

T.C.

**İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ BİLİM DALI**

**DERİN ÖĞRENMEYE DAYALI GÜÇLÜ YÜZ TANIMA
SİSTEMİ İÇİN GAN İLE VERİ ÇOĞALTMA**

YÜKSEK LİSANS TEZİ

Erdal ALİMOVSKI

İstanbul

Aralık,2019

T.C.
İSTANBUL SABAHATTİN ZAİM ÜNİVERSİTESİ
BİLGİSAYAR BİLİMİ VE MÜHENDİSLİĞİ
YÜKSEK LİSANS PROGRAMI

DERİN ÖĞRENMEYE DAYALI GÜÇLÜ YÜZ TANIMA SİSTEMİ İÇİN
GAN İLE VERİ ÇOĞALTMA

YÜKSEK LİSANS TEZİ

ERDAL ALİMOVSKI

Tez Danışmanı

Dr. Öğr. Üyesi Yahya ŞİRİN

İstanbul
Aralık,2019

TEZ ONAY

Fen Bilimleri Enstitüsü Müdürlüğüne,

Bu çalışma, jürimiz tarafından Bilgisayar Mühendisliği Anabilim Dalı, Bilgisayar Bilimleri ve Mühendisliği Bilim Dalında Yüksek Lisans TEZİ olarak kabul edilmiştir.

Danışman Dr. Öğr. Üyesi Yahya ŞİRİN



Üye Dr. Öğr. Üyesi Gökhan ERDEMİR



Üye Dr. Öğr. Üyesi Mehtap YALÇINKAYA



Onay

Yukarıdaki imzaların, adı geçen öğretim üyelerine ait olduğunu onaylarım.



Prof. Dr. Ahmet Korhan BİNARK

Enstitü Müdürü

BİLİMSEL ETİK BİLDİRİMİ

Yüksek lisans tezi olarak hazırladığım “DERİN ÖĞRENMEYE DAYALI GÜÇLÜYÜZ TANIMA SİSTEMİ İÇİN GAN İLE VERİ ÇOĞALTMA” adlı çalışmanın öneri aşamasından sonuçlandığı aşamaya kadar geçen süreçte bilimsel etiğe ve akademik kurallara özenle uyduğumu, tez içindeki tüm bilgileri bilimsel ahlak ve gelenek çerçevesinde elde ettiğimi, tez yazım kurallarına uygun olarak hazırladığımı, bu çalışmamda doğrudan veya dolaylı olarak yaptığım her alıntıya kaynak gösterdiğimi ve yararlandığım eserlerin kaynakçada gösterilenlerden oluştuğunu beyan ederim.

Erdal Alimovski

ÖNSÖZ

Bu çalışmayı tamamlamamda bana yardımcı olan kişilere içten teşekkür ve saygılarımı sunarım. Öncelikle, bu tezin kalitesine katkıda bulunan önerileri, destekleri ve rehberliği için değerli tez danışmanım Dr. Öğr. Üyesi Yahya ŞİRİN'e teşekkür ederim. Ayrıca yüksek lisans eğitimim ve tez çalışmam sırasında benden desteklerini esirgemeyen Dr. Öğr. Üyesi Gökhan ERDEMİR'e teşekkür ediyorum. Son olarak, aileme ve arkadaşlarıma destekleri için teşekkür etmek istiyorum.

Erdal Alimovski

İstanbul 2019

ÖZET

DERİN ÖĞRENMEYE DAYALI GÜÇLÜ YÜZ TANIMA SİSTEMİ İÇİN GAN İLE VERİ ÇOĞALTMA

Erdal ALİMOVSKI

Yüksek Lisans, Bilgisayar Bilimi ve Mühendisliği

Tez danışmanı: Dr. Öğr. Üyesi Yahya ŞİRİN

Aralık-2019, 77 Sayfa

Yüz tanıma sistemi, kişiyi dijital bir görüntüden doğrulamak veya tanımlamak için kullanılan biyometrik sistem türüdür. Yüz tanıma sistemleri güvenlik amaçlı kullanılmalarının yanı sıra eğitim, sağlık ve buna benzer birçok alanda da kullanılırlar. Literatür’de geçmişten günümüze birçok yüz tanıma yöntemi önerilmiştir. Bunları iki gruba ayırabiliriz. Geleneksel yöntemler ve Makine öğrenmesi veya Derin öğrenme (DÖ) tekniklerine dayalı modern yöntemler.

Geleneksel yüz tanıma yöntemleri çeşitli poz, aydınlatma, tıkanıklık vs. gibi farklı koşullar altında çekilmiş görüntülerde oldukça düşük oranda performans göstermektedirler. Bu oranı iyileştirmek amacıyla modern yüz tanıma yöntemleri önerilmiştir. Modern yüz tanıma yöntemleri DÖ tekniklerine dayanmaktadır. DÖ’ye dayalı yüz tanıma sistemleri büyük miktarda veri ile eğitildikleri için yüksek doğruluk performansı sergilemektedirler. DÖ modelleri ne kadar fazla veri ile eğitilirse, sistemin doğruluk performansının artma olasılığı o kadar fazla olur.

Çalışmamız iki bölümden oluşmaktadır. İlk bölümde DÖ’ye dayalı yüz tanıma sistemi oluşturulacaktır. Yüz tanıma sistemimiz sırasıyla: tespit aşaması, hizalama aşaması, yüze ait 128 temsillin (embeddings) üretilmesi, sınıflandırma aşaması, doğrulama ve kümeleme olarak altı ana aşamadan oluşmaktadır

İkinci bölümde, yüz tanıma sistemimizin doğruluk performansında veri artırma tekniklerinin etkisini analiz etmek için Derin Evrimsel Çekişmeli Üretici Ağlar (Deep Convolutional Generative Adversarial Networks) ile sentetik yüzler üretilenmiştir. Her iki bölümde gerçekleştirilen eğitim ve test işlemlerinde LFW veri seti kullanılmıştır.

Test işlemlerinden elde edilen sonuçlar göre, önerilen veri artırma tekniđi sonuca olumlu yansımış ve yüz doğrulamada %2.00, yüz sınıflandırmada ise %2.26'lık bir artış elde etmiştir.

Ancak DEÇÜA modeli küçük veri seti ile eğitildiğinden dolayı, önerilen veri artırma tekniğinin etkisi beklendiğİ oranda olmamıştır. Aynı model daha büyük veri seti ile eğitildiğİ takdirde yüz tanıma sisteminde etkisinin daha yüksek olacağını tahmin ediyoruz.



Anahtar Kelimeler: Yüz tanıma, derin öğrenme, çekişmeli üretici ağlar, makine öğrenmesi, veri çoğaltma.

ABSTRACT
**DATA AUGMENTATION WITH GAN FOR ROBUST FACE
RECOGNITION SYSTEM BASED ON DEEP LEARNING**

Erdal ALİMOVSKI

Master of Science, Computer Science and Engineering

Advisor: Asst. Doc. Dr. Yahya ŞİRİN

December-2019,77 Pages

Since 1970's, facial recognition is one of the most studied subjects in image processing and biometry. Face recognition technology aims to perceive and identify faces in images in a manner similar to the operation of human visual system in computers. With the development of these systems, it is directed to build a more reliable and easier to live world. Many techniques of facial recognition have been proposed since its first development but nowadays best performing systems are based on Deep Learning. The high performance of deep learning based facial recognition systems is mainly dependent on the size of the datasets in which the model and classifier are trained. Therefore, the main purpose of this study is to generate synthetic faces using DCGANs in order to enlarge the dataset and analyze the effects of facial recognition system on verification and classification tasks. In order to perform the analysis properly, it is therefore essential to establish a robust face recognition system. In this case, another objective of our study is to build a robust face recognition system. Experimental results shows that the proposed data augmentation technique increases the accuracy of face recognition system. In the face verification process %2.00 increment was obtained. Whereas in classification tasks 2.26%.

Key Words: Face recognition, deep learning, generative adversarial networks, machine learning, data augmentation.

İÇİNDEKİLER

TEZ ONAY	i
BİLİMSEL ETİK BİLDİRİMİ	ii
ÖNSÖZ	iii
ÖZET	iv
ABSTRACT	vi
İÇİNDEKİLER	vii
ŞEKİL LİSTESİ	ix
TABLO LİSTESİ	xi
KISALTMALAR	xii
BÖLÜM 1	1
GİRİŞ	1
1.1. Problemin tanımı	3
1.2. Tezin amacı	3
1.3. Tezin Genel Görünümü	4
BÖLÜM 2	5
İLGİLİ ÇALIŞMALAR	5
2.1. Geleneksel Yöntemler	5
2.1.1 Özellik Tabanlı Metot	5
2.1.2 Bütünsel Karşılaştırmalı Metot	6
2.3. Modern Yöntemler	7
2.4. Veri Çoğaltma Teknikleri.....	9
BÖLÜM 3	11
DERİN ÖĞRENME’NİN TEORİK ARKA PLANI	11
3.1. Denetimli Öğrenme	11
3.2. Denetimsiz Öğrenme	11
3.2 Yapay Sinir Ağları (Artificial Neural Networks).....	12
3.2.1. Yapay Sinir Ağları'nın Yapısı	12
3.2.2. Yapay Sinir Ağları'nın Çalışma Prensibi	14
3.2.3. Yapay Sinir Ağları'nın Eğitilmesi.....	18
3.3 Derin Öğrenme	24
3.3.1. Derin Sinir Ağlar	25

3.3.2. Evrişimsel Sinir Ağlar	26
3.4. Üretken Modeller (Generative Models)	28
3.4.1. Çekişmeli Üretici Ağlar	28
3.4.2. Çekişmeli Üretici Ağlar'ın zorlukları/dezavantajları	30
3.4.3. Alternatif Çekişmeli Üretici Ağlar	30
BÖLÜM 4.....	33
METODOLOJİ	33
4.1. Sentetik Yüz Üretimi	33
4.2. Yüz Tespit Etme	35
4.3. Yüz Hizalama	38
4.4. 128-B Yüz Temsillerinin Üretilmesi	41
4.5. Doğrulama, Sınıflandırma ve Kümeleme	43
BÖLÜM 5.....	45
DENEYLER, SONUÇ VE TARTIŞMA	45
5.1. Araştırma Araçlar	45
5.1.1. Tensorflow	46
5.1.2. Keras	46
5.2 Veri Seti.....	46
5.2.1. Orijinal LFW veri seti.....	46
5.2.2. DEÇÜA ile çoğaltılmışı veri seti	47
5.3. LFW veri setinde Yüz Doğrulama	48
5.4. LFW veri setinde Yüz Sınıflandırma	49
5.5. Yüz kümeleme.....	51
5.6. Sonuç ve Tartışma	52
5.6.1. Sistemin Doğruluk performansı.....	52
5.6.2. Sonuçların karşılaştırılması.....	53
5.7 Zorluklar ve Sınırlamalar	55
BÖLÜM 6.....	56
SONUÇ VE GELECEKTEKİ ÇALIŞMALAR	56
6.1. Sonuç	56
6.2. Gelecekteki Çalışmalar.....	57
ÖZGEÇMİŞ	63

ŞEKİL LİSTESİ

Şekil 3. 1. Perseptronun yapısı.	12
Şekil 3. 2. Çok katmanlı bir YSA'nın temel yapısı.	14
Şekil 3. 3. Biyolojik sinir hücresi parçaları.	15
Şekil 3. 4. Sigmoid fonksiyonunun grafiksel karşılığı.	16
Şekil 3. 5. Tanjant fonksiyonunun grafiksel karşılığı.	17
Şekil 3. 6. ReLU fonksiyonunun grafiksel karşılığı.	18
Şekil 3. 7. Gradient Descent algoritmasının örneği. (Hochreiter, Younger ve Conwell 2001).	20
Şekil 3. 8. En dik iniş ve konjüğe gradyan yöntemlerinin arama yolları (Chan ve Ewins, 2011).	20
Şekil 3. 9. Optimizasyon algoritmalarının hız ve hafıza açısından karşılaştırılması (Quesada, 2019).	22
Şekil 3. 10. Sinir ağının her katmandan özellikler nasıl öğrendiğinin bir örneği. (Ian G ve arkd. 2016)	25
Şekil 3. 11. ESA 'ın yapısı. Görüntülerin boyutu, tam bağlı katmana varıncaya kadar düşmektedir.	27
Şekil 3. 12. ÇÜA mimarisi.	29
Şekil 3. 13. ÇÜA çalışma prensibi'nin basit örneği.	29
Şekil 3. 14. DEÇÜA'da kullanılan Üretici Ağı (Radford ve arkd. 2015).	32
Şekil 3. 15. DEÇÜA tarafından LSUN veri seti kullanılarak üretilen yatak odası örnekleri (Radford, Metz ve Chintala, 2016).	32
Şekil 4. 1. DEÇÜA modelin akış şeması.	35
Şekil 4. 2. Yüze ait bir görüntünün ana özelliklerinin HOG yöntemi ile gösterimi. (Felzenszwalb vd., 2009).	37
Şekil 4. 3. Orijinal görüntüden yüz kısmının algılanması ve sınırlayıcı kutu içerisinde alınmasının gösterimi.	38
Şekil 4. 4. iBUG 300-W veri setinden üretilen 68 yüz işaretinin görseli.	39
Şekil 4. 5. Görüntüdeki yüze affın dönüşümü işleminin uygulanma görseli. (Amos, Ludwiczuk ve Satyanarayanan, 2016).	40
Şekil 4. 6. Algılanan yüz resminin yüz noktalarına göre hizalandırılmasının görseli.	40
Şekil 4. 7. 128 yüz temsillinin görseli.	42
Şekil 4. 8. Üçlü kayıp: Öğrenme sürecinden önce ve sonra.	43

Şekil 5. 1. LFW veri seti örnekleri.....	47
Şekil 5. 2. DEÇÜA ile üretilen en iyi sentetik yüzler.	48
Şekil 5. 3. Orijinal LFW veri setinde sistemin yüz doğrulama peformansı gösterilmektedir.....	49
Şekil 5. 4. DEÇÜA ile çoğaltılmış veri setinde sistemin yüz doğrulama peformansı gösterilmektedir.....	49
Şekil 5. 5. Yüz tanıma sisteminin doğru olarak tanıdığı kişiler.....	50
Şekil 5. 6. Yüz tanıma sistemi tarafından hatalı olarak tanınan kişiler.....	51
Şekil 5. 7. LFW veri setinde temsillerin görsel dağılımı.	52
Şekil 5. 8. Çalışmada uygulanan yüz tanıma sistemi ile Geleneksel yöntemlere dayalı yüz tanıma sistemlerinin karşılaştırılması.	54



TABLO LİSTESİ

Tablo 5.1: Yüz doğrulama işleminde elde edilen sonuçların karşılaştırılması.....	53
Tablo 5.2: Yüz sınıflandırma işleminde elde edilen sonuçların karşılaştırılması.....	53
Tablo 5.3: Uygulanan yüz tanıma sisteminin diğer sistemler ile karşılaştırılması.....	54



KISALTMALAR

DÖ- Derim Öğrenme

YSA-Yapay Sinir Ağları

TBA-Temel Bileşenler Analizi

ESA- Evrimsel Sinir Ağları

DSA-Derin Sinir Ağlar

ÇÜA-Çekişmeli Üretici Ağlar

DEÇÜA-Derin Evrimsel Çekişmeli Üretici Ağlar



BÖLÜM 1

GİRİŞ

Son zamanlarda teknolojinin hızlı gelişimi ile günlük veri üretiminde ciddi artış olduğu görülmektedir. Bu artış, işlemcilerin güç ve hız gelişimi ile birleştirildiğinde araştırmacılar için güçlü yüz tanıma sistemleri geliştirmek için fırsatlar doğurmuştur. Yüz tanıma konusu, 1970'li yıllardan başlayarak günümüze kadar görüntü işleme ve biyometride en çok incelenen konulardan biridir. Yüz tanıma teknolojisi, görüntüdeki yüzleri, bilgisayarlara insan görsel sisteminin işleyişine benzer bir şekilde görmeyi ve tanımlamayı amaçlamaktadır. Bu sistemlerin geliştirilmesiyle daha güvenilir ve yaşanması daha kolay bir dünya inşa etmek amaçlanmaktadır.

İlk geliştirilmeye başladıklarından günümüze kadar yüz tanıma sistemlerinin doğruluk performansında çok ciddi bir artış gözlemek mümkündür. Son dönemde yapılan çalışmalara baktığımızda geleneksel olarak kullanılan parmak izi ve iris tanıma gibi biyometrik yöntemlerin yerine yüz tanıma yöntemleri tercih edilmeye başlanmıştır. Yüz tanıma sistemlerini diğer yöntemlere göre daha çekici kılan bir tarafı da müdahaleci olmayan doğasıdır. Örneğin, parmak izi tanımada, kullanıcıların parmaklarını bir algılayıcıya yerleştirmeleri gerekmektedir. İris tanıma için kullanıcıların bir kameraya iyice yaklaşmaları gerekir. Aynı şekilde ses tanıma kullanıcıların yüksek sesle konuşmalarıyla mümkün olabilmektedir. Buna karşılık olarak, modern yüz tanıma sistemleri, kullanıcıların yalnızca kameranın (kameradan makul bir mesafede olmaları şartıyla) görüş alanı içerisinde olmaları yeterlidir. Bu durum, yüz tanıma sistemlerini en kullanışlı biyometrik yöntem yapmaktadır. Ayrıca bu durum, yüz tanıma uygulamalarının potansiyeli daha geniş olduğu anlamına gelmektedir. Çünkü kullanıcıların herhangi bir gözetleme sistemi ile temas halinde olmasının istenmediği durumlarda da kullanılabilirler. Günümüzde yüz tanıma sistemleri güvenlik, sağlık, eğitim, parkende, etkinlik ve eğlence olmak üzere birçok farklı alanda kullanılmaktadırlar.

Literatürde önerilen birçok yüz tanıma yöntemi vardır. Tezimizde bunları iki gruba ayırıyoruz; DÖ veya Makine Öğrenmesi tekniklerine dayalı modern yöntemler ve geleneksel yöntemler.

Geleneksel yöntemler genellikle geometrik yaklaşıma dayanmaktadırlar. Bu yöntemler yüzün belirgin yerlerinde ki özelliklerden yüzü tanımaya çalışırlar. Geleneksel yöntemlerde en meşhur ve en fazla kullanılan algoritmalar arasında Eigenfaces (Belhumeur, Hespanha ve Kriegman 1997) ve Fishervector (Matthew ve Alex, 1991) algoritmaları bulunmaktadır. Her ne kadar geleneksel yöntemler kısıtlanmış ortamda iyi performans gösteriyor olsalar da, kısıtlanmamış ortamlarda iyi performans gösteremezler. Bunun nedeni ise bu yöntemlerde yüze ait özelliklerin araştırmacılar ve uzmanlar tarafından seçilmesidir. Dolayısıyla yapılan seçimlerin bazıları doğru, bazıları ise eksik olabilmektedirler. Bu tarz sorunları çözmek için daha sağlam yöntemler geliştirilmiştir.

Yukarıda açıklanan zorlukları çözmeye daha kolay çözümler ve daha başarılı sonuçlar sunmasından dolayı son zamanlarda, geleneksel yüz tanıma yöntemleri yerine modern yöntemler ön plana çıkmıştır. Modern yöntemler derin öğrenme tekniklerine dayanırlar. Geleneksel yöntemlerinin aksine, derin öğrenmeye dayalı yöntemlerde özellikler (features) veya temsiller (embeddings), model tarafından verilerden otomatik olarak seçilir. Derin öğrenme yöntemlerinin temel avantajı, verilerden en iyi özellikleri elde etmek için çok büyük veri setleri ile eğitilebilir olmalarıdır. Büyük veri setleri ile eğitilen Derin öğrenme yöntemleri, eğitim sırasında farklı varyasyonlarda olan görüntülerdeki özellikleri ve temsilleri öğrenebildikleri için bu yöntemler yüksek doğruluk oranları elde ederler. Buradan esinlenerek biz de modelimizi derin öğrenmeye dayalı Evrimsel Sinir Ağları ile oluşturacağız.

Derin öğrenme modellerinin başarı performansının yüksek olması eğitildikleri veri setlerinin büyüklüğüne bağlıdır. Günümüzde en iyi derin öğrenmeye dayalı yüz tanıma sistemleri milyonlarca görüntü ile eğitilmişlerdir. Bu derece büyük veri setlerini sıfırdan oluşturmak neredeyse imkânsızdır. Bu durum araştırmacıları çeşitli veri artırma teknikleri ve üretken modeller geliştirmeye itmiştir. Veri artırma teknikleri ve üretken modeller hakkında yapılmış çalışmalar detaylı bir şekilde 2.Bölümde ele alınmıştır. Verileri çoğaltmak için kullandığımız yöntem ise 4. Bölümde detaylı açıklanmıştır.

1.1. Problemin tanımı

İnsan görsel sistemi milisaniyeler içinde yüzleri kolaylıkla algılayabilmekte ve tanımlayabilmektedir. Ancak bilgisayarlar görüntü üzerinde objeyi tanıma gibi bir yeteneğe sahip değildirler. Bu tür bir yeteneğe sahip olmaları için bilgisayarın adım adım eğitilmesi gerekmektedir. Yüz tanıma, görüntü işleme ve görüntü analizi araştırmacıları için her zaman zor bir problem olmuştur. Geleneksel yöntemlerin, farklı aydınlatma, tıkanıklık ve yüz ifadeleri gibi durumlarda düşük performans elde ettikleri gözlenmektedir. Derin öğrenme yöntemlerinin son zamanlarda kullanımı, çeşitli poz, aydınlatma, tıkanıklık vs. gibi farklı koşullar altında yüz tanıma performansını önemli ölçüde artırmıştır. Ancak yüz tanıma, araştırmacılar için hala zor bir problem olarak görülmektedir. Çünkü bu sistemleri doğru şekilde oluşturmak (eğitmek) için elimizde geniş bir veri seti ve sağlam hesaplama gücüne sahip bir donanım gerekmektedir. Geniş veri seti oluşturmak zor ve aynı zamanda uzun süreçler gerektiren bir durumdur. Derin öğrenme modelinin çok iyi eğitilmiş olabilmesi için milyonlarca görüntünün gerekeceğini varsayarsak böyle bir veri setini sıfırdan oluşturmak bir hayli zordur. Günümüzde var olan en iyi (state-of-the-art) yüz tanıma yöntemleri milyonlarca görüntü ile eğitilmişlerdir. Milyonlarca görüntü içeren bu veri setleri internet ortamında açık kaynak olarak henüz paylaşılmamışlardır. İnternet ortamında açık kaynak olarak paylaşılan yüz veri setlerinin görüntü sayısı yüzbinleri bulmaktadır. Dolayısıyla milyonlarca görüntüye sahip veri setleri ile kıyasladığımızda açık kaynak yüz veri setlerinin görüntü sayısı oldukça düşüktür. Buradan yola çıkarak, Çekişmeli Üretici Ağlar yöntemi ile sentetik yüzler üretilen veri setimizdeki görüntü sayısını arttırarak derin öğrenme modelimizi daha fazla veriyle eğitip doğruluk performansında kayda değer bir artış hedeflemekteyiz.

1.2. Tezin amacı

Çalışmamız iki ana aşamadan oluşmaktadır. İlk aşamada DÖ'ye dayalı yüz tanıma sisteminin oluşturulması amaçlanmıştır. İkinci aşamada ise yüz tanıma sistemimizin doğruluk performansında veri artırma tekniklerinin etkisini analiz etmek için Derin Evrimsel Çekişmeli Üretici Ağlar (Deep Convolutional Generative Adversarial Networks) ile sentetik yüzler üretilecektir. Buna göre çalışmamızın ana amacı veri artırma tekniklerinin yüz tanıma sistemlerinde etkisini analiz etmektir. Ancak analizin

düzgün bir şekilde yapılması için sağlam bir yüz tanıma sistemi oluşturmak zaruridir. Bu sebeple bir diğer amacımız da sağlam yüz tanıma sistemi oluşturmak olacaktır.

1.3. Tezin Genel Görünümü

Bu çalışma altı farklı bölüme ayrılmıştır:

Bölüm 1: Bu bölümde tezin tanıtımı, giriş ve amacı ele alınmıştır.

Bölüm 2: Bu bölüm önceden yapılmış çalışmalar hakkında bilgi vermektedir.

Bölüm 3: Teorik olarak YSA, Derin Öğrenme ve ÇÜA'lar hakkında bilgi verildiği bölümdür.

Bölüm 4: Bu bölümde, tez projesinde kullanılan tasarım ve metodoloji açıklanacaktır.

Bölüm 5: Test Sonuçlarının ele alındığı bölüm.

Bölüm 6: Son bölüm, genel sonuç yorumu ve gelecekteki yapılacak çalışmaları içermektedir.

BÖLÜM 2

İLGİLİ ÇALIŞMALAR

Bu bölüm’de, yüz tanıma ve veri artırma teknikleri hakkında yapılmış çalışmalar incelenecektir. İnceleme aşaması, çalışmamız için kullanılacak en iyi yöntemleri tercih etme konusunda bize oldukça yararlı olmuştur. Aşağıda ilgili çalışmaları, geleneksel yöntemler, modern yöntemler ve veri artırma teknikleri başlıkları altında ele alınacaktır.

2.1. Geleneksel Yöntemler

Yüz tanıma ile ilgili uzun yıllar aktif araştırmalar yapılmıştır. Bilimsel çalışmalara bakıldığında yüz tanıma probleminin başlangıçta düşünülenenden daha zor olduğu hem fikirlik hâkimdir. Yüz tanıma sistemleri, yaş, poz, yüz ifadeleri, aydınlatma ve yüz bozulmaları gibi bir görüntüdeki yüz değişikliklerine karşı dayanıklı olmalı ve aynı zamanda iki farklı insanın yüzlerini birbirinden ayırt edebilmelidirler. Ondokuzuncu yüzyılın sonlarına kadar geliştirilen birçok teknik ve yöntem vardır. Genel olarak 1970’li yıllar’dan 2000’li yılların sonlarına kadar geliştirilen teknik ve yöntemleri iki gruba ayırıyoruz: Özellik tabanlı yöntemler (feature-based methods) ve Bütünsel Karşılaştırmalı Yöntemler (holistic matching methods).

2.1.1 Özellik Tabanlı Metot

Özellik tabanlı yaklaşımlar, kaş, göz, burun, çene vs. gibi yüz özelliklerinin birbirleri arasındaki mesafeleri hesaplayarak çalışmaktadırlar. Elde edilen ağırlıklar özellik vektörü olarak kaydedilmektedirler.

Yüz tanıma üzerinde ilk çalışmalardan biri 1973 yılında yapılmıştır. Araştırmacılar, 16 yüz parametresini içeren bir vektör elde etmek için basit görüntü işleme teknikleri uygulamışlardır. Yüz tanıma işlemi için Öklid mesafe hesaplama tekniğini kullandılar ve iki kişiye ait 40 görüntü içeren veri setinde yaklaşık % 75 doğruluk performansı elde ettiler (Kanade, 1973).

1992 yılında, Kanade’nin çalışmasından esinlenerek, araştırmacılar vektörün geometrik boyutunu 35’e çıkararak 188 görüntüden oluşan veri setinde %90 oranında doğruluk performansı elde etmişler (Brunelli ve Poggio, 1992).

Birkaç yıl sonra, bir grup arařtırmacı yukarıda bahsedilen iki bilimsel alıřmada kullanılan küçük veri setlerine nispeten 658 görüntüden oluşan veri setinde yüz tanıma sistemlerini test ettiler. Mevcut alıřmada arařtırmacılar geometrik vektörü oluşturmak için yüze ait 35 özelliđi diđer alıřmalarda olduđu gibi manuel olarak ıkarıp %95 oranında dođruluk performansı yakaladılar (Cox, Ghosn ve Yianilos, 1996).

Literatüre baktığımızda, özellik tabanlı yüz tanıma ile ilgili yapılmıř birok alıřma daha mevcuttur. Ancak arařtırma ařamasında, özellik tabanlı yöntemler için ana sorunların aynı kaldığı tarafımızca gözlemlenmiřtir. Özellik tabanlı yaklařımlarda yüz özellikleri manuel olarak ıkartılır. Dolayısıyla arařtırmacının, hangi özelliđin yüz tanımda daha etkili olduđuna karar vermesi gerekmektedir. Bu durum birok alıřmada ana zorluk olarak gösterilmiřtir.

2.1.2 Bütünsel Karřılařtırmalı Metot

Bütünsel karřılařtırmalı yaklařımlar, yukarıda belirtildiđi gibi yüzün yerel özelliklerine odaklanmak yerine, yüzün genel özelliklerine odaklanarak tanıma iřlemini gerekleřtirirler. Bütüncül Karřılařtırmalı yaklařımlar, genellikle istatistiksel yaklařımlar olarak da bilinirler.

İstatistiksel yaklařımlar, görüntüyü 2B deđerler dizisi olarak gösterirler. Daha sonra bu deđerleri dođrudan veri kümesindeki görüntülerle karřılařtırrlar. Bu yöntem, kısıtlanmıř ortamlarda iyi performans sergilemektedir. Ancak kısıtlanmamıř ortamlarda iyi performans sergileyememektedir (Huang, 1998).

1987 yılında arařtırmacılar Temel Bileřenler Analizi (Principal component analysis) kullanarak istatistiksel yöntemin boyutlarını ekonomik bir düzeye düşürecek tekniđi geliřtirdiler. Yayınladıkları bilimsel alıřmada arařtırmacılar, TBA'nın öz görüntüler koordinat uzayında herhangi bir yüzü etkin řekilde temsil edebileceđini, dolayısıyla yüz öz görüntüler veya izdüşümlerin bir kombinasyonu ile kolayca yeniden oluşturulabileceklerini kanıtlamıřtır (Sirovich ve Kirby, 1987).

İlerleyen yıllarda (Sirovich ve Kirby, 1987) alıřmasından esinlenerek, yeni bir yüz tanıma sistemi geliřtirildi (Turk Ve Pentland,1991). Bu alıřmadaki ana fikir, izdüşümleri öz görüntüler üzerinde sınıflandırma özellikleri olarak kullanarak yüzlerin tanınmasıydı. Geliřtirdikleri sistem, kiři baři 156 görüntü olmak üzere toplamda 2500 görüntüden oluşan veri tabanında test edilmiř ve o yıllarda en iyi performans gösteren

sistemler arasında yerini almıştır. İlerleyen dönemlerde araştırmacılar oluşturdukları bu yüz tanıma sistemini geliştirip daha büyük veri setleri üzerinde test ettiler. Sistemin geliştirilmesiyle birlikte performansta iyileşme gerçekleşmiştir. Kişiyeye ait tek görüntüden oluşan veri setleri ile sistem iyi performans sergilemiştir. Veri setinde aynı kişiyeye ait birden fazla görüntü olduğunda, TBA gerek duyulmayan değişkenlerini tuttuğu için sistemin performansı düşmektedir.(Turk ve Pentland, 1991)

Fisher yüz (Fisherfaces) metodu Doğrusal Ayırtaç Analizi (Linear Discriminant Analysis) esas alınarak geliştirildi. Araştırmacılar bu metodun düşük ve yüksek aydınlatma, farklı yüz ifadeleri, değişik pozlar vs. gibi çeşitli koşullar altında test edildiğinde TBA'ya göre daha iyi performans sergilediğini bildirmişlerdir (Adini, Moses ve Ullman, 1994).

Literatürde, Fisher yüz metodunda var olan eksiklerini tamamlamak için geliştirilen birçok yöntem ve teknik mevcuttur. Ancak tezimizde sınırlı alan olduğu için bu yöntemler ve teknikler hakkında ayrıntı sunamıyoruz. Bahsi geçen yöntemler için kapsamlı açıklama (Rabia ve Hamid, 2009) çalışmasında bulunabilir.

2.3. Modern Yöntemler

Modern yöntemler Yapay Sinir Ağları (YSA) ve Makine Öğrenmesi teknikleri kullanarak yüz tanıma işlemini gerçekleştirirler. Geleneksel yöntemler ile kıyaslandığında modern yöntemlerin çok daha iyi performans sergilediği apaçiktir. Özellikle 2010 yılından bu yana, DÖ tabanlı yüz tanıma sistemlerin doğruluk performansında muazzam artışlar olmuştur. Bu bölümümüzde mevcut yöntemler ile yapılmış en iyi çalışmaları (state-of-the-art) ele alacağız.

Bir ilk olarak (Lawrence vd., 1997) çalışmasında Evrişimsel Sinir Ağlar (Convolutional neural networks) kullanılarak yüz tanıma sistemi geliştirilmiştir. O zamandan beri geliştirilen birçok yüz tanıma sistemi için yukarıdaki çalışmayı referans olmuştur.

Bir grup araştırmacı Facebook araştırma laboratuvarında DeepFace adlı yüz tanıma sistemini geliştirdiler (Taigman vd., 2014). Mevcut yüz tanıma sistemi derin mimariye sahipti. Araştırmacılar 120 milyon bağlı ağırlık (weight) olmak üzere 9 katman'dan oluşan bir sinir ağı kullandılar. Evrişimsel Sinir Ağlarını (ESA), 4.4 milyon etiketli görüntüden (4000 farklı birey) oluşan bir veri seti ile eğitmişlerdir. Dolayısıyla ESA'nı özellik çıkarıcı olarak kullanılmıştır. DeepFace %97.35 doğruluk oranıyla o zamanda

en iyi performans gösteren yüz tanıma sistemi olarak literatüre geçmiştir. O zamandan günümüze kadar araştırmacılar DeepFace adlı sistemi geliştirmiş ve DeepID - 2, 2+ ve 3 olmak üzere çeşitli formlar sunmuşlardır. Bu geliştirilmiş formlar LFW (Huang vd., 2008) veri setinde test edildiklerinde %99.47 oranında başarı elde ettikleri gözlemlenmektedir.

Diğer bir çalışmada mimarisi ESA'ya dayanan Derin Yüz Tanıma (Deep Face Recognition) sistemi önerilmiştir (Parkhi, Vedaldi ve Zisserman, 2015). Derin yüz tanıma sistemi, DeepFace'e göre daha düşük veri seti ile eğitilmiş ve LFW veri setinde %98.95 oranında doğruluk elde etmiştir. Bu denli başarılı performans sergilemesindeki ana neden kullanılan ESA'nın yapısında öğrenebilen yerel süzgeçlerin mevcut olmasıdır.

Günümüzün en popüler yüz tanıma sistemlerinden olan Facenet, 2015 yılında facebook'un yapay zekâ ile ilgilenen araştırmacılar tarafından oluşturulmuştur (Schroff, Kalenichenko ve Philbin, 2015). Mevcut sistem, yüz görüntülerini doğrudan bir Öklid uzayına eşlemeyi (mapping) öğrenir. Daha sonra Facenet temsillerini (embeddings) özellik vektörü olarak kullanarak yüz doğrulama, tanıma ve kümeleme işleri gerçekleştirilir. Facenet 8 milyon farklı bireye ait toplamda 200 milyon görüntüye sahip veri seti ile eğitilmiştir. Facenet mimarisi DeepFace mimarisine çok benzemektedir. Bununla beraber, Facenet, aynı kişiye ait iki yüz görüntüsünün Öklid mesafesini en aza indirir, farklı kişiye ait üçüncü görüntü ile aynı kişiye ait ilk iki görüntü arasında Öklid mesafesini ise maksimuma çıkarır. Bu yöntem %99.63 başarı oranıyla günümüzde LFW veri setinde test edilen çalışmalar arasında en yüksek performansı sergilemiştir.

2016 yılında OpenFace adlı yüz tanıma kütüphanesini yayınladı (Amos, Ludwiczuk ve Satyanarayanan, 2016). OpenFace, yüz tanıma amaçlı genel bir kütüphanedir. Araştırmacıların Facenet mimarisini kullanarak tipik bir işlemci (CPU) üzerinde uygulanacak kadar hafif bir yüz tanıma sistemi geliştirdiler. Bu yüz tanıma sistemi, internet üzerinde açık kaynak olarak bulunan Facescrub (Hong ve Winkler, 2014) ve CasiaWeb Face (Yi vd., 2014) veri setleri ile eğitilmiştir. Bahsi geçen iki veri setinin içerdiği görüntü sayısı 0.6M civarındadır. Milyonlarca görüntü ile eğitilen sistemlere göre daha küçük veri seti ile eğitilmesine rağmen LFW veri setinde %92.93 başarı oranı elde etmiştir. Daha sonra, mevcut sistemin önceden eğitilmiş (pre-trained) ağırlıkları internet ortamında açık kaynak olarak paylaşılmıştır.

Yukarıda bahsettiğimiz en iyi ve en popüler yöntemlerin yanı sıra, literatürde yapılmış birçok çalışma daha mevcuttur. Aşağıda bu çalışmalar hakkında kısaca bilgi verilecektir.

Derin öğrenmeye dayalı yüz tanıma yöntemlerinde göze çarpan bir diğer sistem WebFace yüz tanıma sistemidir (A.g.e.). Araştırmacılar, (Simonyan ve Zisserman, 2014) ve (Szegedy vd., 2014) çalışmalarından esinlenerek 10 evrişimsel, 5 havuzlama ve 2 tam bağlı katman olmak üzere toplamda 17 katman derinliğinde bir mimari oluşturdular. Aynı mimari CasiaWebFace veri seti ile eğitilerek LFW veri setinde %97.73 oranında başarı elde etti. Ayrıca günümüzde oldukça popüler olan Casia Web Face veri setin bu çalışmada kullanılmak amacıyla oluşturulmuştur. Sonrasında açık kaynak olarak paylaşılmıştır. Diğer bir çalışmada araştırmacılar temsil tabanlı (embedded based) yüz tanıma performansını iyileştirmek için orijinal yüz görüntüsünü ve görüntünün aynasını birleştirdiler, böylece yüz tanıma performansında kayda değer bir artış elde ettiler (Xu vd., 2014). Yüz tanıma sistemleriyle alakalı yapılan bir diğer çalışmada, yüz görüntüsü ağız, gözler ve burun olmak üzere üç bölgeye ayrılmıştır. Sonrasında her bölge ayrı ayrı sinir ağına beslenmiş ve üç ayrı gölge için farklı çıktı alınmıştır. En son adımda Fuzzy Sugeno integrali kullanılarak üç ayrı çıktı birleştirilmiş ve yüz tanıma gerçekleştirilmiştir. Yüz tanıma sistemi 20 kişilik küçük veri setinde test edilmiş ve yüzü üç bölgeye ayırmanın tek bölgeye ayırmaktan daha iyi sonuçlar verdiği kanıtlanmıştır (Melin, Felix ve Castillo, 2005).

2.4. Veri Çoğaltma Teknikleri

Yukarıda ele alınan çalışmalarda gördüğümüz üzere yüz tanıma sistemleri insan seviyesine yakın hatta bazı sistemler insan seviyesinden üstün performans sergilemektedirler. Ancak sağlam yüz tanıma sistemlerini oluşturmakta hala büyük zorluklar vardır. Bu sistemlerin eğitimi, büyük miktarda veri ve yüksek hesaplama gücüne sahip bilgisayar donanımları gerektirir. Günümüzde bu denli büyük veri setleri açık kaynak olarak bulmak imkânsızdır. Dolayısıyla bu durum, araştırmacıları veri çoğaltma tekniklerine doğru yöneltmiştir. Aşağıda veri çoğaltma ile ilgili yapılan çalışmaları inceleyeceğiz.

Derin öğrenmeye dayalı yüz tanıma sistemlerinin doğruluk performansını artırmak amacıyla, (Lv vd., 2017) bilimsel çalışmada veri setine ait yüz görüntülerine değişik saç modelleri, gözlük şablonları, poz ve aydınlatma uygulanmıştır. Yüz hizalama işlemi gerçekleştirildikten sonra, 87 farklı saç modeli oluşturup bütün modeller her bir

kişiyeye uygulanmıştır. İkinci aşamada 100 farklı gözlük şablonu oluşturuldu ve aynı gözlük şablonları yüz görüntülerine uygulandı. Poz işleminde ise (Jiang vd., 2005) çalışmasında önerilen 3B yeniden yapılanma yöntemini kullanarak farklı pozlarda görüntüler elde ettiler. Son olarak bütün yüzlere ortam ışığı ve hareketli yer ışığı uyguladılar. Deneysel sonuçlar önerilen yöntemlerin yüz tanıma performansını büyük ölçüde iyileştirdiğini göstermektedir. Diğer (Seyyedsalehi Z. ve Seyyedsalehi A., 2014) çalışmada, araştırmacılar doğrusal olmayan manifold ayırıcı (nonlinear manifold separator) sinir ağı kullanarak farklı ifadelerden oluşan görsel yüz görüntüleri üretme yöntemini önermişlerdir. Bu yöntemin, kısıtlı ortamlarda oldukça başarılı olduğu mevcut çalışmada kanıtlanmıştır. Farklı bir bilimsel çalışmada, her bir kişiyeye özel yeni ifade görüntülerini sentezlemek amacıyla ifade alt uzay projeksiyon (expression subspace projection) yöntemi önerilmiştir (Mohammadzade ve Hatzinakos, 2013). Aynı zamanda bu yöntemin sonuçlara olumlu yansıdığı bildirilmiştir. Eğitim veri setini genişletmek için, (Yi vd., 2014) ve (Zeiler ve Fergus, 2014) çalışmalarından esinlenerek (Howard, 2013) çalışmasında ters çevirme (flipping) ve kırpma (cropping) yöntemlerini benimsemiştir. Günümüzde yüz veri çoğaltma teknikleri ile ilgili en fazla ilgi çeken çalışmaların başında (Karras vd., 2017) çalışması gelmektedir. Dünya çapında meşhur ekran kartı üreticisi Nvidia bünyesinden bir grup araştırmacı (Goodfellow vd., 2014) çalışmasından esinlenerek, dolayısıyla Çekişmeli Üretici Ağlar (Generative Adversarial Networks) kullanarak 1024x1024 boyutunda sentetik yüzler üretmişlerdir. Elde edilen sonuçlar oldukça etkileyiciydi, sentetik yüzler insan gözüyle sentetik oldukları fark edilemeyecek kadar yüksek kalitedeydiler. (Zhimi, Chen ve Tong, 2017) çalışmasında, araştırmacılar Wasserstein ÇÜA (Wasserstein GANs) kullanarak 64x64 boyutunda sentetik yüzler üretmişlerdir. Wasserstein ÇÜA'nı eğitmek için CelebA (Liu vd., 2015) veri seti kullanılmış ve elde edilen sonuçlar memnun edici olmuştur.

BÖLÜM 3

DERİN ÖĞRENME’NİN TEORİK ARKA PLANI

Bu bölümde Derin Öğrenme kavramının açıklaması yapılacaktır. Derin Öğrenmenin açıklamasını yapmadan önce, Makine Öğrenmesi ve Yapay Sinir Ağlarının (YSA) ne olduğunu açıklamamız gerekmektedir.

Makine Öğrenmesi, bilgisayarların geçmiş bilgilerden elde edilen deneyimlere göre hareket ederek, gelecekteki olayları tahmin etmesine ve modelleme yapmasına imkân tanıyan bir Yapay Sinir Ağı dalıdır. Makine öğrenmesi algoritmalarını verinin yapısına göre iki gruba ayrılmaktadır: Denetimli Öğrenme ve Denetimsiz Öğrenme.

3.1. Denetimli Öğrenme

Denetimli Öğrenme algoritmaları en yaygın şekilde kullanılan algoritmalarlardır. Bu gruptaki algoritmalar, geçmişten öğrendikleriyle yola çıkarak gelecekteki olayları tahmin etmek için etiketli (labeled) verileri kullanırlar. Aslında, eğitim sırasında kullanılacak veri ve veriye ait sınıflar önceden bilinmektedir. Bu bilgiler doğrultusunda sistem öğrenir (eğitilir) ve yeni verilecek olan veriyi bu öğrendiklerine göre değerlendirir. Sistemin iyi bir şekilde eğitilmesi, gelecekte yapacağı tahminlerin daha tutarlı olması anlamına gelmektedir. Denetimli öğrenme algoritmaları, sınıflandırma ve regresyon dâhil olmak üzere birçok makine öğrenme problemini çözmek için iyi bir yöntemdir. Günümüzde makine öğreniminde çözülen problemlerin çoğu denetimli öğrenme ile eğitilmektedir.

3.2. Denetimsiz Öğrenme

Denetimsiz öğrenme algoritmalarında, yalnızca girdi verisi mevcuttur. Dolayısıyla girdi verisine karşılık gelecek bir çıktı değişkeni yoktur. Mevcut algoritmalar verilerdeki temel yapıyı modelleyerek veriler hakkında daha fazla bilgi edinmeyi amaçlarlar. Öğrenme algoritmasına etiketler verilmez ve verilerdeki yapıyı bulmak için algoritmadan kendi kendine öğrenmesi beklenmektedir. Bazı örüntü tanıma problemlerinde, eğitim verileri, karşılığı amaç değerleri olmayan bir dizi giriş vektöründen x oluşur. Bu tür denetimsiz öğrenme problemlerinde amaç, kümeleme (clustering) olarak adlandırılan veri içindeki benzer örnek gruplarını bulmak veya yoğunluk tahmini (density estimation) olarak bilinen, uzayda

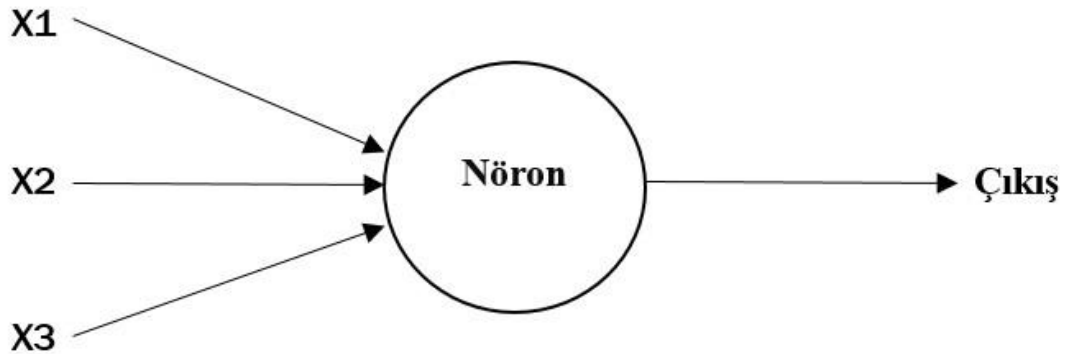
verilerin nasıl dağıldığını tespit etmektir. Özellikle kümeleme işleminde birçok denetimsiz öğrenme algoritması başarılıdır.

3.2 Yapay Sinir Ağları (Artificial Neural Networks)

Yapay zekâ tekniklerinden biri olan YSA, biyolojik sinir ağlarının başka bir deyişle insan beyninin işleyişini taklit ederek var olan verileri analiz edip, çeşitli algoritmalar ile bu verilerden yeni bilgiler oluşturan sistemlerdir. YSA'nın temel işlevi bir girdi verisi almak ve bir dizi karmaşık hesaplama yaptıktan sonra sonuçları ağın çıkış bölümüne aktarmaktır. Bu sistemler, başta iş, sanayi, bilim vs. olmak üzere birçok alanda çok çeşitli görevler için kullanılırlar. YSA tarihçesi eskilere dayanmaktadır. Ancak literatürde XOR problemi olarak bilinen problemdeki elde edilen başarısızlık sonucu belli bir müddet bu ağlara ilgi azalmıştır. Bir müddet sonra (Hopfield ve Tank, 1985) çalışmasında bilgisayar programlama ile çözülmesi zor olan problemleri yapay sinir ağlarının yeteneği ile çözüme ulaştırmanın mümkün olduğu kanıtlanmıştır. Bundan birkaç yıl sonra, 1988 yılında geri beslemeli algoritmalar da geliştirilmiş ve XOR gibi problemlerin çözümü YSA ile mümkün olacağı bildirilmiştir. Bundan sonraki dönemlerde mevcut ağlara ilgi artmıştır.

3.2.1. Yapay Sinir Ağları'nın Yapısı

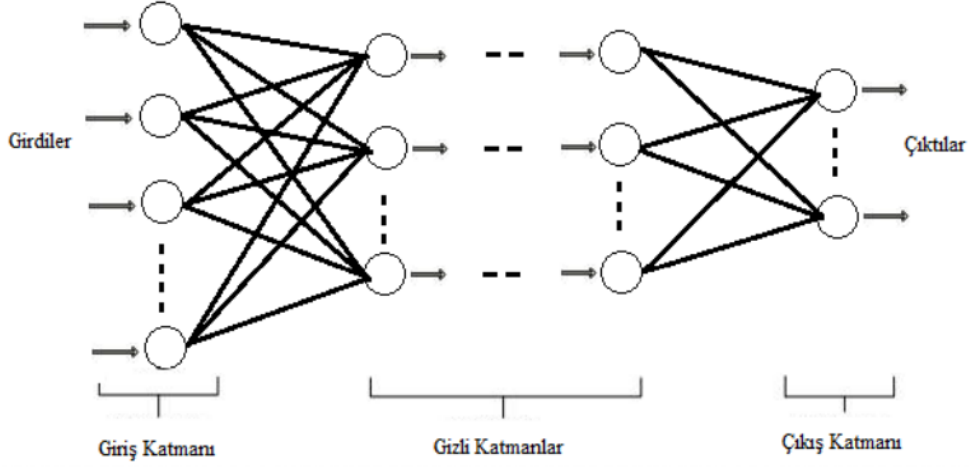
YSA gelişimi (Rosenblatt, 1958) Tek katmanlı algılayıcı (Perceptron) kavramını ortaya atmasıyla başlamıştır. Tek katmanlı algılayıcı en basit ağ türlerinden biridir. Şekil 3.1. Tek katmanlı algılayıcının basit yapısını gösterir. Çıktı sonucunu 0 veya 1 olarak elde edebilmek için birkaç girdi almaktadır ve bu girdi değerlerinin ağırlıklarının toplamını hesaplayarak çalışmaktadır.



Şekil 3. 1. Perseptronun yapısı.

Tek katmanlı algılayıcının bu basit yapısı, daha sonra geliştirilen mimariler için bir başlangıç noktası olarak görülmüştür. YSA'nın ilk ve temel modelleri olan tek katmanlı algılayıcılar doğrusal olan problemleri çözebilme konularında oldukça yetenekli olduklarını gözlemlemek mümkündür. Ancak doğrusal olmayan problemlerde yetersiz kalmaktadırlar. Bu sorunun çözülmesi amacıyla çok katmanlı algılayıcılar geliştirilmiştir. Çok katmanlı algılayıcılar giriş, ara ve çıkış olmak üzere üç temel bölümden oluşur. İlk bölüm giriş katmanı olarak bilinmektedir. Dış dünyadan girdilerin geldiği katmandır ve dış dünyadan girdinin geldiği sayıda nöron bulunmaktadır. Giriş katmanında girdiler herhangi bir işleme tabi tutulmadan bir sonraki katmana iletilmektedirler. Ara bölüm ise gizli katman veya katmanlar olarak adlandırılırlar. Giriş katmanından çıkan bilgiler bu katmana iletilmektedir. Her katman birkaç nörondan oluşur ve bir katmandaki her bir nöron önceki katmandaki tüm nöronlarla bağlıdır. Son bölüm başka farklı bir ifadeyle çıkış katmanı, ara katmandan gelen bilgilerin işlendiği ve giriş katmanından gelen verilere karşılık olarak çıktılarının üretildiği katmandır. Yapılan işleme bağlı olarak çıkış katmanında hücre sayısı birden fazla olabilmektedir. Bu katmanda bulunan her hücre bir evvelki katmana ait tüm hücrelere bağlıdır. Şekil 3.2'de çok katmanlı bir YSA'nın temel yapısı gösterilmektedir. Ağıdaki hücreler birbirlerine ağırlıklar ile bağlanır. Ağırlıklar, hücrelerin birbirleriyle olan bağlantılarının sayısal değerini ifade eder ve hücre üzerindeki etkisini gösterir. Bazı durumlarda bias denilen ek bir dal nöronlara bağlanır. Bias, +1 veya -1 değerine sahiptir ve önceki katmanın hücrelerine bağlı değildir. Bias'ın asıl amacı sinir ağlarının problem çözme performansını arttırmaktır.

YSA yapılarına göre ileri beslemeli ve geri beslemeli olmak üzere ikiye ayrılırlar. Bu yapıların detaylı açıklaması bir sonraki bölümde yapılacaktır. Tekrarlayan Sinir Ağlar (Recurrent Neural Networks), Evrişimsel Sinir Ağlar (Convolutional Neural Networks) vs. gibi YSA'nda katman sayısını artırarak oluşturulan farklı DÖ mimarileri de mevcuttur.

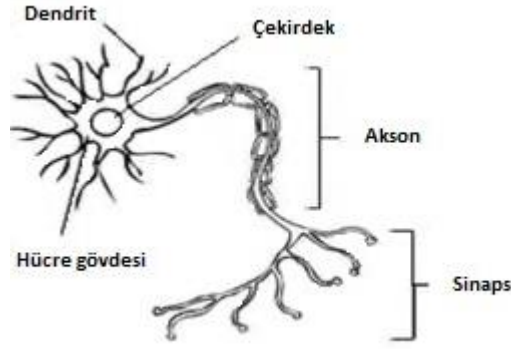


Şekil 3. 2. Çok katmanlı bir YSA'nın temel yapısı.

3.2.2. Yapay Sinir Ağları'nın Çalışma Prensibi

Beynimizin ana işlem ünitesi sinir hücreleri ya da başka bir deyişle nöronlardır. İnsan sinir sisteminde mevcut nöron sayısının yaklaşık olarak 100 milyar olduğu öngörülmektedir. Yapılan araştırmalara göre normal bir sinir hücresinin 50.000-250.000 arasında farklı nöronlarla bağlantılı olduğu tahmin edilmektedir. Beynimizdeki hücreler ve bunlar arasındaki bağlantıları göz önünde bulundurduğumuzda yaklaşık olarak 2.5-3 milyar Gigabyte'e karşılık gelen hafızaya sahibiz demektir. Biyolojik sistemlerde, nöronlar arasında sinaptik bağlantılar ayarlanarak öğrenme gerçekleşmektedir. Dolayısıyla, insanların dünyaya gelme anından itibaren bir yaşayarak öğrenme süreci içine girmiş bulunmaktadır ve aynı süreç içerisinde elde edilen tecrübelerden sinaptik bağlantılar ayarlanır. Yeni bağlantılar ortaya çıkar ve öğrenme gerçekleşir. Biyolojik sinir hücresi çekirdek (nucleus), akson (axon), dendrit (dendrite) ve sinaps (synapse) gibi dört ana parçadan oluşmaktadır. Şekil 3.3'de biyolojik sinir hücresinin ana yapısı gösterilmiştir.

Gelen sinyalleri çekirdeğe iletme görevini dendritler üstlenmektedir. Çekirdek ise bütün mevcut sinyalleri toplayarak aksone aktarır. Akson mevcut sinyalleri işleyerek sinaplara gönderir. Sinapslar ise yeni üretilen sinyalleri diğer sinir hücrelerine iletirler.



Şekil 3. 3. Biyolojik sinir hücresi parçaları.

Yapay Sinir hücrelerinin yapısı, biyolojik sinir hücrelerin yapısına benzemektedir. Yapay sinir hücreleri veya başka bir deyişle yapay nöronlar aralarında bağlantı oluşturarak YSA'nı oluştururlar. Yapay sinir hücresi beş ana bölümden oluşmaktadır:

- Girdi: Kullanıcı tarafından veya başka bir hücreden YSA gelen bilgilerdir.
- Ağırlıklar: Hücreye gelen bilginin değerini ve hücre üzerindeki etkinlik seviyesini belirleyen katsayılardır. Ağırlıkların değerleri sıfır, pozitif veya negatif olabilir. İlgili girdinin ağırlık değerinin yüksek olması yapay sinir hücresine güçlü bağlanması anlamına gelmektedir. İlgili girdinin ağırlık değerinin küçük olması ise daha zayıf bağlanması anlamına gelir. Ağırlık değeri sıfır olduğunda hücrenin herhangi bir etkinliği olmamaktadır.
- Toplama Fonksiyonu: Hücreye gelen girdiyi ilgili ağırlıklarla çarpıp toplar. Böylece hücrenin net girdisi hesaplanır. Bçok farklı toplama fonksiyonu mevcuttur. Dolayısıyla probleme bağlı olarak farklı toplama aktivasyonları kullanılmaktadır. Ancak genelde deneme-yanılma yoluyla en ideal fonksiyon belirlenmektedir.
- Aktivasyon Fonksiyonu: Net girdiyi matematiksel işleme tabi tutarak hücrenin çıktısının belirlenmesini sağlamaktadır. Bu fonksiyonlar genelde doğrusal olmayan fonksiyonlardır. İlgili fonksiyonları seçerken, araştırmacılar türevlerinin basit hesaplanabilir olmasını dikkat ederler
- Çıktılar: Aktivasyon fonksiyonları tarafından işleme tutulduktan sonra elde edilen çıktı değerleridir. Probleme bağlı olarak elde edilen çıktı kullanıcılar, farklı katmana ait hücrelere veya kendisine girdi olarak gönderilebilir

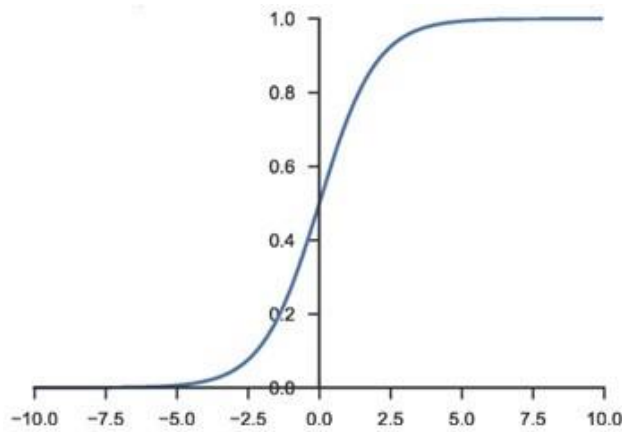
Yapay Sinir Ağlarının çalışma şekli şu şekilde gerçekleşmektedir: İlk olarak giriş katmanında giriş vektörleri olarak da bilinen girdiler alınır, daha sonra gizli

katmandaki her nöron giriş katmanındaki tüm değerleri ağırlıklar vasıtasıyla alır ve bunun yanı sıra bias (eşik veya sapma) nöron olarak seçilen fonksiyona bağlı olarak hesaplamayı yapar ve çıktı sonuçları üretir. Bir sonraki katman, bir önceki katmanın çıktı sonuçlarını girdi değeri olarak alır ve bir kez daha aktivasyon fonksiyonuna göre hesaplamayı yapar ve çıktı sonuçları üretir. Gizli katmanların sayısına bağlı olarak çıkış katmanına kadar bu durum devam eder ve çıkış katmanından sonuç dış dünyaya çıkar. Bu, bilginin her zaman ileri beslendiği ve asla geri beslenmediği anlamına gelir. Dolayısıyla bu tarz mimarilere İleri beslemeli sinir ağları (Feedforward neural networks) denmektedir. Bir sonraki alt başlıkta bilginin çıkış katmanından geriye doğru beslendiği, YSA'nın eğitiminde oldukça sık kullanılan Geri yayılım (Backpropagation) yöntemini ele alacağız. YSA'nın iyi performans sergilemelerinde aktivasyon fonksiyonları kilit rol oynarlar. En yaygın olarak kullanılan aktivasyon fonksiyonları aşağıda ele alınacaktır. Daha basit sınıflandırma problemlerinde Sigmoid veya diğer adıyla Lojistik, doğrulamanın önemli olduğu problemlerde ise ReLu veya Sızıntı (leaky) ReLU aktivasyon fonksiyonu kullanılmaktadır.

1.Sigmoid Aktivasyon Fonksiyonu

Matematiksel denklemler ve sınıflandırma problemleri için kullanılan bu fonksiyon her bir girdi değeri için 0 ila 1 arasında değer üretmektedir. Denklem 3.1 ve Şekil 3.4 Sigmoid fonksiyonunu göstermektedir:

$$F(Net) = \frac{1}{1+e^{-Net}} \quad (3.1)$$

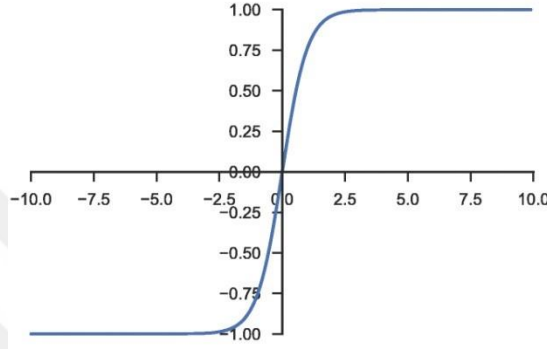


Şekil 3. 4. Sigmoid fonksiyonunun grafiksel karşılığı.

2. Tanjant aktivasyon fonksiyonu

İlgili fonksiyon, eşik değeri (threshold value) kullanarak giriş değerlerini -1 ve +1 arasında ölçeklendirir. Aşağıdaki 3.2 denklemini ve Şekil 3.5 Tanjant fonksiyonunu göstermektedir.

$$F(x) = \frac{e^x}{1+e^x} \quad (3.2)$$

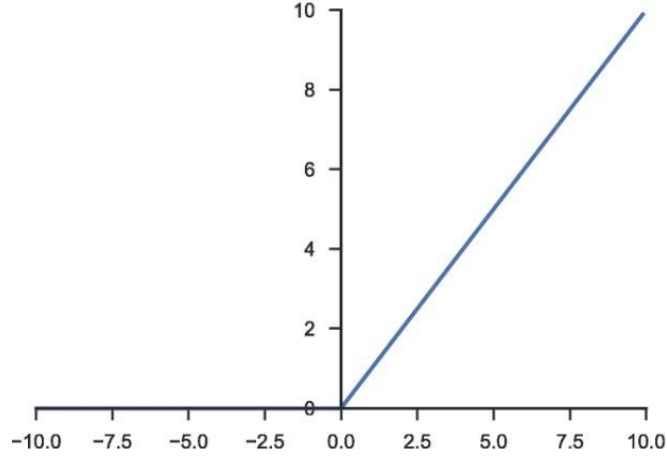


Şekil 3. 5. Tanjant fonksiyonunun grafiksel karşılığı.

3. ReLU (Rectified Linear Unit) aktivasyon fonksiyonu

Sadece pozitif değerlerin geçmesine izin veren ve tüm negatif değerleri sıfırlayan güçlü bir fonksiyondur. Derin Öğrenmede en yaygın kullanılan aktivasyon fonksiyondur ve yalnızca gizli katmanlarda uygulanabilmektedir (Krizhevsky, Sutskever ve Hinton, 2012). Denklem 3.3 ve Şekil 3.6, ReLU fonksiyonunu temsil etmektedir. ReLU aktivasyon fonksiyonunun sık karşılaştığı problem negatif girdiler için gradyan'ın sıfır olmasıdır, bu da ağırlıkların güncellenmeyeceği anlamına gelmektedir. Dolayısıyla bu durum hiçbir değişikliğe duyarlı olmayacak ve bir daha aktif hale gelmeyecek hücreler meydana getirecektir. Böylece ağın bir bölümü pasif hale gelecektir. Bu sorunu çözmek için Sızıntılı (leaky) ReLU önerilmiştir. Sızıntılı ReLU negatif değerlerde 0 yerine çok düşük değerler almaktadır.

$$F(x) = \max(0, x) \quad (3.3)$$



Şekil 3. 6. ReLU fonksiyonunun grafiksel karşılığı.

Bir yapay sinir ağında hangi aktivasyon fonksiyonunun kullanılacağını seçmek her zaman basit-kolay bir iş değildir. Bu durum, çözülecek probleme bağlıdır, bazı durumlarda bir problemi çözmek için 2 veya 3 farklı aktivasyon fonksiyonunu birleştirmeye ihtiyaç duyulur.

YSA'nın gizli katmanlarında neler olup bittiğini, yani işlemlerin nasıl gerçekleştiğini tam olarak detaylı bir şekilde anlaşılmadığı için ``Kara kutu`` yakıştırılması yapılmaktadır. Bireysel olarak her bir aktivasyon fonksiyonunun nasıl çalıştığı bilinmektedir, ancak binlerce veya milyonlarca nöron istenilen çıktıyı elde etmek için birleştirildiğinde nasıl çalıştıklarını anlamak oldukça zordur.

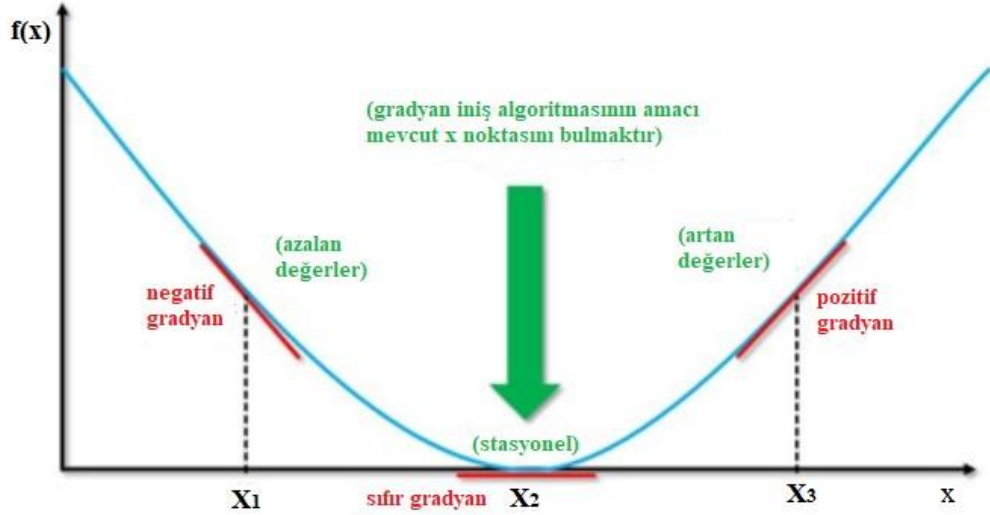
Genel olarak, bir problemi çözmek için sinir ağları tasarlarken, kullanılacak katman sayısı, kullanılacak nöron sayısı ve kullanılacak aktivasyon fonksiyonlarının türü veya türleri, alınacak en önemli kararlardır. Her problem farklıdır, dolayısıyla tüm problemlere veya iki benzer probleme kesin uygundur denilecek bir ağ mimarisi yoktur. Yapay sinir ağı mimarisi belirlendikten sonra, ağı eğitmek için iyi bir strateji gerekmektedir. Bir sonraki alt başlıkta YSA'nın nasıl eğitileceğine dair bilgi verilecektir.

3.2.3. Yapay Sinir Ağları'nın Eğitilmesi

Düzenli çalışabilecek bir YSA'nı eğitmek için elinizde büyük miktarda veri olması gerekmektedir. Yeterli veri olmadan, ilgili problemleri çözebilecek bir yapay sinir ağını eğitmek mümkün değildir. Bu durum, yukarıda açıklandığı gibi açık bir problem olarak görülmüş ve veri artırma teknikleri ile çözüme kavuşturulmuştur.

Öğrenme süreci YSA'da dış ortamdan giriş katmanına girdiler alınarak başlar ve ağırlıklar rastgele belirlenir. Alınan girdiler aktivasyon fonksiyonunda matematiksel işleme tabi tutulduktan sonra çıkış katmanında bir veya birden fazla çıktı elde edilir. Bir sonraki adımda, sistemden elde edilen sonuç ile beklenen sonuç karşılaştırılır ve kayıp fonksiyonu (loss function) veya diğer adıyla maliyet fonksiyonu (cost function) kullanılarak hata oranı bulunur. Artık hata oranı belirlendikten sonra, bir optimizasyon problemi olarak görünür ve çeşitli optimizasyon algoritmaları kullanılarak hata oranını geri beslemeli yöntem ile en düşük seviyeye indirmek amaçlanmaktadır. YSA'nın eğitiminde en yaygın kullanılan algoritma geri beslemeli algoritmasıdır. Geri beslemeli sinir ağlarında, bir hücrenin çıktısı bir önceki veya aynı katmanda bulunan herhangi bir hücreye girdi olarak beslenebilmektedir. Geri beslemeli yöntem üç ana işlevden oluşur; sonucun hesaplanması açısından ileri besleme, hata oranını geriye doğru işletmek (backward) için gradyan alma ve herhangi bir optimizasyon algoritması ile ağırlıkların güncellenmesi işlemi. Son zamanlarda meşhur ve oldukça sık kullanılan optimizasyon algoritmaları olan: Gradyan iniş (Gradient Descent), Momentum ile gradyan iniş (Gradient descent with momentum), Ölçeklenmiş Konjüge Gradyan (Conjugate gradient), Broyden Fletcher Goldfarb Shanno Quasi-Newton, Levenberg-Marquardt ve Kare Ortalamalarının Karekökü (Route means squareprop) hakkında bilgi yazının devamında verilmektedir.

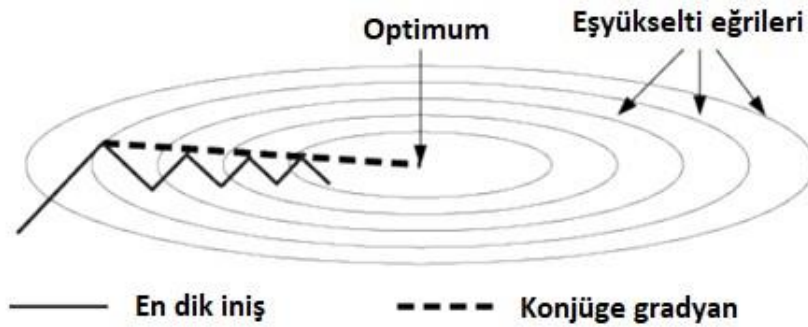
Gradyan İniş algoritması: Optimizasyon algoritmaları arasında en popüler ve en yaygın kullanılan algoritmaların başında Gradyan İniş algoritması gelmektedir. Literatürde Gradyan İniş algoritması aynı zamanda En dik iniş (Steepest descent) olarak da bilinmektedir. Gradyan iniş algoritması, bir fonksiyonun minimum değerini bulmak için kullanılan yinelemeli bir optimizasyon algoritmasıdır. Algoritmanın ana amacı, parametreleri rasgele değerlere sıfırlamak ve ardından her bir yinelemede gradyan yönünde küçük adımlar atmaktır. Bu algoritma, derin öğrenme modellerinde sinir ağının ağırlıklarını geri yayılım yoluyla güncellemek için kullanılır. Şekil 3.7 'de gösterildiği gibi geri yayılım algoritması ağırlıkları gradyan'ın negatifine doğru, yani performans fonksiyonu'nun en hızlı azaldığı yöne ayarlar ve hata gradyanını hesaplayarak en optimum noktayı bulmaya çalışır.



Şekil 3. 7. Gradient Descent algoritmasının örneği.
(Hochreiter, Younger ve Conwell 2001).

Momentum ile Gradyan İniş algoritması: Mevcut gradyanların eksponansiyel ağırlıklarının ortalamasını hesaplar ve sonrasında ağırlıkları güncellemek için aynı gradyanı kullanır. Yüzeydeki küçük özellikleri göz ardı ederek gradyan vektörlerinin hızlanmasına yardımcı olur (Sharma ve Venugopalan, 2014) . Ayrıca güncellemeyi kolaylaştırmak için önceki gradyanları dikkate alır.

Konjüge Gradyanlı İniş algoritması: Mevcut tabanlı algoritmalar, çeşitli problemlerde doğrusal yakınsama (linear convergence) göstermektedirler. Yukarıda bahsedilen iki diğer algoritmaya kıyasen, adım boyutu ölçeklendirme (step size scaling) mekanizması sayesinde daha hızlıdır. En dik iniş (steepest descent) ve Konjüge Gradyanlı yöntemlerinin arama yolları Şekil 3.8'de gösterilmiştir.

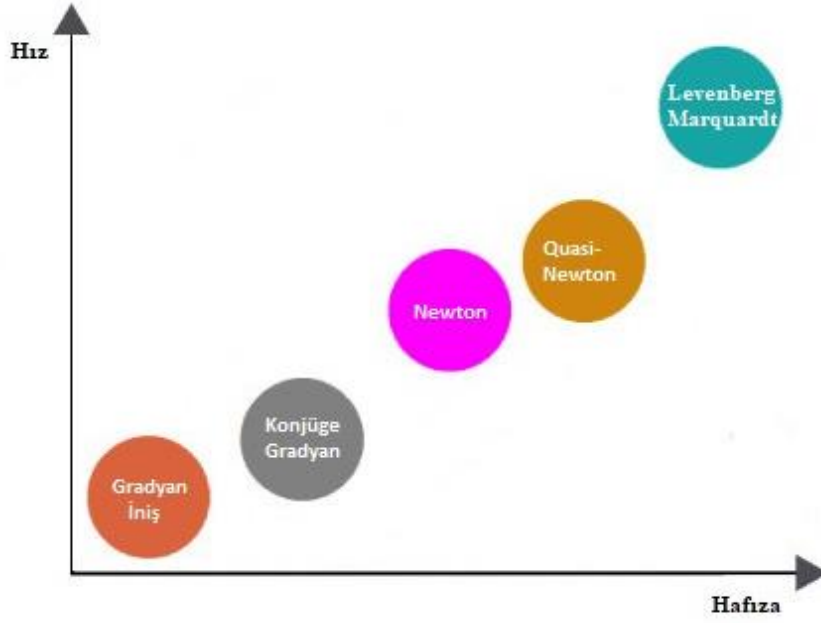


Şekil 3. 8. En dik iniş ve konjüge gradyan yöntemlerinin arama yolları (Chan ve Ewins, 2011).

BFGS, Quasi-Newton algoritması: Ağ eğitiminde yaygın olarak kullanılan bir başka algoritma da BFGS quasi-Newton algoritmasıdır. BFGS quasi algoritması, bir fonksiyonun sabit bir noktasını arayan, Newton'un tepe tırmanma tekniklerine benzemesiyle göze çarpmaktadır. Bu tekniğin derin olmayan ağlarda bile iyi performans sergilediği uzun zamandır bilinmektedir (Curtis ve Que, 2015).

Levenberg-Marquardt algoritması: İlgili algoritma özellikle, kare hata tipi kayıp fonksiyonlarının toplamı üzerinde çalışır. Tam Hessian matrisini hesaplamak yerine, gradyan vektörünü ve Jacobian matrisini hesaplar. Jacobian hesaplamasına bağımlılığı söz konusu olması ve büyük Jacobian matrisinin çok fazla hafıza gerektirmesi, ağlarda ve büyük veri kümelerinde iyi performans sergilemesini zorlaştırmaktadır.

Kare ortalamanın karekökü algoritması: Bu algoritma ilk olarak, Toronto Üniversitesi üyesi Geoffrey Hinton tarafından sinir ağları üzerine bir çevrimiçi eğitim platformu olan Coursera ders slaytlarında ortaya atılmıştır. Hinton, mevcut algoritmayı resmi bir akademik makalede yayınlamamıştır, ancak günümüzde hala derin öğrenme için en popüler gradyan iniş optimizasyon algoritmalarından biridir. Kare ortalamanın karekökü algoritması, kare gradyan'ın hareketli ortalamasını kullanarak Adagrad algoritmasında hızlı bir şekilde azalan öğrenme oranlarını çözmeyi amaçlar. Gradyanı normalleştirmek için son gradyan inişlerinin büyüklüğünden yararlanmaktadır. Mevcut algoritmada öğrenme hızı otomatik olarak ayarlanır ve her parametre için farklı bir öğrenme hızı seçilmektedir. Yukarıda bahsedilen bir kısım optimizasyon algoritmalarının performans karşılaştırması Şekil 3.9'da görülebilir.



Şekil 3. 9. Optimizasyon algoritmalarının hız ve hafıza açısından karşılaştırılması (Quesada, 2019).

Önceden de bahsettiğimiz gibi geri beslemeli ağların ana amacı, eğitim sırasında ağırlıkları güncelleyerek hata oranını en aza düşürmektir. Hata oranını en az seviyeye düşürme sürecinde, yapay sinir ağlarının ağırlıkları yenilenir ve bununla aktivasyon fonksiyonunu daha efektif bir şekilde kullanılması hedeflenir. Bu süreç probleme bağlı olarak yüzlerce, binlerce hatta milyonlarca kez tekrar edilir. Her bir tekrarlanma işlemine eğitim tur sayısı (epoch) denilmektedir. Ağırlıkların devamlı yenilenip, beklenen sonuç elde edilinceye kadar ki geçen zaman dilimine ise öğrenme denir. Tüm ağırlıklara sıfıra yakın bir hata oranı vermek üzere ince ayar (fine-tuning) yapıldıktan sonra, yapay sinir ağı eğitilmiş olarak kabul edilir ve ilgili problemi yüksek doğrulukla çözmesi beklenir. Yapay sinir ağı eğitildikten sonra, ağı daha önce hiç rastlamadığı girdi verileri ağına verilir ve ürettiği sonuç ile istenilen sonuç arasındaki bağlantı incelenir. Eğer yeni verilen girdi verilerinden istenilen-beklenen sonuç üretilmişse, yapay sinir ağı doğru eğitilmiştir demektir, aksi sonuçlar üretildiğinde ise hiper-parametrelerde düzeltmeler yapılması gerekmektedir. Hiper-parametrelerinin doğru seçilmesi geri beslemeli ağlarda eğitim başarısını etkilemektedir. Dolayısıyla iyi performans elde etmek için hiper-parametrelerin titizlikle seçilmesi gerekmektedir. Yazının devamında derin öğrenme uygulamalarında en sık kullanılan hiper-parametreler hakkında bilgi verilecektir.

Veri seti boyutu (Data set size): Önceden de bahsettiğimiz gibi YSA başarılı bir şekilde eğitmek için veri setinin büyük olması en önemli faktörlerin başında gelmektedir. Veri setimizin büyüklüğü ne derecede fazla olursa, YSA mimarisinin öğrenmesi de o derece iyi oranda olacaktır. Ancak veri setinin büyüklüğü iyi performans sergileyecek bir model için yeterli olmayabilir, aynı zamanda veri setinin çeşitliliği fazla olması da önem taşımaktadır. Çünkü veri seti artıka başarı oranı bir noktaya kadar artmaktadır, veri setinin çeşitliliği az ise belli noktadan sonra ağ çeşitlilik göremediği için başarı oranında artış söz konusu olmaz.

Batch boyutu (Batch size): Veri setimizde bulunan bütün verilerin aynı anda işlenmesi zaman ve bellek açısından oldukça maliyetli bir iştir. Çünkü yukarıda bahsettiğimiz gibi geri yayılım işlemi ile gradyan hesaplanması yapılır. Dolayısıyla gradyan hesaplama işleminde veri sayısı ne kadar çok olursa hesaplama işlemi de okadar uzun sürmektedir. Bu problemi çözüme kavuşturmak için veri setindeki verilerin parça parça işlenmesi önerilmiştir. Bu sebeple öğrenme işlemi bu parçalar üzerinden yapılmaktadır. Veriseti'nin parçalar halinde işlenmesi literatür'de batch boyutu olarak adlandırılır. Modeli oluştururken batch boyutu olarak belirlenen değer, modelin aynı anda o değer kadar veri işleyeceği anlamını taşımaktadır. Belirlenen batch boyutu değeri kullanılan ekran kartı hafızasına sığacak orantıda belirlenmesi gerekmektedir. Batch boyutunun değeri 2,4,8,16,32,64,128...,512 vs. gibi değerler ile belirlenmediğinde performans'da ani düşüşler yaşanabilmektedir. Aynı zamanda batch boyutu seçiminde verilerin rastgele seçilmesi eğitim aşamasında meydana gelebilecek olası ezberlemelere (overfitting) karşı modeli daha dayanıklı yapacaktır.

Öğrenme hızı (Learning rate): Öğrenme hızı, modelin mevcut probleme ne kadar hızlı adapte olduğunu kontrol eder. Öğrenme hızının değeri büyük seçilmesi durumunda salınım (oscillation) sebep olacaktır. Küçük olması durumunda ise adım adım ilerleyeceğinden öğrenme süresi daha uzun sürecektir. Öğrenme hızı değerinin büyük olması, ilk başlarda iyi sonuçlara yol açacaktır ancak daha sonrasında sonuçlarda düşüş meydana gelmektedir. Daha düşük öğrenme hızı oranı ile öğrenme daha uzun sürecektir fakat sonuç daha şeffaf olacaktır. Dolayısıyla, öğrenme aşamasında, öğrenme hızı oranı ilk başlarda büyük seçilip iterasyonlar artıka düşürüldüğü takdire istenilen sonuçlar alınacaktır (Kriesel, 2009).

Eğitim turu sayısı (Epoch): Geri yayılım yönteminde veri setinin hesaplama işlemine tabi tutulmasında kullanılan iki öğrenme yöntemi mevcuttur. Bu yöntemler literatürde

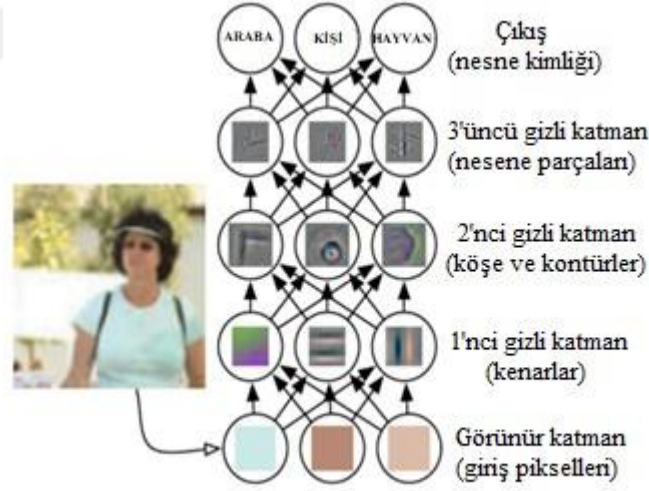
tekil (online) eğitim ve batch eğitimi olarak adlandırılırlar. Tekil eğitimde, ağırlıkların güncelleşmesi veri setindeki her bir örnek verinin ağı yüklenmesiyle meydana gelen hata oranının geri yayılımı ile gerçekleşmektedir. Batch eğitiminde yukarıda da bahsettiğimiz gibi ağı veri setinin bir kısmı parça olarak yüklendikten sonra elde edilen ortalama hatanın geri yayılımı ile ağırlıklar güncellenir (Haykin, 2009). Daha sonra tekrar bir parça ile model eğitilir ve ağırlıklar yine güncellenir. Bu işlem model için optimum ağırlık değerleri bulununcaya kadar tekrarlanmaktadır. Bir modelin eğitilmesi normalde uzun sürmektedir, probleme bağlı olarak eğitim günlerce, haftalarca hatta aylarca sürebilmektedir. Eğitim tur sayısının büyüklüğü probleme göre değişmektedir, ancak örüntü öğrenilen modellerde eğitim tur sayısı diğer modellere göre daha yüksek tutulmalıdır. Eğitim turu sayısı arttıkça modelin başarı oranı da artacaktır, başarı oranı belli bir eğitim turundan sonra çok düşük olacağı için eğitim süreci sonlandırılabilir.

Momentum katsayısı: Genel itibari ile ağı hızlı bir şekilde toparlanmasını sağlamaktadır. Momentum katsayısı hata oranının düşmesinde önemli rol oynamaktadır. Hem bu nedenle hem de öğrenme sürecinde ağı salınımını engellediği için momentum katsayısı kullanmak önemlidir. Ancak momentum katsayısı kullanmanın her yerde deva olmadığını altını çizelim. Momentum katsayısı değeri 0 ile 1 arasında olmalıdır (Sağıroğlu, Çolak ve Bayındır, 2006; Hines, Tsoukalas ve Uhrig, 1997).

3.3 Derin Öğrenme

Derin öğrenme, Yapay sinir ağlarına dayanan bir Makine öğrenmesi dalıdır. Tez konusuyla bağlantılı olması açısından, derin öğrenme açıklamasını Bilgisayar görüşü (Computer Vision) üzerinden yaptık. Yukarıda açıklandığı gibi, yüz tanıma probleminin başarıyla sonuçlanması için, görüntü üzerinde mevcut olan yüz veya yüzlerden özelliklerin (features) çıkarılması gerekmektedir. Aynı durum, görüntü üzerinden nesne tanıma için de geçerlidir. Yüzün kendine has çok fazla özelliği vardır, ancak görüntü veya video üzerinden yüze veya nesneye ait birbirinden farklı özellikleri manuel çıkarmak oldukça zor ve zaman gerektiren bir iştir. Dolayısıyla bu zorluğu aşmak için birçok algoritma geliştirilmiştir. İnsanoğlunun araba, ev, otobüs, köpek ve buna benzer nesnelere tanıması ve tespit etmesi üst düzey özelliklerin sayesinde kolaydır. Örneğin arabanın 4 tekerleği olması, metalden oluşması, ön cama sahip olması vs. gibi. Bir görüntü üzerinden bilgisayarın bu üst düzey özellikleri tanıması

hiç de kolay değildir. Özet olarak geleneksel yöntemler mevcut sorunları etkili bir şekilde çözememektedirler. Derin öğrenmenin devreye girdiği yer tam burasıdır, derin öğrenme görüntü ve video üzerinde var olan çeşitli zorluklara rağmen bu üst düzey özellikleri çıkarabilme kabiliyetine sahiptir. Basit bir ifade ile Derin öğrenme, birçok gizli katmana sahip sinir ağlarıdır. Bu katmanlar onlarca, yüzlerce, binlerce veya milyonlarca olabilir. Derin öğrenme, ağa görüntü beslendiğinde, kenarlar, şekil ve köşeler gibi basit özellikleri birleştirerek yüz, araba, tablo vs. gibi nesnelere öğrenebilme yeteneği vardır. Görüntü bütün katmanların üzerinden geçerken, bir sonraki katmana girdi olarak gitmeden her bir katman görüntüden bir basit özellik öğrenmektedir dolayısıyla katman sayısı arttıkça ağ daha karmaşık özellikleri öğrenebilme yeteneğine sahip olur ve en sonunda bütün özellikleri birleştirip görüntüdeki nesneyi tahmin eder. Derin öğrenme ismi, derin katmanların üst düzey özellik çıkarma kabiliyetlerinin olmasından gelmektedir. Şekil 3.10 Derin öğrenmenin çalışma şeklini grafiksel olarak gösterir.



Şekil 3. 10. Sinir ağının her katmandan özellikler nasıl öğrendiğinin bir örneği. (Ian G ve arkd. 2016)

3.3.1. Derin Sinir Ağlar

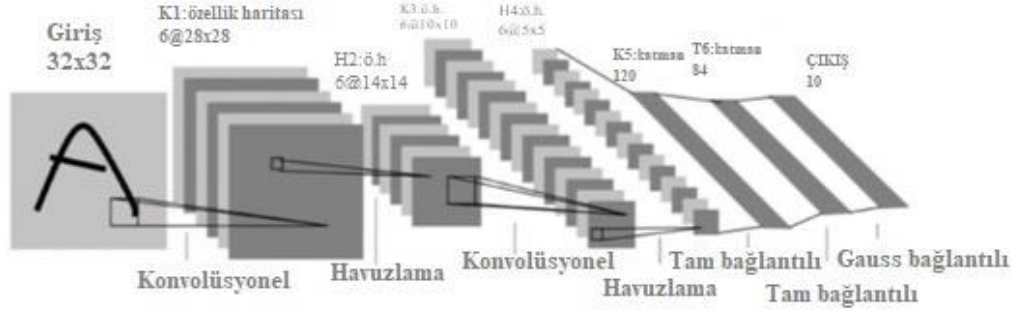
En popüler Derin Sinir Ağlar (DSA) yukarıda açıklanan YSA dayanmaktadır, DSA yapısı normal bir sinir ağından daha fazla gizli katmana sahip olacak şekilde ayarlanmıştır. Gizli katmana kaç katman ekleneceği, ilgili probleme bağlıdır.

Denetimli öğrenme ile eğitilen ve en az üç gizli katmana sahip olan Çok katmanlı algılayıcılar (Multi layer perception), DSA’da en yaygın kullanılan sinir ağlarıdır. Bu yapılarda gizli katmanların fazla olması, ağırlıkların ve nöronların sayısını da artırmaktadır. Bu karmaşık yapı ve düğümler nedeniyle, mevcut ağların eğitilmesi için çok miktarda veri ve çok yüksek hesaplama gücüne sahip bilgisayar donanımı gerekmektedir. Bölüm 4’te göreceğimiz gibi, DSA’mızı sıfırdan eğitememizin ana nedeni budur, kendi ağıımızı oluşturmak ve eğitmek için gerekli bilgisayar donanımına sahip değiliz. YSA’lar ve Geri yayılım metodu onlarca yıldır mevcut. Ancak, 2000’li yılların öncesinde, internet ve sosyal medya kullanımında patlama yaşanmadan önce, yeterli derecede eğitim verisi elde etmek zordu. Ayrıca YSA’ları doğru şekilde eğitmek için yeterince güçlü bilgisayarlar yoktu. Sevindirici haber 2006 yılında geldi. Hinton (2006) çalışmasında, geri yayılım yönteminin her bir katmanı bir Kısıtlı Boltzmann makinesi olarak görerek bir YSA’nın nasıl eğitileceği gösterilmiştir. O yıla kadar, YSA’nın eğitilmesi konusunda yapılmış çalışmalar olmasına rağmen, bu çalışma YSA ve uygulamalarına karşı yeniden ilgi duyulmasına sebep olmuştur. Son yıllarda büyük veri ve Grafik işleme birimlerinin (GPU) ortaya çıkışı, daha ilginç eğitim yöntemlerinin geliştirilmesine yol açmıştır. YSA ve DSA’nın insan seviyesine yakın bir başarı performansı elde etme kabiliyetlerinin olması, son zamanlarda araştırmacıların bu sistemlere ilgisini arttırmıştır.

3.3.2. Evrışimsel Sinir Ağlar

Evrışimsel sinir ağları (Convolutional Neural Networks), çok katmanlı algılayıcıların mimarisini kullanan benzersiz bir DSA türüdür. ESA genellikle Bilgisayar görüşündeki (CV) sorunları çözmek ve görüntülerdeki nesnelere tanımak için kullanılmaktadır. ESA, hayvanların ve insanların görsel korteksinin işleyiş şeklini taklit eder. Mevcut ağlar son zamanlarda Bilgisayar görüşü ve görüntü analizi gibi konularda yüksek başarı elde etmişlerdir. ESA, matematikte “evrışim” adı verilen doğrusal işlem kullandığı için “evrışimsel” ismi verilmiştir. Alışılmış matris çarpımı yerine, katlanmaların bir veya daha fazlasında evrışim işlemini kullanırlar. Bu ağlar, bir görüntüdeki nesneyi tanımak için birlikte çalışan birçok farklı katmana sahiptirler, Şekil 3.11’de ESA’nın yapısı gösterilmektedir. Girdi görüntüsünü alarak çalışmaya başlar, ilk gizli katmanda nöronlar kenarlar, kulakların şekli veya burun gibi basit kalıpları öğrenebilirler. Sonra, görüntü bir sonraki katmana iletilir ki bu katman bir önceki katman ile kıyaslandığında daha belirgin özellikleri öğrenebilmektedir.

Katmanlar arttıkça nöronlar, yüz veya araba gibi üst düzey özellikleri öğrenmeye başlar. ESA'nın çalışma prensibi, standart YSA'dan biraz farklıdır. Bir katmandaki her bir nöron görüntünün sadece küçük bir bölümüne yoğunlaşır, tek bir nöron belli özelliğe odaklanır ve görüntünün başka bir bölümünde çeşitli özelliklerin aranması için farklı nöronlar görevlendirilir.



Şekil 3. 11. ESA'nın yapısı. Görüntülerin boyutu, tam bağlı katmana varıncaya kadar düşmektedir.

Çok Katmanlı Algılayıcılar gibi, ESA da Gradyan İnişli algoritması kullanılarak geri yayılım yöntemiyle eğitilir. Tipik bir ESA mimarisi evrişimsel katman (convolutinal layer), ReLu katmanı (ReLU layer), havuzlama katmanı (pooling layer) ve tam bağlı (fully connected) katmanlarından oluşmaktadır.

Evrişimsel katmanı: Evrişimsel katman, ESA'nın ilk gizli katmanıdır. Bu katmanda bir görüntü, filtrelenmiş görüntüler yığına dönüşür. Genelde ESA'lar ağdaki ağırlık sayısını azaltır ve böylelikle daha hızlı eğitilebilir hale getirilirler. Kernel olarak da bilinen, bir ağırlık seti olan filtre, bir sonraki katmana göndermek üzere daha küçük ağırlık seti üretmek için giriş görüntüsü ile birleştirilir.

ReLU katmanı: ReLU katmanının amacı tüm negatif değerleri sıfır olarak değiştirerek ve tüm pozitif değerleri koruyarak görüntünün piksellerini normalleştirmektir. ESA mimarisinde genelde Relu katmanı ikinci veya dördüncü katmandır.

Havuzlama katmanı: Havuzlama katmanı normalde evrişimsel katmandan sonra gelir (Castelluccio vd., 2015). Bu katmanın amacı, bir önceki katmandan elde edilen görüntü yığınlarını ve parametreleri azaltmaktır. Bu sayede ağdaki uyumsuzluk problemi kontrol altına alınır. Bu katmanda 2 veya 3 boyutunda bir çerçeve oluşturulur ve belli bir adım atma değerine göre görüntü üzerine gezdirilir. Her geçişte, maksimum değer korunurken, geriye kalan değerler atılır. Bu tarz bir işlem, orijinal görüntüden

yaklaşık % 75 kadar daha küçük boyutlu bir görüntü verecektir. Küçük görüntüler ağır eğitim süresini ve ezberleme olasılığını azaltır.

Tam bağlı katman: İsminden de anlaşılacağı üzere, bu katmanın her nöronu kendinden önceki katmanın bütün nöronlarıyla bağlıdır. Her zaman ESA mimarisinin son kısmındadır ve çıkış katmanı olarak kullanılabilir.

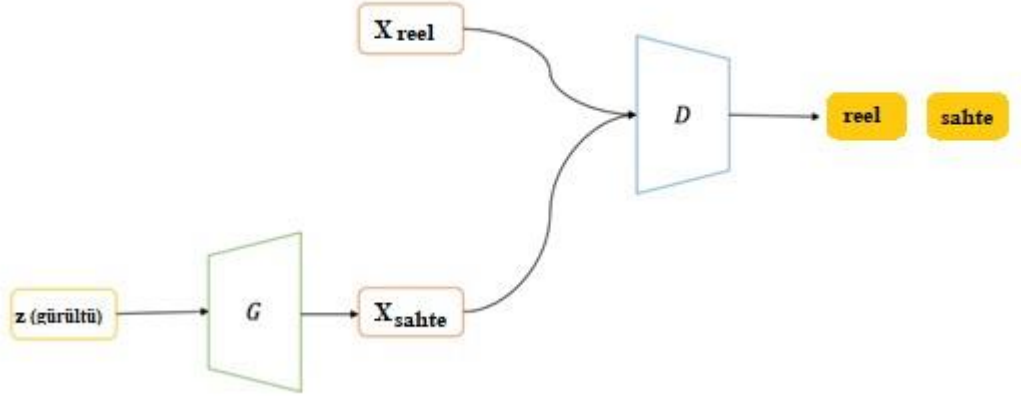
3.4. Üretken Modeller (Generative Models)

Üretken modeller uzun zamandır makine öğrenmesinde önemli ve aktif bir araştırma alanı olmuşturlar. Üretken Modeller, bir veri kümesinin doğal özelliklerini öğrenen ve mevcut verilerden ayırt edilemeyen yeni örnekler üreten modellerdir. Bu modeller Denetimsiz öğrenme kullanılarak eğitilirler. En popüler Üretken modeller arasında Naiv Bayes, Derin İnanç Ağlar (Deep Belief Networks), Varyasyonel Otomatik Kodlayıcılar (Variational Autoencoders) ve Çekişmeli Üretici Ağlar (Generative Adersarial Networks) bulunmaktadır. Biz tez çalışmamızda Çekişmeli Üretici Ağlar üzerinde odaklanacağız.

3.4.1. Çekişmeli Üretici Ağlar

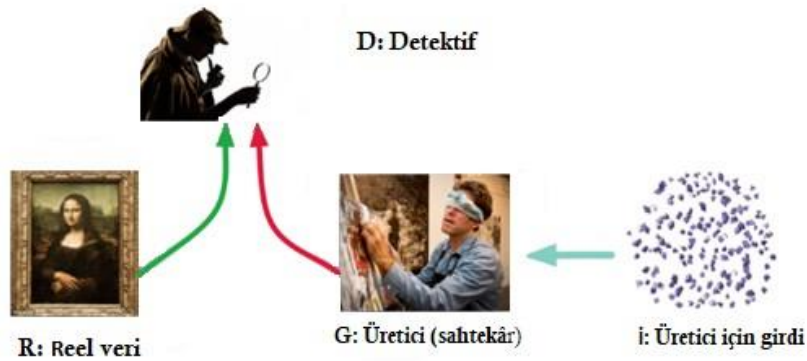
Çekişmeli Üretici Ağlar iki rekabetçi sinir ağından oluşan bir Makine Öğrenmesi algoritmasıdır. Yaygın olarak Görüntü işleme uygulamalarında kullanılmaktadır. Bu uygulamalar arasında; fotogerçekçi görüntü üretme, metinden görüntü üretme (Reed vd., 2016), çözünürlük artırma, video tamamlama (Mathieu, Couprie ve Lecun, 2015) ve çok daha fazlası bulunmaktadır. Görüntü işleme alanındaki kullanımlarının yanı sıra, ÇÜA'lar müzik üretme, metin üretme ve ses iyileştirme gibi diğer görevlerde de kullanılmıştır. İlk olarak İ. Goodfellow tarafından geliştirilen bu ağlar, kısa sürede araştırmacıların ilgisini çekmeyi başarmıştır (Goodfellow vd., 2014).

Kısaca, ÇÜA, mevcut verilerden öğrenerek yeni örnekler üretmek için kullanılmaktadırlar. İlgili ağlar, üretici ağ G (generator) ve ayırmacı ağ D (discriminator) olmak üzere iki ağa sahiptirler. Üretici ağ G, doğrudan örnekler üretmektedir. Ayırmacı ağ D ise, eğitim verilerinden alınan örnekler ile üretici ağdan alınan örnekler arasında ayırım yapmaya çalışır. Şekil 3.12'de modelin mimarisi gösterilmiştir.



Şekil 3. 12. ÇÜA mimarisi.

Üretici G girdi olarak bir gürültü (noise) vektörü z alır ve x (sahte) örnekleri üretir. Ayrımcı D hem üreticiden hem de x (reel) eğitim verilerinden örnekler alır ve iki kanaldan aldığı verileri karşılaştırarak üretici G den aldığı verinin gerçekliğini kontrol eder, böylece 0 ila 1 arasında bir çıktı değer verir. Çıktı değerinin 1 olması özgünlük 0 olması ise sahte anlamını taşır. Nitekim üretici gittikçe daha gerçekçi örnekler üretmeyi öğreniyor, ayrımcı ise üretilen verileri gerçek verilerden ayırt etmede daha iyi olmayı öğreniyor. Dolayısıyla üretici ve ayrımcı dönüşümlü olarak eğitiliyor. Goodfellow, ÇÜA'ın çalışma şeklini basit bir sahtekâr/detektif örneği ile açıklamıştır: Üretici ağ, reel (gerçek) tabloya benzeyen tabloyu üretmek için uğraşan bir "sahtekâr" dır. Ayrımcı ağ ise, gerçek ile sahteyi ayırmaya çalışan bir "detektif" dir. Dolayısıyla "sahtekâr" üreticinin gözleri bağlıdır ve orijinal veriyi hiçbir zaman göremez, sadece "detektif" ayrımcının kararlarını duyma yeteneği vardır (Goodfellow vd. 2014 ; Nag, 2017). ÇÜA basit çalışma şeklinin görseli şekil 3.13'de gösterilmiştir.



Şekil 3. 13. ÇÜA çalışma prensibi'nin basit örneği.

ÇÜA'ın kullandığı bu yöntemin aslı “Minimax Ağacı” adı verilen bir oyun teorisine dayanmaktadır. Bu teori, karar verme mekanizması olarak bir tarafın kaybı karşı taraf için kazanç olan satranç, tictactoe vs. gibi oyunlarda kullanılır ve asıl amacı oyundaki kaybı en alt düzeye düşürmektir. Bizim tezimiz dâhil olmak üzere ÇÜA ile birçok probleme çözüm üretilebilirken, bu yöntemin sahte habercilik ve buna benzer durumlarda suistimal edilebileceği aşikârdır.

3.4.2. Çekişmeli Üretici Ağlar'ın zorlukları/dezavantajları

ÇÜA'da var olan bazı yetersizlikler-dezavantajlar, 2014 yılından bu yana araştırmacıların çözüm hedefi haline gelmiştir. Ana dezavantajlar ÇÜA'ın eğitimi ile ilgilidir. Birincisi ÇÜA'ın eğitimi hiper-parametrelere bağlıdır. İkincisi ve en önemlisi, kayıp fonksiyonları (hem üreticinin hem de ayırmacının) bilgilendirici değildir. Basit ÇÜA mimarisinin bir diğer zorluğu da işlevsellik ile ilgilidir. Şekil 2.2'de gösterildiği gibi orijinal çapraz entropi (cross entropy) kaybını kullanarak, hangi verilerin üretileceği kontrol etmek mümkün değildir, kategorik veri üretilemez ve özellik olarak kullanmak amacıyla gizli uzaya erişilemez. Kategorik veri üretmek, ÇÜA'lar için özellikle zor bir problemdir. Bu problemi, Goodfellow, anlaşılması kolay bir şekilde açıklamıştır: yapay veriler devamlı (continuous) sayılara dayanıyorsa küçük değişiklikler yapabilirsiniz. Ancak tam sayılara dayanıyorsa, küçük bir değişiklik yapmanın yolu yoktur. Örneğin, piksel değeri 1,0 olan bir görüntü çıktılırsanız, sonraki pikselde bu piksel değerini 1.0001 olarak değiştirebilirsiniz. Ancak “Penguin” kelimesini çıktılırsanız, bir sonraki adımda “penguin + .001” olarak değiştiremezsiniz, çünkü “penguin + .001” diye bir kelime yoktur. Bu durumda penguenden deve kuşuna kadar her şeyi denemeniz gerekmektedir. Temel fikir şudur ki, üreticinin bir varlıktan (ör. Penguenden) diğer varlığa (ör. Deve kuşuna) gitme yeteneği yoktur.

3.4.3. Alternatif Çekişmeli Üretici Ağlar

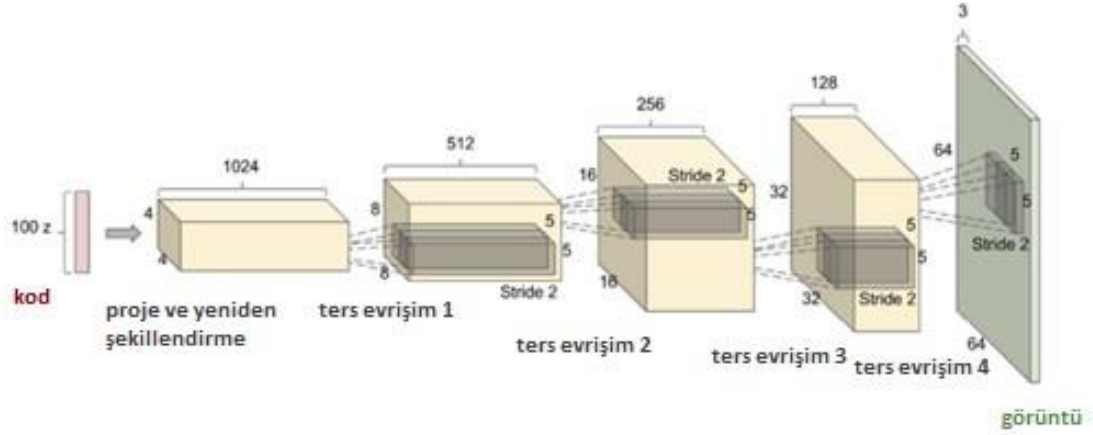
Özgün ÇÜA ile ilgili sorunları çözmek için, araştırmacılar, bir takım farklı eğitim yaklaşımları ve mimarileri geliştirmişlerdir. Yazının devamında, daha yaygın kullanılan alternatif ÇÜA hakkında kısaca bilgi verilirken, tezimizde kullandığımız Derin Evrimsel ÇÜA hakkında yazının son kısmında daha kapsamlı bilgi verilmektedir.

Koşullu ÇÜA'lar(Conditional GANs), ÇÜA'ların ilgili uzantılarıdır. Bu ağlarda, üretici ve ayırıcı sınıf etiketleri gibi bazı dış şartlara bağlıdırlar (Mirza, Mehdi, ve Osindero, 2014). ÇÜA'lar problemlerimizin çoğunu çözüyor gibi görünse de, verilerin gizli uzay temsillerine-gösterimlerine (representaion) erişmemize-ulaşmamıza olanak tanımıyorlar. Bu gösterimleri bulmak, yalnızca gizli uzayda sürekli hareket ederek hangi verilerin üretileceğini kontrol etmek için değil, aynı zamanda özellik çıkarma için de faydalı olabilir. İki yönlü ÇÜA'lar (Bidirectional GANs) bu sorunu çözme amaçlı bir girişimdir. Sadece üretici ağı değil, aynı zamanda verileri üretici (generator) gizli uzayına eşleyen bir kodlayıcı ağı öğrenerek çalışmaktadır. Bilgi (Info GANs) ÇÜA'lar, ÇÜA'nın bilgi-teorik bir uzantısıdır (Chen vd., 2016) . BÇÜA'lar bazı denetimli metotlar ile öğrenilen temsillerle rekabet edebilecek anlamlı temsilleri(representations) öğrenme yetenekleri vardır. Süper Çözünürlük ÇÜA'lar ilk olarak (Ledig vd., 2016) çalışmasında önerilmiştir. SÇÇÜA'lar, derin ağı çekişmeli ağ ile birleştirerek daha yüksek çözünürlüklü görüntüler üretmeyi amaçlar.

Derin Evrişimsel ÇÜA(Deep Convolutional GANs) ilk olarak 2016 yılında Radford ve arkadaşları tarafından önerilmiştir. DEÇÜA, ÇÜA'nın popüler ve başarılı ağ tasarımlarından biridir. Genelde maksimum havuz katmanı olmadan veya tamamen bağlı katmanlar olmadan evrişim katmanlarından oluşmaktadır (Radford, Metz ve Chintala, 2016).

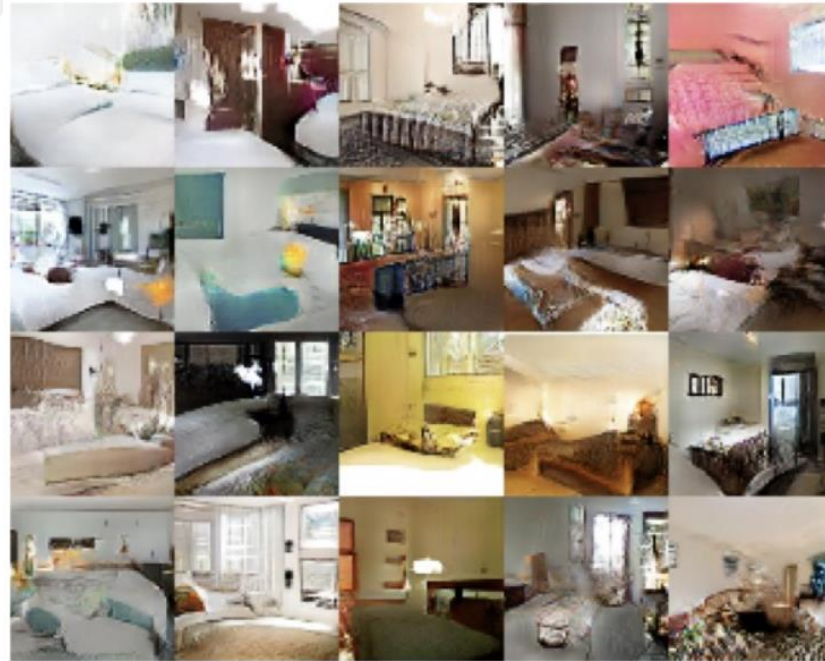
DEÇÜA mimarisinin bazı kilit noktaları şunlardır:

- Ayırıcı ve üreticinin çoğu katmanında yığın normalleştirme (batch normalization) katmanları kullanılmalıdır. Üreticinin son katmanı ve ayırıcının ilk katmanda yığın normalleştirme kullanmayınız, böylece model veri dağılımının doğru ortalamasını ve ölçeğini öğrenebilir (Ioffe ve Szegedy, 2015).
- Genel ağ mimarisi, evrişimsel ağlardan esinlenerek oluşturulur. Bu mimari hiçbir havuz katmanı içermemektedir. Üretici, temsilin uzaysal boyutunu artırması gerektiğinde, 1'den büyük bir adım ile aktarılmış evrişim (transposed Convolution) kullanır (Springenberg vd., 2015).
- Momentum ile Olasılıklı Dereceli Azalma yerine Adam optimizer kullanılmalıdır.



Şekil 3. 14. DEÇÜA’da kullanılan Üretici Ağı (Radford ve arkd. 2015)

ÇÜA’ların düşük sayıda eğitim turu ile (epoch) yüksek çözünürlüklü görüntüler oluşturmayı öğrenen ilk modeli DEÇÜA'lardır. Şekil 3.14'de gösterildiği gibi, sınırlı görüntü alanlarında eğitildiklerinde, DEÇÜA’lar, düşük eğitim turu ile orta kaliteli görüntüler üretebilmektedirler. Ancak sentetik yüz üretiminde bu durum değişmektedir, sentetik yüz üretiminde eğitim turu sayısı binlere hatta on binlere kadar çıkabilmektedir.



Şekil 3. 15. DEÇÜA tarafından LSUN veri seti kullanılarak üretilen yatak odası örnekleri (Radford, Metz ve Chintala, 2016).

BÖLÜM 4

METODOLOJİ

Bu bölümde, sentetik yüz üretimi ve yüz tanıma sistemimizi oluşturmak için seçilen yöntem ve algoritmaların tasarım ve uygulamasının detaylı bir tanımını vermek amaçlanmıştır. Sentetik yüz üretimi için bir önceki bölümde bahsedilen sebeplerden dolayı modelimizi DEÇÜA dayalı oluşturuyoruz.

Yüz tanıma farklı sorunların bir birleşimidir. Bu nedenle sistemi adım adım geliştirmek zorunda kaldık. Her bir aşamadaki sorunları teker teker çözerek elde edilen sonuçları bir diğer aşamaya taşıdık. Yüz tanıma sistemimiz dört ana aşamadan oluşmaktadır. Bunlar sırasıyla: Yüz tespiti, yüz hizalama, 128 boyutlu yüz temsillerin (face embeddings) üretilmesi, doğrulama, sınıflandırma ve kümeleme. Yüz tespit etme işlemleri için Histogram of Gradient algoritması ve OpenCV kütüphanesi uygulanmıştır. Yüz hizalam aşamısında 68 yer işaretçi (landmark) algoritması kullanıldı. Görüntülerden yüze ait 128-B temsilleri (Schroff, Kalenichenko ve Philbin, 2015) çalışmasında NN4 adlı modelinden yararlanarak elde edilmiştir. Sınıflandırma işlemi için Destek Vektör Makineleri, kümeleme için ise t- dağıtılmış Stokastik Komşu Gömme (t-distributed Stochastic Neighborhood Embedding) algoritmaları uygulanmıştır.

4.1. Sentetik Yüz Üretimi

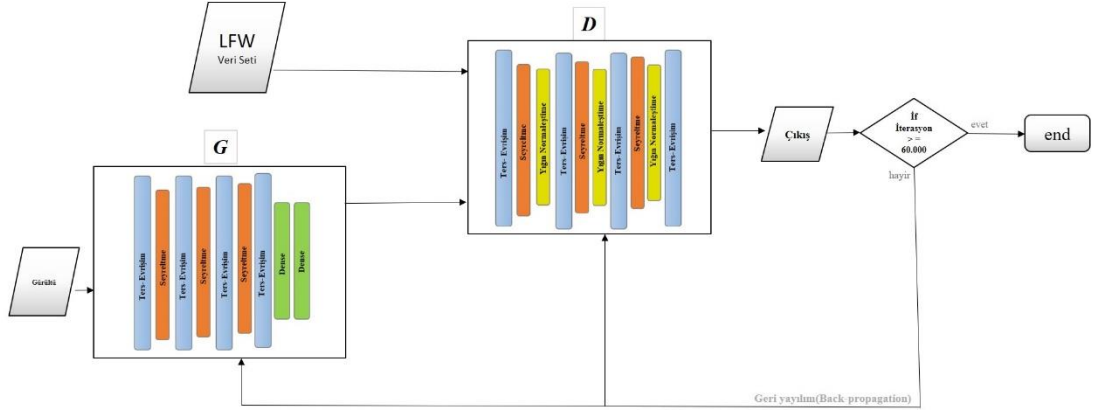
İlk aşamadaki işlemimiz, LFW (Labeled faces in the wild) veri seti üzerine DEÇÜA uygulayarak sentetik yüzler üretmektir. Temel ÇÜA gibi DEÇÜA da üretici ve ayırıcı olmak üzere iki sinir ağından oluşmaktadırlar. Mevcut ağlardaki ana fikir, her iki ağ parçasının da birbirleriyle rekabet ettiğidir bir bakımdan da birbirlerine yardımcı olduklarıdır. Ayırıcı daha iyi hale geldiğinde, üreticinin de daha iyi olması gerekmektedir, aksi halde ayırıcıyı daha fazla aldatma şansı kalmaz. DEÇÜA fikri teoride basit olsa da, işe yarayan bir model oluşturmak pek de kolay değildir. Ancak sadece teori veya saf mantıktan ziyade, gözlem ya da deneyim yoluyla oluşturulduğunda hem üretilen numunelerin kalitesi hem de yakınsama hızı açısından iyi sonuçlar ortaya koymaktadır (Chavdarova ve Fleuret, 2018). Dolayısıyla DEÇÜA modelini oluştururken (Radford, Metz ve Chintala, 2016; Atienza, 2017) çalışmalarından esinlenerek Keras ve Tensorflow kütüphanelerini kullanıyoruz.

DEÇÜA modelimizi oluşturmaya başlamadan önce veri setimizi hazır hale getirmek zorundayız. Bilgisayarımızın donanım gücünü göz önünde bulundurarak veri setimizdeki görüntülerin boyutlarını 256x256 piksel'den 130x130 piksel'e indirerek veri setimizi bir ön işleme tabi tutma mecburiyetindeyiz. Veri setimizi hazır duruma getirdikten sonra DEÇÜA modelini oluşturmaya başlıyoruz. Yazının devamında DEÇÜA modelimizin özellikleri hakkında bilgi verilecektir.

Ayrımcı ağımızın ilk 3 katmanı ters evrişim katmanlarından oluşmaktadır her üç katmanın arasında da seyreltme (dropout) katmanları mevcuttur. Ağın son iki katmanı ise yoğun (dense) katmanlardan oluşur. Son katmanda Sigmoid olmak üzere diğer katmanlarda sızıntılı ReLu (leakyReLu) aktivasyon fonksiyonu kullanılır.

Üretici ağımız 4 ters evrişimli katmanından oluşmaktadır. Son katman hariç her katman arasına seyreltme ve yığın normalleştirme (batch normalization) katmanları yerleştirildi. Ayrımcı ağımızda olduğu gibi üretici ağımızda da son katmanda Sigmoid diğer katmanlarda ise sızıntılı ReLu (leaky ReLu) aktivasyon fonksiyonu kullanıldı. Hem ayrımcı hem üretici olmak üzere her iki ağımızda kernel boyutu (kernel size) 5x5, kaydırma adımı (strides) 1 ve dolgu (padding) same olarak ayarlandı.

Ayrımcı ağ, gerçek görüntüleri alırken, yüksek değerleri hesaplamayı öğrenmelidir (1'e yakın). Sahte görüntüleri alırken de düşük değerleri (0'a yakın) hesaplamalıdır. Bu iki tür kaybı hesaplamak için ikili çapraz entropi (binary cross entropy) kayıp fonksiyonu kullanılır. Her iki ağ için optimizasyon algoritması olarak RMSprop (Root Mean Square Propagation) algoritmasını kullanıldı. Momentum değeri her iki ağ için 0.5 ayarlanırken öğrenme hızı ise ayrımcı ağ için 0.0001 üretici ağ için 0.0002 ayarlanmaktadır. Son olarak eğitim aşamasında kullanılmak üzere yığın boyutu (batch size) 64, eğitim adımı (training step) 60.000 olarak ayarlanır. Şekil 4.1'de DEÇÜA modelimizin akış şeması gösterilmektedir.



Şekil 4. 1. DEÇÜA modelin akış şeması.

4.2. Yüz Tespit Etme

İkinci aşama ise görüntüdeki yüzler tespit etmek ve sonrasında tespit edilen yüzleri hizalamaktır. Bu işlemi uygulamamız için literatür’de pek çok farklı yöntem bulunmaktadır. İncelememiz sırasında en iyi yüz algılama yöntemlerinden birinin standart Viola-Jones algoritması olduğunu tespit ettik (Viola ve Jones, 2001). Ancak daha kapsamlı araştırma yapıldıktan sonra, bizim istediğimiz kadar güvenilir olmadığını gördük, yüzler farklı bir pozisyonda veya kötü aydınlatma koşullar altında bulduklarında, algoritmanın yüzleri algılamakta zorlandığı gözlemlendi. Diğer yüz algılama yöntemleri ayrıntılı bir şekilde incelendikten sonra, çeşitli zor koşullar altında başarılı performans sergilemesi nedeniyle sistemimizde Dlib kütüphanesinde Histogram of Oriented Gradient (Yönlendirilmiş Gradyanların Histogramı-HOG) tabanlı yüz tanımlayıcıyı (descriptor) kullanma kararı aldık (Dalal ve Triggs, 2015). HOG yöntemi, Viola-Jones yöntemi ile kıyaslandığında çeşitli aydınlatma, ölçeklendirme vs. gibi zorlu koşullar altında daha hızlı ve başarılı performans sergilemektedir (Geitgey, 2017).

HOG yönteminin çalışma prensibini araştırdığımız sırada, dlib kütüphanesinde ESA tabanlı yüz tanımlayıcının de mevcut olduğunu fark ettik. ESA tabanlı HOG tanımlayıcı, orjinal HOG dedektörünün aksine neredeyse her açıdan yüzleri algılama yeteneği vardır. ESA tabanlı HOG yönteminin, orjinal HOG tabanlı yöntemle aynı hızda çalışabilmesi için güçlü Nvidia ekran kartına sahip olmak gerekmektedir. Elimizde bulunan kısıtlı bilgisayar donanım gücü nedeniyle mevcut yöntemi sistemimize uygulayamadık. ESA tabanlı HOG tanımlayıcıyı genel amaçlı bir işlemci (CPU) üzerinde çalıştırdığımızda, doğruluk oranı yüksek olmasına rağmen orjinal HOG tanımlayıcıya göre oldukça yavaş olduğunu gözlemledik. Dolayısıyla,

projemizde 96x96 boyutunda hizalanmış RGB (Red, Green, Blue) yüz görüntüleri elde etmek için, yüz algılama yöntemi olarak orijinal HOG tanımlayıcıyı, dönüştürme ve kırma işlemleri için OpenCv kütüphanesini tercih ettik

Bu paragraf'da, HOG tanımlayıcının genel metodolojisi hakkında bilgi verilecektir. HOG yöntemi ilk olarak yaya tanıma sistemlerinde kullanılmak üzere önerilmiştir (Shashua, Gdalyahu ve Hayon, 2014). Çok uzun bir zaman geçmeden, HOG tanımlayıcıları, insan tanıma problemlerinde kullanılmak üzere uyarlandı (Dallal ve Triggs, 2015). HOG kavramı statik özellik çıkarma yöntemine dayanmaktadır.

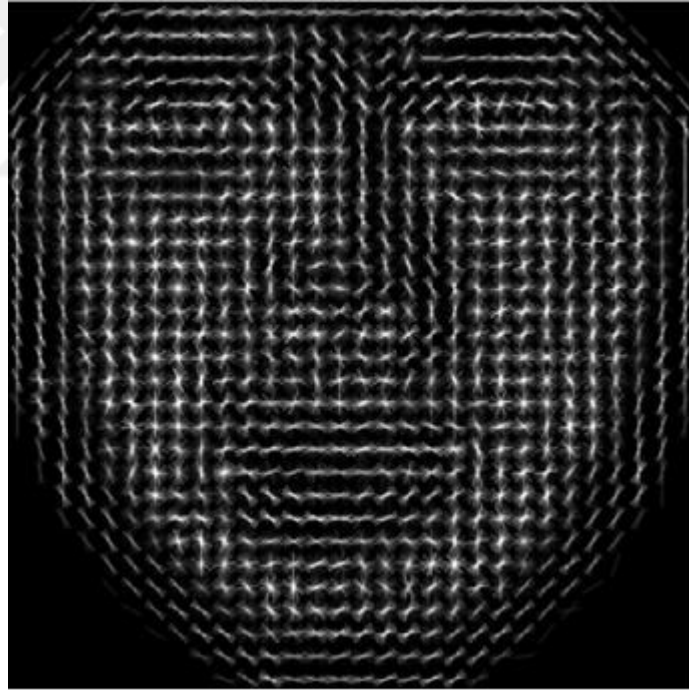
İlk aşama olarak, LFW veri setinde 250x250 boyutlarında olan görüntüleri 96x96 boyutlarına indiriyoruz, aynı zamanda renkli olan orijinal görüntüyü siyah beyaza çevirerek ön işlemden geçirmiş oluyoruz. Görüntüleri daha fazla ön işleme tabi tutmak için gamma düzeltme tekniğini de önerilmiştir (A.g.e.). Ancak performans tatmin edici olmadığı için gamma tekniğini uygulamayı uygun görmedik. Görüntünün ön işleme kısmı yapıldıktan sonra, ikinci adımımız 96x96 boyutunda olan görüntünün HOG tanımlayıcısını hesaplamaya başlamaktır. Görüntünün yatay ve dikey gradyanlarını hesaplamak için farklı yöntemler bulunmaktadır. Sobel işleci bu gradyanları hesaplamak için kullanılan en etkili yöntemlerden biridir. Daha sonra gradyanlar'ın yönünü ve büyüklüğünü 4.1 ve 4.2 denklemleri ile buluyoruz.

$$G(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)} \quad (4.1)$$

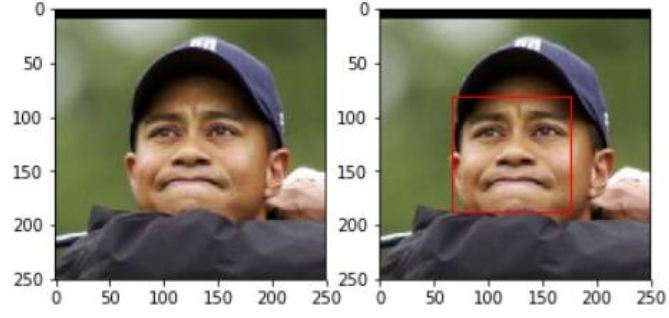
$$\alpha(x, y) = \tan^{-1}\left(\frac{G_y(x, y)}{G_x(x, y)}\right) \quad (4.2)$$

Yön ve büyüklük olmak üzere, her piksel bir gradyan ile değiştirilir. Bu işlem, görüntüdeki yüzü tespit etmek için kullanışlı olmayan birçok bilgiyi ortadan kaldırır ve görüntünün ana hatları gibi sadece gerekli olan bilgileri muhafaza etmektedir. Gradyanların yönü, görüntüdeki aydınlık bölgesinden daha karanlık (dark) bölgeye doğru yönelmektedir. Görüntünün bütün bölümleri gradyanlar ile temsil edildiklerine göre bir sonraki adıma geçebiliriz. Bu adımımızda, işlemleri kolaylaştırmak açısından 96x96 boyutunda olan görüntüyü ilk aşamada hücrelere bölerek, 16x16 boyutunda görüntüler elde ediyoruz, daha sonra aynı görüntüyü 8x8 boyutlarından oluşan 2x2 hücrelere bölmekteyiz. Bu bölme işlemleri, yüz temsillerini çeşitli gürültülere-

tıkanıklıklara(noise) karşı dayanıklı kılar. Daha sonra gradyan yönelimleri 9 kutulu histogramlara nicelendirilir. Son aşamada, nihai özellik vektörlerini seçmek için histogramları birleştiriyoruz. Bu işlem ile birlikte, orijinal görüntüyü basit bir özellik vektörüne dönüştürüyoruz. Özellik vektörü, görüntüde bulunan yüzün temel yapısını temsil etmektedir. Daha sonra HOG özellik tanımlayıcısını eğitmek amacıyla, bu işlemi yüze ait görüntüler ile yüzlerce kez tekrarlıyoruz. Tamamen eğitildikten sonra yeni bir yüz içerikli görüntüden yeni özellikler çıkarmak için kullanılabilir ve daha sonra görüntüde yüzün olup olmadığını sınıflandırmak için doğrusal Destek Vektör Makineleri algoritması uygulanır. Şekil 4.2, HOG yöntemi uygulanarak işleme tabi tutulmuş bir resmi göstermektedir. Şekil 4.3 ise HOG yöntemi ile OpenCV kütüphanesi kullanılarak görüntüden yüz resminin algılanmasını ve kare kutu (bounding box) içerisinde alınmasını göstermektedir. Yüz algılandıktan sonra bir sonraki aşama olan yüz hizalama bölümüne aktarılır.



Şekil 4. 2. Yüze ait bir görüntünün ana özelliklerinin HOG yöntemi ile gösterimi. (Felzenszwalb vd., 2009).

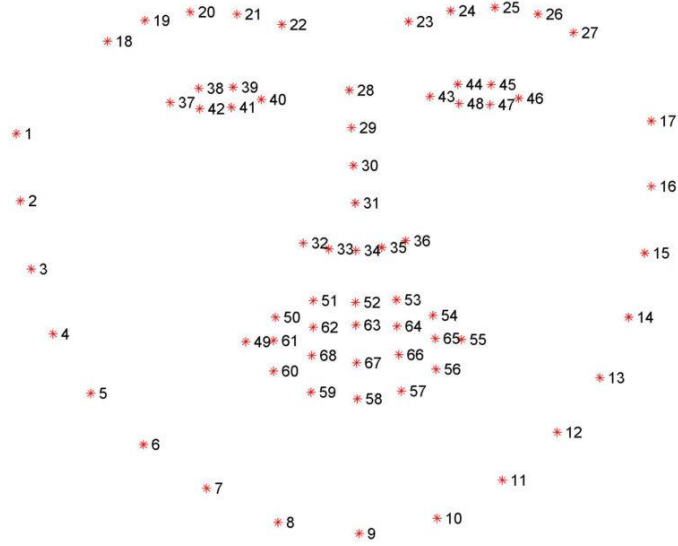


Şekil 4. 3. Orijinal görüntüden yüz kısmının algılanması ve sınırlayıcı kutu içerisinde alınmasının gösterimi.

4.3. Yüz Hizalama

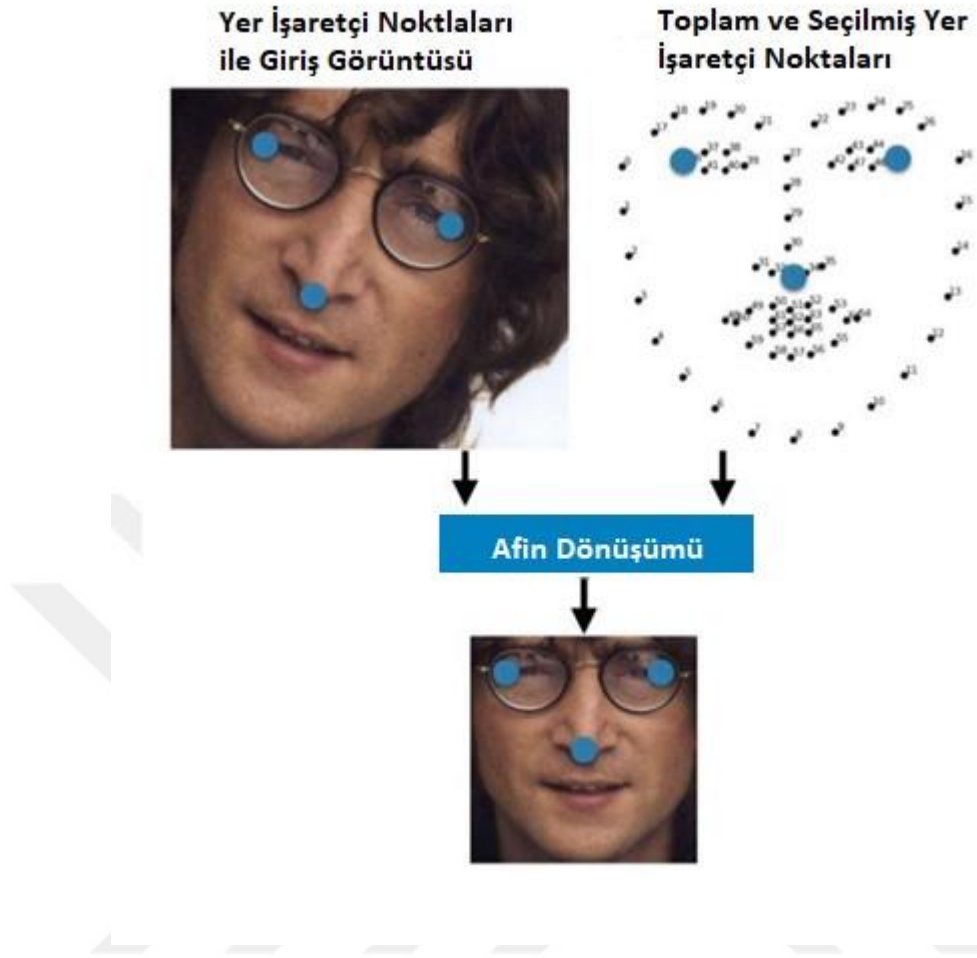
Yüz hizalama aşaması oldukça önemli bir aşamadır, çünkü bazı durumlarda fotoğraf çekildiğinde yüzler aynı konumda veya aynı yönde olmayabilirler dolayısıyla bilgisayar yüzleri farklı konumda ve yönde görüp farklı kişilere ait yüzler olarak algılama olasılığı bulunmaktadır. Yüzleri hizalamak sistem performansı açısından oldukça önemlidir. Literatür taraması kısmında, aslında birçok yöntemin mevcut olduğunu gördük ancak tüm yöntemler temel olarak ağız, sağ kaş, sol kaş, sağ göz, sol göz, burun ve çene gibi yüz bölgelerini konumlandırmaya ve etiketlemeye çalışmaktadır. Hızlı ve açık kaynak olması sebebiyle (Kazemi ve Josephine, 2014) çalışmada geliştirilen yer işaretçi (landmark) tahmin algoritmasını kullanmayı kararlaştırdık.

Bu yöntem ile şekil 4.4'de gösterildiği gibi yüz işaretleri olarak bilinen 68 özel nokta görüntüdeki belirli yüz bölgelerinden üretilmektedir.



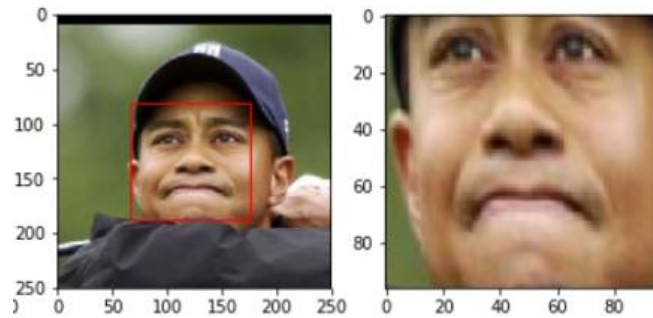
Şekil 4. 4. iBUG 300-W veri setinden üretilen 68 yüz işaretinin görseli.

Daha sonra makine öğrenme algoritması bu 68 noktayı öğrenmesi için eğitilir. Eğitim tamamlandıktan sonra, algoritmaya yeni bir görüntü verildiğinde, algoritma otomatik olarak görüntüdeki yüze ait yeni 68 yüz işareti üretmektedir. Gözler, ağız, burun ve yanaklar görüntünün üzerinde belirlendikten sonra, çevirme (rotate), ölçeklendirme gibi basit 2 boyutlu görüntü işleme dönüşümleri, görüntüde herhangi bir tıkanıklık olmadan yüzün düzgün bir şekilde merkezlenmesi için kullanılmıştır. Bu dönüşüme Şekil 4.5'da gösterildiği gibi afin dönüşümü adı verilir. Mevcut yöntemin çalışma prensibini araştırma esnasında, Helen veri setinde eğitilebilecek 194 yer işareti üreten yöntemlerin de mevcut olduğuna rastladık. Ancak tercih ettiğimiz yöntemin hızlı ve başarılı bir performans sergilemesi nedeniyle, diğer yöntemler ilgi odağımız dışında kalmıştır.



Şekil 4. 5. Görüntüdeki yüze afin dönüşümü işleminin uygulanma görseli. (Amos, Ludwiczuk ve Satyanarayanan, 2016).

Algoritma başarılı bir şekilde eğitildikten sonra, Şekil 4.6'da gösterildiği gibi yüzün yönüne bakılmaksızın, ağız burun ve göz kısımlarını yüz ile aynı konumda konumlandırabiliriz. Bu, yüz tanıma performansını daha doğru olmasını sağlayacaktır.



Şekil 4. 6. Algılanan yüz resminin yüz noktalarına göre hizalandırılmasının görseli.

4.4. 128-B Yüz Temsillerinin Üretilmesi

İki yüz arasındaki benzerliği veya farkı karşılaştırmak için, mevcut yüz temsillerinin (embedings) elde edilmesi gerekmektedir. Bu bölümdeki amacımız yüz içerikli görüntülerden yüze ait temsillerin çıkarılmasıdır. Yüz temsilleri, yüzün kimliğini temsil eden çok boyutlu sayısal bir vektördür.

Mevcut en başarılı temsill çıkarma mimariler hakkında kapsamlı inceleme yapıldıktan sonra, ESA'na (CNN) dayalı mimari kullanılması kararlaştırılmıştır. Kullanılan ESA mimarisi (Schroff, Kalenichenko ve Philbin, 2015) çalışmasında nn4 olarak adlandırılan ve (Szegedy vd., 2014) çalışmasından esinlenen Inception modeline dayanmaktadır. Bahse konu modeli tecih etmemizin ana nedeni yüz doğrulama ve tanıma işlemlerinde başarılı sonuçlar elde etmesidir. Önceden eğitilmiş mimarinin ağırlıklarının açık kaynak olması ve ilgili mimariyi sıfırdan eğitecek güçlükte bilgisayara sahip olmayışımız bu mimariyi tercih etmemizde bir diğer etken olmuştur. Kullanılan mimari Toplamda 22 katmandan oluşmaktadır ve literatür'de mevcut en iyi performans sergileyen diğer mimariler ile kıyaslandığında oldukça derin bir mimariye sahiptir. Diğer mimarilerin yapılarıyla kıyaslandığında ilgili mimarinin yapısında farklılıklar mevcuttur. Bu mimarilerde tek bir katmanda, çok sayıda “özellik çıkarıcı” bulunur. Bu durum dolaylı olarak ağın daha iyi performans göstermesine yardımcı olur, çünkü eğitim sırasında ağ ilgili görevi çözerken seçebileceği birçok seçeneği vardır. Yukarıda bahsedildiği gibi ilgili mimari, görüntüdeki yüzü 128 boyutlu vektör uzayına çıkarmak için önceden eğitilmiştir, dolayısıyla bir görüntüyü ağa yüklediğimizde yüze ait 128 temsil üretmektedir. Mimarinin tamamı model.py uzantılı olarak Martin Krasser tarafından 2018 yılında açık kaynak olarak paylaşılmıştır. Önceden eğitilmiş modelin ağırlıkları ise (<https://cmusatyalab.github.io/openface/models-and-accuracies/#pre-trained-models>, 2016) web sitesinde nn4.small2 isminde açık kaynak olarak mevcuttur. Şekil 4.7'de veri setimizdeki bir kişiye ait 128 yüz temsili gösterilmiştir.



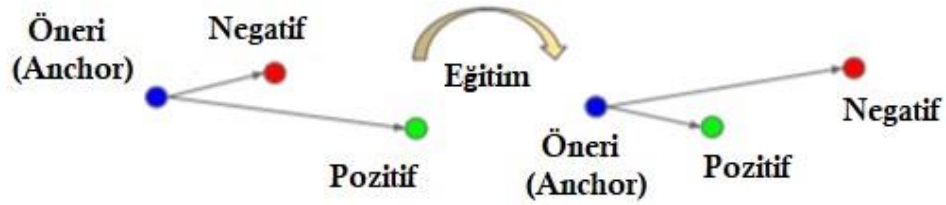
```
[ 0.02374097 0.27338293 -0.08832834 -0.02017033 -0.03959543 0.10389523
-0.00433165 -0.13540013 0.06080468 0.04500411 0.12436974 -0.04816889
-0.10898244 -0.04962858 0.001954 -0.13249727 -0.06721593 0.0388394
0.03809119 -0.12826076 -0.11662748 0.11496108 -0.05124535 -0.07089288
0.0692036 -0.10371653 0.10031309 0.11778706 0.0871548 0.0979897
0.01406911 -0.12927455 -0.03523886 -0.0891747 0.05126474 -0.02049835
-0.0416955 -0.12082877 0.10998838 0.16013248 -0.03772257 -0.03804448
-0.10523985 0.07586426 -0.05832255 0.03664732 -0.04430651 0.03637106
-0.01409994 -0.00979992 0.03453114 0.32950062 -0.05584567 -0.05129259
-0.17865911 0.00793244 -0.04987326 0.04181097 -0.12140195 0.08732796
-0.10436495 -0.04223024 0.19682471 -0.02833547 -0.05046778 -0.06155857
0.09020671 -0.044496 -0.07585879 0.12682435 0.09720209 0.03654587
0.02258595 0.04431985 0.01364174 -0.05537705 -0.04788952 -0.08790142
-0.1001888 0.04200108 0.01681684 0.03017174 0.0385743 0.07505066
0.01457727 0.00827035 -0.08118908 -0.14483356 0.00942834 0.02633035
-0.03568657 -0.10779966 0.00042803 0.03783602 0.06515275 -0.20981832
-0.0216811 -0.10797793 0.04932071 -0.00627633 -0.02696804 0.11923192
0.0163664 -0.01008751 -0.0505329 0.07923488 0.11772538 0.04633122
0.01298601 0.06369941 0.16918525 -0.04993364 0.00640496 0.12529694
-0.07358854 -0.04600717 -0.00926945 0.09514777 0.18824673 -0.01311349
0.0501525 -0.12385129 -0.04699646 -0.01096694 0.13203394 -0.00887628
0.07181319 0.02416481]
```

Şekil 4. 7. 128 yüz temsillerinin görseli.

İlgili mimarının eğitimi şu şekilde gerçekleşmektedir: ilk olarak derin sinir ağına aynı kişiye ait iki görüntü yüklenmektedir, akabinde ise farklı kişiye ait bir görüntü yüklenir. Ardından algoritma 3 görüntünün yüze ait 128 temsilini oluşturur. Bundan sonraki aşamada, ilk 2 görüntünün temsillerini karşılaştırır ve sinir ağı parametrelerini güncelleyerek ilk 2 görüntünün temsil ölçümlerinin birbirine yakınlaşmasını, birinci ve üçüncü görüntülerin arasında temsil ölçümlerinin de birbirinden uzaklaşmasını sağlar. Başka bir deyişle, Şekil 4.8'de gösterildiği gibi aynı kişiye ait Öneri (Anchor) ve pozitif (Positive) olarak adlandırılan görüntülerin yüz temsilleri arasındaki mesafe daha düşük, Öneri ve Negatif (farklı kişiye ait) görüntüler arasında yüz temsilleri mesafesi ise daha yüksektir. Mevcut sinir ağı üçlü kayıp fonksiyonu kullanmaktadır, dolayısıyla bu sürece üçlü kayıp eğitimi denmektedir. Şekil 4.8'de

öğrenme sürecinden önce ve sonra temsillerin birbirine yaklaşma ve uzaklaşma süreci gösterilmektedir.

Dolayısıyla, farklı kişilere ait milyonlarca görüntü ile eğitim sürecini tekrarlanmaktadır. Eğitim sürecinin sonunda, algoritma'nın tek kişiye ait 10 farklı görüntü için benzer temsiller üretmesi beklenmektedir. Daha önce de belirtildiği gibi, bu derin sinir ağının eğitimi Nvidia Tesla gibi maliyetli ve yüksek performanslı ekran kartıyla bile haftalar sürmektedir. Elimizde bur tür imkânlar bulunmamaktadır, neyse ki uygulamış olduğumuz bu yüz tanıma yöntemi önceden eğitilmiştir ve ağırlıkları açık kaynak olarak internet ortamında mevcuttur. Dolayısıyla sıfırdan eğitmemize gerek kalmayarak deneyler bölümünde sadece veri setlerimizin %90'lık kısmını eğitiyoruz ve geriye kalan %10'luk kısmını test için kullanıyoruz.



Şekil 4.8. Üçlü kayıp: Öğrenme sürecinden önce ve sonra.

4.5. Doğrulama, Sınıflandırma ve Kümeleme

Yüz temsilleri elde edildikten sonra, aynı temsilleri özellik vektörleri olarak tanımlayarak yüz doğrulama, tanıma ve kümeleme gibi işlemleri kolayca uygulayabiliriz. Bu aşama, yüz tanıma sistemimizin son aşamasıdır.

Yüz doğrulama işlemi genellikle bire-bir (one-to-one) karşılaştırma işlemi olarak tanımlanır. Algoritma, görüntüye ait yüz temsillini veritabanında depolanan belirli bir görüntü temsilli ile karşılaştırarak, görüntüdeki ilgili kişiyi doğrulamakta veya yalanlamaktadır. Doğrulama işlemi Eşik skor (Treshold score) seçilerek uygulanır, eğer skor belirlenen eşiğin altındaysa pozitif olarak algılanır ve dolayısıyla ilgili kişi doğrulanır. Aksi bir durumda eğer skor belirlenen eşiğin üstündeyse negatif olarak algılanır. Skor, söz konusu iki görüntüye ait yüz temsillerinin arasında Öklid mesafesi uygulanarak hesaplanır. Düşük skor, tespit edilen yüzün, kişi tarafından yüklenen görüntüdeki yüze yakın olduğu ve dolayısıyla doğrulandığı anlamına gelir. Yüksek skor ise her iki görüntüdeki yüzün de farklı olduğu anlamına gelir. En uygun eşik

değeri bulmak için, yüz doğrulama performansının bir dizi mesafe eşik değeri üzerinde değerlendirilmesi gerekmektedir. Gerekli deneyler yapıldıktan sonra mevcut mimari için en uygun eşik değerin 0.64 olduğu saptandı. F1 skoru ise doğruluk yerine değerlendirme metriği olarak kullanıyoruz.

Modelimiz görüntüye ait 128 temsili çıkaracak duruma geldiği ve en uygun eşik değeri belirlendiği göz önüne alınırsa artık sisteme görüntü yüklediğimizde, sistem o görüntünün kime ait olduğunu belirlemeye hazır durumdadır. Bu aşamada birkaç basit makine öğrenme sınıflandırma algoritması uyguladık. Hız, doğruluk ve basit uygulama olanakları sunduğu için Destek Vektör Makinası (SVM) sınıflandırma algoritmasını tercih ettik. Görüntüdeki yüze ait yeni 128 temsilli elde etmek ve elde edilen temsilleri veri setinde bulunan temsiller ile karşılaştırmak için sınıflandırıcıyı eğittik. Görüntü temsili ve veri setimizdeki bütün temsiller karşılaştırılarak Öklid mesafesi hesaplanır böylece eğer sonuç eşik değerin altındaysa sistem mevcut kişiyi "doğru kişi" olarak algılar. Veri setimiz etiketli olduğundan dolayı sistem çıkış olarak algılanan kişinin ismini yazdırır.

Yüz kümeleme, insanların yüzlerini kümeler halinde gruplandırma yöntemidir. Bu kümelerin her biri birden fazla görüntü içermektedir. Literatürde birçok kümeleme algoritması mevcuttur. Kümeleme algoritmaları genel olarak hiyerarşik ve bölümsel olarak iki gruba ayrılırlar (Jain, 2010). Ancak birçok kümeleme algoritmasının odak noktası, yüksek boyutlu veri setindeki benzerliği boyutsallığı azaltacak şekilde tanımlamaktır. Kümeleme işlemini gerçekleştirmemiz için, (Maaten ve Hinton, 2008) çalışmasına dayanarak 128 temsil vektörleri üzerine T- dağıtılmış Stokastik Komşu Gömme (t-distributed Stochastic Neighborhood Embedding) algoritmasını uyguluyoruz. Böylece veri setimizde bulunan görüntüleri iki boyutlu uzaya dökerek kümelemiş olduk.

Bu alt bölümde yapılan işlemlerin sonuç diyagramlarını ve tablolarını detaylı bir şekilde 5. Bölümümüzde sunuyoruz.

BÖLÜM 5

DENEYLER, SONUÇ VE TARTIŞMA

Tezimizde yüz tanıma sistemimizin doğruluk, sınıflandırma ve kümeleme gibi işlemlerde başarı oranını artırma amacıyla DEÇÜA ile sentetik yüzler üretilmiştir. Bu bölümün ilk aşamasında üretilen sentetik yüz görüntüleri gösterilmiştir, ikinci aşamasında ise üretilen sentetik yüzlerin yüz tanıma sistemi üzerindeki performans etkisi değerlendirilmiş ve sonuçlar sunulmuştur.

5.1. Araştırma Araçlar

Araştırma, mevzuatı anlamak için veri toplama, analiz etme ve yorumlama olmak üzere sistematik bir süreçtir. Ayrıca araştırma, özellikle yeni bilgi keşfetmek veya yeni bir anlayışa ulaşmak için, bir konunun ayrıntılı bir çalışması olarak da tanımlanmaktadır. Araştırma yönteminin çözüm aradığımız probleme uygun olması gerekmektedir (Paul, Leedy ve Ormrod, 2014). Aşağıda kullanılan araştırma araçları liste halinde belirtilmiştir:

Donanım:

- RAM:8 GB
- İşlemci: Intel Core i7 CPU 6700 @ 3.40 GHz
- Ekran kartı: Intel® HD Graphics 530
- Sabit (Hard) Disk: 256 GB SDD

Yazılım:

- İşletim sistemi: Windows 10 Enterprise
- İşletim sistemi türü:64-bit
- Yazılım dili: Python 3.6.8, Anaconda Navigator, Jupyter Notebook 5.7.8

Derin Öğrenme Kütüphaneleri:

- Tensorflow 1.9.0
- Keras 2.2.4

5.1.1. Tensorflow

TensorFlow, derin öğrenme modelleri de dahil olmak üzere makine öğrenmesi algoritmalarını uygulamak için Google tarafından geliştirilen bir arayüzdür (Abadi vd., 2016). İsmi Tensor (Tensor) kelimesinden almıştır. Tensor tüm veri tiplerinin karşılığına denk gelen bir vektör veya n-boyutunda bir matris olarak açıklanabilir. Tensorflow'da bütün hesaplama işlemleri bir graf'ın (graph) içerisinde yapılır. Graf bir eğitim sırasında gerçekleşen bütün hesaplamaları tanımlamaktadır. Birden fazla CPU ve GPU üzerinde çalışacak şekilde tasarlanan Tensorflow tanıdığı bu ve buna benzer olanaklar sayesinde araştırmacıların ilgisini çekip kısa sürede en çok kullanılan kütüphaneler arasında girmeyi başarmıştır.

5.1.2. Keras

Keras, Python'da yazılmış açık kaynaklı bir sinir ağı kütüphanesidir. TensorFlow, Microsoft Cognitive Toolkit, Theano gibi kütüphaneler üzerinde çalışabilme yeteneğine sahiptir. Derin sinir ağları ile hızlı deneyler sağlamak için tasarlanan ürün, aynı zamanda kullanıcı dostu ve genişletilebilir olmaya odaklanmaktadır. Keras, görüntü ve metin verileriyle çalışmayı kolaylaştırmak için katmanlar, aktivasyon fonksiyonları, optimizasyon algoritmaları gibi bir dizi yaygın olarak kullanılan sinir ağı yapı taşlarının çok sayıda uygulamasını içerir. Bu kütüphane, klasik sinir ağlarına ek olarak, evrimsel (convolutional) ve tekrarlayan (recurrent) sinir ağlarını da desteklemektedir. DEÇÜA modelimizi oluşturmak için TensorFlow ve Keras kütüphanelerini kullandık.

5.2 Veri Seti

Çalışmamızda kullanılan veri setleri aşağıda işleme tabi tutulmamış orijinal LFW veri seti ve DEÇÜA ile çoğaltılmış veri seti olarak tanımlanmıştır.

5.2.1. Orijinal LFW veri seti

İşleme tabi tutulmamış Labeled Faces in the Wild (LFW) veri seti Amherst, Massachusetts Üniversitesi'ndeki araştırmacılar tarafından web üzerinden toplanan görüntüler ile oluşturulmuştur. LFW veri seti 5.749 kişiye ait toplamda 13.233 görüntüden oluşmaktadır. Mevcut veri seti web ortamında var olan görüntülerden oluşturulduğu için kısıtlanmamış (unrestricted) ortama ait görüntüler içerir. Dolayısıyla açığı, poz, yaş vs. gibi konularda farklılıklar söz konusudur. LFW veri setindeki görüntülerin boyutu 250 x 250 pikseldir. Birkaç görüntü hariç, çoğu görüntü renklidir ve JPEG formatındadır (Huang vd., 2008). Adı geçen veri seti açık kaynak

olarak <http://vis-www.cs.umass.edu/lfw/#download> web sitesinde mevcuttur. Şekil 5.1’de LFW veri setinden rasgele olarak seçilmiş görüntüler gösterilmektedir.



Şekil 5. 1. LFW veri seti örnekleri.

Çalışmamızda LFW veri seti sentetik yüz üretiminde, ESA’nın eğitiminde ve test aşamasında kullanıldı.

5.2.2. DEÇÜA ile çoğaltılmışı veri seti

Sentetik yüz üretme aşamasında, beklendiği gibi kişi başı görüntü sayısı yüksek olan %70’lik bölüm için, DEÇÜA modelimiz temel gerçeğe yakın sentetik yüzler üretmiştir. Diğer %30’luk kısımda ise kişi başı görüntü sayısı oldukça düşük olduğundan sentetik yüz üretimi verimli bir şekilde gerçekleşmemiştir. Görüntü sayısı düşük olan kısım için afin dönüşümü (affine transform) olarak da bilinen, görüntüyü döndürme (rotation), yansıtma (mirroring), ölçeklendirme (scaling) ve biçme

(shearing) gibi veri çoğaltma teknikler uygulayarak verileri çoğalttık. Böylece çoğaltılmış veri setinde görüntü sayısı 21.000'e çıkmış oldu. DEÇÜA ile üretilen en iyi kalitede sentetik yüz görüntüleri Şekil 5.2'de gösterilmiştir.



Şekil 5. 2. DEÇÜA ile üretilen en iyi sentetik yüzler.

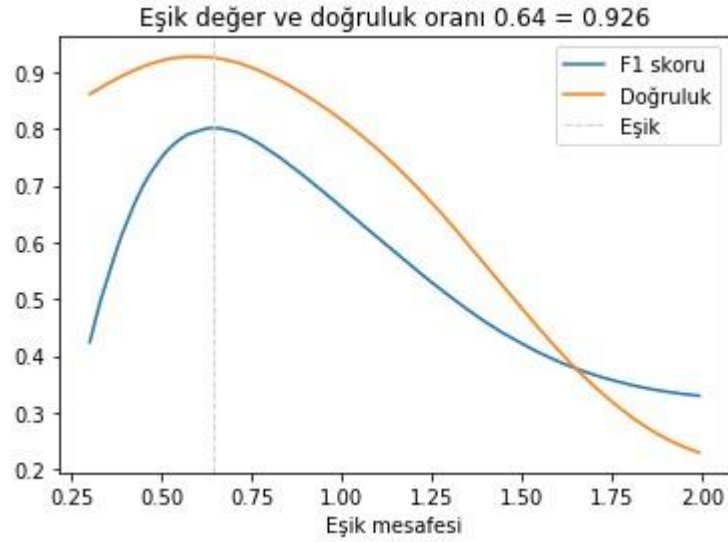
DEÇÜA modeli sadece 13.233 görüntü ile eğitilmesine rağmen elde edilen sonuç memnun ediciydi. DEÇÜA eğitim aşamasında, 3.000 eğitim turundan sonra görüntülerde yüz noktaları oluşmaya başlamış olup eğitim turu arttıkça sentetik yüzün daha belirgin hale geldiği gözlemlenmiştir.

Gerçekleştirilen deneylerde her iki veri setin %90'ını eğitim için kalan %10'luk bölümü sistemi test etmek için kullanılmıştır.

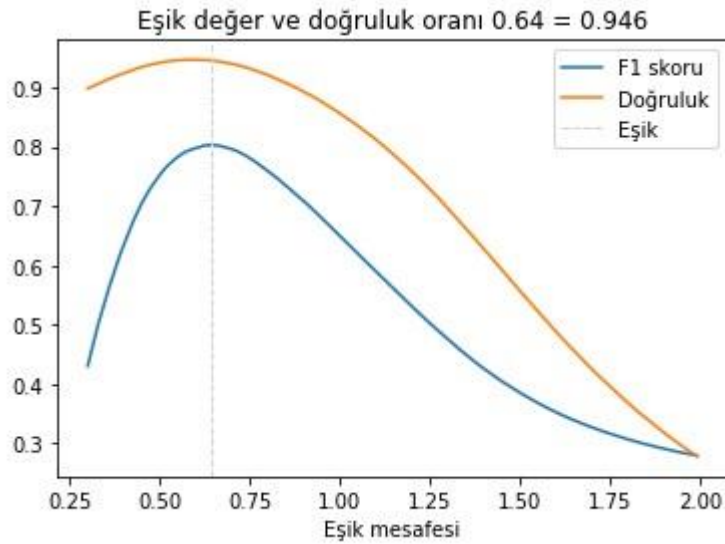
5.3. LFW veri setinde Yüz Doğrulama

Yüz doğrulama işleminde ana hedefimiz, LFW veri setine DEÇÜA uygulandıktan sonra sistemimizin yüz doğrulama performansında artış meydana gelip gelmediğini test etmektir. İlk aşamada orijinal veri seti üzerinde yüz tanıma sistemimizi test ediyoruz ve yüz doğrulama işleminde belli bir doğruluk performansı elde ediyoruz. Elde edilen doğruluk performansı Şekil 5.3'de gösterilmektedir. İkinci aşamada DEÇÜA ile çoğaltılmış veri seti üzerinde yüz tanıma sistemimizi test ediyoruz. Test

işlemi yapıldıktan sonra sistemimiz tarafından elde edilen yüz doğrulama performansı Şekil 5.4'de gösterilmektedir. Elde edilen sonuçlara baktığımızda, uygulanan veri artırma tekniklerinin sonuca olumlu yansıdığını ve doğruluk performansında kayda değer bir artış meydana geldiği açıktır.



Şekil 5. 3. Orijinal LFW veri setinde sistemin yüz doğrulama peformansı gösterilmektedir.



Şekil 5. 4. DEÇÜA ile çoğaltılmış veri setinde sistemin yüz doğrulama peformansı gösterilmektedir.

5.4. LFW veri setinde Yüz Sınıflandırma

Yüz sınıflandırma bölümünde çoğaltılmış veri setimizde yüz tanıma sistemimizin başarı performansını test ediyoruz. İlk aşamada veri setimizde bulunan görüntülerin

%90'lık kısmını rastgele (random) seçerek sınıflandırma algoritmasını eğitiyoruz. Kalan %10'luk kısmı test için kullanıyoruz. Test aşamasından sonra, sistemimizin yüz tanıma oranı oldukça yüksek olduğu ve görüntülerin büyük bir kısmı doğru şekilde tanıdığı gözlemlenmiştir. Sistemimiz tarafından doğru şekilde tanınan yüz görüntülerinin bir kısmı Şekil 5.5'de gösterilmektedir.



Şekil 5. 5. Yüz tanıma sisteminin doğru olarak tanıdığı kişiler.

Şekil 5.5'de gördüğümüz gibi gözlük, şapka veya değişik yüz ifadeleri gibi farklı koşullar altında bile sistemimiz bireyleri başarıyla tanımaktadır.

Aşırı farklı koşullar altında sistemimizin bireyleri tanımakta zorlandığı görülmüştür. Özellikle LFW veri setinde beyaz tenli kişi sayısı ile siyah tenli kişi sayısı arasında aşırı bir denge farkı vardır. Dolayısıyla ESA ağırlıklı olarak beyaz tenli kişiler ile eğitildiği için, sistemimiz siyah tenli kişileri tanımakta zorlanmıştır. Bununla birlikte aşırı farklı şapka, sıra dışı gözlük, saç şekli vs. gibi koşullar da performansı

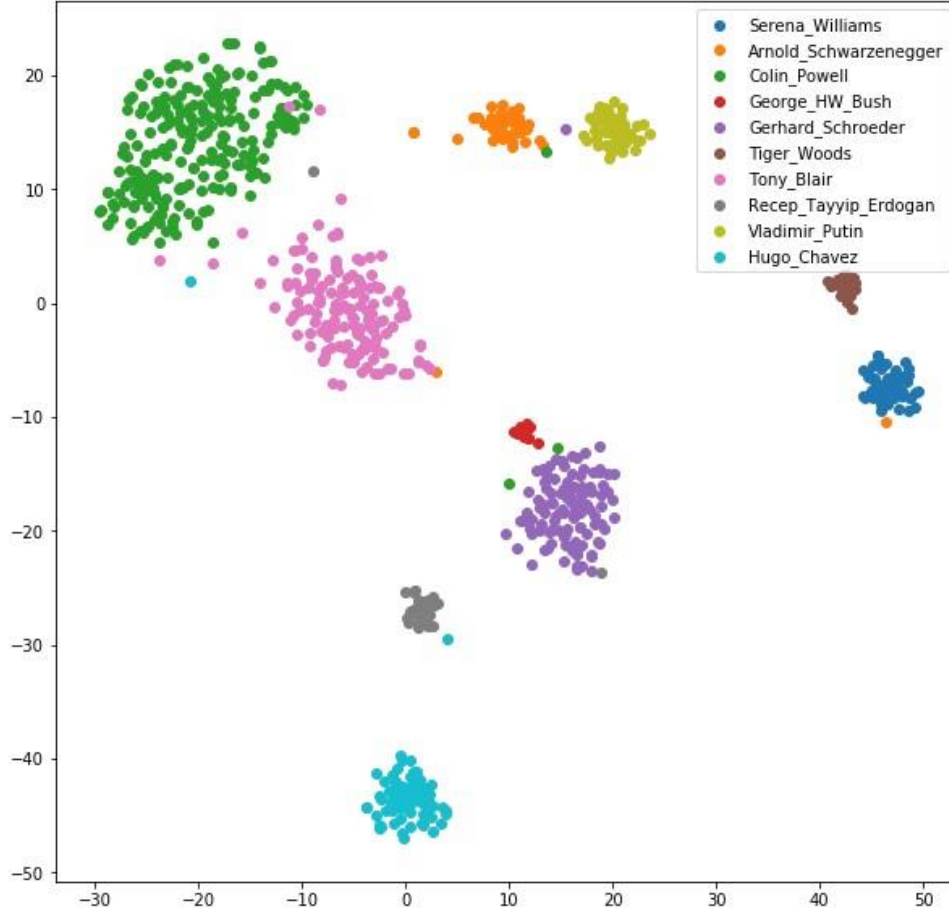
etkilemektedir. Şekil 5.6’da sistemimiz tarafından hatalı olarak tanınan görüntülerin bir kısmı gösterilmiştir.



Şekil 5. 6. Yüz tanıma sistemi tarafından hatalı olarak tanınan kişiler.

5.5. Yüz kümeleme

Yüz kümeleme işlemi ile birbirine benzeyen yüz görüntülerini kümeliyoruz. Böylece birbirine benzeyen yüz görüntülerini görsel üzerinden değerlendirme imkânı buluyoruz. Görselin daha belirgin ve anlamlı olması açısından veri setimizden rastgele olarak seçilmiş sadece 10 kişi için kümeleme görseli sunuyoruz. Her farklı renk veri setimizde bulunan bir bireyi, her nokta ise görüntü sayısını ifade eder. Şekil 5.7’de görüldüğü gibi, istisnalar hariç kimliksel kümeler doğru bir şekilde farklı bölümlere ayrılmıştır.



Şekil 5. 7. LFW veri setinde temsillerin görsel dağılımı.

5.6. Sonuç ve Tartışma

Bu kesite deney aşamalarından elde edilen sonuçları sunuyoruz. Elde edilen sonuçları önceki sonuçlar ve farklı algoritmalar ile karşılaştırmak için grafik (graph) veya tablo halinde gösteriyoruz.

5.6.1. Sistemin Doğruluk performansı

Yüz tanıma sistemimiz orijinal LFW veri setinde yüz doğrulama işlemi için %92.6 oranında doğruluk performansı elde etti. DEÇÜA ile çoğaltılmış veri setinde ise bu oran %94.6'ya çıkmıştır. Yüz sınıflandırma işleminde sistemimizi orijinal LFW veri seti üzerinde test ettiğimizde %92.99 oranında performans elde etti. Aynı işlem için DEÇÜA ile çoğaltılmış veri seti üzerinde sistemimizi test ettiğimizde bu oranın %95.25'e çıkmıştır. Dolayısıyla önerilen veri artırma tekniğinin yüz tanıma sisteminde doğrulama ve sınıflandırma performansına olumlu yansımıştır. Böylece her iki işlem için kayda değer artış elde edilmiştir.

5.6.2. Sonuçların karşılaştırılması

Tablo 5.1'de yüz doğrulama işleminde veri artırma teknikleri gerçekleştirilmeden önce ve gerçekleştirildikten sonra elde edilen sonuçlar karşılaştırılmıştır.

Tablo 5.1. Yüz doğrulama işleminde elde edilen sonuçların karşılaştırılması.

Nr.	Yöntem	Veri Seti	Doğruluk%
1	Önerilen Metod (NN4)	Orijinal LFW	92.60
2	Önerilen Metod (NN4)	DEÇÜA ile çoğaltılmış	94.60

Tablodaki sonuçlardan anlaşılacağı üzere veri artırma teknikleri ile sistemin yüz doğrulama performansında %2.00'lık kayda değer bir artış elde edildi.

Tablo 5.2'de yüz sınıflandırma işleminde veri artırma teknikleri gerçekleştirilmeden önce ve gerçekleştirildikten sonra elde edilen sonuçlar karşılaştırılmıştır.

Tablo 5.2. Yüz sınıflandırma işleminde elde edilen sonuçların karşılaştırılması.

Nr.	Yöntem	Veri Seti	Doğruluk%
1	Önerilen Metod (NN4)	Orijinal LFW	92.99
2	Önerilen Metod (NN4)	DEÇÜA ile çoğaltılmış	95.25

Tablo 5.2'de sonuçlarda görüldüğü gibi sistemin yüz sınıflandırma performansında önerilen veri artırma teknikleri sayesinde %2.26'lık bir artış elde edilmiştir.

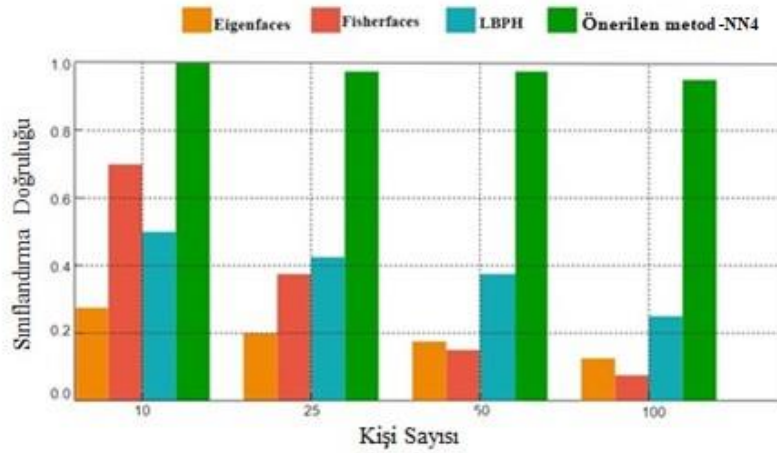
Bu çalışmada uygulanan yüz tanıma sistemi ve günümüzün mevcut en iyi (state-of-the-art) yüz tanıma sistemleri Tablo 5.1'de karşılaştırılmıştır. Çoğu sisteme göre daha az veri ile eğitilmesine rağmen ilgili sistem bahse konu sistemlerin doğruluk oranına yaklaştığı görülmüştür.

Genel anlamda sistemimizin doğruluk performansı en iyi yüz tanıma sistemlerinin performanslarına yaklaştığını söyleyebiliriz.

Tablo 5.2. Uygulanan yüz tanıma sisteminin diğer sistemler ile karşılaştırılması.

Nr.	Yöntem	Görüntü Sayısı	Doğruluk%
1	DeepFace (Taigman vd., 2014)	4M	97.35
2	OpenFace (Amos, Ludwiczuk ve Satyanarayanan, 2016)	0.5M	92.92
3	FaceDescriptor (Parkhi, Vedaldi ve Zisserman, 2015)	2.6M	98.95
4	FaceNet (Florian, Dmitry ve James, 2015)	200M	98.87
5	FaceNet + Alignment (Florian, Dmitry ve James, 2015)	200M	99.63
6	Önerilen Metod (NN4)	0.6M	95.25

Çalışmamızda uygulanan yüz tanıma sistemi ile geleneksel yöntemlere dayanan yüz tanıma sistemlerinin performansı Şekil 5.8’de karşılaştırılmıştır. Uygulanan DÖ dayalı yüz tanıma sistemi geleneksel yöntemlere dayalı yüz tanıma sistemlerine göre açık ara daha iyi performans sergilediği gözlemlenmiştir.



Şekil 5. 8. Çalışmada uygulanan yüz tanıma sistemi ile Geleneksel yöntemlere dayalı yüz tanıma sistemlerinin karşılaştırılması.

5.7 Zorluklar ve Sınırlamalar

Çalışmamızda karşılaştığımız en önemli zorluklardan birisi, gerçek örneklere benzeyecek şekilde sentetik yüz üretecek DEÇÜA modelini oluşturmaktı. Genel olarak, derin öğrenme modellerinde ayarlanmak üzere çok sayıda hiper-parametrelerin yanı sıra birçok model parametresi mevcuttur. Dolayısıyla bütün bu parametreler bir derin öğrenme modelinin performansını etkilemektedir. DEÇÜA iki derin öğrenme modeli içerir. Böylece zaten derin öğrenmede karmaşık olan eğitim aşaması daha da karmaşık bir hal alır. DEÇÜA modelleri, farklı veri setleri için farklı ayarlar gerektirebilirler (Goodfellow vd., 2014). Bu bahsedilen etkenler nedeniyle, derin öğrenme modeline ince ayar (fine tunning) yapma işlemi tahminimizden uzun sürdü. Karşılaştığımız diğer zorluklardan biri de, Tensorflow kütüphanesinin desteklediği güçlü bir ekran kartına sahip olmayışımızdır. Dolayısıyla DEÇÜA modelimizi işlemci (CPU) üzerinden eğitmek zorunda kaldık. Ve bu da eğitimin haftalarca sürmesine neden oldu. Orta seviye hesaplama gücüne sahip bir Nvidia ekran kartıya DEÇÜA modelimizin eğitim süreci 3 gün sürmektedir. Çalışmamızda DEÇÜA eğitim süreci ise 23 gün sürdü.

BÖLÜM 6

SONUÇ VE GELECEKTEKİ ÇALIŞMALAR

6.1. Sonuç

Çalışmamızda, DEÇÜA ve AD veri artırma tekniklerinin yüz tanıma sistemlerinin doğruluk performansında etkisi gösterilmeye çalışılmıştır. Çalışma boyunca edindiğimiz tecrübe, DEÇÜA'nın temel gerçeğe benzer sentetik yüzler üretme kabiliyetine sahip olduğudur. Ancak bunun gerçekleşmesi, veri seti genişliği, kişiye ait görüntü sayısında denge ve ekran kartı performansı gibi etkenlere bağlıdır.

Doğruluk performansı söz konusu olduğunda, yüz tanıma sistemimizin performansında artış tespit edilmiştir. Ancak hedeflediğimiz oranda bir artış elde edemedik. Bunun temel nedeni DEÇÜA modeli ile üretilen sentetik yüzlerin bir bölümünün yüz tanıma sistemi tarafından doğru şekilde algılanabilecek düzeyde olmamasından kaynaklanmaktadır. Hedeflenen oranda bir artışın elde edilememesine rağmen DEÇÜA ve AD veri artırma tekniklerinin sistemin doğruluk performansına olumlu yansımaları, pozitif gelişme olarak algılanmış ve ileride yapılacak ince ayar işlemleriyle bu oranın artabileceği kanaati oluşmuştur.

Genel anlamda DEÇÜA'da model oluşturma ve eğitim aşamalarında zorluklarla karşılaşmıştır. Temel gerçeğe yakın örnek üretecek modeli oluştururken orijinal ve diğer çalışmalardaki tavsiyeler göz önünde bulundurulmalıdır. Aksi takdirde sonuçların kötü olması kaçınılmazdır. DEÇÜA'da hem üretici ağının hem de ayırmacı ağının aynı anda eğitilme zorunluluğunun bulunması bu ağlarda eğitim sürecini zorlaştırmaktadır. Eğitim süreci, üretici ağ ve ayırmacı ağ arasında dengeli değilse problemler meydana gelecektir. Ayırmacı ağ, sentetik verileri kesin doğrulukla sınıflandırır, üretici ağ hızlandırabilir. Aynı şekilde üretici ağ, ayırmacı ağında bir takım zaafı tespit ederse performansını düşürebilme ihtimali vardır. DEÇÜA iki derin öğrenme modeli içerdiğinden eğitim sürecinde yüksek miktarda hesaplama işlemi gerçekleşmektedir. Dolayısıyla hesaplama gücü yüksek ekran kartıyla bile eğitim süreci çok zaman alacaktır. Çalışmamızda yüksek hesaplama gücüne sahip ekran kartına sahip olmadığımızdan dolayı eğitim süreci tahminimizden çok daha uzun sürmüştür. Bu da DEÇÜA modelimize ince ayar yapma imkânını kısıtlamıştır.

6.2. Gelecekteki Çalışmalar

DEÇÜA modeli tasarlanmış, uygulanmış ve yüz tanıma sisteminin performansında etkisi test edilmiştir. Test sonuçları, bu veri artırma tekniğinin sistemin performansına olumlu yansıdığını göstermektedir. Hem elde edilen sonuçları iyileştirmek için hem de ÇÜA'ın farklı modellerini incelemek için gelecekte bazı çalışmaların yapılması öngörülmüştür. DEÇUA modelimizin daha geniş veri seti ile eğitildiği takdirde üreteceği sentetik yüz görüntülerinin kalitesinde iyileşme meydana geleceğini tahmin ediyoruz. Dolayısıyla ilk gerçekleştireceğimiz işlem, modelimize ince ayar yapmak olacaktır.

Gelecekte yapacağımız diğer bir çalışma ÇÜA'ın farklı modellerini incelemek olacaktır. ÇÜA'ın gelişimi halen devam etmektedir, bununla birlikte birçok ilgi çekici model tasarlanmaktadır. Araştırmacılar tarafından oldukça ilgi gören bir diğer model de Çift Yönlü Çekişmeli Üretici Ağlar (Bidirectional Generative Adversarial Networks- BiGAN) dır. Modeli tasarlayan araştırmacılar bilimsel çalışmalarında dengeli bir eğitim ve diğer ağlara göre daha kaliteli sentetik görüntü üretimi vaad ediyorlar (Donahue, Krähenbühl ve Darrell, 2016). ÇÜA'ın farklı modellerini incelerken aynı zamanda yapacağımız bir başka bir işlem de Çift Yönlü Çekismeli Üretici Ağlar ve Derin Evrişimsel Çekismeli Üretici Ağların başarı performansını karşılaştırmak olacaktır.

Referanslar

AbdAlmageed, W., Wu, Y., Rawls, S., Harel, S., Hassner, T., Masi, I., ... & Nevatia, R. (2016, March). Face recognition using deep multi-pose representations. In *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)* (pp. 1-9). IEEE.

Adini, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7), 721-732.

Agarwal, A., Singh, R., Vatsa, M., & Ratha, N. (2018, October). Are image-agnostic universal adversarial perturbations for face recognition difficult to detect?. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)* (pp. 1-7). IEEE.

Ahonen, T., Hadid, A., & Pietikainen, M. (2006). Face description with local binary patterns: Application to face recognition. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12), 2037-2041.

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017, October). Boosting cross-age face verification via generative age normalization. In *2017 IEEE International Joint Conference on Biometrics (IJCB)* (pp. 191-199). IEEE.

Antipov, G., Baccouche, M., & Dugelay, J. L. (2017, September). Face aging with conditional generative adversarial networks. In *2017 IEEE International Conference on Image Processing (ICIP)* (pp. 2089-2093). IEEE.

Atienza, 2017 - <https://towardsdatascience.com/gan-by-example-using-keras-on-tensorflow-backend-1a6d515a60d0>

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (7), 711-720.

Brunelli, R., & Poggio, T. (1992, May). Face recognition through geometrical features. In *European Conference on Computer Vision* (pp. 792-800). Springer, Berlin, Heidelberg.

Castelluccio, M., Poggi, G., Sansone, C., & Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. *arXiv preprint arXiv:1508.00092*.

Chan, Y. J., & Ewins, D. J. (2011). The amplification of vibration response levels of mistuned bladed disks: its consequences and its distribution in specific situations. *Journal of Engineering for Gas Turbines and Power*, 133(10), 102502.

Chavdarova, T., & Fleuret, F. (2018). Sgan: An alternative training of generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 9407-9415).

- Chen, X., Duan, Y., Houthoofd, R., Schulman, J., Sutskever, I., & Abbeel, P. (2016). Infogan: Interpretable representation learning by information maximizing generative adversarial nets. In *Advances in neural information processing systems* (pp. 2172-2180).
- Chen, Zhimin, and Yuguang Tong. "Face super-resolution through wasserstein gans." arXiv preprint arXiv:1705.02438 (2017).
- Cox, I. J., Ghosn, J., & Yianilos, P. N. (1996, June). Feature-based face recognition using mixture-distance. In *cvpr* (Vol. 96, pp. 209-216).
- Curtis, F. E., & Que, X. (2015). A quasi-Newton algorithm for nonconvex, nonsmooth optimization with global convergence guarantees. *Mathematical Programming Computation*, 7(4), 399-428.
- Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection.
- Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9), 1627-1645.
- Geitgey, 2017 - <https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems* (pp. 2672-2680).
- Haykin S., (2009). *Neural Networks and Learning Machines* (3rd ed.), Pearson. Upper Saddle River, NJ, USA.
- Hines, J., Tsoukalas, L. H., & Uhrig, R. E. (1997). *MATLAB supplement to fuzzy and neural approaches in engineering*. John Wiley & Sons, Inc..
- Hochreiter, S., Younger, A. S., & Conwell, P. R. (2001, August). Learning to learn using gradient descent. In *International Conference on Artificial Neural Networks* (pp. 87-94). Springer, Berlin, Heidelberg.
- Hong W., S. Winkler. (2014). Proc. IEEE International Conference on Image Processing (ICIP), Paris, France, Oct. 27-30
- Hopfield, J. J., & Tank, D. W. (1985). "Neural" computation of decisions in optimization problems. *Biological cybernetics*, 52(3), 141-152.
- Howard, A. G. (2013). Some improvements on deep convolutional neural network based image classification. *arXiv preprint arXiv:1312.5402*.
- Huang, G. B., Mattar, M., Berg, T., & Learned-Miller, E. (2008, October). Labeled faces in the wild: A database for studying face recognition in unconstrained environments.
- Huang, R. J. J. (1998). *Detection strategies for face recognition using learning and evolution*. George Mason University.
- Ian G., Yoshua B., Aaron C. (2016). *Deep Learning* (1st ed.) MIT Press.

- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jafri, R., & Arabnia, H. R. (2009). A survey of face recognition techniques. *Jips*, 5(2), 41-68.
- Jiang, D., Hu, Y., Yan, S., Zhang, L., Zhang, H., & Gao, W. (2005). Efficient 3D reconstruction for face recognition. *Pattern Recognition*, 38(6), 787-798.
- Kanade, T. (1974). Picture processing system by computer complex and recognition of human faces.
- Kanede T. (1973). Picture processing system by computer complex and recognition of human faces, Doctoral dissertation Kyoto University, 83–97.
- Karras, T., Aila, T., Laine, S., & Lehtinen, J. (2017). Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.
- Kriesel, (2009) - http://www.dkriesel.com/en/science/neural_networks
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (pp. 1097-1105).
- Lawrence, S., Giles, C. L., Tsoi, A. C., & Back, A. D. (1997). Face recognition: A convolutional neural-network approach. *IEEE transactions on neural networks*, 8(1), 98-113.
- Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., ... & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4681-4690).
- Liu, Z., Luo, P., Wang, X., & Tang, X. (2015). Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 3730-3738).
- Lv, J. J., Shao, X. H., Huang, J. S., Zhou, X. D., & Zhou, X. (2017). Data augmentation for face recognition. *Neurocomputing*, 230, 184-196.
- Mathieu, M., Couprie, C., & LeCun, Y. (2015). Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*.
- Melin, P., Felix, C., & Castillo, O. (2005). Face recognition using modular neural networks and the fuzzy Sugeno integral for response integration. *International Journal Of Intelligent Systems*, 20(2), 275-291.
- Mirza, M., & Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.
- Mohammadzade, H., & Hatzinakos, D. (2012). Projection into expression subspaces for face recognition from single sample per person. *IEEE Transactions on Affective Computing*, 4(1), 69-82.
- Nag,(2017) - <https://medium.com/@devnag/generative-adversarial-networks-gans-in-50-lines-of-code-pytorch-e81b79659e3f>

Parkhi et al., 2015] Parkhi, O. M., Vedaldi, A., and Zisserman, A. (2015). Deep face recognition. In British Machine Vision Conference.

Quesada,(2019)

https://www.neuraldesigner.com/blog/5_algorithms_to_train_a_neural_network

Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*.

Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., & Lee, H. (2016). Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*.

Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6), 386.

Sagiroglu, S., Colak, I., & Bayindir, R. (2006). Power factor correction technique based on artificial neural networks. *Energy Conversion and Management*, 47(18-19), 3204-3215.

Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

Seyyedsalehi, S. Z., & Seyyedsalehi, S. A. (2014). Simultaneous learning of nonlinear manifolds based on the bottleneck neural network. *Neural processing letters*, 40(2), 191-209.

Sharma, B., & Venugopalan, K. (2014). Comparison of neural network training functions for hematoma classification in brain CT images. *IOSR Journal of Computer Engineering*, 16(1), 35.

Shashua, A., Gdalyahu, Y., & Hayun, G. (2004, June). Pedestrian detection for driving assistance systems: Single-frame classification and system level performance. In *IEEE Intelligent Vehicles Symposium, 2004* (pp. 1-6). IEEE.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Sirovich, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3), 519-524.

Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2014). Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).

Taigman, Y., Yang, M., Ranzato, M. A., & Wolf, L. (2014). Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1701-1708).

Turk, M. A., & Pentland, A. P. (1991, June). Face recognition using eigenfaces. In *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (pp. 586-591). IEEE.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86.

Viola, P., & Jones, M. (2001). Robust real-time object detection. *International journal of computer vision*, 4(34-47), 4.

Xu, Y., Li, X., Yang, J., & Zhang, D. (2014). Integrate the original face image and its mirror image for face recognition. *Neurocomputing*, 131, 191-199.

Yi, D., Lei, Z., Liao, S., & Li, S. Z. (2014). Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*.

Zeiler, M. D., & Fergus, R. (2014, September). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818-833). Springer, Cham.

ÖZGEÇMİŞ

Erdal ALİMOVSKİ İstanbul Sabahattin Zaim Üniversitesi Bilgisayar Bilimi ve Mühendisliği bölümünde yüksek lisans öğrencisidir. Üniversitenin bünyesinde bulunan “Gömüllü sistemler” laboratuvarında çalışmalarına ağırlıklı olarak Görüntü İşleme üzerinde devam etmektedir. Mevcut araştırma alanları arasında makine öğrenmesi, derin öğrenme, yapay sinir ağları bulunmaktadır.

