedir.

Herhangi bir veri seti için olasılık dağılımını tahmin eden Denklem 2.1'deki denklemin logaritması alındığında Denklem 2.2 elde edilmektedir (Goodfellow 2016a).

$$\hat{\theta} = \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^n \log \rho_{\text{model}}(x_{(i)}; \theta) \quad (2.2)$$

Karmaşık veri setlerinin türetildiği dağılım parametrelerini belirlemek zorlu bir problemdir. Bu nedenle, genel olarak yaklaşık dağılım parametreleri belirlenmeye çalışılmakta ve bu doğrultuda sinir ağlarından yararlanılmaktadır. Yakın zamanda maksimum olabilirlik ilkesi ile çalışan ve sinir ağlarından yararlanan en dikkat çekici yaklaşımlara otobağlayıcı ağlar (autoregressive networks) (Frey vd. 1995, Frey vd. 1998), değişimsel otokodlayıcılar (variational autoencoders (VAE)) (Kingma vd. 2013, Rezende vd. 2014) ve ÜÇA'lar (Goodfellow vd. 2014) örnek olarak verilebilir. Şekil 2.1'de yoğunluk tahminine (density estimation) göre sınıflandırma yapıldığında bu modellerin hiyerarşideki konumları gösterilmektedir. ÜÇA'larda diğer modellerden farklı olarak, yoğunluk fonksiyonu açık olarak tanımlanmamakta ve modelden örnekleme yapılarak eğitim işlemi gerçekleştirilmektedir.



Şekil 2.1 Üretken modellerin sınıflandırılması (Goodfellow 2016a)

Maksimum olabilirlik ilkesiyle çalışan üretken modeller, olabilirliği nasıl temsil ettiklerine veya yaklaşık olarak nasıl tahmin ettiklerine göre farklılık göstermektedir. Hiyerarşinin sol tarafındaki modeller açık (explicit) bir yoğunluk fonksiyonu tanımlamakta ve bu açık fonksiyonun olabilirliğini eniyilemeye çalışmaktadır. Açık yoğunluk modelleri; yoğunluğun hesaplamalı olarak izlenebildiği (polinom zaman algoritması ile çözülebilen, alt sınırı polinom olan problemler) veya izlenemediği (bir polinom zaman algoritması ile çözülemeyen, alt sınırı üstel olan problemler) olmak üzere iki kısma ayrılmaktadır. Yoğunluğun izlenebilir olmadığı (yaklaşık yoğunluk) modeller de olabilirliği en üst düzeye çıkarmak için değişimsel tahmin veya farklı yaklaşımlar (örneğin, *Markov* zinciri yaklaşımı (Hinton vd. 1984, Hinton 2007)) kullanmalarına göre sınıflandırılmaktadır. Hiyerarşinin sağ tarafında bulunan örtük yoğunluk modellerinde ise model için açık bir olasılık dağılımı temsil edilmemekte ve olasılık dağılımı ile daha az doğrudan etkileşim kurmanın yolu aranmaktadır; bu modeller gerçek dağılımdan örnek alabilen bir fonksiyon elde etmeye çalışmaktadır.

2.1.1 Otobağlayıcı ağlar

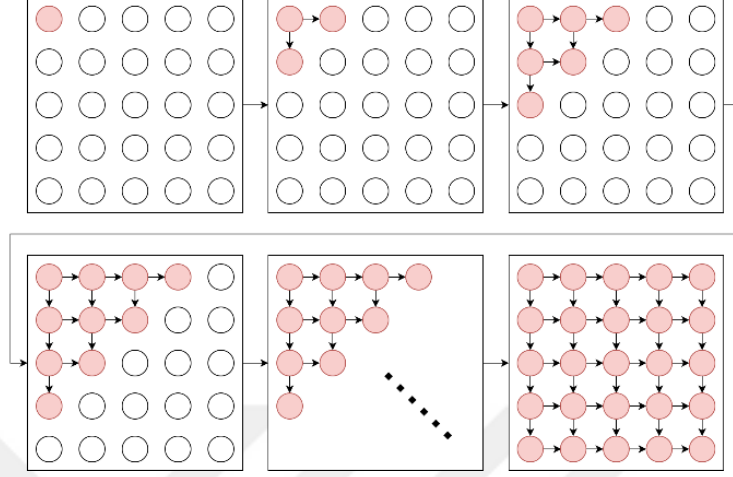
Otobağlayıcı ağlar yapısında gizli vektör içermeyen olasılıksal modellerdir. Bu ağlar zincir olasılık kuralını kullanarak n boyutlu bir x vektörü için ortak olasılık dağılımını (joint probability distribution), tek boyutlu koşullu olasılık dağılımlarının çarpımına dönüştürmektedir. Bu modeller tamamen görülebilir inanç ağları (fully visible belief networks) (Frey vd. 1995, Bengio ve Bengio 2000) olarak adlandırılmaktadır. Veri setinin imgelerden oluştuğu varsayıldığında, bir x imgesinin olabilirliğini bulmak için zincir kuralı kullanılarak x imgesi Denklem 2.3'te verildiği gibi 1-boyutlu koşullu olasılıkların çarpımları şekline dönüştürülmektedir. Uygulamada imgenin ardışık bir diziye dönüştürülmesi farklılık gösterebilmektedir. Satır tabanlı, kolon tabanlı veya diyagonal olacak şekilde farklı yöntemler kullanılmaktadır.

$$p_{model}(x) = \prod_{i=1}^n p_{model}(x_i | x_1, x_2 \dots x_{i-1}) \quad (2.3)$$

Denklem 2.3'te $p_{model}(x)$ bir x imgesinin olabilirliği, $p_{model}(x_i | x_1, x_2 \dots x_{i-1})$ ise $x_1, x_2 \dots x_{i-1}$ pikselleri verildiğinde i . pikselin olasılığını temsil etmektedir. Bu modellerdeki koşullu olasılık dağılımlar sinir ağları ile ifade edilmektedir (Goodfellow vd. 2016b). Yakın zamanda bu modeller kullanılarak yapılan en dikkat çekici yaklaşımlara PixelRNN (Oord vd. 2016a), PixelCNN (Oord vd. 2016b) ve Wavenet (Oord vd. 2016c) çalışmaları örnek olarak verilebilir. Şekil 2.2'de PixelRNN çalışmasında, her bir pikselin olasılığının hesaplanmasında bir önceki piksellere olan bağıllık temsili olarak gösterilmiştir. Şekil 2.2'de gösterildiği gibi bir pikselin olasılığını hesaplayabilmek için kendisinden önceki piksellerin belirlenmiş olması gerekmektedir. Bu işlem için de piksellerin belirlenme sıralamasının tanımlanması gerekmektedir. Şekil 2.2'de sol üst köşeden başlayarak sağ alt köşeye doğru diyagonal olarak diğer piksellerin olasılığı belirlenmektedir.

Özet olarak otobağlayıcı ağlar; açık bir şekilde bir örneğin olabilirliğini hesaplamaya olanak sağlamakta, eğitim verilerinin açık bir şekilde olabilirliğini hesaplamak iyi bir değerlendirme ölçütü sağlamakta ve iyi örneklerin üretilmesine olanak sağlamaktadır. Fakat bu modeller ardışık olarak piksel üretimi yaptığından yavaş çalışmakta ve

yapısında gizli vektör barındırmadığından, üretken modellerin sağladığı bazı avantajları sağlayamamaktadır.



Şekil 2.2 PixelRNN çalışması diyagonal koşullu olasılıkların belirlenmesi için temsili gösterim. Sol üst köşeden sol alt köşeye doğru ardışık olarak piksel üretimi yapılmaktadır. Renkli olanlar üretilen pikselleri temsil etmektedir

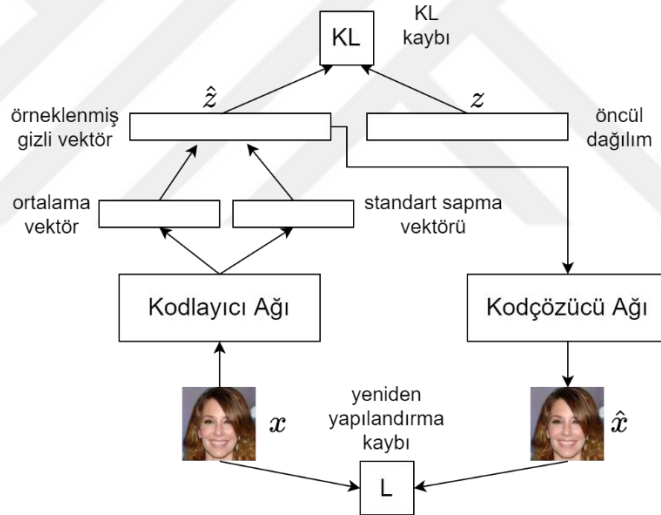
2.1.2 Değişimsel otokodlayıcılar

Değişimsel otokodlayıcılar; izlenebilir yoğunluk fonksiyonlarına sahip modellerin tasarım gerekliliklerinin getirdiği bazı dezavantajlardan (örneğin, ardışık piksel üretimi) kaçınmak için, yaklaşık çıkarımı ağı kullanan ve yalnızca gradyan tabanlı yöntemler ile eğitilebilen modellerdir. VAE’ler olasılıksal bir Kodlayıcı ağı (yaklaşık çıkarım ağı) ve bir olasılıksal Kodçözücü ağı içermektedir. Kodlayıcı ağı ($q_\phi(z|x)$) bir veri örneği x ’i girdi olarak almakta ve z ile gösterilen ayrık bir gizli vektöre gömmektedir. Bu model gizli vektör üreten model olarak da adlandırılmaktadır. Kodçözücü ağı ($p_\theta(x|z)$) ise gizli vektör z ’yi girdi olarak almakta ve x örneğini yeniden yapılandırmaktadır. VAE’lerde kullanılan kayıp fonksiyonu Denklem 2.4’te verilmiştir.

$$L(\theta, \phi; x_i) = \mathbb{E}_{z \sim q_\phi(z|x_i)} \log(p_\theta(x_i|z)) - KL(q_\phi(z|x_i) || p_\theta(z)) \quad (2.4)$$

Denklem 2.4’te x_i gerçek veri dağılımından alınan i . örneği, ϕ ve θ sırasıyla Kodlayıcı-Kodçözücü modellerinin parametrelerini, denklemin ilk kısmı yeniden yapılandırma

kaybını, denklemin ikinci kısmı ise yaklaşık ardıl dağılım (örneğin, diyagonal kovaryans yapısına sahip çok değişkenli bir *Gauss*) ile öncül dağılımı (örneğin, ortalananmış izotropik çok değişkenli *Gauss*) birbirine yakınlaştırmaya çalışan *Kullback–Leibler (KL)* (Kullback ve Leibler 1951) ıraksamasını ifade etmektedir. Şekil 2.3'te imge verisetlerinde kullanılan VAE mimarisi verilmiştir. Kodlayıcı ağı girdi olarak aldığı x imgesini ortalama ve standart sapma vektörlerine dönüştürmekte ve yaklaşık ardıl dağılım kullanılarak örneklenmiş gizli vektör elde edilmektedir. Daha sonra bu gizli vektör Kodçözücü ağına verilerek imge yeniden oluşturulmaktadır. Şekil 2.3'te görüldüğü üzere, imge uzayında yeniden yapılandırma kaybı ve gizli vektör uzayında *KL* kaybı kullanılarak Kodlayıcı ve Kodçözücü modellerinin parametreleri güncellenmektedir. Yakın zamanda bu yaklaşım kullanılarak yapılan en dikkat çekici yaklaşımlara Kingma vd. (2016) ve Chen vd. (2016) çalışmaları örnek olarak verilebilir.



Şekil 2.3 Değişimsel otokodlayıcı mimarisi. Kodlayıcı ağı, imgeleri girdi olarak alıp ortalama ve standart sapma vektörüne dönüştürmektedir. Kodçözücü ağı ise örneklenmiş gizli vektörü girdi olarak almakta ve imgeyi yeniden oluşturmaktadır. Mimaride, gizli vektör uzayında *KL* kaybı, imge uzayında ise yeniden yapılandırma kaybı kullanılmaktadır.

VAE'ler, otobağlayıcı ağlar ile kıyaslandığında uygulamada daha kolay olarak gerçekleştirilmekte ve pikseller ardışık olarak üretilmediğinden test aşamasında daha hızlı imge üretimi yapılabilmektedir. VAE'ler yapısında gizli vektör içermekte ve bu vektörden rastgele örnekleme yapılarak veri setine benzer imgeler üretilebilmektedir; ayrıca yapısında Kodlayıcı ağı barındırdığından imgeden gizli vektöre dönüşüm de

yapılabilmektedir. VAE'lerin temel dezavantajı ise imge veri seti kullanılarak eğitilen modellerde, üretilen imgelerin bulanık olması ve bu duruma neyin sebep olduğunun bilinmemesidir (Goodfellow vd. 2016b).

2.1.3 Üretken çekişmeli ağlar

Üretken çekişmeli ağlar (Goodfellow vd. 2014) yapılarında anlamlı bir gizli vektör barındırmaları ve kaliteli imgeler üretebilmeleri açısından diğer üretken modellerden farklılaşmaktadır (Karras vd. 2017). ÜÇA'lar bir yoğunluk fonksiyonunun açıkça tanımlanmasına gerek duyulmadan, modelden örnekleme yapılarak eğitilmektedir; diğer bir deyişle model örtük (implicit) olarak yoğunluk fonksiyonunu öğrenmektedir. ÜÇA'larda bir amaç fonksiyonunu eniyilemek yerine Ayırıştırıcı ve Üretici ağları arasında iki oyunculu bir oyun kurulmakta ve bu oyunun *Nash* dengesi (Kreps 1989) bulunmaya çalışılmaktadır; ancak bu işlev bir amaç fonksiyonunu eniyilemekten daha zorlu olabilmektedir. Bu nedenle eğitim aşamasında model çoğu zaman istikrarsız olmaktadır. ÜÇA mimarisi Şekil 1.2'de verilmiştir. Daha önce de belirtildiği gibi ÜÇA mimarisi, birbiri ile rekabet halindeki iki parametrelendirilmiş ağdan (Üretici ve Ayırıştırıcı) oluşmaktadır. Türevlenebilir bir fonksiyon olan Üretici ağı $\hat{U}$, öncül bir dağılımdan örneklenen z gizli vektörünü girdi olarak verilmekte ve x örneği ($x = \hat{U}(z)$) üretilmektedir. Türevlenebilir Ayırıştırıcı ağı ise girdi olarak gerçek veya sahte imgeler verilmektedir. Ayırıştırıcı ağı kullanılan kayıp fonksiyonuna göre girdi olarak verilen imge için bir skor veya imgenin gerçek olma ihtimalini vermektedir.

ÜÇA eğitim aşamasında iki senaryo söz konusudur. İlk senaryoda, gerçek veri setinden, $x \in P_{veri}$ ve rastgele üretilen imgelerden, $x' = \hat{U}(z)$ ve $z \in P_z$ örnekler alınarak Ayırıştırıcı ağı girdi olarak verilmektedir. Bu senaryoda Ayırıştırıcı ağın amacı, gerçek x ve sahte imgeleri x' birbirinden ayırt etmektir. Diğer bir deyişle, Ayırıştırıcı ağ Denklem 2.5'i enbüyütmeye (maximization) çalışmaktadır.

$$\max_A D(A) = \mathbb{E}_{x \sim P_{veri}(x)} \log(A(x)) + \mathbb{E}_{z \sim P_z(z)} \log(1 - A(\hat{U}(z))) \quad (2.5)$$

Denklem 2.5'te z , herhangi bir öncül dağılımdan rastgele örneklenen gizli vektör, A Ayırıştırıcı ve $\tilde{U}$ Üretici ağını temsil etmektedir. Bu senaryoda sadece Ayırıştırıcı ağı parametreleri eniyilenmektedir. Uygulamada, Ayırıştırıcı ağı eğitimi farklılık göstermektedir. Denklem 2.5'te gösterildiği gibi gerçek ve sahte imgeler tek bir yığında (batch) ağa girdi olarak verilebildiği gibi gerçek ve sahte imgeler farklı yığınlarda da girdi olarak verilebilmektedir. Bu durumda, eğitim aşamasında her yinelemede, Üretici ağı parametreleri bir kez eniyilenirken, Ayırıştırıcı ağı parametreleri birden fazla eniyilenmektedir.

İkinci senaryoda ise Üretici ağ kullanılarak rastgele imgeler üretilmekte ve Ayırıştırıcı ağa girdi olarak verilmektedir. Temel ÜÇA çalışmasında, Ayırıştırıcı ağ bir *sigmoid* birim ile bittiğinden, 0 ile 1 arasında çıktı üretmektedir. Üretici ağı amacı, Ayırıştırıcı ağını kandırmaktır; Ayırıştırıcı ağı, üretilen imgeler için yüksek skorlar üretmesini sağlamaktır. Diğer bir deyişle, Üretici ağ Denklem 2.6'yı enküçültmeye (minimization) çalışmaktadır. Bu senaryoda ise sadece Üretici ağı parametreleri eniyilenmektedir.

$$\min_{\tilde{U}} D(\tilde{U}) = \mathbb{E}_{z \sim P_Z(z)} \log(1 - A(\tilde{U}(z))) \quad (2.6)$$

Özet olarak, Üretici ve Ayırıştırıcı ağları, Denklem 2.7'de verilen değer fonksiyonu $D(\tilde{U}, A)$ 'yı kullanarak iki oyunculu enküçük-enbüyük (minimax) oyununu oynamaktadır (Goodfellow vd. 2014).

$$\min_{\tilde{U}} \max_A D(A, \tilde{U}) = \mathbb{E}_{x \sim P_{veri}(x)} \log(A(x)) + \mathbb{E}_{z \sim P_Z(z)} \log(1 - A(\tilde{U}(z))) \quad (2.7)$$

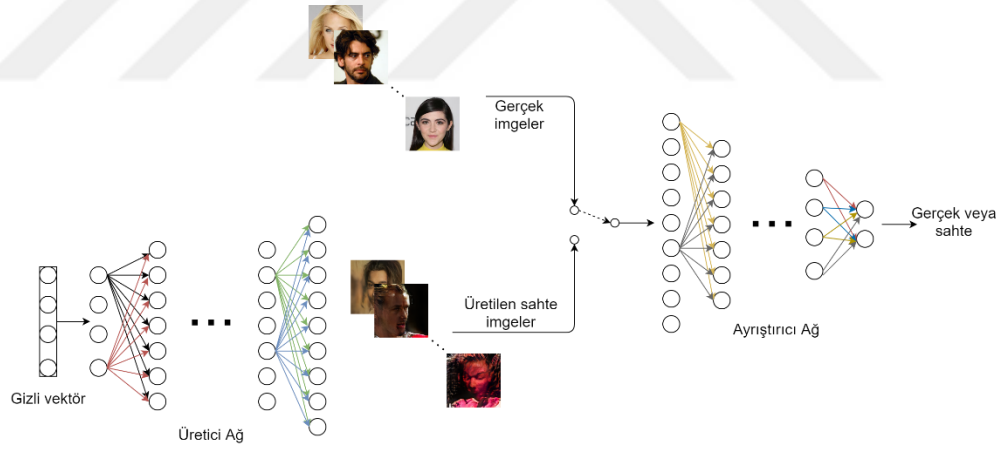
Nihai olarak, bu rekabet sonucu her iki ağ arasında bir *Nash* dengesi kurulmakta ve Ayırıştırıcı ağ girdi olarak aldığı herhangi bir imge için 0.5 değerine yakın bir olasılık değeri üretmektedir (*sigmoid* birim ile biten bir ağ kullanıldığında). Bu durum, üretilen örneklerin gerçek veri seti ile aynı dağılımdan geldiğini göstermektedir (Goodfellow 2016a).

2.2 Üretken Çekişmeli Ağ Mimarileri

Bu kısımda tez çalışması esnasında yararlanılan veya bahsi geçen mimariler ele alınmış ve her mimarinin özet olarak güçlü ve zayıf noktaları belirtilmiştir.

2.2.1 Temel ÜÇA mimarisi

Temel ÜÇA mimarisi ilk olarak Goodfellow vd. (2014) tarafından önerilmiştir. Tam bağlantılı ÜÇA mimarisi olarak da adlandırılan bu mimaride, hem Üretici hem de Ayırıştırıcı ağları için tam bağlantılı katmanlar kullanılmaktadır. İlgili çalışmada *MNIST* (LeCun vd. 2010), *Toronto Face Database* (Susskind vd. 2010) ve *CIFAR-10* (Krizhevsky ve Hinton 2009) veri setleri kullanılmıştır. Bu veri setleri imgelerden oluşmaktadır. Bu nedenle uygulamada bu imgeler tek boyutlu piksel dizisine dönüştürülmektedir. Şekil 2.4'te tam bağlantılı ÜÇA mimarisi verilmiştir.



Şekil 2.4 Tam bağlantılı ÜÇA mimarisi (Goodfellow vd. 2014). Mimari, çok katmanlı algılayıcı (multilayer perceptrons) yapısında kurulmakta olup imgeler piksel dizisine dönüştürüldükten sonra girdi olarak verilmektedir. Karmaşıklığı önlemek amacıyla nöronlar arasındaki tüm bağlantılar çizilmemiştir

Mimaride Ayırıştırıcı ağın son katmanında *sigmoid* geri kalan katmanlarında *maxout* (Goodfellow vd. 2013) aktivasyon fonksiyonu, Üretici ağında ise *Rectified Linear Unit (ReLU)* (Jarrett vd. 2009, Glorot vd. 2011) ve *sigmoid* aktivasyon fonksiyonları karışık olarak kullanılmaktadır. Ayırıştırıcı ağın eğitiminde aşırı öğrenmeden (overfitting)

kaçınmak için iletim sönümü (dropout) (Hinton vd. 2012) düzenlileştirme (regularization) katmanı uygulanmaktadır. Eğitim esnasında Ayırıştırıcı ağı k defa eniyilenirken, Üretici ağı bir defa eniyilenmektedir; bu Üretici ağı yavaş değiştiği durumda, Ayırıştırıcı ağı en uygun (optimal) çözüme yakın tutulmasını sağlamaktadır.

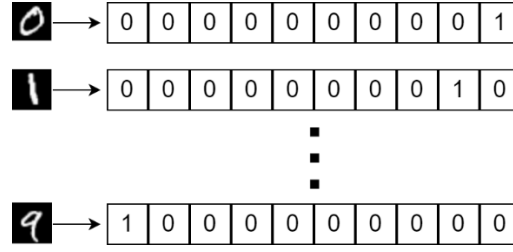
Bu mimari ÜÇA’larda ilk çalışma olmasından dolayı önem arz etmektedir. Ancak tam bağlantılı katman kullanımından dolayı imge çözünürlüğü arttıkça parametre sayısı çok artmakta ve bu da eğitim işlemini zorlaştırmaktadır. Bu nedenle ilgili çalışmada düşük çözünürlüklü (64x64 piksel çözünürlüğünden daha düşük) imge veri setleri kullanılmıştır.

2.2.2 Koşullu ÜÇA mimarisi

Koşullu ÜÇA (Conditional GAN) mimarisi Mirza ve Osindero (2014) tarafından önerilmiştir. Bu mimaride Üretici ve Ayırıştırıcı ağlarına koşul olarak ilave bir bilgi, y , verilmektedir. Bu ilave bilgi; sınıf etiketi, metin veya eskiz gibi problem alanına göre değişkenlik göstermektedir. İlgili çalışmada *MNIST* veri seti kullanılmış ve y sınıf etiketini temsil etmektedir. Denklem 2.8’de koşullu ÜÇA mimarisinde kullanılan amaç fonksiyonu verilmiştir.

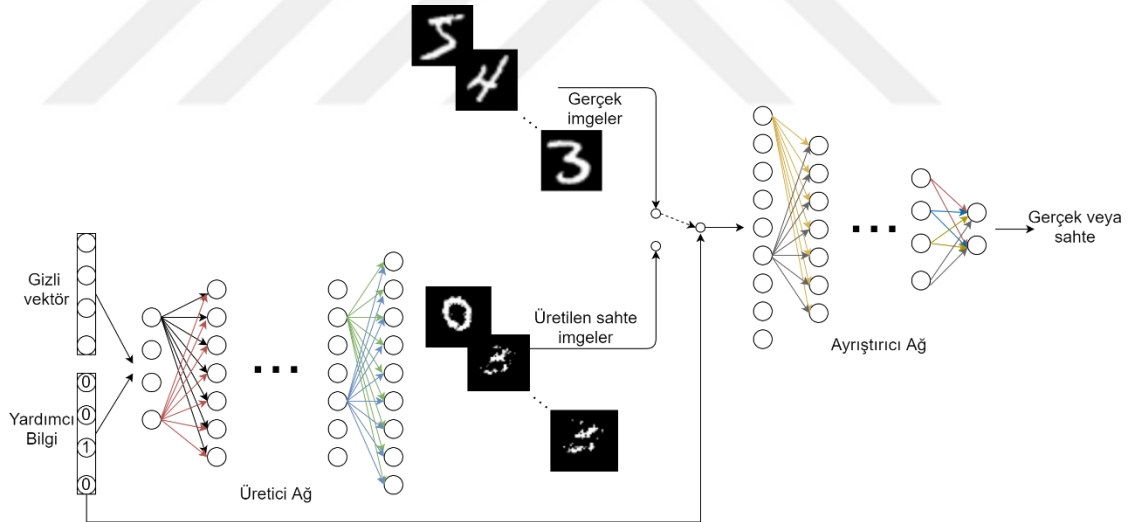
$$\min_{\tilde{U}} \max_A D(A, \tilde{U}) = \mathbb{E}_{x \sim P_{veri}(x)} \log(A(x|y)) + \mathbb{E}_{z \sim P_z(z)} \log(1 - A(\tilde{U}(z|y))) \quad (2.8)$$

Denklem 2.8’de A Ayırıştırıcı ağına, $\tilde{U}$ Üretici ağına, z bir öncül dağılımdan örneklenen gizli vektöre, y ise koşula karşılık gelmektedir. Bu çalışmada sınıf etiketi bir-elemanı-bir olan vektöre (one-hot vector) kodlanmakta ve her iki ağa girdi olarak verilmektedir. Şekil 2.5’te *MNIST* veri setindeki bazı örneklerin bir-elemanı-bir olan vektöre dönüştürme işlemi gösterilmiştir. Şekil 2.5’te görüldüğü gibi vektör boyutu sınıf sayısı kadar olmakta ve her vektörde ilgili sınıfı temsil eden bir “1” bulunmaktadır.



Şekil 2.5 Sınıf etiketinin bir-elemanı-bir olan vektöre dönüştürülmesi

Şekil 2.6’da koşullu ÜÇA mimarisi verilmiştir. Temel ÜÇA mimarisinde Üretici ağına herhangi bir öncül dağılımdan örneklenen bir gizli vektör, z , girdi olarak verilmekteyken, bu mimaride gizli vektör bir yardımcı bilgi, y , ile birleştirilerek girdi olarak verilmektedir. Bu birleştirme işlemi uygulamada farklılık gösterebilmektedir. Örneğin, orijinal çalışmada bu girdiler ilk başta bağımsız olarak ara bir katmana haritalanıp aktivasyon fonksiyonundan geçirildikten sonra birleştirilmiş ve Üretici ağına girdi olarak verilmiştir.



Şekil 2.6 Koşullu ÜÇA mimarisi (Mirza ve Osindero 2014). Mimaride, girdi imgesinin dışında ilgili imge ile ilgili yardımcı bir bilgi hem Üretici hem de Ayrıştırıcı ağına girdi olarak verilmektedir

Bu mimari genel olarak etiket bilgisinin sağlanması durumunda, istenilen koşul doğrultusunda imge üretimi, imge öznetelik düzenleme vb. problemlerde kullanılmaktadır.

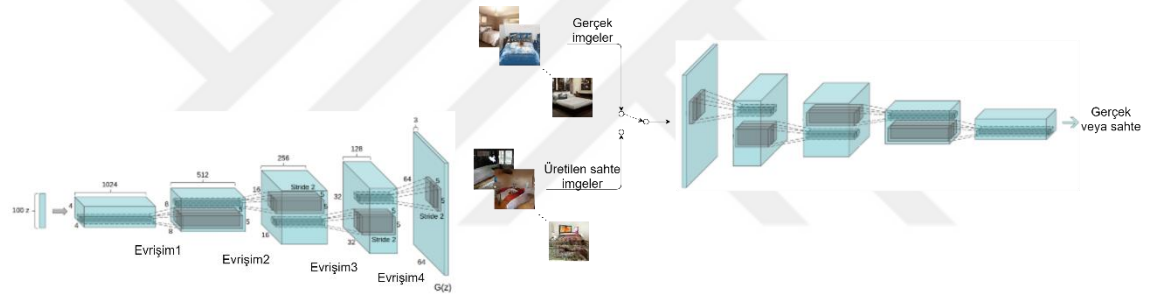
2.2.3 Derin evrişimsel ÜÇA

Derin Evrişimsel ÜÇA (Deep Convolutional GAN (*DCGAN*)) mimarisi Radford vd. (2016) tarafından önerilmiştir. Daha önce belirtildiği gibi temel ÜÇA çalışması hem Üretici hem de Ayırıştırıcı ağı için tam bağlantılı sinir ağlarını kullanmaktadır. Bu yapı imge çözünürlüğü arttığında, parametre sayısının çok fazla artmasına ve eğitim işleminde problemlerin oluşmasına sebep olmaktadır. Bu nedenle temel ÜÇA mimarisi, yüksek çözünürlüklü imge veri setlerinde iyi genelleme performansı gösterememektedir. *DCGAN* çalışmasından önce ÜÇA'ların daha yüksek çözünürlüklü veri setlerinde çalışmasını sağlamak için, denetimli öğrenme çalışmalarında önemli başarı sağlayan Evrişimsel Sinir Ağ (ESA)'lar kullanılmış; ancak bu çalışmalarda başarı elde edilememiştir (Radford vd. 2016).

Radford vd. (2016) Üretici ağında, ESA'lardaki katmanların özelliklerini görselleştirmek için Zeiler ve Fergus (2014) tarafından önerilen ters evrişimsel (deconvolution) katmanına benzer kesirli kademeli evrişimsel (fractional strided convolution) katmanını kullanmayı önermiştir. Bu katman ile Üretici ağında tam bağlantılı katman kullanma gereksinimi ortadan kalkmıştır; bu da daha yüksek çözünürlüklü imge üretimine olanak sağlamıştır. *DCGAN* çalışması ÜÇA'larda bir dönüm noktasıdır ve bu çalışmada ÜÇA'larda ilk defa kullanılan kesirli kademeli evrişimsel (yakın zamandaki çalışmalarda aralarında çok fazla fark olmadığından ters evrişimsel katmanı daha çok kullanılmaktadır) hemen hemen tüm ÜÇA çalışmalarında standart hale gelmiştir. Ayrıca *DCGAN* çalışmasında yüksek çözünürlüklü modelleme ve daha istikrarlı eğitim sağlamak için bazı kritik önerilerde bulunulmuştur:

- İlk olarak Ayırıştırıcı ağın kendi uzaysal alt örneklemesini (downsampling) öğrenmesi için, uzaysal biriktirme katmanları (örneğin, en büyükleri biriktirme (max-pooling)) kademeli evrişimsel (strided convolution) katmanlarıyla, Üretici ağın ise kendi uzaysal üst örneklemesini (upsampling) öğrenmesini sağlamak için üst örnekleme katmanları kesirli kademeli evrişimsel (fractional strided convolution) katmanlarıyla değiştirilmiştir.

- İkinci olarak hem Ayırıştırıcı hem de Üretici ağlarında yığın normalleştirme (batch normalization) (Ioffe ve Szegedy 2015) katmanları kullanılmıştır. Bu, hem üretilen hem de gerçek örneklerin sıfır ortalama ve birim varyansa sahip olacak şekilde normalleştirilmelerini sağlamaktadır; diğer bir deyişle, üretilen ve gerçek örnekler için benzer istatistiklerin kullanılmasına olanak sağlanmaktadır.
- Üçüncü olarak genellikle evrimsel katmanların sonunda kullanılan tam bağlantılı katmanlar yerine global maksimum havuzlama (global max-pooling) katmanlarının kullanılması önerilmiştir; bu katmanın daha derin modellerde istikrarı arttırdığı ancak yakınsama hızında yavaşlamaya neden olduğu belirtilmiştir.
- Son olarak Üretici ağının son katmanında *tanh* ve geri kalan tüm katmanlarda *ReLU* aktivasyon fonksiyonu, Ayırıştırıcı ağının tüm katmanlarında ise *Leaky Rectified Linear Unit (LReLU)* (Maas vd. 2013) aktivasyonu kullanılması önerilmiştir.



Şekil 2.7 DCGAN mimarisi (Radford vd. 2016). Mimaride, hem Üretici hem de Ayırıştırıcı ağında evrimsel katmanlar kullanılmaktadır. Sol taraf: Üretici ağını, Sağ taraf: Ayırıştırıcı ağını temsil etmektedir

Şekil 2.7’de DCGAN mimarisi verilmiştir. DCGAN; LSUN (Yu vd. 2015), ImageNet (Deng vd. 2009) ve özelleştirilmiş bir yüz imge veri seti kullanılarak 64x64 çözünürlüğünde eğitilmiştir. Temel ÜÇA mimarisi ile karşılaştırıldığında imge çözünürlüğü daha yüksek ve imge kalitesinin daha iyi olduğu belirtilmiştir; ancak eğitim esnasında istikrar (stability) probleminin devam ettiği ve eğitimin uzun tutulması durumunda imge çeşitliliğinin (mode collapse) azaldığı belirtilmiştir.

ÜÇA’larda ESA’ların başarılı bir şekilde kullanılması ile birlikte daha önceden tanıtılan koşullu ÜÇA mimarisinde (Kısım 2.2.2) de bu yapı kullanılmaya başlanmıştır.

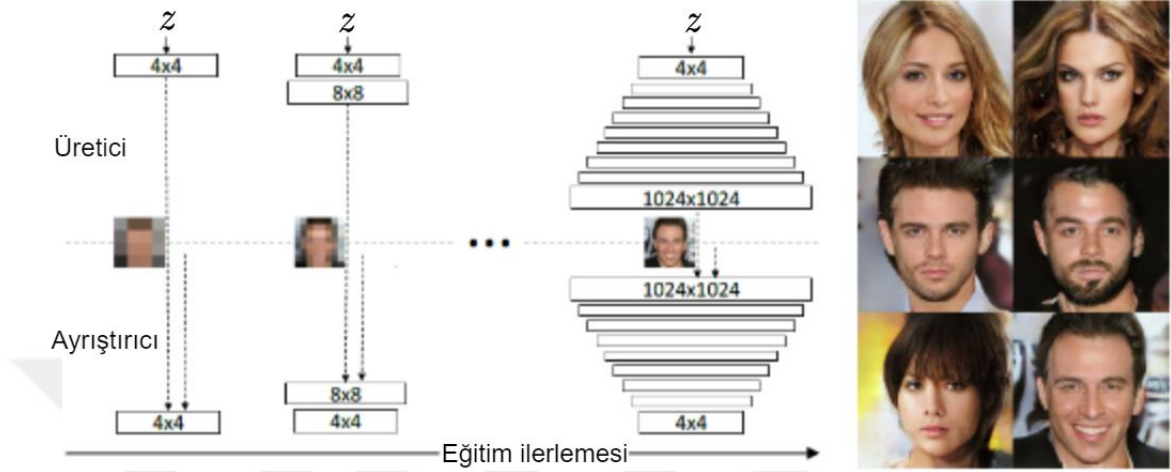
2.2.4 Kademeli büyüyen ÜÇA

Kademeli Büyüyen ÜÇA (Progressive Growing GAN (PGGAN)) mimarisi Karras vd. (2017) tarafından önerilmiştir. ÜÇA’larda genel olarak imge çözünürlüğü arttıkça Üretici ve Ayırıştırıcı ağları arasındaki rekabet dengesi bozulmaktadır. Bunun sebebi Ayırıştırıcı ağı, Üretici ağı göre daha basit bir görev tahsis edilmesinden kaynaklanmaktadır; Üretici ağı veri setindeki örneklerin altında yatan dağılımı (underlying distribution) öğrenirken, Ayırıştırıcı ağı gerçek ve Üretici ağıın öğrendiği dağılımı ayırt etmeyi öğrenmektedir. Diğer bir deyişle, imge çözünürlüğü yükseldikçe Ayırıştırıcı ağı, gerçek ve üretilen imgeleri daha iyi ayırt etmekte ve model mimarisinde kullanılan kayıp fonksiyonuna bağlı olarak bazı problemlerin (gradyan azalması, istikrar, çeşitlilik vb.) oluşmasına neden olmaktadır.

PGGAN çalışmasında iki ağı arasındaki rekabeti sürdürülebilir hale getirmek için, bu ağların kademeli eğitilmesi önerilmiştir. Şekil 2.8’de PGGAN mimarisi verilmiştir. Bu mimaride eğitim hem Üretici hem de Ayırıştırıcı ağı için düşük çözünürlükte (4x4) başlatılmakta ve ağlar kararlı duruma geldiğinde (belli sayıda iterasyon kadar eğitim yapıldığında) çözünürlük iki katına yükseltilmektedir. Bu yöntem hedeflenen çözünürlüğe ulaşılan kadar uygulanmaktadır. Eğitim aşamasında çözünürlük artırıldığında önceki katmanlardaki parametreler eğitilebilir olarak tutulmaktadır; eğitim işlemi devam ettiği sürece tüm parametreler güncellenmektedir. Bu eğitim stratejisi ile yüksek çözünürlüklü (1024x1024) veri setleri kullanılmasına rağmen iki ağı arasındaki rekabetin dengeli olduğu ve istikrarın sağlandığı gözlenmektedir.

PGGAN çalışmasında farklı veri setleri için farklı çözünürlüklerde eğitim işlemi yapılmıştır; bu kapsamda CIFAR-10 için 32x32, LSUN için 256x256 ve CelebA-HQ için 1024x1024 çözünürlüğü kullanılmıştır. CelebA (Liu vd. 2015) veri seti 202599 imgeden oluşmaktadır. CelebA-HQ olarak adlandırılan veri seti ise CelebA veri setinden alınan yüksek çözünürlüklü 30000 imgeyi içermektedir. PGGAN çalışmasında model başarısını arttırmak için bu imgelere çeşitli ön işlemler (yapay doku temizleme (artifact removal), Gauss filtreleme vb.) uygulanmıştır. İmge çözünürlüğü arttıkça hafıza problemleri meydana gelebilmektedir. Bu problemten kaçınmak için ilk başta yüksek

mini yığın (minibatch) boyutu seçilmekte ve çözünürlük arttıkça mini yığın boyutu kademeli olarak azaltılmaktadır.



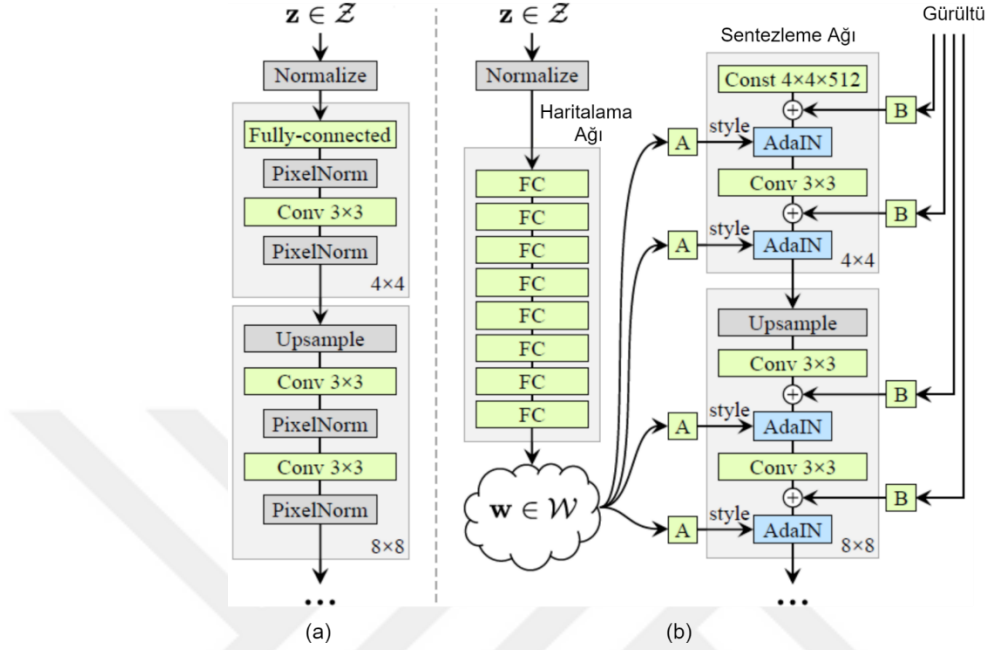
Şekil 2.8 PGGAN mimarisi (Karras vd. 2017). Şekil ilgili çalışmadan alınmıştır. Mimaride, eğitimin başlangıcında hem Üretici hem de Ayrıştırıcı ağı için 4x4 piksel çözünürlüğü kullanılmaktadır. Bu çözünürlükte eğitim kararlı duruma geldiğinde, her iki ağın çözünürlüğü iki katına çıkarılmaktadır. Bu işleme, hedef çözünürlüğe ulaşılan kadar devam edilmektedir

Kademeli eğitim stratejisi ÜÇA çalışmalarının başarısını oldukça artırmıştır. ÜÇA’larda sıklıkla meydana gelen problemlerin üstesinden geldiği gibi, imgelerdeki detayların daha iyi öğrenilmesine olanak sağlamıştır. Bu modelin dezavantajı ise eğitim işleminin çok uzun sürmesidir.

2.2.5 Stil tabanlı ÜÇA

Stil Tabanlı ÜÇA (*StyleGAN*) çalışması (Karras vd. 2018), PGGAN çalışmasının yükseltilmiş bir sürümü olarak ifade edilmektedir. Bu çalışmada imgelerin farklı görsel özelliklerini kontrol etmeyi sağlayacak yeni bir Üretici ağ mimarisi önerilmektedir. Stil transfer literatüründen referans alınarak geliştirilen bu Üretici ağ mimarisi, iki imge arasındaki stillerin (saç rengi, ten rengi, yaş vb.) birbirine aktarılmasını sağlamaktadır. Şekil 2.9 (a)’da PGGAN Üretici ağ mimarisi, (b)’de ise *StyleGAN* Üretici ağ mimarisi verilmiştir. Genel olarak ÜÇA’larda (PGGAN da dâhil) herhangi öncül bir dağılımdan rastgele örneklenen gizli vektör z , Üretici ağına girdi olarak verilmekte ve kullanılan

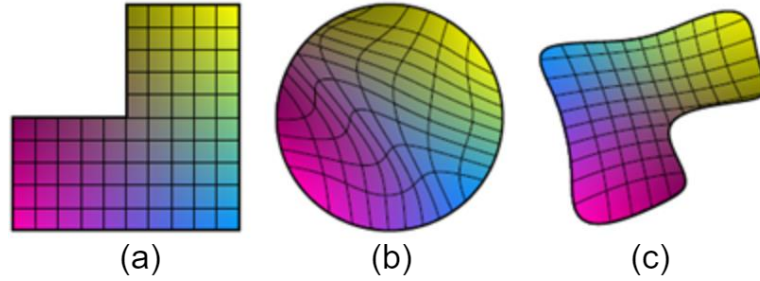
katmanlar (ters evrişimsel) vasıtasıyla bu girdi hedef çözünürlükteki imgeye dönüştürülmektedir.



Şekil 2.9 StyleGAN mimarisi (Karras vd. 2018). Şekil ilgili çalışmadan alınmıştır. (a) PGGAN Üretici ağ mimarisini, (b) ise StyleGAN Üretici ağ mimarisini temsil etmektedir

StyleGAN çalışmasında ise Üretici mimarisi, haritalama ağı (mapping network) ve sentezleme ağı (synthesis network) olmak üzere iki kısma ayrılmaktadır. Şekil 2.9 (b)'nin sol tarafında gösterildiği gibi haritalama ağı, herhangi bir öncül dağılımdan örneklenen 512 boyutlu z vektörünü girdi olarak almakta ve birkaç tam bağlantılı katmandan (ilgili çalışmada 8 katman kullanılmış) geçirip aynı boyuttaki bir vektöre dönüştürmektedir. StyleGAN çalışmasında haritalama ağının kullanılma amacı öz niteliklerin iç içe geçtiği (entangled) gizli vektör uzayı z 'yi olabildiğince herbirinin bir öz niteliği değiştirdiği alt uzaylara dönüştürmektir; haritalama ağının çıktısı w herhangi bir sabit dağılımdan (örneğin Normal veya Düzgün dağılım) örneklenmediği için öz niteliklerin daha iyi ayrıştığı (disentangled) belirtilmiştir. Şekil 2.10'da bu kapsamda ilgili çalışmadan alınan bir örnek gösterilmektedir. Şekil 2.10 (a)'da veri setindeki "erkek" ve "uzun saç" öz niteliklerinin (uzun saçlı erkekler dağılıma dâhil edilmemiştir) dağılımı verilmektedir. (b)'de bu imgelere karşılık gelen z vektörlerinin öz niteliklere haritalanması sonucu, (c)'de ise bu imgelere karşılık gelen w vektörlerinin

ilgili özniteliklere haritalanma sonuçları verilmektedir. Şekil 2.10 incelendiğinde; (b)’de bu özniteliklerin iç içe geçtiği, (c)’de ise özniteliklerin görece daha iyi ayrıştığı gözlenmektedir.



Şekil 2.10 *StyleGAN*’da kullanılan farklı uzayların incelenmesi (Karras vd. 2018). Şekil ilgili çalışmadan alınmıştır. (a) veri集中的 imgelerde “erkek” ve “uzun saç” özniteliklerinin dağılımını, (b) bu imgeleri temsil eden z vektörlerinin özniteliklere haritalanma sonucunu ve (c) ise bu imgeleri temsil eden w vektörlerinin özniteliklere haritalanma sonucunu göstermektedir

Şekil 2.9 (b)’nin sağ tarafında ise sentezleme ağı mimarisi verilmiştir. Sentezleme ağı, haritalama ağı sonucu elde edilen w vektörü, gürültü verisi ve sabit bir girdi ($4 \times 4 \times 512$) olarak sentetik imge üretmektedir. Sentezleme ağın her bir çözünürlüğünde iki tane olmak üzere stil transfer literatüründe kullanılan ve imgelere stilleri enjekte etmeyi sağlayan uyarlanabilir örnek normalleştirme (adaptive instance normalization (*AdaIn*)) katmanı kullanılmaktadır. Haritalama ağı sonucu elde edilen w vektörleri bir afın dönüşümü (affine transform) ile stillere dönüştürülmekte ve *AdaIn* katmanına girdi olarak verilmektedir.

StyleGAN çalışmasında, haritalama ağı sonucu aynı değerleri içeren 18 tane 512’lik (1024 çözünürlüğü için) vektör elde edilmektedir. İmgenin farklı çözünürlüklerine girdi olarak verilen bu 512’lik w vektörleri imgenin farklı özelliklerini (kaba, orta ve ince) değiştirmeye olanak sağlamaktadır. Çalışmada 4 ile 8 çözünürlüğüne gönderilen w vektörlerinin imgedeki kaba özellikleri, 16 ile 32 arasındaki vektörlerin orta özellikleri ve 64 ile 1024 arasındaki vektörlerin imgedeki ince özellikleri değiştirdiği belirtilmektedir. Şekil 2.11’de bu kapsamda 256×256 çözünürlüğü için sıfırdan eğitilerek elde edilen modelde bir imgedeki kaba, orta ve ince özellikleri değiştirmek için test işlemi yapılmıştır. Şekil 2.11’de 1. imge kaba, orta ve ince özelliklere karşılık

gelen w vektörlerinin alındığı imgeyi, 2. imge ise geri kalan w vektörlerinin alındığı imgeyi göstermektedir. Kaba kolundaki imge incelendiğinde, kaba özellikler 1. imgeden alındığından dolayı imgedeki poz, cinsiyet, yüz şekli vb. gibi özelliklerin 1. imgeden, geri kalan özelliklerin (saç, aydınlatma, yaş vb.) ise 2. imgeden transfer edildiği gözlenmektedir. Orta kolundaki imgede ise orta özelliklere (yüz özellikleri, saç stili, gözler açık /kapalı vb.) karşılık gelen w vektörleri 1. imgeden geri kalan özellikler ise 2. imgeden alınmıştır. İnce kolununundaki imgede ise ince özellikler (renkler, mikro yapı vb.) 1. imgeden geri kalan özellikler 2. imgeden alınmıştır. Şekil 2.11’de bir imgedeki bazı özneliklerin diğer imgeye aktarılabildiği gözlenmektedir; ancak bu mimari bir imgedeki sadece bir özneliğin aktarılmasını sağlayamamaktadır.



Şekil 2.11 StyleGAN kullanılarak imgelerde stil değiştirme. 1. imge: düşük, orta veya yüksek seviyeli özneliklerden birinin alındığı imgeyi, 2. imge: ilk imgede alınan öznelikler dışındaki özneliklerin alındığı imgeyi temsil etmektedir. Kaba; 1. imgeden kaba özneliklerin ve 2. imgeden diğer özneliklerin, Orta: 1. imgeden orta özneliklerin ve 2. imgeden diğer özneliklerin, İnce: 1. imgeden ince özneliklerin ve 2. imgeden diğer özneliklerin alınması sonucunu göstermektedir

Sentezleme aşında imgelerdeki varyasyonu ve gerçekliği arttırmak için, her çözünürlüğe rastgele örneklenen gürültü verisi girdi olarak verilmektedir. Şekil 2.12 (a)’da gürültü verisi kullanılmamasının, (b)’de ise gürültü verisi kullanılmasının etkisi

gösterilmektedir. Şekil 2.12’de gürültü verisi kullanılmaması durumunda elde edilen imgelerin, yüksek seviyeli özellikleri içermediği ve nispeten bulanık çıktılar oluşturduğu gözlenmektedir.



Şekil 2.12 Gürültü verisinin imgeler üzerindeki etkisi. (a) sentezleme ağına gürültü verisinin verilmemesi, (b) ise sentezleme ağına rastgele gürültü verisinin girdi olarak verilmesi sonucunu göstermektedir

Eğitim işleminin kademeli olarak yapıldığı mimarilerde, ağın genel olarak istikrarlı bir şekilde eğitimini sürdürdüğü ve imge çeşitliliğinin yüksek olduğu gözlenmektedir; ancak bu mimarilerde daha önceden belirtildiği gibi eğitim aşaması çok uzun sürmektedir.

2.3 Üretken Çekişmeli Ağlarda İstikrar ve Çeşitlilik

Temel ÜÇA mimarisi ile düşük çözünürlüğe sahip veri setleri kullanılarak nispeten iyi sonuçlar üretilebilmektedir. Fakat yüksek çözünürlüklü veri setleri kullanıldığında genel olarak iki ana problem meydana gelmektedir. İlk problem, ÜÇA’nın istikrarsız duruma düşmesidir. Bu problem Ayırıştırıcı ağın, Üretici ağa üstünlük sağlaması sonucu ortaya çıkmaktadır; diğer bir deyişle Ayırıştırıcı ağ, Üretici ağı için yüksek gradyan değerleri üretmekte ve Üretici ağını istikrarsız hale getirmektedir. Yakın zamanda bu problemle başa çıkmak için;

- Kayıp fonksiyonları (Zhao vd. 2016, Arjovsky vd. 2017, Lim ve Ye 2017, Jolicoeur-Martineau 2018, Park ve Kwon 2019),
- Düzenleştirme teknikleri (Gulrajani vd. 2017, Miyato vd. 2018, Wei vd. 2018, Chen vd. 2019),
- ÜÇA mimarileri (Radford vd. 2016, Berthelot vd. 2017, Karras vd. 2017, Gong vd. 2019, Karras vd. 2019) önerilmektedir.

ÜÇA çalışmalarında, sıklıkla karşılaşılan diğer problem ise mod çöküşü (mode collapse) problemidir. Mod çöküşü; Üretici ağın çok boyutlu uzayda benzer özelliklere sahip çok fazla örnek ürettiği veya üretilen örnekler arasında minimum çeşitlilik olduğu durumu göstermektedir. Bu problemin üstesinden gelmek için ise Ayrıştırıcı ağı girdi olarak her bir yığındaki imgelerin birbiriyle olan benzerlik istatistikleri verilmektedir (Salimans vd. 2016, Karras vd. 2017).

Temel ÜÇA mimarisinde kullanılan kayıp fonksiyonları (Ayrıştırıcı ağı için Denklem 2.5 ve Üretici ağı için Denklem 2.6), yüksek çözünürlüklü imgeler ve Derin Sinir Ağ (DSA)'ları kullanılarak oluşturulan mimarilerde yakınsama problemine neden olmaktadır. Eğitimin ilk aşamalarında, Üretici ağın ürettiği imgeler veri setindeki imgelerden farklı olduğu için, gradyan değerleri çok düşük çıkmakta ve Üretici ağı yeterince geri bildirim alamamaktadır. Dolayısıyla Üretici ağ öğrenememekte ve model genel olarak yakınsamamaktadır. Yapılan ilk çalışmalarda bu durum gözlemlenmiş ve Üretici için Denklem 2.9'daki amaç fonksiyonunun kullanılması önerilmiştir.

$$\min_{\mathbf{U}} D(\mathbf{U}) = \mathbb{E}_{z \sim P_z(z)} [-\log(A(\mathbf{U}(z)))] \quad (2.9)$$

Denklem 2.9'da Goodfellow vd. (2014) tarafından gradyan azalması (vanishing gradient) problemini çözmek için önerilen amaç fonksiyonu gösterilmiştir. Bu fonksiyon sayesinde gradyan akışı sağlanmaktadır; ancak eğitimin ilk kısımlarında büyük gradyan değerleri olduğu için, model istikrarsız hale gelmektedir. Belirtilen bu problemlerin üstesinden gelmek için, gerçek ve üretilen imgelerin dağılımları arasındaki *wasserstein* uzaklığını hesaplayan *Wasserstein GAN* (WGAN) (Arjovsky vd. 2017)

çalışması önerilmektedir. Kullanılan bu kayıp fonksiyonu ile yumuşak gradyan değerleri üretilebilmektedir. İlgili çalışmada Ayrıştırıcı ağ bir imgenin gerçek olup olmadığından ziyade bir skor üretmektedir ve bu nedenle Kritik ağ olarak adlandırılmaktadır. Sürekli uzayda verilen dağılımlar arasındaki *wasserstein* mesafesini kolay olarak hesaplamak için, Kritik ağın *1-Lipschitz* özelliğini sağlaması gerektiği belirtilmektedir. *WGAN*'da, *1-Lipschitz* özelliğini sağlamak için ağırlık kırpma (weight clipping) olarak adlandırılan bir yöntem kullanılmaktadır. Bu yöntem ile ağırlık değerleri bir hiperparametre vasıtasıyla sınırlandırılmaktadır. Fakat bu hiperparametre doğru ayarlanmazsa daha önceden ifade edilen gradyan azalması veya gradyan patlaması problemleri yeniden meydana gelmektedir.

WGAN-Gradient Penalty (WGAN-GP) (Gulrajani vd. 2017) çalışmasında, Kritik ağın *1-Lipschitz* özelliğini sağlaması için ağırlık kırpma yerine gradyan cezası olarak adlandırılan ilave bir maliyet değeri hesaplanmış ve Kritik ağın amaç fonksiyonuna dâhil edilmiştir. *WGAN-GP*'de ilave maliyetin hesaplanması için, her iterasyonda aradeğerleme ile yeni imgeler üretilmekte ve bu imgeler Kritik ağa verilerek ileri yayılım ve geri yayılım yapılmaktadır. Yapılan bu işlemler ilave hesaplama yükü oluşturmakta ve model eğitimi yavaşlatmaktadır. Bununla birlikte büyük öğrenme oranı kullanılması durumunda model istikrarsız hale gelebilmektedir. Bu doğrultuda, büyük öğrenme oranı kullanıldığı zaman kolaylıkla istikrarsız duruma düşmeyen ve fazla hesaplama yükü getirmeyen *Spectral Normalization GAN (SNGAN)* (Miyato vd. 2018) çalışması önerilmiştir. *SNGAN* çalışmasında, Kritik ağın *1-Lipschitz* özelliğini koruması için Spektral Normalleştirme (SN) olarak ifade edilen bir normalleştirme tekniği önerilmiştir. Kritik ağın *1-Lipschitz* özelliğini koruması için önerilen yaklaşımlar, ağın kapasitesini daraltmakta ve daha yavaş eniyilemesine sebep olmaktadır. Bundan dolayı, Kritik ağ daha fazla sayıda eniyilenerek denge kurulmaktadır.

Son zamanlarda önerilen *Self Attention GAN (SAGAN)* (Zhang vd. 2019) çalışmasında, SN işlemini hem Kritik ağa hem de Üretici ağa uygulamak ve bu ağları farklı sayıda eniyilemek yerine, *Two Time-scale Update Rule (TTUR)* (Heusel vd. 2017) yaklaşımının kullanılması önerilmiştir. *TTUR* yaklaşımında, *Nash* dengesine ulaşmak

için Üretici ağında küçük bir öğrenme oranı, Ayırıştırıcı ağında ise büyük bir öğrenme oranı (Üretici ağındaki öğrenme oranının dört katı) kullanılmaktadır; bu hem istikrarlı bir model oluşturmakta hem de eğitim sürecini hızlandırmaktadır. Ayrıca SAGAN çalışmasında, Öz-Dikkat (ÖD) mekanizması olarak tanımlanan bir teknik önerilmektedir; bu teknik derin olmayan ağlarda imgedeki global bağımlılıkların yakalanmasını sağlamaktadır.

Belirtilen bu yöntemler ÜÇA'ların eğitim işlemini istikrarlı duruma getirmektedir. Fakat istikrarın yakalanmasına rağmen imge kalitesinin istenilen seviyelere çıkamadığı gözlenmektedir. İmge çözünürlüğü yükseldikçe, Kritik ağ (veya Ayırıştırıcı ağ) veri setinden alınan ve üretilen imgeleri daha kolay ayırt edebilmektedir; bu da ağlar arasındaki dengenin bozulmasına neden olmaktadır (Odena vd. 2017). Bu problemin üstesinden gelmek için, her iki ağın kademeli olarak büyümesini sağlayan mimariler önerilmektedir (Karras vd. 2017).

ÜÇA mimarisinde kullanılan Kritik ağ (veya Ayırıştırıcı ağ) üretilen imgelerin gerçek olup olmadığı ile ilgilenmemektedir. Kritik ağ, Üretici ağın ürettiği imgelerin aynı veya birbirine benzer olup olmadığı konusunda bilgi sahibi olmamaktadır. Üretilen tüm imgeler aynı olsa bile Kritik ağın ürettiği skor değişmemektedir. Bu nedenle, genel olarak Üretici ağ, Kritik ağın yüksek skor ürettiği noktalarda yoğunlaşmakta ve bu da mod çöküşü problemini beraberinde getirmektedir. Bu problemin üstesinden gelmek için, Kritik ağın aynı yığındaki örneklerin benzerliklerine bakması sağlanmaktadır (Salimans vd. 2016). Bu amaç doğrultusunda önerilen bir yöntem MiniYığın Standart Sapmadır (MSS) (Karras vd. 2017). Bu yöntemde, üretilen imgeler ile veri setinden alınan imgelerin çeşitliliği farklı ise elde edilen MSS değerleri de farklı olmakta ve amaç fonksiyonunda ek bir maliyet oluşturmaktadır.

ÜÇA'larda meydana gelen istikrar ve çeşitlilik problemlerinin üstesinden gelmek için belirtilen bu yöntemlerin dışında birçok yöntem önerilmektedir. Ancak bu yöntemlerin de bu problemleri tam olarak çözemediği ve hala açık bir problem alanı olarak kalmaya devam ettiği gözlenmektedir.

2.4 Modelleri Karşılaştırma Metrikleri

Üretken modellerin karşılaştırılması veya eğitim aşamasında ağların gelişimini izlemek için genel olarak *Inception Score (IS)* (Salimans vd. 2016) ve *Fréchet Inception Distance (FID)* (Heusel vd. 2017) metrikleri kullanılmaktadır. İmge veri setleri için geliştirilen üretken modellerin karşılaştırılmasında takip edilen yol birbirine benzemektedir (Xu vd. 2018). *IS* metriğinde, Üretici ağ ile elde edilen imgeler bir özellik çıkarıcı ağından geçirilerek sınıf dağılımları elde edilmektedir. Özellik çıkarıcı ağ olarak, *ImageNet Large Scale Visual Recognition Challenge (ILSVRC)* yarışmasında yüksek skor üreten önceden eğitilmiş *Inception v3* (Szegedy vd. 2013) ağı kullanılmaktadır. *IS*'de *Inception v3* ağının çıktı katmanı kullanılmakta olup Denklem 2.10 kullanılarak hesaplanmaktadır.

$$IS = \mathbb{E}_{x \sim p_g} D_{KL}(p(y|x) || p(y)) \quad (2.10)$$

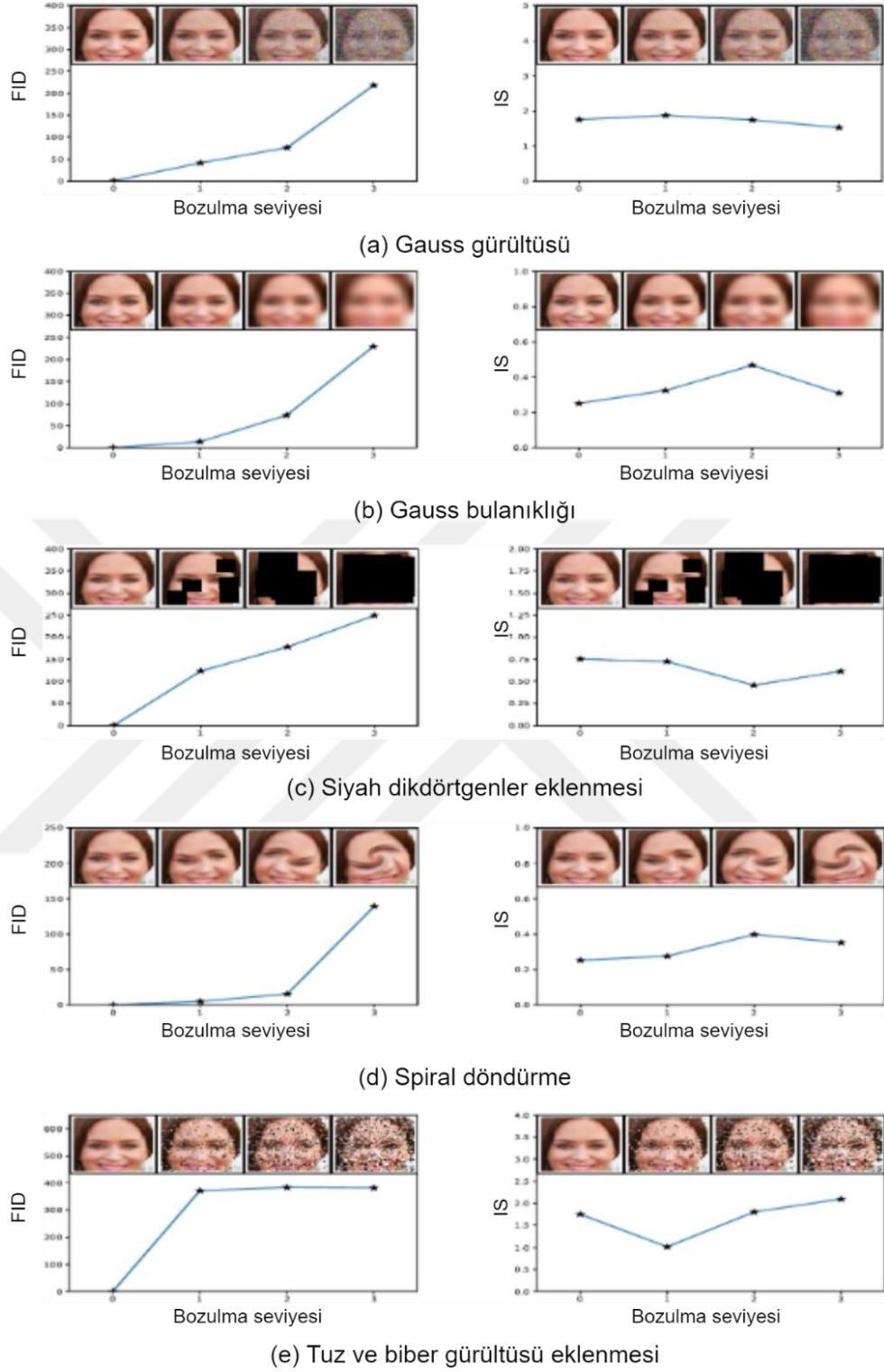
Denklem 2.10'da $x \sim p_g$ x 'in p_g 'den örneklendiğini (Üretici ağ kullanılarak rastgele üretilen örnekler), $D_{KL}(p||q)$ p ve q dağılımları arasındaki *KL* ıraksamasını, $p(y|x)$ koşullu sınıf dağılımını, $p(y)$ ise marjinal sınıf dağılımını göstermektedir. *IS* metriği sadece üretilen imgeler kullanarak hesaplanmaktadır; diğer bir deyişle gerçek veri setindeki imgeler kullanılmamaktadır. *FID* metriğinde ise hem gerçek hemde üretilen imgeler bir özellik çıkarıcı ağından geçirilmekte ve belirlenen bir katmanda elde edilen özellikler kullanılarak iki örnek grubu arasındaki benzerlik düzeyi belirlenmektedir. Bu kapsamda, *Inception v3* ağının *pool3* katmanındaki 2048 boyutlu özellik vektörü kullanılmaktadır. *FID* metriği hesaplanırken hem gerçek ($\mathcal{N}(\mu_r, \Sigma_r)$) hemde üretilen imgelerin ($\mathcal{N}(\mu_g, \Sigma_g)$) oluşturduğu çok değişkenli *Gauss* dağılımları (multivariate Gaussians) arasındaki istatistikler kullanılmaktadır. *FID* skoru Denklem 2.11 kullanılarak hesaplanmaktadır.

$$FID = \|\mu_r - \mu_g\|_2^2 + Tr(\Sigma_r + \Sigma_g - 2(\sqrt{\Sigma_r \Sigma_g})) \quad (2.11)$$

Denklem 2.11'de μ_r ve Σ_r referans veri setindeki örneklerin bir *Gauss* dağılımına uydurulması sonucu elde edilen veri dağılımının ortalama vektörü ve kovaryans

matrisini μ_g ve Σ_g , bir üretici model kullanılarak üretilen imgelerin bir *Gauss* dağılımına uydurulması sonucu elde edilen ortalama vektörü ve kovaryans matrisini, Tr ise doğrusal cebirde iz işlemini (kare matrisin ana köşegeni boyunca elemanların toplamı) temsil etmektedir. *IS* skorunun yüksek, *FID* skorunun ise düşük çıkması modelin daha iyi olduğunu göstermektedir.

TTUR çalışmasında, yüz imgeleri üzerinde farklı bozulmalar oluşturularak bunları tespit edebilmeleri bakımından *IS* ve *FID* metrikleri karşılaştırılmıştır. Şekil 2.13'te düşük seviyeden yüksek seviyeye doğru; (a)'da imgelere *Gauss* gürültüsü ekleme, (b)'de *Gauss* bulanıklığı uygulama, (c)'de imgelerdeki bazı bölgelerin siyah maske ile kapatılması, (d)'de spiral döndürme ve (e)'de tuz ve biber gürültüsü eklenmesi sonucu metriklerin etkisi karşılaştırılmıştır (Heusel vd. 2017). Şekil 2.13'te *TTUR* çalışmasında, yapılan hemen hemen tüm testlerde, *FID*'nin yapılan bozulmalarla ters orantılı olarak tepki verebildiği; ancak *IS* metriğinin bozulmaları iyi yakalayamadığı gözlenmektedir. Bu nedenle tez çalışmasında, model gelişimini izlemek ve modellerin karşılaştırmak için *FID* metriği kullanılmaktadır.



Şekil 2.13 *FID* ve *IS* metriklerinin karşılaştırılması (Heusel vd. 2017). Şekil ilgili çalışmadan alınmıştır. *FID*: *Fréchet inception distance*, *IS*: *Inception score*. (a) imgelere *Gauss* gürültüsü eklenmesi, (b) *Gauss* bulanıklığı uygulanması, (c) siyah dikdörtgenler eklenmesi, (d) spiral döndürme yapılması, (e) ise tuz biber gürültüsü eklenmesi sonucu metriklerin davranışlarını göstermektedir

3. YARI DENETİMLİ İMGE ÖZNETELİK DÜZENLEME

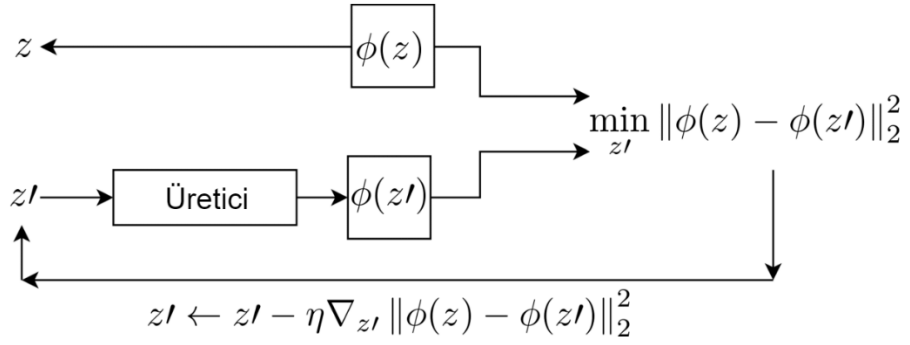
İmge öznitelik düzenleme, son zamanlarda üretken ağlar kullanılarak birçok araştırmacı tarafından incelenen zorlu bir alandır. Buradaki zorluk, imgede istenen öznitelikler dışındaki diğer ayrıntıları korumaktır. Literatürdeki hemen hemen tüm çalışmalar, bu amaçla denetimli bir ortamda etiketli veri setlerini kullanmakta veya öznitelik başına yeni bir model eğitimi gerektirmektedir. Tez kapsamında hedeflenen yöntem, bir imgeyi temsil eden doğru gizli vektör temsilini ve bir özniteliğe karşılık gelen doğru vektör yönünü bularak bu işlemi gerçekleştirmektedir. Bu amaç kapsamında, denetimsiz bir ortamda bir Kodlayıcı ağ aracılığıyla, Üretici ağın tersi işlevini doğru bir şekilde öğrenmeye olanak sağlayan CRG adlı bir mimari önerilmiştir. Daha sonra, önerilen CRG mimarisi kullanılarak gizli uzayda istenen öznitelik temsilleri elde edilmiş ve imgede öznitelik düzenleme amacı ile kullanılmıştır.

Bu bölüm aşağıdaki şekilde düzenlenmiştir. İlk kısımda, Üretici ağın ters işlevini öğrenmeyi sağlayan modeller ile ilgili literatür bilgisi verilmiştir (Kısım 3.1). İkinci kısımda, *CelebA* veri seti için uygun mimari ve hiperparametrelerin belirlenmesi için yapılan deneysel çalışmalar belirtilmiştir. Bu kısımda, ÜÇA mimarisindeki problemleri daha iyi anlamak için düşük çözünürlüğe sahip ve gri değerlerden oluşan *MNIST* veri seti ile başlanmıştır (Kısım 3.2). Üçüncü kısımda, ilgili veri seti için iyi bir Üretici ağ elde edildikten sonra döngüsel hata enküçültmesi işlevi aracılığıyla imgelerin yeniden yapılandırılmasını sağlamak için önerilen yöntem tanıtılmıştır (Kısım 3.3). Dördüncü kısımda, önerilen yöntem aynı amaç doğrultusunda geliştirilen modellerle nitel ve nicel olarak karşılaştırılmış ve önerilen yöntemin daha başarılı olduğu gösterilmiştir (Kısım 3.4). Beşinci kısımda, önerilen yöntem ile bir çift referans imge kullanılarak gizli öznitelik temsillerinin nasıl elde edildiği gösterilmiştir (Kısım 3.5). Altıncı kısımda, hesaplanan öznitelik yönlerinin hedef özniteliklerle alakalı olduğunu göstermek için deneysel analizler yapılmıştır (Kısım 3.6). Yedinci kısımda, aynı öznitelik için farklı imge kullanımının etkisi incelenmiştir (Kısım 3.7). Sekizinci kısımda, bir öznitelik yönünde ne kadar ilerlenebildiği ve bu yöndeki ilerlemenin imgelerde nasıl bir değişime yol açtığı ile ilgili deneysel çalışmalar ve analizler sunulmuştur (Kısım 3.8). Son

kısımda ise öznitelik düzenleme sonuçları verilmiş ve değerlendirmeler yapılmıştır (Kısım 3.9).

3.1 Literatür

Üretken çekişmeli ağlarda, imgelerde istenen öznitelik/öznitelikleri manipüle etme çalışmaları yarı denetimli ve denetimsiz olmak üzere iki ana gruba ayrılabilir. İlk grupta eğitim, koşullu bir ÜÇA mimarisi ve her bir özniteliği temsil eden bir kod kullanılarak gerçekleştirilmektedir (Mirza ve Osindero 2014, Perarnau vd. 2016, Antipov vd. 2017, Jaiswal vd. 2018, Choi vd. 2018, He vd. 2019). Bu yöntemlerde eğitim aşamasından sonra herhangi bir imgenin seçilen öznitelikleri değiştirilebilmektedir. Ancak etiketli veri seti gerekliliği bu yöntemleri kullanışsız kılmaktadır. Ayrıca modelin öznitelik değiştirme yeteneği, eğitim veri setinde sağlanan etiketlerle sınırlı kalmaktadır. İkinci grupta, yani denetimsiz yöntemlerde temel amaç, üretken modellerdeki gizli yapıyı öğrenen bir model oluşturmak ve öznitelik etiketleri kullanmadan üretim sürecini tersine çevirmektir. Önerilen yöntem bu gruba aittir. Bu alt gruptaki çalışmalar üç kategori altında incelenmektedir: İlki Gradyan Tabanlı Teknik (GTT)'lerdir (Lipton ve Tripathi 2017, Creswell ve Bharath 2018). Bu yaklaşımlarda, bir imgeyi temsil eden gizli vektör eniyileme hedefi olarak alınmakta ve $\phi(z)$ imgesine karşılık gelen gizli vektör z 'nin bulunması amaçlanmaktadır. Bu yöntemler, önceden eğitilmiş bir Üretici ağ kullanmakta ve ilave bir Kodlayıcı ağı gerektirmemektedir. İlk olarak, bir öncül dağılımdan rastgele örneklenen z' vektörü Üretici ağı girdi olarak verilmektedir. Üretilen imge ile hedef imge arasındaki farktan elde edilen gradyana göre z' vektörü güncellenmektedir. Çok sayıda yinelemeden sonra, z' vektörünün, $\phi(z)$ imgesini temsil eden hedef vektör z 'ye yakınsaması beklenmektedir. GTT'lerde, gizli vektör temsilinin kodlaması, çok sayıda gradyan iniş (gradient descent) adımından sonra elde edilmektedir; bu da ilgili yaklaşımları kullanışsız kılmaktadır. GTT'nin yapısı ve eniyileme işlevi Şekil 3.1'de gösterilmiştir.



Şekil 3.1 GTT yapısı (Lipton ve Tripathi 2017)

İkinci kategori, Üretici ve Ayırıştırıcı ağlara ilave olarak bir Kodlayıcı ağ içeren mimarilerdir (Larsen vd. 2016, Donahue vd. 2016, Dumoulin vd. 2016, Luo vd. 2017). Bu çalışmalar, Kodlayıcı ağın mimaride nasıl kullanıldığına ve eğitildiğine bağlı olarak farklılık göstermektedir. VAE-GAN'da (Larsen vd. 2016), Üretici ağ VAE'nin Kodçözücüsü olarak kullanılmakta ve Kodlayıcı-Kodçözücü-Ayırıştırıcı ağları tek bir ağ olarak uçtan uca birleştirilmektedir. Bu yapı, VAE'lere kıyasla daha gerçekçi ve daha keskin imgelerin yeniden yapılandırılmasını sağlamaktadır. VAE-GAN'a benzer şekilde, α -GAN (Rosca vd. 2017) çalışmasında da VAE ve çekişmeli öğrenme yapıları birleştirilmektedir. Bu çalışma aynı zamanda gizli uzayda ilave bir Ayırıştırıcı ağ ve imge uzayında piksel tabanlı yeniden yapılandırma kaybı kullanmaktadır. Bu iki modelden farklı olarak, BiGAN (Donahue vd. 2016) ve ALI (Dumoulin vd. 2016) çalışmalarında ne Üretici ne de Kodlayıcı ağ birbirlerinin çıktılarını görmemektedir; her iki çalışmada da Ayırıştırıcı ağına, imge ve gizli vektör çifti girdi olarak verilmektedir. Bu yöntem ile Ayırıştırıcı, imgenin veri setinden alınıp alınmadığını belirlemenin yanında, gizli vektörün rastgele mi yoksa bir Kodlayıcı ağ ile mi kodlandığını da öğrenmektedir. İkinci kategoriye ait çalışmalarda genellikle üç ağ eş zamanlı olarak eğitilmektedir; bu, önceden eğitilmiş bir Üretici ağ kullanarak Kodlayıcı ağı eğitmeyi zorlaştırmaktadır. Bu kategorideki çalışmaların bir diğer dezavantajı, yüksek çözünürlüklü sentetik imgeler oluşturmak için yetersiz olmalarıdır; bu çalışmaların çoğu 64x64 çözünürlüğün ötesine geçememektedir (Heljakka vd. 2018). Ayrıca bu çalışmaların çoğu, bir imgenin gizli vektör temsilini çıkarmakta başarısız olmaktadır; bu da diğer öznitelikleri koruyup, yalnızca istenen bir öznitelik alt kümesini değiştirmeyi sağlayamamaktadır. Bu kategorideki bir başka çalışma da AEGAN mimarisidir (Luo vd.

2017). Bu mimaride, Kodlayıcı ağı diğer ağlarla eşzamanlı olarak eğitilmemektedir. Bu çalışmada, önceden eğitilmiş bir Üretici model kullanılarak Kodlayıcı-Üretici mimarisi oluşturulmakta ve Kodlayıcı ağı sıfırdan eğitilmektedir. Kodlayıcı ağı, sadece üretilen imgeler kullanılarak eğitilmekte ve eğitim aşamasında gerçek imgeler modele girdi olarak verilmemektedir. Bu modelde, üretilen imgelerin yeniden yapılandırma sonuçları gerçek imgelerinkinden daha iyidir; ancak her ikisinin de çok başarılı olmadığı gözlenmektedir.

Üçüncü kategori, yapısında bir Ayrıştırıcı ağı içermeyen ve doğrudan Kodlayıcı ile Üretici ağları arasında çekişmeli bir oyun kurulan Çekişmeli Üretici-Kodlayıcı (Adversarial Generator-Encoder) (AGE) mimarisidir (Ulyanov vd. 2018, Heljakka vd. 2018). Bu kategoride eğitim işlemi, Kodlayıcı ve Üretici ağlarının karşılıklı (reciprocal) olmasını sağlamak için çekişmeli kayıp ve yeniden yapılandırma kaybının bir kombinasyonu kullanılarak gerçekleştirilmektedir. Bu mimarinin üç ana avantajı vardır:

- Kodlayıcı ve Üretici ağları arasında çekişmeli kayıp (adversarial loss) hesaplanırken bir Ayrıştırıcı ağına ihtiyaç duyulmamakta ve öğrenilen parametrelerin sayısı azalmaktadır.
- ÜÇA'larda dağılımlar genellikle yüksek boyutlu imge uzayında, AGE'de karşılaştırma daha basit bir gizli uzayda hesaplanmaktadır.
- Kodlayıcı ağına girdi olarak hem gerçek hem de sahte imgeler verilmektedir; böylece gerçek imgeler daha iyi kodlanmaktadır.

Bu mimarinin çıktıları incelendiğinde, 64x64 piksel çözünürlüğünde diğer uçtan uca mimarilere göre daha iyi sonuçlar verdiği gözlenmektedir. Ancak imgelerin yeniden yapılandırılması için 128x128 piksel çözünürlüğünün ötesine geçilemediği ve bu çözünürlükteki sonuçların istenilen düzeyde olmadığı gözlenmektedir. Bu kategorideki çalışmalar incelendiğinde, yapısında Ayrıştırıcı ağı içeren modellere göre imge kalitesi ve çeşitliliği açısından zayıf oldukları gözlenmektedir. Ayrıca AGE mimarisinde yeniden yapılandırma kaybı kullanılmaması durumunda, Kodlayıcı ve Üretici eşlemesi karşılıklı olmamaktadır. Ağları karşılıklı yapmak için amaç işlevinde (objective function) bir hiperparametre tarafından kontrol edilen yeniden yapılandırma kaybı

kullanılmaktadır; ancak bu hiperparametre değerin belirlenmesi ekstra yük getirmektedir. Bu kategoride bahsedilen problemlerin üstesinden gelmek için, birbirleriyle döngüsel bağlantıları olan iki ayrı kısımlı bir yöntem önerilmiştir. Önerilen yöntemde, eğitim işlemi iki aşamada gerçekleştirilmektedir. İlk aşamada, Üretici ve Ayrıştırıcı ağları arasında çekişmeli kayıp kullanılarak bir Üretici ağ eğitilmektedir (Karras vd. 2017). İkinci aşamada, Kodlayıcı ağ, Kodlayıcı ve Üretici ağları arasında döngüsel bir bağlantı kullanılarak sıfırdan eğitilmektedir. Bu mimari, Üretici ağın rastgele oluşturulmuş veya kodlanmış bir gizli vektörü girdi olarak almasını ve Kodlayıcı ağın eğitim esnasında gerçek veya sahte imgeleri girdi olarak almasını sağlamaktadır. Bu yaklaşım, eğitim esnasında Kodlayıcı ağın yakınsama problemine yardımcı olmaktadır. Ayrıca farklı alanlarda mevcut önceden eğitilmiş modellerin karşılıklı eşlemelerini mümkün kılmaktadır.

Kodlayıcı ağın başarısı, Üretici ağın gizli vektörleri imgelere eşleme başarısıyla doğrudan ilişkilidir. Üretici eğitiminde, özellikle yüksek çözünürlüklü imgeler kullanıldığında çeşitli problemler (gradyan azalması, istikrar, çeşitlilik vb.) meydana gelmektedir. Yaygın veri setleri için yüksek kaliteli imgeler üreten çeşitli ÜÇA araştırmaları olmasına rağmen hangi algoritmanın diğerlerine göre daha iyi olduğu açık değildir. Yakın zamanda yapılan bir çalışmada, son teknoloji modeller, değerlendirme ölçütleri (*FID*, *IS* vb.) kullanılarak nesnel olarak karşılaştırılmıştır (Lucic vd. 2018). Bu çalışmada, modellerin performansının büyük ölçüde veri setlerine ve hiperparametre eniyilemesine bağlı olduğu; tüm veri setlerinde sürekli olarak yüksek kaliteli imgeler üreten ideal bir modelin olmadığı belirtilmiştir. Bu nedenle, istenen veri seti için uygun mimarinin ve uygun hiperparametrelerin belirlenmesi gerekmektedir.

3.2 Üretici Ağın Elde Edilmesi

Bu kısımda, istikrarlı ve çeşitli imge üretimine olanak sağlayan bir ÜÇA modeli oluşturmak için yapılan çalışmalar ele alınmıştır. Yapılan deneysel çalışmalarda, ilk olarak düşük çözünürlüğe sahip (28x28) ve gri piksel değerlerinden oluşan *MNIST* veri seti kullanılmıştır. Model olarak ise *DCGAN* (Radford vd. 2016) modeli referans alınmıştır. Mimari Üretici ağ, bir tam bağlantılı katman (*dense*), 4 ters evrişimsel ve

bir tane evrişimsel katmandan oluşmaktadır. Üretici ağı, son katmanında *tanh* aktivasyon fonksiyonu, diğer katmanlarında ise *LReLU* kullanılmış ve eğim 0.2 olarak alınmıştır. Üretici ağı, düzgün dağılımdan $[-1,1]$ aralığında rastgele örneklenmiş 100 boyutlu z gizli vektörü girdi olarak verilmiştir. Çizelge 3.1’de Üretici ağ katman ve detayları verilmiştir.

Çizelge 3.1 Üretici ağ katman ve detayları

Girdi boyutu: 100				
Katman	Filtre boyutu	Atlama	Filtre sayısı	Aktivasyon fonksiyonu
<i>Dense</i>	-	-	128*7*7	<i>LReLU</i>
<i>Reshape</i>	-	-	(7x7)*128	-
<i>Deconv2D</i>	5x5	2	128	<i>LReLU</i>
<i>Deconv2D</i>	5x5	2	64	<i>LReLU</i>
<i>Conv2D</i>	5x5	1	1	<i>Tanh</i>

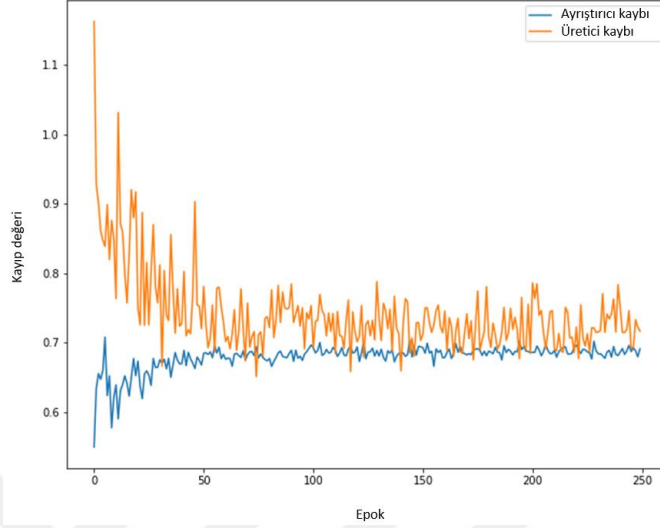
Ayrıştırıcı ağ ise 2 evrişimsel ve bir tam bağlantılı katmandan oluşmaktadır. Ağı evrişimsel katmanlarında *LReLU* (eğim 0.2), son katmanda ise *sigmoid* aktivasyon fonksiyonu kullanılmıştır. Çizelge 3.2’de Ayrıştırıcı ağ katman ve detayları verilmiştir.

Çizelge 3.2 Ayrıştırıcı ağ katman ve detayları

Girdi boyutu:28x28x1					
Katman	Filtre boyutu	Atlama	Filtre sayısı	Aktivasyon fonksiyonu	Dropout
<i>Conv2D</i>	5x5	2	64	<i>LReLU</i>	Evet
<i>Conv2D</i>	5x5	2	128	<i>LReLU</i>	Evet
<i>Flatten</i>	-	-	7x7x128	-	Hayır
<i>Dense</i>			1	<i>Sigmoid</i>	Hayır

Oluşturulan modelde, Ayrıştırıcı ağ Denklem 2.5’te verilen amaç fonksiyonunu enbüyütmeye etmeye çalışırken Üretici ağ ise Denklem 2.6’da verilen amaç fonksiyonunu enküçültmeye çalışmaktadır. Mimaride kayıp fonksiyonu olarak *Binary Cross Entropy (BCE)* kullanılmış olup Ayrıştırıcı ağın çıkışında hesaplanmaktadır. İki ağın parametrelerinin eniyilemesi için ise *Adam* (Kingma ve Ba 2014) eniyileme

yöntemi kullanılmıştır. Şekil 3.2’de 250 epok için Üretici ve Ayırıştırıcı ağların kayıp grafiği gösterilmiştir.



Şekil 3.2 Üretici ve Ayırıştırıcı ağların kayıp grafiği

Şekil 3.2 incelendiğinde, Ayırıştırıcı ağı eğitimin ilk aşamalarında gerçek ve sahte imgeleri kolaylıkla ayırt edebilmektedir. Ancak eğitim ilerledikçe Üretici tarafından üretilen imgeler, veri dağılımına yaklaşmakta ve Ayırıştırıcı ağı imgeleri ayırt etmekte zorlanmaktadır. Nihai olarak Ayırıştırıcı ağı, hem gerçek hem de sahte imgeler için yaklaşık 0.69 kayıp değeri üretmektedir. Ayırıştırıcı ağın son katmanında *BCE* kayıp fonksiyonu kullanıldığından;

$$-\log q_i \cong 0.69 \quad (3.1)$$

Olmaktadır. Denklem 3.1’den de $q_i \cong 0.5$ bulunmaktadır. Ayırıştırıcı ağı, *sigmoid* bir birim ile bittiğinden 0 ile 1 arasında çıktı üretmektedir. Diğer bir deyişle, gerçek veri setinden alınan imgeler için 1, Üretici ağı tarafından üretilen imgeler için 0 üretmesi amaçlanmaktadır. Ancak eğitim aşamasından sonra, Ayırıştırıcı ağı kendisine verilen herhangi bir imgeyi 0.5 olasılığında gerçek veya sahte olarak ayırt etmektedir. Bu da Ayırıştırıcı ağın, üretilen imgeleri gerçek veya sahte olarak ayırt edemediğini, diğer bir deyişle üretilen imgelerin gerçek veri dağılımına yakınsadığını göstermektedir.

Eğitim işlemi tamamlandıktan sonra Ayırıştırıcı ağ atılmakta, Üretici ağ ise veri setine benzer imgeler üretmek amacıyla kullanılmaktadır. Şekil 3.3'te Üretici ağ kullanılarak *MNIST* veri setindeki her bir sınıf için rastgele üretilen bazı imgeler verilmiştir.



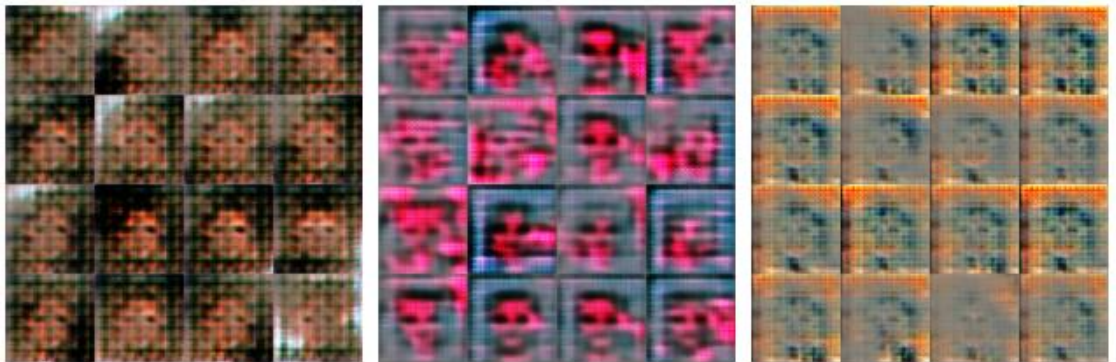
Şekil 3.3 Rastgele üretilen imgeler

MNIST veri seti, düşük çözünürlüklü ve ikili değerlerden oluştuğu için kullanılan ağlar derin olmamasına rağmen iyi sonuçlar elde edilmektedir. Fakat benzer mimari 64x64'lük *CelebA* veri seti için kullanıldığı zaman Üretici ağ için kullanılan amaç fonksiyonuna bağlı olarak iki farklı durum ortaya çıkmaktadır. Bunlardan ilki, Üretici ağında Denklem 2.6'daki amaç fonksiyonu kullanıldığı zaman gradyan azalması problemi oluşmakta ve üretilen imgeler belli bir seviyenin üzerine çıkamamaktadır. Bu durum Şekil 3.4'de gösterilmiştir. Ayırıştırıcı ağ, eğitimin ilk aşamalarında yeterli düzeyde eniyilenmediğinden, Üretici ağ için doğru yönde gradyanlar üretebilmektedir; bu da veri setine benzer imgelerin üretilmesine olanak sağlamaktadır. Fakat bir noktadan sonra Ayırıştırıcı ağ, üretilen ve veri setinden alınan imgeleri tam olarak ayırt etmekte ve Üretici ağ için yeterli gradyan akışı olmamaktadır. Bu durumda gradyan akışı olmadığından, eğitim işlemine devam edilse de imge kalitesinde değişiklik oluşmamaktadır.



Şekil 3.4 Gradyan azalmasının imge kalitesi üzerindeki etkisi

İkinci durum ise gradyan akışını sağlamak için Üretici ağında Denklem 2.9’da belirtilen amaç fonksiyonunun kullanılmasıdır. Bu durumda, eğitimin ilk aşamalarında üretilen ve veri setinden alınan imgeler birbirine benzemediği için, büyük gradyan değerleri üretilmektedir; bu Üretici ağındaki parametrelerin çok fazla değişmesine neden olmaktadır. Bu yöntemde, model istikrarsız olmakta ve eğitim esnasında imge kalitesi düzensiz olarak artmakta veya azalmaktadır. Ancak Ayrıştırıcı ağı henüz eğitimini tamamlamamıştır ve bu örneklerin gerçek olmadığını anladığında büyük gradyan değerleri üretmeye devam etmektedir. Böylece imge kalitesi gittikçe düşmeye başlamaktadır. Şekil 3.5’te bu durum gösterilmiştir. Şekil 3.5 incelendiğinde, soldaki imgelerde, Ayrıştırıcı ağı yüksek skor ürettiği noktalarda örneklerin çoğalmaya başladığı gözlenmektedir. Fakat Ayrıştırıcı ağı iterasyon sayısı arttıkça üretilen örneklerin Üretici ağından geldiğini öğrenmekte ve imge kalitesi gittikçe düşmektedir.



Şekil 3.5 Gradyan patlamasının imge kalitesi üzerindeki etkisi

Bu modellerden ikincisinde gradyan akışı devam ettiği için uygun hiperparametreler kullanılarak model iyileştirilebilmektedir. Deneysel çalışmalarda, modelin iyileşme gösterdiği; ancak imge kalitesinin görsel olarak istenilen seviyelere ulaşamadığı gözlenmiştir. İmge kalitesini arttırmak için son zamanlarda *ImageNet* veri setinde yüksek skor üreten *SAGAN* çalışması referans alınmıştır ve *CelebA* veri seti için (128x128 çözünürlüğü için) performansı incelenmiştir. Bu modelde; *hinge* amaç fonksiyonu kullanılmış, Üretici ve Ayrıştırıcı ağırlarına spektral normalleştirme uygulanmış ve *TTUR* yaklaşımı ile Üretici ve Kritik ağırları için farklı öğrenme oranları (1/4) ayarlanmıştır. Ayrıca yüksek seviyeli özelliklerin olduğu (32x32 ile 64x64) çözünürlükler arasında ÖD kullanılmış, Üretici ve Kritik ağırları $k = 1:1$ oranında eniyilenmiştir. Bu model, Çizelge 3.3'te M1 ile ifade edilmiştir.

Çizelge 3.3 Modellerin karşılaştırılması. KF: Kayıp fonksiyonu, SN: Spektral normalleştirme, ÖD: Öz dikkat, MSS: Miniyiğin standart sapma, *TTUR*: Two Time-scale Update Rule, *FID*: Fréchet inception distance

Model	KF	SN	ÖD	MSS	<i>TTUR</i>	<i>FID</i>
M1	<i>Hinge</i>	E	E	H	Ü: 1e-4 A: 4e-4 k = 1:1	137
M2	<i>WGAN-GP</i>	E	E	H	Ü: 1e-4 A: 4e-4 k = 1:1	85.83
M3	<i>WGAN-GP</i>	E	E	H	Ü: 1e-4 A: 2e-4 k = 1:2	33.75
M4	<i>WGAN-GP</i>	E	E	E	Ü: 1e-4 A: 2e-4 k = 1:2	24
M5	<i>WGAN-GP</i>	E	H	E	Ü: 1e-4 A: 2e-4 k = 1:2	14.16

Şekil 3.6'da *SAGAN* mimarisi kullanılarak elde edilen imgeler gösterilmiştir. İmgeler incelendiğinde, temel ÜÇA mimarisine göre daha iyi sonuçların elde edildiği gözlenmektedir. Ancak elde edilen imgeler incelendiğinde, *FID* skorun yüksek ve imge kalitesinin istenen seviyede olmadığı gözlenmiştir. Aynı zamanda, farklı gizli vektörler modele girdi olarak verilmesine rağmen benzer imgelerin üretildiği gözlenmektedir. Bu da mod çöküşü probleminin olduğunu göstermektedir.



Şekil 3.6 SAGAN mimarisi kullanılarak rastgele üretilen imgeler

Şekil 3.6’da verilen imgeler ve Çizelge 2.3’te verilen *FID* değeri modelin 100 epok eğitilmesi sonucu elde edilen sonuçlardır. *FID* değeri hesaplanırken kaç tane imgenin kullanıldığı önem arz etmektedir. Bu sayıyı belirlemek için gerçek imgeler kullanılarak bir test işlemi yapılmıştır. Sonuçlar Çizelge 3.4’te gösterilmiştir. Örnekler veri setinden alınmasına rağmen örnek sayısı az seçildiğinde nispeten büyük *FID* skorların oluştuğu gözlenmektedir. Çizelge 3.4’ten yola çıkarak modellerin değerlendirilmesi aşamasında yapılan test işlemlerinde 50000 imge kullanılmıştır. Daha çok sayıda imge kullanılarak bu değer sıfıra yakınlştırılabilir; ancak hafıza probleminden dolayı örnek sayısı 50000 olarak alınmıştır.

Çizelge 3.4 Örnek sayısına bağlı olarak *FID* skorları

Örnek sayısı	<i>FID</i>
2048	5.40
8192	1.21
16384	0.73
32768	0.47
50000	0.1

SAGAN mimarisi kullanılarak elde edilen skor, üretilen imgelerin hem kalitesinin düşük olduğunu hem de imge çeşitliliğinin az olduğunu göstermektedir. İmge kalitesini arttırmak için, *WGAN-GP* çalışmasında önerilen düzenleme tekniği (gradyan cezası) kullanılmıştır. Bu güncelleme ile *FID* değeri 137'den 85.83'e inmiştir. *WGAN-GP* ve *SNGAN* yöntemleri benzer bir amacı, yani Ayrıştırıcı ağı *1-Lipshitz* özelliğini sağlaması için birbirine alternatif olarak geliştirilmiştir. Ancak Çizelge 3.3'te M2 modeli incelendiğinde, kayıp fonksiyonu dışındaki tüm parametreler aynı tutulduğunda *FID* skorunun belirgin bir şekilde düştüğü gözlenmektedir.

Şekil 3.7'de kayıp fonksiyonunun *WGAN-GP* ile değiştirilmesi sonucu rastgele üretilen imgeler gösterilmiştir. Dikkat edilirse, imge kalitesinde yükselme olduğu ve imge çeşitliliğinin arttığı gözlenmektedir.



Şekil 3.7 Modelde kayıp fonksiyonu olarak *WGAN-GP* kullanılmasının etkisi

Kritik ağına herhangi bir düzenleme tekniği uygulandığında, ağı kapasitesi azalmakta ve daha yavaş eniylenmektedir. Bundan dolayı M1 modelinde, Kritik ağ için Üretici ağına göre daha büyük bir öğrenme oranı kullanılarak iki ağ arasındaki denge kurulmuştur. Fakat *WGAN-GP*'de yüksek öğrenme oranları kullanıldığında model

istikrarsız oluşturmaktadır. Dolayısıyla M3 modelinde, Kritik ağıın öğrenme oranı yarıya düşürölüp Kritik ağıın 2 kat fazla eniyilenmesinin etkisi incelenmiştir. Bu güncelleme ile birlikte imge kalitesi artmış ve *FID* skoru önemli oranda düşmüştür.

Şekil 3.8’de M3 modeli kullanılarak elde edilen sonuçlar gösterilmiştir. Şekil 3.8 incelendiğinde bir önceki modele göre imge kalitesinin arttığı gözlenmektedir. Bu durum elde edilen *FID* skorundan da anlaşılmaktadır. Bir önceki modele göre *FID* değeri 85.83’ten 33.75’e düşmüştür.



Şekil 3.8 Büyük öğrenme oranı yerine farklı sayıda eniyilemenin etkisi

FID metriğinin imge kalitesi ile birlikte imge çeşitliliğini de göz önünde bulundurduğu belirtilmektedir. Bu doğrultuda, M4 modelinde MSS eklenmiş ve *FID* değerinin 24’e düştüğü gözlenmiştir. Şekil 3.9’da bu model ile elde edilen imgeler verilmiştir. Burada dikkat edilirse imge kalitesinde çok belirgin bir artış gözlemlenmekte ve ilk defa gözlüklü bir yüz imgesinin (sol altta) üretildiği gözlenmektedir.



Şekil 3.9 Miniyığın standart sapmanın imge çeşitliliği üzerindeki etkisi

SAGAN'da, ÖD mekanizmasının doğru katmanlar arasına yerleştirilmemesi halinde imge kalitesinin azaldığı belirtilmiştir. Bu nedenle M5 modelinde, ÖD'nin doğru yere konumlandırılıp konumlandırılmadığını araştırmak için sistemden çıkarılmış ve *FID* skorunun düştüğü görülmüştür. Bu iyileşme, ÖD mekanizmasının doğru katmanlar arasına yerleştirilmediği ile ilişkilendirilmiştir. Fakat ÖD'nin farklı çözünürlükler arasına konumlandırılmasına rağmen *FID* skorunda bir azalma gözlenmemiştir.

Şekil 3.10'da ÖD'nin modelden çıkarılması sonucu elde edilen imgeler verilmiştir. Burada çok belirgin olmasa da imge kalitesinde bir iyileşme olduğu gözlenmektedir.



Şekil 3.10 Öz-dikkat mekanizmasının imge kalitesi üzerindeki etkisi

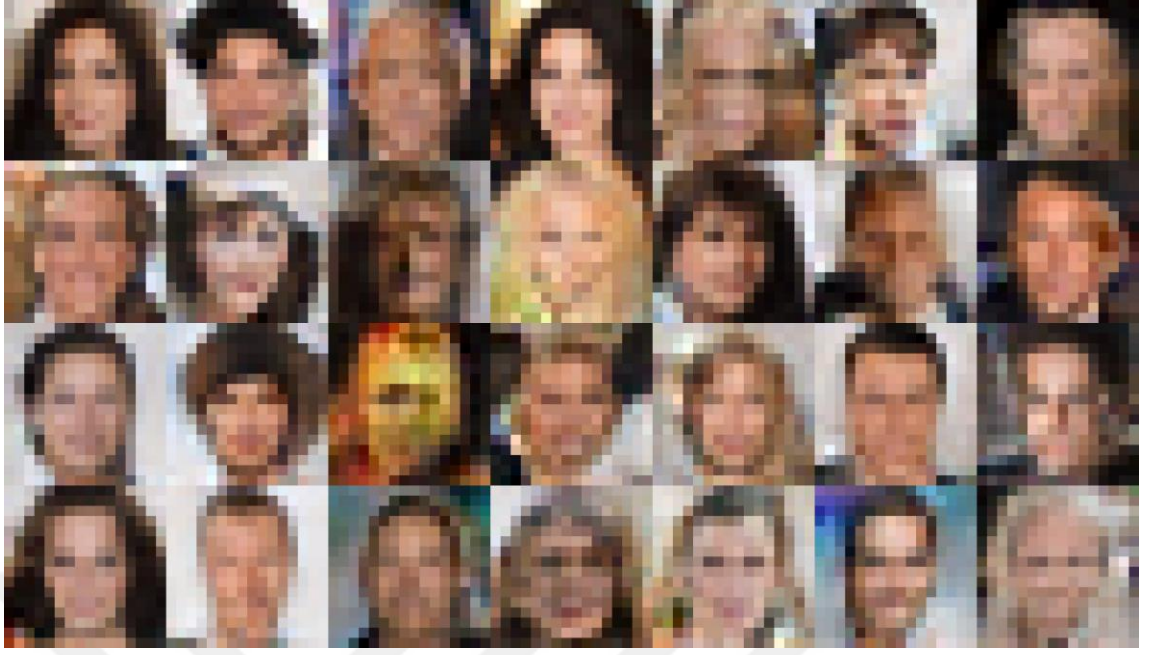
M5 modeli kullanılarak rastgele üretilen imgeler diğer modeller ile görsel olarak karşılaştırıldığında, daha iyi olmalarına rağmen detayların üretilmediği gözlenmektedir. Bu Üretici ağı tek seferde yüksek çözünürlükte imge üretme görevi verilmesinden kaynaklanmaktadır. Bu problemi çözmek için, *PGGAN* mimarisi referans alınmış ve M5 modeli 4x4 çözünürlüğünden başlanarak kademeli bir şekilde eğitilmiştir. Her iki ağ istikrar yakaladığında (600 bin gerçek imge gördüğünde), çözünürlük arttırılmıştır. Bu kademeli büyüme, imgeler hedef çözünürlüğe (128x128) ulaşana kadar sürdürülmüştür. Şekil 3.11 - 3.17 imgelerinde her bir çözünürlük için ağ kararlı duruma geldiğinde rastgele üretilen imgeler gösterilmiştir. Dikkat edilirse, çözünürlük arttıkça imge kalitesi artmakta ve imgedeki detaylar daha iyi yakalanmaktadır.



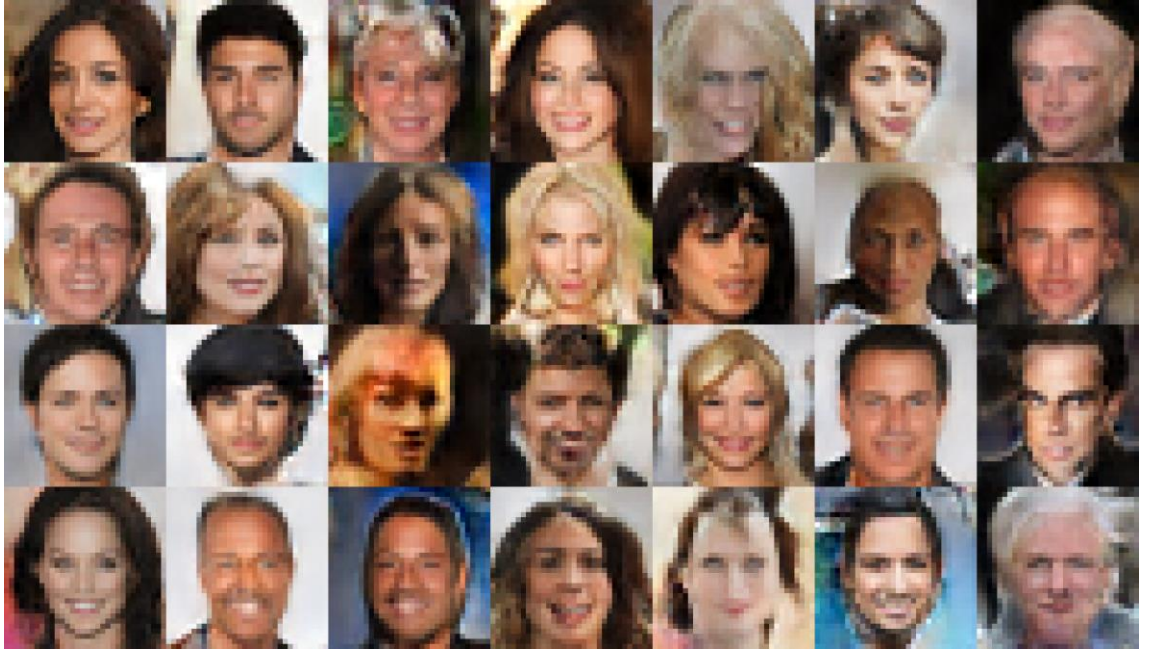
Şekil 3.11 PGGAN çalışması kullanılarak elde edilen ilk imgeler



Şekil 3.12 4x4 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler



Şekil 3.13 8x8 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler



Şekil 3.14 16x16 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler



Şekil 3.15 32x32 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler

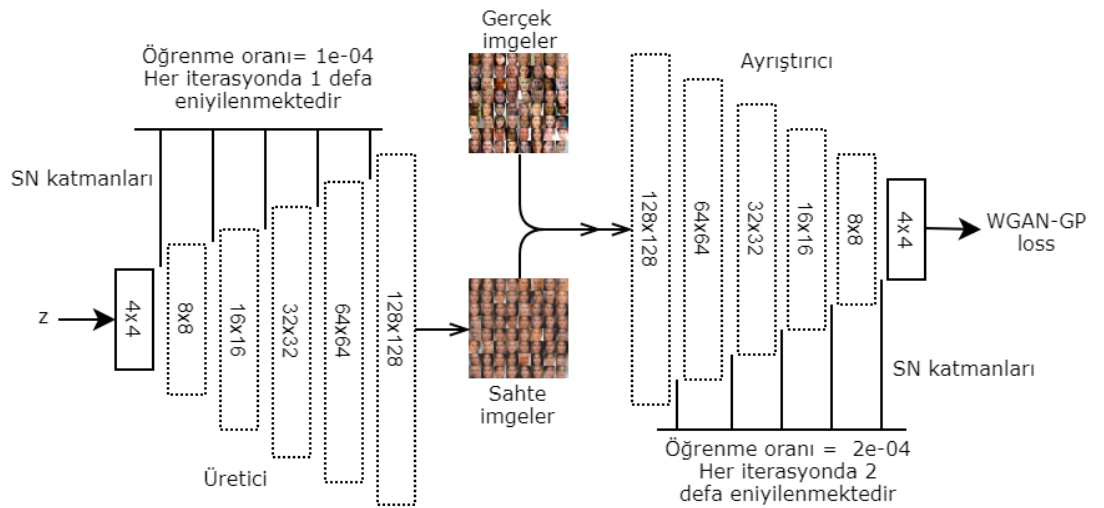


Şekil 3.16 64x64 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler



Şekil 3.17 128x128 çözünürlüğü için ağ istikrarlı olduğunda elde edilen imgeler

Sonuç olarak, M5'ün kademeli olarak eğitilmesi ile *FID* değeri 9.4'e düşürülmüştür. ÜÇA alanında yakın zamanda önerilen yöntemler kullanılarak daha istikrarlı ve daha fazla çeşitliliğe sahip modellerin oluşturulabildiği gözlenmiştir. Fakat ÜÇA mimarisi kullanılan hiperparametrelere göre çok fazla değişkenlik göstermektedir. Bu nedenle hiperparametrelerin en uygun olarak ayarlanması gerekmektedir. Şekil 3.18'de *CelebA* veri seti için nihai olarak kullanılan ÜÇA mimarisi verilmiştir.



Şekil 3.18 CelebA veri seti için oluşturulan ÜÇA mimarisi

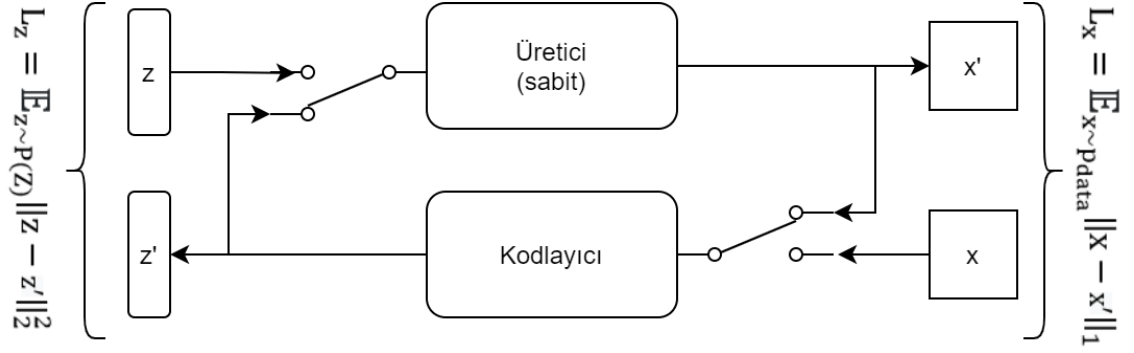
Yapısında gizli temsil barındıran üretken modellerin en önemli avantajlarından biri, iki farklı imge gizli temsilleri arasında aradeğerlemeye (interpolation) olanak sağlaması ve aradeğerleme sonucu üretilen imgelerin, ağların genelleme yeteneği için destekleyici bir kanıt olarak kullanılmasıdır. Şekil 3.19’da Üretici ağ kullanılarak rastgele üretilen imgeler için aradeğerleme sonuçları verilmiştir. Şekil 3.19’da, Üretici ağın aradeğerleme sonuçlarının başarılı olduğu gözlenmektedir.



Şekil 3.19 Üretilen iki imge arasında aradeğerleme sonuçları

3.3 Önerilen Yöntem

Bu kısımda, bir imgenin yeniden yapılandırılması için önerilen model ele alınmıştır. Model iki parametrik ağdan oluşmaktadır: Bunlar, gizli vektör z 'yi imge uzayı x 'e eşleyen Üretici ağı $g_{\omega}(z)$ ve x imgesini veri uzayından gizli vektör z 'ye eşleyen Kodlayıcı ağı $e_{\epsilon}(z)$ 'dir. Amaç, Kodlayıcı ve Üretici ağların gerçek ve sahte imgelerin yeniden yapılandırılması ve kodlanması için çift yönlü eşlemeye sahip olmalarını sağlamaktır. Önerilen modelin tasarımı Şekil 3.20’de gösterilmiştir. Kodlayıcı ağ eğitiminde, yeniden yapılandırma kayıpları, gizli uzayda (Şekil 3.20’de L_z) ve imge uzayında (Şekil 3.20’de L_x) kullanılarak hesaplanmaktadır.



Şekil 3.20 Önerilen CRG mimarisi

CRG eğitim işleminden önce, seçilen veri seti için en iyi skorlar elde edilene kadar ÜÇA mimarisi, yani Üretici ve Ayırıştırıcı ağların eğitilmesi gerekmektedir. Çünkü optimum bir Üretici ağ elde edildikten sonra, imge uzayı x' 'den gizli uzay z' 'ye başarılı bir eşleme yapılabildiği gözlenmiştir. Bu kapsamda, Kısım 3.2'de ÜÇA için en iyi skorlar elde edilene kadar deneysel çalışmalar yapılmıştır. Modelden ilk beklenti, rastgele üretilmiş örneklerin gizli temsillerinin yeniden yapılandırılmasıdır. Verilen rastgele bir z için, Üretici bir x' imgesi oluşturmakta; bu imgenin Kodlayıcı ağı verilmesi ile tahmini bir z' elde edilmektedir. Tahmin edilen z' , yani $e_\epsilon(g_\omega(z))$ ile z arasındaki hatayı en aza indirmek için Denklem 3.4'teki amaç fonksiyonu kullanılmıştır.

$$\hat{\epsilon} = \underset{\epsilon}{\operatorname{argmin}}(\mathbb{E}_{z \sim P(Z)} \|z - e_\epsilon(g_\omega(z))\|_2^2) \quad (3.4)$$

Modelden ikinci beklenti ise gerçek imgeleri yeniden yapılandırmasıdır. Bir özneteliği temsil eden vektör yönünü bulmak için gerçek imgeler kullanılmaktadır. Bu nedenle, modelin gerçek imgeleri kodlaması önemli olmaktadır. Verilen gerçek bir imge x , ilk başta Kodlayıcı ağı girdi olarak verilerek gizli uzayda temsil edilmekte ve bu gizli temsil Üretici ağına verilerek x' , yani, $g_\omega(e_\epsilon(x))$ elde edilmektedir. Modelde, ikinci aşamada ise amaç fonksiyonu 3.5'teki hatanın en aza indirgenmesi için aynı şekilde Kodlayıcı ağ parametreleri eniyilenmektedir.

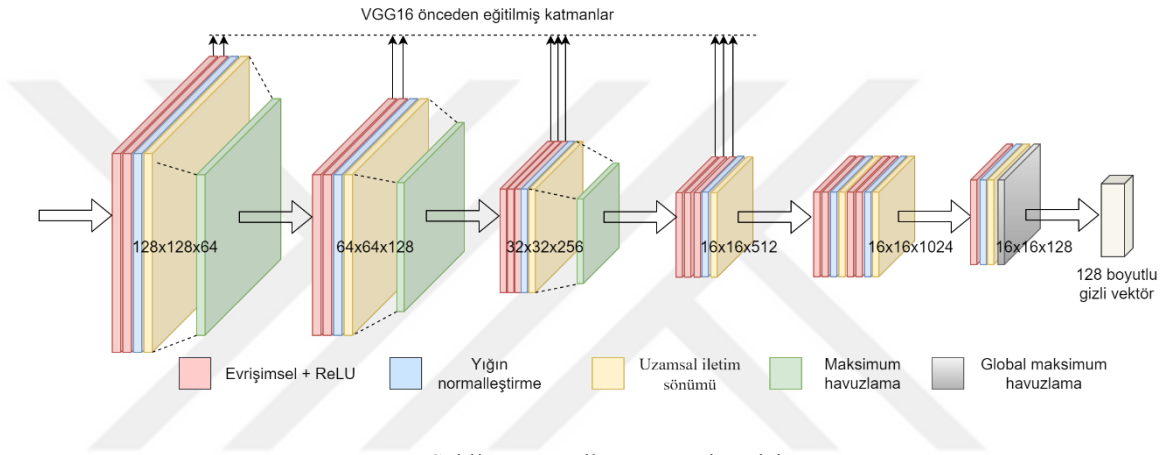
$$\hat{\epsilon} = \underset{\epsilon}{\operatorname{argmin}}(\mathbb{E}_{x \sim P_{data}} \|x - g_\omega(e_\epsilon(x))\|_1) \quad (3.5)$$

Her bir iterasyonda, sırasıyla amaç fonksiyonu 3.4 ve 3.5 kullanılarak Kodlayıcı ağı parametreleri iki defa güncellenmektedir. Modelde; Kodlayıcı ağ hem gerçek hem de üretilen imgeleri, Üretici ağ ise hem rastgele türetilen hem de kodlanmış gizli vektörü girdi olarak almaktadır. Bu durum, gerçek ve üretilen imgelerin yeniden yapılandırılmasına olanak sağlamaktadır. Genel olarak, uçtan uca yaklaşımlar incelendiğinde üç ağı eşzamanlı olarak eğitildiği gözlenmektedir. Bu ağlardan bir tanesinin diğerlerine üstünlük kurması durumunda istikrar problemi ortaya çıkmaktadır. Bu nedenle, önceden eğitilmiş Üretici ağlar bu çalışmalarda kullanılmamaktadır. Ancak önerdiğimiz CRG mimarisinde, herhangi bir istikrar problemi ile karşılaşılmadan, önceden eğitilmiş Üretici ağlar kullanılabilir. Bu durum, farklı veri setleri için önceden eğitilmiş Üretici ağları kullanmaya olanak sağlamaktadır.

Kodlayıcı ağ: Kodlayıcı ağı, çok katmanlı evrimsel sinir ağı olarak tasarlanmıştır. Üretici ağı, yaklaşık olarak 22 milyon eğitilebilir parametre barındırmaktadır. Yapılan deneysel çalışmalarda, genel olarak Kodlayıcı ağ kapasitesini Üretici ağından yüksek tutmanın daha iyi kodlama sağladığı gözlenmiştir. Kodlayıcı ağı, 6 bloktan oluşmakta ve yaklaşık olarak 26 milyon eğitilebilir parametre içermektedir. Derin Evrimsel Sinir Ağ (DESA)'nın sıfırdan eğitilmesi, ağ eğitimi sırasında yakınsama (convergence) sorunlarına neden olabilmekte ve eğitim süresini önemli ölçüde arttırabilmektedir (Zeiler ve Fergus 2014). DESA ilk birkaç katmanda düşük seviyeli özellikleri kodladığından, Kodlayıcı ağı ilk 4 bloğu için önceden eğitilmiş VGG16 (Simonyan ve Zisserman 2014) ağı ağırlık değerlerini kullanılmıştır. Ancak mimaride birkaç değişiklik yapılmıştır. Bunlar:

- İmge çözünürlüğünü çok fazla düşürmemek için dördüncü maksimum havuzlama (max-pooling) katmanını kaldırılmıştır.
- Kodlayıcı ağı son katmanına eklenen ilave katmanlar sonucu meydana gelen gradyan azalması probleminin üstesinden gelmek için her evrimsel bloğun sonuna bir yığın normalleştirme katmanı eklenmiştir.
- Aşırı öğrenmeyi (overfitting) önlemek için yığın normalleştirme katmanlarından sonra uzamsal iletim sönümü (spatial dropout) (Srivastava vd. 2014) katmanı kullanılmıştır.

Modelde ayrıca ağın kapasitesini artırmak ve *CelebA* veri setine özgü özellikleri çıkarmak için iki yeni blok eklenmiştir. Son olarak, Kodlayıcı ağı son katmanında global maksimum havuzlama (global max-pooling) katmanı kullanılmıştır. İlk blokta 64 filtre kullanılmış ve ardışık bloklarda filtre sayısı son blok dışında ikiye katlanmıştır; gizli vektör boyutu 128 olduğundan, son blokta 128 filtre kullanılmıştır. Son evrişimsel katmanında 1x1 filtre boyutu, diğer evrişimsel katmanlarda ise 3x3 filtre boyutu kullanılmıştır. Kodlayıcı ağı genel yapısı Şekil 3.21’de gösterilmiştir.



Şekil 3.21 Kodlayıcı ağ mimarisi

Eğitim detayları: Kodlayıcı ağı önerilen CRG mimarisi kullanılarak eğitilmiştir. Kodlayıcı ağ eğitim aşamasında, önceden eğitilmiş *VGG16* ağının ağırlıkları da dâhil, tüm evrişimsel katman ağırlıkları için aynı öğrenme oranı kullanılmıştır. Eniyileme algoritması olarak *Root Mean Square Propagation (RMSProp)* (Tieleman ve Hinton, 2012) kullanılmış olup, yığın boyutu 128, *rho* değeri 0.9 ve *epsilon* 1e-08 olarak atanmıştır. Kodlayıcı ağındaki tüm parametreleri eniyilemek için küçük bir öğrenme hızı, yani 1e - 4 kullanılmıştır; bu mevcut *VGG16* katmanlarının ağırlık değerlerini doğru bir şekilde uyarlamayı sağlamıştır. Asgari sağlama (validation) kaybı 10 epok için iyileşme göstermediği zaman öğrenme oranı 2 kat azaltılmıştır. Aşırı öğrenmeden kaçınmak için, eğitim aşamasında iki düzenleme tekniği kullanılmıştır. Bunlar, veri arttırma ve %50 uzamsal iletim sönümüdür. Veri arttırmada, imgeler 30 derece aralığında döndürülmüş ve yatay-dikey çevirme işlemleri uygulanmıştır. 20 epok boyunca sağlama kaybında bir iyileşme elde edilmemesi durumunda, eğitim süreci sonlandırılmıştır. Eğitim işlemi

tamamlandıktan sonra modelden beklenti, üretilen bir imgenin hemen hemen aynı şekilde ve gerçek imgelerin kabul edilir seviyede yeniden yapılandırılmasıdır. Kabul edilir seviye; gerçek imgeleri kullanarak bir özniteliğe karşılık gelen gizli vektörü bulabilmektir.

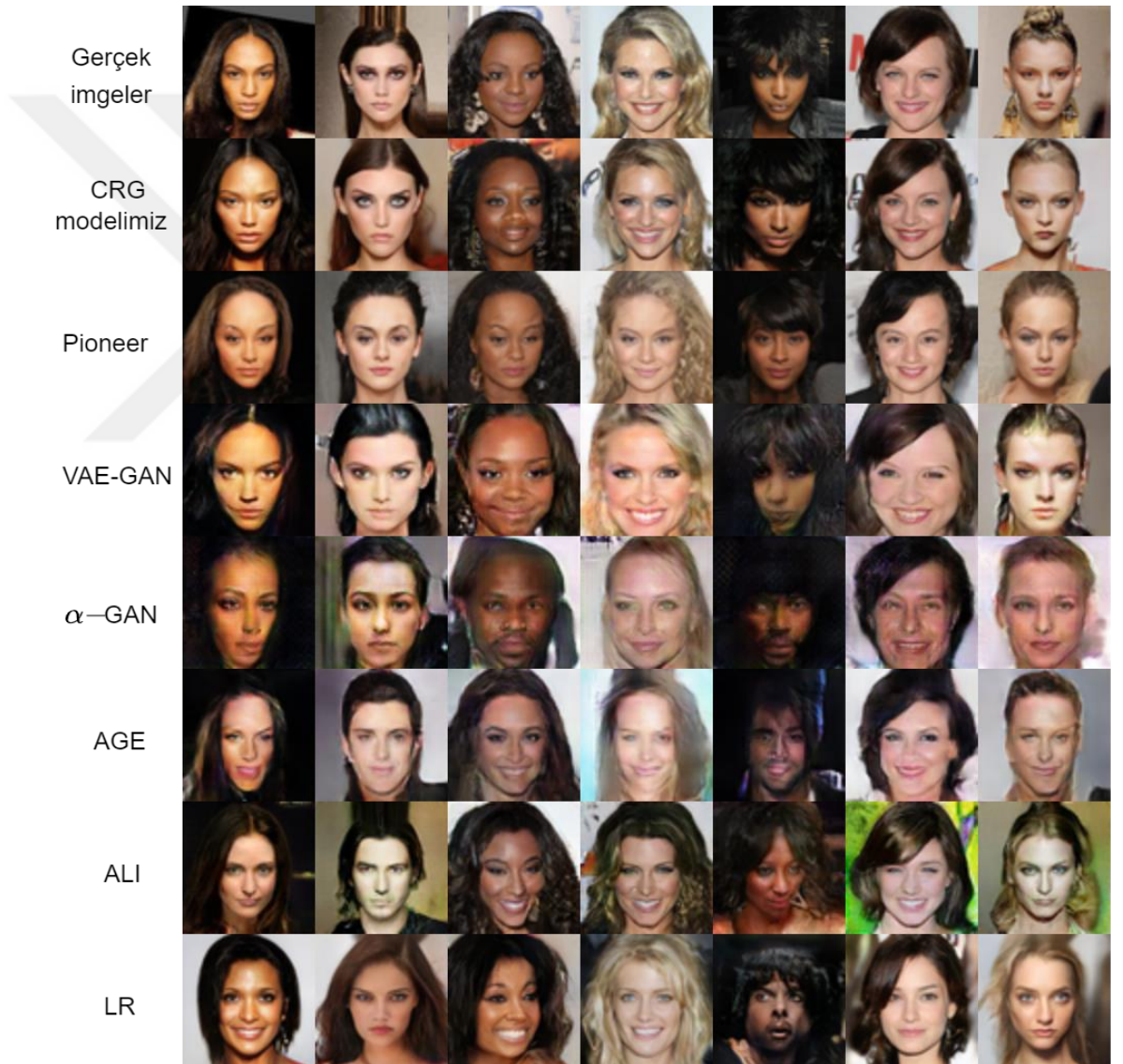
Üretici ağ parametrelerinin güncellenme etkisi: Önerilen CRG mimarisinde eğitim işleminden önce, ÜÇA mimarisi kullanılarak Üretici ağ eğitilmekte ve bu ağın parametreleri sabitlenmektedir. Şekil 3.22’de üst satırdaki imgeler veri setinden alınan, ortadaki imgeler ise önerilen CRG mimarisi kullanılarak yeniden yapılandırılan imgelerdir. Sonuçlar incelendiğinde, yeniden yapılandırılan imgelerin gerçek imgelere benzer ve keskin olduğu gözlenmektedir. İkinci aşamada, Üretici ağ parametrelerinin de eniyilendiği CRG(TG) modeli eğitilmiştir. Şekil 3.22’de en alt satırda bu model kullanılarak elde edilen imgeler gösterilmiştir. Yeniden yapılandırılan imgeler incelendiğinde, sonuçların bulanık olduğu gözlenmektedir. Bu nedenle, çalışmanın devamında Üretici ağ parametrelerinin sabit tutulduğu CRG mimarisi kullanılmıştır.



Şekil 3.22 Gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılması: (birinci sıra) gerçek imgeler, (ikinci sıra) önerilen CRG mimarisi (Üretici parametreleri sabit) ve (üçüncü sıra) CRG(TG) mimarisi (Üretici parametreleri de eğitilmektedir) sonuçları gösterilmiştir

3.4 Modellerin Karşılaştırılması

Herhangi bir özneliğe (saç rengi, gözlük vb.) karşılık gelen gizli vektör yönünü tespit etmek için bir gerçek referans imge çifti (ilgili özneliği içeren ve içermeyen) kullanılması planlanmaktadır. Bu nedenle, gerçek imgenin yeniden yapılandırma performansı, ilgili özneliğe karşılık gelen gizli vektör yönünü doğru bir şekilde elde etme ile orantılı olmaktadır. Bu kısımda, üretken ağlar kullanılarak imgelerin yeniden yapılandırılması için tasarlanmış modeller karşılaştırılmıştır.



Şekil 3.23 Modellerin gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılması. *Pioneer*, *LR* ve *CRG* mimarisi 128x128, diğer modeller ise 64x64 piksel çözünürlüğünde imgeler üretmektedir

Şekil 3.23'te önerilen CRG mimarisi, *Pioneer*¹, *VAE-GAN*², α -*GAN*³, *AGE*⁴, *ALI*⁵ ve *LR* modelleri ile gerçek imgelerin yeniden yapılandırılması bakımından karşılaştırılmıştır. Karşılaştırma için genel olarak bu modellerin önceden eğitilmiş resmi modelleri kullanılmıştır; önceden eğitilmiş modelleri mevcut değilse ya modeller sıfırdan yeniden eğitilmiş ya da orijinal makale ile aynı olacak şekilde tasarlanmış başka bir modelin önceden eğitilmiş modeli kullanılmıştır. CRG, *Pioneer* ve *LR* modelleri için 30 bin imge, diğerleri için ise yaklaşık 200 bin imge kullanılarak eğitim işlemi yapılmıştır. Sonuçlar nitel olarak karşılaştırıldığında, önerilen modelin gerçek imgeleri yeniden yapılandırma bakımından, diğer modellere üstünlük kurduğu gözlenmektedir.

Gerçek ve yeniden yapılandırılan imgeler arasındaki benzerliği nicel olarak karşılaştırmak için fark adresleme (difference hash (*dhash*)) (Niu ve Jiao 2008), algısal adresleme (perceptual hash (*phash*)) (Niu ve Jiao 2008) ve dalgacık adresleme (wavelet hash (*whash*)) (Petrov 2016) metrikleri kullanılmıştır. Bu yöntemlerde, imgeler başlangıçta sabit bir adresleme koduna dönüştürülmektedir. Ardından, orijinal imge adresleme kodu ile yeniden yapılandırılan imgenin adresleme kodu arasındaki benzerlik, iki imgenin ne kadar benzer olduğunu belirlemek için kullanılmaktadır. Çizelge 3.5'te, daha önceden belirtilen modeller için bu algısal değerlendirme ölçütleri ile Ortalama Mutlak Hata (OMH) ve Ortalama Karesel Hata (OKH) skorları verilmiştir. *dhash*, *phash* ve *whash* metrikleri, algısal olarak daha alakalı skorlar üretmektedir; yeniden yapılandırılan imgeler keskin ve girdi imgesine benzer olduğunda skor yükselmektedir. Öte yandan, OMH ve OKH metrikleri piksel bazındaki farkların ortalamalarını sağlamaktadır. Çizelge 3.5'ten de görüldüğü üzere, CRG'nin algısal skorları diğer tüm yöntemlerden daha iyidir; ancak CRG'nin OMH ve OKH skorları *Pioneer* modelinden daha düşüktür. *Pioneer* modelinin bulanık yeniden yapılandırma sonuçları piksel bazında daha az hatayla sonuçlanmaktadır. OMH ve OKH skorları CRG (TG) modeli kullanılarak hesaplandığında ise sırasıyla 0.215 ve 0.034 değerleri elde edilmiştir. Bu skorlar diğer yöntemlerden daha iyi gibi görünse de CRG (TG)'nin yeniden

¹ <https://github.com/AaltoVision/pioneer>

² <https://github.com/PrateekMunjhal/Autoencoding-beyond-pixels-using-a-learned-similarity-metric>

³ <https://github.com/PrateekMunjhal/Variational-Approaches-for-Auto-Encoding-Generative-Adversarial-Networks>

⁴ <https://github.com/DmitryUlyanov/AGE>

⁵ <https://github.com/IshmaelBelghazi/ALI>

yapılandırma sonuçları, CRG'ye kıyasla belirgin şekilde daha bulanıktır. Bu nedenle, OMH ve OKH metriklerinin imgelerin yeniden yapılandırma kalitelerini karşılaştırmak için çok yararlı olmadıkları görülmüştür. Algısal metrikler, Şekil 3.23'ü desteklemekte ve önerilen CRG mimarisinin, gerçek imgelerin yeniden yapılandırılması bakımından, diğer modellerden daha iyi olduğunu göstermektedir.

Çizelge 3.5 Gerçek imgelerle yeniden yapılandırılan imgelerin benzerlikleri. *dhash*: Difference hash, *phash*: Perceptual hash, *whash*: Wavelet hash, OMH: Ortalama mutlak hata, OKH: Ortalama karesel hata

Model	dhash	phash	whash	OMH	OKH
CRG (TG)	0.846	0.792	0.855	0.215	0.034
CRG	0.871	0.848	0.892	0.306	0.067
<i>Pioneer</i>	0.852	0.804	0.864	0.261	0.051
VAE-GAN	0.842	0.779	0.861	0.355	0.095
α -GAN	0.837	0.756	0.855	0.443	0.122
AGE	0.832	0.754	0.863	0.449	0.136
ALI	0.759	0.668	0.745	0.487	0.159
LR	0.697	0.585	0.674	0.613	0.263

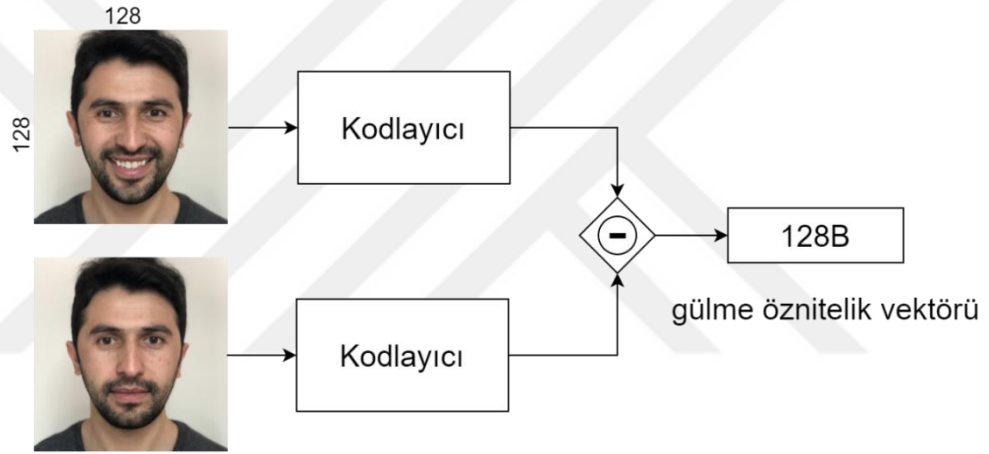
Üretilen bir imgenin herhangi bir öz niteliğini değiştirmek için üretilen imgenin doğru bir şekilde yeniden yapılandırılması gerekmektedir. Şekil 3.24'te, önerilen model kullanılarak üretilen bazı imgelerin yeniden yapılandırılma sonuçları verilmiştir. Örneklerden görüldüğü gibi Kodlayıcı ağı, Üretici ağı karşılıklı eşlemesini başarıyla öğrenmiştir.



Şekil 3.24 Üst sıra: rastgele üretilmiş imgeler, alt sıra: önerilen model kullanılarak yeniden yapılandırılan imgeler

3.5 Gizli Öznitelik Temsillerinin Elde Edilmesi

CRG mimarisinde, Kodlayıcı ağ denetimsiz olarak eğitilmektedir; ÜÇA eğitimi sırasında hiçbir etiket veya koşullu ÜÇA mimarisi kullanılmamaktadır. Bir özniteliğe karşılık gelen gizli vektörü elde etmek için, referans olarak keyfi gerçek bir imge ve kontrol etmek istediğimiz belirli bir özniteliği içeren ikinci bir imge kullanılmaktadır. Şekil 3.25'te gülme özniteliğine karşılık gelen gizli vektör yönünün nasıl elde edildiği gösterilmiştir. İlk önce, Kodlayıcı ağ kullanılarak referans imgelere karşılık gelen gizli vektörler elde edilmektedir. Daha sonra, bu gizli vektörler kullanılarak ilgili öznitelik için normalleştirilmiş yön belirlenmektedir.



Şekil 3.25 Gülme özniteliğine karşılık gelen gizli vektörün çıkarılması

Üretici ağ ve karşılıklı eşleme fonksiyonunun (Kodlayıcı ağ) doğruluğu yüksek olduğunda, bu yön imgedeki diğer öznitelikleri koruyarak yalnızca ilgili özniteliği doğru bir şekilde değiştirmektedir. Bir f özniteliğini temsil eden gizli vektör yönü z_f 'yi bulmak için Denklem 3.6 kullanılmaktadır.

$$z_f = \frac{z_2 - z_1}{\|z_2 - z_1\|_2^2} \quad (3.6)$$

Denklem 3.6'da, z_1 bir referans imge kullanılarak Kodlayıcı tarafından kodlanan, z_2 ise istenen özniteliği içeren bir imge kullanılarak elde edilen gizli kodu temsil etmektedir. z_f elde edildikten sonra herhangi bir imgede istenen özniteliği değiştirmek için Denklem 3.7 kullanılmaktadır. Duruma bağlı olarak öznitelikler pozitif veya negatif

yönde kodlanabilmektedir. Örneğin, sarı saç kahverengi saça dönüştürmek için $(z_1 - z_2)$ kullanılıyorsa tersi işlemi yapmak için, yani kahverengi saç sarı saça dönüştürmek için $(z_2 - z_1)$ kullanılabilir. Bunun için Denklem 3.6'da ilgili değişikliğin yapılması gerekmektedir. z_f elde edildikten sonra istenen özneliğe sahip bir imge oluşturmak için Denklem 3.7 kullanılmaktadır.

$$z_a = z_p + k \times z_f \quad (3.7)$$

Denklem 3.7'de z_p , herhangi rastgele bir imge kullanılarak Kodlayıcı tarafından tahmin edilen gizli vektör, k imgeye empoze etmek istediğimiz öznelik miktarı, z_a ise özneliğin kodlandığı gizli vektörü temsil etmektedir. Öznelik kodlama, istenildiği gibi Denklem 3.7'de k 'nın işareti değiştirilerek pozitif veya negatif yönlerde gerçekleştirilebilmektedir. Bilindiği kadarıyla, ÜÇA'ları kullanan mevcut yeniden yapılandırma modelleri henüz imgelerde yalnızca seçilen öznelikleri doğru bir şekilde kontrol etme yeteneğine sahip değildir. Bu yöntemi kullanarak bir imgede istenen özneliği değiştirmek bazı avantajlar sağlamaktadır:

- Öznelikleri kodlamak için etiketli veri setine ihtiyaç duyulmamaktadır.
- Öznelik kümesi sabit değildir; bir çift referans imgesi sağlanması durumunda herhangi bir öznelik vektörü elde edilebilmektedir.
- Hedef imgeye öznelik empoze etme miktarı, bir k parametresi ayarlanarak kolayca kontrol edilebilmektedir.

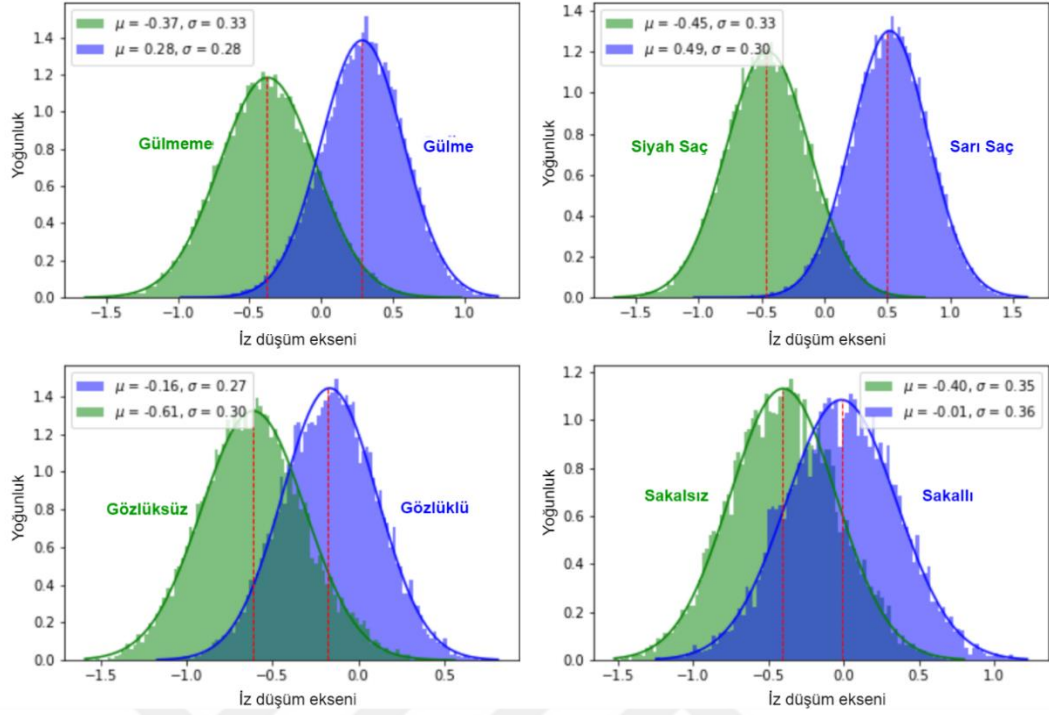
3.6 Öznelik Yönlerinin Analizi

Bu kısımda, *CelebA* gerçek veri setinden rastgele alınan ve ÜÇA eğitim aşamasında kullanılmayan, ilgili bir özneliği içeren ve içermeyen rastgele imgeler kullanılarak bir dizi özneliğin (gülme, saç rengi, gözlük ve sakal) deneysel bir değerlendirilmesi sağlanmıştır. *CelebA* veri setinden seçilen bu imgelerin etiketi bilinmektedir. Bu etiketli imgeler, sadece ilgili özneliği içeren ve içermeyen imgelerin izdüşüm eksenindeki konumlarını analiz etmek için kullanılmıştır. Her bir öznelik⁶ için yarısı ilgili özneliği

⁶ Gözlük özneliği içeren örnek sayısı yetersiz olduğundan 13000 imge kullanılmıştır.

içeren ve diğer yarısı içermeyen olacak şekilde toplamda 40000 imge kullanılmıştır. Burada gözlemlemek istenilen durum, ilgili özniteliği içeren ve içermeyen imgelere karşılık gelen vektörlerin, bu öznitelikler için elde edilen gizli vektörlere izdüşümü alındığında, izdüşüm uzayındaki konumlarını analiz etmektir. Önerilen yönteme göre ilgili özniteliği içermeyen bir imgeye karşılık gelen gizli vektör, herhangi bir özniteliğe karşılık gelen vektör yönünde hareket ettirildiğinde ilgili özniteliğin kademeli olarak değişeceği; bu nedenle izdüşüm uzayında iki farklı dağılımın görülmesi beklenmektedir. Bu dağılımlardan ilki ilgili özniteliği içeren imgelerin, ikincisi ise bu özniteliği içermeyen imgelerin oluşturduğu dağılım olması beklenmektedir.

Bir önceki kısımda (Kısım 3.5) tanımlandığı gibi her bir öznitelik için 50 farklı referans imge çifti kullanılarak öznitelik vektörleri elde edilmiş, daha sonra bunların ortalaması alınmıştır. Şekil 3.26 incelendiğinde, her bir öznitelik için elde edilen öznitelik vektörlerine izdüşüm alındığında iki farklı dağılımın olduğu görülmüştür. Bu dağılımlarda, yeşil renk ile boyanan dağılım ilgili özniteliği içermeyen, mavi ile boyanan dağılım ise ilgili özniteliği içeren imgelerin oluşturduğu dağılımı göstermektedir. Şekil 3.26'da sol-üst gülme, sağ-üst saç rengi, sol-alt gözlük ve sağ-alt sakal özniteliğine karşılık gelen dağılımları göstermektedir.



Şekil 3.26 Gizli öz nitelik vektörleri üzerine iz düşümler: sol-üst: gülme, sağ-üst: saç rengi, sol- alt: gözlük, sağ-alt: sakal öz niteliği

CelebA veri setinde ilgili öz niteliği içerdiği şeklinde etiketlenmiş (diğer bir deyişle, gülümseme, sarı saç, gözlüklü veya sakallı olarak etiketlenmiş) imgelerin ilgili öz niteliklere iz düşümleri alındığında, dağılımın farklı konumlarına düştükleri gözlenmiştir. Şekil 3.27’de, her bir öz nitelik için, Normal dağılımın sağ ve soldaki uç kısımlarına (yani, $\mu \pm 3\sigma$ ’nın ötesine) ve ortasına ($\cong \mu$) düşen bazı imgeler gösterilmiştir. Şekil 3.26’da mavi ile boyalı dağılımlardaki her bir imge, *CelebA* veri setinde ilgili öz niteliğe sahip olarak etiketlenmiştir. Ancak bu dağılımlardan mavi olanının sol-uç ve yeşil dağılımın, sağ-uç bölgelerindeki imgelerin doğru etiketlenip etiketlenmediğinin incelenmesi gerekmektedir. Şekil 3.27’de, her bir öz nitelik için, üst satırdaki imgeler ilgili öz niteliği içeren imgelerin oluşturduğu dağılımın (mavi dağılım) sağ-uç kısmından ($\mu + 3\sigma$) alınmıştır. Bu imgeler incelendiğinde, imgelerin iz düşümü, beklendiği gibi ilgili öz niteliği içermeyen dağılımın ortalamasından uzak olduğu ve ilgili öz niteliği güçlü bir şekilde sağladığı gözlenmiştir. Orta satırdaki imgelerin iz düşümü, mavi dağılımın ortalamasına yakın olmaktadır. Bu imgeler incelendiğinde, ilgili özelliği sağladığı; ancak sağ-uç noktalardan alınan imgeler kadar güçlü bir şekilde öz niteliği

barındırmadığı görülmüştür. Alt satırdaki imgeler ise mavi dağılımın sol-uç kısmından ($\mu - 3\sigma$) alınmıştır. Dikkat edilirse, bu imgelerin izdüşümü, ilgili özniteliği içermeyen dağılımın ortalamasına yakın veya ilerisindedir. Bu bölgeden alınan imgelerin ilgili özniteliği çok az içerdiği veya tamamen içermediği gözlenmektedir. Bu imgeler *CelebA* veri setinde ilgili öznitelige sahip olarak etiketlenmelerine rağmen çoğunun özniteliği barındırmadığı gözlenmektedir. Örneğin, Şekil 3.27’de sol-alt kısımda gözlüklü olarak etiketlenen imgelerin, gözlüklü olmadıkları gözlenmektedir. Bu durum, diğer öznitelikler için de benzer olmaktadır. Bu durum *CelebA* veri setinde yanlış etiketlemeden kaynaklanmaktadır. Önerilen yöntemde, veri setindeki etiketler kullanılmadığından, model bu yanlış etiketlemeden etkilenmemektedir. Ancak bu durum koşullu ÜÇA mimarisi ve etiketli veri seti kullanan modelleri etkileyebilmektedir.

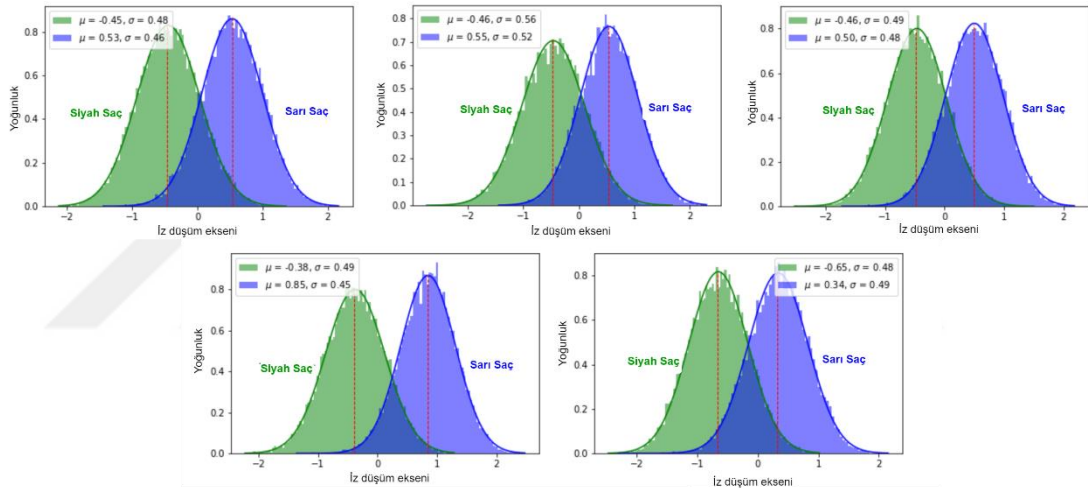


Şekil 3.27 Dört farklı öznitelik için mavi ile boyanan dağılımın orta ve uç noktalarından alınan örnekler

3.7 Öznitelik Kodlamada Rastgele Referans İmgelerin Etkisi

Bu kısımda amaç, farklı referans imge çiftlerini kullanarak aynı öznitelik için elde edilen öznitelik vektörlerinin tutarlılığını test etmektir. Bu kapsamda herhangi bir öznitelik için ortalama öznitelik vektörü yerine, farklı referans imgeler kullanılarak elde edilen

öznitelik vektörlerine izdüşüm alınmıştır. Burada amaç, farklı referans imgelerinin tutarlılığını incelemektir. Bu amaç doğrultusunda, saç rengi özneliği için ortalama öznitelik vektörü elde edilirken kullandığımız referans imgelerden rastgele 5 tanesi kullanılarak bunların tutarlılığı deneysel olarak incelenmiştir. Şekil 3.28’de bu 5 referans image ayrı ayrı kullanılarak elde edilen öznitelik vektörlerine izdüşüm sonuçları gösterilmiştir. Dikkat edilirse, ortalama ve standart sapmalardaki çok küçük farklılıkların dışında, Şekil 3.26’ya benzer olarak iki farklı dağılımın olduğu görülmüştür. Bu gözlemler, belirli bir özneliğin birbirine yakın bir dizi doğrusal yön ile kodlanabileceğini göstermektedir; dolayısıyla, ilgili öznitelik yönlerin tamamıyla farklı olmadığı ve her öznitelik için bir dizi olası yön olduğu gözlenmiştir.



Şekil 3.28 Saç rengi özneliği için rastgele 5 farklı referans image çifti kullanılarak elde edilen dağılımlar

Çalışmanın devamında, öznitelik düzenlemede keyfi referans image seçiminin etkisini göstermek için görsel örnekler sunulmuştur. Şekil 3.29’da 4 farklı öznitelik (gülme, saç rengi, gözlük ve sakal) için elde edilen sonuçlar verilmiştir. Şekil 3.29’da her bir öznitelige karşılık gelen çerçevelerin solundaki imgeler, bu özneliklere karşılık gelen gizli vektör yönlerini bulmak için kullanılan referans imgelerdir. Sağdaki imgeler ise ilgili özneliği içermeyen bir imgeye bu özneliğin uygulanması sonucu verilmiştir. Tüm örnekler için k değeri 2 olarak alınmıştır.

Bir özneliğe karşılık gelen gizli vektörü elde etmek için kullanılan referans imgeler oldukça farklı olmasına rağmen tüm seçilen referans imgelerde *nötr* ve ilgili özneliği

içeren imgeler tutarlı olduğundan, elde edilen gizli vektör yönlerinin tutarlı ve benzer olduğu görülmüştür. Sonuçlar incelendiğinde, referans imgelerin farklı olmasından kaynaklı, üretilen imgelerde küçük farklılıkların (arka plan, saç kenarları vb.) olduğu gözlenmiştir. Bu gibi farklılıklar, hesaplanan gizli vektör yönlerindeki küçük varyasyonlardan kaynaklanmaktadır ve beklenen bir durumdur. Bu sonuçlar, herhangi bir öznitelik karşılık gelen gizli vektör yönünü hesaplamak için farklı referans imgelerin kullanabileceğini göstermektedir.



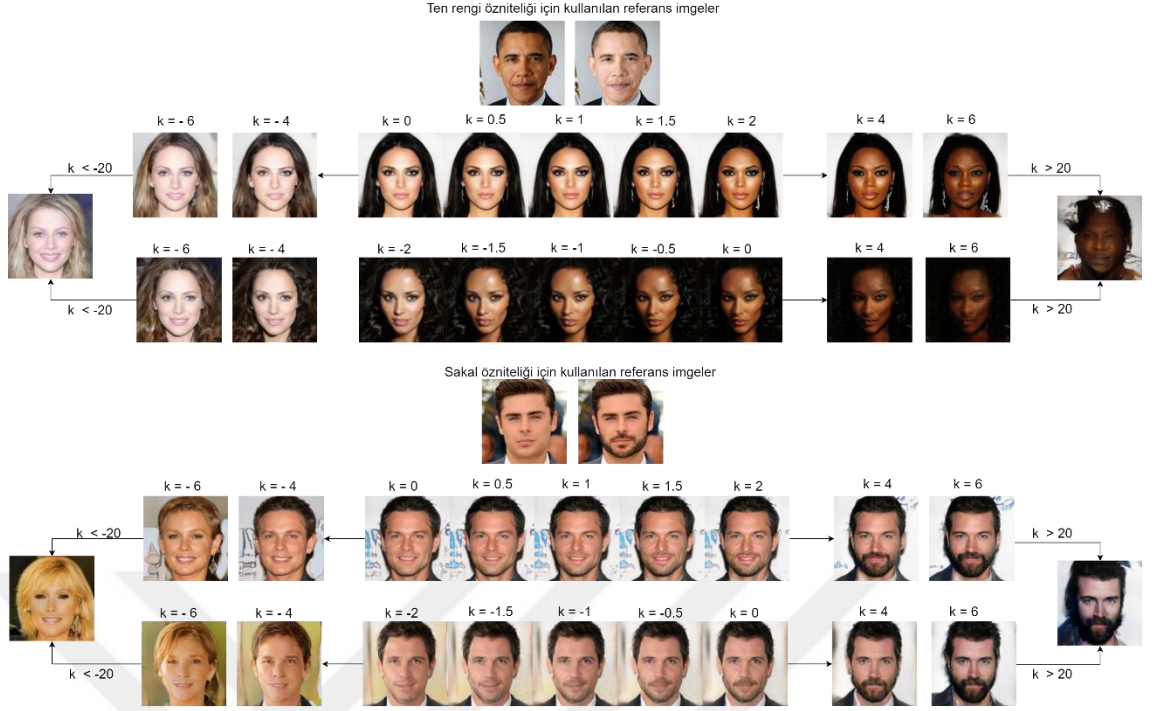
Şekil 3.29 Dört öznitelik için farklı referans imge çiftleri kullanılarak öznitelik düzenleme. Sol üst çerçeve: gülme, sağ üst çerçeve: saç rengi, sol alt çerçeve: gözlük, sağ alt çerçeve: sakal

3.8 k Parametresi için Etkili Aralığın Belirlenmesi

Bu kısımda, bir öznitelik yönünde imgelerin nasıl değiştiğini anlamak için, k 'nın geniş bir aralığında pozitif ve negatif yönlerde bir dizi gözlem gerçekleştirilmiştir. Şekil 3.30'da, ten rengi ve sakal öznitelikleri için sonuçlar gösterilmiştir. Ten rengi öznitelğinde görüldüğü üzere, ilgili öznitelik beklendiği gibi pozitif/negatif yönlerde kademeli olarak artmakta/azalmaktadır. Üretilen bir imgenin komşuluklarına

bakıldığında, ten rengi ve sakal dışındaki özniteliklerin korunduğu görülmüştür. Öznitelik vektörü yönünde k değeri artırmaya/azaltmaya devam edildiğinde, yüz imgelerinin yavaş yavaş farklı yüzlere dönüştüğü gözlenmektedir. Bu nedenle, yüz özelliklerini değiştirmeden, ilgili bir özniteliği düzenlemek için belirli bir imgenin yerel komşuluğunda kalınması gerekmektedir.

Herhangi bir öznitelik yönündeki büyük ilerlemelerde, yani $k > 20$ veya $k < -20$ gibi büyük k değerleri için, başlangıç imgesinden bağımsız olarak tüm imgelerin sol ve sağ uçlarda belirli bir çift yüz imgesine yakınsadığı gözlenmiştir. Aslında, bu öznitelik ekseninde bir son olmadığı; ancak bir noktadan sonra eksen üzerindeki aradeğerlemenin yeni imgeler üretmediği görülmüştür. Bu şaşırtıcı bir durum değildir. Çünkü gizli uzaydaki bu eksenlerin en uzak uçlarında, ÜÇA eğitimi sırasında kullandığımız z örnekleme nedeniyle (z 'nin her bir bileşeni $\mu_{z_i} = 0$ ve $\sigma_{z_i} = 1$ olan bir Normal dağılımdan örneklenmiştir) Üretici ağ tarafından gözlemlenen neredeyse hiç örnek olmamaktadır. z bileşenlerinde çok ileri gidildiğinde, yani $\mu_{z_i} \pm 3\sigma_{z_i}$ aşıldığında, üretilen imgelerdeki değişiklik olasılığı sıfıra çok yaklaşmaktadır. Test edilen tüm öznitelik vektörlerinde, benzer durum gözlenmiştir. Genel olarak, küçük adımlarla bir öznitelik ekseninde pozitif veya negatif yöne gidildiğinde, $k < 2$ olduğu sürece diğer girdi özniteliklerinin orijinal girdiye çok benzer kaldığı gözlenmiştir.



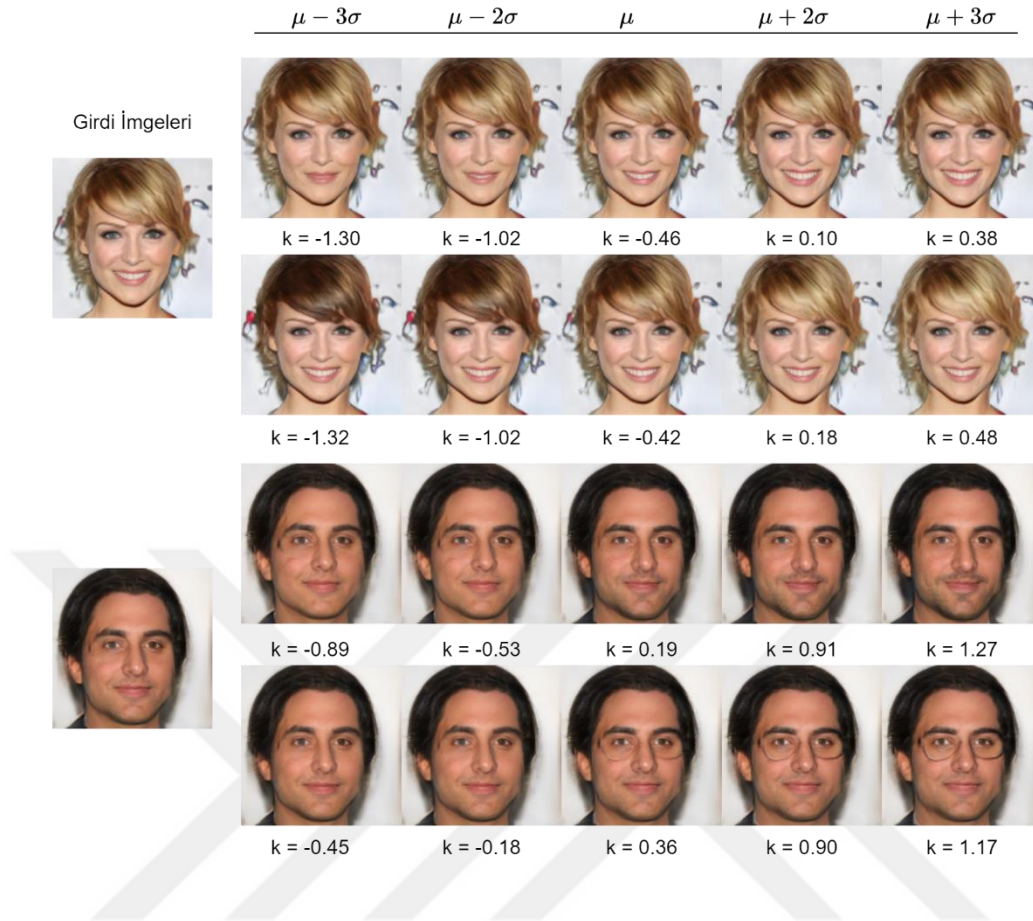
Şekil 3.30 k parametresi için değer aralığının belirlenmesi. Üst: farklı k değerleri için ten rengindeki değişimi, alt: sakal özneliğindeki değişimi göstermektedir. Her bir özellik için kullanılan referans imgeler, her bir örneğin üstünde verilmiştir. $k = 0$ başlangıç imgesine karşılık gelmektedir

Öznitelik değiştirme uygulanacak herhangi bir imge ilgili özneliği içermiyorsa genel olarak $k < 2$ hipotezi iyi çalışmaktadır. Fakat seçilen imge ilgili özneliği içeriyorsa öznitelik yönündeki küçük değişimler bile imgenin bozulmasına neden olabilmektedir. Bu nedenle, k için daha sistematik bir çözüm sunulması gerekmektedir. İmgeye özel bir k aralığını belirlemek için, Şekil 3.26’da elde edilen dağılımlardan yararlanılabilir. Burada, esas amaç k için güvenli sınırları belirlemektir. Bu şekilde, girdi olarak alınan imge ilgili özneliği sağlıyorsa daha küçük k değerleri ile imgedeki temel öznitelikler korunabilir. Şekil 3.26’da daha önceden belirtildiği gibi imgelerin hesaplanan öznitelik vektörlerine izdüşümü alındığında, biri *nötr* ve diğeri ilgili özneliği içeren imgelerin oluşturduğu iki dağılım oluşmaktadır. Bu dağılımlardaki her öznitelik için elde edilen ortalama ve standart sapmalardan yararlanılarak k parametresi için güvenli bir aralık tanımlanmıştır. Şekil 3.27’de görüldüğü gibi öznitelik dağılımlarının sol uç noktalarında, yani 3σ ’nın ilerisinde, öznitelikler çok az görünürken sağ uç noktalarında ise öznitelikler belirgin bir şekilde görünmektedir. Buradan yola çıkarak öznitelik ekseninde $\mu \pm 3\sigma$ ’nın ötesine geçmek yüz özniteliklerinde fark edilebilir değişimler göstermektedir. Bu

şekilde, gizli uzaydaki herhangi bir z 'nin öznitelik eksenindeki izdüşümü ile öznitelik eksenindeki $\mu \pm 3\sigma$ 'ya uzaklığı hesaplanarak k aralığı formüle edilebilir. Daha sonra, k parametresi bu aralıkta tutularak bir imgenin yalnızca ilgili özniteliği güvenli bir şekilde düzenlenebilir.

Şekil 3.31'de daha önceden analiz ettiğimiz öznitelik kümesi için, iki örnek imge üzerinde test işlemi yapılmıştır. Çalışmada, ilk başta Kodlayıcı ağı kullanılarak bu imgelerin z vektörleri elde edilmiştir. Daha sonra, her bir öznitelik için bağımsız olarak z vektörlerinin izdüşümleri (yani a) hesaplanmıştır. Nihai olarak, imgelerin izdüşümü ile öznitelik dağılımlarının $\mu_a \pm 3\sigma_a$ 'ya uzaklığı hesaplanarak k için güvenli sınırlar belirlenmiştir. Hesaplanan aralıklarda, oluşturulan imgeleri gözlemlemek için, k değeri öznitelik eksenleri üzerinde μ_a , $\mu_a \pm 2\sigma_a$ ve $\mu_a \pm 3\sigma_a$ noktalarına denk gelecek şekilde ayarlanmıştır. Her iki imgenin orijinal z değerleri ve Kodlayıcı ağı kullanılarak ilgili dört öznitelik için tanımlanan k değerleri için öznitelik düzenleme gerçekleştirilmiştir. Şekil 3.31'de elde edilen sonuçlar gösterilmiştir. İmgeler incelendiğinde, k değeri öznitelik dağılımlarının $\mu \pm 3\sigma$ 'sı arasındaki değerlere ayarlandığında, öznitelik düzenlemenin yalnızca istenen öznitelikleri düzenlediği ve diğer yüz özniteliklerini başarıyla koruduğu görülmüştür.

Dağılım parametreleri incelendiğinde, *nötr* bir imgenin atfedilen dağılımın sağ kuyruğuna (yani $\mu + 3\sigma$) ortalama mesafesi, seçilen dört öznitelik için 1.26 ile 1.84 arasında değişkenlik göstermektedir. Ancak girdi imgesi hali hazırda ilgili özniteliğe sahipse bu aralık daralmaktadır. Bu gibi durumlarda, orijinal girdi imgesinin izdüşümü hali hazırda öznitelik ortalamasına çok yakın olacağından, hedef öznitelikte çok küçük bir değişim olmaktadır. Bu değerler gözlemlerimizle tutarlıdır. Deneyisel çalışmalarımızda, *nötr* imgelerde bazı öznitelikleri düzenlemek için $k = 2$ 'nin genellikle diğer yüz özniteliklerini değiştirmeden çalıştığı görülmüştür. Bu deneylerde, seçilen öznitelik dağılımlarının aralığı 2'den az olduğu için, Şekil 3.31'de elde edilen sonuçlarda imge düzenleme sırasında imgenin diğer yüz özniteliklerinin korunduğu gözlenmiştir.



Şekil 3.31 Sol sütun: örnek girdi imgeleri, sağ sütun: dört öznelik için, k 'nın farklı değerleri kullanılarak düzenlenmiş imgeleri göstermektedir

3.9 Değerlendirme

İmge öznelik düzenleme çalışmaları incelendiğinde, genel olarak *CelebA* veri setinde tanımlanmış etiketli veri seti kullanılmaktadır. *CelebA* veri setinde 40 öznelik için etiket bilgisi sağlanmıştır. Çizelge 3.6'da bu öznelikler verilmiştir.

Çizelge 3.6 CelebA veri kümesinde tanımlanan tüm öznitelikler

No	Tanım	No	Tanım	No	Tanım	No	Tanım
1	Gölge	11	Bulanıklık	21	Cinsiyet	31	Favoriler
2	Kemerli kaşlar	12	Kahverengi saç	22	Ağız biraz açık	32	Gülme
3	Çekici	13	Gür kaşlar	23	Bıyık	33	Düz saç
4	Göz altı torbalanma	14	Tombul	24	Kısık gözler	34	Dalgalı saç
5	Kel	15	Gerdan	25	Sakalsız	35	Küpe takma
6	Kâkül	16	Gözlük	26	Oval yüz	36	Şapkalı
7	Büyük dudak	17	Top sakal	27	Soluk ten	37	Ruj Sürme
8	Büyük burun	18	Gri saç	28	Sivri burun	38	Kolye takma
9	Siyah saç	19	Ağır makyaj	29	Açık alın	39	Kravat takma
10	Sarı saç	20	Çıkık elmacık kemikleri	30	Pembe yanaklar	40	Yaş

Önerilen yaklaşım kullanılarak etiketli veri seti kullanılmadan, toplamda 15 öznitelik için düzenleme yapılmıştır. Bunlar; cinsiyet, yaş, kâkül, dalgalı saç, düz saç, kel, saç uzunluğu, sakal, gülme, gözlük, güneş gözlüğü, sarı saç, siyah saç, kahverengi saç ve ten rengi'dir. EK Şekil 1 - Şekil 11'de bu öznitelikler için elde edilen sonuçlar gösterilmiştir. Bilindiği kadarıyla, bu öznitelikler mevcut yüz öznitelik düzenleme çalışmalarında kullanılan hemen hemen tüm öznitelikleri kapsamaktadır. Diğer öznitelikler için sonuç gösterilmemesinin sebepleri:

- Bu özniteliklerin bazıları belirsiz veya yüz imgelerinde görsel özniteliklerle yorumlanması zor olmaktadır. Örneğin; bulanıklık, gölge, çekici vb.
- Önerilen yöntemde, bir özniteliği temsil eden gizli vektör yönü bulunurken aynı imge için ilgili özniteliği içeren ve içermeyen olmak üzere bir referans imge çifti kullanılmaktadır. Bu özniteliklerden bazıları için referans imge oluşturmak zor olmaktadır. Örneğin; oval yüz, büyük dudak, büyük burun, gür kaşlar, çıkık elmacık kemikleri, sivri burun vb.
- Bu özniteliklerden bazıları, veri setinde sadece yüz kısmı kırıldığı (crop) için kullanımı zor hale gelmiştir. Örneğin; kravat takma, tombul, kolye takma vb.
- Önerilen model CelebA veri kümesinde sınırlı sayıda imge kullanılarak (30 bin) eğitildiği için, model bazı öznitelikleri kodlamakta zorlanmaktadır. Örneğin; top sakal, bıyık, gri saç, ruj sürme, küpe takma vb.

Önerilen yöntemde daha önceden belirtildiği gibi aynı kişi için, ilgili özneliği içeren ve içermeyen olmak üzere bir referans imge çifti kullanılmaktadır. Ancak cinsiyet özneliğinin kodlanmasında, referans imge çiftlerini tanımlamak zor olmaktadır. Çünkü referans imge seçilirken rastgele bir kadın ve erkek yüz imgesi kullanıldığında, diğer özneliklerin birçoğu düzenleme esnasında değişime uğramaktadır. Bu nedenle, referans imgeleri oluşturmak için, rastgele bir erkek yüz yerine bir kadın yüzüne yapay olarak sakal eklenmiştir. Düzenlemeden sonra ortaya çıkan imgelerde, birincil yüz özneliklerinin korunduğu, yani üretilen erkek/kadın imgelerinin ilk yüz imgesine çok benzediği ve cinsiyetin değiştiği gözlenmiştir (EK Şekil 1). EK Şekil 1 - Şekil 11’de belirtilen 15 öznelik için düzenleme sonuçları verilmiştir. Tüm öznelikler için öznelik düzenleme, soldan sağa yapılmıştır. Bu özneliklerden bazıları yenidir ve Çizelge 3.6’da yer almamaktadır.

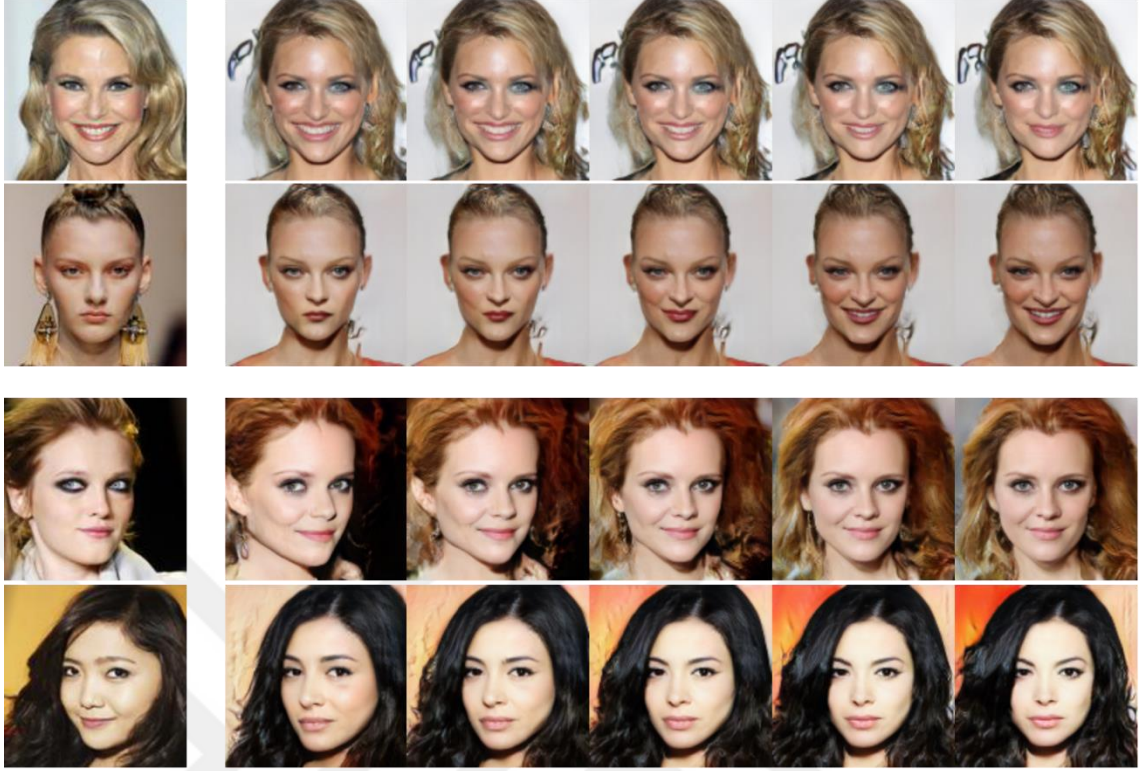
Önerilen yöntem ile imgedeki öznelikler değiştirilebileceği gibi farklı amaçlar için de kullanılabilir. Şekil 3.32’de bu kapsamda bir örnek gösterilmiştir. Diğer özneliklerde olduğu gibi tek bir referans imge çifti kullanılarak imgelerin poz ayarlamasını da yapmak mümkündür. Literatürde belirtilen bu özneliklerden her birini gerçekleştirebilmek için ayrı ayrı model tasarımlarının veya model eğitimlerinin yapıldığı gözlenmektedir. Örneğin; yüz yaşlandırma (face aging) (Palsson vd. 2018, Wang vd. 2019a, Yang ve Lv 2020), poz ayarlama (face frontalization) (Liu ve Chen 2021, Zhang vd. 2020) vb.



Şekil 3.32 Poz ayarlanması

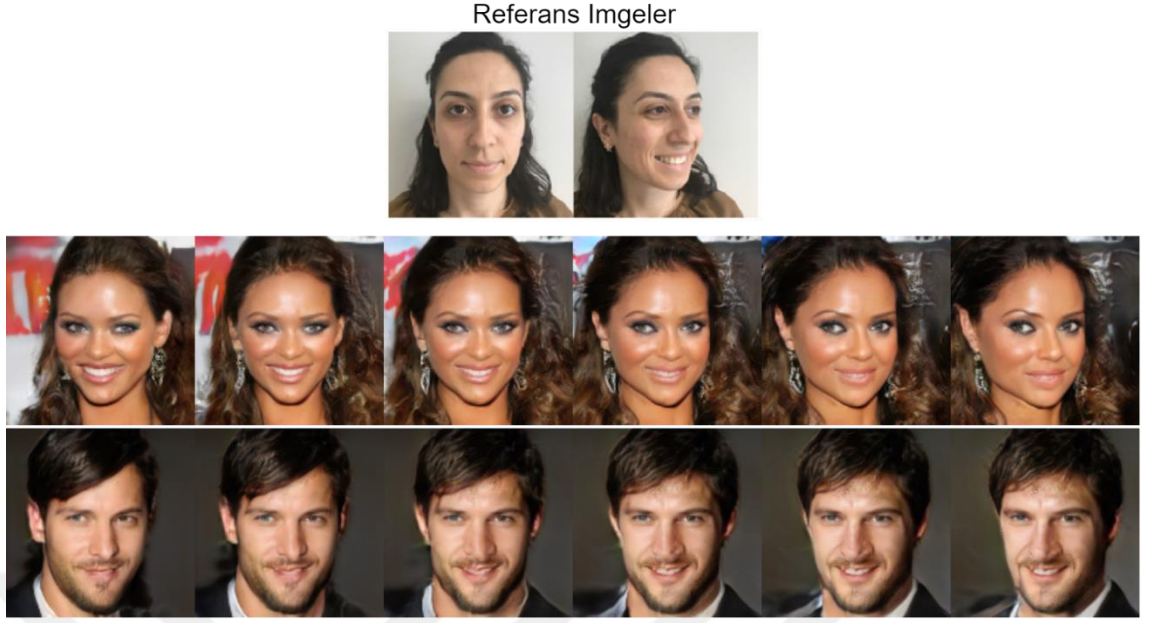
Tez çalışmasında üretilen imgelerin özniteliklerini değiştirmek hedeflense de modelin gerçek imgeler üzerindeki performansını da gözlemlemek adına test işlemleri yapılmıştır. Şekil 3.33'te gerçek imgeler kullanılarak gülme özniteliği ve poz ayarlama için örnekler gösterilmiştir. Şekil 3.33 incelendiğinde gerçek imgelerdeki sonuçların da umut verici olduğu gözlenmektedir. Gerçek imgelerdeki temel problem imgelerin yeniden yapılandırma sonuçlarının istenen seviyede olmamasıdır. Orijinal imge ile yeniden yapılandırılan imge arasındaki farkın, temel olarak ÜÇA eğitimi sırasında yetersiz sayıda örnek, yani 30000 gerçek imge kullanılmasından kaynaklandığı öngörülmektedir. Üretici ağ, eğitim esnasında çok fazla görmediği öznitelikleri oluşturmakta zorlanmaktadır. Örneğin, Şekil 3.33'te ikinci satırdaki örneğin küpeleri, eğitim sırasında Üretici ağ küpelere sahip yeterince numune görmediği için üretilenmemiştir. Bununla birlikte, yeniden yapılandırılan imgelerdeki saç stilleri, jest özellikleri ve pozun başarılı bir şekilde yakalandığı gözlenmektedir. Gerçek imgelerin yeniden yapılandırma doğruluğu, ÜÇA eğitimi sırasında birçok öznitelik varyasyonuna sahip daha fazla örnek kullanılarak artırılabilir. Aslında *CelebA* veri seti 202599 imge içermektedir. Ancak ÜÇA eğitiminde yüksek kaliteli 30000 imgeyi kullanmanın daha iyi sonuçlar ürettiği gözlenmiştir. Üretici ağın başarısı, Kodlayıcı ağın da başarısını etkileyeceğinden, *CelebA* veri setinin alt kümesi kullanılmıştır. Ancak ileride yüksek kalitede ve daha fazla varyasyona sahip imgeler kullanılarak belirtilen problemin üstesinden gelinir.

Gerçek İmgeler



Şekil 3.33 Gerçek imgelerde gülme özniteliđi ve poz ayarlama

Herhangi bir imgenin istenilen bir özniteliđi deđiştirilip diđer öznitelikleri sabit tutulabildiđi gibi istenirse eş zamanlı olarak birden fazla özniteliđi de deđiştirilebilmektedir. Şekil 3.34’te, poz ayarlaması ve gülme özniteliklerinin aynı anda manipüle etme sonuçları gösterilmiştir. Bu örnekte de daha önceden belirtildiđi gibi sadece bir referans imge çifti kullanılmıştır. Bu işlemin gerçekleştirilebilmesi için imgelerden bir tanesinin *nötr*, diđerinin ise çoklu özniteliđe sahip olması gerekmektedir.



Şekil 3.34 Keyfi bir referans imge çifti kullanılarak eş zamanlı olarak gülme özniteliğinin kazandırılması ve poz ayarının yapılması. Üst sıra: referans imge çiftini, ikinci ve üçüncü sıra: referans öznitelik yönünde birden çok örnekleme göstermektedir

Bu bölümde yapılan çalışmalar genel olarak değerlendirilirse ilk olarak ÜÇA'larda istikrar ve çeşitlilik konuları ele alınmış ve özellikle *CelebA* veri seti için yoğun deneysel çalışmalar yapılmıştır (Kısım 3.2). İkinci olarak önceden eğitilmiş bir Üretici ağ kullanılarak karşılıklı eşlemeyi sağlayacak bir Kodlayıcı ağı elde edilmesi için bir yöntem önerilmiştir (Kısım 3.3). Üçüncü olarak önerilen yöntem, bu amaç kapsamında geliştirilen modeller ile nitel ve nicel olarak karşılaştırılmış ve önerilen yöntemin daha başarılı olduğu gösterilmiştir (Kısım 3.4). Dördüncü olarak tek bir referans imge çifti kullanılarak herhangi bir özniteliği temsil eden öznitelik vektör yönünün nasıl elde edildiği gösterilmiştir (Kısım 3.5). Beşinci olarak, öznitelikleri temsil eden gizli vektör yönlerinin analizi yapılmıştır (Kısım 3.6). Bu kısımda, herhangi bir yüz imgesine istenilen bir özniteliğin kazandırılması için, bu imgenin ilgili öznitelik için hesaplanan vektör yönünde hareket ettirilmesi gerektiği belirtilmiştir. Ancak elde edilen sonuçlar incelendiğinde, üretilen bu imgelerin iki imgenin ara değerlemesi sonucu mu yoksa ilgili imgenin öznitelik yönünde hareketi sonucu mu üretildiği açık değildir. Bu kapsamda, önerilen modelin daha önceden görmediği gerçek imgeler kullanarak bu imgeler için elde edilen gizli vektörlerin, ilgili öznitelik eksenine izdüşümü alınmıştır. Sonuçlar incelendiğinde, ilgili özniteliği içeren ve içermeyen imgelerin iki farklı dağılım

oluşturduğu gözlenmiştir. Herhangi bir imge, ilgili öznitelik vektörü yönünde hareket ettirildiğinde, bu imgenin izdüşümünün ilgili özniteliği içermeyen dağılımdan, ilgili özniteliği içeren imgelerin oluşturduğu dağılıma doğru hareket ettiği gözlenmiştir. Bu sonuçlar, önerilen yöntemin doğru çalıştığını göstermektedir. Altıncı olarak farklı imge çiftleri kullanılarak elde edilen gizli vektörlerin tutarlılığı incelenmiştir (Kısım 3.7). Bu kapsamda, nitel ve nicel olmak üzere iki analiz yapılmıştır. Nitel olarak karşılaştırmada, aynı öznitelik için farklı imge çiftleri kullanılmış ve elde edilen öznitelik vektörleri aynı imgeye uygulanmıştır. Sonuçlar incelendiğinde, kullanılan referans imgeler birbirinden oldukça farklı olmalarına rağmen üretilen imgelerdeki değişimlerin çok yakın olduğu gözlenmiştir. Nicel olarak kıyaslamada ise rastgele seçilen referans imgeler için elde edilen öznitelik vektörlerine daha önceden belirtilen gerçek imgelerin izdüşümleri alınmıştır. Sonuçlar incelendiğinde, elde edilen dağılımların benzer olduğu ve sadece ortalama ve standart sapmalarda küçük değişimlerin olduğu gözlenmiştir. Yedinci olarak ise k parametresi için güvenli bir aralık bulunması üzerine bir çalışma yapılmıştır (Kısım 3.8). Yapılan deneysel çalışmalar sonucunda, herhangi bir özniteliği barındırmayan bir imgede $k < 2$ 'nin imgedeki önemli öznitelikleri koruyarak sadece ilgili özniteliği değiştirdiği gözlenmiştir. Fakat girdi imgesi ilgili özniteliği içeriyorsa büyük k değerleri için imgelerde bozulmalar meydana gelmektedir. Çalışmada, imgeleri temsil eden gizli vektörlerin, ortalama öznitelik vektörü üzerindeki izdüşümleri sonucu elde edilen dağılım parametrelerinden yararlanılarak ilgili özniteliği içerip içermediğine bakmaksızın herhangi bir imge için güvenli k aralıklarını belirleme üzerine bir yöntem önerilmiştir. İlave olarak k 'nın sınırları zorlanmış, diğer bir deyişle çok büyük k değerleri için test işlemleri yapılmış ve her bir öznitelik vektörü yönündeki pozitif ve negatif ilerlemelerde tüm imgelerin bir çift imgeye yakınsadığı tespit edilmiştir. Nihai olarak, imge düzenleme çalışmalarında kullanılan ve hemen hemen tümünü kapsayan toplamda 15 öznitelik için elde edilen sonuçlar gösterilmiştir. Sonuçlar incelendiğinde, ilgili öznitelik dışındaki diğer değişimlerin kabul edilebilir seviyede olduğu gözlenmiştir.

4. YÜZ İMGESİ İÇBOYAMA

İmge içboyama yaklaşımlarında amaç, eksik veya maskelenmiş bölgeleri algısal olarak makul içerikle doldurmaktır. Bu, zarar görmüş imgelerin onarılmasını, istenmeyen nesnelerin/parçaların kaldırmasını veya maskelenmiş bölgelerde alternatif içeriklerin sentezlenmesini sağlamaktadır. İçboyama problemi uzun zamandır araştırmacılar tarafından incelenen bir konudur; ancak son zamanlarda özellikle derin öğrenmeye dayalı üretken modellerdeki gelişmelerle, popüler bir konu haline gelmiştir. İmge içboyama modelleri, anlamsal olmayan ve anlamsal bilgi gerektiren yöntemler olmak üzere iki ana gruba ayrılmaktadır. Geleneksel yaklaşımlar olarak da adlandırılan birinci grup, aynı imgenin bilinen bölgelerinden veya bir imge veri tabanından yamalar aramakta ve maskelenmiş bölgeyi yama benzerliği en yüksek çıkan içerikle doldurmaktadır (Simakov vd. 2008, Barnes vd. 2009, Darabi vd. 2012, Huang vd. 2014). Bu yaklaşımlar, arka plan içboyama probleminde (örneğin; gökyüzü, deniz, çimen vb.) doku tutarlı sonuçlar üretmektedir; ancak genellikle yüz, nesne vb. gibi anlamsal bilgi gerektiren karmaşık sahneleri tamamlamada başarısız olmaktadır. Bunun sebebi, maskelenmiş bölgeye karşılık gelen nesne, imgenin diğer bölgelerinde veya veri tabanında mevcut veya uygun olmayabilmektedir.

İkinci grup ise öğrenmeye dayalı imge içboyama yöntemleri (Pathak vd. 2016, Liv d. 2017, Iizuka vd. 2017, Yu vd. 2018, Wang vd. 2018b, Wang vd. 2019b, Nazeri vd. 2019, Zheng vd. 2019) olarak adlandırılmaktadır. Bu gruptaki yöntemler; nesnelerin konumu, şekli, rengi ve dokusu gibi üst düzey anlambilimlerini yakalayabilmekte ve maskelenmiş bölgeyi tutarlı bir şekilde doldurabilmektedir. Yüz imgesi içboyama yaklaşımları bu gruba ait olup anlamsal içerik üretimi gerektirmektedir. Örneğin, bir yüz imgesinin eksik kısmı burun ise diğer yüz parçalarından imge yamaları kullanılarak bu yüze doğal olarak uyan bir burun sentezlemek zor olmaktadır.

Yakın zamanda, derin öğrenmeye dayalı yaklaşımlar, maskelenen bölgeler için görsel olarak gerçekçi içerik doldurma görevinde dikkate değer bir başarı sağlamıştır. Ancak

her bir yaklaşımın, maskeli bölgeyi doldurmada farklı sınırlamaları veya kısıtlamaları olmaktadır:

- Çoğu yöntem, maskelenmiş bölgenin içeriğine, yani atanan piksel değerlerine, çizilen eskizlere vb. duyarlıdır. Bu yöntemler ya maskelenen bölgeye atanan piksel değerini göz ardı etmekte veya eğitim aşamasında olduğu gibi aynı piksel değerlerini kullanmaktadır (Yu vd. 2018, Liu vd. 2018). Bu durum maskelenen bölgeye farklı bir içerik atansa bile çıktının aynı kalmasına neden olmaktadır.
- İmge içboyama alanında önerilen çalışmalar farklı maske türlerini (düzenli, düzensiz ve elle çizilmiş) kullanarak eğitim işlemini gerçekleştirmektedir. Genel olarak bu alanda önerilen yöntemler, eğitim aşamasında kullanılan maske türünde başarılı olurken diğer maske türlerinde istenen çıktıları sağlayamamaktadır (Iizuka vd. 2017, Yu vd. 2018).
- İmge içboyama çalışmaları genel olarak maskelenen bölgeler için tek bir çıktı üretmektedir. Son zamanlarda bazı modeller maskelenmiş bölge için çoklu ve farklı sonuçlar sağlamaktadır; ancak bu modeller genel olarak model eğitimi esnasında eskiz/kenar (Nazeri vd. 2019, Yu vd. 2019, Shao vd. 2020) veya etiket (Chen vd. 2018) bilgisine ihtiyaç duymaktadır.

Literatürde, parametrik (öğrenmeye dayalı) modellerin çoğu Kodlayıcı-Kodçözücü mimarisine dayanmaktadır; bu modeller, maskelenen bölgeleri uçtan uca denetimli bir şekilde doldurmayı öğrenmektedir. Genel olarak bu modellerde, belirli bir imge ve maske için sonuç tek ve sabit olmaktadır. Son zamanlarda, yapısında Üretici ağ barındıran bazı yaklaşımların, maske içeriğini farklılaştırarak farklı ve ümit verici sonuçlar elde ettiği gözlenmektedir (Yeh vd. 2017, Abdal vd. 2020). Önerilen yaklaşımda, maskelenen bölgeyi doldurmak için bir Üretici ağ kullanıldığından, önerilen yaklaşım bu çalışmalara benzemektedir. Ancak önerilen yöntemin bu yaklaşımlardan farkı, yinelemeli olarak gizli vektöre eniyilemek yerine, maskelenen imgeyi Üretici ağ uzayına gömmek (embedding) için Kodlayıcı ağ kullanılmasıdır. Eniyilemeye dayalı yaklaşımlar genel olarak çok sayıda yineleme (birkaç bin) gerektirirken önerilen yöntem ile yalnızca birkaç yinelemeden sonra, ortalama 10 aşamadan sonra, makul bir imge oluşturulmaktadır.

Bu kapsamda, belirli bir imgeyi Üretici gizli uzayına gömmek için Bölüm 3.3'te tanıtılan CRG mimarisi kullanılmıştır. CRG mimarisi, Üretici ağı ters eşlemesini öğrenmek için denetimsiz bir şekilde eğitilmiş, bir Üretici ve bir Kodlayıcı ağı içermektedir. Üretici ağı, *PGGAN* mimarisinin biraz değiştirilmiş bir sürümü olarak tasarlanmış ve *CelebA* veri setindeki 30 bin yüz imgesi kullanılarak eğitilmiştir. İmge içboyama problemi için önerilen çözüm, imgenin bazı bölgeleri maskelenmiş olsa bile, Kodlayıcı ağı imgenin bağlamını kullanarak imgeyi Üretici uzayında gizli bir noktaya gömmesi ile ilişkilidir. Maskelenen bölge içeriği imgenin gömüleceği noktayı doğrudan etkilemektedir; çünkü bağlam tüm imgenin piksel dağılımına bağlı olarak değişmekte ve dolayısıyla üretilen imge buna göre farklılık göstermektedir. İlk döngülerde, maske bölgesi için üretilen içerik bozuk, bulanık vb. olmasına rağmen maskelenmiş bölge için ilk maske içeriği yerine oluşturulan imge içeriğinin kullanılmasıyla, yüz özellikleri yavaş yavaş iyileşmektedir. Bu döngüsel yaklaşım, ilk maske içeriğine bağlı olarak maske bölgelerinde çeşitli alternatif dolgular üretmeye olanak sağlamaktadır; çünkü değişen imge içeriği her seferinde Üretici uzayında farklı bir gömme ile sonuçlanmaktadır. Sonuç olarak, yinelemeli olarak daha iyi bir gizli koda yakınsamak, Üretici alanında daha iyi imge oluşturmayı sağlamakta ve bu da dolayısıyla makul bir imge içeriğine ulaşılmasını sağlamaktadır. Önerilen sezgisel yöntem, döngüler sonucu elde edilen imgeleri skorlayan bir Ayırıştırıcı ağı yardımıyla başarılı bir şekilde çalışmaktadır.

Yüz imgesi içboyama problemi için önerilen yöntem iki aşamadan oluşmaktadır. İlk aşamada, CRG mimarisi ve bir Ayırıştırıcı ağı, maskelenmiş bir girdi imgesi için makul bir imge oluşturmak için kullanılmaktadır. İkinci aşamada, sınır pikselleri etrafındaki yapaylıkları, ten rengi varyasyonları vb. gibi görsel bozuklukları düzeltmek için, Isola vd. (2017) ve Liu vd. (2018) çalışmalarından esinlenerek bir *Unet* modeli eğitilmiştir. İkinci aşamada derin bir ağı kullanılmasının sebebi, manuel işlem sonrası algoritmalarından (*Poisson image blending* (Pérez vd. 2003), *fast marching methods* (Telea 2004) vb.) kaçınmaktır.

Bu bölüm aşağıdaki şekilde düzenlenmiştir. İlk kısımda, yüz imgesi içboyamaya olanak sağlayan modeller ile ilgili literatür taraması yapılmıştır (Kısım 4.1). İkinci kısımda, önerilen iki aşamalı yöntem tanıtılmıştır (Kısım 4.2). Üçüncü kısımda, her döngü sonucu elde edilen imgeleri skorlamayı sağlayan Ayırıştırıcı ağ mimarisi ele alınmıştır (Kısım 4.3). Dördüncü kısımda, maskelenen bölge ile imgenin geri kalan bölgeleri arasında geçiş sürekliliğini sağlamak için farklı çalışmalardan adapte edilen *Unet* modeli açıklanmıştır (Kısım 4.4). Beşinci kısımda, önerilen yöntemin yetenekleri deneysel olarak irdelenmiştir (Kısım 4.5). Altıncı kısımda, önerilen yöntemin yakın zamanda yüz imgesi içboyama problemi için geliştirilen modellerle karşılaştırma sonuçları verilmiştir (Kısım 4.6). Yedinci kısımda, önerilen yöntemin başarısız olduğu durumlar ele alınmıştır (Kısım 4.7). Son kısımda ise önerilen yöntem ile ilgili değerlendirmeler yapılmıştır (Kısım 4.8).

4.1 Literatür

Derin öğrenme modellerindeki ilerlemeler ile birlikte, maskelenen bölgeler için uygun anlamsal içerik üretilabilmektedir. Bu alanda yapılan son çalışmalar incelendiğinde, genel olarak imgelerdeki eksik kısımları doldurmak için daha iyi bir Kodlayıcı-Kodçözücü mimarisi tasarlamaya odaklandıkları gözlenmektedir. Kodlayıcı eksik bir bölgeye sahip imgeyi düşük boyutlu bir gizli uzaya gömmekte, Kodçözücü ise gömülü gizli kodu girdi olarak alıp orijinal imgeyi yeniden yapılandırmaktadır. Otokodlayıcı olarak adlandırılan bu modeller genel olarak piksel tabanlı yeniden yapılandırma kaybı kullanılarak eğitilmektedir; bu da genellikle bulanık çıktıların oluşmasına neden olmaktadır. Bu nedenle yakın zamanda önerilen modeller, yeniden yapılandırma kaybı ile ÜÇA eğitiminde kullanılan çekişmeli kaybı birleştirerek ortak bir amaç fonksiyonu kullanmaktadır. *Context Encoder (CE)* (Pathak vd. 2016) içboyama probleminde çekişmeli kaybı kullanan en eski çalışmalardan biridir. *CE*, *DESA*'lar ile anlamsal önselleri (semantic priors) ve anlamlı gizli gösterimleri (meaningful hidden representations) öğrenerek büyük maskelenmiş bölgeleri doldurmaya olanak sağlamaktadır. *CE* yaklaşımı, maskelenen bölgelerde yeni anlamsal içerik üretebilmektedir; ancak sınırlara yakın bölgelerde tutarsız içerik üretebilmekte ve sabit boyutlu maske kullanmaktadır. Iizuka vd. (2017), *CE* çalışmasını geliştirerek çekişmeli

kayıp hesaplamasında hem global hem de yerel Ayırıştırıcı kullanmış ve görsel kalitenin önemli ölçüde artmasını sağlamıştır. Ayrıca bu çalışmada, tamamen evrışimsel katmanlardan oluşan bir mimarinin kullanılması farklı boyuttaki imgelerin kullanılmasına olanak sağlamıştır. Li vd. (2017), daha uygun ve tutarlı bir sonuç elde etmek için global, yerel ve piksel bazlı yeniden yapılandırma kayıplarına ek olarak önceden eğitilmiş bir yüz sözdizimsel ayırıştırma ağı (face parsing network) kullanmıştır. Bu yaklaşım, hizalanmış yüz imgeleri için iyi sonuçlar üretmektedir. Ancak renk tutarlılığını sağlamak için işlem sonrası yöntemlere (*Poisson image blending* (Pérez vd. 2003) vb.) ihtiyaç duymaktadır. Yu vd. (2018), ESA tabanlı modellerin uzak bağlamsal bölgelerdeki bilgi ile maskelenmiş bölge arasındaki korelasyonları modelleme açısından etkisiz olduğunu ve bunun, maskelenmiş bir bölge ile çevresi arasında çeşitli bozulmalara yol açtığını belirtmiştir. İlgili çalışmada, uzaysal olarak uzak konumlardaki özellik parçaları arasındaki ilişkiyi yakalamak için yeni bir bağlamsal dikkat katmanı (contextual attention) önerilmiştir. Son zamanlarda Wang vd. (2019b), bu mekanizmanın çevreleyen bilgilere (surrounding information) son derece bağımlılık yaptığını ve bir imgenin eksik bölgesindeki tüm bozuklukları kaldıramadığını belirtmiştir. Bu problemin üstesinden gelmek için ilgili çalışmada temsil öğrenme, özellik yeniden yapılandırma ve artık öğrenme (residual learning) olmak üzere üç kısımdan oluşan DSA mimarisi önerilmiştir. İlk kısım, imge içeriğini yakalamayı amaçlamakta ve bunu gizli bir özellik temsiline sıkıştırılmaktadır. İkinci kısım, *Laplacian* piramidi olarak adlandırılmakta ve yüz imgesinin eksik bölgelerini kademeli olarak kabadan inceye doğru onarmaktadır. Üçüncü kısım ise maskelenmiş bölge ve çevresi arasındaki renk farklılıklarını ortadan kaldırmak için kullanılan evrışimsel artık öğrenme mimarisidir. Bu mimari ile maske konumlarından bağımsız olarak gerçekçi ve doğal sonuçların üretildiği belirtilmektedir.

Belirtilen bu modeller dikdörtgensel maskelere çözüm sunmak için tasarlanmıştır. Fakat gerçek hayat senaryolarında, imge içboyama için daha esnek maske tanımlarına ihtiyaç duyulmaktadır. Son zamanlarda serbest biçimli maskeleri desteklemek için çeşitli yöntemler önerilmiştir. Liu vd. (2018), sadece geçerli pikselleri kullanarak evrışimin maskelenip yeniden normalleştirildiği bir kısmi evrışimsel (partial convolution) katmanı önermiştir. Önerilen model düzensiz maskeler için iyi sonuçlar vermektedir; ancak

maskelenen bölgedeki pikselleri göz ardı ettiğinden, eskiz tabanlı içboyamaya izin vermemektedir. Yu vd. (2019), bir dinamik özellik geçitleme mekanizmasını öğrenerek kısmi evrişimi genelleştiren geçitli evrişimsel (gated convolution) katmanı önermiştir. Ayrıca kullanıcı çizimleri doğrultusunda içboyamaya izin vermek için modele bir uzantı eklenmiştir. Nazeri vd. (2019), standart evrişim katmanlarını kullanarak elle çizilmiş maskelere ve eskiz tabanlı içboyamayı izin veren *EdgeConnect* olarak adlandırılan yeni bir yaklaşım önermiştir. Önerilen model, bir kenar üretici ve bir imge tamamlama olmak üzere iki ağdan oluşmakta ve her iki ağdan sonra bir çekişmeli ağ kullanılmaktadır. Kenar üretici ağı imgede maskelenen bölgelerdeki kenarları tespit etmekte, imge tamamlama ise kenar bilgisini koşul olarak alarak imgedeki eksik bölgeleri doldurmaktadır. *EdgeConnect* yöntemi, anlamsal olarak tutarlı içerikler oluşturmaktadır. Fakat maskelenmiş bölge sınırlarında bir kenar yapay dokusu (artifact) oluşabilmektedir. Bu problemi çözmek için, Hong vd. (2019), *Deep Fusion Network (DFNet)* adlı yeni bir perspektif önermiştir. Bu çalışmada, özellikle sınır bölgelerinde daha yumuşak ağırlıklara sahip esnek bir alfa kompozisyon haritası oluşturmak için, *Unet* modelindeki farklı Kodçözücü katmanlarına füzyon blokları yerleştirilmiştir. Bu mimari ile maskelenmiş bölge sınırlarında, daha yumuşak bir geçiş sağlandığı belirtilmiştir.

İmge içboyama problemi için bahsedilen modeller normal, düzensiz veya elle çizilmiş maskeler için makul sonuçlar sağlamaktadır. Fakat bu modeller maskelenen bir imge için sabit ve tek bir sonuç üretebilmektedir. Son zamanlarda, bazı imge içboyama yöntemleri (Chen vd. 2018, Zheng vd. 2019) tek bir maske için çok sayıda ve çeşitli makul çözümler üretmeyi sağlamıştır. Chen vd. (2018), keyfi maskeler için yüksek çözünürlükte yüz imgelerini onarmayı sağlayan tamamen uçtan uca kademeli bir ÜÇA modeli önermiştir. Ayrıca ilgili çalışmada aynı maske bölgesinde birden çok sonuç sağlamak için, kontrol edilebilir özniteliklere (cinsiyet, gülümseme vb.) sahip koşullu bir sürüm önerilmiştir. Zheng vd. (2019), imge tamamlamada çeşitli çözümler üretmek için iki paralel yola sahip olasılıklı bir çerçeve önermiştir. Ayrıca gerçekçiliği iyileştirmek için Kodlayıcı ve Kodçözücü özellikleri arasında uzak ilişkilerden yararlanan kısa + uzun vadeli bir dikkat katmanı sunulmuştur. Bu yaklaşımlar, yüz imgesi içboyamada yararlı olan farklı sonuçların üretilmesine izin vermektedir.

Örneğin, maskelenmiş bölge ağız bölgesi ise hem gülümseyen hem de gülümsemeyen bir yüz imgesi üretilebilmektedir.

Hem imge üretimine hem de bir imgede maskelenmiş bir bölgenin doldurulmasına izin veren çok az sayıda çalışma olduğu gözlenmektedir (Yeh vd. 2017, Abdal vd. 2020). Yeh vd. (2017), anlamsal içboyamayı sınırlandırılmış bir imge üretme problemi olarak ele almayı önermiştir. Bu çalışmada, ÜÇA eğitiminden sonra maskelenmiş imgenin gizli uzayda imgeye en yakın olan kodlaması aranmaktadır. İlgili çalışmada, imgenin maskelenmeyen kısmı kullanılarak orijinal imgeye en yakın imgeyi bulmak için eniyilemeye dayalı bir yöntem önerilmiştir. Abdal vd. (2020) ise başlangıçtaki z gizli uzayı yerine, *StyleGAN* (Karras vd. 2019) çalışmasının W^+ uzayına bir gömme algoritması önermiştir. Önerilen bu yöntem, imge içboyama gibi çoklu imge düzenleme ve işleme uygulamalarına olanak sağlamakta ve eniyilemeye dayalı yaklaşımları ilginç hale getirmektedir. Fakat bu yöntemler, iyi sonuçlar elde etmek için bir imgede birden çok yinleme (örneğin 1500) ve farklı hiperparametre setlerini ayarlamayı gerektirmektedir.

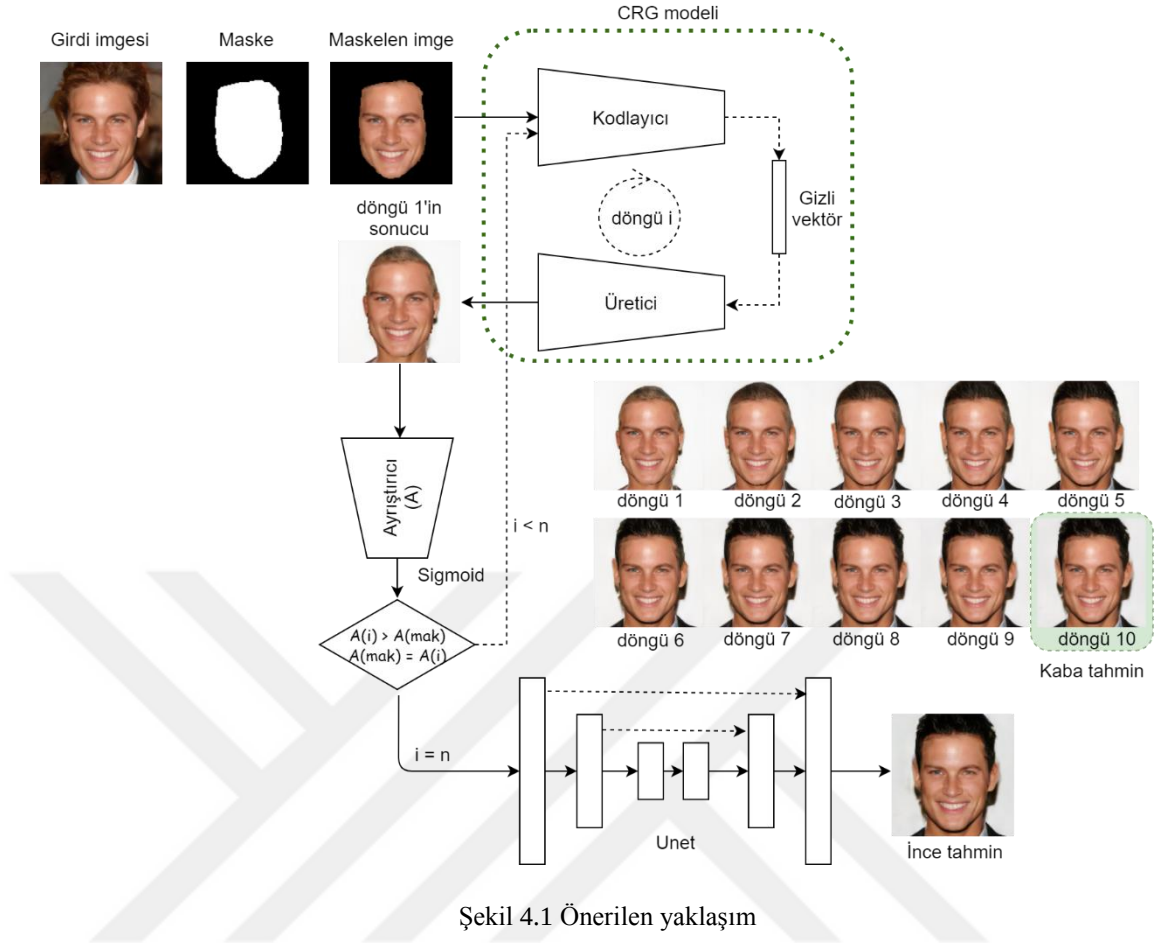
Bu bölümde, Kodlayıcı-Üretici mimarisinde kurulan ve Bölüm 3.3'te tanıtılan CRG mimarisi kullanılarak yüz imgesi içboyama problemi için önerilen çözümden bahsedilmektedir. Üretici ağına sahip diğer çalışmalardan farklı olarak Kodlayıcı ağ kullanılarak maskelenmiş bir bölgeyi anlamsal olarak çok az sayıda yinleme ile dolduran döngüsel bir algoritma önerilmiştir. Önerilen yöntem esnektir; eskiz tabanlı içboyamaya, farklı maske türlerinin kullanılmasına ve maskelenmiş bir bölgeye atanan piksel değerlerine bağlı olarak çok sayıda ve çeşitli sonuçlar üretilmesine olanak sağlamaktadır.

4.2 Önerilen Yöntem

Bu kısımda, yüz imgesi içboyama problemi için önerilen iki aşamalı (kabadan-inceye) yöntem tanıtılmıştır. İlk aşamada, CRG mimarisi kullanılarak Kodlayıcı ve Üretici ağları arasında bir döngü kurulmakta ve her döngü sonucu elde edilen imge Ayırıştırıcı

ağ (Kısım 4.3) ile skorlanmaktadır. Bu döngü, makul bir imge içeriği oluşturulana kadar (genellikle 10'dan daha az sayıda) çalıştırılmakta ve nihai olarak, en yüksek skora sahip imge kaba tahmin olarak seçilmektedir. Genel olarak, ilk döngüde skor düşük, döngü ilerledikçe skor artmakta ve belirli sayıda döngüden sonra skor düşmeye başlamaktadır. Ancak bazen skorlarda dalgalanmalar oluşabilmektedir. Diğer bir deyişle skor düşüşe geçtikten sonra tekrar yükselebilmektedir. Bu nedenle, genel olarak en az 10 döngü çalıştırılmakta ve en yüksek skora sahip imge kaba tahmin olarak seçilmektedir.

Önerilen yöntemin ikinci aşaması işlem sonrası (post-processing) olarak düşünülebilir. Bu kısımda amaç, maskelenmiş bölgenin sınırları içinde veya çevresinde ortaya çıkan bozuklukları iyileştirmek ve ortadan kaldırmaktır. İkinci aşama, oluşturulan kaba imgeyi girdi olarak almakta ve *Unet* tipi derin bir ağ kullanarak iyileştirilmiş sonucu tahmin etmektedir (Kısım 4.4). Bu aşamada bir ağ kullanmamızın amacı, manuel işlem sonrası yöntemlerden (Perez vd. 2003, Telea 2004) kaçınmaktır. Önerilen yaklaşımın genel tasarımı Şekil 4.1'de gösterilmiştir.



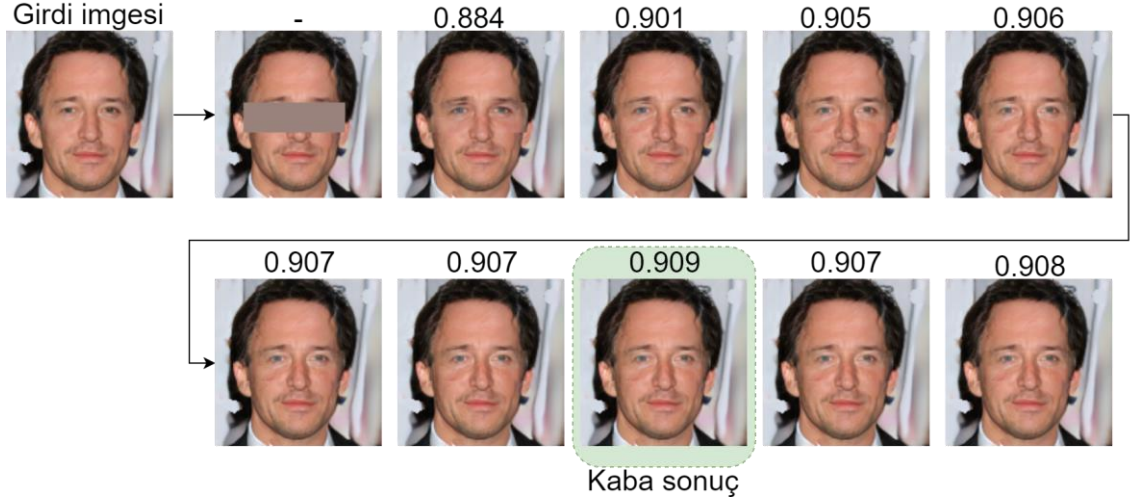
Şekil 4.1 Önerilen yaklaşım

Önerilen yöntemde, bir imgede maskelenen imgeyi doldurmak için CRG mimarisindeki Üretici ağ kullanılmaktadır. Kaba tahmin şu şekilde çalışmaktadır: İlk olarak, maskelenen imge, gizli vektör temsilinin hesaplanması için CRG kodlayıcı ağına girdi olarak verilmektedir. Maskelenen imge, bazı eksik bölgeler içerdiğinden, CRG kodlayıcısı imgenin maskelenmemiş bölgelerdeki bağlamını kullanarak gizli uzayda bir noktaya gömmektedir; maske içeriği de ek bir bağlam sağlamaktadır. Başlangıçta, maskelenen imge için elde edilen gizli kod, orijinal maskelenmemiş imgeye ait olan koddan genel olarak çok farklı olmaktadır. Bu nedenle, birinci döngü sonucu olarak üretilen imge, maskelenen bölgeler için kaba bir içerik barındırmakta ve orijinal imgeden oldukça farklı görünmektedir. Yine de, gerçekçi görünen bir imgeye doğru ilk maskelenen imgeden daha iyi olmaktadır. Bu nedenle, başka bir döngüyü başlatmak için tahmini maske içeriği ilk imgeye kopyalanmaktadır. Teknik olarak bir sonraki döngünün girdisini hazırlamak için Denklem 4.1 kullanılmaktadır.

$$I_0 = M \odot I_{orj}, \quad I_{i+1} = M \odot I_{orj} + (1 - M) \odot I_i \quad (4.1)$$

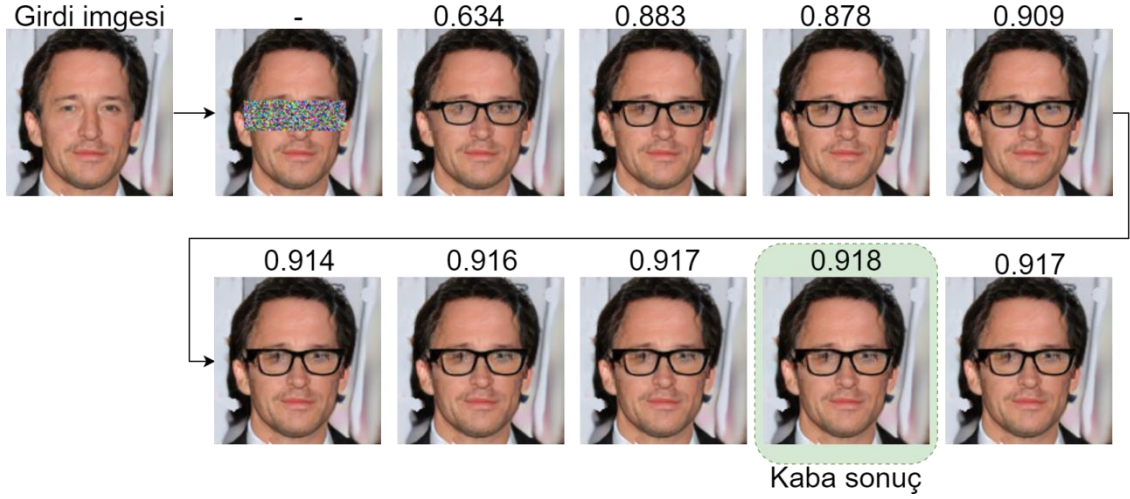
Denklem 4.1’de, I_{orj} orijinal imgeyi, M ikili değerlere sahip maskeyi (maskelenmiş alana 0 diğer alanlara 1 değeri atanmıştır), I_i önceki döngünün çıktısını ve I_{i+1} sonraki döngünün girdisini temsil etmektedir. Şekil 4.1’de, ilk olarak, Üretici ağ kullanılarak rastgele bir girdi imgesi oluşturulmuştur. İlgili örnek için amaç, yüz kısmını değiştirmeden farklı bir saç modeli ve arka plan oluşturmaktır. Bu bağlamda, yüz kısmını koruyacak ve yüzün etrafındaki kısımları doldurmayı sağlayacak şekilde elle bir maske oluşturulmuştur. Denklem 4.1 kullanılarak maske imgeye uygulanmış ve elde edilen imge CRG mimarisine girdi olarak verilmiştir. Her döngünün sonunda orijinal imgede maskelenen bölgeye karşılık gelen piksel değerleri, o döngü sonucu üretilen imgenin uzaysal olarak karşılık gelen pikselleri ile değiştirilmiştir. Bu örnekte, döngüsel süreç 10 kez tekrarlanmış ve en yüksek Ayırıştırıcı ağ skoruna sahip imge kaba tahmin olarak kullanılmıştır. Üreticinin kapasitesine bağlı olarak oluşturulan imgeler, döngü ilerledikçe genel olarak orijinal imgeye (maskelenmeyen bölge için) yakınsamaktadır. Fakat maskelenen bölge pikselleri orijinal görüntüye kopyalandığında, yakınsanan imgedeki poz veya renkteki farklılıklardan dolayı göze çarpan sınır bozuklukları meydana gelebilmektedir. Bu bozuklukları düzeltmek için Isola vd. (2017) ve Liu vd. (2018) çalışmalarından uyarlanan *Unet* modeli sıfırdan eğitilmiştir.

Önerilen yöntemde, maskelenen bölgeye atanan ilk piksel değerlerine bağlı olarak imge gizli uzayda farklı bir noktaya gömülmektedir; bu da farklı alternatif imgelerin elde edilmesine olanak sağlamaktadır. Şekil 4.2’de 128x128 çözünürlüğüne sahip bir yüz imgesinde, gözler kapatılacak şekilde bir maske uygulanmış ve maskelenmemiş imge bölgelerindeki piksel değerlerinin ortalaması atanmıştır. İmgelerin üstünde Ayırıştırıcı ağ skorları verilmiştir. Ayırıştırıcı ağ, 0-1 aralığında skorlar üretmekte ve 1’e yakın değerlerin üretilmesi imgenin kaliteli olduğunu göstermektedir. İmgeler incelendiğinde, ilk üretilen imgelerde geçişlerin daha belirgin olduğu ve ilerleyen döngülerde sonuçların daha kabul edilebilir olduğu gözlenmektedir. Bu örnekte en yüksek skora sahip 8. döngü sonucu kaba tahmin olarak alınmıştır. Şekil 4.2’de nihai olarak gözlüksüz bir yüz imgesi elde edilmiştir.



Şekil 4.2 Maskelenen bölgeye ortalama piksel değeri atandığında elde edilen kaba imge

Şekil 4.3’de ise aynı imgenin maskelenen bölgesine rastgele piksel değeri atanmasıyla elde edilen sonuçlar gösterilmiştir. Maskelenen bölgeye atanan piksel değerine bağlı olarak imge, Üretici uzayında farklı bir noktaya gömülmekte ve bu durum farklı sonuçların elde edilmesine olanak sağlamaktadır. Bilindiği kadarıyla, diğer yüz imgesi içboyama çalışmaları bu özelliği sağlayamamaktadır. Bunun sebebi, diğer çalışmaların maskelenen bölgeye piksel değeri ataması yaparken hem eğitim hem de test aşamalarında aynı yöntemi kullanmaları veya maskelenmiş bölgeye atanan piksel değerlerini göz ardı etmelerinden kaynaklanmaktadır. Dikkat edilirse, maskelen bölgeye ortalama piksel değeri atandığında, gözlüksüz bir yüz imgesi elde edilmişken rastgele piksel değeri atandığında gözlüklü bir yüz imgesi elde edilmiştir. Bu her zaman sabit bir durum değildir. Bu örnek için rastgele piksel değeri atanması sonucu gözlüklü bir yüz imgesi elde edilmişken farklı bir imgede farklı bir içerik oluşabilmektedir. Bu maskelenen imgenin Üretici uzayında gömüldüğü nokta ile doğrudan ilişkilidir.



Şekil 4.3 Maskelenen bölgeye rastgele piksel değeri atandığında elde edilen kaba imge

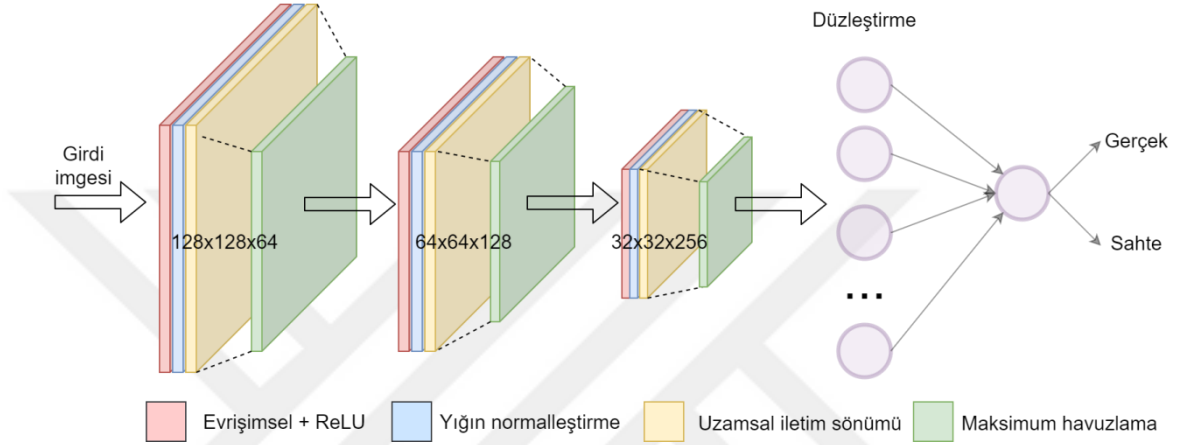
Önerilen yöntemde, CRG mimarisinin eğitim aşamasında, herhangi bir maske kullanılmadığı için maskelerin türü ve içeriği konusunda herhangi bir kısıtlama olmamaktadır. Şekil 4.4'te, önerilen yöntem rastgele bir düzensiz maske kullanılarak 5 döngü için test edilmiştir. Daha önceden belirtildiği gibi her döngünün sonunda, orijinal imgede maskelenen bölgeye karşılık gelen piksel değerleri ile üretilen imgedeki piksel değerleri değiştirilmiştir (Denklem 4.1). Bu örnek 5 CRG döngüsü kurularak gerçekleştirilmiş ve 5. döngü sonucunda Üretici ağı ürettiği imge sağ-üst köşede gösterilmiştir. 5. döngü sonucu elde edilen imgenin orijinal görüntüye yakınsadığı ve elde edilen kaba tahminin döngüler ilerledikçe iyileştiği gözlenmektedir.



Şekil 4.4 Rasgele düzensiz bir maske kullanılarak elde edilen kaba imge

4.3 Ayırıştırıcı Ağ

Bu kısımda, önerilen Ayırıştırıcı ağ mimarisi tanıtılmıştır. Yüz imgesi içboyama problemi için önerilen yöntemde, hangi imgenin görsel olarak daha gerçekçi olduğunu belirlemek önemlidir. Bu amaçla, sigmoid bir birim ile biten çok katmanlı evrişimsel bir sinir ağı oluşturulmuştur. Önerilen Ayırıştırıcı ağ yapısı Şekil 4.5’de gösterilmiştir.



Şekil 4.5 Ayırıştırıcı ağ mimarisi

Genel olarak, imge içboyama çalışmalarında maskelenen bölge ile imgenin geri kalanı arasında parlaklık, bulanıklık ve karışıklık farklılıkları oluşmaktadır. Bu kısımda amaç, bu farklılıkları dikkate alarak skor üretecek bir model sağlamaktır. Bu amaç doğrultusunda, öncelikle düzenli ve düzensiz maskeler üretilmiş ve daha önceden belirtilen görsel bozukluklar belli bir miktar maskelenmiş bölgeye uygulanarak bir veri seti oluşturulmuştur. Bu kapsamda, yüz imgelerine bulanıklık [1.0 – 2.5], parlaklık [0.4 – 0.8] ve karışıklık [0.4 – 0.8] için ilgili aralıklarda rastgele değerler üretilerek yapay doku uygulanmıştır. Bu yapay doku eklenmesinde, bulanıklık için yüksek değer, diğerleri için ise düşük değer bozukluk oranını arttırmaktadır. Daha sonra gerçek imgeler 1, diğerleri ise 0 olarak etiketlenmiştir. Etiketleme işlemi otomatik olarak basit bir betik ile gerçekleştirilmiştir. Eğitim aşamasında, Denklem 4.2’de verilen ÜÇA’ya dayalı bir kayıp fonksiyonu kullanılmıştır.

$$mak_A \mathbb{E}_{x \sim P_{veri}(x)} \log(A(x)) + \mathbb{E}_{y \sim P_{GörBoz}(y)} \log(1 - A(y)) \quad (4.2)$$

Denklem 4.2’de x ve y sırasıyla gerçek imgeyi ve görsel bozukluk içeren imgeyi temsil etmektedir. Burada amaç, gerçek veri dağılımını, yani P_{veri} ’yi görsel bozukluk içeren veri dağılımından, yani $P_{GörBoz}$ ’dan ayırt etmeyi sağlayacak şekilde Ayırıştırıcı ağını eniyilemektir. Ancak Ayırıştırıcı ağına verilen görev basit olduğundan, ağ hızla yakınsamakta ve girdi imgeleri için yalnızca 0 ve 1 skorları üretmektedir; bu durum önerilen yöntem için yararlı değildir. Önerilen yöntemde, imgelerin görsel kalitesini yansıtan ve ara skorlar üretebilen yapıya ihtiyaç duyulmaktadır. Bu problemin üstesinden gelmek için eğitim işleminde iki farklılık yapılmıştır. İlk olarak, gerçek imgeler 1 yerine 0.9 ve yapay doku içeren imgeler 0 yerine 0.1 olarak etiketlenmiştir. İkinci olarak, düşük oranda yapay doku uygulanan bazı imgeler, (bulanıklık [0.3 – 0.8], parlaklık [0.85 – 0.95] ve karışıklık [0.85 – 0.95]) gerçek olarak etiketlenmiştir. Oluşturulan veri seti toplamda 60 bin imge içermektedir; bu imgelerden 22 bin tanesi bozulmamış imgeleri ve 8 bin tanesi düşük oranda yapay doku uyguladığımız imgeleri içermektedir ve gerçek olarak etiketlenmiştir, geri kalan 30 bin tanesine yüksek oranda yapay doku uygulanmış ve sahte olarak etiketlenmiştir.

Ayırıştırıcı ağ ve eğitim detayları: Ayırıştırıcı ağ Şekil 4.5’de gösterildiği gibi çok katmanlı bir evrişimsel sinir ağı olarak tasarlanmıştır. Ayırıştırıcı, 3 blok ve tam bağlantılı bir katmandan oluşmakta ve toplamda yaklaşık 437 bin eğitilebilir parametre içermektedir. İlk blokta 64 filtre kullanılmış ve ardışık bloklarda filtre sayısı ikiye katlanmıştır. Evrişimsel katmanlarda 3x3 filtre boyutu kullanılmıştır. Aşırı öğrenmeyi engellemek için, %50 uzamsal bırakma (spatial dropout) katmanı uygulanmıştır. Ayırıştırıcı ağı, sıfırdan 300 epok boyunca *RMSPProp* eniyileme algoritması kullanılarak eğitilmiştir. Eniyileme algoritmasında, öğrenme oranı $1e-4$, ρ 0.9 ve ϵ $1e-08$ olacak şekilde ayarlanmıştır. 15 epok boyunca doğrulama veri setinde iyileşme olmaması durumunda öğrenme oranı yarıya indirgenmiş ve doğrulama veri setindeki başarıya göre en iyi model kaydedilmiştir.

Nihai olarak, eğitim işlemi sonrası elde edilen Ayırıştırıcı ağ imgenin görsel kalitesine bağlı olarak ara skorlar üretmektedir.

4.4 Görsel Bozuklukların Ortadan Kaldırılması

Bu kısımda amaç, maskelenen bölge ile çevresindeki bağlam arasında yapısal sürekliliği sağlamaktır. Bu amaçla, daha önceden belirtilen çalışmalardan yararlanılarak bir *Unet* modeli uyarlanmış ve sıfırdan eğitilmiştir. Eğitim aşamasında, *Unet* modeline girdi olarak maske ve kaba sonuç verilmiş ve çıktı olarak orijinal imgenin elde edilmesi sağlanmıştır. Model piksel başına yeniden yapılandırma kaybı ve hem I_{Unet} hem de I_{komp} için hesaplanan stil kaybından oluşan Denklem 4.4'teki birleşik kayıp fonksiyonu enküçültülecek şekilde eğitilmiştir. CRG sonucu I_{CRG} , ikili maske M (maskelenen bölge için 0) ve *Unet* tahmini I_{Unet} verildiğinde, I_{komp} Denklem 4.3'de verildiği gibi tanımlanmaktadır.

$$I_{komp} = M \odot I_{CRG} + (1 - M) \odot I_{Unet} \quad (4.3)$$

$$L_{toplaml} = L_{yapı} + 150(L_{stil_{Unet}} + L_{stil_{komp}}) \quad (4.4)$$

Denklem 4.5 birleşik kayıp fonksiyonunda kullanılan piksel başına yeniden yapılandırma kaybını göstermektedir; burada I_{orj} orijinal imgeyi, N_{orj} ise I_{orj} 'deki piksel sayısını temsil etmektedir.

$$L_{yapı} = \frac{1}{N_{orj}} \|I_{Unet} - I_{orj}\|_1 \quad (4.5)$$

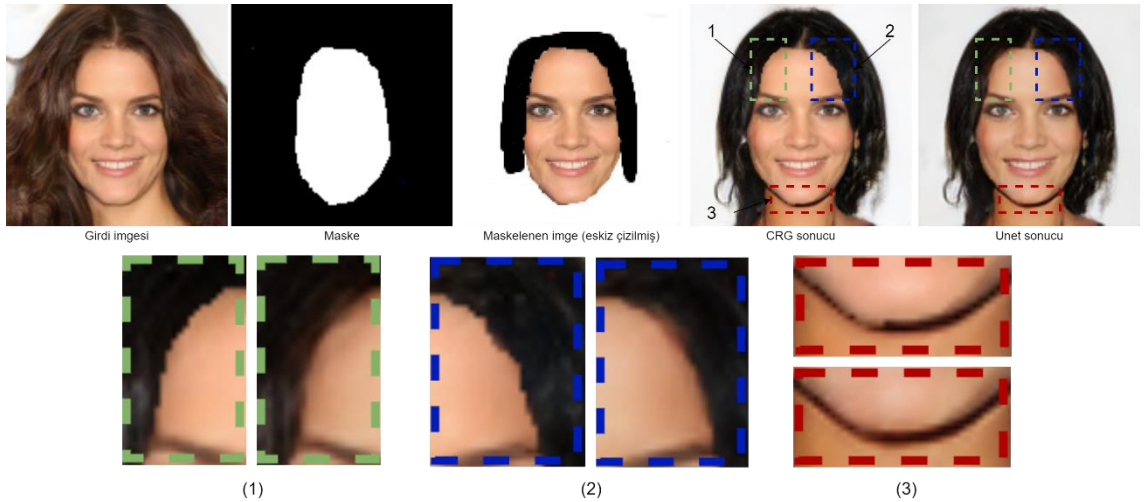
Denklem 4.6 ve 4.7'de sırasıyla I_{Unet} ve I_{komp} için kullandığımız stil kayıp fonksiyonları gösterilmiştir. Stil kaybı hesaplarırken *ImageNet* veri setinde (Russakovsky vd. 2015) önceden eğitilmiş *VGG16* ağıın *pool1*, *pool2* ve *pool3* katmanları kullanılmıştır.

$$L_{stil_{Unet}} = \sum_{j=0}^{J-1} \frac{1}{c_j c_j} \left\| K_j \left(G_{\phi_j^{I_{Unet}}} - G_{\phi_j^{I_{orj}}} \right) \right\|_1 \quad (4.6)$$

$$L_{stil_{komp}} = \sum_{j=0}^{J-1} \frac{1}{c_j c_j} \left\| K_j \left(G_{\phi_j^{I_{komp}}} - G_{\phi_j^{I_{orj}}} \right) \right\|_1 \quad (4.7)$$

Denklem 4.6 ve 4.7’de, ϕ_j^{I*} , I_* girdisi için j. katmanın (yani *pool1*, *pool2* ve *pool3*) aktivasyon haritasına karşılık gelmektedir. ϕ_j^{I*} , $C_j \times H_j \times W_j$ boyutlarına sahiptir; burada C, H ve W sırasıyla ilgili katmanın özellik haritası sayısı, yüksekliği ve genişliğine karşılık gelmektedir. $G_{\phi_j^{I*}}$ ise $C_j \times C_j$ gram matrisidir, yani ϕ_j^{I*} aktivasyon haritasının kovaryansları ve K_j j. aktivasyonu için normalizasyon katsayısıdır, örneğin j. aktivasyon katmanı için $1/C_j H_j W_j$ ’dir.

Şekil 4.6’da, *Unet* modeli ile sınır bozukluklarının (boundary artifacts) kaldırılması ile ilgili örnek verilmiştir. Bu örnekte, ilk önce yüz kısmını değiştirmeden tutmak için elle çizilmiş bir maske oluşturulmuş ve oluşturulan maske imgeye uygulanmıştır. Daha sonra saç stilini daha koyu hale getirmek için maskelenmiş imgeye kabataslak siyah saç çizilmiş ve bu imge CRG mimarisine girdi olarak verilmiştir. CRG mimarisi kullanılarak içboyama sonucu elde edildikten sonra karşılaştırma için hem CRG hem de *Unet* sonucundan aynı kısma denk gelen üç bölge seçilmiştir. Sınır bölgelerindeki işlem sonrası iyileşme, alt sıradaki imge çiftlerinde açıkça gözlenmektedir; işlem sonrası maskelenen alan ile çevresi arasında daha iyi yapısal tutarlılık ve daha pürüzsüz renk geçişi gözlenmiştir.



Şekil 4.6 *Unet* modeli kullanılarak maskelenen bölge ile diğer bölgeler arasındaki geçiş sürekliliğinin sağlanması. Yeşil, mavi ve kırmızı kesik çizgilerle gösterilen bölgelere %300 yakınlaştırma uygulanmıştır. (1) - sol, (2) - sol ve (3) - yukarı imgeler CRG çıktıları, (1) - sağ, (2) - sağ, (3) - alt imgeler, *Unet* modeli kullanılarak işlem sonrası elde edilen imgelerdir

Unet mimarisi ve eğitim ayrıntıları: Unet mimarisi Isola vd. (2017) ve Liu vd. (2018) çalışmalarından uyarlanmıştır. Modelde, tüm kısmi evrişimsel katmanlar standart evrişimsel katmanlarla değiştirilmiştir. Uyarlanan modelde, maskeler sadece stil kayıplarını hesaplamak için kullanılmıştır; bu nedenle maske güncelleme adımları takip edilmemiştir. Çizelge 4.1’de, mimarideki diğer ayrıntılar verilmiştir. Her evrişimsel katmandan sonra bir yığın normalleştirme katmanı kullanılmıştır. Liu vd. (2018) çalışmasına benzer şekilde, aşağı örnekleme ve üst örnekleme 2 faktörüyle gerçekleştirilmiştir. Kodlayıcı ağıdaki tüm evrişimsel katmanlarda *ReLU* aktivasyon fonksiyonu, Kodçözücü’de ise 0.2 eğimli *LReLU* aktivasyon fonksiyonu kullanılmıştır.

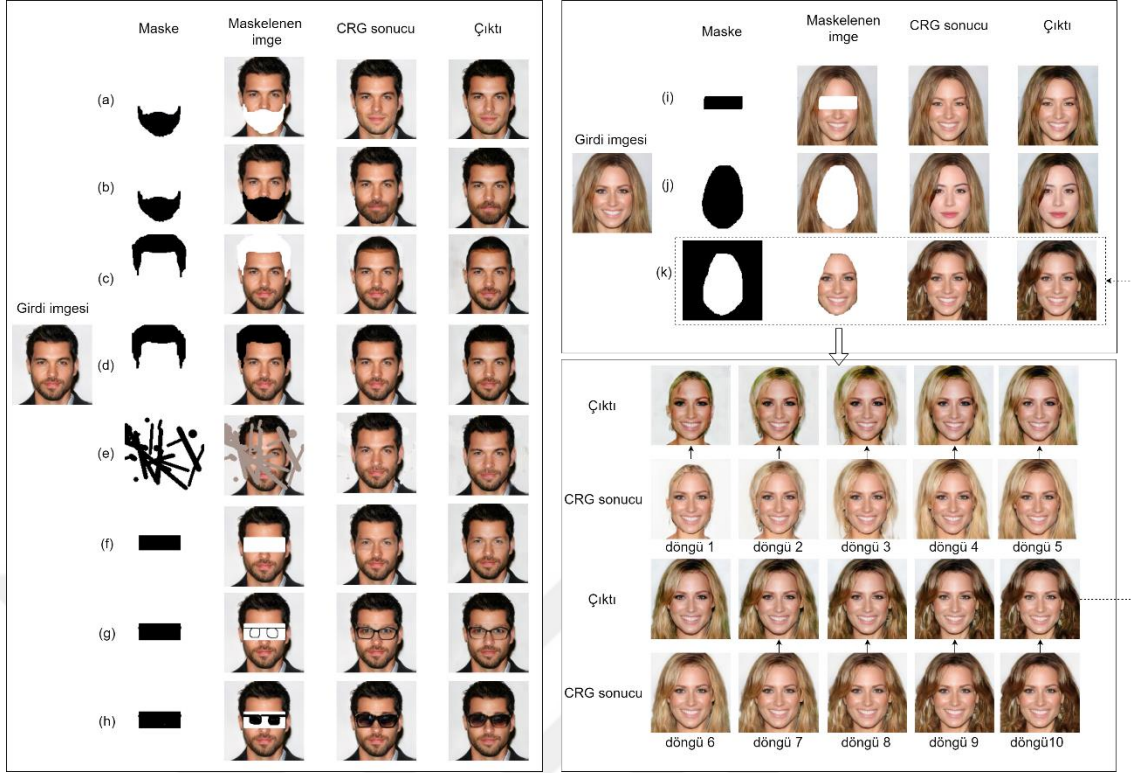
Çizelge 4.1 Unet mimarisi

Kodlayıcı		
Katman adı	Filtre boyutu	Filtre sayısı
<i>Conv1</i>	7x7	64
<i>Conv2</i>	5x5	128
<i>Conv3</i>	5x5	256
<i>Conv4</i>	3x3	512
<i>Conv5</i>	3x3	512
<i>Conv6</i>	3x3	512
Kodçözücü		
<i>Upsample1</i>	-	512
<i>Concat (Conv6 ile)</i>	-	512+512
<i>Conv7</i>	3x3	512
<i>Upsample2</i>	-	512
<i>Concat (Conv5 ile)</i>	-	512+512
<i>Conv8</i>	3x3	512
<i>Upsample3</i>	-	512
<i>Concat (Conv4 ile)</i>	-	512+512
<i>Conv9</i>	3x3	512
<i>Upsample4</i>	-	512
<i>Concat (Conv3 ile)</i>	-	512+256
<i>Conv10</i>	3x3	256
<i>Upsample5</i>	-	256
<i>Concat (Conv2 ile)</i>	-	256+128
<i>Conv11</i>	3x3	128
<i>Upsample6</i>	-	128
<i>Concat (Conv1 ile)</i>	-	128+64
<i>Conv12</i>	3x3	64
<i>Upsample7</i>	-	64
<i>Concat (Girdi ile)</i>	-	64+3
<i>Conv13</i>	3x3	3

Unet modeli, yığın boyutu 16 ve küçük bir öğrenme oranı, yani 0.0002'e ayarlanan Adam eniyileme algoritması kullanılarak 100 epok boyunca sıfırdan eğitilmiştir. Aşırı öğrenme veya konumsal önyargıdan (positional bias) kaçınmak için, veri arttırma (data augmentation) yaklaşımı kullanılmıştır. Bu kapsamda, imgeler sola, sağa, yukarı veya aşağı kaydırılmış ve yatay-dikey çevirme işlemleri uygulanmıştır.

4.5 Önerilen Yöntemin Yetenekleri

Bu kısımda, önerilen yaklaşımın yetenekleri deneysel olarak incelenmiştir. Öncelikle önerilen yöntem, maske içeriğine bağlı olarak maske bölgesinde çeşitli alternatif dolgular üretmeye olanak sağlamaktadır. Şekil 4.7'de yüz imgesinde sakal bölgesi maskelenerek piksel değeri (a) beyaz ve (b) siyah ile doldurulmuştur. Önerilen yöntem ile maske içeriğine bağlı olarak maskelenen bölgede farklı ve ilişkili içerik üretildiği gözlenmiştir; maske içeriği beyaz olduğunda sakalsız bir imge ve siyah olduğunda sakallı bir imge elde edilmiştir. İkinci olarak önerilen yöntem farklı maske türlerinin, yani elle çizilmiş (örneğin; (a), (b) vb.), düzensiz (örneğin, (e)) ve düzenli maskelerin (örneğin; (f), (g) vb.) kullanımına olanak sağlamaktadır. Üçüncü olarak imgenin maskelenen bölgesinde eskiz çizilmesi durumunda, Üretici uzayında ilişkili bir noktaya düşeceğinden, önerilen yöntem eskiz tabanlı içboyama yapmaya olanak sağlamaktadır. Örneğin (g)'de göz bölgesinde kabaca gözlüğe benzeyen ince eskizle bir maske oluşturulmuş ve sonuç olarak gözlüklü bir yüz imgesi elde edilmiştir. Benzer şekilde (h)'de kalın çizgilerle bir maske (kabaca güneş gözlüklerine benzer) oluşturulmuş ve sonuç olarak güneş gözlüklü bir yüz imgesi elde edilmiştir. Son olarak önerilen yöntem çoklu ve çeşitli sonuçlara izin vermektedir. Bunu göstermek için, sağ alt blokta (k) örneği için her döngünün sonucu sunulmuştur. Dikkat edilirse, birkaç döngü sonucunun yararlı olduğu gözlenmektedir; birden çok sonuç için Ayırıştırıcı ağ devre dışı bırakılmalı ve her döngü sonucu *Unet* modeline girdi olarak verilmelidir.



Şekil 4.7 Soldan-Sağa girdi imgesi, farklı maske türleri (düzenli, düzensiz veya elle çizilmiş), maskelenen imge (maskelenmiş bölgede farklı piksel değeri veya eskiz kullanılmıştır), kaba sonuç (n. CRG döngü sonucu), ve ince sonuç verilmiştir. Sağ alttaki blokta ise (k) örneği için tüm döngü sonuçları verilmiştir

ÜÇA’lardaki ilerlemelere rağmen eğitim işlemi zor olmakta ve genel olarak iki ana problem ortaya çıkmaktadır (Kısım 2.3). İlk problem, istikrar sorunudur; iki ağ arasındaki rekabet, yani Üretici ve Ayrıştırıcı, biri diğerine hâkim olduğu için istikrarsızlık oluşturmaktadır. İkinci problem, mod çökme sorunudur; yani Üretici ağı sınırlı sayıda örnek üretebilmektedir. Burada önerilen yaklaşım ile üretilen imgelerde çarpık (distorted) olan veya farklı sonuçlar üretmek istediğimiz parçalar maskelenerek bu parçaların yenilenmesine olanak sağlanmıştır. Şekil 4.8’de bu amaç doğrultusunda bir örnek gösterilmiştir. İlk olarak CRG Üreticisi kullanılarak rastgele bir girdi imgesi üretilmiştir. Bu imge incelendiğinde yüz kısmının makul olduğu; ancak diğer kısımlarının bozuk olduğu gözlenmektedir. Daha sonra yüz kısmını koruyacak şekilde elle çizilmiş bir maske oluşturulmuştur. Döngü sayısı ilerledikçe maskelenen kısmın iyileştiği ve yüz için daha uygun hale geldiği gözlenmiştir.



Şekil 4.8 İstenmeyen bölgelerin düzeltilmesi

Üretilen imgelerin yanı sıra önerilen yöntem, gerçek imgelerin boyanması için de yararlı ve etkili olmaktadır. Şekil 4.9'da görüldüğü gibi farklı maske türleri kullanılarak farklı boyutlardaki ve farklı yerleşimlerdeki maskeler için gerçekçi sonuçların elde edildiği gözlenmektedir. Şekil 4.9'da, üretilen imgelerde olduğu gibi farklı maske türlerinin kullanılabildiği gösterilmiştir. (1) ve (2) örneklerinde düzensiz maske, (3) ve (4)'de elle çizilmiş maske ve (5) - (11) örneklerinde ise düzenli maskeler kullanılmıştır. Sonuçlar analiz edildiğinde, önerilen yöntemin orijinal içerikten çok farklı sonuçlar üretebildiği gözlenmektedir. Örneğin (3) numaralı imge incelendiğinde, nihai olarak oluşturulan içeriğin orijinal imgeden çok farklı olduğu gözlenmektedir. Ayrıca ara döngüler sonucu elde edilen içeriklerin de kullanışlı olduğu gözlenmektedir. İlgili örnek için döngü 2, döngü 3 ve döngü 4 imgeleri de *Unet* modelinden geçirilerek kullanılabilir. Literatür kısmında, bazı yüz imgesi içboyama çalışmalarının sadece hizalanmış imgelerde başarılı sonuçlar ürettiği belirtilmiştir. Önerilen yöntem için (6)'da farklı poza sahip ve (7)'de hizalanmış imgeye aynı maske uygulanarak test işlemi yapılmıştır. (6) numaralı imge farklı poza sahip olmasına rağmen burun objesinin makul olarak yerleştirildiği gözlenmiştir. (8) numaralı orijinal imge gülme özneliğini taşımasına rağmen ağız bölgesi maskelendiğinde, önerilen yöntemin yüz ifadesini yakaladığı ve gülme özneliğine sahip bir imge oluşturduğu gözlenmektedir. (9) ve (10) numaralı örnekler ise döngüsel olarak sonuçların nasıl iyileştiğini görmek açısından önemlidir. Örneğin (10) numaralı örnek incelendiğinde, yüz imgesi ile hiç uyumlu olmamasına rağmen ilgili bölgede olması gereken bir göz objesinin oluştuğu ve üretilen göz objesinin döngüler ilerledikçe yüz ile daha uyumlu hale geldiği gözlenmektedir. Bu sonuçlar, önerilen yöntemin gerçek imgeler kullanılarak da iyi sonuçlar üretebildiğini göstermektedir.



Şekil 4.9 Gerçek imgeler kullanılarak elde edilen sonuçlar

4.6 Modellerin Karşılaştırılması

Literatürde, yüz imgesi içboyama modelleri farklı maske türleri ve farklı çözünürlüklerde eğitilmiştir. Önerilen yöntemde, maskelere bağlı olarak herhangi bir denetime ihtiyaç duymayan CRG mimarisi kullanılmıştır. Bu nedenle önerilen yöntem farklı maske türlerinin kullanımına izin vermektedir. Şekil 4.10’da, önerilen yöntem yüz

imgeleri için içboyamaya izin veren ve yakın zamanda önerilen modellerle, yani *PIC*⁷ (Zheng vd. 2019), *GLCIC*⁸ (Iizuka vd. 2017), *GIICA*⁹ (Yu vd. 2018), *GMCNN*¹⁰ (Wang vd. 2018b), *EdgeConnect*¹¹ (Nazeri vd. 2019), *DFNet*¹² (Hong vd. 2019), *Pconv* (Liu vd. 2018) karşılaştırılmıştır. Bunun için farklı maske türleri (elle çizilmiş (1-5), düzenli (6-7) ve düzensiz (8-9)) kullanılmıştır. Bu modellerden bazıları modellerini düzenli, örneğin *GIICA* ve *GLCIC*, bazıları düzensiz, örneğin *EdgeConnect*, *DFNet* ve *Pconv* veya hem düzenli hem de düzensiz maskeler, örneğin *PIC* ve *GMCNN* kullanarak eğitmiştir. Modellerle karşılaştırma yapılırken ilgili modellerin önceden eğitilmiş resmi modelleri kullanarak imge içboyama yapılmıştır; resmi modelleri mevcut olmaması durumunda model yazarlarına maske ve girdiler gönderilerek sonuçları alınmış veya orijinal makalelerde olduğu gibi yapılandırılmış önceden eğitilmiş bir model kullanılmıştır. Farklı çözünürlüklerde eğitilmiş modeller için orijinal imge ve maske yeniden boyutlandırılmıştır. *PIC* modelinde, maskelenen bölgenin, imgenin merkezinde olması veya rastgele yerleştirilmesi açısından yayınlanmış iki önceden eğitilmiş model mevcuttur. Benzer şekilde *GMCNN* modeli, düzenli ve düzensiz maskeler için önceden eğitilmiş iki farklı modele sahiptir. Bu modellerde kullanılan maskeye bağlı olarak ilgili önceden eğitilmiş modelleri kullanılmıştır. Ayrıca *PIC* modeli tek bir maske için birden fazla sonuç üretebilmektedir; bu nedenle, Ayrıştırıcı puanına göre bu modelin ürettiği en iyi sonuç seçilmiştir. Modellerin karşılaştırması Şekil 4.10'da gösterilmiştir. Sonuçlardan da görüldüğü gibi önerilen yöntem, farklı maske türlerinde iyi sonuçlar vermekte ve elle çizilmiş (1-5) maske tipinde diğer modellere üstünlük kurabilmektedir. Önerilen yöntem dışındaki hemen hemen diğer tüm modellerin, özellikle 3. ve 4. imgelerde başarısız olduğu görülmüştür. Bunun nedeni, maskeli alanın çok büyük olması ve bu modellerin eğitim aşamasında bu maskeleri kullanmamasıdır. Önerilen yöntemde de eğitim aşamasında herhangi bir maske kullanılmamıştır. Fakat Üretici ağ sayesinde önerilen yöntem büyük maskelerin doldurulmasına olanak sağlamıştır.

⁷ <https://github.com/lyndonzheng/Pluralistic-Inpainting>

⁸ <https://github.com/otenim/GLCIC-PyTorch>

⁹ https://github.com/JiahuiYu/generative_inpainting/tree/v1.0.0

¹⁰ https://github.com/shepnerd/inpainting_gmcnn

¹¹ <https://github.com/knazeri/edge-connect>

¹² <https://github.com/hughplay/DFNet>



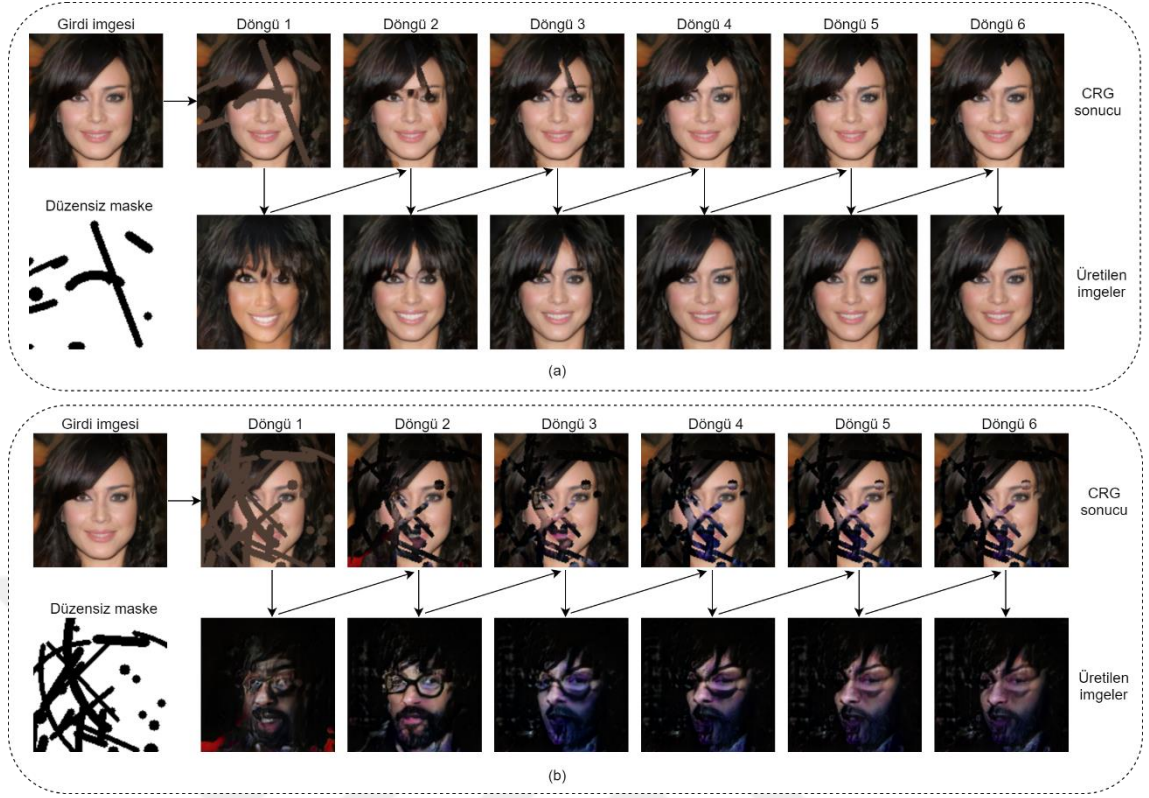
Şekil 4.10 Model sonuçlarının karşılaştırılması

Yüz imgesi içboyama modellerini nicel olarak karşılaştırmak için, genel olarak $L1$ kaybı, $PSNR$, $SSIM$ (Wang vd. 2014) ve IS metriklerini kullanılmaktadır. Fakat birçok çalışma Yeh vd. (2017), Yu vd. (2018) ve Wei vd. (2019) orijinal imge içeriğinden farklı birçok olası çözüm nedeniyle iyi bir değerlendirme ölçütü olmadığını belirtmiştir. Bu metrikler, üretilen imgelerin orijinal imgeye benzerliğine dayalı olarak skorlar üretmektedir. Ancak Şekil 4.8’de gösterildiği gibi ilgili problem alanı için bu her zaman istenen bir durum olmamaktadır.

4.7 Modelin Başarısız Olduğu Durumlar

Önerilen yöntemin başarısı, Üretici uzayının yeterliliği (sufficiency) ile doğrudan ilişkilidir; yeterli bir Üretici modelinde, öncül z dağılımından örneklenen hemen hemen her nokta için (yani üretilen imgeler), makul anlamsal içerik oluşturulabilmektedir. Bu bağlamda maskelenen imgelerin, Üretici gizli uzayında yeterli değişirliğe (variance) sahip bir bölgeye gömülmesi önemli olmaktadır. Ancak gizli uzaydaki bazı noktalarda, eğitilmiş örneklerin *manifold*'u yeterli yoğunluğa sahip olmamaktadır. Bu bölgeler ölü bölgeler (dead zones) olarak adlandırılmaktadır (White 2016). Bu tür bölgelerden yapılan örneklemelerde, gizli uzaydaki bu noktanın etrafında değişirlik olmaması nedeniyle bir sonraki döngü için, maske içeriği düzgün bir şekilde doldurulamamaktadır.

Bir imgeye uygulanan kusur seviyesi, yetersiz anlamsal içeriğe neden olmakta ve bu ölü bölgelerden örnekleme olasılığını güçlendirmektedir. Şekil 4.11'de bu problem ile ilgili bir örnek gösterilmiştir; (a)'da, maske uygulanan girdi imgesinin anlamsal içeriğinde oluşturulan kusur, (b)'de tasvir edilen kadar fazla değildir. (b)'deki maskede imgedeki önemli anlamsal içerik gizlendiğinden, CRG kodlayıcı maskelenen imgeyi anlamlı bir gizli alana gömmekte zorlanmaktadır. Her iki üretmenin ikinci satırlarında görüldüğü gibi ilk maskelenen imge olan (a) yeterli değişirliğe sahip bir bölgeye, ikincisi ise yani (b), bir ölü bölgeye gömülmektedir. Oluşturulan (b) imgesinde görüldüğü üzere, gömülü nokta etrafındaki yoğunluk, o bölgede makul bir yüz imgesi elde etmek için yeterli olmamaktadır.



Şekil 4.11 Önerilen yöntemin zayıflığı. (a) başarılı örnek (b) başarısız örnek. Her örnek için, üst sıra: her bir CRG döngüsünün sonucunu, alt sıra: ilgili döngü için Üretici uzayında üretilen imgeyi göstermektedir

4.8 Değerlendirme

Yüz imgesi içboyama çalışmaları incelendiğinde, Kodlayıcı-Kodçözücü mimarisinde kurulan modellerin genel olarak farklı kısıtlamalara sahip oldukları gözlenmektedir. Bu bölümde, Kodlayıcı-Üretici mimarisinde kurulan bir modelin sağladığı avantajlar detaylı olarak ele alınmıştır. Bu kapsamda bir yöntem önerilmiş ve bu alanda geliştirilen çalışmalarla nitel olarak karşılaştırılmıştır. Önerilen yöntemin, ilgili modellerin eğitim aşamasında kullandıkları maske türlerinde ilgili modellere yakın performans sergilediği ve yüz veya arka planın tamamen maskelendiği (diğer bir deyişle, büyük maskeler kullanıldığında) imgelerde daha başarılı olduğu gösterilmiştir.

Yapılan deneysel çalışmalarda, önerilen modelin başarısının Üretici uzayının yeterliliği ile doğrudan ilişkili olduğu gözlenmiştir. Bu nedenle maskelenen imge Üretici uzayında değişirliğin az olduğu bir noktaya gömülmesi durumunda, döngülerdeki ilerlemelere

rağmen imge kalitesinde iyileşme olmamaktadır. Bu problemin üstesinden gelmek için, Üretici uzayını daha iyi temsil eden bir modelin ve bu uzaydaki değişirliği arttırmayı sağlayacak daha büyük bir veri setinin kullanılması gerektiği öngörülmektedir.



5. SONUÇ

Tez çalışmasında ÜÇA'larda imgeden gizli vektöre haritalamayı sağlayan CRG mimarisi önerilmiştir; bu mimari gerçek ve sahte imgelerin anlamlı bilgiler barındıran düşük boyutlu bir uzaya indirgenmesine ve bu imgelerin yeniden oluşturulmasına olanak sağlamaktadır. Bununla birlikte, CRG mimarisi kullanılarak imge öznitelik düzenleme ve yüz imgesi içboyama problemleri için etkili çözüm sunulmuştur. İmge öznitelik düzenleme probleminde, tüm veri setini etiketleme ihtiyacı gerektirmeden, öznitelik başına sadece bir referans gerçek imge çifti kullanılarak, öznitelik düzenleme işlemine olanak sağlanmıştır. Bu doğrultuda, farklı gerçek referans imge çiftleri kullanılarak elde edilen öznitelik vektörlerinin tutarlılığı analiz edilmiş ve bir özniteliliğin istenilen oranda değiştirilmesine olanak sağlayan k parametresi için güvenli aralık bulmayı sağlayacak bir yaklaşım önerilmiştir. Yüz imgesi içboyama probleminde ise bu kapsamda önerilen modellerin sahip olduğu farklı kısıtlamaların üstesinden gelmeyi sağlayan bir yöntem önerisinde bulunulmuştur. Önerilen yöntem, farklı maske türlerinin kullanılmasına, maske içeriği ile ilişkili sonuç üretmeye, çoklu sonuç üretmeye ve ezki tabanlı içboyamaya olanak sağlamaktadır. Aşağıda tez çalışması boyunca yapılan araştırmalar sonucu elde edilen bulgular, öneriler ve gelecekte yapılması planlanan çalışmalar detaylı olarak ele alınmıştır.

5.1 Bulgular ve Öneriler

Üretken çekışmeli ağlarda üretilen imgelerin çeşitliliğini arttırmak ve istikrarı sağlamak için yakın zamanda önerilen yaklaşımlar kullanılarak deneysel araştırmalar yapılmıştır. Farklı veri setlerinde başarılı sonuçlar sağlayan modellerin, aynı parametreler kullanılarak *CelebA* veri seti için test edildiğinde, benzer sonuçların alınmadığı gözlenmiştir. Lucic vd. (2018)'nin de belirttiği gibi çoğu modelin başarısı, kullanılan veri seti ve hiperparametrelerin doğru ayarlanması ile orantılı olduğu gözlenmektedir. Kademeli olmayan modeller kullanılarak *CelebA* veri seti için hiperparametre ayarlaması yapıldığında, üretilen imgelerin kalitesinin arttığı gözlemlenmiş; ancak istenilen seviyede sonuçlar elde edilememiştir.

Üretken çekişmeli ağ eğitim deneylerinde, kullanılan veri seti çözünürlüğü, kırpma (crop) metodu ve yapılan ön işlemlerin model başarısında önemli olduğu gözlenmiştir. *CelebA* veri seti için standart bir kırpma metodu olmadığı gözlenmektedir. Ayrıca modellerin performansı incelenirken model eğitiminde kullanılan kırpma metodunun dışında bir metot kullanılması durumunda, model performanslarının farklılık gösterdiği gözlenmiştir. Bu nedenle, model karşılaştırılması yapılırken ilgili modelin kullandığı metot dikkate alınmıştır. Bu kapsamda, Karras vd. (2017) tarafından önerilen, veri setinin çeşitli ön işlemlerden geçirildiği (yapay doku temizleme, *Gauss* filtreleme vb.), yüz imgesinde önemli noktaların (landmark locations) göz önünde bulundurularak kırpma işleminin yapıldığı ve öznelilikleri daha iyi göstermeye olanak sağlayan bir metot kullanılmıştır.

Yapılan deneyler sonucunda, yüksek çözünürlüklü imgelerde kademeli modellerin daha başarılı olduğu gözlenmiştir. Diğer modellerin imgeleri belirli seviyede iyileştirdiği; ancak detayları tam olarak yakalamadığı görülmüştür. Ayrıca ilgili veri seti için kullanılan kayıp fonksiyonu ve düzenleme tekniğinin de model başarısında önemli rol oynadığı gözlenmiştir.

Üretken modelleri karşılaştırmak için farklı metrikler (*IS*, *SSIM*, *FID* vb.) kullanılmaktadır. Bu metriklerin hiçbiri tam anlamıyla modellerin başarısını iyi yansıtamamaktadır. Son zamanlarda genel olarak *FID* metriğinin kullanımı yönünde bir eğilim söz konusudur. Bu nedenle model eğitimini sürdürmek veya modelleri karşılaştırmak için yapılan deneylerde bu metrik kullanılmıştır. Üretken modelin, uzayı iyi öğrenip öğrenmediğini incelemenin en iyi yöntemlerden bir tanesi de aradeğerlemeye olanak sağlayıp sağlamadığını incelemektir. Aradeğerleme sonuçlarının iyi olması, ağın genelleme yeteneğinin de iyi olduğunu göstermektedir.

İmge öznelilik düzenleme çalışmaları genel olarak etiketli veri seti ve koşullu ÜÇA mimarisi kullanılarak gerçekleştirilmektedir. Derin öğrenme çalışmaları çok büyük boyutlu veri setlerine ihtiyaç duyduğundan, bu veri setlerini etiketlemek zor ve zaman alıcı olmaktadır. Bu işlemi gerçekleştirmenin diğer bir yolu ÜÇA'lardaki doğrusal

yapıyı kullanmaktır. Bu yapıyı kullanabilmek için herhangi bir imgeyi temsil eden gizli vektörün bulunması gerekmektedir. Bu kapsamda, genel olarak önceden eğitilmiş bir Üretici ağı kullanılarak eniyilemeye dayalı yöntemler tercih edilmektedir; ancak bu yöntemler tek bir imge için çok sayıda yineleme gerektirmektedir. Bu işlevi tek bir yinelemede gerçekleştirmenin yolu parametrik ağların kullanımından geçmektedir. Fakat özellikle imge çözünürlüğü arttıkça parametrik ağlar ile bu işlevi gerçekleştirmek zor olmaktadır. Bu nedenle, bu alanda önerilen birçok modelin düşük çözünürlükte kaldığı gözlenmektedir. Yapılan deneysel çalışmalar neticesinde nispeten yüksek çözünürlükte (128x128) imge yeniden yapılandırma için CRG olarak adlandırılan parametrik bir model mimarisi önerilmiş ve başarılı sonuçlar elde edilmiştir. Eğitim aşamasında, çeşitli yakınsama ve istikrar problemleri ile karşılaşmıştır. *CelebA* veri setinde bu problemlerin üstesinden gelmek için çeşitli öneriler sunulmuştur; ancak farklı veri setleri, farklı ağ mimarileri veya farklı çözünürlükte farklı problemler ile karşılaşılabilir.

Önerilen CRG mimarisinde, bir imgenin düşük boyutlu bir uzayda temsil edilip yeniden yapılandırılmasını sağlamak için, Üretici ağ parametrelerinin sabit tutulup Kodlayıcı ağ parametrelerinin eniyilenmesi önerilmiştir. Kodlayıcı ağ eğitimi esnasında, Üretici ağ parametrelerinin de eniyilenmesinin bulanık çıktılar oluşturduğu gözlenmiştir.

Bir Üretici ağ ve bunun ters işlevini sağlayan bir Kodlayıcı ağ elde edilmesi durumunda, tek bir referans imge çifti kullanılarak ilgili özniteliğe karşılık gelen gizli vektörün elde edilebildiği gösterilmiştir. Ayrıca bu öznitelik vektörü kullanılarak farklı bir imgenin ilgili özniteliği pozitif veya negatif yönde değiştirilmiştir. Genel olarak imge öznitelik düzenlemede etiketli veri seti kullanılmaktadır; bu yöntem öznitelik düzenlemenin etiketlenen veri setinde tanımlanmış özniteliklerle sınırlı kalmasına neden olmaktadır. Öte yandan, önerilen yöntem ile bir çift referans imge elde edilebilmesi durumunda veri setlerinde tanımlanmamış öznitelikleri de değiştirmek mümkün hale gelmektedir. İlave olarak, herhangi bir öznitelik Denklem 3.7'deki k parametresi değiştirilerek istenilen oranda imgeye empoze edilebilmektedir.

Bir özniteliği temsil eden gizli vektör yönünü kullanarak bir imgede öznitelik düzenlemenin yapılabildiğini göstermek için bir analiz yöntemi sunulmuştur. Bu kapsamda, eğitim aşamasında kullanılmayan ilgili özniteliği içeren ve içermeyen imgeleri temsil eden gizli vektörlerin, öznitelik eksenine izdüşümleri alındığında, iki farklı dağılım olduğu görülmüştür. Bu dağılımlardan ilkinin, ilgili özelliği içeren diğerini ise ilgili özelliği içermeyen imgeler oluşturmuştur. Bu iki farklı dağılım, ilgili özniteliği içermeyen bir imgeyi temsil eden gizli vektörün, ilgili öznitelik yönünde hareket ettirilmesi durumunda diğer dağılım bölgesine geçeceğini, yani ilgili özniteliği taşıyacağını göstermektedir. Yapılan deneysel çalışmalarda bu durum gösterilmiştir.

Öznitelik eksenine izdüşüm alındığında, iki ayrı dağılımın oluşması beklenmektedir. Fakat yapılan testlerde bu dağılımların iç içe geçtiği görülmüştür. Bunun sebebini anlamak için, her iki dağılımın sol ve sağ uç kısımlarındaki ve ortadaki imgeler analiz edilmiştir. Örneğin ilgili özniteliği içeren dağılım incelendiğinde; sağ uçtaki imgelerin ilgili özniteliği çok güçlü bir şekilde yansıttığı, ortadaki imgelerin ilgili özniteliği taşıdığı ancak özniteliğin zayıf olduğu ve sol uçtaki imgelerin genel olarak ilgili özniteliği hiç taşımadığı görülmüştür. Bunun sebebi sol uçtaki imgelerin izdüşümü, ilgili özniteliği içermeyen imgelerin oluşturduğu dağılımın ortalamasına yakın düşmektedir. Bu problemin *CelebA* veri setindeki yanlış etiketlemeden kaynaklandığı görülmüştür. Önerilen yöntemde etiketli veri seti kullanılmadığından bu durum problem oluşturmamaktadır; ancak etiketli veri seti kullanan çalışmalarda bu durum problem meydana getirebilmektedir.

Yapılan analizlerde herhangi bir özniteliği temsil eden gizli vektörü belirlemek için, 50 çift referans imge kullanılmış ve bunların ortalaması alınmıştır. Çalışmanın devamında, tek bir referans imge çifti kullanarak bu işlemin yapıp yapılamayacağı ve farklı referans imge çiftleri kullanarak elde edilen öznitelik vektörlerin tutarlılığı incelenmiştir. Bu kapsamda, belirtilen bu 50 çift imge setinden rastgele 5 çift imge alınıp aynı analiz işlemleri uygulanmıştır. Sonuçlar incelendiğinde, benzer olarak iki dağılımın olduğu ve dağılım parametrelerinin çok az farklılık gösterdiği gözlenmiştir. Bu durum aynı öznitelik için farklı referans imge çiftlerinin kullanılabileceğini ve

öznitelik yönleri birebir aynı olmadığı için sonuçlarda çok az varyasyon oluşabileceğini göstermektedir.

Bir öznitelik yönünde hareket edildiğinde imgelerin nasıl değişkenlik gösterdiği araştırılmıştır. Bu kapsamda, k parametresinin geniş bir aralığı için gözlemler yapılmıştır. Herhangi bir imgenin yerel komşuluğunda kalındığında, yani küçük k değerlerinde, imgede ilgili öz niteliğin değiştiği ve diğer özniteliklerin korunduğu görülmüştür. Öznitelik vektörü yönünde ilerlemeye devam edildiğinde, yüz imgesinin farklı yüz imgesine dönüştüğü gözlenmiştir. Herhangi bir öznitelik vektörü yönünde çok fazla hareket edildiğinde ise girdi imgesinden bağımsız olarak tüm imgelerin pozitif ve negatif yönlerde belirli bir yüz imgesine yakınsadığı tespit edilmiştir.

Yapılan deneysel araştırmalarda, herhangi bir imge bir öz niteliği içermiyorsa genel olarak $k < 2$ seçilmesi durumunda ilgili öznitelik dışındaki diğer özniteliklerin korunduğu gözlenmiştir. Fakat eğer imge ilgili öz niteliği taşıyorsa bu öznitelik yönündeki küçük değişimler bile diğer özniteliklerin değişmesine neden olabilmektedir. Bu problemin üstesinden gelmek için daha önceden oluşturulan dağılım parametreleri kullanılarak (50 referans imge çifti kullanılarak elde edilen) sistematik bir çalışma yapılmış ve k parametresi için imgeye özel güvenli aralıkların belirlenmesi sağlanmıştır.

Üretilen imgelerin yeniden yapılandırma sonuçları oldukça iyi olmasına rağmen gerçek imgelerde bu başarı yeterince sağlanamamıştır. Bu problemin ÜÇA eğitimi esnasında yetersiz sayıda örnek kullanıldığından kaynaklandığı öngörülmektedir. ÜÇA eğitim aşamasında, daha çok varyasyona sahip ve daha büyük veri seti kullanılarak bu problemin üstesinden gelinebilir. Nihai olarak, önerilen yöntem kullanılarak 15 öznitelik (cinsiyet, yaş, kâkül, dalgalı saç, düz saç, kel, saç uzunluğu, sakal, gülme, gözlük, güneş gözlüğü, sarı saç, siyah saç, kahverengi saç ve ten rengi) için öznitelik düzenleme sonuçları gösterilmiştir. Bu öznitelikler, mevcut öznitelik düzenleme çalışmalarında kullanılan hemen hemen tüm öznitelikleri kapsamaktadır.

Yüz imgesi içboyama problemi kapsamında geliştirilen modeller incelendiğinde, her birinin farklı kısıtlamalara ve sınırlamalara sahip oldukları gözlenmektedir. Bu modeller genel olarak belli bir amaca yönelmekte ve diğer durumlarda başarısız olmaktadır. Yapısında Üretici ağ barındıran çalışmaların daha esnek sonuçlar sağladığı gözlenmektedir. Fakat bu yöntemler genel olarak eniyilemeye dayalı olduğundan, tek bir imge için çok sayıda yineleme gerektirmekte ve zaman maliyeti oluşmaktadır. Araştırmalar sonucunda, Kodlayıcı-Üretici mimarisinde kurulan bir modelin daha az sayıda yineleme ile sonuca ulaştığı gösterilmiştir. Bu kapsamda, Kısım 3.3'te önerilen CRG mimarisi kullanılarak yüz imgesi içboyama problemi için yeni bir yöntem önerilmiştir.

Yüz imgesi içboyama çalışmaları değerlendirildiğinde, modellerin genel olarak tek bir maske türünde iyi sonuçlar gösterdiği gözlenmektedir. Bu çalışmalarda araştırmacılar mümkün olması durumunda, her bir maske türü için farklı model eğitimleri yapmaktadır. Bu alanda yapılan ilk çalışmalarda ise model kısıtlamalarından (örneğin, lokal Ayırıştırıcı ağ kullanılması) dolayı farklı boyuttaki maske kullanımlarına izin verilmemektedir. Önerilen yöntemde herhangi bir kısıtlamanın olmadığı ve farklı maske türlerinin (düzenli, düzensiz ve elle çizilmiş) kullanılabildiği gösterilmiştir.

Bazı modeller küçük maske boyutlarında iyi sonuçlar sağlamakta; ancak büyük maskeler kullanıldığında istenmeyen sonuçlar üretilebilmektedir. Yapılan deneylerde önerilen yöntem kullanılarak imgenin yaklaşık %50'si maskelenmesine rağmen başarılı sonuçların elde edildiği gösterilmiştir. Özellikle yüzün veya arka planın tamamen maskelendiği örneklerde daha iyi sonuçlar üretilebildiği gözlenmiştir.

Bilindiği kadarıyla, yüz imgesi içboyama çalışmalarında, modellerin maskelenen kısma atanan piksel değerlerine bağlı olarak farklı sonuçlar üretemediği gözlenmektedir. Çünkü bu çalışmalar, hem eğitim hem de test aşamasında aynı piksel değerlerini atamakta ya da maskelenen bölgeye atanan piksel değerlerini göz ardı etmektedir. Yapılan deneysel çalışmalarda, önerilen yöntem kullanılarak maskelenen bölgeye

atanan farklı piksel değerlerine bağlı olarak farklı sonuçlar elde edilebildiği gösterilmiştir.

İlgili problem alanı için geliştirilen çalışmalar, genel olarak aynı girdi ve maske için tek ve sabit sonuç üretmektedir. Yakın zamanda bazı yaklaşımların, çoklu ve çeşitli sonuçlar üretebildiği gözlenmektedir; ancak bu çalışmalar, genel olarak eğitim aşamasında etiketli veri setlerine ihtiyaç duymaktadır. Önerilen yöntemde, Ayrıştırıcı ağ devre dışı bırakılarak bazı imgeler için birkaç döngü sonucunun kullanılabileceği gösterilmiştir; bu da çoklu ve çeşitli sonuç elde etmeyi sağlamıştır.

Yüz imgesi içboyama probleminde, kullanıcı yönlendirilmesi ile çıktı üretimi istenen bir durumdur; bu, kullanıcının maskelenen bölgede çeşitli çizimler yaparak uyumlu çıktı üretmesini sağlamaktadır. Ancak bu özelliği sağlayan çok az çalışma mevcuttur ve bunların bir kısmı eğitim aşamasında ön bilgiye (örneğin imgenin kenar haritası) ihtiyaç duymaktadır. Önerilen yöntem, bu kapsamda herhangi ön bilgi kullanılmamasına rağmen eskiz tabanlı içboyamaya imkân sağlamaktadır.

Önerilen yöntemin başarısı, Üretici uzayının yeterliliği ve Kodlayıcı ağın eşlemeyi doğru yapmasıyla ilişkilidir. Bu nedenle, maskelenen imge Üretici uzayında değişirliğin az olduğu bir noktaya düşmesi durumunda, önerilen yöntem başarısız olmaktadır. Kısım 3.3'te tanıtılan CRG mimarisi, gerçek imgelerin yeniden yapılandırılması bakımından diğer modellere üstünlük kurmasına rağmen hala imgeleri istenen seviyede yeniden oluşturamadığı gözlenmektedir. Özellikle gerçek imgelerde, büyük bir bölgenin maskelenmesi sonucu, maskelenen imge genel olarak Üretici uzayında iyi tanımlanmamış bir bölgeye gömülmektedir; bu durum, döngüler ilerlemesine rağmen imgenin iyileşmemesi ile sonuçlanmaktadır. Bu problemin üstesinden gelmek için önerilen modelde üç iyileştirme yapılabilir. Bunlar:

- Üretici uzayını daha iyi temsil eden, veri dağılımını daha iyi öğrenen bir ÜÇA mimarisi kullanılabilir,
- Modelin yüz imgesindeki öznitelikleri daha iyi öğrenmesi için daha büyük boyutlu veri seti kullanılabilir,

- Üretici ağıın karşılıklı eşlemesini öğrenen Kodlayıcı ağı için daha yoğun ve derin bir model mimarisi kullanılabilir.

5.2 Gelecekte Yapılması Planlanan Çalışmalar

Tez kapsamında önerilen CRG mimarisi ve çözüm sunduğu; imge öznitelik düzenleme ve yüz imgesi içboyama problemleri ele alınmıştır. Önerilen model benzer birçok problem alanında kullanılabilir. Bu kapsamda, önerilen modelin hem ele alınan problemlerde daha iyi sonuçlar almasını sağlamak hem de benzer birçok problemde kullanışlı hale getirmek için performansını arttırmaya yönelik çalışmaların yapılması planlanmaktadır. Model performansını arttırmak için:

1. Üretici uzayını daha iyi temsil eden bir ÜÇA modeli kullanılması planlanmaktadır. Mevcut çalışmada, *PGGAN* modeli referans alınmıştır. Ancak son zamanlarda *StyleGAN* ve *StyleGAN2* çalışmaları kullanılarak daha iyi sonuçların alındığı gözlenmektedir; diğer bir deyişle, daha düşük *FID* skorları elde edilmektedir. Bu nedenle Üretici ağıın, bu modellerden biri kullanılarak elde edilmesi planlanmaktadır.
2. Üretici ağıın ters eşlemesini sağlamak için *VGG16* tabanlı bir mimari kullanılmıştır. Bu model sığ olmasına rağmen çok fazla parametre barındırmaktadır. Tez çalışmasından sonra *DenseNet*, *ResNet50* (He vd. 2016) vb. tabanlı daha derin bir model kullanılması planlanmaktadır.
3. Tez çalışmasında gizli uzay z 'ye kodlanma yapılmaktadır. Yakın zamanda önerilen *StyleGAN* çalışmasında, mimaride mevcut olan W veya W^+ uzaylarına kodlamanın daha iyi olduğu gözlenmektedir. Çünkü bu uzaylarda özniteliklerin daha ayrık bir yapıya sahip oldukları belirtilmiştir. Bu nedenle, geliştirilecek modelde bu uzaylara kodlama yapılması planlanmaktadır.
4. Tez çalışması esnasında, parametrik modeller kullanılarak ters eşleme probleminde 128×128 çözünürlüğünün ötesine geçilemediği gözlenmektedir. Yakın zamanda yapılan çalışmalarda istenen seviyede olmasa da 256×256 çözünürlüğünde sonuçların üretildiği gözlenmektedir. Bu nedenle 256×256 çözünürlüğü için model geliştirilmesi planlanmaktadır.

KAYNAKLAR

- Abdal, R., Qin, Y. and Wonka, P. 2020. Image2StyleGAN++: How to Edit the Embedded Images?. IEEE Conference on Computer Vision and Pattern Recognition. 13 - 19 Jun 2020; Seattle/United States.
- Antipov, G., Baccouche, M. and Dugelay, J. L. 2017. Face aging with conditional generative adversarial networks. In 2017 IEEE International Conference on Image Processing (ICIP). 17-20 September 2017; Beijing/China.
- Arjovsky, M., Chintala, S. and Bottou, J. 2017. Wasserstein generative adversarial networks. 34 th International Conference on Machine Learning (ICML). 6-11 August 2017; Sydney/Australia.
- Barnes, C., Shechtman, E., Finkelstein, A., and Goldman, D. B. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph, 28(3); 24.
- Bengio, Y. and Bengio, S. 2000. Modeling high-dimensional discrete data with multi-layer neural networks. Advances in Neural Information Processing Systems, 12; 400-406.
- Berthelot, D., Schumm, T. and Metz, L. 2017. Began: Boundary equilibrium generative adversarial networks. arXiv preprint arXiv:1703.10717.
- Chen, T., Zhai, X., Ritter, M., Lucic, M. and Houlsby, N. 2019. Self-supervised gans via auxiliary rotation loss. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 15-21 June 2019; Long Beach/United States.
- Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., ... and Abbeel, P. 2016. Variational lossy autoencoder. arXiv preprint arXiv:1611.02731.
- Chen, Z., Nie, S., Wu, T. and Healey, C. G. 2018. High resolution face completion with multiple controllable attributes via fully end-to-end progressive generative adversarial networks. arXiv preprint arXiv:1801.07632.
- Choi, Y., Choi, M., Kim, M., Ha, J. W., Kim, S. and Choo, J. 2018. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 18-22 June 2018; Salt Lake City/Utah.
- Creswell, A. and Bharath, A. A. 2018. Inverting the generator of a generative adversarial network. IEEE transactions on neural networks and learning systems, 30(7); 1967-1974.

- Darabi, S., Shechtman, E., Barnes, C., Goldman, D. B., and Sen, P. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)*, 31(4); 1-10.
- Deng, J., Dong, W., Socher, R., Li, L. J., Li, K. and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 20-25 Jun 2009; Miami, FL/USA.
- Dogan, Y. and Keles, H. Y. 2019. Stability and Diversity in Generative Adversarial Networks. In *2019 27th Signal Processing and Communications Applications Conference (SIU)*. 24-26 April 2019; Sivas/Turkey.
- Dogan, Y. and Keles, H. Y. 2020. Semi-supervised Image Attribute Editing using Generative Adversarial Networks. *Neurocomputing*, 401; 338-352.
- Dogan, Y. and Keles, H. Y. 2021. Iterative Facial Image Inpainting using Cyclic Reverse Generator. *arXiv preprint arXiv:2101.07036*.
- Donahue, J., Krähenbühl, P. and Darrell, T. 2016. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*.
- Dumoulin, V., Belghazi, I., Poole, B., Mastropietro, O., Lamb, A., Arjovsky, M. and Courville, A. 2016. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*.
- Frey, B. J., Brendan, J. F. and Frey, B. J. 1998. *Graphical models for machine learning and digital communication*. MIT press, 717;718.
- Frey, B. J., Hinton, G. E. and Dayan, P. 1995. Does the wake-sleep algorithm produce good density estimators?. In *Advances in Neural Information Processing Systems (NIPS)*. 27-30 November 1995; Denver, CO/USA.
- Glorot, X., Bordes, A. and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *International conference on artificial intelligence and statistics (AISTATS)*. 11 – 13 April 2011; Fort Lauderdale, Florida/USA.
- Gong, X., Chang, S., Jiang, Y. and Wang, Z. 2019. Autogan: Neural architecture search for generative adversarial networks. *IEEE International Conference on Computer Vision (ICCV)*, 27 October - 2 November 2019; Seoul/Korea.
- Goodfellow, I. 2016a. NIPS 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*.
- Goodfellow, I., Bengio, Y., Courville, A. and Bengio, Y. 2016b. *Deep learning*. MIT press, Cambridge.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. ... and Bengio, Y. 2014. Generative adversarial nets. In *Advances in Neural*

Information Processing Systems (NIPS), 8-13 December 2014; Montréal/Canada.

- Goodfellow, I., Warde-Farley, D., Mirza, M., Courville, A. and Bengio, Y. 2013. Maxout networks. In International Conference on Machine Learning (ICML), 16-21 June 2013; Atlanta/USA.
- Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. 2017. Improved training of wasserstein gans. In Advances in Neural Information Processing Systems (NIPS). 4-9 December 2017; Long Beach, CA/ USA.
- He, K., Zhang, X., Ren, S. and Sun, J. 2016. Deep residual learning for image recognition. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 26 June – 1 July 2016; Las Vegas/Nevada.
- He, Z., Zuo, W., Kan, M., Shan, S. and Chen, X. 2019. Attgan: Facial attribute editing by only changing what you want. IEEE Transactions on Image Processing. 28(11); 5464-5478.
- Heljakka, A., Solin, A. and Kannala, J. 2018. Pioneer networks: Progressively growing generative autoencoder. In Asian Conference on Computer Vision (ACCV). 2-6 December 2018; Perth, WA/Australia.
- Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Advances in Neural Information Processing Systems (NIPS), 4-9 December 2017; Long Beach, CA/ USA.
- Hinton, G. E. 2007. Learning multiple layers of representation. Trends in cognitive sciences, 11(10); 428-434.
- Hinton, G. E., Sejnowski, T. J. and Ackley, D. H. 1984. Boltzmann machines: Constraint satisfaction networks that learn. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.
- Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.
- Huang, J. B., Kang, S. B., Ahuja, N., and Kopf, J. 2014. Image completion using planar structure guidance. ACM Transactions on graphics (TOG), 33(4); 1-10.
- Iizuka, S., Simo-Serra, E., and Ishikawa, H. 2017. Globally and locally consistent image completion. ACM Transactions on Graphics (ToG), 36(4); 1-14.
- Ioffe, S. and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning (ICML). 6-11 July 2015; Lille/France.

- Isola, P., Zhu, J. Y., Zhou, T., and Efros, A. A. 2017. Image-to-image translation with conditional adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21-26 July 2017; Honolulu, Hawaii/USA.
- Jaiswal, A., AbdAlmageed, W., Wu, Y. and Natarajan, P. 2018. Bidirectional conditional generative adversarial networks. In Asian Conference on Computer Vision (ACCV). 2-6 December 2018; Perth, WA/Australia.
- Jarrett, K., Kavukcuoglu, K., Ranzato, M. A. and LeCun, Y. 2009. What is the best multi-stage architecture for object recognition?. IEEE International Conference on Computer Vision (ICCV), 27 September - 4 October 2009; Kyoto/Japan.
- Jolicœur-Martineau, A. 2018. The relativistic discriminator: a key element missing from standard GAN. arXiv preprint arXiv:1807.00734.
- Karras, T., Aila, T., Laine, S. and Lehtinen, J. 2017. Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv preprint arXiv:1710.10196.
- Karras, T., Laine, S. and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 15-21 June 2019; Long Beach/United States.
- Kingma, D. P. 2013. Fast gradient-based inference with continuous latent variable models in auxiliary form. arXiv preprint arXiv:1306.0733.
- Kingma, D. P. ve Ba, J. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. 2016. Improved variational inference with inverse autoregressive flow. In Advances in Neural Information Processing Systems (NIPS). 5-10 December 2016; Barcelona/Spain.
- Kreps, D. M. 1989. Nash equilibrium. In Game Theory (pp. 167-177). Palgrave Macmillan, London.
- Krizhevsky, A. and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report, University of Toronto.
- Kullback, S. and Leibler, R. A. 1951. On information and sufficiency. The annals of mathematical statistics, 22(1); 79-86.
- Larsen, A. B. L., Sønderby, S. K., Larochelle, H. and Winther, O. 2016. Autoencoding beyond pixels using a learned similarity metric. In International Conference on Machine Learning (ICML). 19-24 June 2016; New York City, NY/ USA.

- LeCun, Y., Cortes, C. and Burges, C. J. 2010. MNIST handwritten digit database. AT&T Labs [Online]. Available: <http://yann.lecun.com/exdb/mnist>, 2.
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... and Shi, W. 2017. Photo-realistic single image super-resolution using a generative adversarial network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21-26 July 2017; Honolulu, Hawaii/USA.
- Li, Y., Liu, S., Yang, J., and Yang, M. H. 2017. Generative face completion. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21-26 July 2017; Honolulu, Hawaii/USA.
- Lim, J. H. and Ye, J. C. 2017. Geometric gan. arXiv preprint arXiv:1705.02894.
- Lipton, Z. C. and Tripathi, S. 2017. Precise recovery of latent vectors from generative adversarial networks. arXiv preprint arXiv:1702.04782.
- Liu, G., Reda, F. A., Shih, K. J., Wang, T. C., Tao, A., and Catanzaro, B. 2018. Image inpainting for irregular holes using partial convolutions. In Proceedings of the European Conference on Computer Vision (ECCV). 8-14 September (85-100); Munich/Germany.
- Liu, Y. and Chen, J. 2021. Unsupervised face Frontalization for pose-invariant face recognition. Image and Vision Computing, 106; 104093.
- Liu, Z., Luo, P., Wang, X. and Tang, X. 2015. Deep learning face attributes in the wild. IEEE International Conference on Computer Vision (ICCV), 13-16 December 2015; Santiago/Chile.
- Lucic, M., Kurach, K., Michalski, M., Gelly, S. and Bousquet, O. 2018. Are gans created equal? a large-scale study. In Advances in Neural Information Processing Systems (NIPS). 2-8 December 2018; Montréal/Canada.
- Luo, J., Xu, Y., Tang, C. and Lv, J. 2017. Learning inverse mapping by autoencoder based generative adversarial nets. In International Conference on Neural Information Processing (ICONIP). 14-18 November 2017; Guangzhou/China.
- Maas, A. L., Hannun, A. Y. and Ng, A. Y. 2013. Rectifier nonlinearities improve neural network acoustic models. In International Conference on Machine Learning (ICML), 16-21 June 2013; Atlanta/USA.
- Mirza, M. and Osindero, S. 2014. Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784.
- Miyato, T., Kataoka, T., Koyama, M. and Yoshida, Y. 2018. Spectral normalization for generative adversarial networks. International Conference on Learning Representations (ICLR), Apr 30 - May 3 2018; Vancouver, BC/Canada.

- Nazeri, K., Ng, E., and Ebrahimi, M. 2018. Image colorization using generative adversarial networks. In International Conference on Articulated Motion and Deformable Objects (AMDO). 12-13 July 2018; Palma de Mallorca/ Spain.
- Nazeri, K., Ng, E., Joseph, T., Qureshi, F. Z., and Ebrahimi, M. 2019. Edgeconnect: Generative image inpainting with adversarial edge learning. arXiv preprint arXiv:1901.00212.
- Niu, X. M. and Jiao, Y. H. 2008. An overview of perceptual hashing. *Acta Electronica Sinica*, 36(7); 1405-1411.
- Odena, A., Olah, C. and Shlens, J. 2017. Conditional image synthesis with auxiliary classifier gans. 34 th International Conference on Machine Learning (ICML). 6-11 August 2017; Sydney/Australia.
- Oord, A. V. D., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., ... and Kavukcuoglu, K. 2016c. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499.
- Oord, A. V. D., Kalchbrenner, N. and Kavukcuoglu, K. 2016a. Pixel recurrent neural networks. In International Conference on Machine Learning (ICML). 19-24 June 2016; New York City, NY/ USA.
- Oord, A. V. D., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A. and Kavukcuoglu, K. 2016b. Conditional image generation with pixelcnn decoders. arXiv preprint arXiv:1606.05328.
- Palsson, S., Agustsson, E., Timofte, R. and Van Gool, L. 2018. Generative adversarial style transfer networks for face aging. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 18-22 June 2018; Salt Lake City/Utah.
- Park, S. W. and Kwon, J. 2019. Sphere generative adversarial network based on geometric moment matching. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 15-21 June 2019; Long Beach/United States.
- Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., and Efros, A. A. 2016. Context encoders: Feature learning by inpainting. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 26 June – 1 July 2016; Las Vegas/Nevada.
- Perarnau, G., Van De Weijer, J., Raducanu, B. and Álvarez, J. M. 2016. Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355.
- Pérez, P., Gangnet, M. and Blake, A. 2003. Poisson image editing. In ACM SIGGRAPH. 313-318.
- Petrov, D. Wavelet image hash in Python, 2016. (<https://fullstackml.com/wavelet-image-hash-in-python-3504fdd282b5/>). [Online; accessed 19-June- 2019].

- Radford, A., Metz, L. and Chintala, S. 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. The International Conference on Learning Representations (ICLR). 2-4 May 2016; San Juan/Puerto Rico.
- Rezende, D. J., Mohamed, S. and Wierstra, D. 2014. Stochastic backpropagation and approximate inference in deep generative models. In International Conference on Machine Learning (ICML). 21-26 June 2014; Beijing/China.
- Rosca, M., Lakshminarayanan, B., Warde-Farley, D. and Mohamed, S. 2017. Variational approaches for auto-encoding generative adversarial networks. arXiv preprint arXiv:1706.04987.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., ... and Fei-Fei, L. 2015. Imagenet large scale visual recognition challenge. International journal of computer vision, 115(3); 211-252.
- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A. and Chen, X. 2016. Improved techniques for training gans. In Advances in Neural Information Processing Systems (NIPS). 5-10 December 2016; Barcelona/Spain.
- Shamsolmoali, P., Zareapoor, M., Wang, R., Jain, D. K. and Yang, J. 2019. G-GANISR: Gradual generative adversarial network for image super resolution. Neurocomputing, 366; 140-153.
- Shao, H., Wang, Y., Fu, Y. and Yin, Z. 2020. Generative image inpainting via edge structure and color aware fusion. Signal Processing: Image Communication, 87; 115929.
- Simakov, D., Caspi, Y., Shechtman, E., and Irani, M. 2008. Summarizing visual data using bidirectional similarity. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 23-28 June 2008; Anchorage, Alaska/USA.
- Simonyan, K. and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research, 15(1); 1929-1958.
- Susskind, J., Anderson, A. and Hinton, G. E. 2010. The Toronto face dataset. Technical Report, University of Toronto.
- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. and Fergus, R. 2013. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199.

- Tavakolian, M. and Hadid, A. 2018. Deep discriminative model for video classification. In Proceedings of the European Conference on Computer Vision (ECCV). 8-14 September; Munich/Germany.
- Telea, A. 2004. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1); 23-34.
- Ulyanov, D., Vedaldi, A. and Lempitsky, V. 2018. It takes (only) two: Adversarial generator-encoder networks. *AAAI Conference on Artificial Intelligence*. 2-7 February 2018; New Orleans, Louisiana/USA.
- Wang, C. C., Liu, H. H., Pei, S. C., Liu, K. H. and Liu, T. J. 2019a. Face aging on realistic photos by generative adversarial networks. In 2019 IEEE International Symposium on Circuits and Systems (ISCAS). 26-29 May 2019; Sapporo, Hokkaido/Japan.
- Wang, C., Xu, C., Wang, C., and Tao, D. 2018a. Perceptual adversarial networks for image-to-image transformation. *IEEE Transactions on Image Processing*, 27(8); 4066-4079.
- Wang, Q., Fan, H., Sun, G., Cong, Y. and Tang, Y. 2019b. Laplacian pyramid adversarial network for face completion. *Pattern Recognition*, 88; 493-505.
- Wang, Y., Tao, X., Qi, X., Shen, X. and Jia, J. 2018b. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*. 3-8 December 2018; Montréal/Canada.
- Wang, Z., Bovik, A. C., Sheikh, H. R. and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4); 600-612.
- Wei, J., Lu, G., Liu, H. and Yan, J. 2019. Facial image inpainting with deep generative model and patch search using region weight. *IEEE Access*, 7; 67456-67468.
- Wei, X., Gong, B., Liu, Z., Lu, W. and Wang, L. 2018. Improving the improved training of wasserstein gans: A consistency term and its dual effect. *arXiv preprint arXiv:1803.01541*.
- White, T. 2016. Sampling generative networks. *arXiv preprint arXiv:1609.04468*.
- Xu, Q., Huang, G., Yuan, Y., Guo, C., Sun, Y., Wu, F., and Weinberger, K. 2018. An empirical study on evaluation metrics of generative adversarial networks. *arXiv preprint arXiv:1806.07755*.
- Yan, Y., Ren, W. and Cao, X. 2018. Recolored image detection via a deep discriminative model. *IEEE Transactions on Information Forensics and Security*, 14(1); 5-17.

- Yang, C. and Lv, Z. 2020. Gender based face aging with cycle-consistent adversarial networks. *Image and Vision Computing*, 100; 103945.
- Yeh, R. A., Chen, C., Yian Lim, T., Schwing, A. G., Hasegawa-Johnson, M. and Do, M. N. 2017. Semantic image inpainting with deep generative models. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 21-26 July 2017; Honolulu, Hawaii/USA.
- Yin, G., Liu, B., Sheng, L., Yu, N., Wang, X., and Shao, J. 2019. Semantics disentangling for text-to-image generation. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 15-21 June 2019; Long Beach/United States.
- Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T. and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. and Huang, T. S. 2018. Generative image inpainting with contextual attention. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 18-22 June 2018; Salt Lake City/Utah.
- Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X. and Huang, T. S. 2019. Free-form image inpainting with gated convolution. *IEEE International Conference on Computer Vision (ICCV)*, 27 October - 2 November 2019; Seoul/Korea.
- Zeiler, M. D. and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision (ECCV)*. 6-12 September 2014; Zurich/Switzerland.
- Zhang, H., Goodfellow, I., Metaxas, D. and Odena, A. 2019. Self-attention generative adversarial networks. In *International conference on machine learning (ICML)*. 9-15 June 2019; Long Beach, CA/USA.
- Zhang, R., Isola, P. and Efros, A. A. 2016. Colorful image colorization. In *European Conference on Computer Vision (ECCV)*, 8-16 October 2016. (649-666); Amsterdam/The Netherlands.
- Zhang, Z., Liang, R., Chen, X., Xu, X., Hu, G., Zuo, W. and Hancock, E. R. 2020. Semi-Supervised Face Frontalization in the Wild. *IEEE Transactions on Information Forensics and Security*, 16; 909-922.
- Zhang, Z., Xie, Y., and Yang, L. 2018. Photographic text-to-image synthesis with a hierarchically-nested adversarial network. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 18-22 June 2018; Salt Lake City/Utah.
- Zhao, J., Mathieu, M. and LeCun, Y. 2016. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*.

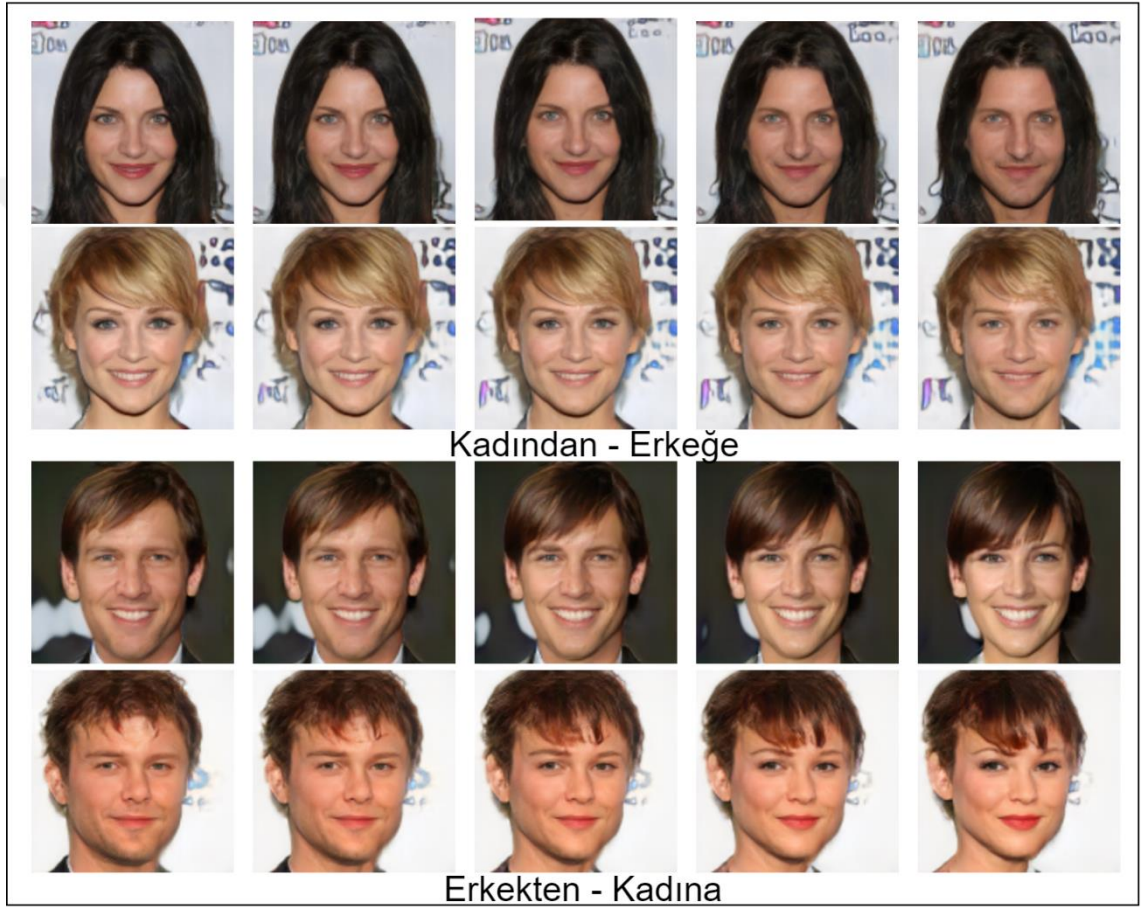
Zheng, C., Cham, T. J. and Cai, J. 2019. Pluralistic image completion. IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 15-21 June 2019; Long Beach/United States.

Zou, H., Ak, K. E. and Kassim, A. A. 2020. Edge-Gan: Edge Conditioned Multi-View Face Image Generation. In 2020 IEEE International Conference on Image Processing (ICIP). 25-28 October; United Arab Emirates.



EK Öznitelik Düzenleme Sonuçları

Cinsiyet özneliği için kullanılan referans imgeler



Kadından - Erkeğe

Erkekten - Kadına

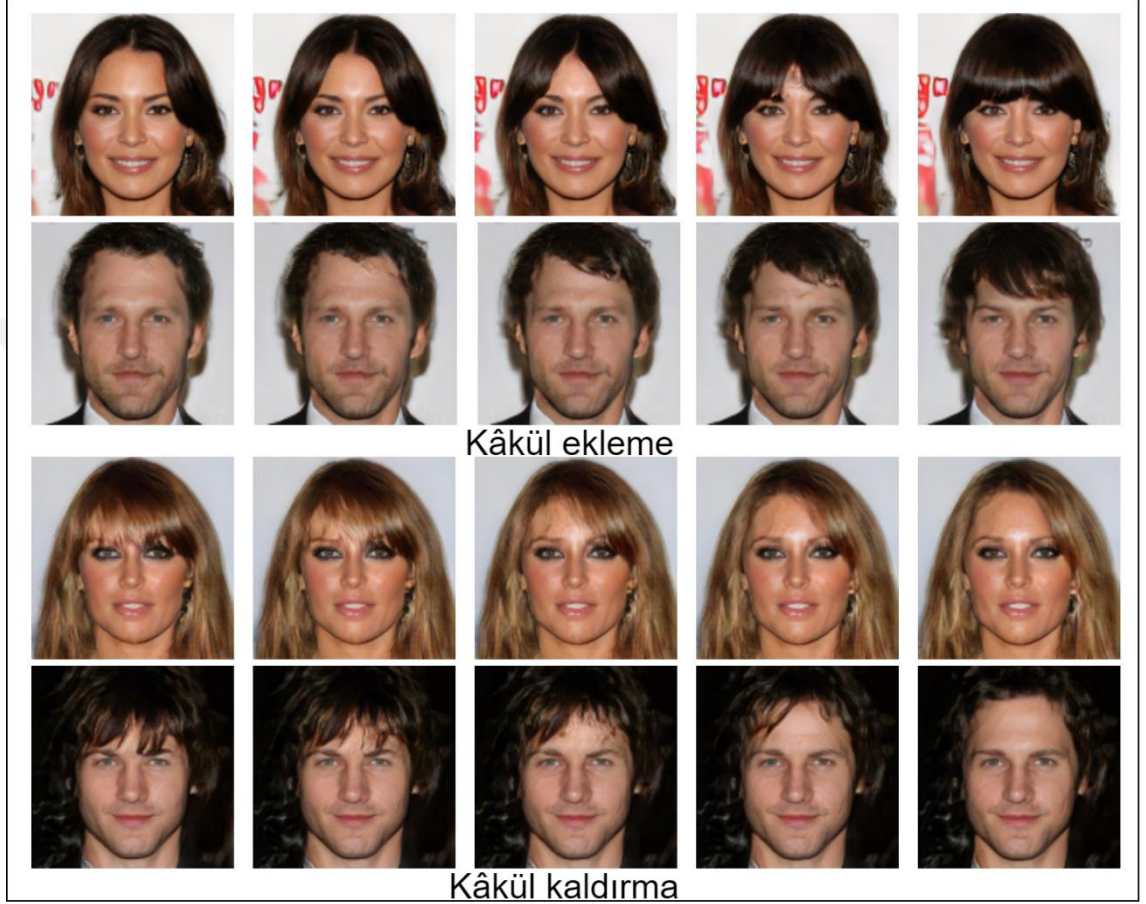
Şekil 1 Cinsiyet özneliğinin değiştirilmesi

Yaş özniteliği için kullanılan referans imgeler



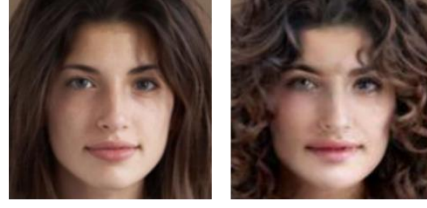
Şekil 2 Yaş özniteliğinin değiştirilmesi

Kâkûl özniteliği için kullanılan referans imgeler



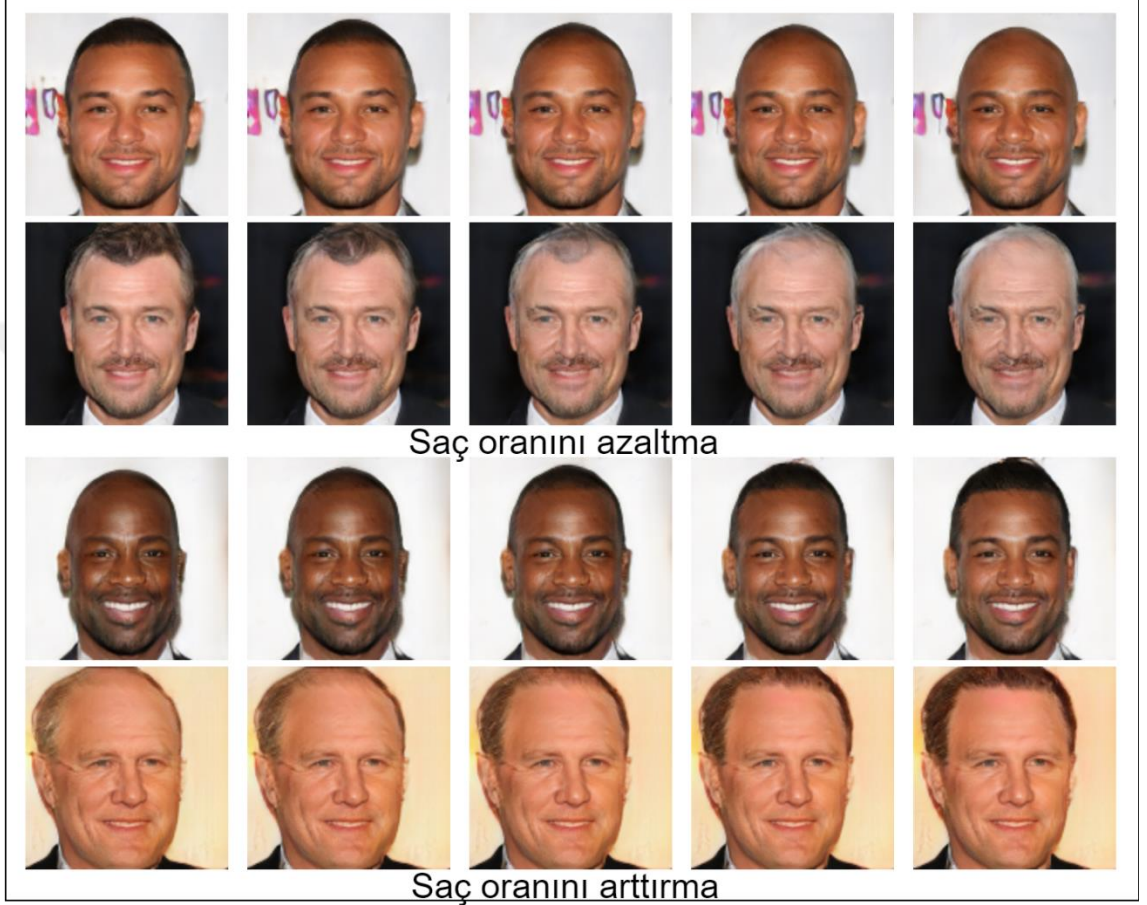
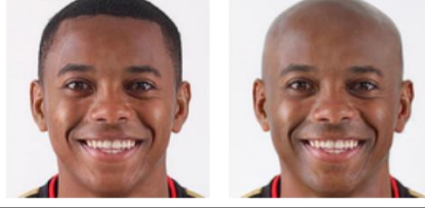
Şekil 3 Kâkûl özniteliğinin değiştirilmesi

Düz ve dalgalı saç özniteliği için kullanılan referans imgeler



Şekil 4 Dalgalı ve düz saç özniteliklerinin değiştirilmesi

Kel özniteliği için kullanılan referans imgeler



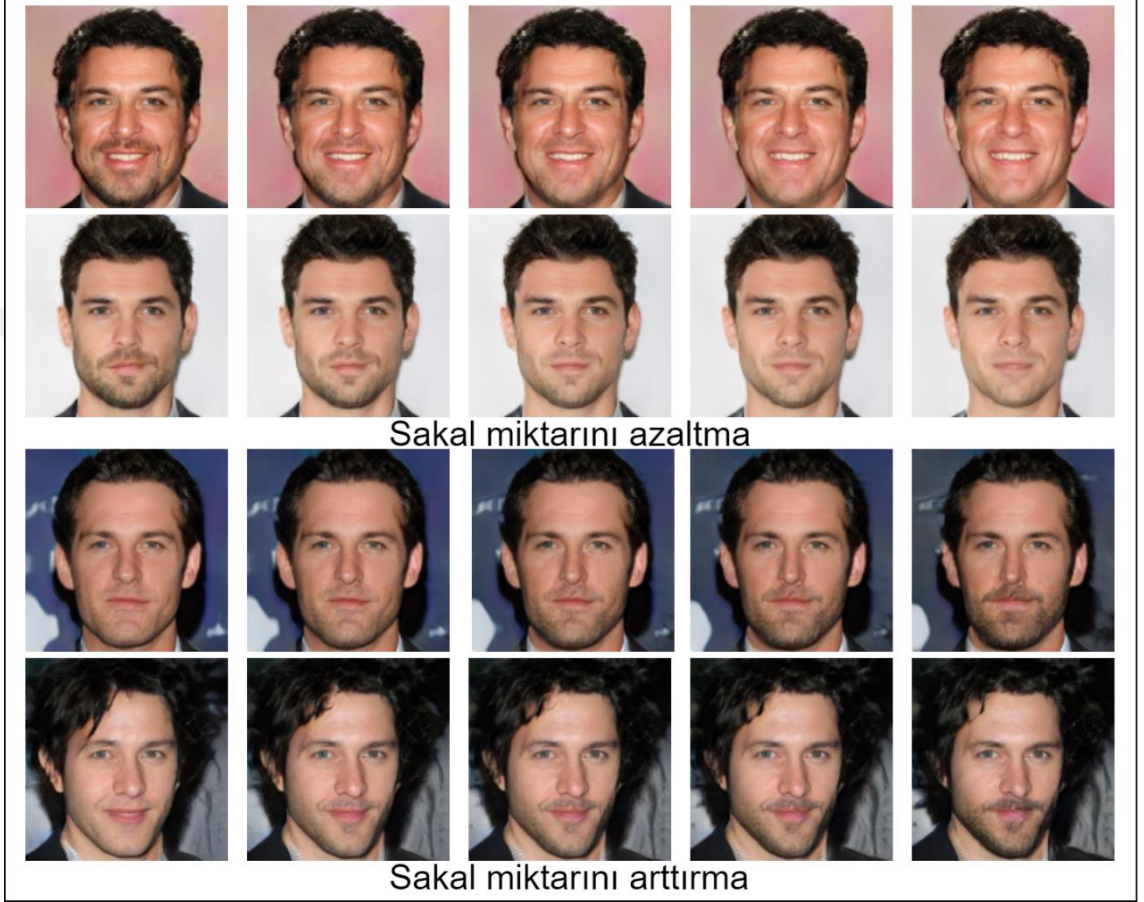
Şekil 5 Kel özneliğinin değiştirilmesi

Saç uzunluğu özniteliği için kullanılan referans imgeler



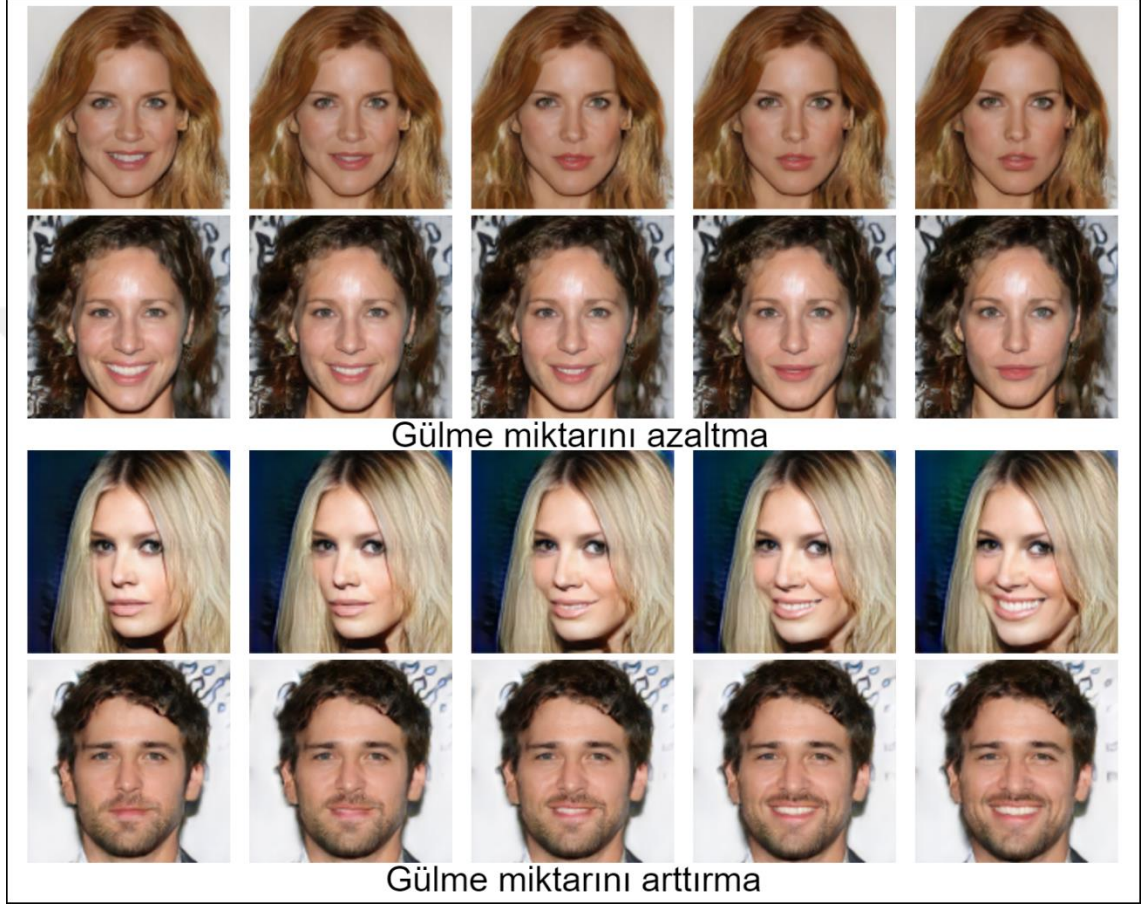
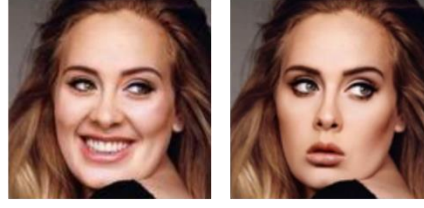
Şekil 6 Saç uzunluğu özniteliğinin değiştirilmesi

Sakal özniteliği için kullanılan referans imgeler



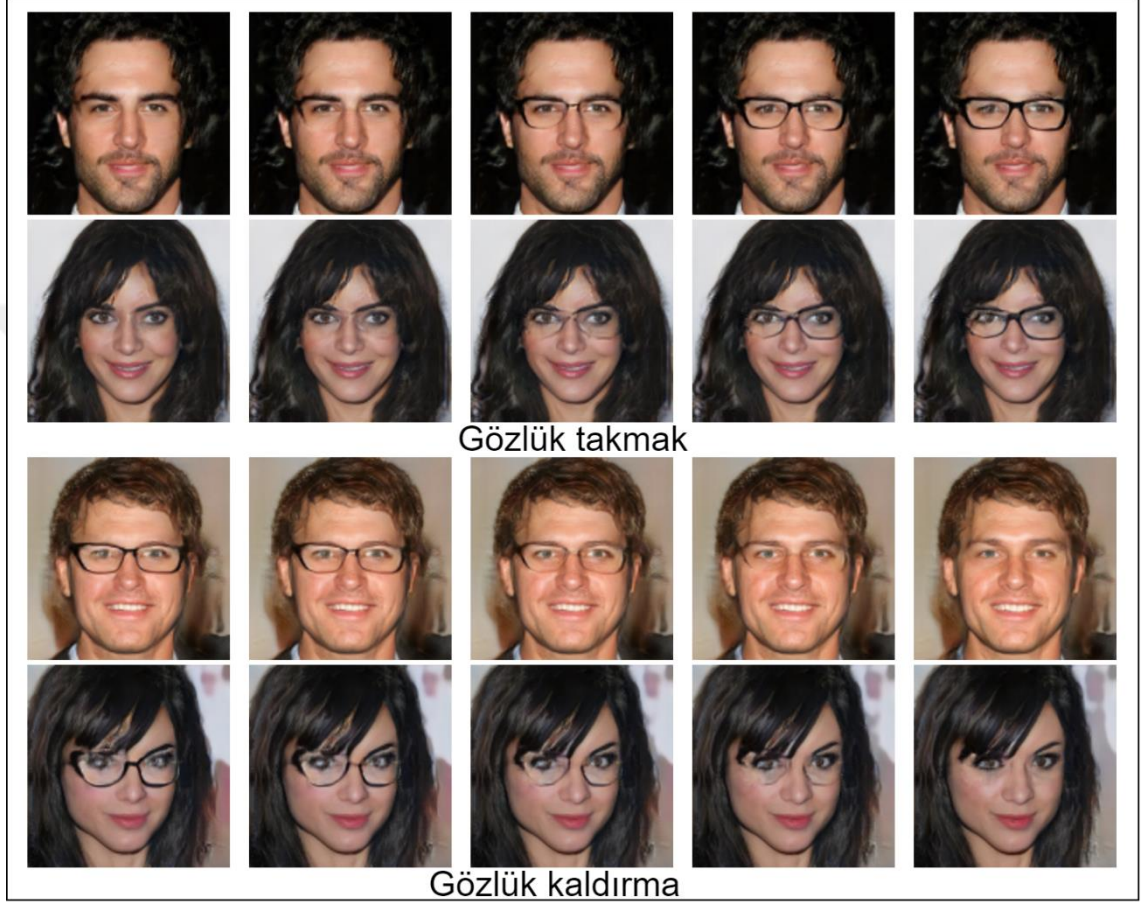
Şekil 7 Sakal özniteliğinin değiştirilmesi

Gülme özneliği için kullanılan referans imgeler



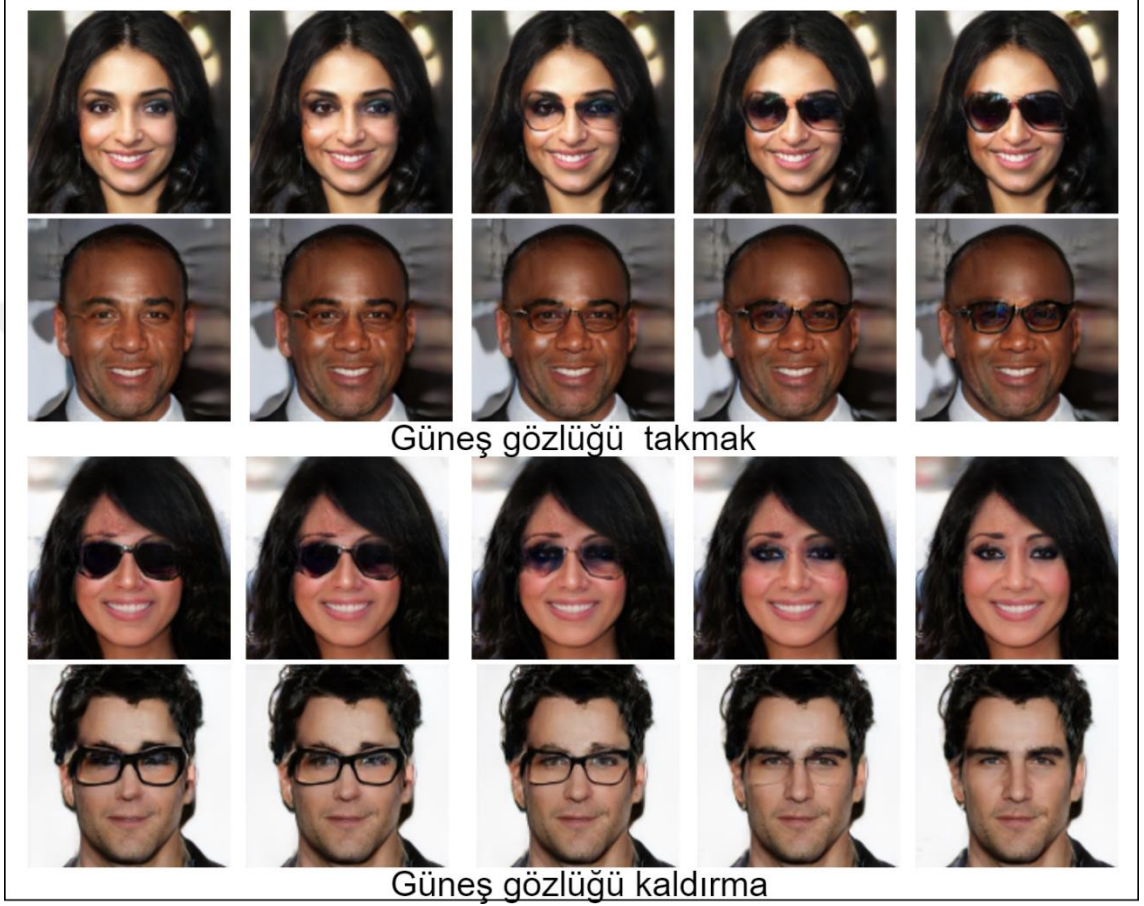
Şekil 8 Gülme özneliğinin değiştirilmesi

Gözlük özniteliği için kullanılan referans imgeler



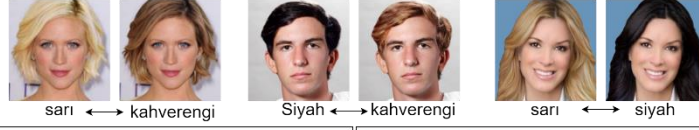
Şekil 9 Gözlük özniteliğinin değiştirilmesi

Güneş gözlüğü özniteliği için kullanılan referans imgeler



Şekil 10 Güneş gözlüğü özniteliğinin değiştirilmesi

Sarı, siyah ve kahverengi saç renkleri öznitelikleri için kullanılan referans imgeler



Şekil 11 Saç rengi ve ten rengi özniteliklerinin değiştirilmesi