



**T.C.**  
**KONYA TEKNİK ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**DERİN ÖĞRENME TEKNİKLERİ**  
**KULLANILARAK YÜZ TANIMA TABANLI**  
**MÜŞTERİ DOĞRULAMA İLE**  
**BANKAMATİKLERDE SAHTEKÂRLIK**  
**TESPİTİ**

**Mehmet YILDIRIM**

**YÜKSEK LİSANS**

**Bilgisayar Mühendisliği Anabilim Dalı**

**Temmuz-2019**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Mehmet YILDIRIM tarafından hazırlanan “Derin Öğrenme Teknikleri Kullanılarak Yüz Tanıma Tabanlı Müşteri Doğrulama İle Bankamatiklerde Sahtekârlık Tespiti” adlı tez çalışması 19/07/2019 tarihinde aşağıdaki jüri tarafından oy birliği ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

**Başkan**  
Doç. Dr. Halife KODAZ

**Danışman**  
Dr. Öğr. Üyesi Sait Ali UYMAZ

**Üye**  
Dr. Öğr. Üyesi Burak YILMAZ

### İmza







Yukarıdaki sonucu onaylarım.

Prof. Dr. Hakan KARABÖRK  
Enstitü Müdürü

## **TEZ BİLDİRİMİ**

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

## **DECLARATION PAGE**

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



Mehmet YILDIRIM  
Tarih: 19.07.2019

## ÖZET

### YÜKSEK LİSANS

# DERİN ÖĞRENME TEKNİKLERİ KULLANILARAK YÜZ TANIMA TABANLI MÜŞTERİ DOĞRULAMA İLE BANKAMATİKLERDE SAHTEKÂRLIK TESPİTİ

**Mehmet YILDIRIM**

**Konya Teknik Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Bilgisayar Mühendisliği Anabilim Dalı**

**Danışman: Dr. Öğr. Üyesi Sait Ali UYMAZ**

**2019, 66 Sayfa**

**Jüri**

**Dr. Öğr. Üyesi Sait Ali UYMAZ  
Doç. Dr. Halife KODAZ  
Dr. Öğr. Üyesi Burak YILMAZ**

Günümüzde bankamatikler banka müşterilerinin finansal işlerini yerine getirmeleri için çok yaygın olarak kullanılmaktadır. Bankamatikler manyetik bir kart ve dört haneli bir PIN ile müşteri doğrulama sağladıktan sonra kullanıma açılmaktadır. Bu durum, sahtekârlara bankamatik donanımlarını kullanarak para çalmak için fırsat oluşturmaktadır. Dijital bankacılık platformlarının sayısı arttıkça banka müşterileri banka şubelerini fiziki olarak ziyaret etmek zorunda kalmadan hesap açabilir hale gelmiştir. Benzer şekilde banka kartları müşterilere kurye aracılığı ile şubeye gelmeye gerek kalmadan ulaştırılabilmektedir. Bu durum, kötü niyetli kurye çalışanlarına kart bilgilerini çalmak için fırsat vermektedir. Buna ek olarak banka müşterileri genellikle dört haneli PIN kodunu; doğum tarihi, yaşadıkları şehirlerin plakaları ve ulusal kimlik numaraları gibi hatırlanması kolay şifrelerden seçmektedir. Sahtekârlar bu kodu tahmin edebilmekte veya kendilerini banka çalışanı gibi tanıtırak bu kodları kolayca elde edebilmektedir. Banka kartı ve bankamatiklerdeki buna benzer dezavantajlar nedeniyle dolandırıcılar banka müşterilerinden kolayca para çalma potansiyeline sahiptir.

Bu çalışmada, bankamatiklerde ilgili sahtekârlıkların önüne geçmek için bankaların standart şifre esaslı kimlik doğrulama yöntemine ek olarak yüz tanıma teknolojilerini kullanarak bir kimlik doğrulama yöntemi ortaya konulmuştur. Müşteri bankada hesap açarken müşteriden alınan yüz örnekleri derin öğrenme modelinden geçirilerek benzersiz bir kimlik üretilip kaydedilecektir. Daha sonra müşteri bankamatikte işlem yaparken tekrar alınan yüzler derin modelden geçirilip çıkan sonuç, kaydedilen kimlik ile karşılaştırılacaktır. Bu karşılaştırma sonucu, belirlenen eşik değerinin altında kalırsa müşteri doğrulaması sağlanacaktır. Aksi takdirde, kart ve PIN bilgisiyle dahi bankamatiğe giriş sağlanamayacaktır. Böylece bankaların bankamatik aracılığıyla verdikleri hizmetlerde çok daha güvenli bir platform sunmaları sağlanacaktır.

**Anahtar Kelimeler:** Banka, Bankamatik, Derin Öğrenme, Yüz Tanıma, Yüz Doğrulama, Sahtekârlık

## **ABSTRACT**

### **MS THESIS**

# **FACE RECOGNITION BASED CUSTOMER AUTHENTICATION BY USING DEEP LEARNING TECHNIQUES FOR DETECTING ATM FRAUD**

**Mehmet YILDIRIM**

**Konya Technical University  
Institute of Graduate Studies  
Department of Computer Engineering**

**Advisor: Asst. Prof. Dr. Sait Ali UYMAZ**

**2019, 66 Pages**

**Jury**

**Asst. Prof. Dr. Sait Ali UYMAZ  
Assoc. Prof. Dr. Halife KODAZ  
Asst. Prof. Dr. Burak YILMAZ**

Today, cash dispensers are widely used for bank customers to perform their financial business. The cash dispensers are put into use after providing customer verification with a magnetic card and a four-digit PIN. This provides an opportunity for fraudsters to steal money using the hardware of cash dispensers. As the number of digital banking platforms increased, bank customers have been able to open accounts without having to physically visit the bank branches. Similarly, bank cards can be delivered to customers by courier without having to visit the branch. This gives evil-minded courier employees the opportunity to steal card information. In addition, bank customers typically set their four-digit PIN code as the date of birth, the license plates of the cities they live in, or their national identification numbers. Fraudsters can guess this code or identify themselves as bank employees and can easily obtain them. Due to such disadvantages in credit and debit cards, fraudsters have the potential to easily steal money from bank customers.

In this work, in order to prevent related fraud in ATMs, an authentication method by using face recognition technologies in addition to the standard password based authentication method of banks has been introduced. When the customer opens an account at the bank, the face samples taken from the customer will be passed through a deep learning model and a unique identity will be produced and recorded. Afterwards, when the customer is processing at the cash dispenser, the faces taken again will be passed to the deep model and the result will be compared with the recorded identity. When this comparison result is below the threshold value, customer verification will be approved. Otherwise, access to the cash dispenser will not be possible even with card and PIN information. Thus, it will be ensured that banks provide a much safer platform in the services they provide through ATMs.

**Keywords:** Bank, ATM, Deep Learning, Face Recognition, Face Verification, Fraud

## ÖNSÖZ

Bu çalışma süresince ve yüksek lisans öğrenimim boyunca, bilgisini, desteğini, ilgi ve alakasını benden esirgemeyen saygıdeğer danışman hocam Sayın Dr. Öğr. Üyesi Sait Ali UYMAZ'a; ders dönemim boyunca desteklerini esirgemeyen, Konya Teknik Üniversitesi Mühendislik ve Doğa Bilimleri Fakültesi Bölümü'ndeki saygıdeğer hocalarıma teşekkürlerimi sunarım.

Birçok konuda olduğu gibi yüksek lisans sürecinde de beni destekleyip motive eden sevgili eşim Leyla YILDIRIM'a ve biricik kızım Eylül YILDIRIM'a sevgilerimi ve teşekkürlerimi sunarım.

Mehmet YILDIRIM  
KONYA-2019

## İÇİNDEKİLER

ÖZET .....	iv
ABSTRACT .....	v
ÖNSÖZ .....	vi
İÇİNDEKİLER .....	vii
ŞEKİLLER LİSTESİ .....	ix
ÇİZELGELER LİSTESİ .....	xi
SİMGELER VE KISALTMALAR.....	xii
<b>1. GİRİŞ.....</b>	<b>1</b>
<b>2. KAYNAK ARAŞTIRMASI .....</b>	<b>3</b>
<b>3. MATERYAL VE YÖNTEM.....</b>	<b>6</b>
3.1. Yapay Sinir Ağları .....	6
3.1.1. Biyolojik sinir ağları .....	6
3.1.2. Yapay sinir hücresinin yapısı .....	7
3.1.3. Aktivasyon fonksiyonları .....	8
3.1.4. Tek katmanlı yapay sinir ağları .....	12
3.1.5. Çok katmanlı yapay sinir ağları .....	13
3.1.6. Geri beslemeli yapay sinir ağları .....	13
3.1.7. Maliyet fonksiyonları .....	14
3.1.8. Gradyan iniş algoritması (Gradient Descent) .....	15
3.1.9. Geri yayılım algoritması.....	18
3.2. Derin Öğrenme .....	21
3.2.1. Evrimsel sinir ağları .....	21
3.2.2. Derin öğrenme modelleri.....	28
3.3. Derin Yüz Tanıma .....	31
3.3.1. Yüz tespiti.....	33
3.3.2. Yüz hizalama .....	34
3.3.3. Özellik çıkarımı .....	36
3.3.4. GoogLeNet (Inception V1) modeli.....	40
<b>4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA .....</b>	<b>43</b>
4.1. Yüz Servisi .....	44
4.2. Bankacılık Uygulaması .....	46
4.3. Bankamatik Uygulaması .....	47
4.4. Kullanılan Modelin Değerlendirilmesi .....	51
4.5. Yüz Servisi Testleri .....	56

<b>5. SONUÇLAR VE ÖNERİLER .....</b>	<b>61</b>
5.1. Sonuçlar .....	61
5.2. Öneriler .....	62
<b>KAYNAKLAR .....</b>	<b>63</b>
<b>ÖZGEÇMİŞ.....</b>	<b>66</b>





## ŞEKİLLER LİSTESİ

Şekil 3.1. Biyolojik nöral sistemin üç aşaması (Kawaguchi, 2001) .....	6
Şekil 3.2. Biyolojik bir sinir hücresi (Kriesel, 2007) .....	7
Şekil 3.3. YSA örneği (Keskenler ve Keskenler, 2017).....	8
Şekil 3.4. Lineer aktivasyon fonksiyonu (Gadoue, 2007).....	9
Şekil 3.5. Eşik aktivasyon fonksiyonu (Nordbotten, 2006).....	9
Şekil 3.6. Sigmoid aktivasyon fonksiyonu (Maladkar, 2018) .....	10
Şekil 3.7. Tanh aktivasyon fonksiyonu (Maladkar, 2018) .....	11
Şekil 3.8. ReLU aktivasyon fonksiyonu (Maladkar, 2018).....	11
Şekil 3.9. ReLU ve Sızıntılı ReLU aktivasyon fonksiyonları (Maladkar, 2018).....	12
Şekil 3.10. Tek katmanlı yapay sinir ağı (Yiğit, 2017).....	12
Şekil 3.11. Çok katmanlı yapay sinir ağı (Patterson ve Gibson, 2017) .....	13
Şekil 3.12. Geri beslemeli yapay sinir ağı (Yiğit, 2017).....	14
Şekil 3.13. Yapay nöron (Şahin, 2017) .....	15
Şekil 3.14. Ağırlık ve kayıp değerine göre global minimum eğrisi (Patterson ve Gibson, 2017).....	16
Şekil 3.15. Bir nörona aktivasyon ve geri yayılım uygulanmasının formülle gösterilmesi (Kim, 2017).....	16
Şekil 3.16. Çok katmanlı sinir ağı örneği (Patterson ve Gibson, 2017).....	18
Şekil 3.17. Çıkış katmanındaki ağırlıkların güncellenmesi (Patterson ve Gibson, 2017).....	19
Şekil 3.18. Gizli katmanlar arası bağlantıların gösterimi (Patterson ve Gibson, 2017).....	21
Şekil 3.19. Örnek LeNet-5 ağının mimarisi (Gu ve ark., 2018).....	22
Şekil 3.20. Evrişim sinir ağı giriş katmanı (Andrej Karpathy ve Johnson, 2019).....	23
Şekil 3.21. 5x5x3 boyutta bir giriş görüntüsüne 3x3'lük filtrenin uygulandığı konvolüsyon işlemi (Özkan ve Ülker, 2017).....	24
Şekil 3.22. Evrişim ağlarına genel bakış (Afonja, 2018) .....	25
Şekil 3.23. ReLU aktivasyon fonksiyonu (Sharma ve ark., 2016) .....	26
Şekil 3.24. Bir ESA modelinde evrişim katmanı ve ReLU katmanının giriş görüntüsüne yapmış olduğu etki (Özkan ve Ülker, 2017).....	26
Şekil 3.25. 4x4'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı maksimum havuzlama işleminin uygulanması (Özkan ve Ülker, 2017).....	27
Şekil 3.26. ESA modelinde havuzlama katmanının bir önceki katmandan gelen görüntüye uygulanması sonucu oluşan görüntü (Özkan ve Ülker, 2017).....	27
Şekil 3.27. Alex Net modelinin yapısı (Krizhevsky ve ark., 2012).....	29
Şekil 3.28. ZF Net modelinin yapısı (Zeiler ve Fergus, 2014).....	29
Şekil 3.29. VGG16 ve VGG19 mimarisi (Rosebrock, 2017).....	30
Şekil 3.30. Microsoft ResNet mimarinin yapısı (He ve ark., 2016) .....	31
Şekil 3.31. Residual bloğu (He ve ark., 2016).....	31
Şekil 3.32. Yüz sınıflandırma ve doğrulama örnekleri .....	32
Şekil 3.33. Yüz tanıma işlem adımları .....	33
Şekil 3.34. Çoklu evrişim ağının ilk aşamasına verilmek için oluşturulan görüntü piramidi (Zhang ve ark., 2018) .....	34
Şekil 3.35. Teklif (P-Net) evrişim ağı (Zhang ve ark., 2016).....	34
Şekil 3.36. Teklif Ağı (P-Net) sonucunda ortaya çıkan yeni resim (Zhang ve ark., 2016).....	35
Şekil 3.37. Arıtım (R-Net) evrişim ağı (Zhang ve ark., 2016) .....	35
Şekil 3.38. Arıtım ağı (R-Net) sonucunda ortaya çıkan yeni resim (Zhang ve ark., 2016).....	35

Şekil 3.39. Çıkış (O-Net) evrişim ağı (Zhang ve ark., 2016) .....	36
Şekil 3.40. Çıkış ağı (O-Net) ile yüzün beş temel noktasının tespiti (Zhang ve ark., 2016).....	36
Şekil 3.41. DeepFace mimarisi (Taigman ve ark., 2014).....	37
Şekil 3.42. FaceNet mimarisi (Schroff ve ark., 2015) .....	39
Şekil 3.43. Üçlü kayıp örneği (Schroff ve ark., 2015) .....	39
Şekil 3.44. GoogLeNet mimarinin yapısı (Deshpande, 2019) .....	41
Şekil 3.45. Bir Inception detayı (Szegedy ve ark., 2015).....	41
Şekil 3.46. Bir Inception V1 detayı (Szegedy ve ark., 2015).....	42
Şekil 4.1. Derin öğrenme kullanılarak sahtekârlık tespiti uygulama mimarisi .....	43
Şekil 4.2. Yüz servisi yüz kaydetme işlemi.....	44
Şekil 4.3. Yüz servisi yüz öznitelik çıkarma istek örneği .....	45
Şekil 4.4. Yüz servisi yüz öznitelik çıkarma cevap örneği .....	45
Şekil 4.5. Yüz servisi yüz doğrulama adımları.....	45
Şekil 4.6. Yüz doğrulama metodunun örnek bir isteği.....	46
Şekil 4.7. Yüz doğrulama metodunun örnek bir dönüşü.....	46
Şekil 4.8. Müşteri bilgileri ve yüz görüntülerinin alındığı ekran.....	47
Şekil 4.9. Bir resmin derin modelden geçirildikten sonra elde edilen öznitelik vektörü	47
Şekil 4.10. Veri tabanı yapısı.....	47
Şekil 4.11. Manyetik kart simülasyon ekranı ve doğrulamayı başlatan ekran .....	48
Şekil 4.12. Şifre doğrulama ekranı.....	48
Şekil 4.13. Yüz örneklerinin alındığı ekran.....	49
Şekil 4.14. Yüz doğrulamanın yapıldığı bekleme ekranı .....	49
Şekil 4.15. Yüz doğrulaması sonucunda açılan işlem ekranı .....	50
Şekil 4.16. Hesap sahibinden farklı bir kişinin yüz doğrulama adımını geçme denemesi .....	50
Şekil 4.17. Hesap kitlenme ekranı .....	51
Şekil 4.18. Yüz tanıma sistemlerine ağ mimarisinin etkisini göstermek için kullanılan mimariler (Simonyan ve Zisserman, 2014) .....	52
Şekil 4.19. LWF veri setinin eğitilmiş model üzerindeki değerlendirme grafiği (Sandberg, 2018).....	55
Şekil 4.20. LWF veri setinin ve eğitim aşamasındaki modelin çapraz entropi kaybının gösterimi (Sandberg, 2018).....	55
Şekil 4.21. LWF veri setinin değerlendirilmesi ve eğitim veri seti ile karşılaştırılması (Sandberg, 2018).....	56
Şekil 4.22. LWF veri setinde bulunan resimler arasındaki uzaklık grafiğine bağlı olarak eşik değerinin gösterimi.....	57
Şekil 4.23. LWF veri setinden alınan bir kişinin resimleri .....	57
Şekil 4.24. LWF test verisinden alınan bir kişinin yüz hizalanması (ilk sıradaki resimler orijinal resimler ikinci sıradaki resimler hizalanan yüz resimleridir). .....	58
Şekil 4.25. LWF test verisinden alınan bir kişinin başarı grafiği .....	58
Şekil 4.26. LWF veri setinden test yapılan kişinin doğru tespit edilen resimleri .....	59
Şekil 4.27. LWF veri setinden test yapılan kişinin yanlış tespit edilen resimleri.....	59
Şekil 4.28. Bir katılım bankasının şubesinden ve laboratuvarından alınan resimlerin hizalanması ve birbirine olan uzaklıklarının gösterilmesi .....	60

## ÇİZELGELER LİSTESİ

<b>Çizelge 3.1.</b> Geri yayılım algoritması için kullanılan terimler (Patterson ve Gibson, 2017).....	19
<b>Çizelge 3.2.</b> LFW veri seti üzerinde DeepFace ve FaceNet mimarilerinin karşılaştırılması (Parkhi ve ark., 2015).....	37
<b>Çizelge 4.1.</b> Ağ mimarisi, veri seti, üçlü kayıp, yüz hizalamanın yüz tanıma başarısına etkisi (Parkhi ve ark., 2015).....	52
<b>Çizelge 4.2.</b> Kamuoyuna açık olan yüz veri setleri (Parkhi ve ark., 2015).....	53
<b>Çizelge 4.3.</b> Kamuoyuna açık olmayan yüz veri setleri (Parkhi ve ark., 2015) .....	53



## SİMGELER VE KISALTMALAR

### Kısaltmalar

<b>CPU</b>	: Central Processing Unit (Merkezi İşlem Birimi)
<b>ESA</b>	: Evrimsel Sinir Ağları
<b>GPU</b>	: Graphic Processing Unit (Grafik İşlem Birimi)
<b>ILSVRC</b>	: ImageNet Large-Scale Visual Recognition Challenge (ImageNet Büyük Ölçekli Görüntü Tanıma Yarışması)
<b>ReLU</b>	: Rectified Linear Units Layer (Düzleştirilmiş Doğrusal Birim Katmanı)
<b>YSA</b>	: Yapay Sinir Ağları
<b>DVM</b>	: Destek Vektör Makinaları



## 1. GİRİŞ

Bankamatik cihazlarının hayatımıza girmesi ile birlikte bankacılık alanında birçok kolaylık sağlamıştır. Banka şubelerinde oluşan uzun kuyruklar ve günlerce süren para transferi işlemleri bankamatik'in keşfedilmesi ile büyük ölçüde son bulmuştur. Aynı zamanda bankacıların çalışma şartlarını zorlayan etkenleri ve operasyonel hataların meydana gelmesini de büyük ölçüde engellemiştir. Banka müşterileri için bankacılık işleri ev ve iş yerlerine yakın hale gelmiştir. 7/24 bankacılık işlemlerinin devam etmesini sağlamıştır. Ancak bu avantajlar ile beraber bazı dezavantajları da beraberinde getirmiştir (Awodele ve Akanni, 2012).

Medyada bankamatik dolandırıcılığı ile ilgili birçok haber yer almaktadır. Ticari bir banka müşterilerin güvenliğini sağlamakla sorumludur (Kale ve Nanda, 2014). Bankamatikler genellikle iki faktörlü doğrulama ile çalışmaktadır. Bunlar müşteriye özgü olan manyetik kartın bankamatik yuvasına yerleştirilmesi ve dört haneli bir PIN kodunun girilmesidir (Aru ve Gozie, 2013). Günümüzde birçok insan dört haneli PIN'in akılda kalması için doğum tarihi, vatandaşlık numarasının ilk dört hanesi gibi kolay tahmin edilebilir PIN'ler seçmektedir. Kart ise erişilebilir olduğu için PIN sosyal mühendislik, baskı gibi yöntemler ile elde edilebilir.

Bu çalışma kapsamında son zamanlarda çok popüler olan, neredeyse bir uzman kadar doğru karar verebilen derin öğrenme yöntemleri kullanılarak ikinci bir kimlik doğrulama yöntemi sunulmaktadır. Bankamatiklerde bulunan manyetik kart doğrulama ve dört haneli PIN doğrulamadan sonra bankamatik cihazında bulunan kamera yardımıyla müşteri yüz resminin alınacağı bir doğrulama yöntemi kullanılmıştır. Bu doğrulama yöntemi ile müşteri banka ile ilk iletişime geçtiği zaman, müşterinin diğer bilgileri ile birlikte yüz örnekleri alınacaktır ve banka veri tabanına kaydedilecektir. Müşteri her bankamatik kimlik doğrulaması yaptığında manyetik kart ve PIN doğrulamasından sonra açılan kamera aracılığıyla yüz örnekleri alınıp, bu alınan yüz örnekleri daha önce veri tabanında kayıtlı olan yüz örneğiyle karşılaştırılacaktır. Bu karşılaştırma sonucunda doğrulama sağlanırsa müşteri işlem yapabilecektir. Aksi takdirde doğrulama sağlanmadığı için işlem yapamayacaktır. Böylelikle müşteri bankamatik kanalıyla daha güvenli bankacılık işlemi yapacaktır.

Bu çalışma kapsamında yapılan uygulama üç kısımdan oluşmaktadır. Bunlar yüz tanıma servisi, bankacılık uygulaması ve bankamatik uygulamasıdır. Yüz tanıma servisi yüz kaydetme ve doğrulama işlemi yapabilmek için kullanılmaktadır. Bu servis

bankacılık uygulamasından alınan yüzlerin özniteliklerini çıkararak kaydedilmesini sağlar. Aynı zamanda daha önce veri tabanında kayıtlı olan öznitelikler ile bankamatiklerden alınan yüzlerin karşılaştırıp yüz doğrulanmasını sağlar. Yüz doğrulama işlemi için Google'ın bulduğu ve %99.63 doğruluk ile çalışan FaceNet mimarisi kullanılmıştır. Bu mimarinin bu kadar başarılı olmasının sebebi daha önceki yüz tanıma yöntemlerinden farklı olarak ağı eğitiminde üçlü kayıp yöntemini kullanmasıdır. Üçlü kayıp yöntemi kendisiyle aynı kimliğe sahip olan resimleri kendisine yakınlaştırırken, kendisinden farklı olan kişilerden alınan resimleri ise uzaklaştıracak şekilde ağı günceller. Böylelikle sadece kendi kimliğine ait resimleri bulurken aynı zamanda kendi kimliğinden olmayan resimleri de tespit eder.

Bankacılık uygulaması müşteriden aldığı yüzleri yüz servisine gönderip öznitelikleri alır ve müşterinin diğer bilgileri ile beraber veri tabanına kaydeder.

Bankamatik uygulaması ise veri tabanında kayıtlı olan öznitelikler ile kameradan aldığı resimleri yüz servisine gönderir ve standart bankamatik doğrulamaya ek olarak yüz doğrulamanın yapılmasını sağlar.

## 2. KAYNAK ARAŞTIRMASI

Diğer bilim dallarında olduğu gibi yapay sinir ağlarında (YSA) da dönem dönem iniş çıkışlar yaşanmıştır. YSA'nın geçmişi 1940 yılına dayanmaktadır. Elektronik bilgisayarların programlanması ile gündeme gelmiştir (Macukow, 2016). 1943 yılında nörofizyolog ve bir genç matematikçi olan Walter Pitts ve Warren McCulloch ilk nörolojik ağ modelini tanıttılar (McCulloch ve Pitts, 1943). Bu aynı zamanda ilk yapay sinir hücresinin oluşturulma tarihidir. Bu denli basit yapılı bir ağ bile tüm mantık ve aritmetik fonksiyonları hesaplayabilmekteydi. Aynı zamanda birden çok yapay sinir hücresinin birlikte çalışması gerektiğini belirtmişlerdir. Ayrıca ilk öğrenme kurallarını belirlemişlerdir (Keskenler ve Keskenler, 2017).

1949'da, bir psikolog olan Donald O. Hebb genel YSA'nın temelini atan klasik Hebbian kuralını formülize etmiştir. Bu kurala göre; iki nöron aynı anda aktifken, bu iki nöron arasındaki bağlantının güçlendirilmesi gerektiğini ortaya koymuştur. Bu kural önemli bir noktaya değinmiştir. Ancak bu alandaki araştırmaların kısıtlılığından dolayı bunu doğrulayamamıştır (Hebb, 1962).

1950 yılında nöropsikolog Karl Lashley, beyin bilgi deposunun dağıtılmış bir sistem olarak gerçekleştirildiği tezini savunmuştur. Tezi, farelerde yapılan deneylere dayanıyordu. Ancak burada tahrip olmuş sinir dokusunun yeri farelerin bir labirentten çıkış yolunu bulma performansını etkiliyordu (Lashley, 1950).

1951 yılında Marvin Minsky ilk nörobilgisayar olan Snark'ı geliştirdi. Ancak, hiçbir zaman pratikte uygulanmamıştır (Kriesel, 2007).

1957'de Frank Rosenblatt, Charles Wightman ve iş arkadaşları, 20 × 20 piksel görüntü sensörü ile basit sayısal tanıyabilen ve elektromekanik olarak çalışabilen ilk başarılı nörobilgisayar Mark I perceptron'u geliştirdiler (Rosenblatt, 1961).

1960 yılında Bernard Widrow ve Marcian E. Hoff, ADALINE (ADaptive LInear NEuron)'ı geliştirdiler. Bu model, mühendislikte kullanılan ilk YSA çalışmasıdır. Bu model telefon hatlarında yankıları filtrelemek için kullanılmıştır. Günümüzde halen kullanılmaya devam edilmektedir (Widrow ve Hoff, 1960).

1960 yılından sonra YSA düşüşe geçmiştir (Kriesel, 2007).

Fukushima 1971 yılının başlarında, örüntü tanıma ve görüntü işleme konularını ele alan NEOCOGNITRON modelini geliştirmiştir (Fukushima, 1980). Bu model ile beraber YSA daha fazla mühendislik alanında kullanılmaya başlanılmıştır (Kriesel, 2007).

1980 yılı ve sonrasında bu alandaki çalışmalar tekrar hızlanmıştır. 1982’de Kohonen tarafından yazılan “Kendi kendine öğrenme nitelik haritaları” isimli makale yayınlanmıştır. Rumelhart’ın YSA açısından devrim niteliğinde olan çok katmanlı ağların ortaya atılmasını sağlayan eseri ile bir kez daha YSA’nın temelleri atılmış ve yeniden popüler olmuştur (Kriesel, 2007).

1986 yılında Rumelhart ve McClelland tarafından YSA için çok büyük öneme sahip olan geriye yayımlı algoritması geliştirilmiştir (Chan ve ark., 1987).

2006 yılında Geoffrey Hinton YSA’da birden çok katmanın kullanılmasıyla derin öğrenmenin temellerini atmıştır. 2012 yılında ImageNet (Büyük Ölçekli Görsel Tanıma Yarışması) yarışmasındaki başarısı ile beraber çalışmalar daha da hızlanmış ve bugünkü halini almıştır (Hinton ve Salakhutdinov, 2006). Ayrıca daha başarılı sinir ağlarının tasarlanması için derinliğin artması gerektiği belirtilmiştir (Zhang ve ark., 2018).

Derin öğrenmenin temeli çok geçmişte olmasına rağmen son zamanların en popüler konusu olmasının başlıca sebepleri vardır. Bunlardan birincisi eğitim için kullanılacak yeterli verinin olmasıdır. Günümüzde internet ve sosyal medyanın aktif olarak kullanılmasıyla veri miktarı artmıştır. Örneğin, Flickr her gün yaklaşık 3.6 TB veri oluşturur ve Google her gün yaklaşık 20.000 TB veri işler. Ulusal Güvenlik Ajansı her gün internette yaklaşık olarak 1.8 PB veri toplamaktadır (Zhang ve ark., 2018). İkinci olarak birden fazla katmanın ve her katmanda gerçekleştirilecek olan işlemleri yapabilecek güçlü donanımların olmasıdır. Gizli katmanların artmasıyla büyük belleklere ve daha hızlı işlem yapabilen hızlı bilgisayarlara ihtiyaç duyulmuştur. Bundan dolayı eğitim için Central Processing Unit (CPU) çok yavaş ve çok uzun süreler alacaktır. Bunun yerine Graphic Processing Unit (GPU) kullanılmış ve çok büyük hızlara ulaşıp kısa sürede eğitimler tamamlanmıştır (Özkan ve Ülker, 2017).

Derin öğrenme, karmaşık sistemlerden değerli bilgileri elde eder. Her katmanda eğitim verisine ait farklı özellikler bulmaktadır. Her art arda gelen gizli katman önceki çıktıyı alır (Şeker ve ark., 2017). Alt katmanlarda öğrenilen özelliklerin sonraki katmanlarda kullanılmasına ayrıcalık sağlar. Böylece tespit edilmeye çalışılan özelliklere daha çok odaklanmış olunur (Şahin, 2017).

Derin öğrenme; resim analizi, ses tanıma ve metin anlama gibi birçok uygulamada büyük başarılar elde etmiştir (Zhang ve ark., 2018).

Derin öğrenme konusunda birçok model geliştirilmiştir. En temel derin öğrenme modelleri, yığılı oto kodlayıcı(SAE), derin inanç ağları(DBN), evrimsel sinir ağı (ESA) ve devirli sinir ağı (RNN) dir. Bu modeller en çok kullanılan modellerdir. Diğer derin



öğrenme modellerinin çoğu, bu dört derin mimarinin versiyonlarıdır (Zhang ve ark., 2018).

1959 yılında Hubel ve Wiesel, hayvanların görme merkezinde bulunan ve reseptörlerdeki ışığı yakalamaktan sorumlu hücreleri keşfetmişlerdir. 1990 yılında LeCun, modern ESA'nın yapısı hakkında yeni fikirler içeren bir bildiri yayınlamıştır. Daha sonra bunu geliştirmiştir. LeNet 5 isminde el yazısı rakamları tanıyabilen çok katmanlı bir sinir ağı geliştirmiştir. Diğer sinir ağları gibi LeNet 5 de çok katmana sahiptir ve geri yayılım algoritması ile eğitilebilmektedir. Çok az veya hiç ön işleme tabi tutulmadan, satır bazlı piksellerden görsel örüntüleri tanımaya olanak sağlayan orijinal resim özelliklerini elde edebilmektedir. Zhang ve arkadaşları tarafından yürütülen paralel bir çalışma ile resim içerisindeki karakterlerin tespiti için “resim yer değişikliğinden etkilenmeyen, çok büyük değişikliklere karşı dayanıklı” bir sinir ağı (shift-invariant) yani SIANN'i (Shift Invariant Neural Network) kullanmışlardır. Ancak, büyük eğitim verisi eksikliğinden ve hesaplama gücünden dolayı bu ağlar; karmaşık problemler, büyük resimler ve video sınıflandırma işlemlerinde performans gösterememiştir (Gu ve ark., 2018).

2006 yılından beri ESA'nın eğitimi sırasında karşılaşılan zorlukları aşmak amacıyla birçok metot geliştirilmiştir. Krizhevsky ve arkadaşları, önceki resim sınıflandırma metotlarına nazaran gözle görülür gelişmelere sahip bir klasik ESA mimarisi önermiştir. Önerdikleri metot Alex Net olarak geçmektedir. Genel olarak Alex Net metodundaki mimari LeNet-5 mimarisine benzemektedir. Ancak daha derin bir mimariye sahiptir. Alex Net'in başarı sağlamasının ardından metodun performansını arttırmak amacıyla birçok çalışma yapılmıştır. Bu çalışmalara ZF Net, VGG Net, GoogleNet ve ResNet örnek verilebilir. Mimarilerin gelişmesinde, ağların daha çok derinleşmesi şeklinde bir yönelim vardır. Yani kısaca mimari geliştikçe ağ derinleşmektedir. Alex Net'ten 20 kat ve VGG Net'ten 8 kat daha derin olan ResNet; ImageNet isimli “Nesne tanıma ve büyük boyuttaki resimlerin sınıflandırılmasında kullanılan algoritmaları değerlendiren” yarışmanın kazananı olmuştur. Derinliğin artması ve doğrusalsızlığın artmasıyla sinir ağı, hedef fonksiyonuna daha çok yaklaşmakta ve daha iyi özellik gösterimleri yapabilmektedir. Ancak bunlar beraberinde ağın optimizasyonunu güçlendirmekte ve aşırı uyuma (overfitting) sebep olabilmektedir. Ayrıca karmaşıklık artışına da sebep olabilmektedir (Gu ve ark., 2018).

### 3. MATERYAL VE YÖNTEM

#### 3.1. Yapay Sinir Ağları

YSA, insan beyninin biyolojik özellikleri örnek alınarak geliştirilmiştir. Sinir sisteminde bulunan algılayıcıları örnek alan, önceden öğrenmiş olduğu bilgileri kullanarak yeni bilgiler oluşturan bir bilgisayar programıdır. YSA görüntü işleme, sınıflandırma, bilgisayar görüşü gibi birçok alanda aktif olarak kullanılmaktadır. Aynı zamanda YSA, giriş olarak verilen verilere karşılık bir çıktı vermektedir. Bunu yapabilmesi için önceden ağına belirli veriler ile eğitilmesi gerekmektedir. Bu eğitim verisinden bir genel kural seti oluşturup karar verecek olgunluğa ulaşmaktadır (Keskenler ve Keskenler, 2017).

##### 3.1.1. Biyolojik sinir ağları

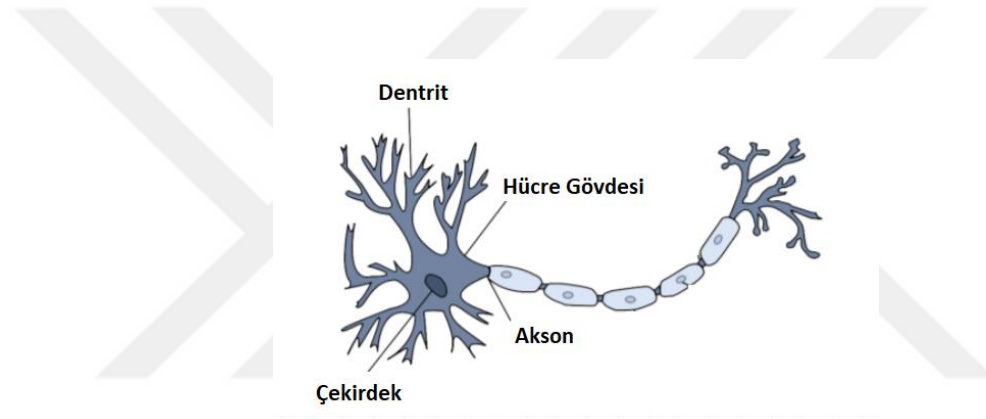
YSA, insan sinir sisteminden esinlenerek geliştirildiği için biyolojik sinir sistemini incelemek fayda vardır. İnsan beyni sayısız sinir hücresinden meydana gelmektedir. Bu sinir hücrelerinde bulunan nöronların arasındaki bağlantı sayısı 60 trilyon olduğu söylenmektedir. Bu hücreler duyu organlarından aldığı uyarıları bir sonraki sinire sinyal olarak iletir ve bunu merkezi sinir sistemine kadar taşır. Merkezi sinir sistemi bu sinyali anlamlandırdıktan sonra geri dönüş sinyalini üretir. Bu sinyal tepki sinyalleri ile geri duyu organlarına iletilir (Çayıröglü, 2015).

İnsan sinir sistemi temel olarak reseptör, sinir ağı ve efektörler olmak üzere üç aşamadan oluşmaktadır. Reseptörler dış dünyadan duyu organları ile uyarıları alırlar ve bilgiyi elektriksel sinyaller biçiminde nöronlara iletirler. Nöronlardan oluşan ağ ise bu sinyali işler ve yorumlar. Son olarak efektörler ise sinyali sinir ağından dış çevreye verilen yanıtlara dönüştürür. Şekil 3.1’de sinir ağlarının çift yönlü iletişimi gösterilmektedir (Kawaguchi, 2001).



Şekil 3.1. Biyolojik nöral sistemin üç aşaması (Kawaguchi, 2001)

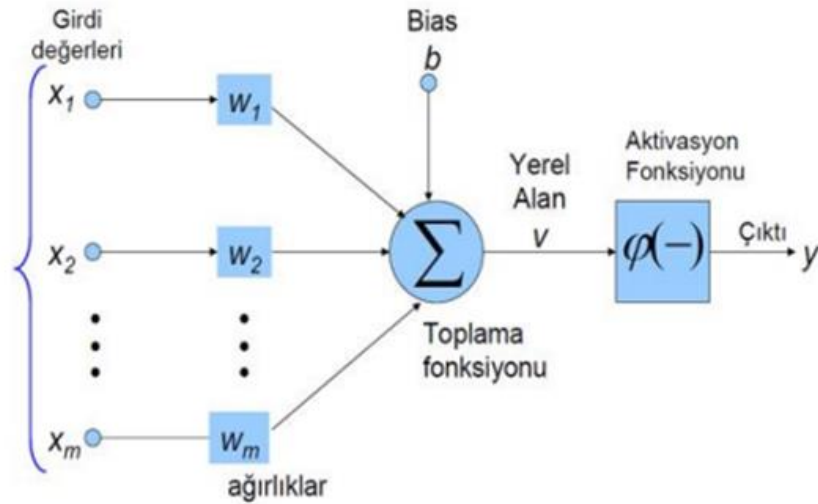
Sinir sisteminin en temel birleşeni nöronlardır. Nöron, özünde diğer nöronlardan bilgi almak, bu bilgiyi benzersiz bir şekilde elde etmek ve sonuçlarını diğer hücrelere göndermek için optimize edilmiştir (Buduma ve Locascio, 2017). Şekil 3.2’de görüldüğü gibi nöronun bir girişi (dendrit), hücre gövdesi (çekirdek), çıkışı (akson) ve bağlantıları vardır (Bell, 2014). Dendritler, ana hücre gövdesinden ortaya çıkan ince yapılardır. Hücrenin bağlı olduğu komşu nöronlardan gelen sinyalleri alır ve dendritin ağırlık değeri ile çarpım gerçekleştirir (Patterson ve Gibson, 2017). Dendritlerin görevi bağlantılı olduğu diğer nöronlar ve duyu organlarından aldığı sinyalleri çekirdeğe aktarmaktır. Çekirdek gelen bu sinyalleri birleştirerek aksonlara gönderir. Bir araya gelen bu sinyalleri akson işleyerek nöronun diğer kısmında bulunan bağlantılara iletir. Bu bağlantıdan da diğer nöronlara gönderir (Çayıroğlu, 2015).



Şekil 3.2. Biyolojik bir sinir hücresi (Kriesel, 2007)

### 3.1.2. Yapay sinir hücresinin yapısı

Beynimizdeki nöronların bu işlevsel anlayışını bilgisayarda temsil edebileceğimiz yapay bir modele çevirebiliriz (Buduma ve Locascio, 2017). Bunu açıklayacak olursak temel bir YSA; girdi, ağırlık, toplama fonksiyonu, aktivasyon fonksiyonu ve çıkış değerinden oluşmaktadır.



Şekil 3.3. YSA örneği (Keskenler ve Keskenler, 2017)

Şekil 3.3'te görüldüğü gibi  $x_1, x_2$  ve  $x_3$  sinir hücresinin giriş değeridir.  $w_1, w_2$  ve  $w_3$  değerleri her bir giriş değerinin ağırlığını ifade eder. Dışarıdan gelen giriş sinyali düğüme ulaşmadan önce ağırlık ile çarpılır. Düğüme ağırlıklı sinyaller toplandıktan sonra bu değerler ağırlıklı toplam olarak eklenir (Kim, 2017). Bu toplama fonksiyonuna bias değeri denilen bir değerde eklenir. Bu değer tüm girdilerin sıfır olması durumunda çıktının ne olduğunu belirtmektedir (Şahin, 2017).

Toplama fonksiyonu matematiksel olarak Denklem 3.1'deki gibi ifade edilir.

$$net = \sum_{i=0}^n w_i x_i + b = w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_n x_n + b \quad (3.1)$$

Toplama fonksiyonun çıkışı bir aktivasyon fonksiyonundan geçirilip çıkış elde edilir. Matematiksel olarak Denklem 3.2'deki gibi ifade edilir (Gadoue, 2007).

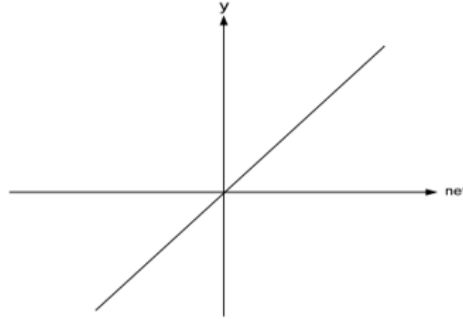
$$y = f(net) = (\sum_{i=1}^n w_i x_i + b) \quad (3.2)$$

### 3.1.3. Aktivasyon fonksiyonları

#### 3.1.3.1. Lineer aktivasyon fonksiyonu

Lineer aktivasyon fonksiyonu doğrusallık problemini çözmek için kullanılır. Çok tercih edilen bir aktivasyon fonksiyonu değildir. Toplama fonksiyonunun değerini sabit

bir sayıyla çarpıp çıkışa iletir. Şekil 3.4'te lineer aktivasyon fonksiyonu örneği verilmiştir (Çayıroğlu, 2015).

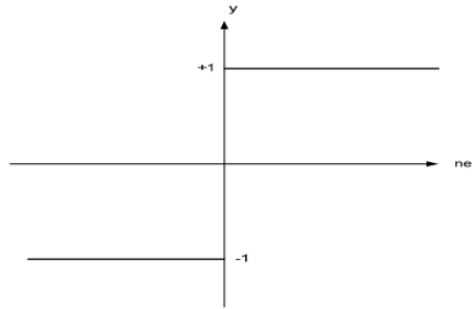


Şekil 3.4. Lineer aktivasyon fonksiyonu (Gadoue, 2007)

### 3.1.3.2. Eşik aktivasyon fonksiyonu

Bu aktivasyon fonksiyonu çıktısını -1 ve +1 ile sınırlandırır. Ağın çıkışı matematiksel olarak Denklem 3.3'teki gibi ifade edilir. Şekil 3.5'te eşik aktivasyon fonksiyonu örneği verilmiştir (Çayıroğlu, 2015).

$$y = \begin{cases} +1, & \text{if } net > 0 \\ -1, & \text{if } net < 0 \end{cases} \quad (3.3)$$



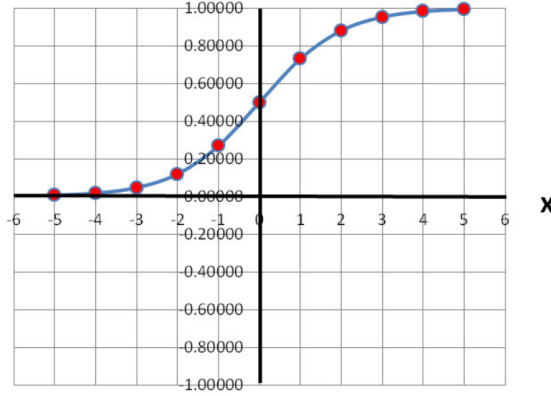
Şekil 3.5. Eşik aktivasyon fonksiyonu (Nordbotten, 2006)

### 3.1.3.3. Sigmoid aktivasyon fonksiyonu

Sigmoid aktivasyon fonksiyonu, toplama fonksiyonu çıktısına bir eşik uygulayarak ağın çıkışını 0 ile 1 değerleri arasına sıkıştırır. Yani negatif değerleri sıfıra yaklaştırırken büyük pozitif değerleri ise bire ölçeklendirir. Sigmoid fonksiyonu Denklem

3.4'teki gibi ifade edilmektedir. Şekil 3.6'da sigmoid aktivasyon fonksiyonu örneği verilmiştir.

$$f(x) = 1.0/(1.0 + e^{-x}) \quad (3.4)$$

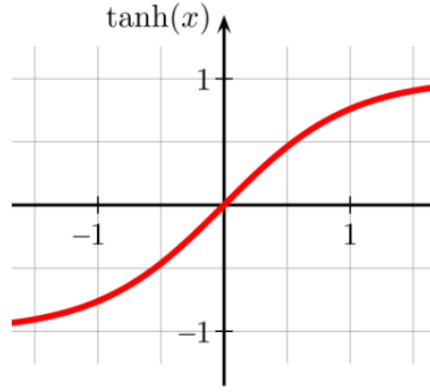


Şekil 3.6. Sigmoid aktivasyon fonksiyonu (Maladkar, 2018)

Şekil 3.6'da eşik değeri 0.5 olarak ayarlanmış. Değer 0.5'in üzerindeyse 1'e doğru yaklaşır. 0.5'in altında ise 0'a doğru yaklaşır. Geçmiş yıllarda çok popüler bir aktivasyon fonksiyonu olmasına rağmen, günümüzde çok tercih edilmemektedir (Maladkar, 2018).

### 3.1.3.4. Tanh aktivasyon fonksiyonu

Tanh aktivasyon fonksiyonu, sigmoid aktivasyon fonksiyonunda olduğu gibi toplam fonksiyonunun çıkışına bir eşik uygulayarak çıkış değerini -1 ile 1 arasında ölçeklendirir. Büyük negatif sayıları -1'e doğru ölçeklendirir ve büyük pozitif sayıları ise 1'e doğru ölçeklendirir. Avantajı negatif değerleri de kapsamasıdır. Şekil 3.7'de tanh aktivasyon fonksiyonu örneği verilmiştir (Maladkar, 2018).



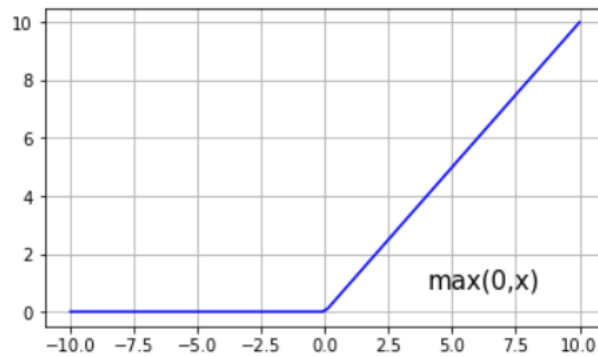
Şekil 3.7. Tanh aktivasyon fonksiyonu (Maladkar, 2018)

### 3.1.3.5. ReLU aktivasyon fonksiyonu

ReLU aktivasyon fonksiyonu dünyada en yaygın olarak kullanılan aktivasyon fonksiyonudur. ReLU'nun avantajı çok hızlı olmasıdır. Derin bir ağda sigmoid gibi bir aktivasyon fonksiyonu kullanıldığında oldukça maliyetli olacaktır. Her bir nöron için bu hesaplama tekrarlanacaktır. ReLU ise sıfırdan küçük olan değerler için hesaplama yapmadığı için çok hızlıdır. Bu sebepten dolayı evrimsel ağlarda çok sık olarak kullanılır. Denklem 3.5'teki gibi matematiksel olarak ifade edilir.

$$f(x) = (0, maks) \quad (3.5)$$

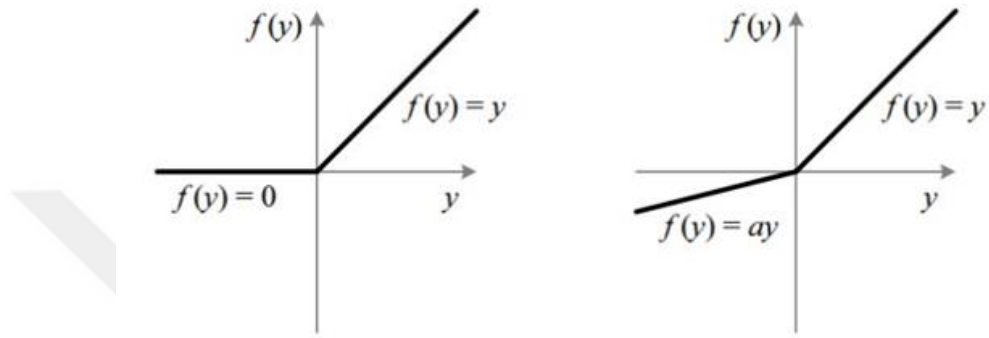
Denklem 3.5'te görüldüğü gibi yalnızca maksimum değerler geçecektir. Bu durum hız kazandırmasının yanında negatif olan nöronları sıfıra çekeceği için ölü nöronları arttıracak ve öğrenme sürecini zorlaştıracaktır. Şekil 3.8'de ReLU aktivasyon fonksiyonu verilmiştir (Maladkar, 2018).



Şekil 3.8. ReLU aktivasyon fonksiyonu (Maladkar, 2018)

### 3.1.3.6. Sızıntılı ReLU (Leaky ReLU)

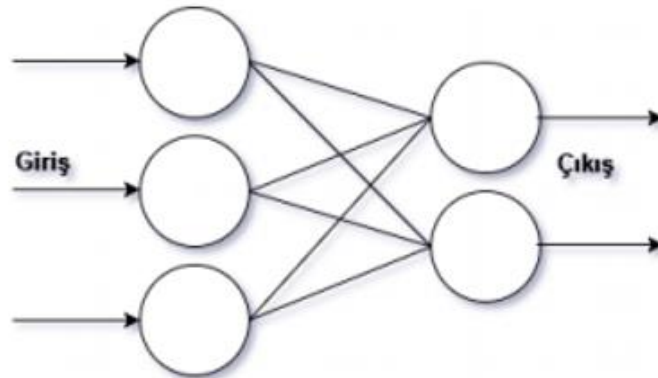
ReLU aktivasyon fonksiyonundaki ölü nöron problemine çözüm olarak geliştirilmiştir. Negatif değerleri çok küçük bir değer olan 0.01 ile çarparak nöronun aktif kalmasını sağlar. Şekil 3.9'da sızıntılı ReLU aktivasyon fonksiyonu verilmiştir (Maladkar, 2018).



Şekil 3.9. ReLU ve Sızıntılı ReLU aktivasyon fonksiyonları (Maladkar, 2018)

### 3.1.4. Tek katmanlı yapay sinir ağları

Tek katmanlı YSA Şekil 3.10'da gösterildiği gibi giriş katmanları direk çıkışa bağlıdır arada gizli bir katman bulunmaz. Doğrusal problemlerin hesaplanmasında kullanılır (Yiğit, 2017).

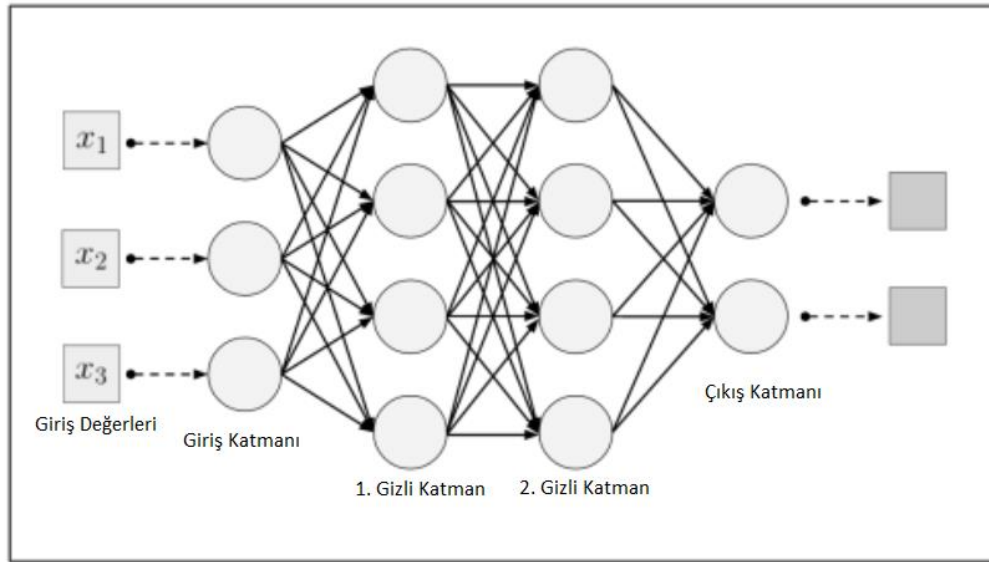


Şekil 3.10. Tek katmanlı yapay sinir ağı (Yiğit, 2017)



### 3.1.5. Çok katmanlı yapay sinir ağları

Çok katmanlı YSA Şekil 3.11’de gösterildiği gibi giriş ve çıkış katmanını dışında bir ara katmana daha sahip olan ağlardır. Daha yüksek seviyeli hesaplama yetenekleri elde etmek için, daha karmaşık bir ağ yapısına ihtiyaç duyulmaktadır. Giriş düğümleri bilgiyi ilk gizli katmandaki birimlere iletir. Daha sonra yapılan hesaplamalar sonucu bir sonraki gizli katmana iletilir ve bu şekilde tamamlanır. Ağ tasarımcısı yapacağı hesaplama karmaşıklığına bağlı olarak gizli katman sayısını arttırabilir. Tek katmanlı ağlara kıyasla daha karmaşık problemler çözerler (Kawaguchi, 2001).

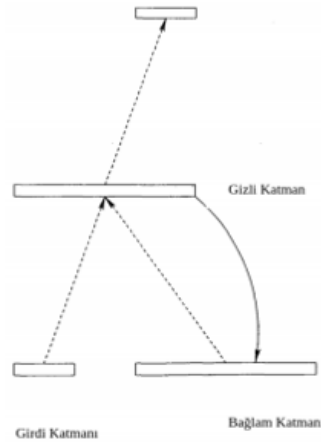


Şekil 3.11. Çok katmanlı yapay sinir ağı (Patterson ve Gibson, 2017)

### 3.1.6. Geri beslemeli yapay sinir ağları

Şekil 3.12’de gösterildiği gibi geri beslemeli YSA ileri beslemeli ağlardan farklı olarak bir nöronun çıktısı sadece kendisinden sonra gelen nöronun girişine verilmez. Aynı zamanda kendisinden önceki katmana veya kendi katmanından bir nörona da giriş olarak verilir. Bu yapı sayesinde doğrusal olmayan bir hareket sergiler (Çayiroğlu, 2015).

Bu ağların çıkış değeri giriş değerine verilirken belli bir gecikme olmaktadır. Bu sebepten dolayı bu ağlar dinamik hafızaya sahiptir (Taşova, 2011).



Şekil 3.12. Geri beslemeli yapay sinir ağı (Yiğit, 2017)

### 3.1.7. Maliyet fonksiyonları

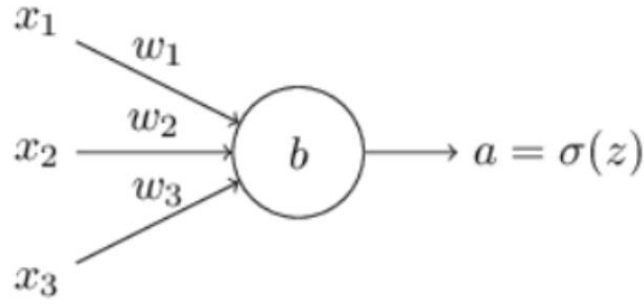
YSA'nın eğitimi sırasında ağırlıkların alabilecekleri tüm değerlerinin denenerek, ağın en başarılı olduğu ağırlıkların tespit edilmesi çok karmaşık ve zaman alacaktır. Bu zaman kaybı ve karmaşıklıktan kurtulmak için en başarılı ağırlıkların bulunabilmesinde maliyet fonksiyonları kullanılmaktadır. Maliyet fonksiyonları ağın çıkışında beklenen değer ile hesaplanan değer arasındaki uzaklığı hesaplamaktadır. Diğer bir deyişle hatayı hesaplamaktadır. En sık kullanılan maliyet fonksiyonları karesel maliyet fonksiyonu (quadratic cost) ve çapraz entropi (cross entropy) fonksiyonlarıdır.

Karesel maliyet fonksiyonu Denklem 3.6'daki gibi ifade edilir.

$$c(w, b) \equiv \frac{1}{2n} \sum_x ||y(x) - a||^2 \quad (3.6)$$

Denklem 3.6'da kullanılan  $n$  eğitim verisi içerisindeki eleman sayısını,  $a$  girdi indisine bağlı olarak ağın çıktısını,  $w$  ağın ağırlıklarını ve  $b$  ise bias değerini ifade etmektedir. Bu işlem tüm eğitim verisine uygulanmaktadır. Tahmin edilen  $y(x)$  değeri  $a$  çıktısına yaklaştığında, bu fonksiyon da sıfıra yaklaşmaktadır. Bu durum ağın öğrenmesinin iyi olduğunu ifade etmektedir.

Şekil 3.13'te çapraz entropi uygulamak için örnek bir nöron verilmiştir. Burada  $x_1, x_2, x_3$  girdi değerleri,  $w_1, w_2, w_3$  ağırlıklarını ve  $b$  bias değerini göstermektedir. Ayrıca  $a = \sigma(z)$  değeri nöronun çıktı değerini ve  $z$  giriş değerlerinin toplamını ifade etmektedir.



Şekil 3.13. Yapay nöron (Şahin, 2017)

Çapraz entropi fonksiyonu nörona Denklem 3.7'deki gibi uygulanmaktadır.

$$c = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)] \quad (3.7)$$

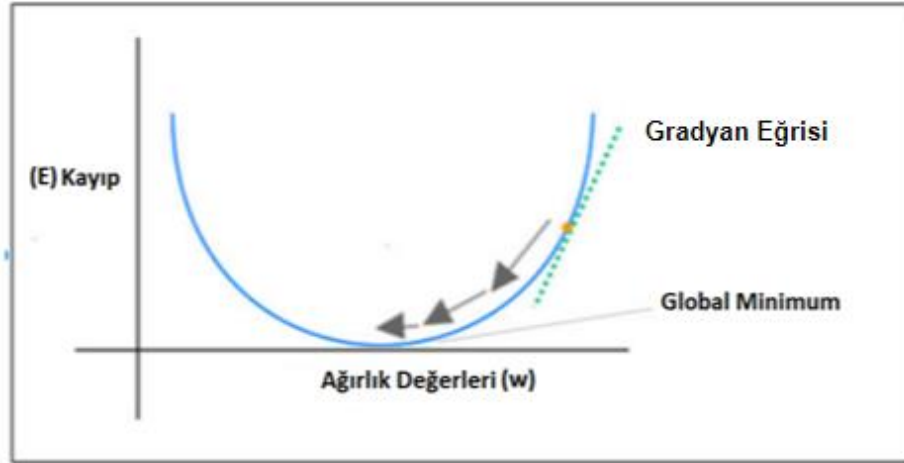
Denklem 3.7'de  $n$  eğitim verisindeki toplam eleman sayısını,  $x$  eğitim verisi içerisindeki girdilerin toplamını,  $y$  değeri beklenen çıktıyı ifade etmektedir (Şahin, 2017).

### 3.1.8. Gradyan iniş algoritması (Gradient Descent)

Gradyan iniş algoritması, maliyet fonksiyonu aracılığıyla tespit edilen hata oranını en aza indirmeyi amaçlamaktadır. Bu algoritma eğitim aşamasında ağırlık ( $w$ ) ve bias ( $b$ ) değerini değiştirerek optimum sonuca ulaşmayı amaçlamaktadır. Bu aşama YSA eğitiminin en önemli aşamalarından birisidir.

Gradyan iniş, maliyet fonksiyonunun türevini alarak herhangi bir noktanın teğet eğimini verir. Bu şekilde ağırlık ve bias değerini güncelleyerek en alt noktaya yani en iyi ağırlık değerlerine ulaşır.

Öğrenme hızı (learning rate), algoritmanın bir sonraki adımı için ne kadar gradyan iniş yapacağını belirten bir katsayıdır. Şekil 3.14'te ağırlıklarına göre hata oranı verilmiştir (Şahin, 2017).



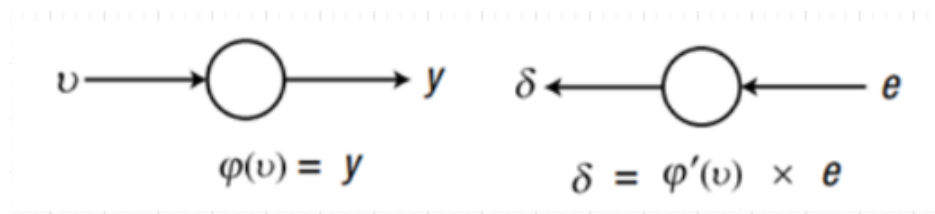
Şekil 3.14. Ağırlık ve kayıp değerine göre global minimum eğrisi (Patterson ve Gibson, 2017)

Şekil 3.14'te görülen  $w$  değeri ağırlıkların  $E$  değeri ağırlıklara bağlı olarak hata miktarını ifade etmektedir. Ağırlıkların optimum olduğu noktada hata değeri en düşüktür. Ağırlık değerleri çok küçük veya çok büyük seçildiğinde yüksek seviyelerde hataya sebep olmaktadır. Ağırlıklar güncellenerek grafikteki en minimum nokta bulunmaktadır.

Eğim azaltmak için üç yöntem bulunmaktadır. Bunlar sırasıyla; toplu eğim azaltma (batch gradient descent), rastgele eğim azaltma (stochastic gradient descent) ve parçalı eğim azaltma (mini-batch gradient descent) yöntemleridir (Şahin, 2017).

Toplu eğim azaltma (batch gradient descent) yöntemi bütün eğitim verisi tek bir seferde okunduktan sonra alçaltma işlemi yapar. Bu işlem eğitim verisinin tümünde bir kere yapılır.

Şekil 3.15'te bir nörona aktivasyon ve geri yayılım uygulanma formülleri gösterilmiştir. Buradan yola çıkarak toplu eğim azaltma yöntemi ağırlık güncellemesini şu şekilde hesaplar:



Şekil 3.15. Bir nörona aktivasyon ve geri yayılım uygulanmasının formülle gösterilmesi (Kim, 2017)

$$\varphi(v) = y \quad (3.8)$$

Denklem 3.8'de gösterilen  $\varphi(v)$  nöron girişine verilen  $y$  değerinin aktivasyon fonksiyonundan geçirilmesi sonucunda çıkan yeni değeri ifade eder.

$$\delta = \varphi'(v) x e \quad (3.9)$$

Denklem 3.9'da belirtilen  $\delta$  değeri hata miktarı ile aktivasyon fonksiyonu türevinin çarpılması sonucunda ortaya çıkan değerdir.

$$\Delta_{w_{ij}} = a \delta_i x_j \quad (3.10)$$

Denklem 3.10'da gösterilen  $\Delta_{w_{ij}}$  yeni ağırlık değerini,  $a$  öğrenme oranını,  $\delta_i$  i indeksli nöronun hata oranını,  $x_j$  değeri önceki nöronun çıkışını ifade etmektedir.

$$\Delta_{w_{ij}} = \frac{1}{N} \sum_{k=1}^N \Delta_{w_{ij}}(k) \quad (3.11)$$

Denklem 3.11'de gösterilen  $\Delta_{w_{ij}}(k)$  k-inci eğitim verisinin ağırlık güncellemesini ifade eder. N ise eğitim verisinin toplam sayısını ifade eder.

Rastgele eğitim azaltma (stochastic gradient descent) yönteminde, her eğitim elemanı sonunda alçaltma hesaplanıp ağırlıklar güncellenir. Eğitim verisindeki her bir eleman için hesaplama yapar. Örneğin 100 adet eğitim verisi varsa 100 defa eğitim hesaplanır.

Rastgele eğitim azaltma, ağırlık güncellemelerini Denklem 3.12'deki gibi hesaplar:

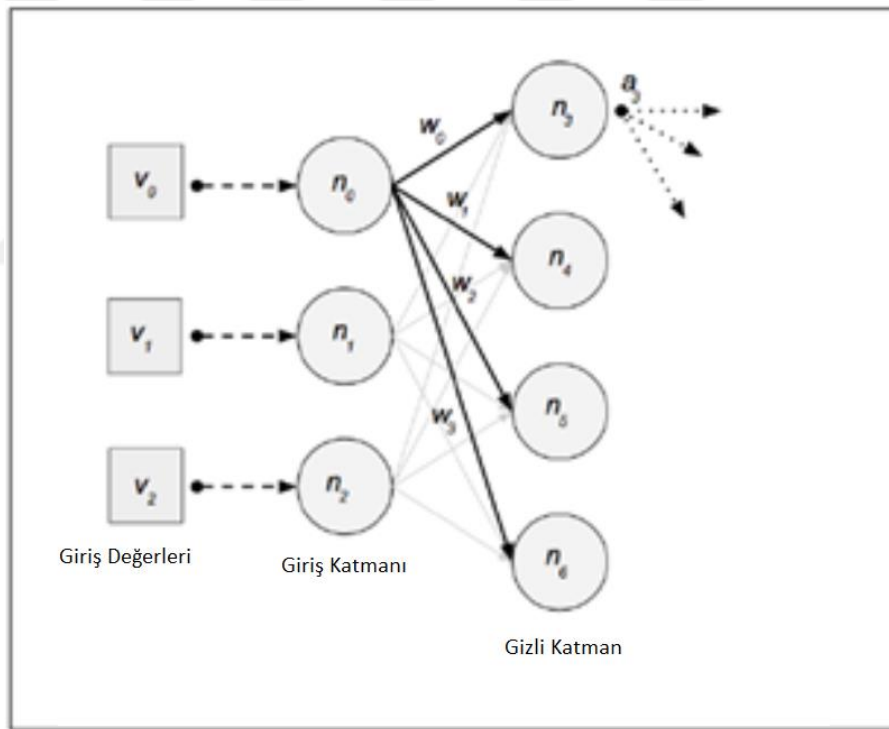
$$\Delta_{w_{ij}} = a \delta_i x_j \quad (3.12)$$

Parçalı eğitim azaltma (mini-batch gradient descent), toplu eğitim azaltma ve rastgele eğitim azaltma arasında bir yaklaşımdır. Tüm verilerin okunarak geri beslemenin yapılması veya teker teker her eğitim elemanına bakılarak yapılması arasında bir yöntemdir. Eğitim veri kümesindeki elemanların bir kısmını seçer ve bunları toplu olarak kullanarak ağırlık hesaplar. Örneğin, 100 adet eğitim elemanı içeren bir eğitim verisi bulunuyor. Bu elemanlar içerisinde 20 rastgele veri noktası seçilirse, toplu iniş yöntemi 20 eğitim verisine uygulanır. Bu durumda tüm veri noktaları için eğitim sürecini tamamlamak üzere toplam beş ağırlık ayarı yapılır ( $5 = 100/20$ ). Parçalı eğitim azaltma

yöntemi, uygun sayıda veri noktası seçtiğinde, her iki yöntemden de faydalanır. Rastgele eğitim azaltmadan daha hızlı, toplu eğitim azaltmadan daha istikrarlı çalışır (Kim, 2017).

### 3.1.9. Geri yayılım algoritması

Geril yayılım ilk kez 1969 yılında Bryson ve Ho tarafından ortaya atılmıştır. 1980 yılına kadar YSA alanındaki durgunluk yüzünden göz ardı edilmiştir. Geril yayılım ağ modelindeki hatanın azaltılması açısından önemli rol oynamaktadır. Bir sinir ağı eğitilirken çıktı değeri ile beklenen değeri aynı ise hiç bir şey yapılmaz. Ancak farklıysa ağı ağırlıklarının güncellenmesi gerekmektedir. Geril yayılım işlemi, eğitim girdisine bağlı olarak beklenen çıkış ile üretilen çıkış arasındaki hatayı en aza indirmeyi amaçlamaktadır. Her bir ağırlık için hata oranını bölmek için pragmatik bir yaklaşımdır.



Şekil 3.16. Çok katmanlı sinir ağı örneği (Patterson ve Gibson, 2017)

Şekil 3.16'da çok katmanlı bir YSA verilmiştir. Bu ağ geril yayılım için kullanılacaktır. Ancak öncesinde denklemlerin daha iyi anlaşılması için kullanılan terimler Çizelge 3.1'de ifade edilmiştir.

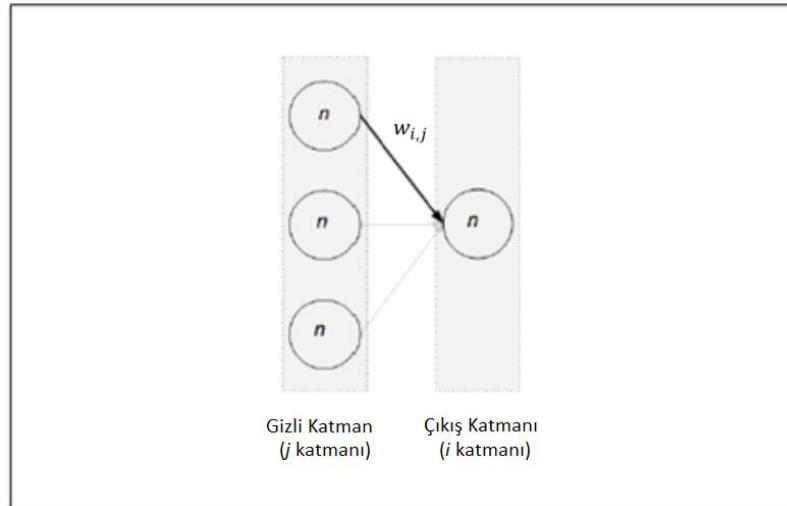
**Çizelge 3.1.** Geri yayılım algoritması için kullanılan terimler (Patterson ve Gibson, 2017)

Gösterim	İfade ettiği anlam
$i$	Her bir nöronun indeksini ifade eder.
$n_i$	İndeks $i$ 'de bulunan nöronu ifade eder.
$j$	Nöron $i$ 'ye bağlanan önceki katmandaki nöronun indeksini ifade eder.
$a_i$	$i$ nöronunun aktivasyon değerini ifade eder. ( $i$ nöronunun çıkışı )
$A_i$	Girişlerden nöron $i$ 'ye kadar olan aktivasyon değerleri vektörünü ifade eder.
$g$	Aktivasyon fonksiyonunu ifade eder.
$g'$	Aktivasyon fonksiyonunun türevini ifade eder.
$Err_i$	Ağın çıkışı ve eğitim örnekleri için hesaplanan çıkışlar arası farkı ifade eder.
$w_i$	$i$ nöronunun ağırlığını ifade eder.
$w_{i,j}$	Önceki nöron $j$ 'den nöron $i$ 'ye gelen bağlantı üzerindeki ağırlığı ifade eder.
$input\_sum_i$	Nöron $i$ 'ye gelen ağırlıklı girişlerin toplamını ifade eder.
$input\_sum_j$	Önceki katmanda yer alan nöron $j$ 'ye ait ağırlıklı girişlerin toplamını ifade eder. (Geri yayılım için kullanılır. )
$a$	Öğrenme oranını ifade eder.
$\Delta_j$	Önceki katmanda yer alan bağlantılı nöron $j$ 'nin hata miktarını ifade eder.
$\Delta_i$	Nöron $i$ için hata ifadesi;= $Err_i \times g'(input\_sum_i)$

Ağın çıkış ağırlığı Denklem 3.13'teki gibi hesaplanır.

$$w_{i,j} \leftarrow w_{i,j} + a \times a_j \times Err_i \times g'(input\_sum_i) \quad (3.13)$$

Bu kural aynı zamanda  $j$  nöronu ile  $i$  nöronu arasındaki tüm bağlantıların güncellenmesi için kullanılan ağırlık güncelleme kuralıdır. Çıkış katmanına etki eden ağırlıklar Şekil 3.17'de gösterilmiştir.

**Şekil 3.17.** Çıkış katmanındaki ağırlıkların güncellenmesi (Patterson ve Gibson, 2017)

Denklem 3.14'te gösterilen nöron  $i$  için, aktivasyon fonksiyonuna gelen toplam ağırlık girişinin hesaplanması; gelen ağırlık vektörü  $w_j$  için iç çarpımlarını ve aktivasyon vektörü  $A_j$  hesaplanır ve sonra bias değeri ile toplanır.

$$input\_sum_i = W_i \cdot A_i + b \quad (3.14)$$

Nöron  $j$  için aktivasyon değeri Denklem 3.15'teki gibi hesaplanmaktadır.

$$a_j = g(input\_sum_j) \quad (3.15)$$

Nöron  $i$ 'deki hata miktarı ( $e$ ),  $Err_i$  olarak gösterilmiştir. Aktivasyon fonksiyonunun türevi ise  $g'(x)$  olarak gösterilmiştir. Nöron  $i$ 'nin net girişine Denklem 3.16'daki formül uygulanmıştır.

$$g'(input\_sum_j) \quad (3.16)$$

Hata miktarı Denklem 3.17'deki gibi hesaplanmaktadır.

$$\Delta_i = Err_i \times g'(input\_sum_i) \quad (3.17)$$

Ağırlık güncelleme fonksiyonu ise Denklem 3.18'deki gibi hesaplanmaktadır.

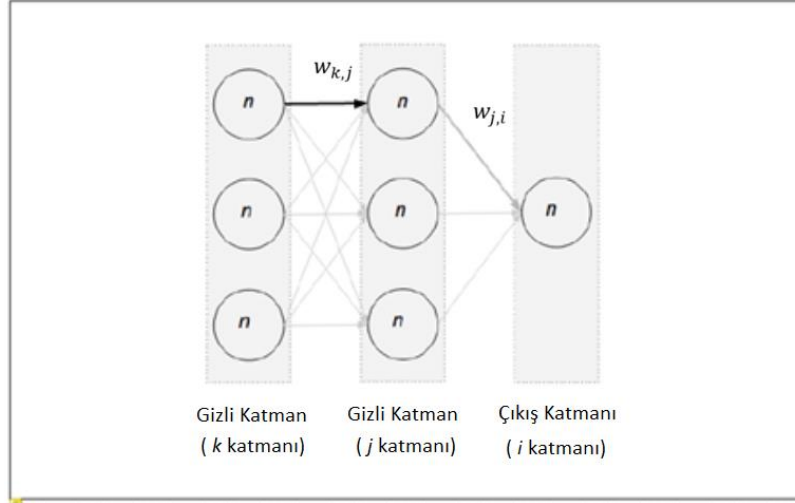
$$w_{j,i} \leftarrow w_{j,i} + a \times a_j \times \Delta_i \quad (3.18)$$

Hata değerinin yeni yayılım kuralı: Bu kural giriş ile gizli katman arasındaki ağırlık güncelleme kuralını verir. Denklem 3.19'daki formül ile hesaplanır.

$$\Delta_j \leftarrow g'(input\_sum_j) \sum_i W_{j,i} \Delta_i \quad (3.19)$$

Gizli katmanların ağırlıklarının güncellenmesi ise giriş katmanına varıncaya kadar Şekil 3.18'de görüldüğü gibi önceden hesaplanan küçük hata oranı alınır ve önceki katmandan gelen aktivasyon ve öğrenme oranıyla çarpılır.





Şekil 3.18. Gizli katmanlar arası bağlantıların gösterimi (Patterson ve Gibson, 2017)

Ağırlıklar hesaplandıktan sonra Denklem 3.20’de gösterildiği gibi hesaplanan değerler alınır ve önceki ağırlığı ile toplanır.

$$\Delta_j \leftarrow g'(input\_sum_j) \sum_i W_{j,i} \Delta_i \quad (3.20)$$

Mevcut ağırlıklar ve bias değerleri meydana gelen hatadan sorumludur. Doğru çıktıyı destekleyen ağırlık ve bias değerleri güçlendirilmektedir. Bu öğrenme adımlarının uzunluğu veya her iterasyonda değişen ağırlıkların miktarı, öğrenme oranı olarak bilinmektedir. Öğrenme oranı ağı tasarlayan kişinin belirlediği bir parametredir (Patterson ve Gibson, 2017).

## 3.2. Derin Öğrenme

### 3.2.1. Evrimsel sinir ağıları

ESA, canlıların görüntü algılama mekanizmasını örnek alan iyi bilinen bir derin öğrenme mimarisidir. Literatürde ESA ile alakalı sayısız mimari vardır. Ancak temel bileşenleri çok benzerdir. LeNet5 mimarisini örnek olarak alırsak evrişim, havuzlama ve tam bağımlı katman olmak üzere 3 tip katmandan oluşmaktadır. Evrişim katmanı, girişlere ait olan öznitelik temsillerini öğrenmeyi hedefler. Evrişim katmanı, birden çok evrişim çekirdeğinden oluşan ve farklı öznitelik haritalarını hesaplamak için kullanılan

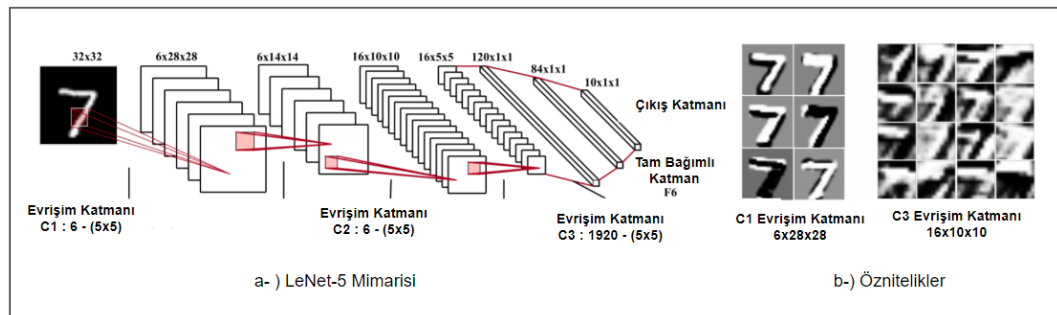
katmandır. Özellik haritasındaki her nöron, önceki katmandaki komşu nöronların bir bölgesine belirgin bir şekilde bağlıdır.

Aktivasyon fonksiyonu ESA'ya doğrusalsızlığı getirir bu da çok katmanlı ağların doğrusal olmayan özellikleri tanımasını sağlamak için istenen bir şeydir. Tipik aktivasyon fonksiyonları sigmoid, tanh ve ReLU'dur. Havuzlama katmanı, özellik haritasının çözünürlüğünü azaltarak yer değişimlerinden etkilenmeyen yapıyı sağlamayı amaçlar. Genelde iki evrişim katmanı arasında yer alır. Havuzlama katmanına ait olan özellik haritası, önceki evrişim katmanında bulunan ve kendisine karşılık gelen özellik haritasına bağlıdır.

Tipik olarak 2 çeşit havuzlama işlemi vardır. Bunlar; ortalama ve maksimum havuzlama işlemleridir. Şekil 3.19(b)'de ilk iki evrişim katman ile öğrenilen 7 rakamına ait özellik haritası gösterilmektedir. Birinci evrişim katmanında yer alan çekirdekler; kenarlar ve kıvrımlar gibi düşük seviyeli özellikleri tanımak için dizayn edilmiştir. Yüksek seviyeli katmanlar ise daha soyut özellikleri kodlamak için vardır. Birden fazla evrişim ve havuzlama katmanını kullanarak daha yüksek seviyeli özellik temsilleri çıkartılabilmektedir.

Birkaç evrişim ve havuzlama katmanından sonra, yüksek seviyeli kararlar vermek amacıyla bir veya daha fazla tam bağlı (fully connected) katman olabilir. Bilgi edebilmek amacıyla, önceki katmandaki tüm nöronları alıp, mevcut katmandaki tüm tek nöronlara bağlarlar. Tam bağlı katmanlar her zaman gerekli değildir. Çünkü onlar 1x1 evrişim katmanlarla yer değiştirebilirler.

Son ESA katmanı, çıktı katmanıdır. Sınıflandırma işlemleri için softmax operatörü yaygın olarak kullanılır. Diğer yaygınca kullanılan metot destek vektör makinaları (DVM) olup farklı sınıflandırma problemlerini çözmek için ESA ile birleştirilebilir.

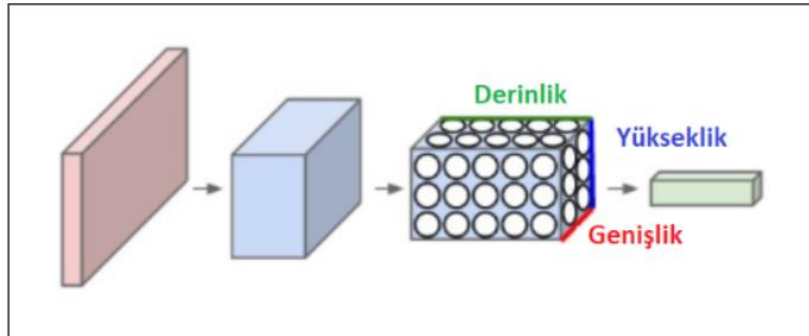


Şekil 3.19. Örnek LeNet-5 ağının mimarisi (Gu ve ark., 2018)

Yukarıda Şekil 3.19(a)'da rakam sınıflandırmada iyi performans sağlayan bir ağ olan LeNet-5 ağının mimarisi verilmiştir. Şekil 3.19(b)'de ise LeNet-5 mimarisine ait özellikler görselleştirilmiştir. Her katmana ait özellik haritaları farklı bloklarda gösterilmiştir (Gu ve ark., 2018).

### 3.2.1.1. Giriş katmanı

ESA'nın başlangıç katmanıdır. Bu katmanda veri herhangi bir işleme tabi tutulmadan olduğu gibi ağa verilir. Oluşturulacak ağ modelinin başarısında giriş katmanına verilen verinin boyutu önemli rol oynamaktadır. Bu verinin çok büyük seçilmesi beraberinde bellek ihtiyacı ve karmaşık işlem gücü getirecektir. Avantaj olarak verinin büyük olması ağın derinliğini arttıracak ve başarı oranı yükselecektir. Başarılı ve hızlı bir model tasarlanmak isteniyorsa hem hesaplama maliyeti hem de modelin başarısı için avantaj sağlayacak bir veri boyutu seçilmelidir (Özkan İnik ve Ülker, 2017). Bu katmanda verilen değer Şekil 3.20'de görüldüğü gibi görüntünün (genişlik, yükseklik, derinlik) değerlerini içerir. Örneğin  $[32 \times 32 \times 3]$  boyutunda bir ham resim olsun. Buradaki genişlik 32, yükseklik 32 ve üç renk kanalı R, G, B ise 3 olarak gösterilmektedir (Andrej Karpathy ve Johnson, 2019).

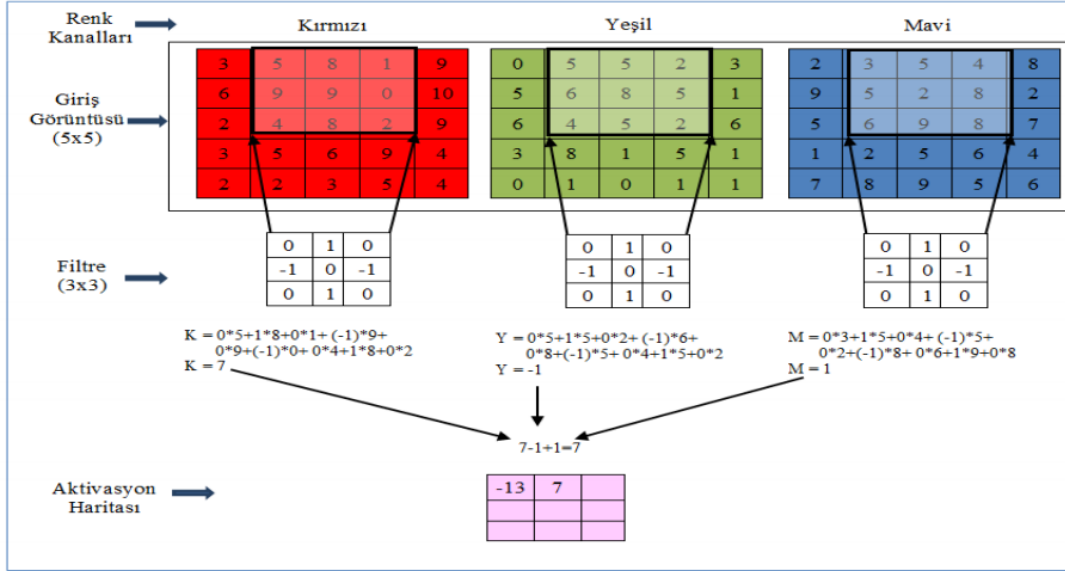


Şekil 3.20. Evrişim sinir ağı giriş katmanı (Andrej Karpathy ve Johnson, 2019)

### 3.2.1.2. Evrişim katmanı (Convolution Layer)

Evrişim katmanı, aynı zamanda dönüşüm katmanı olarak da adlandırılır. Tasarım aşamasında belirlenen filtrenin görüntü üzerinde gezdirilmesi işleminden oluşmaktadır. Filtreler  $2 \times 2$ ,  $5 \times 5$  ve  $3 \times 3$  matris boyutunda olabilir. Evrişim işlemi önceki katmanda oluşan çıktıyı alıp filtre uyguladıktan sonra çıkış verisini bir sonraki katmana iletir. Bu

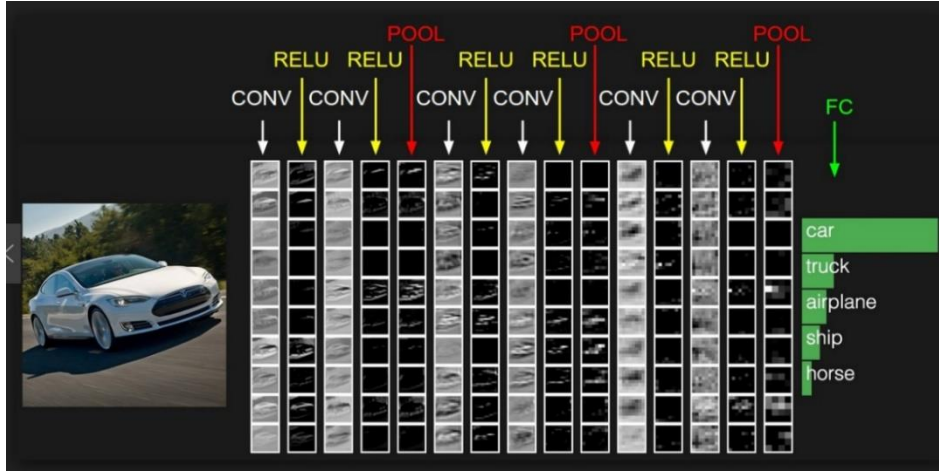
işlem sonucunda özellik haritası oluşur. Özellik haritası her filtrenin kendine has özelliklerin ortaya çıktığı matristir. Eğitim aşamasında bu filtrelerin belirlenen katsayıları, her öğrenme adımında değişir. Böylece verinin hangi kısımlarda önemli olduğu keşfedilir.



**Şekil 3.21.** 5x5x3 boyutta bir giriş görüntüsüne 3x3'lük filtrenin uygulandığı konvolüsyon işlemi (Özkan ve Ülker, 2017)

Şekil 3.21'de evrişim işleminin görüntü üzerine uygulanması verilmiştir. 3x3 boyutunda bir filtrenin görüntü üzerinde uygulanıp özellik haritasının çıkarılması gösterilmektedir. Filtre sağdan sola kaydırılarak her adımda özellik haritası çıkaracak şekilde gezdirilir. Matris sona gelince bir satır aşağıya kayarak tüm matrisi dolaşacak şekilde bu işlemi tamamlar. Filtrenin içerisindeki her eleman denk geldiği hücre ile çarpılarak toplanır. Her üç renk katmanı için bu işlem tekrarlandıktan sonra toplanıp özellik haritasına yazılır.

Filtre matrisinin katsayıları kaç adım kayacağı ve her renkte farklı matrislerin kullanılması gibi konuları tamamen ağ tasarımcısı belirlemektedir. Her uygulanan filtre görüntünün belirli özelliklerini keşfetmektedir.



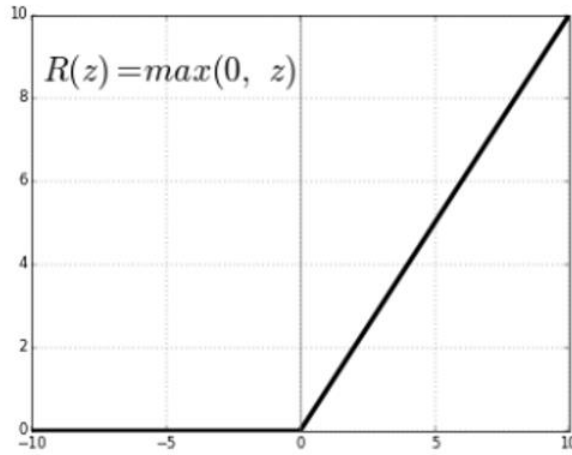
Şekil 3.22. Evrişim ağlarına genel bakış (Afonja, 2018)

Şekil 3.22’de görüldüğü gibi ilk katmana 10 filtre uygulanmış ve her filtrede arabanın farklı bir özelliği ortaya çıkmıştır. İlk filtrede aracın ön hatları keskinleşirken diğer filtrelerde tekerler veya kapılar daha belirgin olmaktadır. Bu da her filtre işleminden sonra resimde bulunan asıl nesnenin bir özelliğini ortaya çıkarmaktadır (Özkan ve Ülker, 2017).

### 3.2.1.3. Düzleştirilmiş doğrusal birim katmanı (Rectified Linear Units Layer (ReLU))

Bu katman evrişim katmanından sonra gelmektedir. ReLU günümüzde kullanılan en yaygın aktivasyon fonksiyonlarından biridir. ReLU fonksiyonu tüm negatif değerleri sıfır yapar. Bu da diğer aktivasyon fonksiyonlarında olduğu gibi bir hesaplama işlemi getirmediği için diğerlerine göre oldukça hızlıdır. ReLU’nun bazı dezavantajları bulunmaktadır. Negatif değerleri sıfıra çektiği için geri yayılım sırasında tüm ağırlıklar güncellenmeyecektir. Bu dezavantajı gidermek için leaky ReLU aktivasyon fonksiyonu da kullanılmaktadır. Ama ReLU’ya göre daha yavaş çalışmaktadır. Matematiksel olarak Denklem 3.21’deki gibi gösterilmektedir. Ayrıca Şekil 3.23’te de ReLU aktivasyon fonksiyonunun grafiği verilmiştir. (Afonja, 2018).

$$f(x) = \begin{cases} 0, & \text{eğer } x < 0 \\ x, & \text{eğer } x \geq 0 \end{cases} \quad (3.21)$$



Şekil 3.23. ReLU aktivasyon fonksiyonu (Sharma ve ark., 2016)



Şekil 3.24. Bir ESA modelinde evrişim katmanı ve ReLU katmanının giriş görüntüsüne yapmış olduğu etki (Özkan ve Ülker, 2017)

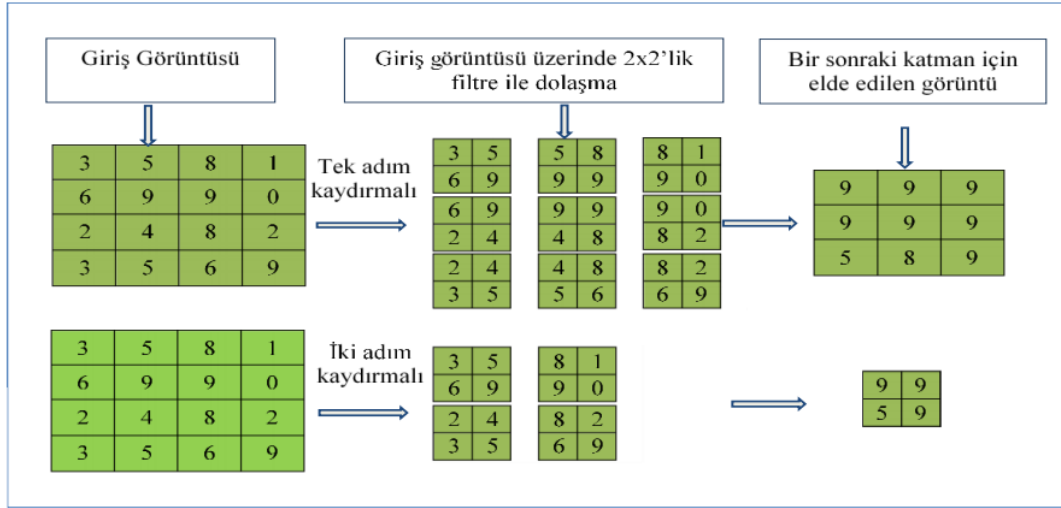
Şekil 3.24'te 3x3 matrise uygulanan filtre sonucunda oluşan çıktı ve bu çıktıya da uygulanan ReLU aktivasyon fonksiyon çıktıları gösterilmektedir

#### 3.2.1.4. Havuzlama katmanı (Pooling Layer)

Bu katman ReLU katmanından sonra gelmektedir. Bu katmandaki amaç; bir sonraki katman için veri boyutunu azaltıp hesaplama maliyetini düşürmektir. Yapılan işlem ağın derinliğinde herhangi bir değişiklik yapmaz. Aynı zamanda veri boyutunu azalttığı için veri kaybına da yol açar. Her ne kadar dezavantaj gibi görünse de hesaplama maliyetini düşürdüğü ve ağın ezberlemesini engellediği için bir o kadar da avantaj sağlamaktadır.

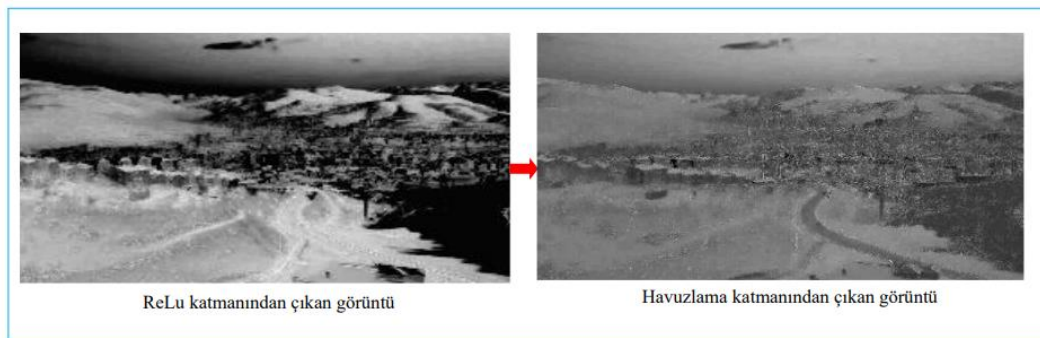
Evrişim katmanında olduğu gibi bir filtre matrisi belirlenir ve bu matris giriş verisi üzerinde gezdirilir. Bu matrisin elemanları içerisinde maksimin olan alınır. Buna

maksimum havuzlama denir. Aynı zamanda ortalaması da alınabilir buna da ortama havuzlama denilir. Genellikle maksimum havuzlama diğer havuzlama yöntemlerine göre daha başarılıdır. Her bir özellik haritasına ayrı ayrı havuzlama işlemi uygulanmaktadır (Şahin, 2017).



**Şekil 3.25.** 4x4'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı maksimum havuzlama işleminin uygulanması (Özkan ve Ülker, 2017)

Şekil 3.25'te 4x4 bir matrisine 2x2'lik bir havuzlama filtresi uygulanmaktadır. Bir adım kaydırılarak uygulanan havuzlamada maksimum değer alınarak 3x3 lük bir matris elde edilmiştir. Ayrıca 2x2'lik bir maksimum havuzlama uygulanılarak 2 birim kaydırılarak 2x2'lik bir matris oluşmuştur (Özkan ve Ülker, 2017).



**Şekil 3.26.** ESA modelinde havuzlama katmanının bir önceki katmandan gelen görüntüye uygulanması sonucu oluşan görüntü (Özkan ve Ülker, 2017)

Şekil 3.26'da ReLU uygulanılarak elde edilen bir görüntüye maksimum havuzlama uygulanmıştır.

### 3.2.1.5. Tam bağımlı katman (Fully Connected Layer)

Bu katman, ağıın çıktı olarak kullanacağı sınıf puanlarını hesaplamak için kullanılmaktadır. Çıkış hacminin boyutları,  $[1 \times 1 \times N]$  şeklindedir. N değeri çıktı sınıflarının sayısıdır. Bu katmadaki tüm nöronların önceki katmandaki her nöron ile arasında bir bağı vardır.

### 3.2.1.6. Sınıflandırma katmanı (Classification Layer)

Bu katman en son katman olan tam bağımlı katmandan önce gelir. Bu katmanda sınıflandırma işlemi yapılır. Bu katmanın çıkışı sınıflandırılacak nesne sayısı kadardır. Bu katmanda birçok sınıflandırma algoritması kullanılmaktadır. Genellikle kullanılan sınıflandırma algoritması softmax sınıflandırıcıdır.

## 3.2.2. Derin öğrenme modelleri

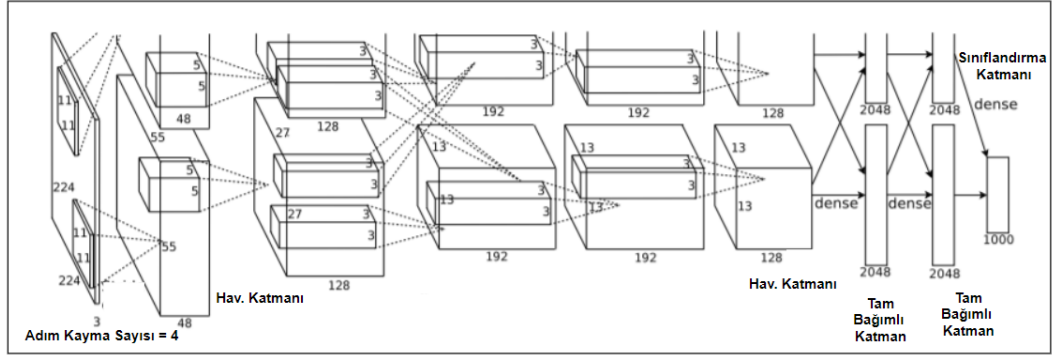
Bu bölümde derin öğrenme konusunda tasarlanan en popüler modeller hakkında bilgi verilecektir. Son 5 yılda gündeme gelen derin öğrenme modelleri ele alınacaktır.

### 3.2.2.1. Alex Net (2012)

Derin öğrenme ile ilgili her şeyi başlatan Yann LeCun'un makalesi olduğu söylene de dünya çapında popüler olmasına “Derin Konvolüsyon Ağları ile ImageNet Sınıflandırması” isimli makale sebep olmuştur. Bu alanda en etkili makalelerden biri olarak kabul edilmektedir. Alex Krizhevsky, Ilya Sutskever ve Geoffrey Hinton, 2012 yılında ImageNet yarışmasını kazanmak için büyük, derin bir evrimsel ağ oluşturmuşlardır. Bu ağa Alex Net adını vermişlerdir. Bu ağda modern mimariye kıyasla basit bir yapı kullanmıştır.

Ağ; 5 evrişim, maksimum havuzlama, düzleştirici ve 3 tam bağımlı katmandan oluşmaktadır. Bu ağ 1000 ayrı kategoride sınıflandırma yapmıştır (Deshpande, 2019). Ağa uygulanan filtreler 11x11 matris boyutunda ve filtrenin kayma adım sayısı 4 olarak kabul edilmiştir (Özkan ve Ülker, 2017).



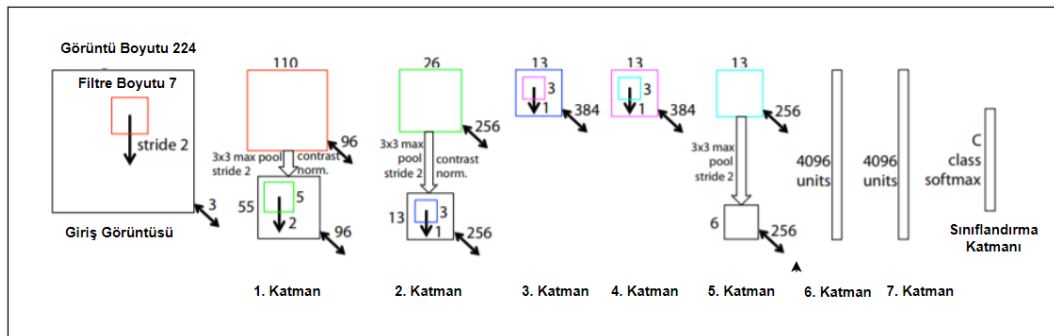


Şekil 3.27. Alex Net modelinin yapısı (Krizhevsky ve ark., 2012)

Şekil 3.27’de Alex Net ağ mimarisi verilmiştir. Bu ağ toplam 22.000’den fazla kategoride ve 15 milyondan fazla açıklamalı resim içeren ImageNet verilerinden eğitildi. Eğitim süresini kısa tutmak için ReLU doğrusal olmayan fonksiyonunu kullanmıştır. Beş ile altı gün boyunca iki GTX 580 GPU üzerinde eğitilmiştir (Deshpande, 2019).

### 3.2.2.2. ZF Net (2013)

Alex Net’in 2012 yılındaki başarısından sonra ImageNet’te 2013’te sunulan modellerde ESA kullanımı artmıştır. 2013 yılındaki yarışmanın galibi Zeiler ve Rob Fergus tarafından geliştirilen bir ağ olmuştur. ZF Net olarak adlandırılan bu ağ nesne tanımadaki hata oranını %11.2’ye geriletmiştir. Bu mimari Alex Net’ten esinlenerek oluşturulmuştur. Alex Net’in eksiklerini giderme temeline dayanmaktadır.



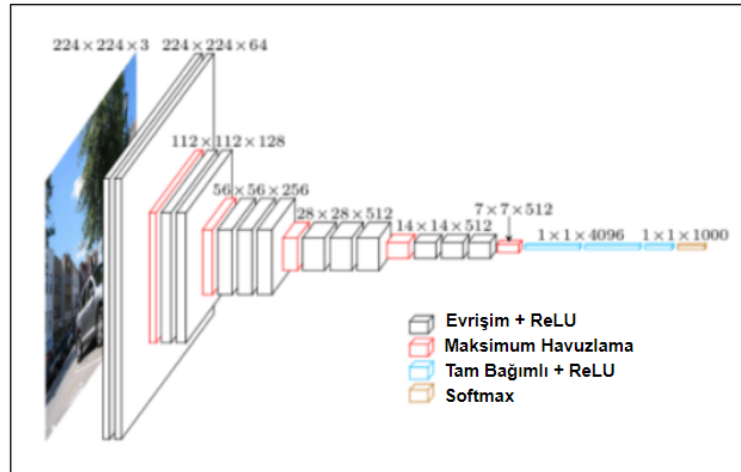
Şekil 3.28. ZF Net modelinin yapısı (Zeiler ve Fergus, 2014)

Şekil 3.28’de ZF Net ağ mimarisi verilmiştir. Bu ağın ana noktalarına bakıldığında Alex Net’te 11x11 olan filtre matrisi boyutunu 7x7 boyutuna düşürmüştür. Bu da giriş katmanındaki birçok orijinal pikselin korunmasına yardımcı olup ağın başarısını arttırmıştır. Ayrıca filtrenin kayma adım sayısı 2 olarak belirlenmiştir. Alex Net, 15

milyon görüntü üzerinde eğitim almıştır, ZF Net ise sadece 1,3 milyon görüntü üzerinde eğitim almıştır. On iki gün boyunca GTX 580 GPU'da eğitilmiştir (Deshpande, 2019). Aktivasyon fonksiyonu olarak ReLU kullanılmış, hata azaltılması için çapraz entropi kayıp fonksiyonu ve ağı eğitimi için ise rastgele eğim azaltma yöntemi kullanılmıştır (Özkan ve Ülker, 2017).

### 3.2.2.3.VGG Net (2014)

Oxford üniversitesi görsel geometri grubu tarafından geliştirilmiştir. Yarışmaya girdiği sene birinci olmamıştır. Ağdaki katman sayısına göre VGG16 ve VGG19 olarak isimlendirilmektedir. Şekil 3.29'da ağ mimarisi verilmiştir. VGG16 toplamda 13 evrişim ve 3 tam bağımlı katmandan oluşmaktadır. Diğer katmanlarla beraber 41 katmandan oluşmaktadır. Bu katmanlar; maksimum havuzlama, tam bağımlı, ReLU ve sınıflandırma katmanlarıdır.  $224 \times 224 \times 3$  boyutunda bir girdi almıştır. %89 başarı oranı elde etmiştir. VGG19, 16 evrişim ve 3 tam bağımlı katmandan oluşmaktadır. VGG19 diğer katmanlarla beraber toplamda 47 katmandan oluşmaktadır. Katman sayısı arttığı için VGG16'ya göre daha iyi başarı elde etmiştir (Doğan ve Türkoğlu, 2018). İki ile üç hafta boyunca 4 Nvidia Titan Black GPU'da eğitilmiştir (Deshpande, 2019).

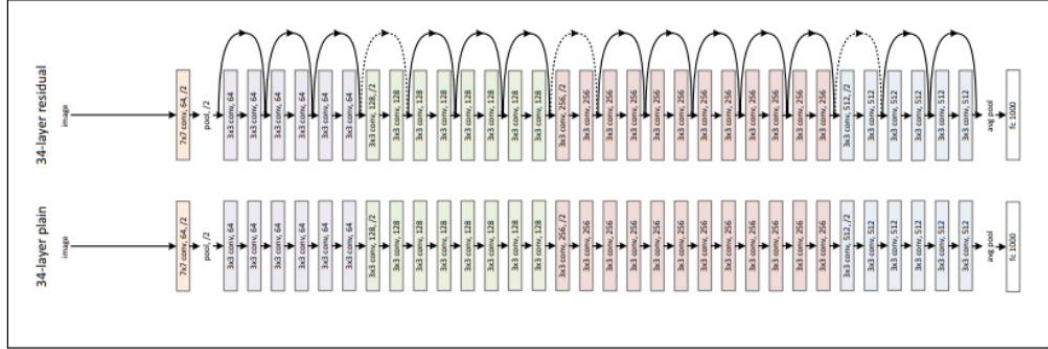


Şekil 3.29. VGG16 ve VGG19 mimarisi (Rosebrock, 2017)

### 3.2.2.4. Microsoft ResNet (2015)

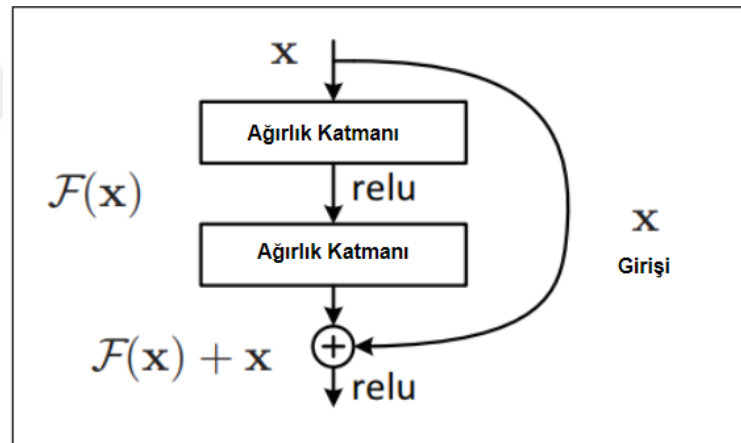
ResNet 2015 yılında hata oranını %3,6'ya düşürerek o yılın birincisi olmuştur. Uzman insanların dahi %5-10 arasında hata yaptığı göz önüne alınırsa bunun ne kadar

yüksek bir başarı olduğu anlaşılır. Bu mimari şimdiye kadar tasarlanan tüm mimarilerden daha derin bir yapıya sahiptir. Toplamda 152 katmandan oluşmaktadır. İki ile üç hafta boyunca 8 GPU makinede eğitilmiştir (Deshpande, 2019).



Şekil 3.30. Microsoft ResNet mimarinin yapısı (He ve ark., 2016)

Şekil 3.30’da ResNet’in ilk 32 katmanı gösterilmiştir.



Şekil 3.31. Residual bloğu (He ve ark., 2016)

Şekil 3.31’de görüldüğü gibi bu mimari residual bloklarından meydana gelmektedir. Bu bloklar  $x$  girişine evrişim, ReLU işlemi sonucu bir  $F(x)$  değeri ortaya çıkar. Bu  $F(x)$  değerine ilk girişte verilen  $x$  değeri eklenerek yeni bir  $H(x) = F(x) + x$  değeri oluşturur.

### 3.3. Derin Yüz Tanıma

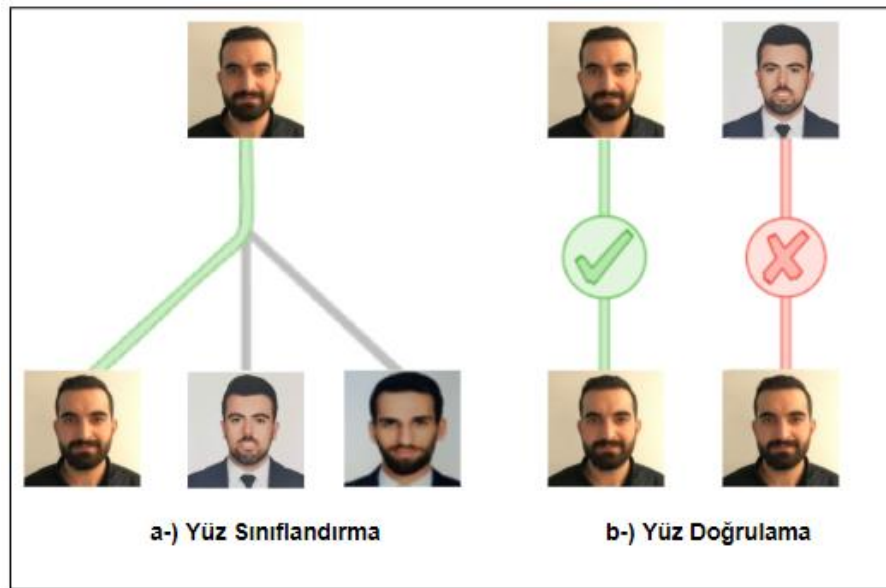
Yüz tanıma, kişiden alınan dijital resimlerden seçilen özneliklerin ön işlenmesiyle, kişinin doğrulanması veya tanınmasına olanak sağlayan bilgisayar tabanlı

uygulamadır. Yüz tanımının performansını artırmak için, diğer kimlik doğrulama sistemleriyle birleştirilmektedir (Taigman ve ark., 2014). Son zamanlarda ESA gibi derin modellerin üst düzey görsel özelliklerin çıkarılmasında etkili olduğu kanıtlanmış, yüz tanıma ve doğrulama alanlarında kullanılmıştır (Sun ve ark., 2014).

Yapılan çalışmada, literatürde sık yöntemler olarak bilinen iyi mühendislik yapılarak özneliklerin elle çıkarılması yöntemi yerine, derin öğrenme modelleri kullanılarak daha yüksek başarı elde edilmiştir. Derin öğrenmenin yüz tanıma alanındaki başarısı insan seviyesine ulaşmıştır (Taigman ve ark., 2014).

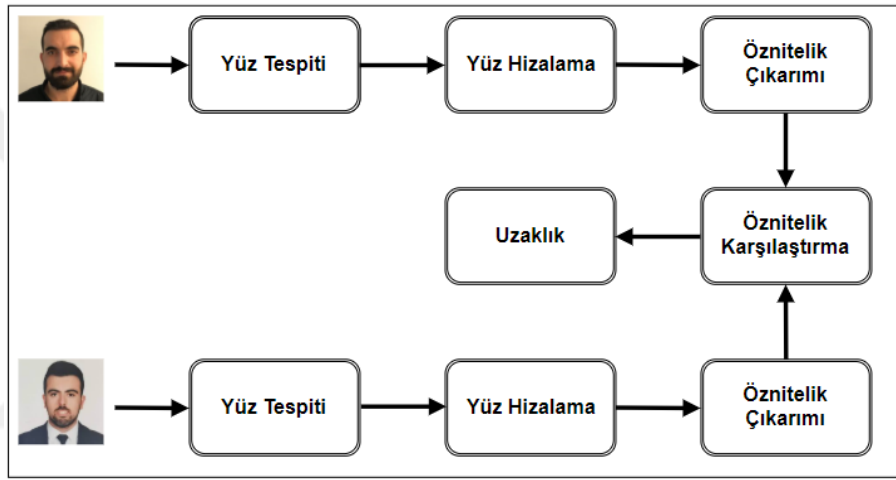
Yüz tanıma iki temel yöntemden oluşmaktadır (Wang ve Deng, 2018). Şekil 3.32'de görüldüğü gibi ilk olarak mevcut bir veri tabanında bulunan yüzler içinde arama yapıp yüz sınıflandırma yapılır. Burada bire çok bir karşılaştırma vardır. Bu yöntemde genellikle kümeleme ve sınıflandırma algoritmaları kullanılır. İkinci yöntem ise yüz doğrulama işlemidir. Girdi olarak verilen yüzün sistemdeki yüz ile karşılaştırılıp, aynı kişiye ait olup olmadığının tespit edilmesinden oluşmaktadır. Bu yöntem bire bir karşılaştırma yapar. Genellikle uzaklık hesaplama yöntemleriyle karşılaştırma yapılır.

Bizim çalışmamızda yüz tanımadan önce kart ve şifre ile doğrulama yapıldığı için sistemde kayıtlı olan kişinin yüzü bilinmektedir. Bu sebepten dolayı sistemde kayıtlı olan müşterinin resmi ile bankamatikten gelen resim karşılaştırılır. Doğrulama bu karşılaştırma sonucunda yapılır.



Şekil 3.32. Yüz sınıflandırma ve doğrulama örnekleri

Yüz tanıma işlemi belli bir işlemler hattından geçtikten sonra sonuçlanır. Bu işlem adımlarının her birisi sistemin başarısını doğrudan etkilemektedir. Tasarlanan sistemin iş hattı Şekil 3.33'te gösterilmiştir. Bu iş hattında yüz problemin temel taşı oluşturulmaktadır. Bundan dolayı ilk olarak verilen resimde yüz tespit edilir. Daha sonra tespit edilen yüz hizalanır ve hizalanan yüz ağı verilip öznitelikler çıkarılır. Bu öznitelikler aynı zamanda kişiye ait özel bir kimliktir. Test verisi için de aynı işlem yapılır. Bu iki resimden elde edilen öznitelikler karşılaştırılır. Karşılaştırma sonucu uzaklıklar elde edilir. Bu uzaklık belirlediğimiz eşik değerinin altında ise yüz doğrulama sağlanır. Eşik değerinin üzerinde ise doğrulama sağlanmaz.



Şekil 3.33. Yüz tanıma işlem adımları

### 3.3.1. Yüz tespiti

Yüz tanıma işleminin ilk aşaması yüz tespittir ve bu nesne tanımanın özel bir şeklidir. Yüz tespiti, dijital resimlerden veya videolardan gelen ön yüz resimlerini tespit eden bilgisayar görü teknolojisidir. Yüz tespiti, burun ve üst dudak sınır çizgisi tespiti ile ön yüz tespiti yapan bilgisayarlar görü teknolojisini kullanır. Burun, yüz sınır çizgilerinin orta noktasıdır (Sharma ve ark., 2016). Yüz tespiti için birçok algoritma geliştirilmiştir. Çalışmamızda diğer yüz tanıma algoritmalarına göre yüksek doğrulukta çalışan çoklu ESA kullanılıp, hem yüz tespiti hem de yüz hizalama işlemi aynı anda yapılmaktadır.

### 3.3.2. Yüz hizalama

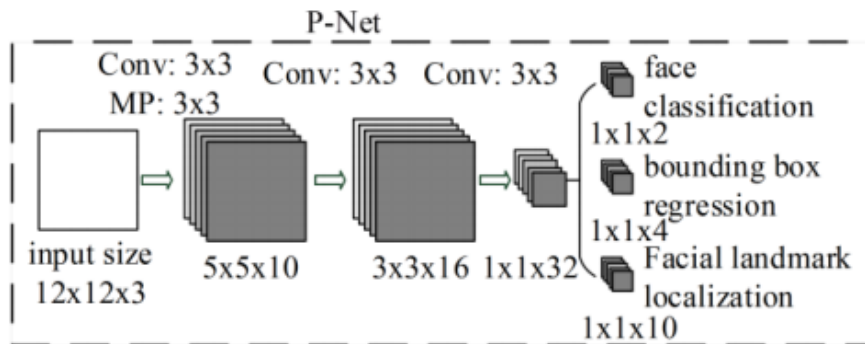
Yüz hizalama işlemi için birçok yöntem mevcuttur. Bizim kullandığımız yöntem çoklu ESA yöntemini kullanmaktadır. Dış ortamlarda yüksek ışık, düşük ışık ve yüzü tespit edilmeye çalışılan kişinin tam kameraya dönmemesi problemlerinden dolayı yüz tespiti ve hizalanması zor bir problemdir. Son araştırmalar derin öğrenme yöntemlerinin bu iki görevde etkileyici performans gösterdiklerini kanıtlamıştır. Yüz tespiti ve yüz hizalama işlemi, çok görevli öğrenmeyle birleştirilmiş basamaklı ESA kullanılarak yapılmaktadır. Bu işlem üç aşamadan oluşmaktadır (Zhang ve ark., 2016).

İlk olarak üç aşamalı ESA'ya başlamadan önce Şekil 3.34'te gösterildiği gibi girişe verilmek üzere çeşitli boyutlarda bir görüntü piramidi oluşturulur.



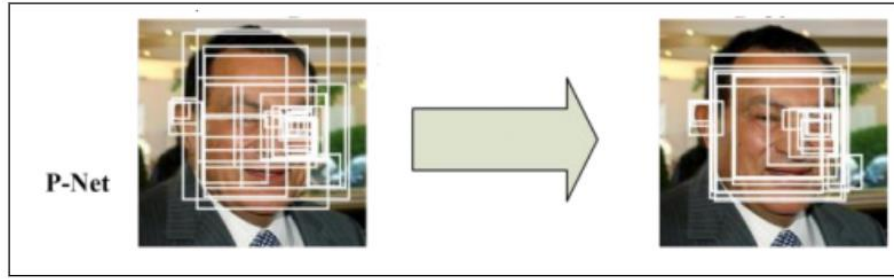
Şekil 3.34. Çoklu evrişim ağının ilk aşamasına verilmek için oluşturulan görüntü piramidi (Zhang ve ark., 2018)

Daha sonra Şekil 3.35'teki yüzeysel bir evrişim ağı olan teklif ağı (P-Net) ile yüzün tespiti için pencereler üretilir.



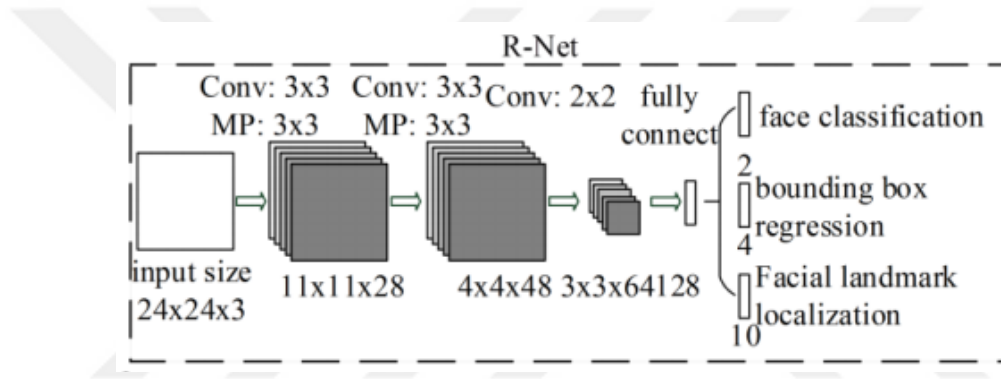
Şekil 3.35. Teklif (P-Net) evrişim ağı (Zhang ve ark., 2016)

Teklif Ağı (P-Net) çıkışında yüze denk gelen birden çok pencere üretilir. Bu pencerelerin hepsi Şekil 3.36'daki gibi yüzü kapsayacak şekildedir.

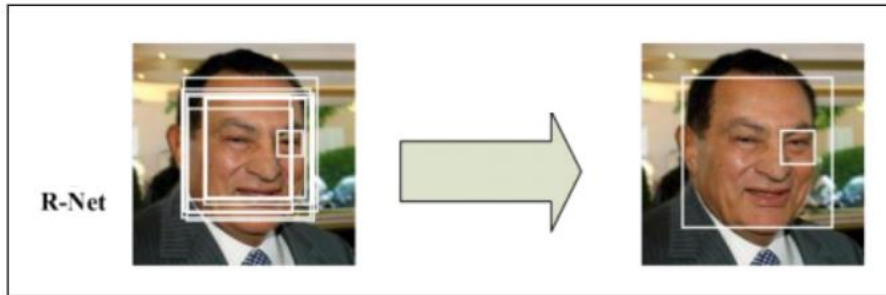


Şekil 3.36. Teklif Ağı (P-Net) sonucunda ortaya çıkan yeni resim (Zhang ve ark., 2016)

Daha sonra Şekil 3.35’te verilen teklif ağından daha karmaşık olan arıtım ağı (R-Net) ile büyük sayıda yüz içermeyen pencereler azaltılır.



Şekil 3.37. Arıtım (R-Net) evrişim ağı (Zhang ve ark., 2016)

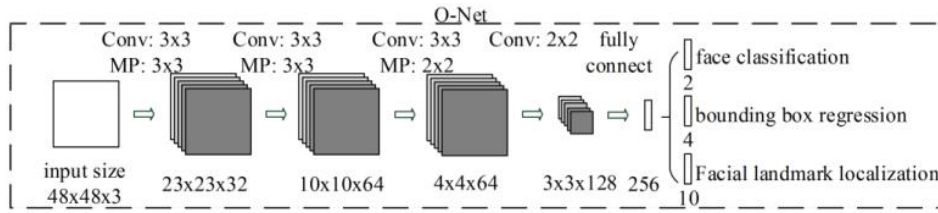


Şekil 3.38. Arıtım ağı (R-Net) sonucunda ortaya çıkan yeni resim (Zhang ve ark., 2016)

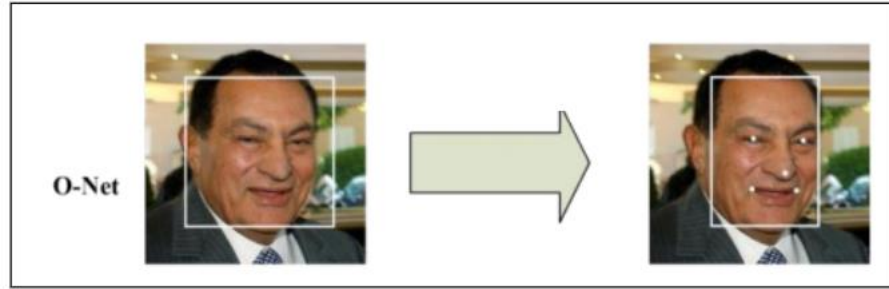
Arıtım ağı (R-net) sonucu Şekil 3.38’de görüldüğü gibi tüm pencereler azaltılmış. Sadece yüzü kapsayan bir aday pencere kalmıştır. Bu adımdan sonra yüzü ifade eden 5 nokta tespit edilip işlem sonlanacaktır.

Son olarak Şekil 3.37’de gösterilen R-Net’ten daha güçlü bir evrişim ağı olan Şekil 3.39’daki çıkış ağı (O-net) ile yüz daha ayrıntılı olarak açıklanmaktadır. Burada Şekil 3.40’taki gibi yüzün beş temel yüz merkezi tespit edilir.





Şekil 3.39. Çıkış (O-Net) evrişim ağı (Zhang ve ark., 2016)



Şekil 3.40. Çıkış ağı (O-Net) ile yüzün beş temel noktasının tespiti (Zhang ve ark., 2016)

Kullanılan üç ağda da yüz tespiti için ağın eğitimi üç aşamaya bağlıdır. Bunlar sırasıyla yüzün sınıflandırılması (face classification), sınırlayıcı kutu regresyonu (bounding box regression) ve yüz noktalarının tespiti aşamalarıdır. Bu üç işlemin sonucuna göre ağ, geri yayılım ile güncellenir. Yüzün sınıflandırılma aşamasında tespit edilen karede yüzün olup olmasına göre geri yayılım yapar. Sınırlayıcı kutu regresyonu her aday pencere ile kendisine en yakın pencere arasındaki öklid uzaklığının tahmin edilmesi ve yüz noktalarının tespiti işlemlerine göre geri yayılım yapar. Ayrıca sınırlayıcı kutu regresyonu resimden nesne tespitinde kullanılır. Burada yüz bizim için bir nesne olduğu için kullanılmıştır (Zhang ve ark., 2016).

### 3.3.3. Özellik çıkarımı

Özellik çıkarımı için derin öğrenme modelleri kullanılmıştır. Derin modeller her katmanda yüze ait benzersiz bir özellik tespit eder ve son katmana doğru kişiye özel özneliklerin ortaya çıkmasını sağlamaktadır. Yüz tanıma alanında birçok çalışma mevcuttur. Google, Facebook gibi dünya devi firmalar bu alanda çeşitli çalışmalar yapmıştır. Biz Facebook'un DeepFace ve Google'ın FaceNet isimli mimarilerini bu çalışma kapsamında inceleyeceğiz ve kendi uygulamamızda kullandığımız FaceNet mimarisinin avantajlarını açıklayacağız.



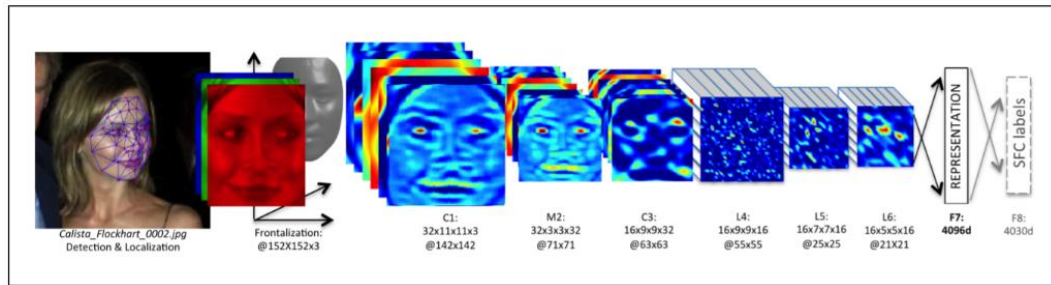
Çizelge 3.2’de bu iki mimaride kullanılan eğitim verisinin miktarı ve başarı oranları verilmiştir (Parkhi ve ark., 2015).

**Çizelge 3.2.** LFW veri seti üzerinde DeepFace ve FaceNet mimarilerinin karşılaştırılması (Parkhi ve ark., 2015)

No	Mimari	Resim	Başarı(%)
1	DeepFace	4 Milyon	97.35
2	FaceNet	200 Milyon	99.63

### 3.3.3.1. DeepFace

DeepFace mimarisi 2014 yılında Facebook araştırmacıları tarafından tasarlanmış, önceki yüz tanıma yöntemleri ve modellerine göre yüksek başarı elde edilmiştir. Bu modelde diğer modellerden farklı olarak işlenmiş veri setleri yerine ham resimler kullanılmıştır. Bu derin yüz mimari 9 katmanlı bir ağdan oluşmaktadır. Bu ağ 120 milyondan fazla parametre içermektedir. Eğitimi 4 milyondan fazla etiketlenmiş, 4.000’den fazla kimliğe sahip yüz resmi içeren veri seti ile yapılmıştır. 3D yüz hizalama kullanılmıştır. YouTube, yüz veri setinde hata oranını %50’den fazla düşürmektedir.



**Şekil 3.41.** DeepFace mimarisi (Taigman ve ark., 2014)

DeepFace, yüz resminin kimliğini belirlemek için Şekil 3.41’de gösterilen mimariyi kullanmıştır. 3D hizalanmış 3 katmanlı (RGB) yüz resmi 152 x 152 piksel olarak ESA katmanının girişine verilmiştir ve buna C1 katmanı ismi verilmiştir. C1 katmanı 32 filtreden oluşmaktadır. Her filtrenin boyutu 11x11’dir. RGB katmanlarından dolayı bu boyut 11x11x3 olmuştur. Bunun gösterim şekli 32x11x11x3 @ 152 x152 şeklindedir. Sonraki katman olan maksimum havuzlama katmanı 32 öznitelik haritası beslemek için kullanılmıştır. Maksimum havuzlama katmanı 3x3 matris şeklinde

uygulanır ve adım sayısı ikidir. Sonraki C3 isimli ESA katmanı 9x9x16 boyutunda 16 filtre kullanır. C3 katmanının amacı, basit yapı ve kenarlar gibi düşük seviyeli öznitelikleri çıkarmaktır. Maksimum havuzlama katmanları, evrişim ağlarının çıkışlarını yerel dönüşümler için daha güçlü yapar. Bununla beraber havuzlama katmanının birçok seviyesi, mikro yapılar ve detaylı yüzün kesin pozisyonu hakkında bilgiyi kaybedebilir. Dolayısıyla maksimum havuzlama, sadece ilk evrişim katmanına uygulanmıştır. Bu ilk katman ön işlemeli adım olarak yorumlanır. Takip eden L4, L5 ve L6 katmanları yerel bağlıdır (tam bağımlı değildir) ve evrişim katmanı gibi filtre seti uygulamalar ancak öznitelik haritasının her yerinde farklı filtreler öğrenir. Örneğin; göz ve kaşlar arasındaki alanlar çok farklı görünüşler sergiler ve burunla ağız arasındakinden daha fazla ayırım yeteneğine sahiptir. Başka bir deyişle giriş resimlerinin hizalanmış olmasının gücüyle DeepFace mimarisi düzenlenmiştir. Yerel katmanların kullanımı öznitelik çıkarımına etki etmez, sadece eğitim için kullanılan parametre sayısını etkiler. DeepFace büyük etiketli veri setine sahip olduğundan, 3 büyük yerel bağlı katman kullanılmıştır. Sonuç olarak; iki üst katman F7 ve F8 tam bağlıdır, her çıkış birimi her girişe bağlıdır. Bu katmanlar, yüz resimlerinin uzak parçalarında yakalanan öznitelikler arası korelasyonu yakalayabilir. Son tam bağlı katmanın çıkışı K-yollu softmax'ı (K sınıf sayısı) beslemiştir. Bu sınıf yüz etiketleri üzerinde bir dağılım üretir.

DeepFace mimarisinin eğitilmesinin amacı, doğru sınıfın bulunma olasılığını maksimize etmektir. Bunu her eğitim örneği için çapraz entropi ile kayıp fonksiyonunu minimize ederek yapar. Girişe verilen yüz, beklenen çıktı ile farklıysa rastgele eğim azaltımı kullanılarak parametreler güncellenir. Parametrelerin güncellenmesi standart geri yayılım algoritması ile hesaplanmıştır. Bu adım 4000 benzersiz kimlik içeren 4 milyon resim üzerinde denenmiştir. Eğitimin amacı aynı kişiye ait olan resimler arasındaki uzaklığı en aza indirmektir.

DeepFace test aşamasında verilen iki yüzün aynı kişiye ait olduğunu tespit etmek için doğrulama metriklerini kullanmaktadır. DeepFace siyam ağlarını doğrulama metriği olarak kullanır. Aynı ağa verilen iki yüz görüntüsünün sonunda çıkan öznitelik vektörlerinin arasındaki uzaklığı hesaplar ve buna bir eşik uygular. Bu eşik değerinin altında kalan resimler aynı kişiye aittir sonucunu ortaya koyar. Eşik değerinden büyük olanlara ise farklı kişiye aittir denir.

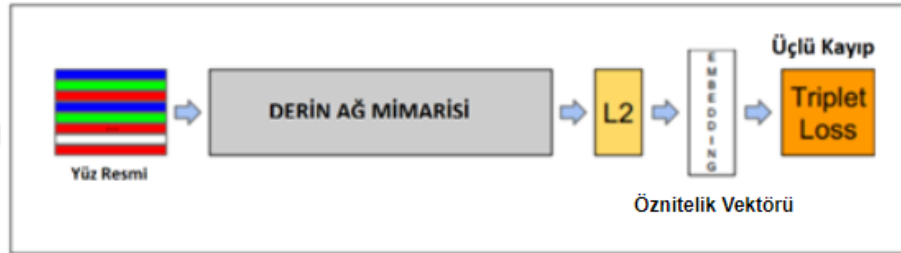
İki yüz resmi arasındaki uzaklık Denklem 3.22'deki gibi hesaplanmaktadır.

$$d(f_1, f_2) = \sum_i a_i |f_1[i] - f_2[i]| \quad (3.22)$$

Denklem 3.22’de gösterilen  $f_1$  ve  $f_2$  DeepFace ağından geçirilen iki yüz resmini vektörel olarak temsil eder.  $d(f_1, f_2)$  ise iki yüz arasındaki uzaklığı hesaplar (Taigman ve ark., 2014).

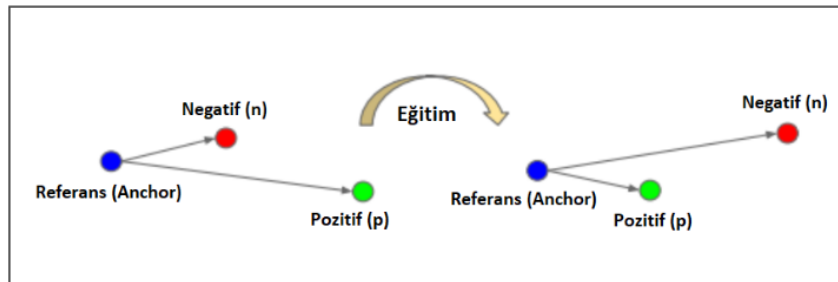
### 3.3.3.2. FaceNet

FaceNet, Google arařtırmacıları tarafından geliştirilen derin yüz tanıma modelidir. Tez kapmasında bu model kullanılmıştır. Bu model kendinden önceki tüm modellerden daha başarılı olmuştur. Bu başarının temel nedeni eğitim aşamasında kullanılan üçlü kayıp (triple loss) yöntemidir. Bu yöntem kendisiyle aynı resmi olan kişileri yakınlılařtırırken kendisinden farklı olanları da uzaklařtırmaktadır. Bu model LFW veri setinde %99.63 doğruluk, YTB veri setinde ise %95.12 doğruluk sağlamıştır. Burada kullanılan ađ modeli Inception ađ modelidir. Bu model 2014 ImageNet’te birinci olmuştur.



Şekil 3.42. FaceNet mimarisi (Schroff ve ark., 2015)

Şekil 3.42’de görüldüğü gibi girişe verilen yüz resmi, derin ađdan geçirilip embedding dimensionality (gömme boyutluluđu) denilen 128 boyutunda bir vektör oluşur. Bu vektör kişiye özel olan yüz kimliğini verir. Mobil cihazlarda çalıştırmak için çıkış vektörünün boyutu daha da azaltılabilir ancak az doğruluk kaybı yaşanabilir.



Şekil 3.43. Üçlü kayıp örneđi (Schroff ve ark., 2015)

Üçlü kayıp, eğitim aşamasında test verisinde bulunan bir kişinin bir resmini referans resim olarak alır. Bu resme anchor denilmektedir. Aynı kişiye ait bir resim daha alınarak karşılaştırma için kullanılır. Aynı kişiye ait olan resme pozitif resim denir. Ek olarak referans olarak alınan kişiden farklı bir kişinin resmi alınır buna da negatif resim denir. Referans resim ile pozitif resim derin ağdan geçirilir ve oluşan 128 boyutunda emmbedding vektörleri karşılaştırılır. Aralarındaki uzaklık hesaplanır. Aynı şekilde negatif resim de derin ağdan geçirilir ve referans resim ile arasındaki uzaklık hesaplanır. Eğer referans resim ile negatif resim arasındaki uzaklık, pozitif resim ile arasındaki uzaklıktan küçükse ağ yanlış çalışıyor demektir. Ağ geri yayılım ile parametreleri günceller ve bu işlem negatif resim uzaklığı pozitif resimden büyük olana kadar devam eder. Belli bir denemeden sonra referans resmin negatif ile pozitif arasındaki uzaklığı aynı olabilir. Bu durumda ağın eğitimi bir dar boğaza girmiş olacaktır. Bu dar boğazı aşmak için pozitif uzaklığa  $a$  denilen bir değer eklenir ve bu değer sayesinde bu dar boğaz aşılmış olacaktır. Şekil 3.43'te pozitif ve negatif resimlerin referans resme olan uzaklıkları eğitimden önce ve sonra verilmiştir. Eğitimden sonra pozitif resim referans resmine yaklaşırken aynı zamanda negatif resim referans resimden uzaklaşmaktadır. Bu da ağın başarısını arttırmaktadır.

Üçlü kayıp yöntemi ile ağın eğitilmesinde kullanılan geri yayılım Denklem 3.23'teki gibi hesaplanır:

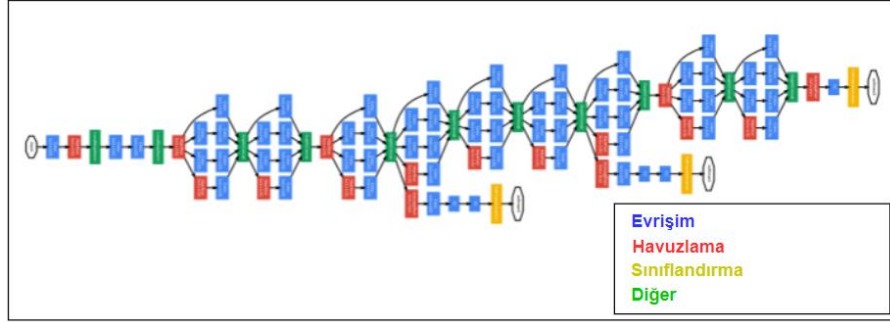
$$\|f(x_i^a) - f(x_i^p)\|^2 + a < \|f(x_i^a) - f(x_i^n)\|^2 \quad (3.23)$$

Denklem 3.23'teki  $f(x_i^a)$  değeri referans resmin vektörel olarak çıktısını,  $f(x_i^p)$  pozitif resmin vektörel olarak çıktısını,  $f(x_i^n)$  değeri negatif resmin vektörel olarak çıktısını ifade eder (Schroff ve ark., 2015).

### 3.3.4. GoogLeNet (Inception V1) modeli

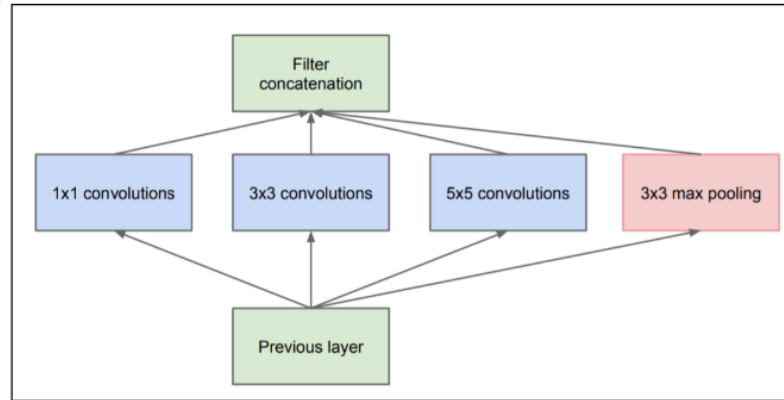
GoogLeNet'in en önemli özelliği, evrişim katmanlarının her zaman sıralı olarak dizilmesi gerekmediğini ortaya atan ilk modeldir. Ağ 22 katmandan oluşan bir ESA modelidir. Havuzlama katmanları da sayılırsa bu 27 katmana çıkmaktadır (Szegedy ve ark., 2015). Toplamda 100 katman ve 9 Inception modülünden oluşmaktadır. ImageNet yarışmasında 2014 yılında hata oranını %6.7 düşürmesiyle birinci olmuştur. Yapısında

Inception modülleri bulunduğu için karmaşıktır. Bütün katmanları art arda ekleyip hepsine ayrı ayrı filtre uygulamak yerine, katmanları paralel olarak kullanarak hem bellek ihtiyacını düşürmüş hem de ağı ezberlemesini zorlaştırmıştır. GoogLeNet mimarisinin genel yapısı Şekil 3.44’te verilmektedir (Deshpande, 2019).



Şekil 3.44. GoogLeNet mimarisinin yapısı (Deshpande, 2019)

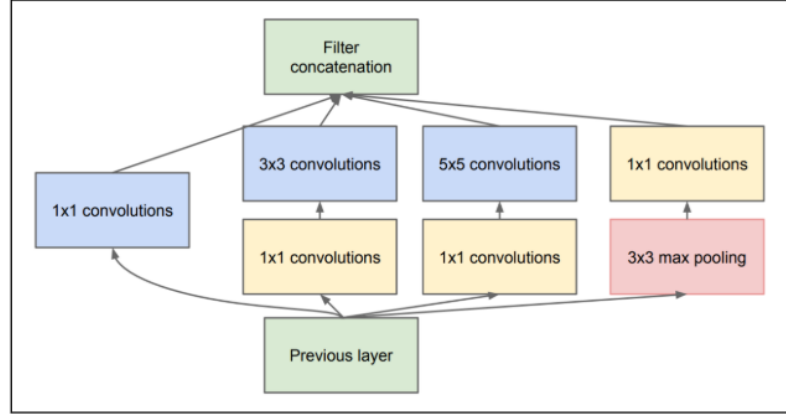
Paralel olarak kullanılan katmanlara Inception denilmektedir. Bir Inception modülü detaylı olarak Şekil 3.45’te verilmiştir. Burada 1x1, 3x3, 5x5 evrişim katmanı ve 3x3 ortaklama katmanı paralel olarak kullanılmıştır.



Şekil 3.45. Bir Inception detayı (Szegedy ve ark., 2015)

Genel olarak, ağı performansını artırmanın en doğrudan yolu ağı derinliğini ve genişliğini artırmaktır. Bu durum çok sayıda parametre üretilmesi anlamına gelir. Bununla birlikte, birçok parametre yalnızca fazla uyarıya neden olmakla kalmayacak, aynı zamanda hesaplamayı da artıracaktır (Szegedy ve ark., 2015). Her bir Inception modülünde paralel yürütülen işlemler yüzünden ağı çıkış boyutu ve parametre sayısı çok artmaktadır. Bu problemin çözümü için Şekil 3.46’da görülen 1x1 boyutunda bir evrişim

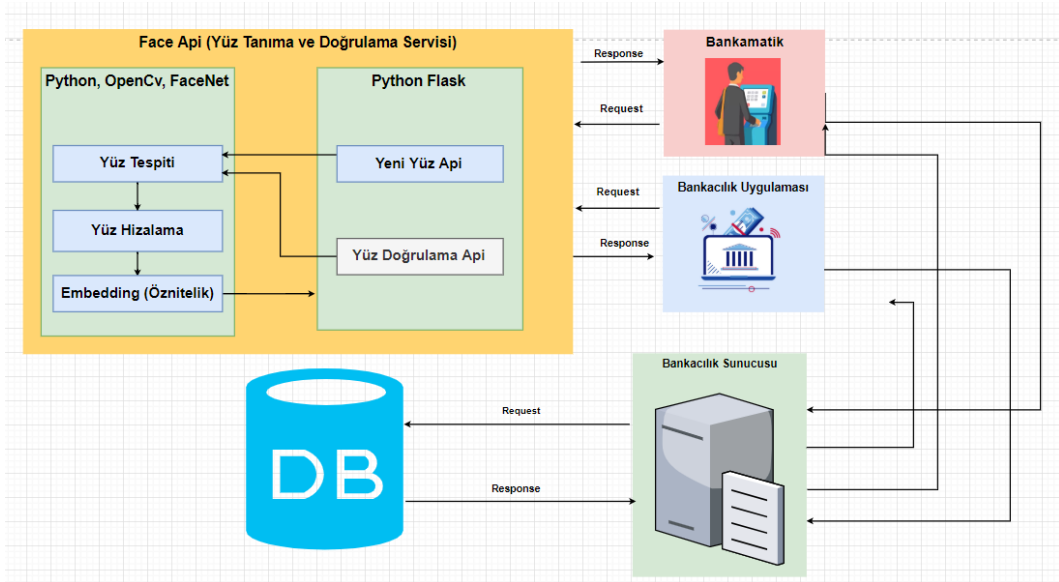
katmanı mevcut evrişim katmanlarından öncesine eklenerek boyut azaltılmaktadır. Bu işlemler sonucu oluşan yeni ağa Inception V1 adı verilmiştir.



Şekil 3.46. Bir Inception V1 detayı (Szegedy ve ark., 2015)

#### 4. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Bu tez çalışmasında derin öğrenme kullanılarak sahtekârlık tespiti işlemi, birbiri ile bağlantılı olarak çalışan 3 uygulamadan oluşmaktadır. Bunlar yüz servisi, bankacılık uygulaması ve bankamatik uygulamasıdır. Uygulamanın mimarisi Şekil 4.1’de gösterilmiştir. Yüz servisi gelen resmin özniteliklerini bulur ve istekte bulunan uygulamaya iletir. Ek olarak öznitelik listesini ve doğrulama yapılacak resimlerin listesini alarak bunları karşılaştırır ve aradaki uzaklık mesafelerinin listesini istek yapan uygulamaya gönderir. İkinci uygulama bankacılık uygulamasıdır. Burada müşteriden alınan resimler yüz servisine gönderilir, yüz servisinden dönen öznitelik listesini müşterinin diğer bilgileri ile birlikte veri tabanına kaydeder. Uygulamamızın son kısmı bankamatik uygulamasıdır. Bu uygulama asıl doğrulamanın yapıldığı yerdir. Bankacılık uygulamasından alınan resimlerin öznitelikleri, bankamatiklerden alınan resimler ile beraber yüz doğrulama servisine gönderilir ve geri dönüş değeri olarak aradaki uzaklıkları alır. Bu uzaklık listesinde bulunan herhangi bir değer eşik değerin altında kalırsa doğrulama işlemi sağlanır. Bütün değerler eşik değerinin üstünde ise doğrulama sağlanmaz.



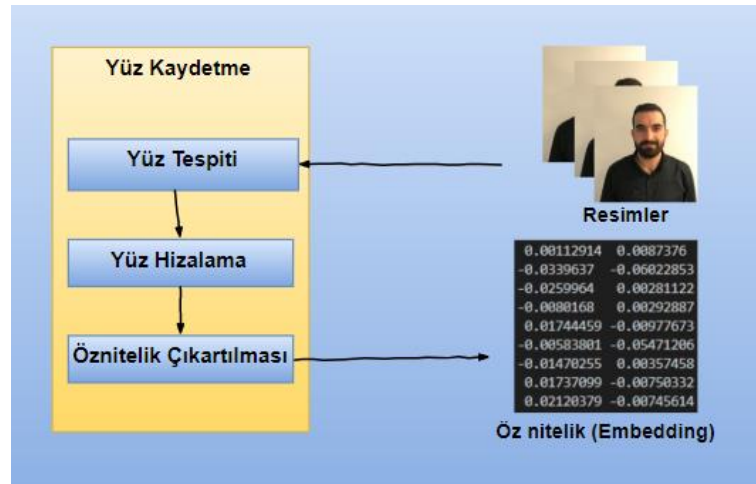
Şekil 4.1. Derin öğrenme kullanılarak sahtekârlık tespiti uygulama mimarisi

#### 4.1. Yüz Servisi

Yüz servisi iki katmandan oluşmaktadır. Bunlardan bincisi yüz işleminin yapıldığı katmandır. İkinci katman ise bunun dış uygulamalar ile haberleşmesini sağlayan katmandır.

İlk katman temel işin yapıldığı katmandır. Burada yüz kaydetme ve yüz doğrulama olmak üzere iki işlem yapılır. İlk olarak gelen resimlerden yüz tespit edilir hizalanır. Bu işlem çoklu ESA ile bulunur. Tespit edilen yüzler daha önceden eğitilmiş modelden teker teker geçirilerek öznitelikler bulunur. Aynı şekilde karşılaştırılma işlemi için de gönderilen tüm resimler, eğitilmiş modelden geçirilir ve öznitelikleri çıkarılır. Ek olarak iletilen öznitelik vektörü ile öznitelikleri çıkarılan resimlerle aralarındaki uzaklıklar teker teker hesaplanır. Bu işlemler python dili aracılığıyla tensorflow, opencv kütüphaneleri kullanılarak yapılmıştır.

İkinci katman, ilk katmandan alınan sonuçların dış uygulamalara iletilmesini sağlayan bir web servistir. Burada Python Flask kütüphanesi kullanılmıştır. İki kısımdan oluşmaktadır. İlk kısım yüz kaydetme işlemini yapar. Şekil 4.2’de görüldüğü gibi liste olarak aldığı resimleri ilk katmana gönderir. İlk katman sırasıyla yüz tespiti, yüz hizalama ve öznitelikleri hesaplar. Hesaplanan öznitelikleri liste halinde istekte bulunan uygulamaya iletir.



Şekil 4.2. Yüz servisi yüz kaydetme işlemi

Şekil 4.3’te örnek bir web servis isteği gösterilmektedir. Burada base64’te çevrilen iki resim alınmaktadır. Şekil 4.4’teki gibi bir öznitelik listesini geri döndürür.



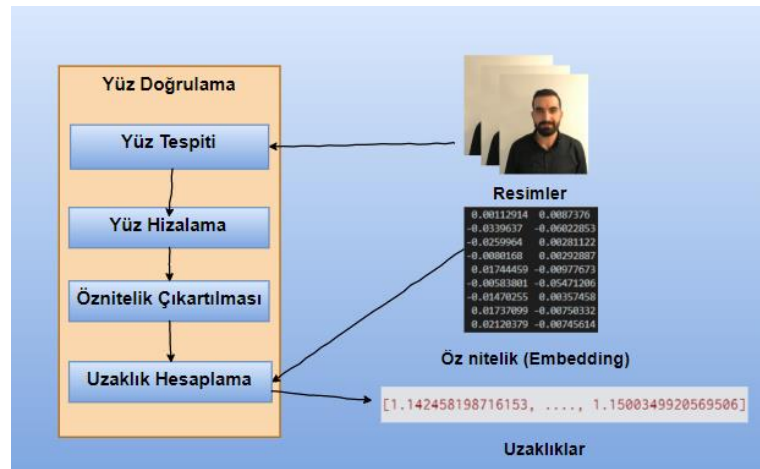
```
{
  "image":["<base64Image>","<base64Image>"]
}
```

Şekil 4.3. Yüz servisi yüz öznitelik çıkarma istek örneği

```
{
  "results":[[
    -0.07120747119188309,
    -0.06242715194821358,
    -0.05528175085783005,
    0.007519483100622892,.....,
    0.0699707418680191,
    0.01730312407016754],[
    -0.07120747119188309,
    -0.06242715194821358,
    -0.05528175085783005,
    0.007519483100622892,.....,
    0.0699707418680191,
    0.01730312407016754]]
}
```

Şekil 4.4. Yüz servisi yüz öznitelik çıkarma cevap örneği

İkinci kısım yüz doğrulama işleminin yapıldığı kısımdır. Daha önce alınıp kaydedilen bir öznitelik listesi ile bir resim listesini alıp ilk katmana iletir. Şekil 4.5'te görüldüğü gibi ilk önce gelen resimler teker teker yüz tespiti, hizalama ve öznitelik çıkarma işlemlerinden geçirilir. Öznitelikleri çıkarılan ve servise gönderilen öznitelikler arasındaki uzaklık hesaplanır ve istek yapan uygulamaya iletilir.



Şekil 4.5. Yüz servisi yüz doğrulama adımları

```
{
  "image": "[<base64Image>, <base64Image>]",
  "emb": "<embedding>"
}
```

**Şekil 4.6.** Yüz doğrulama metodunun örnek bir isteği

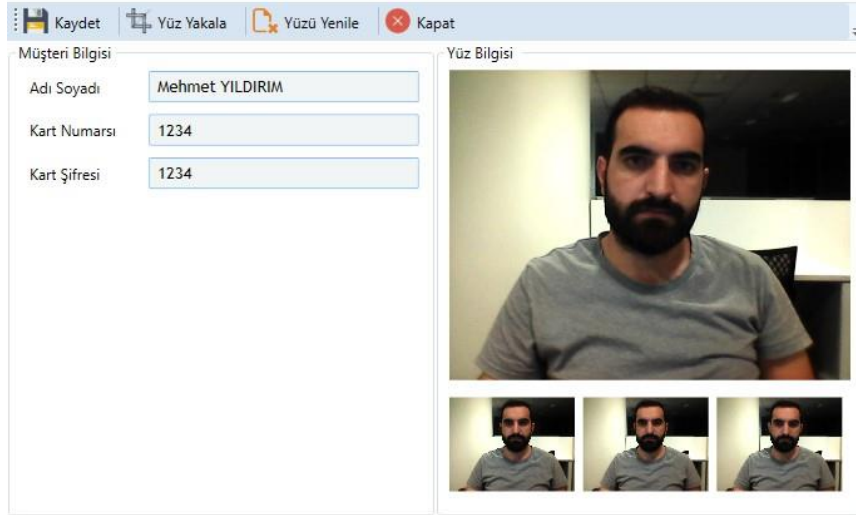
Şekil 4.6’da yüz doğrulama istek örneği görülmektedir. Burada resim listesi (image) ve öznitelik listesi (emb) alınarak ilk katmana iletilir. Bunun sonucunda Şekil 4.7’deki gibi bir uzaklık listesi istek yapan uygulamaya gönderilir.

```
{
  "results": "[1.142458198716153, ..., 1.1500349920569506]"
}
```

**Şekil 4.7.** Yüz doğrulama metodunun örnek bir dönüşü

## 4.2. Bankacılık Uygulaması

Bankacılık uygulaması, bankamatik işlemlerinden önce müşterinin bilgilerinin alınıp sisteme kaydedilmesi için kullanılmaktadır. Şekil 4.8’de görüldüğü gibi bankacılık uygulamasından müşteri adı, kart numarası, PIN doğrulama için şifre ve müşterinin yüz resimlerinin alınıp sisteme kaydedilmesinden oluşmaktadır. Açılan kamera yardımı ile şube personeli tarafından müşterinin üç adet resmi kırpılır diğer bilgiler ile beraber kaydedilir. Kaydetme esnasında müşteriden alınan resimler yüz kaydetme servisine gönderilir. Servis yüzleri derin modelden geçirdikten sonra Şekil 4.9’daki gibi bir öznitelik (embedding) listesi döner. Servisten dönen bu öznitelik vektörü ve müşterinin diğer bilgileri Şekil 4.10’da görüldüğü gibi veri tabanına kaydedilir.



Şekil 4.8. Müşteri bilgileri ve yüz görüntülerinin alındığı ekran

0.00112914	0.0087376	0.06622613	-0.04806245	0.03955342	0.02813745
-0.0339637	-0.06022853	0.05370723	0.00994615	-0.03089539	0.05745071
-0.0259964	0.00281122	-0.06617335	-0.04368978	0.01229524	0.03491386
-0.0080168	0.00292887	-0.01191895	0.00542836	0.03810421	-0.02186024
0.01744459	-0.00977673	-0.00869712	-0.05646815	0.0390574	0.03221302
-0.00583801	-0.05471206	-0.05386182	0.03950235	-0.01191168	0.02786443
-0.01470255	0.00357458	0.03805452	-0.00134095	0.01086083	0.00739649
0.01737099	-0.00750332	-0.06865687	-0.03055882	-0.00417906	-0.04332313
0.02120379	-0.00745614	-0.01293921	-0.06862825	0.03731135	0.04467856
-0.02182399	0.00651766				

Şekil 4.9. Bir resmin derin modelden geçirildikten sonra elde edilen öznelik vektörü

	CustomerId	CustomerName	CardNumber	Password
1	28	M.emin Eroğlu	1234	1234
2	29	Mehmet YILDIRIM	1234	1234

	CustomerFaceId	CustomerId	FaceId
1	31	28	[0.0190811175853014, -0.05680757015943527, -0.0023510416503995657, 0.029326949268579483, 0.037...
2	32	28	[-0.01766585186123848, -0.01972401887178421, -0.0038647635374218225, 0.021749528124928474, 0.02...
3	33	28	[-0.026463976129889488, 0.012973517179489136, 0.021492093801498413, 0.018448013812303543, 0.03...
4	34	29	[-0.04574408754706383, -0.032056037336587906, -0.07362884283065796, -0.00719598669991493, 0.04...
5	35	29	[-0.05421755090355873, -0.0405709482729435, -0.08570580184459686, -0.0038877506740391254, 0.049...
6	36	29	[-0.05480533838272095, -0.056458696722984314, -0.07607481628656387, -0.00017580929852556437, 0.0...

Şekil 4.10. Veri tabanı yapısı

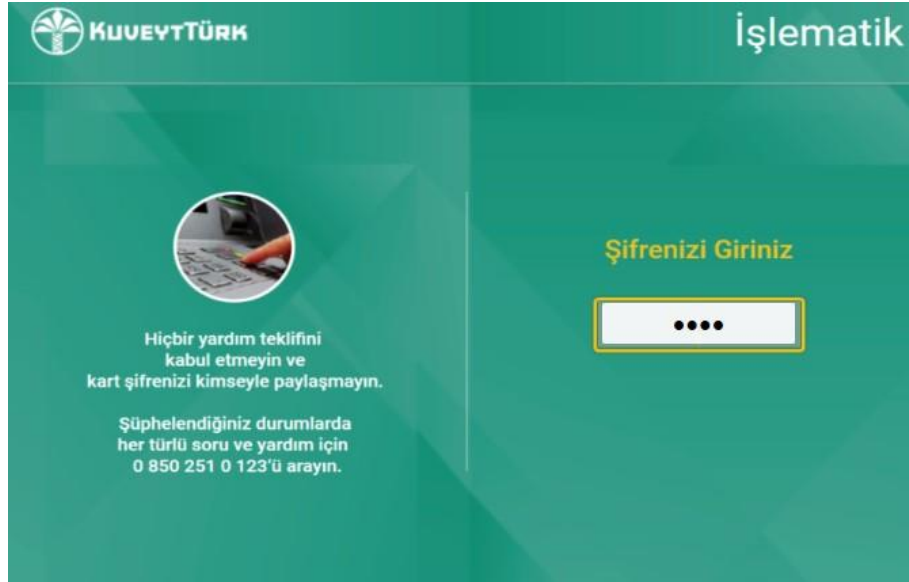
### 4.3. Bankamatik Uygulaması

Bankamatik uygulaması standart bir bankamatik doğrulama süreci oluşturulmak için kullanılmıştır. Üç aşamalı doğrulamadan sonra müşteri işlem yapacağı ekrana ulaşabilmektedir. Şekil 4.11’de ilk olarak manyetik kart doğrulaması için ilk ekrandan müşteri seçimi yapılmaktadır. Bu adım bizim uygulamamızın ilk doğrulama adımını oluşturmaktadır. Bu ekrandaki temel amaç manyetik kart doğrulamayı göstermektir.



Şekil 4.11. Manyetik kart simülasyon ekranı ve doğrulamayı başlatan ekran

Manyetik kart doğrulamadan sonra Şekil 4.12'deki şifre doğrulama ekranı açılmaktadır. Bu adımda dört haneli PIN girişinin olduğu ikinci ekran gelmektedir. Bu ekran bizim senaryomuzun ikinci doğrulama adımını oluşturmaktadır. Burada müşteri hesap açarken belirlemiş olduğu PIN'i girmektedir.



Şekil 4.12. Şifre doğrulama ekranı

PIN doğrulamasından sonra bu çalışma kapsamında önermiş olduğumuz Şekil 4.13'teki üçüncü doğrulama adımını gelmektedir. Bu doğrulama adımında açılan kamera

penceresi yardımıyla 15sn'de bir kare alınacak şekilde 8 kare alınır. Toplam 120 sn kamera penceresi açık kaldıktan sonra alınan yüzler yüz doğrulama servisine gönderilir.



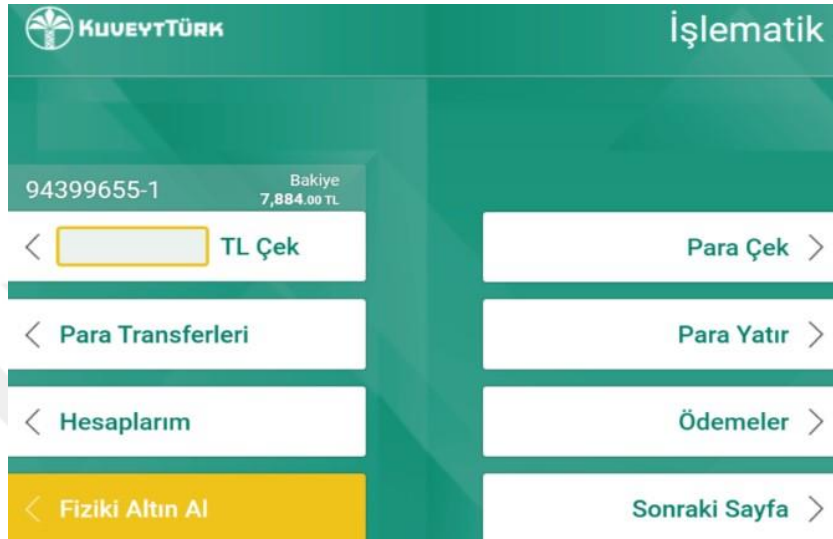
Şekil 4.13. Yüz örneklerinin alındığı ekran

Doğrulama süresinde Şekil 4.14'teki bekleme ekranı açılır. Bu ekranın bekleme süresi boyunca alınan yüz örneklerini derin modelden geçirerek öznitelikleri (embedding) alır. Tüm resimlerin öznitelik vektörleri alındıktan sonra müşterinin veri tabanındaki kayıtlı öznitelikleri ile aralarındaki uzaklık hesaplanıp servisten dönüş yapılır.



Şekil 4.14. Yüz doğrulamanın yapıldığı bekleme ekranı

Yüz doğrulama servisinden dönen uzaklıklardan herhangi birisi eşik değerin altında kalırsa yüz doğrulama başarılı sonlanır. Yüz doğrulama sonucunun başarılı olması ile beraber müşteri Şekil 4.15'teki işlem ekranına yönlendirilir ve üç adımlı bankamatik doğrulaması tamamlanmış olur.



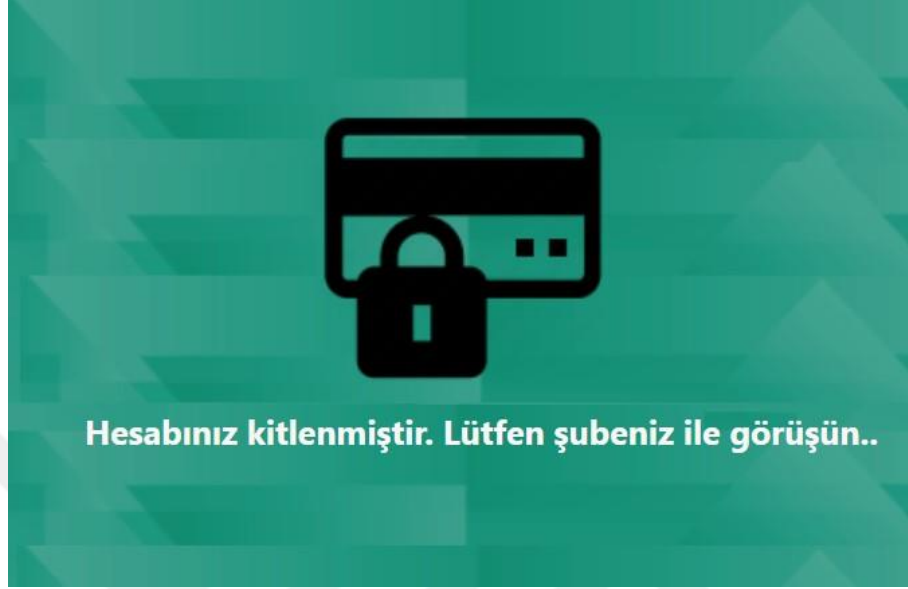
Şekil 4.15. Yüz doğrulaması sonucunda açılan işlem ekranı

Şekil 4.16'da görüldüğü gibi hesap sahibinden farklı birisi bankamatik işlemi yapmaya çalıştığında, iki defa yüz doğrulama ekranı açılıp yüzler alınmaktadır. İlk yüz doğrulama denemesinden sonra ikinci defa tekrar yüz doğrulama ekranı açılmaktadır.



Şekil 4.16. Hesap sahibinden farklı bir kişinin yüz doğrulama adımını geçme denemesi

İki deneme sonucu yüz doğrulama sağlamazsa Şekil 4.17'deki gibi hesap kilit ekranı açılır ve hesabına kısıt konulur. Müşteri kendi şubesine gidip hesabındaki kilidi kaldırmadığı sürece bankamatikten işlem yapamamaktadır.



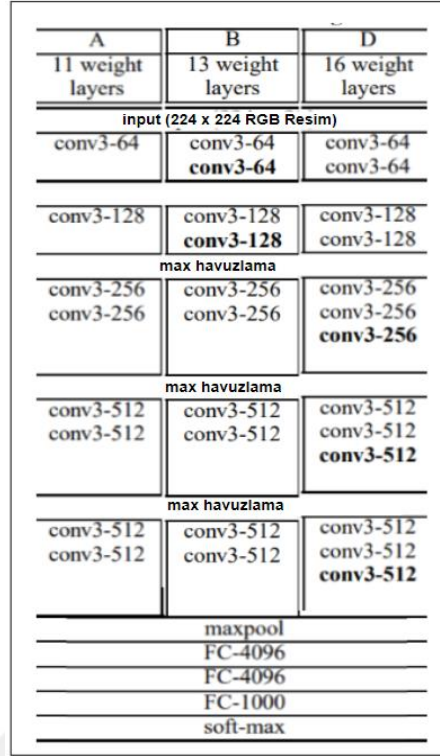
Şekil 4.17. Hesap kilitlenme ekranı

#### 4.4. Kullanılan Modelin Değerlendirilmesi

Yüz tanıma sistemlerinde, derin öğrenme kullanılmadan önce öznitelikler elle çıkartılıp değerlendirme yapılmaktaydı. Yüz tanımadaki başarı elle bulunan bu özniteliklerin doğru tespit edilmesine bağlıydı. Yüz tanıma sistemlerinde derin öğrenmenin kullanılmasıyla beraber bu öznitelikler derin öğrenme tarafından bulunmaya başlandı. Derin öğrenmenin başarısı ise tasarlanan derin modelin büyüklüğüne, yüzün ön işleme adımlarına, eğitim verisinin büyüklüğüne ve eğitim yöntemlerine bağlıdır.

Çizelge 4.1'de ağ mimarisinin, veri setinin, yüzün hizalanmasının ve üçlü kayıp ile ağın eğitilmesinin yüz tanıma sistemlerinin başarısına etkisi gösterilmektedir. Bu çizelgeye göre yüz tanıma başarısının doğrudan bağlı olduğu etkenler açıklanmaktadır.





**Şekil 4.18.** Yüz tanıma sistemlerine ağ mimarisinin etkisini göstermek için kullanılan mimariler (Simonyan ve Zisserman, 2014)

Çizelgede kullanılan A, B ve D ağ mimarilerinin detayları Şekil 4.18’de gösterilmektedir. A mimarisi 11 evrişim katmanı ve 13 milyon parametreden, B mimarisi 13 evrişim katmanı ve 13 milyon parametreden ve son olarak D mimarisi ise 16 evrişim katmanı ve 138 milyon parametreden oluşmaktadır.

**Çizelge 4.1.** Ağ mimarisi, veri seti, üçlü kayıp, yüz hizalamanın yüz tanıma başarısına etkisi (Parkhi ve ark., 2015)

No.	Ağ mimarisi	Veri Seti	Yüz Hizalama(Eğitim)	Yüz Hizalama(Test)	Üçlü Kayıp(Embedding)	Başarı(%)
1	A	C	Hayır	Hayır	Hayır	92.83
2	A	F	Hayır	Hayır	Hayır	95.80
3	A	F	Hayır	Evet	Hayır	96.70
4	B	F	Hayır	Evet	Hayır	97.27
5	B	F	Evet	Evet	Hayır	96.17
6	D	F	Hayır	Evet	Hayır	96.73
7	B	F	Hayır	Evet	Evet	99.13

İlk olarak çizelgede bulunan 1 ve 2 numaralı satır değerleri karşılaştırılmaktadır. Burada diğer işlemler aynı tutulup sadece veri setleri değiştirilmiştir. F veri seti her bir



kimlik için 1.000 resim, C veri seti ise her bir kimlik için 50 resim içermektedir. Toplamda iki veri seti eşit sayıda kimliğe sahiptir. Sonuç olarak F veri seti C veri setine göre yüz tanıma başarı oranını daha fazla etkilemektedir. Bunun temel nedeni F veri setinin C veri setine göre daha fazla veriye sahip olmasıdır. Bundan yola çıkarak yüz tanıma sisteminin başarısını etkileyen etkenlerden birisinin veri seti olduğu sonucu ortaya çıkmaktadır. Yüz tanıma sistemlerinin az veri ile eğitilmesinin temel sebeplerinden birisi büyük veri setlerinin kamuoyuna açık olmamasıdır. Bu yüzden bu alandaki başarılar genellikle ellerindeki devasa veriler sayesinde Facebook ve Google gibi internet devleriyle sınırlı kalmaktadır. Örneğin, en son Google tarafından yapılan yüz tanıma sisteminde 200 milyon resim ve 8 milyon benzersiz kimlik kullanılarak yüz tanıma sisteminin eğitimi yapılmıştır (Schroff ve ark., 2015). Bu veri kümesinin boyutu, herkese açık herhangi bir yüz veri kümesinden neredeyse üç kat büyüklüğündedir. Çizelge 4.2 ve Çizelge 4.3'te görüldüğü gibi kamuoyuna açık olan eğitim veri setleri ve kamuoyuna açık olmayan eğitim veri setleri verilmektedir. Bu sonuçtan ortaya çıkıyor ki bu büyük veri setlerini oluşturmak, özellikle akademik alanlarda, çoğu uluslararası araştırma grubunun yeteneklerinin ötesindedir.

**Çizelge 4.2.** Kamuoyuna açık olan yüz veri setleri (Parkhi ve ark., 2015)

Veri Seti	Kimlik	Resim
LFW	5.749	13.233
WDRRef	2.995	99.773
CelebFaces	10.177	202.599

**Çizelge 4.3.** Kamuoyuna açık olmayan yüz veri setleri (Parkhi ve ark., 2015)

Veri Seti	Kimlik	Resim
FaceBook	4.030	4.4 Milyon
Google	8 Milyon	200 Milyon

Çizelge 4.1'deki 2 ve 3 numaralı satır değerleri karşılaştırıldığında iki satırda da tüm işlemlerin aynı sadece 3 numaralı satırda test aşamasında 2D yüz hizalama kullanıldığı görülmektedir. Elde edilen sonuca göre test aşamasında yüzün 2D hizalanması başarı oranını arttırmaktadır. Ancak aynı çizelgede bulunan 4 ve 5 numaralı satırlar incelendiğinde eğitim aşamasında yapılan 2D yüz hizalama işlemi aynı başarıyı sağlamamıştır. Bunun temel sebebi günlük dışarıda çekilen resimlerdeki arka planın test

aşamasına dâhil edilmemesi ve ağ parametrelerinin bunu kapsayacak şekilde güncellenmemesinden kaynaklanmaktadır.

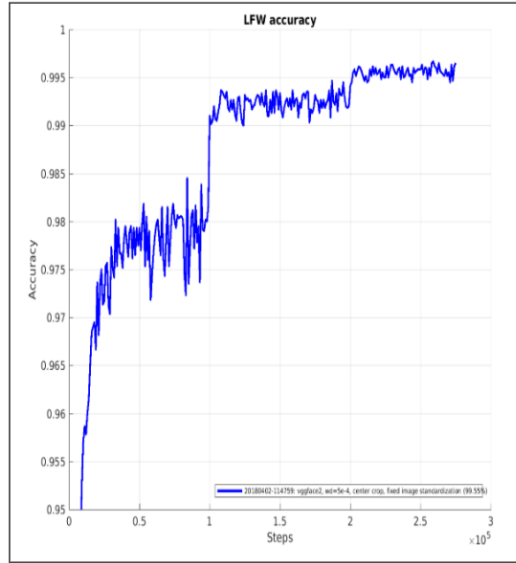
Çizelgedeki 3, 4 numaralı ve 4, 6 numaralı satırlar incelendiğinde Şekil 4.18’de gösterilen ağ mimarileri değiştirilip, bunlar dışındaki etkenler sabit kalacak şekilde karşılaştırılma yapılmaktadır. Bu karşılaştırma sonucunda B mimarisinin A mimarisinden daha başarılı çalıştığı görülmektedir. Ancak D mimarisi B mimarisinden daha derin bir ağ olmasına rağmen daha az başarılı çalışmıştır. Bunun birkaç olası sebebi vardır. D mimarisinin B mimarisinden daha fazla parametre içermesi ağın ezberlemesine sebep olmaktadır. Ayrıca D ağ mimarisi daha derinleştiği için öğrenme oranı gibi parametrelerin daha ince ayarlanması gerekmektedir.

4 ve 7 numaralı satırlar incelendiğinde eğitim aşamasında üçlü kayıp kullanılmıştır ve bunun dışındaki tüm etkenler sabit tutulmuştur. Burada üçlü kayıp yönteminin başarıya etkisi değerlendirilmiştir. Üçlü kayıp yöntemi ile eğitilen ağlarda %1.8 oranında başarı artmıştır. Hata oranını %68 oranında azaltmıştır.

Bu sonuçlar ışığında kullanılan mimarinin, eğitim yönteminin, eğitim verisinin ve yüzün hizalanmasının yüz tanıma için büyük öneme sahip olduğu tespit edilmiştir. Aynı zamanda bu sonuçlardan yola çıkarak kullandığımız uygulamada, yüz hizalama ve üçlü kayıp ile eğitilmiş modelin kullanılmasına karar verilmiştir.

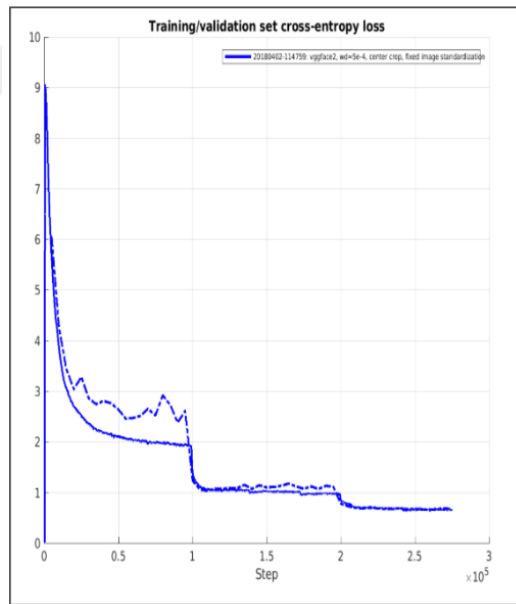
Bu çalışma kapsamında, VGGface veri seti ile eğitilmiş Inception V1 modeli kullanılmıştır. Eğitim 275.000 adım çalışıp ve öğrenme oranına göre sonlanmıştır. Nvidia Pascal Titan X GPU, Tensorflow r1.7, CUDA 8.0 ve CuDNN 6.0’da yaklaşık 30 saat sürede tamamlanmıştır (Deshpande, 2019).

LWF veri seti kullanılarak eğitilmiş modelin değerlendirilme sonuçları aşağıdaki gibi verilmiştir.



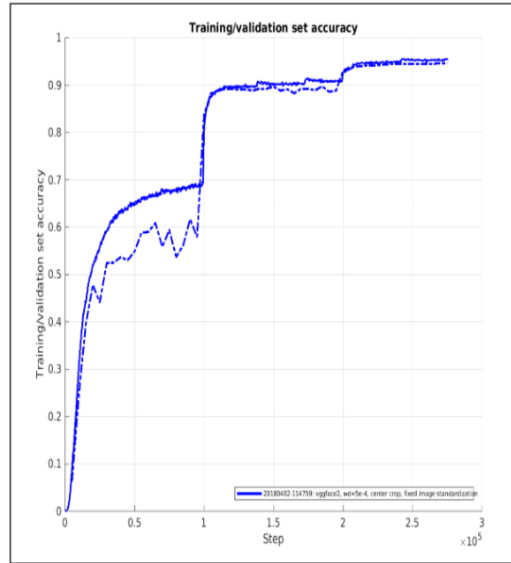
**Şekil 4.19.** LWF veri setinin eğitilmiş model üzerindeki değerlendirme grafiği (Sandberg, 2018)

Şekil 4.19’da görüldüğü gibi LWF veri seti üzerinde yapılan değerlendirmede 0.9950 doğruluk elde edilmiştir.



**Şekil 4.20.** LWF veri setinin ve eğitim aşamasındaki modelin çapraz entropi kaybının gösterimi (Sandberg, 2018)

Şekil 4.20’de gösterilen kesiksiz olan çizgi eğitim aşamasındaki çapraz entropi kaybını, kesikli olan çizgi değerlendirme aşamasındaki çapraz entropi kaybını göstermektedir. Değerlendirme 30.000 resim üzerinden yapılmıştır. Her 5 epoch’da bir değerlendirme yapılır.

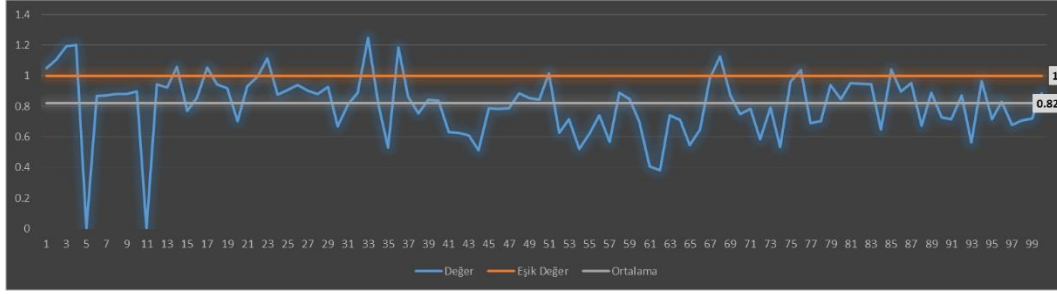


**Şekil 4.21.** LWF veri setinin değerlendirilmesi ve eğitim veri seti ile karşılaştırılması (Sandberg, 2018)

Şekil 4.21’de gösterilen grafik test ve eğitim verisinin doğruluğunu göstermektedir. Kesikli çizgi test veri seti üzerindeki doğruluğu, kesikli olmayan çizgi eğitim veri setinin doğruluğunu göstermektedir. LWF veri setindeki 30.000 resim kullanılmış olup 5 epoch’da bir güncelleme yapılmıştır.

#### 4.5. Yüz Servisi Testleri

Yüz servisinin başarı oranını tespit etmek için iki farklı test yapılmıştır. İlk olarak bir yüz veri seti olan LWF veri seti üzerinde testler yapılmaktadır. Bu veri seti 5.000 farklı kişiden alınan 9.000 resimden oluşmaktadır. Yüz doğrulama işlemi yapacağımız için bir kişiye ait en az iki resmin olması gerekmektedir. O yüzden veri seti içerisinde bulunan tek resme sahip kişiler silinmiştir. Bu işlem sonunda bir kişiye ait resimler içerisinde bir resim seçilip referans resmi olarak alınmaktadır. Bu referans resim, yüz kaydetme servisine gönderilip, öznitelikleri alınmaktadır. Bu öznitelik, aynı kişiye ait diğer resimler ile beraber yüz doğrulama servisine gönderilmektedir. Bu işlem tüm veri seti için tekrarlanmaktadır. Test sonucunda Şekil 4.22’de görülen grafik ortaya çıkmıştır. Grafikte ortalama ağırlık ve eşik değerine göre uzaklıklar gösterilmektedir. Eğitim sonucunda %82 başarı oranı elde edilmiştir. Başarı oranının bu kadar düşük olmasının sebebi, test verisinin ünlülerden oluşması ve yüz açılarının kameraya dönük olamamasından kaynaklanmaktadır.



**Şekil 4.22.** LWF veri setinde bulunan resimler arasındaki uzaklık grafiğine bağlı olarak eşik değerinin gösterimi

Ek olarak LWF veri seti içerisinde bir kişi seçilerek testler yapılmıştır. Veri seti içerisinde alınan kişiye ait resimler Şekil 4.23'te gösterilmiştir. Bu kişiye ait veri setinde 32 adet resim bulunmaktadır.



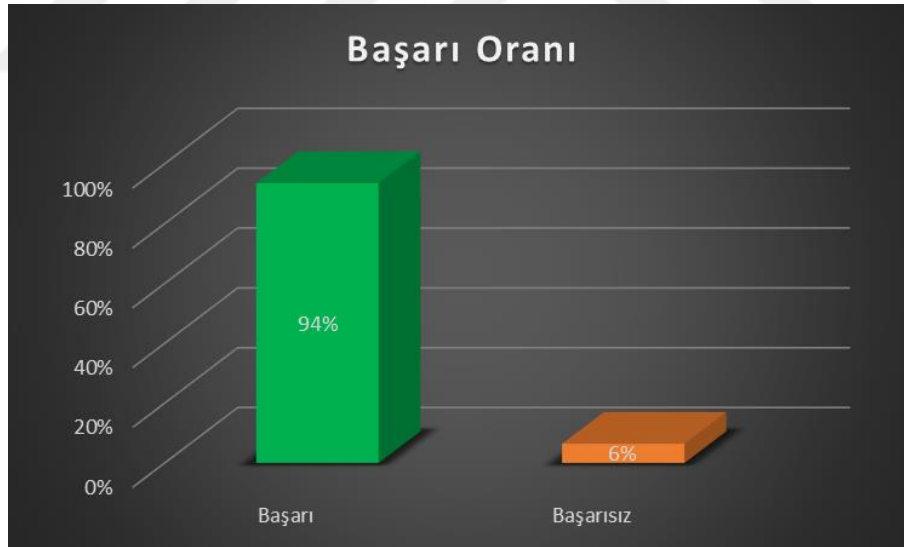
**Şekil 4.23.** LWF veri setinden alınan bir kişinin resimleri

İlk olarak bu resimlere yüz tespit ve hizalama işlemi yapılmaktadır. Hizalama işlemi sonucunda oluşan yeni resimler Şekil 4.24'te gösterilmektedir. İlk satırlar asıl resimleri, ikinci satırlar hizalanmış yüzleri göstermektedir.



**Şekil 4.24.** LWF test verisinden alınan bir kişinin yüz hizalanması (ilk sıradaki resimler orijinal resimler ikinci sıradaki resimler hizalanan yüz resimleridir).

Hizalama sonucu oluşan yeni resimler içerisinde ilk resim referans resim olarak seçilmiştir. Referans resim ile diğer resimler karşılaştırılıp değerlendirilme yapılmıştır. Tüm resimler eğitilmiş derin modelden geçirildikten sonra ilk resmin öznelikleri diğer resimlerle karşılaştırılmıştır ve aralarındaki uzaklıklar hesaplanmıştır. Eşik değeri 1 olarak seçilip doğrulama işlemi yapılmıştır. Bu doğrulama işlemi sonucunda Şekil 4.25'te görüldüğü gibi %94 başarı elde edilmiştir.



**Şekil 4.25.** LWF test verisinden alınan bir kişinin başarı grafiği

Test sonucunda doğru tespit edilen 29 adet resim Şekil 4.26'da gösterilmiştir. Yanlış tespit edilen iki resim ise Şekil 4.27'de gösterilmiştir. Bu iki resmin yanlış tespit edilmesinin sebebi, resimlerin referans resmine göre daha karanlık ve yüz açısının aşağıya doğru bakmasıdır.



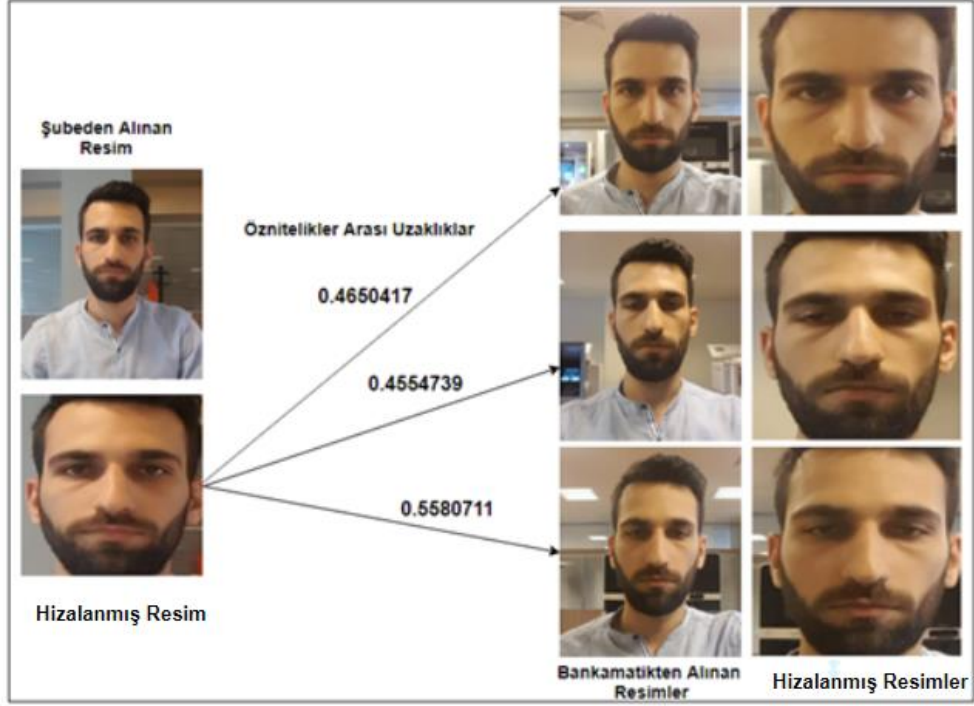
Şekil 4.26. LWF veri setinden test yapılan kişinin doğru tespit edilen resimleri



Şekil 4.27. LWF veri setinden test yapılan kişinin yanlış tespit edilen resimleri

İkinci olarak yapılan test, bir katılım bankasının şubesinden alınan resim ile bankamatik laboratuvarından alınan resimlerin karşılaştırılmasından oluşmaktadır. Şekil 4.28’de görüldüğü gibi şubeden alınan resim ile hizalanmış hali ve bankamatiklerden alınan üç resim ile hizalanmış halleri bulunmaktadır. Ek olarak şubeden alınan resmin diğer resimler ile arasındaki uzaklıkları verilmiştir. Şekilde gösterilen uzaklıklardan da görüldüğü gibi belirlenen 0,7 eşik değerine göre tüm resimler doğrulanmıştır.





**Şekil 4.28.** Bir katılım bankasının şubesinden ve laboratuvarından alınan resimlerin hizalanması ve birbirine olan uzaklıklarının gösterilmesi



## 5. SONUÇLAR VE ÖNERİLER

### 5.1. Sonuçlar

Bu çalışmada, bankamatiklere manyetik kart ve PIN doğrulamaya ek olarak yüz doğrulama eklenmiştir. Yüz doğrulama için VGGFace2 veri seti ile eğitilmiş Inception V1 modeli kullanılmıştır. Eğitilmiş modelin LWF veri seti kullanılarak değerlendirme yapılmıştır. Bu değerlendirme sonucunda 0.995 doğruluk sağladığı gösterilmiştir. Kullanılan model üçlü kayıp yöntemi ile eğitilmiştir. Bu yöntem aynı kimliğe sahip olan resimlerin uzaklıklarını azaltır. Ayrıca diğer ağ eğitim yöntemlerinden farklı olarak farklı kimliğe sahip olan resimler arasındaki uzaklığı artırır ve geri yayılımı buna göre yapar. Üçlü kayıp yöntemi ile ağ eğitildiğinde başarı oranını %1.8 arttırmıştır. Aynı zamanda hata oranını da %68 azalttığı gösterilmiştir.

Yüz doğrulama aşamasından önce çoklu evrişim ağları kullanılarak yüz hizalama yapılmıştır. Yapılan değerlendirmelerde test aşamasında yüz hizalama işlemi yüz doğrulama başarısını 0.9 oranında arttırdığı belirtilmiştir. Kullandığımız yüz tanıma sisteminin yüksek başarısında üçlü kayıp yöntemi ile ağın eğitilmesinin ve yüz hizalanmasının payı yüksektir.

Yüz servisini test etmek için LWF veri seti kullanılmıştır. Yapılan testlerde %82 başarı elde edilmiştir. Başarı oranının düşük çıkmasının sebebi, veri seti içerisinde bulunan yüzü kameraya dönük olmayan ve çok düşük ışıkta çekilen resimlerin çokça bulunmasından kaynaklanmaktadır. Aynı şekilde LWF eğitim seti içerisinde bulunan bir kişiye ait 32 resim alınmıştır. Bu resimlerin ilki referans resim alınarak, diğer resimler ile karşılaştırılma yapılmıştır. Karşılaştırma sonucu %94 başarı elde edilmiştir. Bu resimlerin 29'unun doğrulaması başarılı yapılırken kalan 2'sinin doğrulaması yapılamamıştır.

Bir katılım bankasının şubesinden alınan resim ile bankamatik laboratuvarında farklı açılardan alınan örnek resimler yüz servisi uygulamasını test etmek için karşılaştırılmıştır. Karşılaştırma sonucunda tüm resimler belirlemiş olduğumuz eşik değerinin altında kalmış ve %100 başarı elde edilmiştir.

## 5.2. Öneriler

Yüz tanıma, diğer doğrulama yöntemleri ile birleştirildiğinde çok güvenli sistemleri oluşturmaktadır. Bu doğruluk ile beraber giderek daha fazla yerde kullanılmaya başlanmıştır.

Bu çalışmada, yüz tanıma bankamatiklere uygulanıp daha güvenli hale getirilmiştir. LWF ve VGGFace veri setleri ile eğitilmiş modeller kullanılmıştır. Daha büyük veri setleri ile eğitildiğinde %100 başarı oranına yaklaşabilmektedir. Bu gelişmiş veri setleri ile eğitilip başarı oranı daha da fazla arttırılabilir.

Yüz tanıma sistemi video, maske ve resimlerin kameraya gösterilmesi ile atlatılabilir. Bu atlatma yöntemlerine karşı derinlik sensörleri veya resimdeki ışığın yansımalarının bulunması yöntemleri ile engellenebilir.

Ek olarak, yüz tanımadan önce yaşlandırma efektleri kullanılarak kişinin ileriki yaşlardaki resimlerinden de doğrulama sağlanabilir.

## KAYNAKLAR

- Afonja, T., 2018, Basic Overview of CNN, <https://medium.com/ai-saturdays/aisaturdayagos-basic-overview-of-cnn-cd354470e2bb>: [2406.2019].
- Andrej Karpathy ve Johnson, J., 2019, Convolutional Neural Networks for Visual Recognition, <http://cs231n.github.io/convolutional-networks/>: [24.08.2019].
- Aru, O. E. ve Gozie, I., 2013, Facial Verification Technology for Use In Atm Transactions, *American Journal of Engineering Research (AJER)*, 2.
- Awodele, O. ve Akanni, A., 2012, Combating automated teller machine frauds through biometrics, *International Journal of Emerging Technology and Advanced Engineering*, 2 (11), 441-444.
- Bell, J., 2014, Machine learning: hands-on for developers and technical professionals, John Wiley & Sons, p.
- Buduma, N. ve Locascio, N., 2017, Fundamentals of deep learning: Designing next-generation machine intelligence algorithms, " O'Reilly Media, Inc.", p.
- Chan, L.-W., Fallside, F. J. C. s. ve language, 1987, An adaptive training algorithm for back propagation networks, 2 (3-4), 205-218.
- Çayıröğlü, İ., 2015, İleri Algoritma Analizi-5 Yapay Sinir Ağları, *Karabük Üniversitesi Mühendislik Fakültesi*.
- Deshpande, A., 2019, Deep Learning Papers You Need To Know About, <https://adeshpande3.github.io/The-9-Deep-Learning-Papers-You-Need-To-Know-About.html>:
- Doğan, F. ve Türkoğlu, İ., 2018, Derin Öğrenme Algoritmalarının Yaprak Sınıflandırma Başarımlarının Karşılaştırılması, *Sakarya University Journal of Computer and Information Sciences*, 1 (1), 10-21.
- Fukushima, K. J. B. c., 1980, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position, 36 (4), 193-202.
- Gadoue, S., 2007, Industrial Automation-Artificial Neural Networks', *Newcastle University School of Electrical, Electronic & Computing Engineering*.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G. ve Cai, J., 2018, Recent advances in convolutional neural networks, *Pattern Recognition*, 77, 354-377.
- He, K., Zhang, X., Ren, S. ve Sun, J., 2016, Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778.
- Hebb, D. O., 1962, The organization of behavior: a neuropsychological theory, Science Editions, p.
- Hinton, G. E. ve Salakhutdinov, R. R. J. s., 2006, Reducing the dimensionality of data with neural networks, 313 (5786), 504-507.
- Kale, A. S. ve Nanda, S. K., 2014, A Review Paper on Design of Highly Secured Automatic Teller Machine System by using Aadhaar card and Fingerprint, *International Journal of Advance Research in Computer Science and Management Studies Research Paper*, 2 (1).
- Kawaguchi, K., 2001, A multithreaded software model for backpropagation neural network applications.
- Keskenler, M. F. ve Keskenler, E. F., 2017, Geçmişten Günümüze Yapay Sinir Ağları ve Tarihi, *Takvim-i Vekayi*, 5 (2), 8-18.
- Kim, P., 2017, Matlab deep learning, In: With Machine Learning, Neural Networks and Artificial Intelligence, Eds: Springer, p.

- Kriesel, D., 2007, A brief introduction on neural networks.
- Krizhevsky, A., Sutskever, I. ve Hinton, G. E., 2012, Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, 1097-1105.
- Lashley, K. S., 1950, In search of the engram.
- Macukow, B., 2016, Neural networks state of art, brief history, basic models and architecture, *Computer Information Systems and Industrial Management, Cism 2016*, 9842, 3-14.
- Maladkar, K., 2018, Types Of Activation Functions In Neural Networks And Rationale Behind It, <https://analyticsindiamag.com/most-common-activation-functions-in-neural-networks-and-rationale-behind-it/>: [08.06.2019].
- McCulloch, W. S. ve Pitts, W. J. T. b. o. m. b., 1943, A logical calculus of the ideas immanent in nervous activity, 5 (4), 115-133.
- Nordbotten, S., 2006, Data Mining with Neural Networks, *Svein Nordbotten & Associates, Bergen*.
- Özkan, İ. ve Ülker, E., 2017, Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri, *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, 6 (3), 85-104.
- Parkhi, O. M., Vedaldi, A. ve Zisserman, A., 2015, Deep face recognition, *bmvc*, 6.
- Patterson, J. ve Gibson, A., 2017, Deep learning: A practitioner's approach, " O'Reilly Media, Inc.", p.
- Rosebrock, A., 2017, ImageNet: VGGNet, ResNet, Inception, and Xception with Keras, <https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/>: [24/06/2019].
- Rosenblatt, F., 1961, Principles of neurodynamics. perceptrons and the theory of brain mechanisms, *Cornell Aeronautical Lab Inc Buffalo NY*.
- Sandberg, D., 2018, Training Using The VGGFace2 Dataset, <https://github.com/davidsandberg/facenet/wiki/Training-using-the-VGGFace2-dataset>: [30.06.2019].
- Schroff, F., Kalenichenko, D. ve Philbin, J., 2015, Facenet: A unified embedding for face recognition and clustering, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 815-823.
- Sharma, S., Shanmugasundaram, K. ve Ramasamy, S. K., 2016, FAREC—CNN based efficient face recognition technique using Dlib, *2016 International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, 192-195.
- Simonyan, K. ve Zisserman, A. J. a. p. a., 2014, Very deep convolutional networks for large-scale image recognition.
- Sun, Y., Wang, X. ve Tang, X., 2014, Deep learning face representation from predicting 10,000 classes, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1891-1898.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. ve Rabinovich, A., 2015, Going deeper with convolutions, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9.
- Şahin, Ö., 2017, iOS platformunda görme engelliler için TL tanıma uygulaması, *Selçuk Üniversitesi Fen Bilimleri Enstitüsü*.
- Şeker, A., Diri, B. ve Balık, H. H., 2017, Derin Öğrenme Yöntemleri ve Uygulamaları Hakkında Bir İnceleme, *Gazi Mühendislik Bilimleri Dergisi (GMBD)*, 3 (3), 47-64.

- Taigman, Y., Yang, M., Ranzato, M. A. ve Wolf, L., 2014, Deepface: Closing the gap to human-level performance in face verification, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701-1708.
- Taşova, O., 2011, Yapay sinir ağları ile yüz tanıma, *DEÜ Fen Bilimleri Enstitüsü*.
- Wang, M. ve Deng, W., 2018, Deep face recognition: A survey, *arXiv preprint arXiv:1804.06655*.
- Widrow, B. ve Hoff, M. E., 1960, Adaptive switching circuits, *Stanford Univ Ca Stanford Electronics Labs*.
- Yiğit, A., 2017, İş süreçlerinde insan görüşünü derin öğrenme ile destekleme, *Trakya Üniversitesi Fen Bilimleri Enstitüsü*.
- Zeiler, M. D. ve Fergus, R., 2014, Visualizing and understanding convolutional networks, *European conference on computer vision*, 818-833.
- Zhang, K., Zhang, Z., Li, Z. ve Qiao, Y., 2016, Joint face detection and alignment using multitask cascaded convolutional networks, *IEEE Signal Processing Letters*, 23 (10), 1499-1503.
- Zhang, Q., Yang, L. T., Chen, Z. ve Li, P., 2018, A survey on deep learning for big data, *Information Fusion*, 42, 146-157.

## ÖZGEÇMİŞ

### KİŞİSEL BİLGİLER

**Adı Soyadı** : Mehmet YILDIRIM  
**Uyruğu** : TC  
**Doğum Yeri ve Tarihi** : Malazgirt - 12.01.1992  
**Telefon** : 0539 689 68 66  
**E-Posta** : yildirim.mehmet.49@gmail.com

### EĞİTİM

Derece	Adı, İlçe, İl	Bitirme Yılı
Lise	: Malazgirt Alparslan Lisesi, Malazgirt, Muş	2009
Üniversite	: Selçuk Üniversitesi, Selçuklu, Konya	2015
Yüksek Lisans	: Konya Teknik Üniversitesi, Selçuklu, Konya	
Doktora	:	

### İŞ DENEYİMLERİ

Yıl	Kurum	Görevi
2012 - 2013	İnforce Bilişim Teknolojileri	Yazılım Mühendisi
2013 - Halen	Kuveyt Türk Katılım Bankası	Kıdemli Yazılım Mühendisi

### UZMANLIK ALANI

Web uygulama geliştirme ve altyapı çalışmaları

### YABANCI DİLLER

İngilizce

### YAYINLAR

Derin Öğrenme Teknikleri Kullanılarak Yüz Doğrulamanın Bankamatiklere Uygulanması

*21. Akademik bilişim konferansı, Ordu, Türkiye (13-15.02.2019)*

Mobile Application Price Estimation By Using Artificial Neural Network and Bayesian Classifier

*Data Science, Machine Learning and Statistics 2019, Van, Türkiye (26-29.06.2019)*