

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ÖZ DENETİMLİ YÖNTEMLERLE GÖRSEL ÖZELLİK AYRIŞTIRMA

Abdurrahman Akın AKTAŞ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2023

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

ÖZ DENETİMLİ YÖNTEMLERLE GÖRSEL ÖZELLİK AYRIŞTIRMA

Abdurrahman Akın AKTAŞ

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. İman ASKERBEYLİ
Eş-Danışman: Doç. Dr. Hacer YALIM KELEŞ

Bu çalışmada, öz denetimli bir yöntem kullanılarak, önceden eğitilmiş çekişmeli üretici ağların gizli uzaylarındaki temel anlamsal eksenleri belirleyebilen bir yaklaşım sunulmaktadır. Bu bağlamda, yenilikçi bir yaklaşım olan Yön-Ayrıştırıcı model önerilmektedir. Bu evrimsel model ResNet18 öznitelik çıkarıcısı mimarisini ikiz ağ kurgusunda kullanmaktadır. Bir üretici ağ tarafından öz denetimli olarak oluşturulan görseller ile Yön-Ayrıştırıcı modelin eğitimi için veri kümesi oluşturulmaktadır. Oluşturulan verilerle eğitilen Yön-Ayrıştırıcı model, daha önce rastlamadığı iki verinin görsel özelliklerinin üretici ağın gizli uzayındaki farkını kodlayan yön vektörünü bulabilecek şekilde eğitilmektedir. MNIST ve FreyFaces veri kümeleri ile önceden eğitilmiş InfoGAN ve DCGAN modelleri kullanılarak yeni veri kümeleri elde edilmiştir. Çok modlu MNIST verilerinde tutarlı veri örnekleri oluşturabilmek için sezgisel bir benzerlik metriği kullanılmıştır. Bunun için önceden eğitilmiş birer değişimsel otokodlayıcı ve sınıflandırıcı ile gömme vektörleri oluşturulmuştur. Bu veriler ile elde edilen deneysel sonuçlar, Yön-Ayrıştırıcı model ile belirlenen yönler ile tayin edilen anlamsal özelliklerin farklı veri örneklerine uygulandığında verinin tutarlı bir şekilde değiştirildiğini göstermektedir.

Ocak 2023, 54 sayfa

Anahtar Kelimeler: Öz Denetim, Görsel Özellik Ayrıştırma, Evrimsel Sinir Ağı, Otokodlayıcı, Çekişmeli Üretici Ağlar

ABSTRACT

Master Thesis

VISUAL ATTRIBUTE DISENTANGLEMENT USING SELF-SUPERVISION

Abdurrahman Akın AKTAŞ

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

Supervisor: Prof. Dr. İman ASKERBEYLİ
Co-Supervisor: Doç Dr. Hacer YALIM KELEŞ

In this work, a self-supervised approach that can identify the semantic axes in the latent spaces of pre-trained adversarial generative networks is presented. Thus, a novel Direction-Disentangler model is proposed. This convolutional model employs the ResNet18 feature extractor architecture in a twin network fashion. In a self-supervised manner, a dataset for training the Direction-Disentangler model is created with the images generated by a generator network. This data is used to train the Direction-Disentangler model to identify the direction in the generator's latent space that encodes the discriminative features between two reference images. New datasets were obtained by using pre-trained InfoGAN and DCGAN models for the MNIST and FreyFaces dataset. A heuristic distance metric was used to generate consistent data samples for the multimodal MNIST data. For this purpose, a pre-trained variational autoencoder and classifier were employed to create embedding vectors. The experimental results using these datasets show that the identified direction can be applied to different reference images and is effective for editing their relevant semantic attributes.

January 2023, 54 pages

Key Words: Self-Supervision, Visual Attribute Disentanglement, Convolutional Neural Network, Autoencoder, Generative Adversarial Networks (GAN)

TEŞEKKÜR

Çalışmalarımı yönlendiren ve tez çalışmamın her aşamadinsa bana yol gösteren danışman hocalarım sayın Doç. Dr. Hacer YALIM KELEŞ (Hacettepe Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) ve sayın Prof. Dr. İman ASKERBEYLİ'ye (Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı) en derin duygularla teşekkür ederim.

Çalışmalarım boyunca ve tezimin yazım aşamasında bana destek olan arkadaşlarım Anıl Osman TUR ve Berkay SELBES'e teşekkür ederim.

Son olarak hayatım boyunca maddi ve manevi pek çok fedakârlık göstererek beni destekleyen ve her anımda yanımda olan eşim Zümrüt AKTAŞ, annem Sultan AKTAŞ, babam Kudret AKTAŞ ve ablalarım Belkıs AKTAŞ ve Fatıma AKTAŞ BOETTCHER'a tüm içtenliğimle minnettarlığımı belirterek, en derin duygularla teşekkür ederim.

Abdurrahman Akın AKTAŞ
Ankara, Ocak 2023

İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
TEŞEKKÜR	iv
SİMGELER VE KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
1. GİRİŞ.....	1
1.1 Tezin Amacı	3
1.2 Tezin Ana Katkısı.....	3
2. KURAMSAL TEMELLER VE KAYNAK ÖZETLERİ.....	4
2.1 Üretici Sinir Ağları.....	4
2.1.1 Otokodlayıcı.....	4
2.1.2 Değişimsel otokodlayıcı.....	6
2.1.3 Çekişmeli üretici sinir ağları (GAN)	8
2.2 Ayırıştırma (Disentanglement).....	18
3. YÖNTEM	35
3.1 Model Mimarisi	35
3.2 Kullanılan Üretici Modeller	36
3.3 Veri Kümeleri	37
3.3.1 MNIST veri kümesi.....	37
3.3.2 FreyFaces veri kümesi	38
4. BULGULAR VE TARTIŞMA	40
4.1 InfoGAN MNIST Yön-Ayrıştırıcısı	40
4.2 DCGAN MNIST Yön-Ayrıştırıcısı.....	43
4.3 InfoGAN FreyFaces Yön-Ayrıştırıcısı.....	46
5. SONUÇ	48
KAYNAKLAR	49
ÖZGEÇMİŞ.....	54

SİMGELER ve KISALTMALAR DİZİNİ

Θ	Thita
Φ	Fi
λ	Lamda

Kısaltmalar

GAN	Çekişmeli Üretici Ağ (Generative Adversarial Network)
VAE	Değişimsel Otokodlayıcı (Variational Autoencoder)
β -VAE	Beta-Değişimsel Otokodlayıcı (Beta-Variational Autoencoder)
cGAN	Koşullu GAN (Conditional GAN)
DCGAN	Derin Evrişimsel GAN (Deep Convolutional GAN)
MLP	Çok Katmanlı Algılayıcı (Multilayer Perceptron)
ResNet	Residual Network

ŞEKİLLER DİZİNİ

Şekil 2.1 Üç gizli katmana sahip ve darboğazı boyutsal olarak kısıtlanmış bir otokodlayıcının örnek mimarisi	6
Şekil 2.2 İki μ ve σ olmak üzere dört gizli katmana sahip değişimsel otokodlayıcının örnek mimarisi	8
Şekil 2.3 Çekişmeli üretici ağ mimarisi	9
Şekil 2.4 LSUN veri kümesiyle kullanılan DCGAN mimarisi	11
Şekil 2.5 GAN (sol) ve DCGAN (sağ) ile eğitimin farklı aşamalarında oluşturulan MNIST veri örnekleri (Radford vd., 2016)	12
Şekil 2.6 Yüz verileri ile eğitilmiş DCGAN'ın örnek çıktıları	13
Şekil 2.7 DCGAN'ın gizli uzayına ait vektörler üzerinde aritmetik işlemler uygulanarak oluşan çıktının özelliklerin değiştirilmesi.....	14
Şekil 2.8 InfoGAN mimarisi	16
Şekil 2.9 InfoGAN ve GAN'da c_1 'e değişim uygulanması	17
Şekil 2.10 InfoGAN'da c_2 ve c_3 kodlarına değişim uygulanması	17
Şekil 2.11 InfoGAN ve CelebA ile yapılan deneyler.....	18
Şekil 2.12 InterFaceGAN ile üretilmiş örnekler (Shen vd., 2020).....	20
Şekil 2.13 Gizli uzayda yön keşfi (Jahanian vd., 2020).....	20
Şekil 2.14 Bulunan yönleri uygulayarak elde edilen örnek değişimler (Jahanian vd., 2020).....	21
Şekil 2.15 Koşullu GAN Mimarisi	22
Şekil 2.16 Koşullu GAN ile üretilmiş örnek veriler (Mirza ve Osindero, 2014).....	23
Şekil 2.17 GANControl ile veri üretimi	24
Şekil 2.18 GANControl örnek çıktıları. Yaş ve Poz değerleri değiştirilerek oluşturulan örnekler (Shoshan vd., 2021).....	25
Şekil 2.19 DisCont ile Cars3D ve 3DShapes veri kümelerinden üretilen örnek görseller (Bhagat vd., 2020)	27
Şekil 2.20 DisCo ile ayrıştırılmış bir GAN ile üretilen örnek veriler (Ren vd., 2022) ...	29
Şekil 2.21 Farklı görsel özellikleri temsil eden yönlerde ilerlenerek elde edilen sonuçlar (Voynov ve Babenko, 2020).....	30
Şekil 2.22 WarpedGANSpace yönteminin StyleGAN2'ye uygulanarak elde edilen örnek değişimler (Tzelepis vd., 2021)	31
Şekil 2.23 CRG ile yön bulma örneği (Dogan ve Keles, 2020).....	33
Şekil 2.24 CRG ile tespit edilen yönlerin örneklerle uygulanması (Dogan ve Keles, 2020).....	34

Şekil 3.1 Geliştirilen Yön-Ayrıştırıcı yöntemin akış diyagramı	36
Şekil 4.1 InfoGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması.....	40
Şekil 4.2 InfoGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanması	40
Şekil 4.3 InfoGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği sola yaslanma özelliğinin diğer rakamlara uygulanması.....	42
Şekil 4.4 DCGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması.....	43
Şekil 4.5 DCGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanması	43
Şekil 4.6 DCGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği sağa yaslanma özelliğinin diğer rakamlara başarılı şekilde uygulanması	45
Şekil 4.7 DCGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği kalınlaştırma özelliğinin diğer rakamlara başarısız şekilde uygulanması	46
Şekil 4.8 InfoGAN ile rastgele üretilmiş yüz imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması.....	46
Şekil 4.9 InfoGAN ile rastgele üretilmiş yüz imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanmasının örnekleri.....	47

1. GİRİŞ

Makine öğrenmesi ve görüntü işleme alanlarında son yıllarda gerçekleşen gelişmelerle beraber veri kümelerine olan ihtiyaç artmaktadır (Elyan vd., 2022; Paullada vd., 2021; Voulodimos vd., 2018) fakat yeni ve kaliteli veri kümelerin üretimi çok emek ve kaynak gerektirmektedir. Bilgisayarla görü yöntemleri ile geometrik şekiller ve farklı görsellerin birleşimlerini oluşturmak görece basit olsa da bu verilerin kullanım durumları kısıtlıdır. İnsanlar, hayvanlar, doğa manzaraları ve tıbbi görüntüler gibi gerçekçi ve karmaşık verilerin kullanım durumları daha önemli ve faydalı olmakla birlikte bu verilerin gerçek kaynaklardan toplanması daha maliyetlidir ve Kişisel Verilerin Korunması Kanunu gibi nedenlerle aynı zamanda daha zordur. Rotasyon, taşıma ve gürültü ekleme gibi veri iyileştirme yöntemleriyle az sayıda bulunan gerçek verilerin sayısı suni bir şekilde artırılabilir. Bu işlemlerin sonucunda üretilen görseller farklı olmalarına rağmen gerçek verinin temsil ettiği doğal dağılıma katkıda bulunmamaktadırlar. Bu nedenle, eğitilen makine öğrenmesi algoritmasının yalnızca gerçek veriyi ezberlemesini yavaşlatmaktadırlar (Shorten ve Khoshgoftaar, 2019). Bu bağlamda, gerçek veriye benzeyen ve onun özelliklerini taşıyan sentetik verilerin üretimine duyulan ihtiyaç artmaktadır. Daha fazla alanda fayda sağlayabilmeleri için üretilecek verinin görsel özellikleri kontrol edilebilir olmalıdır.

Derin öğrenmenin gelişmesiyle beraber veri üretiminde de derin ağların kullanımı üretici ağlar ile artmıştır (Chlap vd., 2021). Otokodlayıcılar veri örnekleri üzerinden veriyi tekrar üretmeyi öğrenirler ve aynı zamanda veriler arasında doğrusal bir geçiş sağlayabilmektedirler. Otokodlayıcılar, üretilen çıktının boyutu arttıkça daha bulanık çıktılar ürettikleri için yüksek çözünürlüklü veri üretiminde farklı bir yaklaşım olarak Çekişmeli Üretici Ağlar (GAN) (Goodfellow vd., 2014) sunulmuştur. GAN'lar, yakın zamanda, yapısındaki üretici ağlar ile oldukça gerçekçi çıktılar üretebilecek duruma gelmişlerdir. Eğitim sırasında veri örneklerini tekrar üretmek yerine gerçek veri olarak sınıflandırılabilir veri üretmeyi hedeflemektedirler. Böylece daha gerçekçi ve net çıktılar üretebilmektedirler. Üretilen veriler gerçek verilerin özneliliklerini kapsamalıdır ve onları gerçek bağlamlarında uygulamalıdır. Örneğin bir köpek resmi üretilirken iki göz, iki kulak ve bir ağız ve burun köpeği doğru tanımlamak için gerekli olacaktır fakat bunların

birbirine olan mesafeleri ve resim içindeki lokasyonları yanlış olursa bu sentetik veri başarısızdır. Kediler ve köpekler birer hayvan olsalar da bunların karışımı olan bir hayvan tanımsızdır ve sentetik verilerin arasında yer almamalıdır. Rastgele üretilen gerçekçi sentetik veriler fayda sağlayabilirken, bu verilerin kontrol edilebilmesi daha faydalı olacaktır. Az sayıda gerçek verinin olması, karmaşık yapılara sahip makine öğrenmesi modelleri ve sentetik verinin kalitesini ve doğruluğunu ölçen yöntemlerin eksikliği bu problemi zorlaştıran durumlardan bazılarıdır. Veri üretiminde etiketlenmiş veriye ihtiyaç duyan yöntemler ise etiketleme maliyeti yüzünden kısıtlanmaktadır.

Etiketli veri gerektirmeden eğitilebilen modellerin çıktıları gerçekçidir fakat bu modellerin çıktılarındaki görsel özellikleri istenilen biçimde yönlendirme problemi henüz kusursuz bir çözüme ulaşmamış ve zorlu bir problemdir. Bu problemin kaynağı, üretici ağın gizli uzayındaki eksenlerin iç içe geçmiş biçimde kodlanmış olmasıdır. Bu eksenler veriye ait anlamsal özellikleri ifade etmektedirler. Diğer bir deyişle, veriyi temsil ederken kullanılan gizli vektörün (z) eksenleriyle, veriyi oluşturan modelin kodladığı anlamsal özellikleri temsil eden eksenlerin uyumlu olmamasıdır. Üreticinin gizli uzayındaki anlamsal yönün tespit edilmesi, veriyi anlamsal olarak değiştirme işlemini mümkün kılmaktadır. Bu işlemin tersi olan ve Ayrıştırma (Disentanglement) olarak adlandırılan yöntem, gizli uzaydaki eksenleri ayrıştırarak onları anlamsal özelliklerle hizalamaktadır (Bengio vd., 2013).

Başarılı bir şekilde nitelikli sentetik verinin üretimi, mevcut yöntemlerin daha çok veriyle kullanılabilmesi sayesinde birçok alana katkı sağlayabilmektedir. Sentetik verilerin gerçek verilere benzemelerine rağmen gerçek olmamaları ise onları yasal engellerden ve yüksek maliyetlerden bağımsızlaştırmaktadır. Var olan gerçek bir veri kümesini baz alarak oluşturulan sentetik veriler bu veri kümesine katılarak ilgili veri kümesi iyileştirilebilir. Aynı zamanda az sayıda veri örneği tarafından temsil edilen öznel özelliklerin sayısını artırarak bu veri kümesinin dağılımı iyileştirilebilir. Basitleştirilmiş bir arayüze sahip olan üretim yöntemleri insan makine etkileşiminde büyük bir rol oynayarak, insanın sonsuz yaratıcılığının gerçekleşmesine yardımcı olabilmektedir (Ramesh vd., 2022).

1.1 Tezin Amacı

Bu tez, Görsel Özellik Ayırıştırma probleminin çözümüne katkı sağlamayı amaçlamaktadır. Bu tez çalışmasında, GAN eğitimi farklı stratejiler kullanılmadan, halihazırda özellik ayırıştırma yapılmadan, eğitilmiş bir modelin gizli uzayının analiz edilmesi ve somut referans veriler ile hedeflenen görselleri üretebilen bir yaklaşım sunulmaktadır.

1.2 Tezin Ana Katkısı

Tez kapsamında önerilen yaklaşım, ayırıştırılmış GAN eğitimi veya etiketlenmiş veri gerektirmemektedir. Bu yaklaşım önceden eğitilmiş üretici modeller ile kullanıma uygundur. Bu çalışma, üretici ağı anlamsal eksenlerini öz denetimli yöntemlerle otomatik olarak belirlemeyi hedefleyen çalışmalara (Creswell ve Bharath, 2019; Dogan ve Keles, 2020, 2022; Lipton ve Tripathi, 2017) yakın bir konumda yer almaktadır. Bu yaklaşımın öncü sonuçları SIU 2022 konferansında paylaşılmıştır (Aktaş vd., 2022). Bu bağlamda çalışmamızın sağladığı ana katkılar şu şekilde özetlenebilir:

- GAN'ların gizli uzay analizi için farklı bir yaklaşım olan Yön-Ayrıştırıcı model önerilmektedir. Yön-Ayrıştırıcı model, GAN ile üretilen görselleri analiz ederek, içerdikleri anlamsal özelliklerin arasındaki farkı GAN'ın gizli uzayında temsil eden yön vektörünü tahmin edebilmektedir.
- Öz denetimli bir yöntem geliştirip, maliyeti yüksek olan etiketlenmiş veri gerektirmeyen bir kurgu önerilmektedir.
- Ön eğitilmiş GAN'ların özelleştirilmiş eğitim stratejileri ile tekrar eğitilmelerine gerek kalmadan kullanılabilirlikleri artırılabilir.
- İki yaygın veri kümesi (MNIST ve FreyFaces) ile sonuçlar elde edilerek, bulunan yön vektörü GAN'ın girdi vektörüne eklenerek, birbirinden bağımsız görsellerin kimliğini bozmadan, hedeflenen biçimde özellik verilebildiği gösterilmiştir.

2. KURAMSAL TEMELLER VE KAYNAK ÖZETLERİ

2.1 Üretici Sinir Ağları

Makine öğrenmesinin birçok alanı bir veri kümesinin özelliklerini öğrenerek o veri üzerinde sınıflandırma veya yakınsama gibi işlemler yapmayı hedefler. Üretici algoritmalar ise mevcut verinin dağılımını öğrenerek onun özünü taşıyan fakat veri kümesinde var olmayan özgün veri örnekleri üretmeyi hedeflerler.

2.1.1 Otokodlayıcı

Bir otokodlayıcı (Autoencoder) girdisini tekrar üretmeyi hedefleyen bir yapay sinir ağıdır. Bunu yaparken faydalı temsiller öğrenmeyi hedeflemektedir. Girdi vektörü (x) ilk olarak gizli bir temsile (z) eşleştirilmektedir ($z = f(x)$). f fonksiyonu, kodlayıcı ağ (encoder) tarafından bir veya daha fazla doğrusal olmayan katman ile parametreleştirilmektedir. Ardından gizli temsil (z) çıktıya ($\hat{x}$) kod çözücü fonksiyonunu (g) parametreleştiren kod çözücü ağ (decoder) tarafından eşleştirilmektedir ($\hat{x} = g(z)$). Kodlayıcı ağ gibi, kod çözücü ağ bir veya birden fazla doğrusal olmayan katmandan oluşabilmektedir. Kodlayıcı ve kod çözücü ağların arasında gizli temsil z 'yi içeren ve genelde tek bir katmandan oluşan darboğaz bulunmaktadır. Parametreler Denklem 2.1'de verilen kayıp fonksiyonunu minimize edecek şekilde optimize edilmektedirler.

$$L(x, g(f(x))) = ||\hat{x} - x||_2^2 \quad (2.1)$$

Denklem 2.1'deki kayıp fonksiyonu farklı kısıtlamalar eklenerek optimize edilebilmektedir. Bu kısıtlar otokodlayıcının mimarisine (örneğin darboğazın boyutu) ve hedef fonksiyonuna (örneğin ek bir kısıtlayıcı terim) uygulanabilmektedirler. Şekil 2.1'de örnek bir otokodlayıcı mimarisi gösterilmiştir.

Darboğaz ile birlikte gizli temsilin boyutu girdinin boyutundan daha küçük tutularak kısıtlanması, otokodlayıcının daha faydalı temsilleri öğrenmesini sağlayabilmektedir. Böylece otokodlayıcının, verinin belirgin ve çok anlam taşıyan özniteliklerini öğrenmesi sağlanmaktadır. Doğrusal katmanlardan oluşan ve ortalama kare hatası kriterini kullanan bir otokodlayıcının mimarisindeki gizli katmanların PCA algoritmasına eşit oldukları

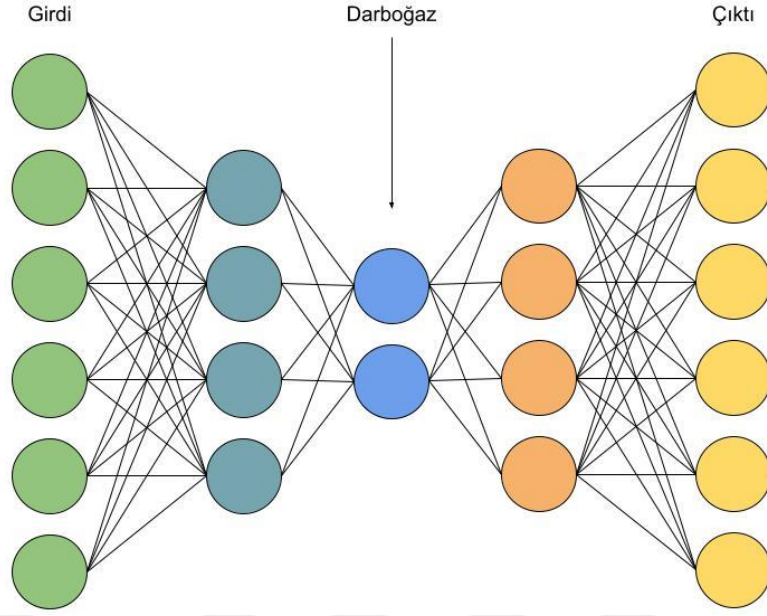
gösterilmiştir (Vincent vd., 2010). Doğrusal olmayan katmanlar kullanıldığında ise otokodlayıcının kapasitesi kayda değer seviyede artmaktadır ve daha faydalı temsiller öğrenebilmektedir (Hinton ve Salakhutdinov, 2006).

Girdiye stokastik değişimler uygulanarak otokodlayıcının farklı bir yöntemle faydalı temsiller öğrenmesi sağlanabilmektedir (Vincent vd., 2008, 2010). Eğitim sırasında girdiye gürültü eklenerek otokodlayıcının aynı girdinin gürültüsüz halini üretmesi hedeflenmektedir. Bu otokodlayıcının kayıp fonksiyonu Denklem 2.2’de verilmiştir.

$$L(x, g(f(\tilde{x}))) = ||\hat{x} - x||_2^2 \quad (2.2)$$

Burada $\tilde{x}$, x ’in bir çeşit gürültü eklenen bir kopyasıdır. Otokodlayıcı bu gürültüyü düzeltmeyi öğrenirken aynı zamanda gürültüden etkilenmeyen daha soyut temsilleri öğrenebilmektedir. Bu şekil öğrenilen gürbüz temsillerin verinin dağılımını temsil ettikleri için veri üretiminde kullanılabildikleri gösterilmiştir (Alain ve Bengio, 2013).

Otokodlayıcıların veri üretimi dışında birçok farklı kullanım senaryoları bulunmaktadır. Denetimsiz biçimde eğitilen otokodlayıcıların öğrendiği özneliliklerden faydalanılarak az sayıda etiketli veri ile denetimli eğitim gerçekleştirilebilmektedir. Yarı denetimli olarak adlandırılan bu yöntem etiketli veriye olan ihtiyacı azaltmaktadır. Bir çalışma denetimsiz öğrenilen öznelilikleri bir sınıflandırıcının girdileri olarak kullanmaktır (Gogna ve Majumdar, 2016). Farklı bir kullanım alanı ise düşük boyutlu gizli kodu kullanarak veriyi kümelemektir. Bu şekilde veriye göre daha düşük boyutlarda ve hızlı bir biçimde benzerlik ölçülebilmektedir (Hinton ve Salakhutdinov, 2006).



Şekil 2.1 Üç gizli katmana sahip ve darboğazı boyutsal olarak kısıtlanmış bir otokodlayıcının örnek mimarisi

2.1.2 Değişimsel otokodlayıcı

Değişimsel otokodlayıcılar (Variational Autoencoder, VAE) (Kingma ve Welling, 2014), otokodlayıcıların bir türevidir. Temel otokodlayıcılardan farklı olarak değişimsel otokodlayıcıların darboğazı stokastiktir. Girdi vektörü (x) tek bir gizli temsile eşleştirilmek yerine bir Gauss dağılımı ifade eden ortalama (μ) ve standart sapma (σ) parametrelerine dönüştürülmektedir. Ağırlıkları ve biasları θ olan kodlayıcı ağın çıktısı $f_{\theta}(z|x)$ olarak ifade edilmektedir. Bu dağılımdan örnekleme yapılarak gürültülü z değerleri elde edilebilmektedir. Kod çözücü ağı ise ϕ olan ağırlıklara ve biaslara sahiptir ve gizli temsil z 'yi girdi olarak almaktadır $g_{\phi}(x|z)$. Gizli temsilin boyutu girdinin boyutundan düşük olduğu için bir veri kaybı söz konusudur. Bu kaybı ölçmek için tekrar üretim log olasılığı (reconstruction log-likelihood) $\log(g_{\phi}(x|z))$ kullanılmaktadır. Bu metrik kod çözücü ağın girdi x 'i gizli temsil z 'yi kullanarak ne kadar başarılı bir şekilde tekrar üretebildiğini ölçmektedir. Şekil 2.2'de örnek bir değişimsel otokodlayıcı mimarisi verilmiştir.

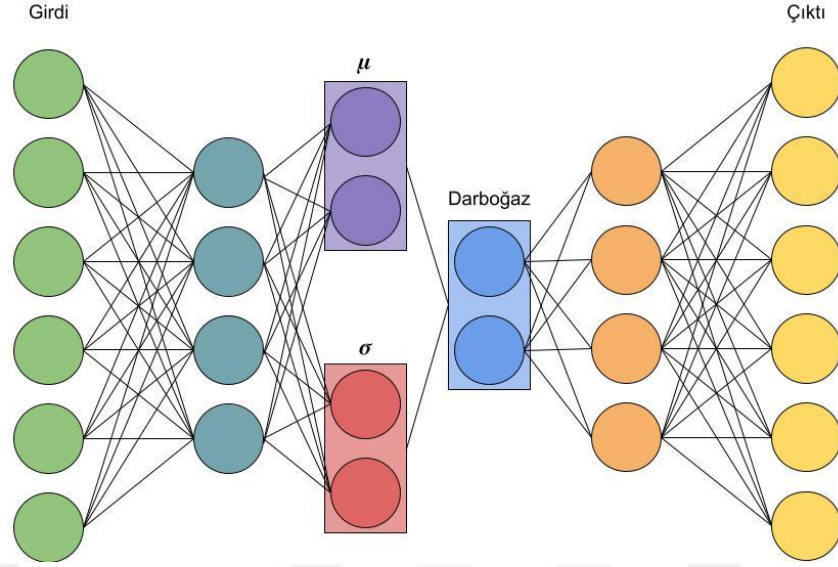
Değişimsel otokodlayıcının kayıp fonksiyonu düzenleyici eklenmiş negatif log olasılığı ile ifade edilmektedir. Bütün veri örneklerini kapsayan bir temsil olmadığı için kayıp

fonksiyonu her bir veri örneği için bölünerek l_i olarak ifade edilmektedir. Bu kayıpların toplamı N veri örneği için $\sum_{i=1}^N l_i$ olarak verilmiştir. Bir veri örneği x_i için kullanılan kayıp fonksiyonu Denklem 2.3’de verilmiştir.

$$l_i(\theta, \phi) = -\mathbb{E}_{f_{\theta}(z|x_i)}[\log(g_{\phi}(x_i|z))] + D_{KL}(f_{\theta}(z|x_i)||g(z)) \quad (2.3)$$

Denklemin ilk terimi tekrar üretim kaybıdır, yani i' inci veri örneği için beklenen negatif log olasılığıdır. Bu beklenti kodlayıcının temsiller üzerinde olan dağılımına ilişkindir. Bu terim kod çözücüyü veriyi tekrar üretmeye teşvik etmek için kullanılmaktadır. Başarısız tekrar üretimde kayıp yükselmektedir. İkinci terim bir düzenleyicidir. Kodlayıcının dağılımı $f_{\theta}(z|x)$ ve $g(z)$ arasındaki Kullback-Leibner uzaklaşmadır (Kullback-Leibner Divergence). Bu metrik f ile g ’yi temsil ederken oluşan bilgi kaybını ölçmektedir. Diğer bir deyişle f ’in g ’ye olan uzaklığını ifade etmektedir.

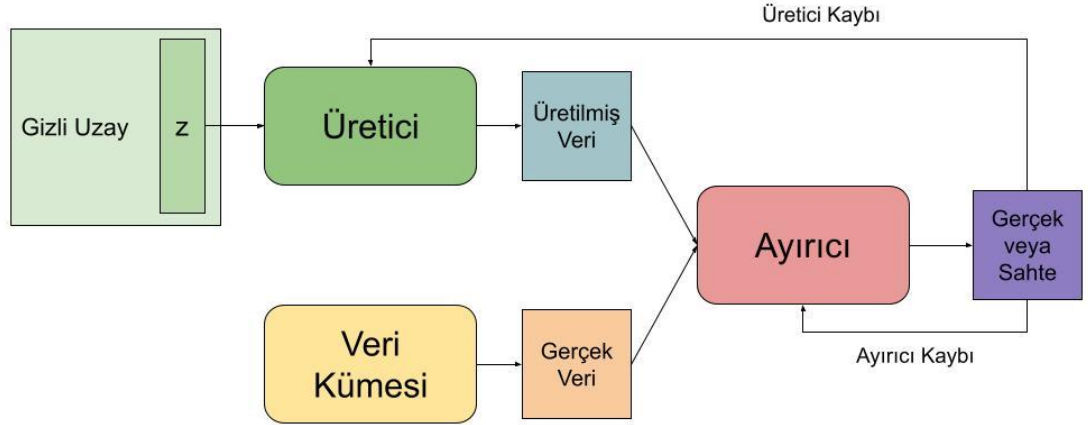
Değişimsel otokodlayıcılarda g ortalaması sıfır ve varyansı bir olan standart normal dağılım olarak tanımlanmıştır $g(z) = Normal(0,1)$. Kodlayıcının normal dağılıma uygun olmayan gizli temsiller üretmesi durumunda kayıp artmaktadır. Bunun amacı farklı girdilerin z ’lerini Öklid uzayında birbirinden olabildiğince uzak tutmaktır. Böylece benzer girdilerin temsillerinin birbirlerine yakın olmaları sağlanmaktadır.



Şekil 2.2 İki μ ve σ olmak üzere dört gizli katmana sahip değişimsel otokodlayıcının örnek mimarisi

2.1.3 Çekişmeli üretici sinir ağları (GAN)

Çekişmeli Üretici Ağlar (Generative Adversarial Networks, GAN) (Goodfellow vd., 2014) diğer bir üretici model yaklaşımıdır. GAN'ların temel fikri bir oyun teorisi senaryosundan türetilmiştir. Bu çekişmeli senaryoda üretici ağ ve ayrıştırıcı ağ olan iki farklı yapay sinir ağı bulunmaktadır. Bu iki ağ birbirine karşı yarışmaktadır. Üretici ağ, gizli temsil vektörü (z) ve kendi ağırlıklarını ve biaslarını (θ_g) kullanarak veri örnekleri üretmektedir ($x = g(z; \theta_g)$) ve bir otokodlayıcının kod çözücü kısmına benzemektedir. Rakibi olan ayrıştırıcı ağ ise gerçek verilerden oluşan veri kümesine ait ve üretici ağ tarafından üretilen verileri ayırt etmeyi hedeflemektedir. Ayrıştırıcı ağ bir sınıflandırıcıdır ve x 'in gerçek bir veri örneği olma olasılığını gösteren değeri üretmektedir ($d(x; \theta_d)$). Şekil 2.3'de örnek bir GAN mimarisi gösterilmiştir.



Şekil 2.3 Çekişmeli üretici ağ mimarisi

Çekişmeli üretici ağların öğrenimleri, toplamı sıfır olan bir oyun (zero-sum game) olarak ifade edilebilmektedir. Bu bağlamda ayrıştırıcı ağın kazancı $v(\theta_g, \theta_d)$ fonksiyonu ile belirlenmektedir. Üretici ağın kazancı ise $-v(\theta_g, \theta_d)$ olarak belirlenmektedir. Eğitim sırasında her iki oyuncu kazançlarını maksimize etmeyi amaçlamaktadır $L = \min \max(v(g, d))$.

$$v(\theta_g, \theta_d) = \mathbb{E}_{x \sim \text{veri}} \log(d(x)) + \mathbb{E}_{x \sim \text{model}} \log(1 - d(x)) \quad (2.4)$$

Denklem 2.4'te verilen fonksiyon GAN'ların bütün eğitim veri kümesi üzerinde hesaplanan kayıp fonksiyonudur. Bu kayıp, ayrıştırıcı ağın veri örneklerinin gerçek veya sahte olarak sınıflandırmayı öğrenmesini sağlamaktadır. Aynı zamanda üretici ağ ürettiği veri örneklerinin ayrıştırıcı ağ tarafından gerçek veri olarak sınıflandırılmasını sağlamayı hedeflemektedir. Yakınsamada üretici ağın ürettiği veri örnekleri, gerçek örneklerden ayırt edilemeyecek kadar gerçekçidir ve ayırıcı ağ bütün veri örnekleri için $\frac{1}{2}$ olasılığını üretmektedir.

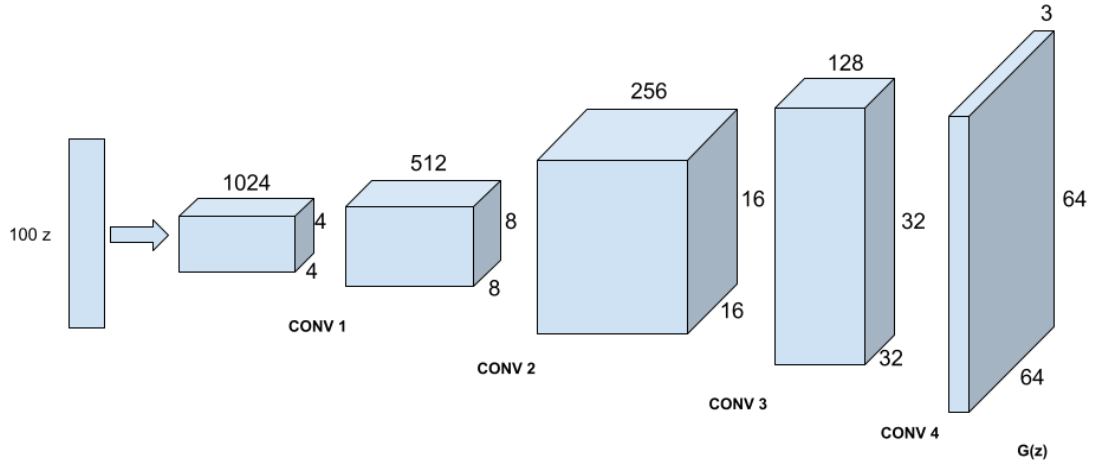
GAN'larda öğrenme, g ve d sinir ağları tarafından temsil edildiğinde ve $\max_d v(g, d)$ dışbükey olmadığından pratikte zor olabilmektedir. Yakınsama, GAN'ların yetersiz öğrenmelerine neden olabilecek bir sorun olarak tanımlanmıştır (Goodfellow vd., 2014). Genel olarak, iki oyuncunun kayıplarının eşzamanlı gradyan inişi ile bir dengeye ulaşmaları garanti edilememektedir. GAN eğitimini zorlaştıran bir örnek olarak bir

oyuncunun a 'yı kontrol ettiği ve ab kaybına uğradığı, diğer bir oyuncunun b 'yi kontrol ettiği ve $-ab$ kaybına uğradığı $v(a, b) = ab$ fonksiyonu örnek verilebilir. Her oyuncunun sonsuz derecede küçük gradyan adımları attığı düşünüldüğünde, her oyuncu diğer oyuncunun zararına kendi kaybını düşürmektedir. Bu durumda a ve b başlangıç noktasındaki denge noktasına varmak yerine sabit, dairesel bir yörüngeye girmektedirler. İki oyuncunun aynı anda minimum ve maksimum sağlamaya çalıştıkları oyunda, oyunun dengeleri v 'nin yerel minimumları değildir. Bunun yerine, her iki oyuncunun kayıpları için aynı anda minimum olan noktalardır. Bu, birinci oyuncunun parametrelerine göre yerel minimum ve ikinci oyuncunun parametrelerine göre yerel maksimum olan v 'nin eyer noktaları oldukları anlamına gelmektedir. Oyuncuların v 'yi eyer noktasına varmadan sonsuz sayıda sırayla artırıp azaltmaları mümkündür. Bu yakınsama sorununun GAN'ları ne ölçüde etkilediği bilinmemektedir.

Gerçekçi deneylerde üretici ağ için en iyi başarı elde eden kayıp fonksiyonu, ayrıştırıcı ağın doğru sınıflandırma yapmasının logaritmik olasılığını düşürmek yerine, yanlış sınıflandırma yapmasının olasılığını arttırmak olarak belirlenmiştir (Goodfellow vd., 2014). Bu reformülasyonun temelindeki motivasyon ise ayrıştırıcı ağın yüksek güven ile bütün verileri sahte olarak sınıflandırması durumunda bile üretici ağın kayıp fonksiyonunun türevinin yine de yüksek değerler ürettiği olmasıdır. Eğitim sırasında bu senaryo özellikle ayrıştırıcı ağın üretici ağdan daha hızlı öğrenmesi durumunda meydana gelmektedir ve üretici ağın kayıp fonksiyonu tarafından üretilen düşük değerli gradyanlar geri yayılımda etkisiz olmaktadır. GAN eğitiminin stabilize edilmesi henüz çözülmemiş bir problemdir fakat model mimarilerin ve hiperparametrelerin doğru seçilmeleri büyük bir etkidir.

Gizli uzay ise, doğrudan yorum yapamadığımız, ancak dışarıdan gözlemlenen olayların anlamlı bir iç temsilini kodlayan özellik değerlerini içeren soyut, çok boyutlu ve devamlı bir alanı ifade etmektedir. Gizli uzayda üreticinin üretebileceği bütün verileri temsil eden bir gizli vektör vardır. Aynı zamanda gizli uzaydan seçilebilen her bir gizli vektör için veri üretilmektedir.

Zaman içinde GAN mimarisine değişiklikler uygulanarak farklı dallar türetilmiştir. Bu bağlamda bir çalışmada sunulan derin evrimsel GAN (DCGAN) mimarisi görsel veri



Şekil 2.4 LSUN veri kümesiyle kullanılan DCGAN mimarisi

üretiminde başarı sergilemiştir ve modelin gizli uzayında özneliklerin önemli varyasyonların ifade edildiğini göstermiştir (Radford vd., 2016). İlk GAN'lar doğrusal katmanlardan oluşurken DCGAN evrimsel katmanlardan oluşmaktadır. Bu sayede üretici ağı kendi uzamsal üst örneklemesini (upsampling) öğrenmesine izin verilmiştir. Üretici modele girdi olarak sağlanan gizli temsil vektörü (z) 4x4'lük bir vektör olarak şekillendirilen işlem evrimsel katmanların ilk aşamasını belirtmektedir. Ardından dört adet fraksiyonel adımlı (fractionally strided) evrimsel katmandan geçirilmektedir. Bunun sonucunda 64x64 çözünürlüğüne sahip olan bir görsel oluşmaktadır. Burada kritik olan nokta, hiçbir doğrusal katmanın kullanılmamasıdır. LSUN veri kümesiyle eğitilmek üzere kullanılan DCGAN mimarisi Şekil 2.4'de gösterilmiştir. Bu mimaride gizli vektörün boyutu 100, üretilen görselin ise boyutu 64x64 olarak seçilmiştir. GAN ve DCGAN'ın eğitiminin farklı aşamalarında üretilmiş örnek MNIST çıktılarının karşılaştırılması Şekil 2.5'de verilmiştir.



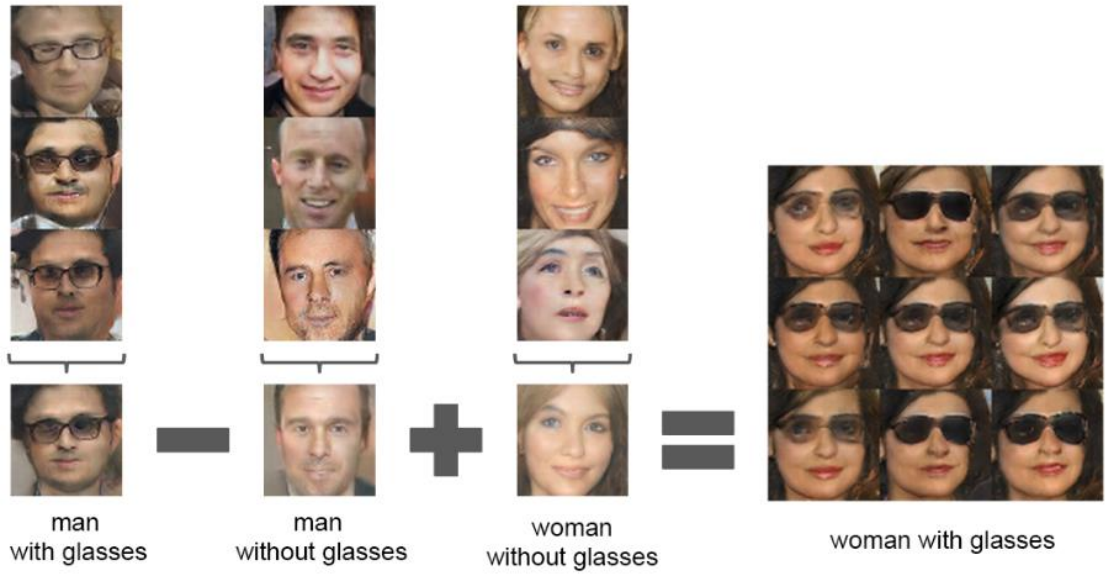
Şekil 2.5 GAN (sol) ve DCGAN (sağ) ile eğitimin farklı aşamalarında oluşturulan MNIST veri örnekleri (Radford vd., 2016)



Şekil 2.6 Yüz verileri ile eğitilmiş DCGAN'ın örnek çıktıları

Yüz verileri ile eğitilmiş DCGAN'ın örnek çıktıları ise

Şekil 2.6'da verilmiştir. DCGAN'ın gizli uzayına ait vektörler üzerinde aritmetik işlemler uygulanarak oluşan çıktının özellikleri değiştirilebilmektedir. Bu işlem manuel olarak farklı çıktıların gizli vektörleri kullanılarak elde edilebilmektedir. “Gözlüklü erkek” ifade eden gizli vektörden “gözlüksüz erkek” gizli vektörü çıkarılıp “gözlüksüz kadın” vektörü eklenilerek “gözlüklü kadın” vektörü elde edilebilmektedir. Bu işlemin örneği Şekil 2.7’de verilmiştir. Bu işlemin sonucunda tutarlı çıktıların oluşması, DCGAN'ın gizli uzayında aritmetik operasyonların geçerliliğini kanıtlamıştır.



Şekil 2.7 DCGAN'ın gizli uzayına ait vektörler üzerinde aritmetik işlemler uygulanarak oluşan çıktının özelliklerin değiştirilmesi

GAN'lardan türeyen farklı bir mimari ise InfoGAN'dır (Chen vd., 2016). GAN formülasyonu, basit çarpanlara ayrılmış, sürekli bir gürültü vektörünü (z) girdi olarak kullanır ve gürültüyü kullanma biçimine ilişkin kısıtlamalar empoze etmemektedir. Sonuç olarak, gürültü üretici tarafından oldukça karışık bir şekilde kullanılabilinmektedir. Bu durum, z 'nin boyutlarının verinin anlamsal boyutlarına karşılık gelmemesine neden olabilmektedir.

InfoGAN'ın varsayımlarından birisi, birçok alanda verilerin doğal olarak varyasyon içeren bir takım anlamsal çarpanlara bölünebilmesidir. Örneğin, MNIST veri kümesinden yeni veriler üretirken, üretici modelin üretilen rakamın sayısını temsil etmek için bir tane ayrık (0-9), ve rakamı oluşturan çizginin kalınlığı ve açısı için birer tane sürekli rastgele değişken tahsis etmesi ideal olacaktır. Bir MNIST verisinin bu şekilde ifade edilebilen üç adet özniteliklerden oluştuğunu belirterek, değişkenlerin denetimsiz bir biçimde bu kurguya uymalarının saplanması faydalı olacaktır.

Bu bağlamda, yapılandırılmamış tek bir gürültü vektörü kullanmak yerine girdi iki parçaya bölünmüştür. Bu vektörlerden ilki, sıkıştırılmaz gürültü kaynağı olarak değerlendirilen z vektörüdür. Diğeri ise, gizli kod (latent code) olarak adlandırılan ve veri dağılımının göze çarpan, yapılandırılmış semantik özelliklerini ifade etmeyi hedefleyen c vektörüdür.

Matematiksel olarak yapılandırılmış gizli değişkenlerin kümesi $c_1, c_2, \dots, c_L$ olarak belirtilmektedir. En basit haliyle, Denklem 2.5'teki gibi çarpanlara ayrılmış bir dağılım varsayılabilir.

$$P(c_1, c_2, \dots, c_L) = \prod_{i=1}^L P(c_i) \quad (2.5)$$

Bu gizli faktörlerin denetimsiz bir şekilde keşfedilmesi için bir yöntem önerilmektedir. InfoGAN'ın geleneksel GAN'dan farkı, üretici modelin sıkıştırılamaz gürültü vektörü z 'nin yanında aynı zamanda gizli kod vektörü c 'yi de girdi olarak kabul etmesidir. Böylece üretici modelin formu $G(z, c)$ olarak ifade edilmektedir. Ancak, standart GAN'da, üretici model Denklem 2.6'yı sağlayan bir çözüm bularak ek olan gizli kodu göz ardı edebilmektedir.

$$P_G(x|c) = P_G(x) \quad (2.6)$$

Anlamsız kodlarla başa çıkmak için bilgi-teorik bir düzenleme önerilmektedir. Gizli kodları c ve üretici dağılımı $G(z, c)$ arasında yüksek bilgi ortaklığı (high mutual information) sağlanmalıdır. Diğer bir deyişle, $I(c; G(z, c))$ yüksek olmalıdır.

Bilgi teorisinde, X ve Y arasındaki ortak bilgi (mutual information) $I(X; Y)$, Y rastgele değişkeni hakkında bilinenlerden X rastgele değişkeni hakkında öğrenilen bilgi miktarını ölçmektedir. Ortak bilgi Denklem 2.7'de verilen denklemde gösterildiği üzere iki entropi terimi arasındaki fark olarak ifade edilebilmektedir.

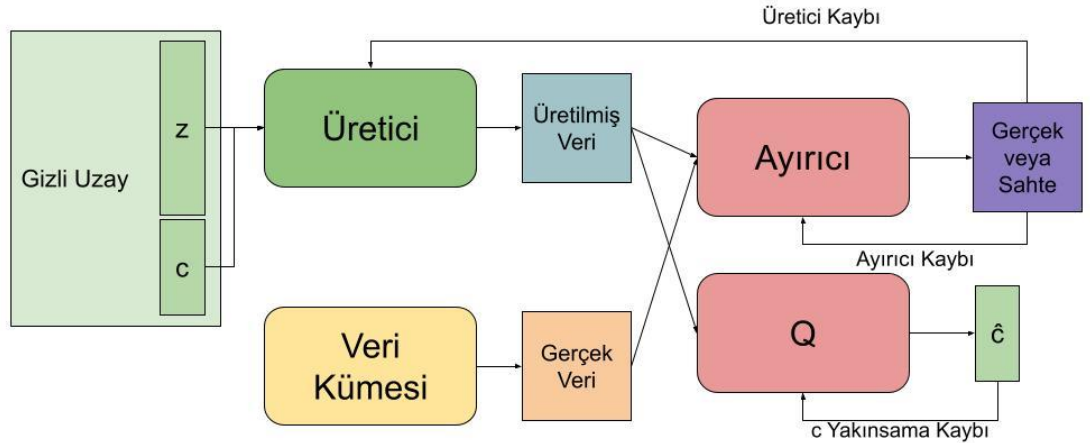
$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) \quad (2.7)$$

Bu tanım sezgisel bir biçimde yorumlanabilmektedir. $I(X; Y)$, Y gözlemlendiğinde X 'teki belirsizliğin azalmasıdır. X ve Y bağımsız olduğunda, $I(X; Y) = 0$ olmaktadır, çünkü bir değişken hakkındaki bilgi diğeri hakkında hiçbir bilgi vermemektedir. Aksine, X ve Y deterministik ve tersine çevrilebilir bir fonksiyonla ilişkili olduğunda, maksimal ortak bilgi elde edilmektedir. Bu yorum, maliyeti formüle etmeyi kolaylaştırmaktadır. Herhangi bir $x \sim P_G(x)$ verildiğinde, $P_G(c|x)$ 'in düşük bir entropiye sahip olması hedeflenmektedir. Yani c gizli kodundaki bilgi üretimi sürecinde kaybolmamalıdır. InfoGAN'ın çözmeyi hedeflediği minimax oyunu Denklem 2.8'te verilmiştir.

$$\min_G \max_D V_I(D, G) = V(D, G) - \lambda I(c; G(z, c)) \quad (2.8)$$

Uygulamada, yardımcı dağılım (Q) bir sinir ağı olarak parametrelendirilmektedir. Ayırıcı ağın (D) ve Q 'nın evrimsel katmanları ortak olarak kullanılmaktadırlar. Koşullu dağılım $Q(c|x)$ için çıkış parametrelerine bağlı son bir doğrusal katman bulunmaktadır. Bu, InfoGAN'ın GAN'a yalnızca ihmal edilebilir bir hesaplama maliyeti eklediği anlamına gelmektedir. $L_I(G, Q)$ kaybının her zaman normal GAN hedeflerinden daha erken yakınsadığı gözlemlenmiştir. Böylece eğitim süresini uzatmamaktadır.

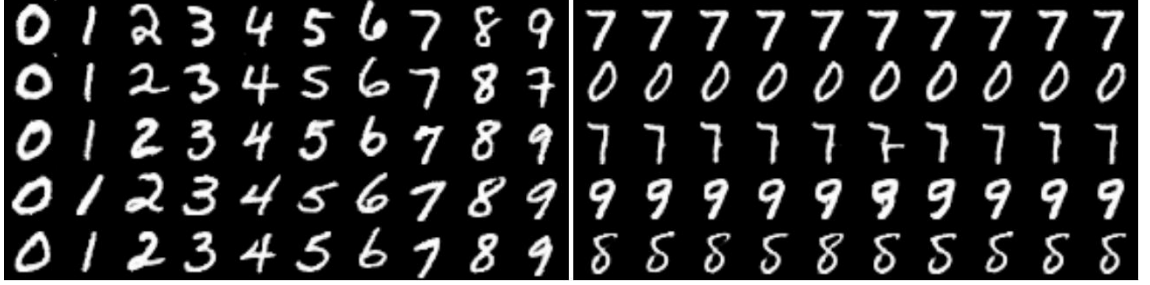
Kategorik gizli kod c_i için, $Q(c_i|x)$ 'i temsil etmek için softmax aktivasyonu kullanılmıştır. Devamlı gizli kod c_j için ise, $Q(c_j|x)$ bir Gauss dağılımı ile temsil edilmiştir. Eğitim sırasında DCGAN tarafından sunulan yöntemler kullanılmış ve yeni yöntemlerin geliştirilmesine ihtiyaç duyulmamıştır. InfoGAN ile yeni eklenen λ hiperparametresi ayrık gizli kodlar ile eğitirken 1 olarak kullanılırken, devamlı gizli kodlar ile eğitirken genelde 1'den küçük bir değer ile kullanılmaktadır. Böylece $\lambda L_I(G, Q)$ kayıp ölçeğinin GAN kaybıyla aynı ölçekte olması sağlanabilmektedir. InfoGAN'ın genel mimarisi Şekil 2.8'de verilmiştir.



Şekil 2.8 InfoGAN mimarisi

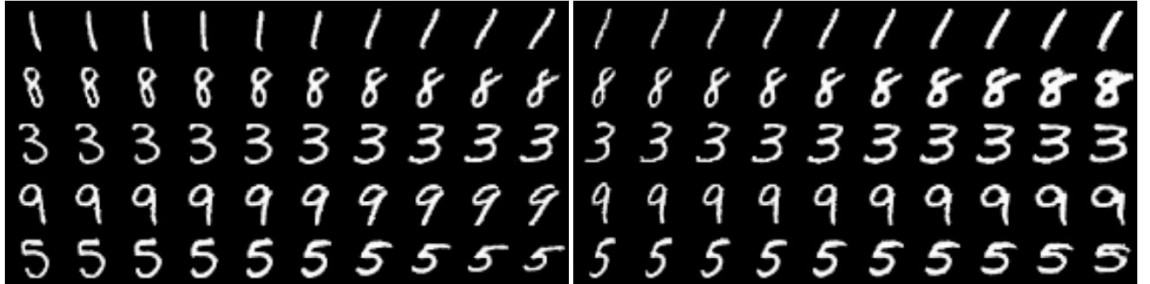
InfoGAN yazarları tarafından MNIST veri kümesi ile yapılan deneylerde, verideki ayrık değişimleri modellemesi için c_1 10 boyutlu kategorik gizli kod olarak kullanılmıştır. Verideki devamlı değişimleri modellemek için ise c_2 ve c_3 gizli kodları $(-1, 1)$ aralığında tekdüze dağılımında devamlı gizli kodlar olarak seçilmiştir. Ayrıca aynı gizli vektöre

sahip bir GAN eğitilmiş ve iki modelin girdilerine yapılan değişikliklerin çıktıya olan etkisi karşılaştırılmıştır. Şekil 2.9'da görüldüğü üzere InfoGAN'da farklı girdilerde c_1 'in değişmesiyle beraber çıktının rakamı çoğunlukla değişmektedir. Aynı girdi GAN'a verildiğinde ise çıktıda anlamlı bir değişim gözlemlenememektedir.



Şekil 2.9 InfoGAN ve GAN'da c_1 'e değişim uygulanması

InfoGAN eğitimde etiket kullanılmamasına rağmen c_1 'in değeri %5 hata payı olan bir sınıflandırıcı olarak kullanılabildiği tespit edilmiştir. Devamlı kodlar c_2 ve c_3 çıktının stilindeki devamlı değişimleri ifade etmektedir. Çıktının rotasyonu c_2 tarafından ve kalınlığı c_3 tarafından modellenmektedir. Bu kodlara değişim uygulandığında oluşan çıktılar Şekil 2.10'da verilmiştir.

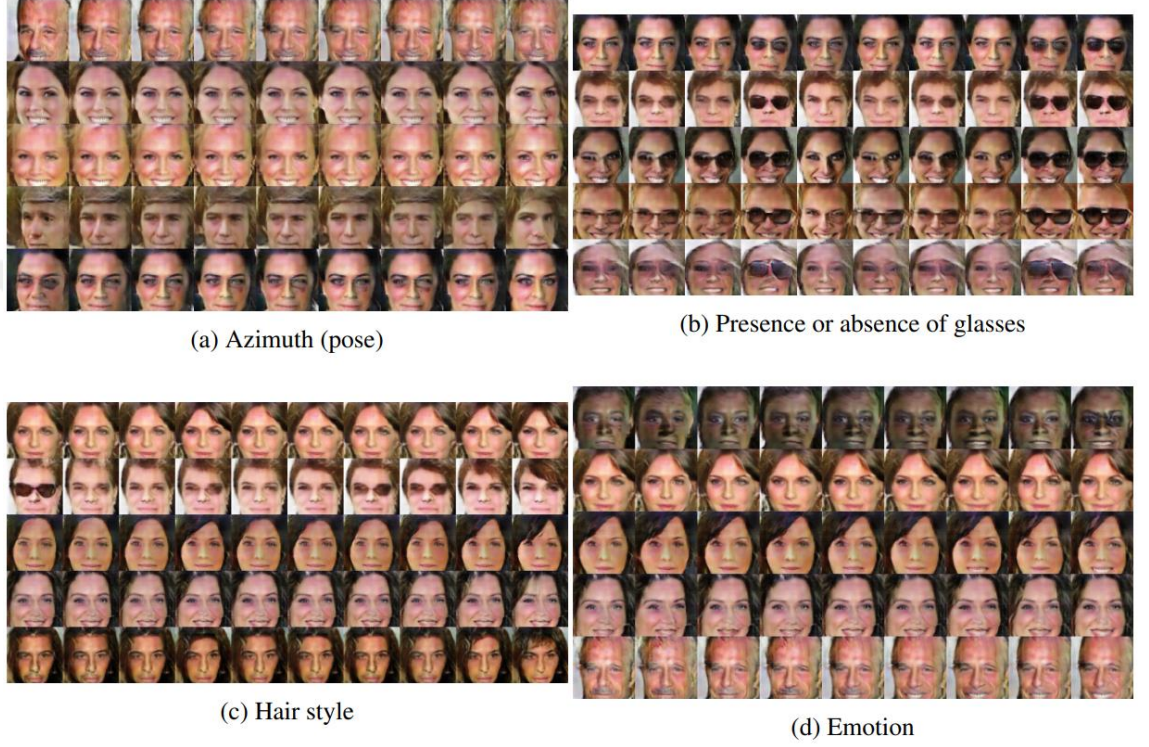


Şekil 2.10 InfoGAN'da c_2 ve c_3 kodlarına değişim uygulanması

Üretici model bu kodlar değiştiğinde çıktıdaki rakamı yalnızca döndürüp esnetmek yerine rakamın çizgi kalınlığını ve açısını değiştirerek çıktının gerçeğe yakın olmasını sağlayacak ayrıntılara önem verdiği dikkat çekmektedir.

Ayrıca InfoGAN ile daha zorlayıcı olan ve 200 bin ünlünün yüz resimlerinden oluşan CelebA (Z. Liu vd., 2015) veri kümesi üzerinde deneyler yapılmıştır. Bu veri kümesi için

gizli kod her biri 10 boyutlu olmak üzere 10 tane kategorik değişken olarak modellenmiştir. Veride hiçbir yüz birden fazla açıdan gösterilmemesine rağmen InfoGAN yüzlerin azimut (poz) bilgisini ayrıştırabilmiştir. Bunun dışında gözlüklerin varlığı, saç stili ve duygu gibi görsel bir anlayış gerektiren farklı anlamsal özellikleri ayrıştırabilmiştir. CelebA deneylerin örnekleri Şekil 2.11’de verilmiştir.



Şekil 2.11 InfoGAN ve CelebA ile yapılan deneyler

2.2 Ayrıştırma (Disentanglement)

Görüntü düzenleme veya üretme gibi birçok görsel işlem için görüntünün içeriklerini ve anlamsal özelliklerini belirlemek temel bir problemdir. Bu özellikler bir GAN’ın gizli uzayında karmaşık bir şekilde temsil edilmektedir. Bu uzaydaki anlamsal eksenleri tespit etmek ve onları birbirinden ayrıştırmak, onların daha kolay ve hedefli bir biçimde düzenlenmelerini ve kontrol edilmelerini sağlayabilmektedir.

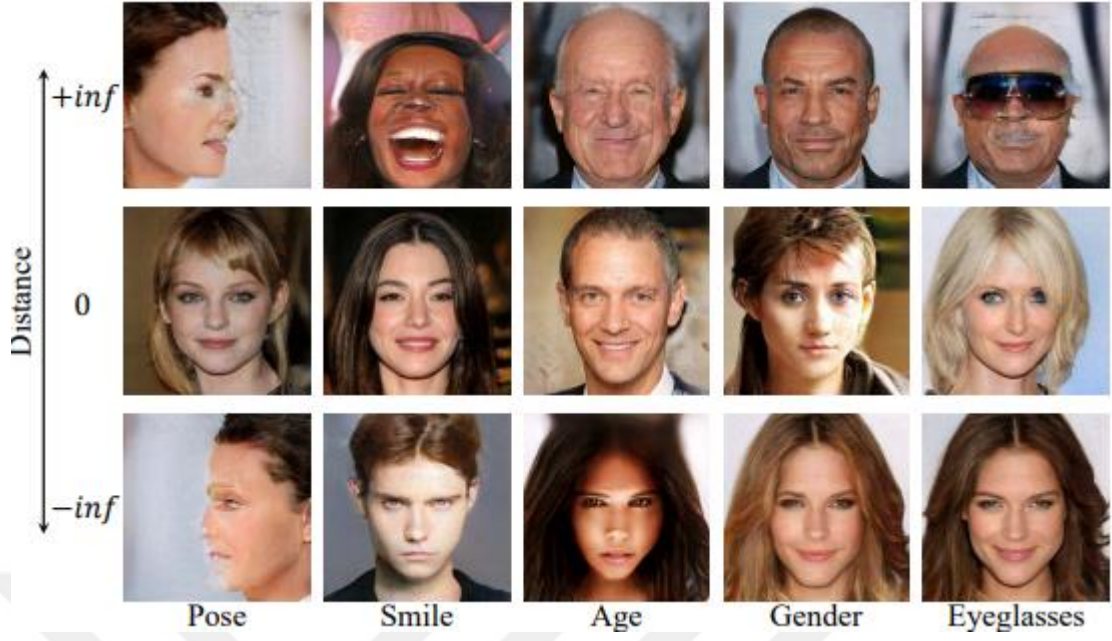
Daha önceki GAN modellerinin ortaya çıkmasından bu yana, GAN gizli uzayının genellikle anlamsal uzay aritmetiğine sahip olduğu bilinmektedir. Örneğin yüz görüntü

verileri için gülümseme veya gözlük eklemeye karşılık gelen yönler vardır (Radford vd., 2016) (Şekil 2.7). Bu yönlerin kullanılması görüntü düzenlemeyi kolaylaştıracağı için, gizli uzaydaki bu tür yönlerin keşfedilmesi şu anda çok fazla araştırmacının ilgisini çekmektedir.

Yakın tarihli bazı çalışmalar (Goetschalckx vd., 2019; Karras vd., 2019; Shen vd., 2020), gizli uzayda yorumlanabilir yönleri belirlemek için insan tarafından sağlanan etiketli veri kümeleri kullanarak denetimli yöntemler önermektedirler. Örneğin, (Karras vd., 2019; Shen vd., 2020), belirli yüz özelliklerini tahmin etmek için CelebA veri seti (Z. Liu vd., 2015) ile önceden eğitilmiş sınıflandırıcıları kullanmaktadır. Bu sınıflandırıcılar daha sonra oluşturulan görüntüler ve bunların gizli kodları için sahte etiketler üretmek için kullanılmaktadır. Bu etiketlere dayanarak, gizli uzayda ayırıcı düzlem oluşturulur ve bu düzlemin bir normali, karşılık gelen niteliği yakalayan bir yönü ifade eder. InterFaceGAN yöntemi (Shen vd., 2020) iyi eğitilmiş GAN'ların gizli uzaylarının ayrıştırılmış olduklarını temel almaktadır. Bu uzayda z_1 ve z_2 gizli vektörlerin arasında doğrusal geçişler sağlandığında çıktıdaki anlamsal özelliklerin de doğrusal bir biçimde değişim sağladıkları gözlemlenmiştir. Bu nedenle cinsiyet gibi anlamsal özellikler için gizli uzayda bu doğrusal hareketi tanımlayan hiper düzlemlerin bulunduklarını öne sürülmektedir. Hiper düzleminin normali n ve gizli vektör z ile, z 'nin hiper düzleme olan mesafesi d olarak verilmiştir (Denklem 2.9). Bu bağlamda d , negatif bir değere sahip olabilirdi için, tam anlamıyla bir uzaklık ölçüsü olarak tanımlanmamaktadır. Mesafe d 'nin sayısal işareti değiştiğinde anlamsal özellik tersine dönmektedir.

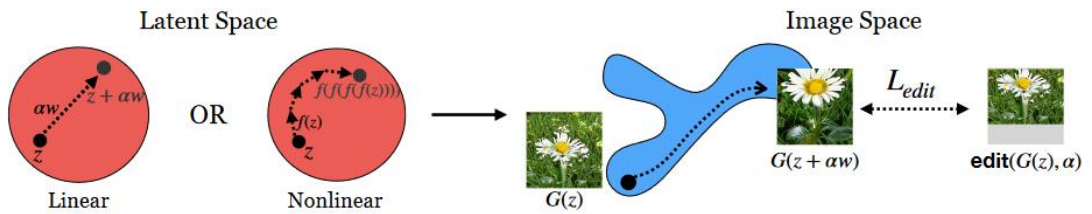
$$d(n, z) = n^T z \quad (2.9)$$

Anlamsal özelliklerin ilgili hiper düzlemlerine olan mesafeleri negatif yönde uzak, sıfır ve pozitif yönde uzak olan örnekler Şekil 2.12'de verilmiştir. Bu bağlamda bir özelliğin hiper düzlemine olan mesafesi sıfır iken, ilgili özellik uçların ortasındaki seviyede mevcuttur. Poz özelliği sağ ve sola bakan pozların arasında geçiş sağlamakta olduğu için, sıfır mesafeli poz düz bakmaktadır. Gülümseme özelliği ise kızgın ve güler yüzlü ifadeler arasında geçiş yapmakta olduğu için sıfır noktasında nötr bir yüz ifadesine tekabül etmektedir.



Şekil 2.12 InterFaceGAN ile üretilmiş örnekler (Shen vd., 2020)

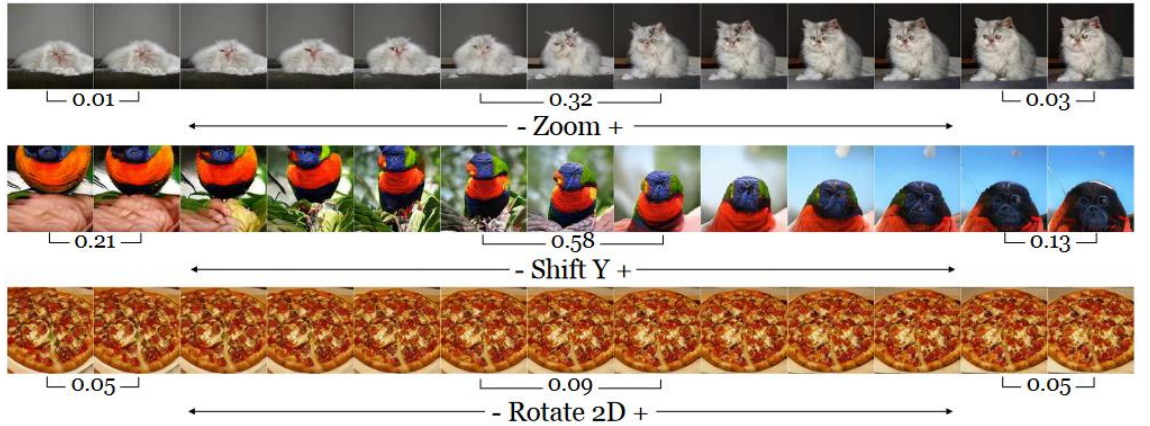
Yukarıdaki denetimli yaklaşımların en önemli zafiyeti, insan tarafından üretilen ve elde edilmesi maliyetli olabilen etiketlere veya önceden eğitilmiş modellere ihtiyaç duymalarıdır. İki çalışma (Jahanian vd., 2020; Plumerault vd., 2022) öz denetimli yaklaşımları kullanmaktadırlar ve yakınlştırma veya rotasyon gibi basit görüntü modifikasyonlarına karşılık gelen gizli uzaydaki vektörleri aramaktadırlar. Bu çalışmalardan GAN'ların yönetilebilirliğini araştıran çalışma (Jahanian vd., 2020), önceden eğitilmiş bir GAN'ın gizli uzayını keşfetmeyi hedeflemektedir. Şekil 2.13'de gösterilen yöntemle bir imaja uygulanan farklı değişimlerin, GAN uzayında hangi yönde ilerlenerek taklit edilebildiğini keşfetmektedirler.



Şekil 2.13 Gizli uzayda yön keşfi (Jahanian vd., 2020)

Orijinal gizli vektör z ile üretilen görsel $G(z)$ 'ye bir katsayı a ile ölçeklendirilen bir değişim uygulanarak $edit(G(z), a)$ görseli elde edilmektedir. Bu bağlamda uygulanan

değişim yakınlaştırma, çevirme veya keşfedilmesi istenen farklı bir işlem olabilmektedir. Aynı ölçek a ile GAN uzayında bir adım w atılır ve oluşan $G(z+aw)$ görseli oluşturulmaktadır. $G(z+aw)$ ve $edit(G(z), a)$ arasındaki L2 kaybı hesaplanarak GAN uzayındaki atılan adımın yönünün ne kadar doğru olduğu ölçülmektedir. Kayıp azaldıkça, gizli uzayda kullanılan yön, hedeflenen değişim operasyonunun etkisini yakınsamaktadır. Bu keşif işlemi tamamen öz denetimli bir biçimde gerçekleşmektedir. Bahsi geçen yöntem, gizli uzayda doğrusal keşifler için geçerlidir. Doğrusal olmayan keşifler için $f^*(z)$ fonksiyonu öğrenilir ve özyinelemeli biçimde uygulanarak adımlar atılmaktadır. Bu yöntem ile keşfedilen yönlerin görsellere uygulanarak oluşturulan örnekler Şekil 2.14’de verilmiştir. Yakınlaştırma, y eksenini kaydırma ve görseli çevirme operasyonları uygulanmıştır. Adım mesafesi w arttıkça, görseller değişmemeye veya gerçekçi olmamaya başlamaktadırlar.

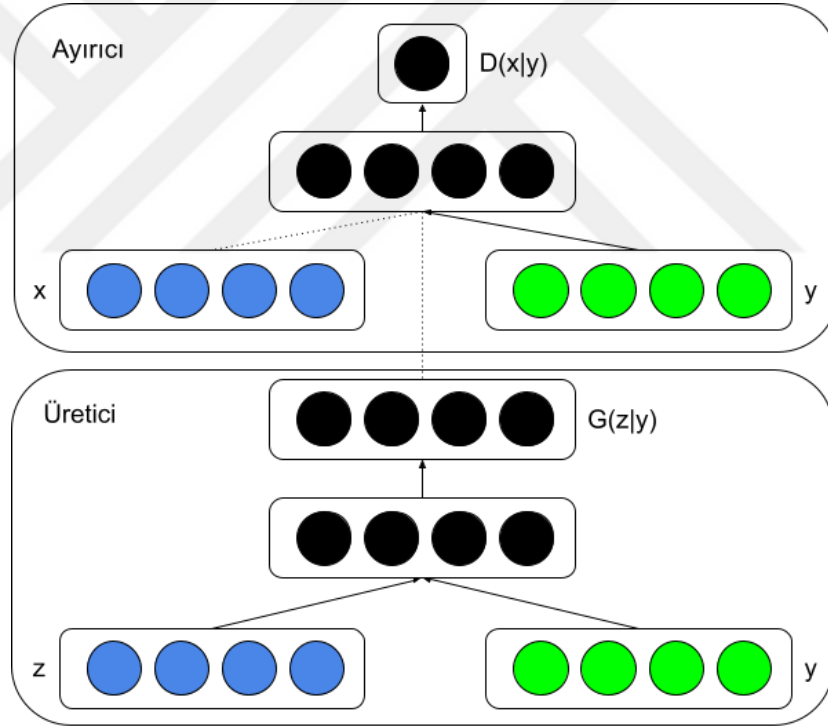


Şekil 2.14 Bulunan yönleri uygulayarak elde edilen örnek değişimler (Jahanian vd., 2020)

Bu yaklaşımlar denetim gerektirmez iken, yalnızca otomatik olarak elde edilebilen basit dönüşümleri yakalayan yönleri bulmak için kullanılabilinmektedirler. Tüm bu yaklaşımlar, yalnızca araştırmacıların tanımlamayı umdukları yönleri keşfedebilmektedirler.

Model yorumlanabilirliği üzerine alternatif bir araştırma alanı, ayrıştırılmış gizli uzaylara sahip üreticiler eğitmeyi amaçlar (Chen vd., 2016; Higgins vd., 2017; Lee vd., 2020; B. Liu vd., 2020; Mirza ve Osindero, 2014; Ramesh vd., 2019; Shoshan vd., 2021). Sınırlı sayıda manuel etiketlenmiş özellikler kullanarak görsel özellik ayrıştırma problemi

kısmen kolaylaşmaktadır fakat farklı alanlara ait bir çok veri kümelerinin tüm özelliklerini etiketlemek ve tanımlamak zor ve maliyetli bir iştir. Bir yöntem (Mirza ve Osindero, 2014) etiketlenmiş veriler ile GAN eğiterek, modelin ilgili özelliklere yönelik duyarlı olmasını sağlamaktadır. Eğitilen model böylece daha önce rastlamadığı özellik kombinasyonlarında oluşan etiketlere uygun veri üretebilmektedir. Koşullu GAN (Conditional GAN, cGAN) olarak adlandırılan bu yöntemde, üretici ve ayırıcı modellere ek bir girdi (y) verilmektedir. Bu girdi, sınıf etiketi veya farklı modalitelere ait bilgilerden oluşabilmektedir. Üretici için z ve y yeni bir katman ile birleştirilerek, birleşik gizli vektör oluşturulmaktadır ($G(z|y)$). Bu vektör daha sonra üreticiye girdi verilerek çıktı üretilmektedir. Ayırıcı model için ise, üreticinin çıktısı ve y girdi olarak sağlanmaktadır ($D(G(z|y) | y)$). Bu veri akışı Şekil 2.15’de gösterilmiştir. Bu kurgunun denklemi ise Denklem 2.10 de verilmiştir.



Şekil 2.15 Koşullu GAN Mimarisi

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} \log(D(x|y)) + \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z|y))) \quad (2.10)$$

Eđitilmiş Koşullu GAN ile veri üretirken, y 'nin ilgili değeri hedeflenen çıktıya göre seçilerek, çıktı yönlendirilebilmektedir. MNIST veri kümesi ile eğitilmiş bir koşullu GAN'ın örnek çıktıları Şekil 2.16'da verilmiştir. Her satır bir rakam sınıfı için koşullandırılmıştır. Her sütun ise farklı bir örnek çıktı içermektedir.

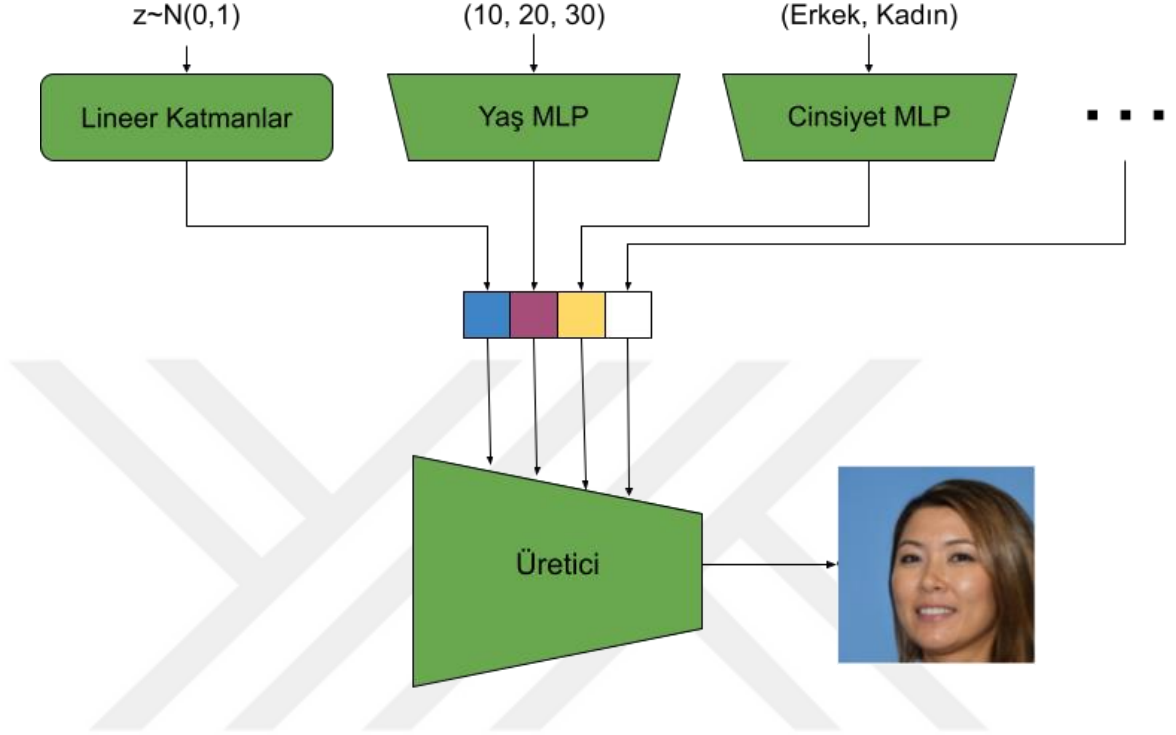


Şekil 2.16 Koşullu GAN ile üretilmiş örnek veriler (Mirza ve Osindero, 2014)

Denetimli eğitilen Koşullu GAN yöntemi, üretilen çıktıların kısıtlı biçimde yönlendirilebilir olmalarını sağlamaktadır. Kompleks verilerin tüm özelliklerini etiketlemek, onların somut olarak tanımlanabilir olmasını gerektirmektedir, fakat bu her zaman mümkün olmamaktadır.

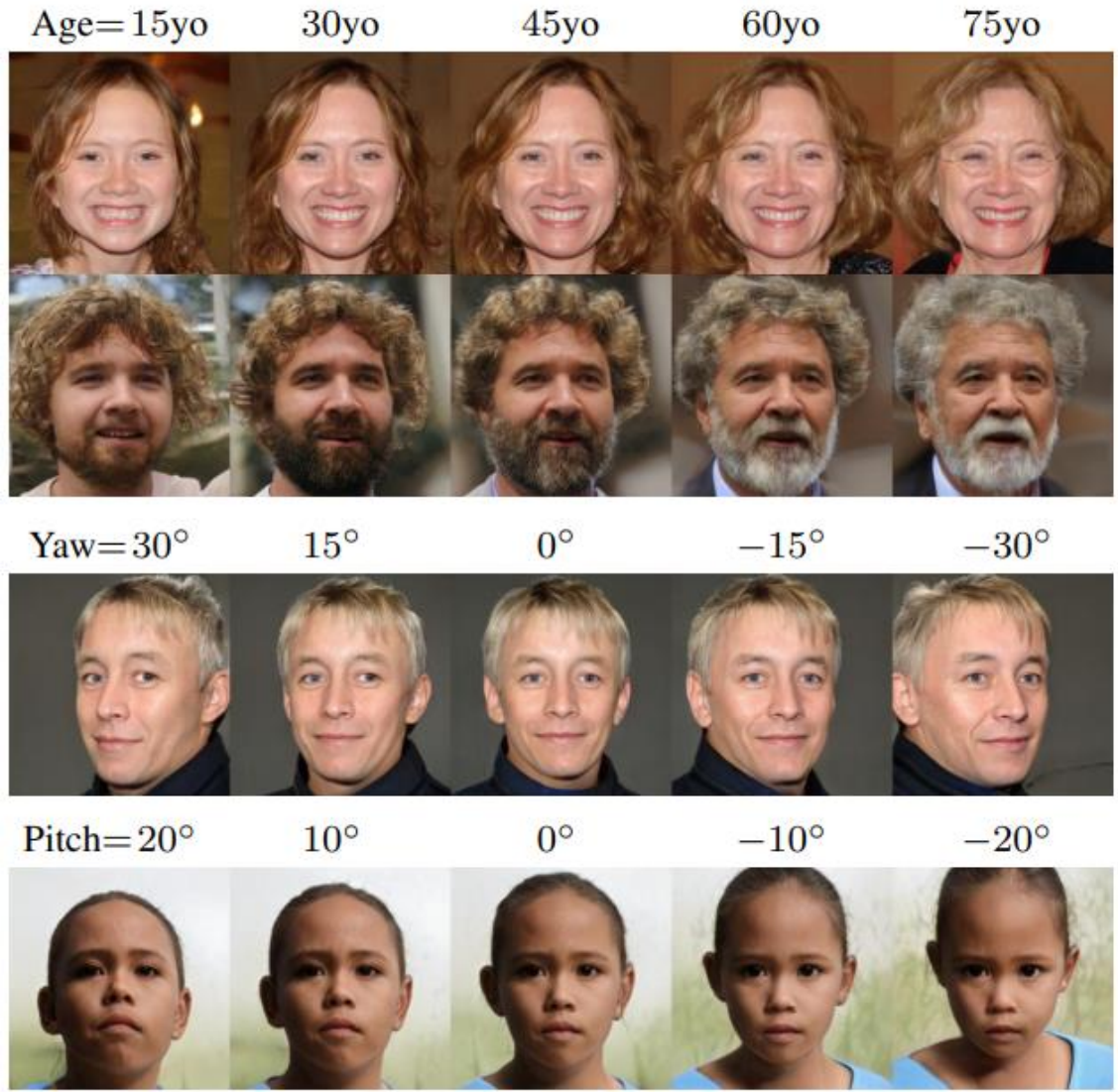
Verilerin görsel özelliklerindeki anlamsal değişimlere sebep olan önemli faktörlerin, verinin kendisinden çıkarılmaları daha genel uygulanabilir bir çözümün bulunması için gerekli olmaktadır. GANControl adında farklı bir denetimli eğitilen çalışma ise (Shoshan vd., 2021) iki fazlı bir eğitim uygulamaktadır. GANControl'un ilk fazında bir GAN ayrıştırılmış özelliklerle eğitilerek, gizli uzay alt bölümlere bölünür. Bu sayede her bölüme açık olarak ayrıştırılmış olan özelliklerin biri denk gelir. Bölümlerin diğerinin ise herhangi bir hedefli özellik için kullanılmayarak, çıktının kontrol edilemeyen özelliklerini temsil etmesine izin verilmektedir. İkinci fazda her özellik için bir çok katmanlı algılayıcı (Multilayer Perceptron, MLP) eğitilmektedir. MLP'lerin görevi, insan tarafından

sağlanan anlamsal girdilerin (örneğin yaş=20, cinsiyet=erkek) gizli uzaydaki ilgili alt uzaylara kodlanmasıdır. Bu mimari ile veri üretiminin bir örneği Şekil 2.17’de gösterilmiştir.



Şekil 2.17 GANControl ile veri üretimi

Böylece açık bir biçimde kontrol edilebilir bir gizli uzay ve bu uzaya bağlı olarak kontrol edilebilir çıktılar elde edilmektedir. GAN’ın çekişmeli kaybını ve eğitimin ilk fazdaki kontrast kaybını birleştirerek eğitim sonrasında kontrollü bir biçimde veri üretebilen bir GAN elde edilmektedir. GANControl yöntemi ile üretilen örnek veriler Şekil 2.18’de verilmiştir. Bu örnekte yaş ve poz değerleri değiştirilerek her satırda bir kaynak örneğin farklı versiyonları görünmektedir. Görseldeki kişinin kimliği değişmeden hedeflenen özelliğin değiştiği gözlemlenebilmektedir.



Şekil 2.18 GANControl örnek çıktıları. Yaş ve Poz değerleri değiştirilerek oluşturulan örnekler (Shoshan vd., 2021)

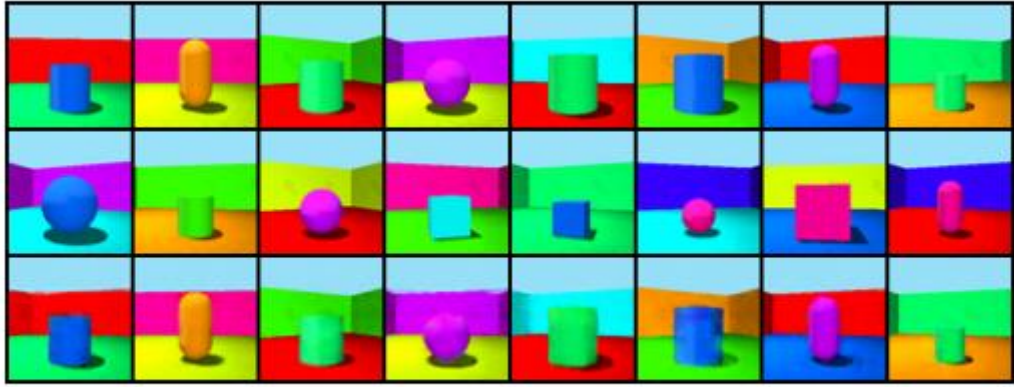
Yakın zamanda daha çok dikkat çeken çalışmalar ise denetimsiz veya öz denetimli olan yöntemler kullanmaktadırlar. Bilgi teorisinden esinlenen öncü bir çalışma olan InfoGAN (Chen vd., 2016), etiketli veri gerektirmeyen bir yaklaşım sunmaktadır. InfoGAN yönteminde GAN'a girdi olarak sağlanan verilerin eksenlerinin, gizli uzaydaki dağılım ile hizalanmaları hedeflenmektedir. Bu yöntem ile eğitilen modeller karmaşık olmayan veri dağılımları ile iyi çalışırken, daha karmaşık verilerde eğitimi ve hedeflenen eksenlerin hepsinin bulunması zorlaşmaktadır. Bu nedenle yalnızca model tarafından dikkate alınan görsel özellikler kontrol edilebilir ve imgeler sadece bu kısıtlı özellik

kümesi üzerinden değiştirilebilirler. Bu yöntem ayrıca seçilen görsel özelliklerin girdi vektöründeki hangi eksenler tarafından kodlanacağına yönelik bir ön tasarım gerektirmektedir. Kayıp fonksiyonları da bunu dikkate alacak şekilde güncellenmeleri gerekmektedir.

DisCont (Bhagat vd., 2020) adlı bir çalışmada ise veriye çeşitli işlemler uygulanarak görsel özelliklerin biri değiştirilip diğer özellikler korunmuş, böylelikle de modelin değişen özelliği ayrıştırması ve öğrenmesi sağlanmıştır. Veri yalnızca eğitim sırasında işlenerek kullanıldığı için bu yöntem öz denetimli bir eğitim sağlamaktadır. Gizli uzayda yapılan birtakım değişikliklerin üretilen görselin özelliklerine nasıl etki ettikleri tespit edilmiştir. Bu bağlamda modelin gizli uzayı iki farklı alt uzaya bölünmektedir. $Z_f \subset R^{d \times k}$ uzayı ayrıştırılmış ve yorumlanabilir özellik içeren öznitelik uzayını belirtir; burada d ve k , sırasıyla alanın boyutsallığını ve öznitelik sayısını göstermektedir. İkinci alt uzay ise $Z_u \subset R^d$ olarak tanımlanmaktadır ve belirsiz öznitelik alanını belirtmektedir. Buna arka plan ve karışık öznitelikleri dahil olmaktadır. Z_u uzayının standart normal dağılıma uyum sağlaması zorlanmaktadır. Şekil 2.19'da verilen örneklerde DisCont yöntemi uygulanarak oluşturulan örnekler verilmiştir. Her iki veri kümesi için ilk iki sıradaki görüntüler, test setinden rastgele seçilen örneklerdir. Alt sıradaki görüntülerde ilk sıradaki görüntüden belirtilen bir öznitelik ikinci satırdaki görüntüdeki karşılık gelen öznitelik ile değiştirilerek oluşturulurken, diğer tüm öznitelikler sabit tutulmaktadır. Cars3D (a) (Reed vd., 2015) ve 3DShapes (b) veri kümeleri için örnekler verilmiştir. Cars3D örnekleri için renk, 3DShapes için ise oryantasyon değerleri değiştirilmiştir.



(a) Cars3D; Specified Attribute: Color



(b) 3DShapes; Specified Attribute: Orientation

Şekil 2.19 DisCont ile Cars3D ve 3DShapes veri kümelerinden üretilen örnek görseller (Bhagat vd., 2020)

Kayda değer farklı bir çalışma, standart VAE'nin hedefindeki KL terimine daha fazla vurgu yapan β -VAE (Higgins vd., 2017) modelini önermektedir. Modelin hedefindeki bu değişiklik, gizli uzayın daha verimli olmasını gerektirir, bunun da ayrıştırılmış gizli uzay ile sonuçlandığı gösterilmiştir. Bu modeller, gizli uzayların ayrıştırılmalarını sağlarken, üretim kalitesi ve çeşitliliği açısından GAN'lara göre genellikle daha zayıf kalırlar.

Birkaç çalışma, orijinal mimarileri ve eğitim protokolleri geliştirerek bu sorunları ele almaktadır. Örneğin, (B. Liu vd., 2020), kod vektörü c 'yi iki tabanına (binary) indirgeyerek, GAN ayırıcıların kodu tahmin etme görevini basitleştirmektedir. Farklı bir çalışmanın yazarları (Lee vd., 2020) VAE tarafından ayrıştırılmış bir kod elde

etmektedirler ve ardından keşfedilen kodu GAN modeline iletmektedirler. VAE ve GAN'ı birleştirerek iyi ayrıştırılmış ve kaliteli görüntüler elde etmektedirler.

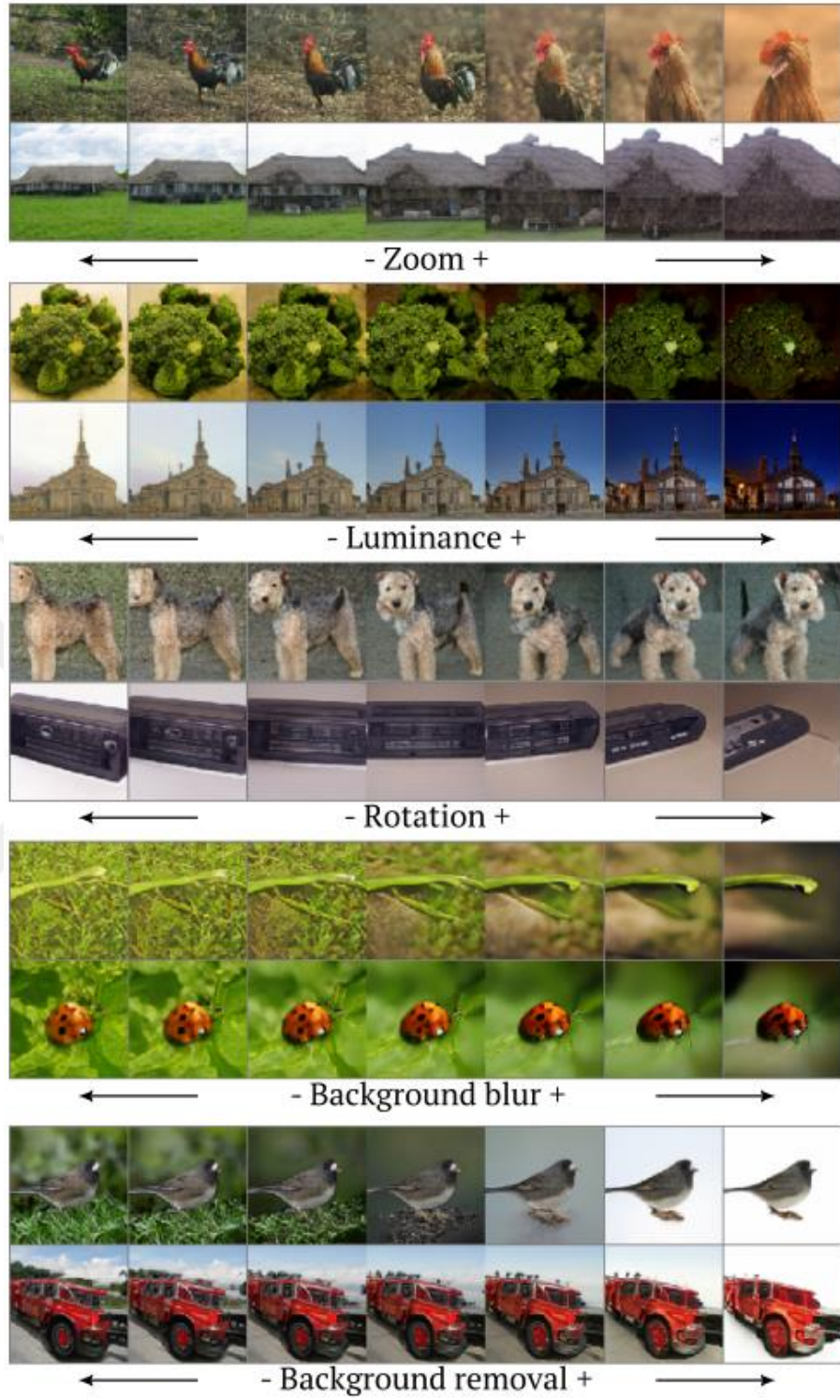
GAN'ın gizli uzayını eğitim sırasında ayrıştırmak uygulanabilir olduğunda geçerli ve başarılı bir çözüm olarak değerlendirilebilir. Halihazırda eğitilmiş ve karmaşık bir gizli uzaya sahip olan bir GAN'ın gizli uzayını ayrıştırmak ise bu yöntemlerle mümkün değildir. Son yıllarda bu problemle ilgilenen birtakım çalışmalar yapılmıştır.

DisCo çalışmasında (Ren vd., 2022), kontrast kaybı kullanılarak önceden eğitilmiş modellerin gizli uzayları ayrıştırılmış bir biçimde modelleyen yeni bir gizli uzay üretilmektedir. Mevcut gizli uzay sistematik bir şekilde keşfedilirken yeni bir uzay eş zamanlı olarak tanımlanmaktadır. Önceden eğitilmiş bir üretici modelden yararlanmanın iki faydası vardır: (i) yüksek kaliteli görüntü üretimine sahip önceden eğitilmiş modeller halihazırda bulunabilmektedir. Bu, ayrıntılı görüntü varyasyonlarını ve kontrol edilebilir üretim gibi sonraki görevleri yansıtmak için önemlidir; (ii) faktörler önceden eğitilmiş modele gömülüdür. Böylece bu model denetimsiz çözülmüş temsil öğrenimi için tümevarımsal bir önyargı görevi görmüştür. DisCo iki temel modülden oluşmaktadır. İlki, gizli uzayda aday geçiş yönleri sağlamak için bir navigatör modülüdür. İkincisi ise görüntü varyasyonlarının temsiliyi çıkarmak ve ayrıştırılmış temsillere dayalı bir varyasyon uzayı oluşturmak için bir Δ -Kontrastör modülüdür. Daha spesifik olarak, Δ -Kontrastör, iki ortak ağırlık ayrıştırıcı kodlayıcıdan oluşmaktadır. İki görüntü arasındaki varyasyon, çözücü kodlayıcılar tarafından çıkarılan kodlanmış temsillerinin farkı olarak modellenmektedir. Varyasyon uzayında, aynı yönde ilerlemekten kaynaklanan varyasyon örneklerini bir araya getirerek ve farklı yönlerden ilerlemekten kaynaklananları iterek, navigatör ayrıştırılmış yönleri faktörler olarak keşfetmeyi öğrenmektedir. Aynı zamanda ayrıştırıcı kodlayıcı, görüntülerden ayrıştırılmış temsilleri çıkarmayı öğrenmektedir. Böylece keşfedilen yönlerde hareket etmek belirgin görüntü varyasyonlarına neden olmaktadır. VAE ve InfoGAN tabanlı yöntemlerden farklı olarak, DisCo yönteminde ayrıştırılmış temsiller ve faktörler iki ayrı uzaydadır. Önceden eğitilmiş GAN'ın poz yönü DisCo ile ayrıştırılarak üretilen örnek veriler Şekil 2.20'de verilmiştir.



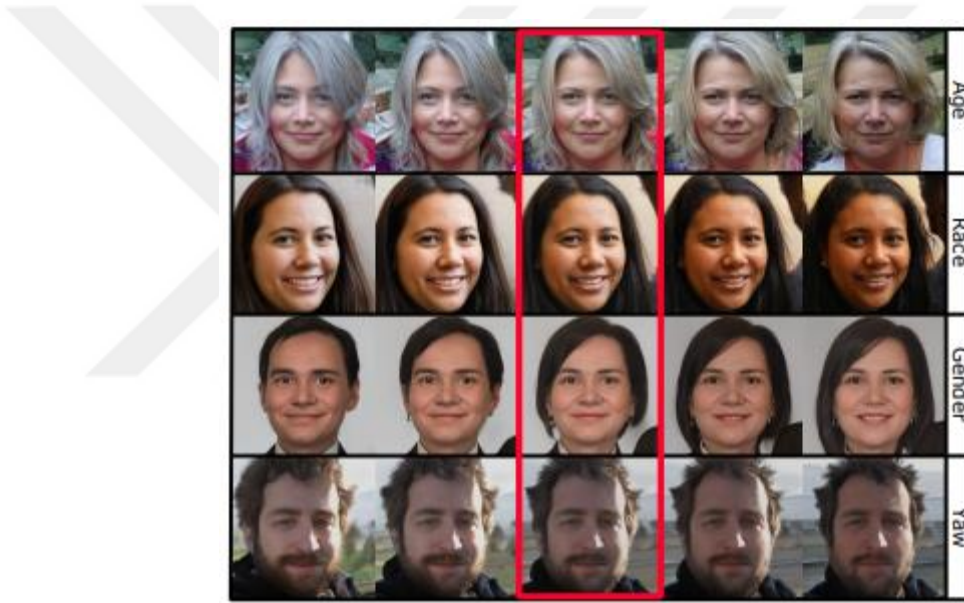
Şekil 2.20 DisCo ile ayrıştırılmış bir GAN ile üretilen örnek veriler (Ren vd., 2022)

Önceden eğitilmiş modellerin gizli uzayını tarayarak anlamsal ve doğrusal yönleri denetimsiz biçimde keşfetmeyi hedefleyen bir çalışmanın (Voynov ve Babenko, 2020) sunduğu yöntem gizli vektörü değiştirip, çıktıya yansıyan farkı ölçerek bir yönün anlamını ölçmektedir. Bu denetimsiz yöntemin amacı, önceden eğitilmiş bir GAN üreticisinin gizli uzayındaki yorumlanabilir yönleri keşfetmektir. Bu bağlamda üretici (G), gizli uzaydan alınan örnekleri ($Z \in R^d$) görsel uzayına (I) eşleştirmektedir ($G: z \rightarrow I$). Bu yöntemde üretici eğitilmemektedir. Eğitilen bileşenler ise iki tanedir. İlki $A \in R^{dxK}$ matrisidir. Burada d G 'nin gizli uzayının boyutsallığına eşittir. K ise yöntemin keşfedeceği yönlerin sayısını belirten bir hiperparametredir. Eğitilen ikinci bileşen yeniden yapıcı model R 'dir. R 'nin girdisi G ile üretilen bir görsel çiftidir. Bu çift, z ve z 'ye A 'dan bir yön eklenerek elde edilen vektörlerin G 'ye verilerek oluşturulan görsellerden oluşmaktadır. Yani ikinci resim, ilk resmin değişime uğramış halidir. R 'nin görevi ilgili bir görüntü dönüşümünü tetikleyen gizli uzaydaki değişimi yeniden üretmektir. K 'nin değeri üretici modelin gizli uzayının boyutsallığına eşit olarak seçilmektedir. Bu değerin büyüklüğü fazla ise (örneğin 128) modelin öğrenmesi gereken anlamsal yön sayısı çok artmaktadır ve K 'ye daha küçük bir değer atanmaktadır (örneğin 64). Bu yöntem MNIST veri kümesi ile önceden eğitilmiş bir GAN'a uygulanarak Şekil 2.21'deki görseller elde edilmiştir. Sırasıyla yakınlık, parlaklık, rotasyon, arka plan bulanıklığı ve arka plan giderilmesine denk gelen yönlerde ilerlenerek her birisi için ikişer örnek sunulmuştur.



Şekil 2.21 Farklı görsel özellikleri temsil eden yönlerde ilerlenerek elde edilen sonuçlar (Voynov ve Babenko, 2020)

GAN uzayındaki yönlerin her zaman doğrusal olmadığını öne süren ve bununla alakalı bir yöntem sunan WarpedGANSpace adlı çalışmada (Tzelepis vd., 2021) K tane eğim fonksiyonu kullanılarak ve K tane eğim yönü bulunarak yorumlanabilir yönler tespit edilmektedir. Yazarlar, görsel özellikleri tarif eden yönlerin doğrusal olmayabildiklerini ispatlarken, bazı özellik yönlerinin doğrusal olabileceğini tespit etmişlerdir. Bir önceki çalışma gibi bu çalışma da hedeflediği tek bir özelliği temsil eden yönü bulmak yerine belirli boyutlara sahip bir anlamsal yönler kümesi bulmaktadır. WarpedGANSpace yöntemi StyleGAN2'ye (Karras vd., 2020) uygulanarak elde edilen çıktılar Şekil 2.22'de verilmiştir. Burada yaş, ırk, cinsiyet ve açı özelliklerini ifade eden yönlerde artan ve azalan yönde ilerlenerek baz resime değişimler uygulanmıştır.



Şekil 2.22 WarpedGANSpace yönteminin StyleGAN2'ye uygulanarak elde edilen örnek değişimler (Tzelepis vd., 2021)

Diğer bir yaklaşım ise, ayırıcı ağ içermeyen ve doğrudan kodlayıcı ile üretici arasında bir rakip oyun kurulan çekişmeli üretici-kodlayıcı (Adversarial Generator-Encoder, AGE) mimarisidir (Ulyanov vd., 2017). Bu yaklaşımda, eğitimde çekişmeli kayıp ve yeniden oluşturma kaybının bir kombinasyonu ile gerçekleştirilmektedir. Böylece kodlayıcı ve üretici karşılıklı olmaya teşvik edilmektedir. Bu mimari üç temel avantaj sağlamaktadır. Bunlardan ilki, karşıt kayıp yalnızca kodlayıcı ve üretici arasında hesaplandığı için, ayırıcı modele gerek yoktur ve öğrenilen parametre sayısı azalmaktadır. İkincisi,

GAN'larda dağılımlar genellikle yüksek boyutlu görüntü uzayında karşılaştırılırken, AGE'lerde karşılaştırma daha basit ve daha az boyuta sahip bir gizli uzayda yapılmaktadır. Üçüncüsü ise, kodlayıcı ağına girdi olarak hem gerçek hem de sahte veriler verildiği için, gerçek görüntüleri kodlamayı daha iyi öğrenebilmektedir. Bu yaklaşımı kullanan çalışmalar incelendiğinde ayırıcı modellere sahip mimarilere göre görüntü kalitesi ve çeşitlilik açısından zayıf oldukları görülmektedir (Karras vd., 2022). Ek olarak, AGE mimarisinde yeniden oluşturma kaybı kullanılmadığında, kodlayıcı ve üretici eşleştirmesi karşılıklı olmamaktadır. Ağları karşılıklı hale getirmek için yeniden oluşturma kaybı, hedef fonksiyonundaki bir hiperparametre tarafından kontrol edilmektedir ancak bu parametrenin değerinin belirlenmesi ekstra yük getirmektedir.

Bu bağlamda AGE'lerin üretici ağı $g(z)$, gizli uzay z 'yi veri uzayı x 'e eşleştirmektedir. Kodlayıcı ağ $e(x)$ ise veri uzayındaki görsel x 'i gizli uzay z 'ye eşleştirmektedir. Kodlayıcı eğitiminde yeniden oluşturma kaybı L , veri uzayındaki kayıp L_x ve gizli uzaydaki kayıp L_z 'nin birleşimi olarak hesaplanmaktadır. Kayıp fonksiyonları Denklem 2.11 ve Denklem 2.12 de verilmiştir. Kodlayıcının ağırlıkları her eğitim döngüsünde bu kayıplardan hesaplanan iki hedef fonksiyonu ile güncellenmektedir.

$$L_x = \mathbb{E}_{x \sim p_{data}} \|x - x'\|_1 \quad (2.11)$$

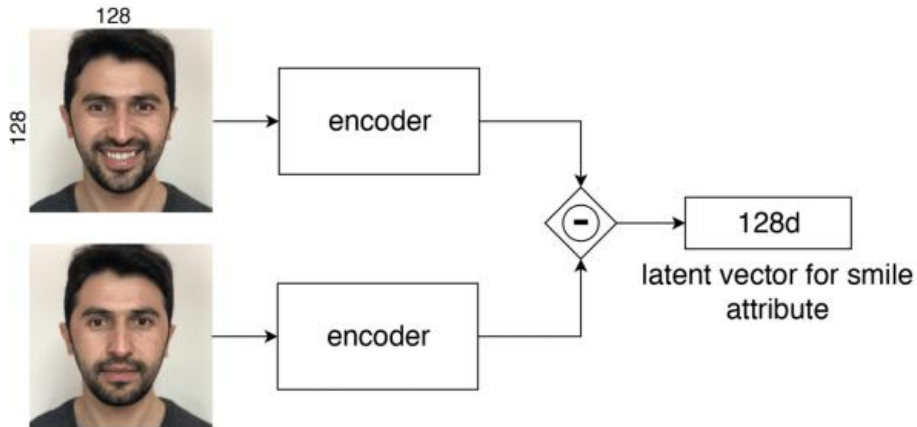
$$L_z = \mathbb{E}_{z \sim P(z)} \|z - z'\|_2^2 \quad (2.12)$$

Bu kategoride belirtilen sorunların üstesinden gelmek için, bir çalışmada (Dogan ve Keles, 2020) birbiriyle döngüsel bağlantıları olan eğitimi iki adımda gerçekleştiren ve iki ayrı bölümden oluşan CRG adlı bir mimari önerilmektedir. İlk adımda, üretici ve ayırıcı arasındaki orijinal çekişmeli kaybı kullanılarak bir üretici eğitilmektedir. Bu adım normal GAN eğitiminden farksızdır. İkinci adımda, kodlayıcı ve üretici arasındaki döngüsel bağlantı kullanılarak kodlayıcı ağı sıfırdan eğitilmektedir. Bu mimari, üretici ağın rastgele oluşturulmuş veya kodlanmış bir gizli vektörü girdi olarak almasına ve kodlayıcı ağın eğitim sırasında girdi olarak gerçek veya sahte görüntüler almasına izin vermektedir. Bu yaklaşım, kodlayıcıların eğitim sırasında yaşadıkları yakınsama sorunlarını gidermek için yardımcı olmaktadır. Ayrıca, farklı alanlarda görüntü üretimi için hazır olan herhangi bir önceden eğitilmiş üreticinin tersini öğrenme imkanını sağlamaktadır. Kodlayıcının başarısı, üreticinin gizli vektörleri görüntülere çevirme başarısıyla doğrudan ilişkilidir.

CRG yöntemi ile bir özelliğin gizli uzaydaki yönünü bulmak için iki görsel gerekmektedir. İlk görsel x_1 , referans görselidir ve hedeflenen özelliği içermemelidir. İkinci görsel x_2 ise hedef görselidir ve hedeflenen özelliği içermelidir. Bu iki görsel kodlayıcıdan geçirilerek z_1 ve z_2 gizli vektörleri yakınsanmaktadır ve vektörlerin farkı Denklem 2.13'te verilen denklem ile hesaplanarak gizli uzaydaki yön z_f bulunmaktadır.

$$z_f = \frac{z_2 - z_1}{\|z_2 - z_1\|_2^2} \quad (2.13)$$

Bu işlemin gülümseme özelliği için nasıl uygulandığı Şekil 2.23'de gösterilmiştir. Yön vektörü z_f elde edildikten sonra Denklem 2.14 kullanılarak ilgili özellik herhangi bir görsele uygulanabilmektedir. Burada z_a özellikli çıktıdır, z_p değiştirilmesi istenen herhangi bir görselin kodlayıcı ile bulunan gizli vektörüdür ve k uygulanacak özelliğin katsayısıdır.



Şekil 2.23 CRG ile yön bulma örneği (Dogan ve Keles, 2020)

$$z_a = z_p + k \times z_f \quad (2.14)$$

Bu yöntem ile oluşturulan örnek çıktılar Şekil 2.24'te verilmiştir. Şekildeki ilk örnekte referans ve hedef görseli (ilk satır, sol) kullanılarak gülümseme özelliğinin gizli uzaydaki yönü tespit edilmiştir ve iki farklı görsele uygulanarak birinci ve ikinci satırlardaki değişimler elde edilmiştir. Birinci satırda gülümseme yönü pozitif yönde artan k değeri ile uygulanarak çıktı görselindeki kişinin gülümseme özelliği artmaktadır. İkinci satırda ise k değeri negatif yönde arttığı için gülümseme özelliği azalmaktadır. Üçüncü ve

dördüncü satırda ise farklı referans ve hedef görselleri (üçüncü satır, sol) ile elde edilen yön vektörü kullanılmaktadır. Bu yön, iki farklı yöne bakan kişilerin pozlarını değiştirmek için kullanılmıştır.



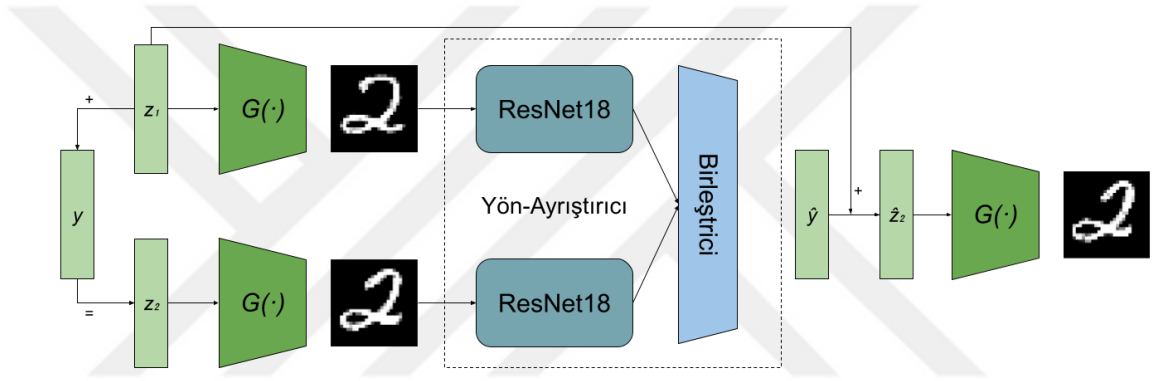
Şekil 2.24 CRG ile tespit edilen yönlerin örneklere uygulanması (Dogan ve Keles, 2020)

3. YÖNTEM

3.1 Model Mimarisi

Bu çalışmada oluşturulan Yön-Ayrıştırıcı modelin hedefi, bir GAN ile üretilen iki farklı görseli gizli uzayda ayırt eden vektöre yakınsamaktır. Bunun için referans görsellerini birbirinden farklı kılan görsel özellikleri gözeterek, onların farkını GAN'ın gizli uzayında temsil eden vektörü bulmayı hedeflemektedir. Yön-Ayrıştırıcı ağın girdileri, en az bir görsel özelliği farklı olan iki görseldir. Çıktısı ise bu farkı ifade eden gizli yön vektörüdür. Bu bağlamda, Yön-Ayrıştırıcı ağ girdi olarak aldığı iki görselin özniteliklerini çıkarıp bunları karşılaştırmayı hedeflemektedir. Yön-Ayrıştırıcı ağ için ResNet18 (He vd., 2016) mimarisi temel alınmıştır. ResNet18, 18 derin katmana sahip, toplam 72 katmanlı bir evrişimsel sinir ağı mimarisidir. Bu ağın mimarisi, büyük miktarlarda evrişimsel katmanların verimli bir şekilde çalışmasını sağlamayı amaçlamaktadır. Ancak, bir ağa fazla sayıda derin katmanın eklenmesi genellikle çıktının bozulmasına neden olmaktadır. Bu problem sinir ağların geri yayılım yoluyla (back propagation) eğitilirken meydana çıkabilen kaybolan eğim sorunu (vanishing gradient) olarak bilinmektedir. Birden çok katmanın varlığı nedeniyle, tekrarlanan çarpım işlemleri, gradyanın giderek küçülmesine ve dolayısıyla ağ performansının doygunluğa ve hatta düşmesine sebep olabilmektedir. ResNet'in ana fikri, çoğunlukla kısayol bağlantıları veya kimlik bağlantıları olarak adlandırılan atlama bağlantılarının kullanılmasıdır. Bu bağlantılar, katmanlar arasında kısayollar oluşturup, bir veya birden fazla katmanın üzerinden atlayarak çalışmaktadır. Kısayol bağlantılarının kullanılmasının amacı, derin ağların karşılaştığı kaybolan eğim sorununu çözmeleridir. Kısayol bağlantıları, bir önceki katmanın aktivasyonlarını kullanarak kaybolan eğim sorununu ortadan kaldırmaktadırlar. Bağlantı atlama işlemi ağı sıkıştırdığı için ağ daha hızlı öğrenmektedir. Bağlantıların sıkıştırılması sayesinde, ağın kalan kısmının daha fazla özellik alanı eğitmesini ve keşfedebilmesini sağlamaktadır. Tezin kapsamında ResNet18 mimarisine birkaç değişiklikler uygulanmıştır. ResNet18'in orijinal girdi katmanı üç kanallı veriler için tasarlandığı için, bu katman tek kanallı veri kabul eden, 7x7'lik evrişimsel filtreye sahip ve 64 öznitelikli çıktı üreten bir katman ile değiştirilmiştir. Tek kanallı olmasının nedeni, kullanılan veri kümelerinin tek kanallı olmalarıdır. Ayrıca ResNet'in çıktı katmanı yeterince karmaşık veriler kapsayabilmesi

için 200 boyutlu çıktı üreten bir doğrusal katman ile güncellenmiştir. Oluşan yeni özellik çıkarma mimarisi ikiz ağ olarak kurgulanmıştır, yani girdiler birbirinden bağımsız ve paralel olarak ağdan geçerek, birer 200 boyutlu öznitelik vektörü üretmektedirler. Bu vektörler iki doğrusal katmandan geçerek birleştirilmektedirler. Nihai katmanın boyutu, hedeflenen yön vektörünün boyutuna eşittir. Modelin çıktısı böylece yakınsamaya çalışılan yön vektörü olarak kullanılabilir. MNIST ile yapılan deneylerde yön vektörü boyutu InfoGAN için 2, DCGAN için ise 100 olarak tanımlanmıştır. FreyFaces ve InfoGAN deneylerinde ise 5 boyutlu bir vektör kullanılmıştır. Yön vektörün boyutu ilgili önceden eğitilmiş GAN'ın z vektör boyutuna eşit olarak seçilmiştir. Yön-Ayrıştırıcı modelin genel mimari Şekil 3.'de verilmiştir.



Şekil 3.1 Geliştirilen Yön-Ayrıştırıcı yöntemin akış diyagramı

3.2 Kullanılan Üretici Modeller

Bu çalışma kapsamında üretici ağ olarak veri kümesinden bağımsız olarak iki farklı GAN mimarisi kullanılmıştır. Bu mimariler Deep Convolutional GAN (DCGAN) ve Information Maximizing GAN'dır (InfoGAN). DCGAN mimarisi GAN'ların genel mimarisinin (Şekil 2.3) evrimsel katmanlar kurgulanmış bir versiyonudur. Evrimsel katmanların sayesinde görsellerin özniteliklerini ve böylece onların gerçeğe yakın sahte örneklerini üretmeye daha uygundur. DCGAN'ın eğitiminde gizli uzayını ayırtırmaya yönelik herhangi bir strateji uygulanmamaktadır ve bundan dolayı gizli uzayın eksenleri üretilen görsellerin özelliklerine direkt olarak karşılık gelmektedir. InfoGAN mimarisinde de evrimsel katmanlar kullanılmıştır, fakat temel GAN mimarisine ek olarak rastgele belirlenen gizli vektörün (z) yanında ikinci bir girdi vektörü (c) kullanılmaktadır (Şekil 2.8). Modelin eğitimi sırasında model c vektörünü görsellerin

özelliklerini ifade edecek şekilde öğrenmeye çalışmaktadır. Ayırık ve devamlı değerler içeren c vektörü, kontrol edilebilmesi hedeflenen görsel özelliklere göre eğitim öncesinde tasarlanması gerekmektedir. Farklı sınıflar için ayırık değişkenler kullanılırken devamlı değerlere sahip olan özellikler için devamlı değişkenler kullanılmaktadır. InfoGAN daha sonra görseli değerlendirerek c vektörünü bir çıktı olarak yakınsamaya çalışmaktadır. InfoGAN c vektörünü eğitim esnasında farklı görsel özelliklerin eksenleriyle hizalamaya çalışır, fakat hangi özelliklere karşılık geleceği bilinmemektedir. Eğitim sonrasında c vektörüne atanan değerler ile eksenleri hizalanan görsel özellikler kontrol edilebilmektedir.

3.3 Veri Kümeleri

Yön-Ayrıştırıcı modelin eğitimi için MNIST ve FreyFaces veri kümeleri ile eğitilmiş farklı GAN'lar kullanılarak bu veri kümelerin türevleri oluşturulmaktadır. Orijinal veri kümelerin temel bilgileri ve türev veri kümelerin nasıl oluşturuldukları hakkında ayrıntılı bilgi verilecektir.

3.3.1 MNIST veri kümesi

MNIST, elle yazılmış rakam görselleri içeren ve yaygın olarak kullanılan bir veri kümesidir (Deng, 2012). Eğitim bölütü 60.000 veri örneğinden oluşurken test için ayrılmış olan küme 10.000 veri örneği içermektedir. İçerdiği veri örneklerin boyutları 28x28 pikseldir ve bütün görseller tek kanallı, yani siyah beyaz, görüntülerdir. 1998'de tanıtılmıştır ve sınıflandırma görevleri için standart bir ölçüt haline gelmiştir. Görsellerin temsil ettiği veri çok karmaşık olmadığı için birçok farklı makine öğrenmesi alanlarında yeni yöntemlerin işlevselliğini kanıtlamak için kullanılmaktadır.

Yön-Ayrıştırıcı modelin eğitimi için veri kümeleri oluşturulmuştur. Rastgele seçilen gizli vektörler (z_1 ve z_2), üretici modele ($G(\cdot)$) girdi olarak verilerek iki imge ($G(z_1)$ ve $G(z_2)$) üretilmektedir. Yön-Ayrıştırıcı modelin girdi olarak aldığı bu iki imgeyle, z_1 'den z_2 'ye giden yön vektörünü ($\hat{y}$) yakınsamayı hedeflemektedir. Veri üretiminde tek bir rakam, örneğin 2, kullanılmaktadır. Bu rakam sabit tutularak, onu etkileyen anlamsal özelliklerin model tarafından öğrenilmesi hedeflenmektedir. Önceki çalışmalar tayin edilen bir yönün GAN gizli uzayında doğrusal vektör operasyonları ile tutarlı bir biçimde

uygulanabildiğini göstermişlerdir (Dogan ve Keles, 2020). Bu davranış, Yön-Ayrıştırıcı modelin gizli uzay eksenlerini bir rakam için çözdüğünde, bulunan yön vektörünün farklı rakamlara uygulandığında aynı etkiye sebep olacağı anlamına gelmektedir. InfoGAN ile veri üretirken çıktının ($G_{IG}(z)$) hangi rakam olacağı herhangi bir gizli vektör (z) için kontrol edilirken, DCGAN için aynısı geçerli değildir. DCGAN ile oluşturulan veri kümesinde tek bir rakama ait örneklerin bulunması için bir referans görseli belirlenmiştir. Daha sonra üretilen tüm verilerin otomatik belirleme için iki bağımsız derin ağ kullanılarak bir sezgisel yöntem uygulanmıştır. Rastgele üretilen referans görsellerin (2 rakamını içeren) birisi seçilmiş ve üretilen diğer verilerin rakam kimliğinin aynı olduğunu doğrulamak için iki farklı yöntem kullanılmıştır. Bu yöntemlerin ilki, MNIST ile önceden eğitilmiş bir sınıflandırıcıdan yararlanmaktadır. Sınıflandırıcının sondan bir önceki katmanın ürettiği öznitelik vektöründen sıkıştırılmış bir gömme vektörü elde edilmiştir. Diğer yöntemin temelinde MNIST ile önceden eğitilmiş bir Değişimsel Otokodlayıcı (VAE) bulunmaktadır. VAE'nin darboğaz katmanında yer alan öznitelik vektörü, girdiyi temsil eden gömme vektörü olarak kabul edilmiştir. Bu yöntemlerle, referans görselin ve kimliği doğrulanacak görselin arasındaki benzerliği tespit etmek için gömme vektörleri arasındaki Öklid mesafesi hesaplanarak, iki görselin benzerliğini temsil eden bir metrik elde edilmiştir. Bu metrik için ampirik bir eşik belirlenmiştir. Böylece oluşan karar mekanizması ile DCGAN verileri ile eğitecek olan Yön-Ayrıştırıcı model için veri kümesi üretiminde görsellerin uygunluğu otomatik olarak belirlenmiştir.

3.3.2 FreyFaces veri kümesi

FreyFaces, tek bir kişinin farklı yüz ifadelerinin görsellerini içeren bir veri kümesidir. Toplamda 1920 veri örneği içermektedir. Her bir görselin boyutları 20x28 pikseldir ve tek kanallı siyah beyaz görüntülerdir.

FreyFaces veri kümesi ile önceden eğitilmiş bir InfoGAN modeli kullanılmıştır. Nötr bir yüz ifadesine sahip olan bir referans görseli oluşturulduktan sonra bu görselin gizli vektörünün c kısmındaki beş devamlı değişkenin değerleri rastgele değiştirilerek referans görselinden yola çıkarak farklı görsel özelliklerinde değişiklik gösteren görseller oluşturulmuştur. Referans görseli ve farklı yönlere sahip olan görsellerin gizli uzaydaki farkını temsil eden yön vektörleri MNIST veri kümesinde olduğu gibi burada da Yön-

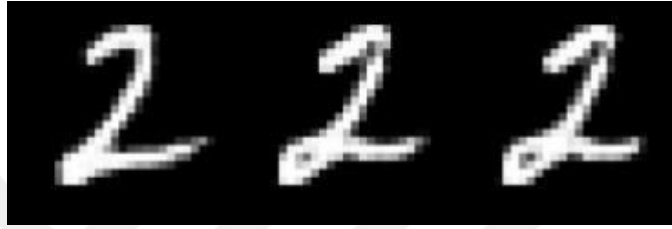
Ayrıştırıcı modelin tahmin etmeye çalışacağı vektörlerdir. Orijinal FreyFaces veri kümesinde yalnızca tek bir şahsın resimleri bulunduğu için MNIST veri kümesi ile yaşanan sınıf sabitleme problemi burada söz konusu değildir. FreyFaces eğitim veri kümesi oluşturulduktan sonra farklı bir referans görseli seçilerek doğrulama veri kümesi ve yine farklı bir referans görseli ile test veri kümesi oluşturulmuştur.



4. BULGULAR VE TARTIŞMA

Eğitimi tamamlanmış Yön-Ayrıştırıcı modeller ile eğitim veri kümesinde yer almayan test verileriyle deneyler yapılmıştır. Bu bölümün devamında, Yön-Ayrıştırıcı model uygulanarak hedeflenen anlamsal özellikleri temsil eden yönlerin kodlanıp kodlanmadığı tüm deneyler için örnekler verilerek paylaşılmaktadır.

4.1 InfoGAN MNIST Yön-Ayrıştırıcısı



Şekil 4.1 InfoGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması

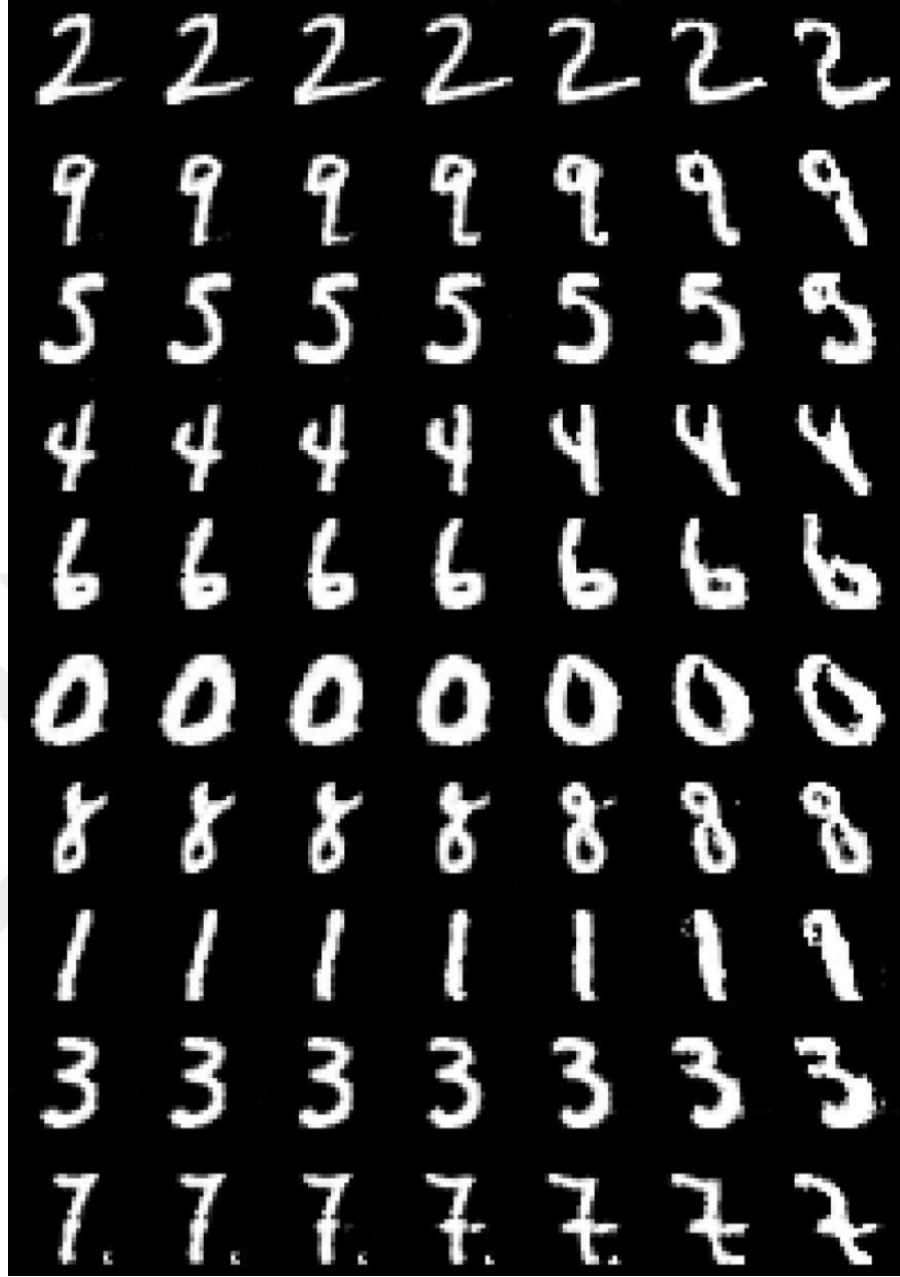


Şekil 4.2 InfoGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanması

InfoGAN kullanılarak MNIST veri kümesinin türevi oluşturulmuştur. Bu küme ile eğitilen Yön-Ayrıştırıcı model, önceden karşılaşmadığı görseller ile beslendiğinde üretilen sonuçlardan birisi Şekil 4.'da yer almaktadır. Şekildeki ilk imge $G_{IG}(z_1)$ görselidir. Ortadaki görsel rastgele üretilmiş olan hedef imgesidir $G_{IG}(z_2)$. Bu imgelerin anlamsal özelliklerinin farkı y vektörü olarak ifade edilmektedir. Yön-Ayrıştırıcı tarafından üretilen $\hat{y}$ vektörü, y vektörüne yakınsamaktadır. $\hat{y}$ ilk görselin gizli vektörüne eklenerek oluşturulan vektör üretici modele beslenerek elde edilen görsel $(G_{IG}(z_1 + \hat{y}))$ hedef görsele çok yakınsamakta olduğu görülmektedir. Bu durum Şekil 4.'da en sağdaki görselde gözlemlenebilmektedir. Bu sonuçtan, Yön-Ayrıştırıcı modelin sadece iki imgeyi kullanarak onların gizli uzaydaki farkını tespit edebildiğini ispatlamaktadır.

Yön-Ayrıştırıcı tarafından iki imge arasında bulunan yön vektörü ($\hat{y}$) bir katsayı ile ölçeklenerek kullanıldığında Şekil 4.'de görüldüğü üzere, istenen anlamsal özelliğin doğrusal olarak hedeflenen yönde değiştiği gözlenmektedir. Burada ilk görsel InfoGAN ile rastgele üretilmiş 2 imgesidir. İkinci rakam ve sonrası ise yakınsanan gizli yön vektörünün bir katsayı (a) ile çarpılarak üretilen görsellerdir ($G_{IG}(z_1 + a * \hat{y})$). Katsayı (a) arttıkça dördüncü imge ($G_{IG}(z_1 + a * \hat{y})$) hedef görseline ($G_{IG}(z_2)$) yakınsamıştır. Katsayı (a) arttırıldıkça, anlamsal özelliğin belirginliği altıncı ve yedinci rakamlarda ($G_{IG}(z_1 + a * \hat{y})$) görüldüğü üzere hedef rakamındaki seviyeyi aşmaktadır.

Yön bulunduktan sonra katsayı kullanarak, özelliğin hedef görselindeki seviyesinden bağımsız olarak, kademeli biçimde kullanılabilinmektedir. Bir rakam için elde edilen bir yön vektörü diğer rakamlara uygulanarak elde edilen görseller Şekil 4.'de verilmiştir. Burada 2 rakamı kullanarak Yön-Ayrıştırıcı ile elde edilen yön vektörü kullanılmıştır. İlgili vektör sola yaslanma özelliğini kodlamaktadır. İlk sütunda InfoGAN ile rastgele üretilmiş imgeler ($G_{IG}(z_1)$) bulunmaktadır. Her satırın devamında ise katsayı ile ölçeklenen yön vektörü ($\hat{y}$) eklenerek oluşturulan görseller ($G_{IG}(z_1 + a * \hat{y})$) verilmiştir. Şekilde görüldüğü üzere, tahmin edilen yön vektörü diğer rakamlara uygulandığında aynı anlamsal görsel özelliği başarılı bir şekilde görsel olarak etkilemektedir. Sonuçlardan hareketle InfoGAN'ın gizli uzayı Yön-Ayrıştırıcı model tarafından anlamsal olarak çözülebildiği anlaşılmaktadır.



Şekil 4.3 InfoGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği sola yaslanma özelliğinin diğer rakamlara uygulanması

4.2 DCGAN MNIST Yön-Ayrıştırıcısı



Şekil 4.4 DCGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması

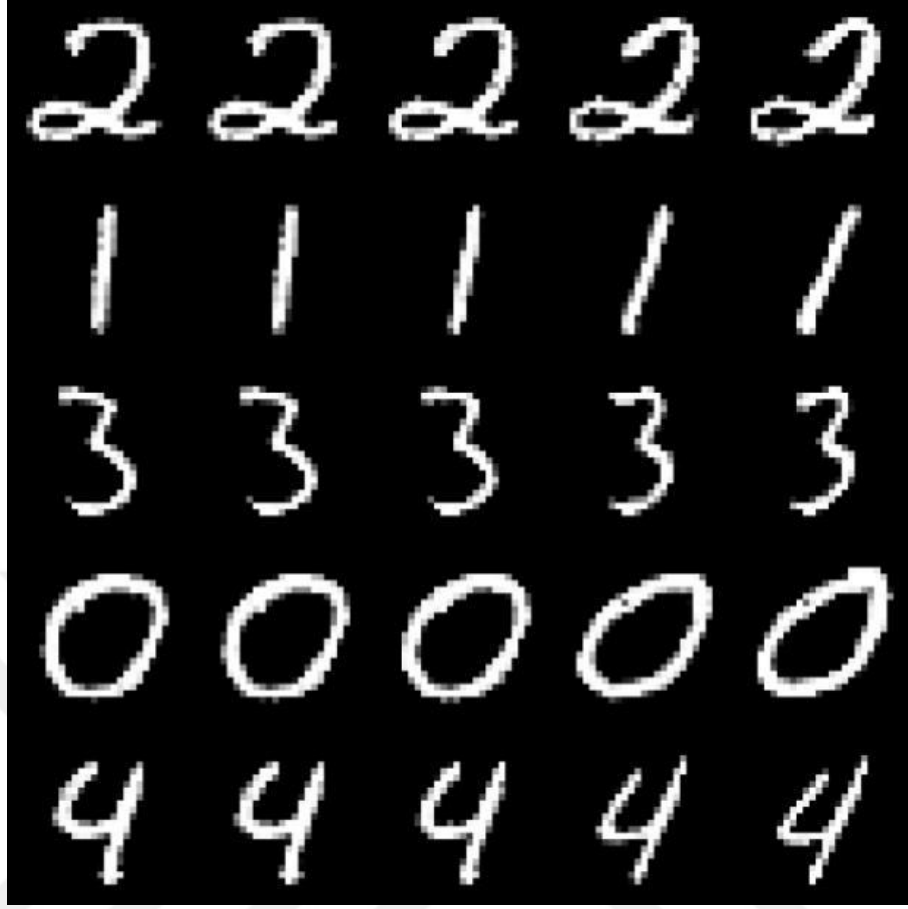


Şekil 4.5 DCGAN ile rastgele üretilmiş 2 imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanması

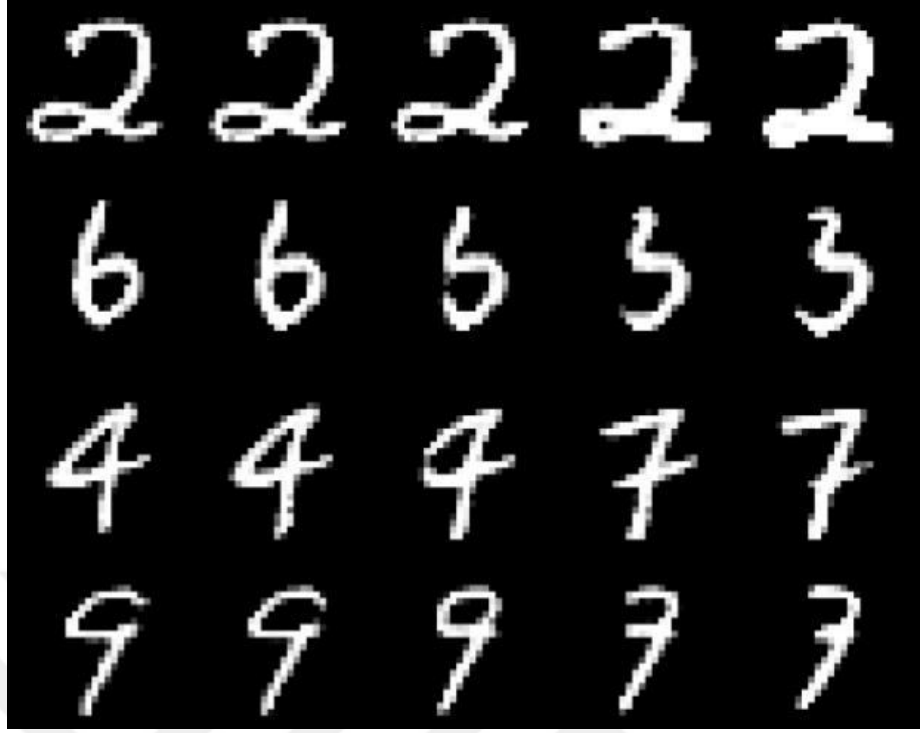
DCGAN kullanılarak ve VAE yardımıyla iyileştirilerek bir MNIST veri kümesi oluşturulmuştur. Bu veri kümesiyle eğitilen Yön-Ayrıştırıcı model kullanılarak 4.1 Bölümünde bahsedilen analiz ve deneyler yapılmıştır. Şekil 4.'da ilk imge $G_{DC}(z_1)$ görselidir. Ortadaki görsel rastgele üretilmiş olan hedef imgesidir $G_{DC}(z_2)$. Yön-Ayrıştırıcı tarafından yakınsanan $\hat{y}$ vektörü ilk görselin gizli vektörüne eklenerek oluşturulan vektör üretici modele beslenerek elde edilen görsel ($G_{DC}(z_1 + \hat{y})$) hedef görsele çok yakınsamakta olduğu görülmektedir. Şekilde görüldüğü üzere, ikinci ve üçüncü imgeler birbirine benzemektedirler. Bu, Yön-Ayrıştırıcı modelin $\hat{y}$ 'yi yakınsayabildiğini ispatlamaktadır.

Yön-Ayrıştırıcı tarafından iki imge arasında bulunan yön vektörü ($\hat{y}$) bir katsayı ile ölçeklenerek kullanıldığında Şekil 'da görüldüğü üzere, istenen anlamsal özelliğin doğrusal olarak hedeflenen yönde değiştiği gözlenmektedir. Burada ilk görsel DCGAN ile rastgele üretilmiş 2 imgesidir. İkinci rakam ve sonrası ise yakınsanan gizli yön vektörünün bir katsayı (a) ile çarpılarak üretilen görsellerdir ($G_{DC}(z_1 + a * \hat{y})$). Katsayı (a) arttıkça dokuzuncu imge ($G_{DC}(z_1 + a * \hat{y})$) hedef görseline ($G_{DC}(z_2)$) yakınsamıştır. Katsayı (a) arttırıldıkça, anlamsal özelliğin belirginliği onuncu ve sonraki rakamlarda ($G_{DC}(z_1 + a * \hat{y})$) görüldüğü üzere hedef rakamındaki seviyeyi aşmaktadır.

Yalnızca 2 rakamı ile elde edilen yön vektörü diğer rakamlara uygulandığında Şekil 'de gösterilen çıktılara benzer başarılı sonuçlar üretilebilinmiştir. İlk satırın solunda, DCGAN ile üretilmiş 2 görseli ($G_{DC}(z_1)$), sağa doğru ise katsayı a ile ölçeklenerek üretilen imgeler ($G_{DC}(z_1 + a * \hat{y})$) gösterilmiştir. Ölçeklenen yön, Yön-Ayrıştırıcı tarafından bulunan yöndür. Aynı yön diğer satırlardaki ilk imgelere ölçeklenerek eklendiğinde, yönün etkisi ilk satırda gösterdiği etkiyle yakın olduğu görünmektedir. Bütün rakamlar aynı şekilde sağa doğru yaslanmaktadır. Hedef görselin anlamsal özelliği daha karmaşık bir dönüşümden kaynaklanıyorsa ise (Şekil), elde edilen yön vektörünün kodladığı bu transformasyon diğer rakamlarda, rakamların karakteristik bütünlüğünden ödün verebilmektedir. Şekil 'de verilen örnekte 2 rakamı referans alınmıştır. Böylece tayin edilen yön başlangıç görseline ölçeklenip kademeli olarak eklenmiştir. Bu işlemin uygulanmasıyla 2 rakamının dikey eksen etrafında dönmesi, yön vektörünün bu özelliği kodladığı görünümü vermektedir. Döndürme etkisi, diğer rakamların karakteristik bütünlüklerini kaybetmelerine sebep olabilmektedir. Bu bozulma arttıkça bir rakamın farklı bir rakama dönüşmesiyle sonuçlanmaktadır. Bu problemin kaynağı, üretici modelinin eğitimi için kullanılan veri kümesinde yeterince karmaşık transformasyonlara sahip veri örneklerinin bulunmaması olarak düşünülmektedir. Bahsi geçen deneyde, çıktıların ara değer kestirimleri (enterpolasyonları) ayrıntılı biçimde çizdirildiğinde belli noktalarda ani atlamalar ve mod çökmesi (mode collapse) görülmektedir.



Şekil 4.6 DCGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği sağa yaslanma özelliğinin diğer rakamlara başarılı şekilde uygulanması



Şekil 4.7 DCGAN ile oluşturulmuş 2 rakamı veri kümesiyle eğitilen modelin tahmin ettiği kalınlaştırma özelliğinin diğer rakamlara başarısız şekilde uygulanması

4.3 InfoGAN FreyFaces Yön-Ayrıştırıcısı



Şekil 4.8 InfoGAN ile rastgele üretilmiş yüz imgesine Yön-Ayrıştırıcı model ile bulunan yönün uygulanması



Şekil 4.9 InfoGAN ile rastgele üretilmiş yüz imgesine Yön-Ayrıştırıcı model ile bulunan yönün katsayı ile ölçeklendirilerek uygulanmasının örnekleri

InfoGAN ve FreyFaces ile oluşturulan eğitim veri kümesi ile eğitilmiş olan Yön-Ayrıştırıcı modeli ile test veri kümesindeki verilerle deneyler yapılmıştır. Şekil 'de solda referans görseli ($G_{IG}(z_1)$), ortada hedef görseli ($G_{IG}(z_2)$) ve sağda Yön-Ayrıştırıcı modelin ilk iki imgeyi kullanarak tahmin ettiği yön vektörü ($\hat{y}$) uygulanarak elde edilen görsel ($G_{IG}(z_1 + \hat{y})$) sunulmuştur. Şekilde görüldüğü gibi, yakınsanan yön vektörü ile oluşturulan görsel hedeflenen görsele benzemektedir. Bu, yön vektörünün doğru yakınsandığı anlamına gelmektedir ve ayrıştırıcı modelin, InfoGAN'ın gizli uzayını ayrıştırarak sadece iki imge kullanarak onların farkını tespit edebildiğini ispatlamaktadır.

Bulunan yön vektörü aynı referans görselinin gizli vektörüne ölçeklenerek eklendiğinde oluşan başlangıç, ara ve sonuç görselleri Şekil 'de paylaşılmıştır. Yön-Ayrıştırıcı modeli MNIST veri kümesiyle eğitilen üretici modellerin gizli uzaylarını ayrıştırmakta başarılı olduğu gibi, daha karmaşık bir veriden oluşan FreyFaces veri kümesiyle eğitilen bir üretici modelde de başarılı olmuştur.

5. SONUÇ

Bu tez çalışmasında, üretici ağların görselleri istenen anlamsal özellikleri sağlayacak biçimde kontrollü üretimde, önceden eğitilmiş GAN'ların gizli uzaylarını ayırıştırabilen yeni bir öz denetimli yaklaşım tasarlanmıştır. Bu yöntemle MNIST ve FreyFaces veri kümeleri ile deneyler yapılmıştır ve tasarlanan yöntem için nitel sonuçlar verilmiştir.

Geliştirilen yönteme girdi olarak iki görsel verildiğinde, görseller arasındaki anlamsal farklılığın gizli uzaydaki yönünü tespit eden bir Yön-Ayrıştırıcı ağ mimarisi sunulmaktadır. Önerilen yöntemde ResNet18 öznitelik çıkarma mimarisi ikiz ağ olarak kullanılmıştır. Etiketli veri kümesi gereksinimini gidermek için öz denetimli bir eğitim stratejisi kurgulanmıştır. Önerilen yöntem özelleştirilmiş eğitim stratejileri ile GAN eğitime veya tekrar eğitime ihtiyaç duymamaktadır. Bu sayede ön eğitilmiş herhangi bir GAN'a uygulanabilmektedir. Böylece var olan GAN'ların kullanılabilirliğini arttırmaktadır.

İki yaygın veri kümesi ile sonuçlar elde edilerek, bulunan yön vektörün GAN'a girdi olarak sağlanan vektöre uygulandığında farklı imgelerin kimliğini değiştirmeden onlara aynı özellikleri verebildiği gösterilmiştir. Deneysel çalışmalarda önceden eğitilmiş DCGAN ve InfoGAN üretici modelleriyle yöntemin işlevselliği ortaya konulmuştur. Yöntemin uygulandığı MNIST ve FreyFaces veri kümelerinden elde edilen sonuçlar üzerinden niteliksel yorumlama yapıldığında kurgulanan yöntemin başarımı incelenebilmektedir. Sonuçlar incelendiğinde yöntemin üretici ağın gizli uzayında yapılan değişiklikleri yakınsayarak gizli uzayı eşleştirilebildiği gözlemlenmektedir.

Yöntemin işlevselliği görece az karmaşıklığa sahip veri kümeleri ile teyit edilmiştir. Daha zorlayıcı bir kurgu için daha karmaşık veri kümeleri ile deneyler yapılması önerilmektedir. Bu bağlamda Yön-Ayrıştırıcı ağın karmaşık verilerin özniteliklerini daha iyi kapsayabilmesi için farklı öznitelik çıkarma mimarilerinden faydalanılabilir.

KAYNAKLAR

- Aktaş, A. A., Yalım Keleş, H., Askerzade, İ. 2022. "A Novel Visual Attribute Disentanglement Approach using Self-Supervision". İçinde *2022 30th Signal Processing and Communications Applications Conference (SIU)* (ss. 1-4).
- Alain, G., Bengio, Y. 2013. "What Regularized Auto-Encoders Learn from the Data Generating Distribution". Geliş tarihi gönderen <https://openreview.net/forum?id=-4IA4WgNAy4Wx>
- Bengio, Y., Courville, A., Vincent, P. 2013. "Representation Learning: A Review and New Perspectives". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8), 1798-1828.
- Bhagat, S., Udandaraao, V., Uppal, S., Anand, S. 2020. "DisCont: Self-Supervised Visual Attribute Disentanglement Using Context Vectors". İçinde A. Bartoli & A. Fusiello (Ed.), *Computer Vision – ECCV 2020 Workshops* (ss. 549-553). Cham: Springer International Publishing.
- Chen, X., Duan, Y., Houthoof, R., Schulman, J., Sutskever, I., Abbeel, P. 2016. "InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets". İçinde *Advances in Neural Information Processing Systems* (C. 29). Curran Associates, Inc. Geliş tarihi gönderen <https://proceedings.neurips.cc/paper/2016/hash/7c9d0b1f96aebd7b5eca8c3edaa19ebb-Abstract.html>
- Chlap, P., Min, H., Vandenberg, N., Dowling, J., Holloway, L., Haworth, A. 2021. "A review of medical image data augmentation techniques for deep learning applications". *Journal of Medical Imaging and Radiation Oncology*, 65(5), 545-563.
- Creswell, A., Bharath, A. A. 2019. "Inverting the Generator of a Generative Adversarial Network". *IEEE Transactions on Neural Networks and Learning Systems*, 30(7), 1967-1974.
- Deng, L. 2012. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]". *IEEE Signal Processing Magazine*, 29(6), 141-142.
- Dogan, Y., Keles, H. Y. 2020. "Semi-supervised image attribute editing using generative adversarial networks". *Neurocomputing*, 401, 338-352.
- Dogan, Y., Keles, H. Y. 2022. "Iterative facial image inpainting based on an encoder-generator architecture". *Neural Computing and Applications*, 34(12), 10001-10021.
- Elyan, E., Vuttipittayamongkol, P., Johnston, P., Martin, K., McPherson, K., Moreno-García, C. F., ... Mostafa Kamal Sarker, M. 2022. "Computer vision and

machine learning for medical image analysis: recent advances, challenges, and way forward.". Artificial Intelligence Surgery, 2.

- Goetschalckx, L., Andonian, A., Oliva, A., Isola, P. 2019. "GANalyze: Toward Visual Definitions of Cognitive Image Properties" (ss. 5744-5753). Program adı: Proceedings of the IEEE/CVF International Conference on Computer Vision. Geliş tarihi gönderen https://openaccess.thecvf.com/content_ICCV_2019/html/Goetschalckx_GANalyze_Toward_Visual_Definitions_of_Cognitive_Image_Properties_ICCV_2019_paper.html
- Gogna, A., Majumdar, A. 2016. "Semi Supervised Autoencoder". İçinde A. Hirose, S. Ozawa, K. Doya, K. Ikeda, M. Lee, & D. Liu (Ed.), *Neural Information Processing* (ss. 82-89). Cham: Springer International Publishing.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... Bengio, Y. 2014. "Generative Adversarial Nets". İçinde *Advances in Neural Information Processing Systems* (C. 27). Curran Associates, Inc. Geliş tarihi gönderen <https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>
- He, K., Zhang, X., Ren, S., Sun, J. 2016. "Deep Residual Learning for Image Recognition". İçinde *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (ss. 770-778). Las Vegas, NV, USA: IEEE.
- Higgins, I., Matthey, L., Pal, A., Burgess, C., Glorot, X., Botvinick, M., ... Lerchner, A. 2017. " β -VAE: LEARNING BASIC VISUAL CONCEPTS WITH A CONSTRAINED VARIATIONAL FRAMEWORK", 22.
- Hinton, G. E., Salakhutdinov, R. R. 2006. "Reducing the Dimensionality of Data with Neural Networks". *Science*, 313(5786), 504-507.
- Jahanian, A., Chai, L., Isola, P. 2020. "On the "steerability" of generative adversarial networks". Program adı: International Conference on Learning Representations. Geliş tarihi gönderen <https://openreview.net/forum?id=HylsTT4FvB>
- Karras, T., Aila, T., Laine, S., Lehtinen, J. 2022. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". Program adı: International Conference on Learning Representations. Geliş tarihi gönderen <https://openreview.net/forum?id=Hk99zCeAb¬eId=Hk99zCeAb&ref=https://githubhelp.com>
- Karras, T., Laine, S., Aila, T. 2019. "A Style-Based Generator Architecture for Generative Adversarial Networks" (ss. 4401-4410). Program adı: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Geliş tarihi gönderen https://openaccess.thecvf.com/content_CVPR_2019/html/Karras_A_Style-

Based_Generator_Architecture_for_Generative_Adversarial_Networks_CVPR_2019_paper.html

- Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T. 2020. "Analyzing and Improving the Image Quality of StyleGAN". İçinde *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (ss. 8107-8116). Seattle, WA, USA: IEEE.
- Kingma, D. P., Welling, M. 2014, Mayıs 1. "Auto-Encoding Variational Bayes". arXiv. Geliş tarihi gönderen <http://arxiv.org/abs/1312.6114>
- Lee, W., Kim, D., Hong, S., Lee, H. 2020. "High-Fidelity Synthesis with Disentangled Representation". İçinde A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Ed.), *Computer Vision – ECCV 2020* (ss. 157-174). Cham: Springer International Publishing.
- Lipton, Z., Tripathi, S. 2017. "Precise Recovery of Latent Vectors from Generative Adversarial Networks".
- Liu, B., Zhu, Y., Fu, Z., Melo, G. de, Elgammal, A. 2020. "OOGAN: Disentangling GAN with One-Hot Sampling and Orthogonal Regularization". *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 4836-4843.
- Liu, Z., Luo, P., Wang, X., Tang, X. 2015. "Deep Learning Face Attributes in the Wild" (ss. 3730-3738). Program adı: *Proceedings of the IEEE International Conference on Computer Vision*. Geliş tarihi gönderen https://openaccess.thecvf.com/content_iccv_2015/html/Liu_Deep_Learning_Face_ICCV_2015_paper.html
- Mirza, M., Osindero, S. 2014. "Conditional generative adversarial nets". İçinde *In NIPS 2014 Workshop on Deep Learning*.
- Paullada, A., Raji, I. D., Bender, E. M., Denton, E., Hanna, A. 2021. "Data and its (dis)contents: A survey of dataset development and use in machine learning research". *Patterns*, 2(11), 100336.
- Plumerault, A., Borgne, H. L., Hudelot, C. 2022. "Controlling generative models with continuous factors of variations". Program adı: *International Conference on Learning Representations*. Geliş tarihi gönderen <https://openreview.net/forum?id=H1laeJrKDB>
- Radford, A., Metz, L., Chintala, S. 2016, Ocak 7. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". arXiv.
- Ramesh, A., Choi, Y., LeCun, Y. 2019, Şubat 5. "A Spectral Regularizer for Unsupervised Disentanglement". arXiv.
- Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M. 2022, Nisan 12. "Hierarchical Text-Conditional Image Generation with CLIP Latents". arXiv.

- Reed, S. E., Zhang, Y., Zhang, Y., Lee, H. 2015. "Deep Visual Analogy-Making". İçinde *Advances in Neural Information Processing Systems* (C. 28). Curran Associates, Inc. Geliş tarihi gönderen <https://proceedings.neurips.cc/paper/2015/hash/e07413354875be01a996dc560274708e-Abstract.html>
- Ren, X., Yang, T., Wang, Y., Zeng, W. 2022. "Learning Disentangled Representation by Exploiting Pretrained Generative Models: A Contrastive Learning View". Program adı: International Conference on Learning Representations. Geliş tarihi gönderen <https://openreview.net/forum?id=j-63FSNcO5a>
- Shen, Y., Gu, J., Tang, X., Zhou, B. 2020. "Interpreting the Latent Space of GANs for Semantic Face Editing" (ss. 9243-9252). Program adı: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Geliş tarihi gönderen https://openaccess.thecvf.com/content_CVPR_2020/html/Shen_Interpreting_the_Latent_Space_of_GANs_for_Semantic_Face_Editing_CVPR_2020_paper.html
- Shorten, C., Khoshgoftaar, T. M. 2019. "A survey on Image Data Augmentation for Deep Learning". *Journal of Big Data*, 6(1), 60.
- Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G. 2021. "GAN-Control: Explicitly Controllable GANs" (ss. 14083-14093). Program adı: Proceedings of the IEEE/CVF International Conference on Computer Vision. Geliş tarihi gönderen https://openaccess.thecvf.com/content/ICCV2021/html/Shoshan_GAN-Control_Explicitly_Controllable_GANs_ICCV_2021_paper.html
- Tzelepis, C., Tzimiropoulos, G., Patras, I. 2021. "WarpedGANSpace: Finding Non-Linear RBF Paths in GAN Latent Space" (ss. 6393-6402). Program adı: Proceedings of the IEEE/CVF International Conference on Computer Vision. Geliş tarihi gönderen https://openaccess.thecvf.com/content/ICCV2021/html/Tzelepis_WarpedGANSpace_Finding_Non-Linear_RBF_Paths_in_GAN_Latent_Space_ICCV_2021_paper.html
- Ulyanov, D., Vedaldi, A., Lempitsky, V. 2017. "Adversarial Generator-Encoder Networks". *Proceedings of the AAAI Conference on Artificial Intelligence*, 32.
- Vincent, P., Larochelle, H., Bengio, Y., Manzagol, P.-A. 2008. "Extracting and composing robust features with denoising autoencoders". İçinde *Proceedings of the 25th international conference on Machine learning* (ss. 1096-1103). New York, NY, USA: Association for Computing Machinery.
- Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A. 2010. "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion", 38.

Voulodimos, A., Doulamis, N., Doulamis, A., Protopapadakis, E. 2018. "Deep Learning for Computer Vision: A Brief Review". Computational Intelligence and Neuroscience, 2018, e7068349.

Voynov, A., Babenko, A. 2020. "Unsupervised Discovery of Interpretable Directions in the GAN Latent Space". İçinde *Proceedings of the 37th International Conference on Machine Learning* (ss. 9786-9796). PMLR. Geliş tarihi gönderen <https://proceedings.mlr.press/v119/voynov20a.html>

