



T.C.  
BURSA ULUDAĞ ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**3 BOYUTLU DERİNLİK KAMERASI İLE  
DERİN ÖĞRENME TABANLI GÜVENLİ YÜZ TANIMA**

**Sedat YILDIZ**  
0000-0003-4632-8186

Doç. Dr. Ahmet Emir DİRİK  
(Danışman)

YÜKSEK LİSANS TEZİ  
ELEKTRONİK MÜHENDİSLİĞİ

BURSA - 2021

## TEZ ONAYI

Sedat YILDIZ tarafından hazırlanan "3 Boyutlu Derinlik Kamerası İle Derin Öğrenme Tabanlı Güvenli Yüz Tanıma" adlı tez çalışması aşağıdaki jüri tarafından oy birliği ile Bursa Uludağ Üniversitesi Fen Bilimleri Enstitüsü Elektronik Mühendisliği Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

**Danışman :** Doç. Dr. Ahmet Emir DİRİK  
0000-0002-6200-1717

**Başkan :** Doç. Dr. Ahmet Emir DİRİK  
0000-0002-6200-1717  
Bursa Uludağ Üniversitesi  
Mühendislik Fakültesi  
Bilgisayar Mühendisliği Bölümü

**Üye:** Doç. Dr. Ersen YILMAZ  
0000-0002-6620-655X  
Bursa Uludağ Üniversitesi  
Mühendislik Fakültesi  
Elektrik-Elektronik Mühendisliği Bölümü

**Üye:** Dr. Öğr. Üyesi Saffet VATANSEVER  
0000-0002-4680-1263  
Bursa Teknik Üniversitesi  
Mühendislik ve Doğa Bilimleri Fakültesi  
Mekatronik Mühendisliği Bölümü

Yukarıdaki sonucu onaylıyorum.

Prof. Dr. Hüseyin Aksel EREN  
Enstitü Müdürü  
16.10.2021 (Tarih)

**B.U.Ü. Fen Bilimleri Enstitüsü, tez yazım kurallarına uygun olarak hazırladığım bu tez çalışmada;**

- tez içindeki bütün bilgi ve belgeleri akademik kurallar çerçevesinde elde ettiğimi,
- görsel, işitsel ve yazılı tüm bilgi ve sonuçları bilimsel ahlak kurallarına uygun olarak sunduğumu,
- başkalarının eserlerinden yararlanması durumunda ilgili eserlere bilimsel normlara uygun olarak atıfta bulunduğumu,
- atıfta bulunduğum eserlerin tümünü kaynak olarak gösterdiğimi,
- kullanılan verilerde herhangi bir tahrifat yapmadığımı,
- ve bu tezin herhangi bir bölümünü bu üniversite veya başka bir üniversitede başka bir tez çalışması olarak sunmadığımı

**beyan ederim.**

11/02/2021  
Sedat YILDIZ  
İmza



## ÖZET

Yüksek Lisans Tezi

### 3 BOYUTLU DERİNLİK KAMERASI İLE DERİN ÖĞRENME TABANLI GÜVENLİ YÜZ TANIMA

**Sedat YILDIZ**

Bursa Uludağ Üniversitesi  
Fen Bilimleri Enstitüsü  
Elektronik Mühendisliği Anabilim Dalı

**Danışman:** Doç. Dr. Ahmet Emir DİRİK

Derin öğrenme ile yüz tanıma teknikleri, son yıllarda çok hızlı gelişim gösteren ve günlük hayatta pek çok alanda uygulaması olan bir araştırma konusudur. Suçluların takip edilmesi, personellerin şirkete giriş çıkış takibi gibi alanlar yüz tanıma sistemlerinin kullanılabileceği alanlardır. Yüz tanıma ile birlikte bir diğer önemli nokta ise yüz tanıma sistemlerine karşı yapılan saldırıların önlenmesidir. Örneğin yüz tanıma sistemleri, vesikalık fotoğraf, yazıcıdan kişinin yüz fotoğraf çıktısının alınması, telefon veya tableten yüz fotoğrafı, video görüntüleri, maske kullanılması gibi yöntemlerle yanıltılabilmektedir. Bu nedenle başarılı bir yüz tanıma sistemi geliştirmek kadar yüz tanıma sisteminin aldatılmasını önlemek de önemli bir konudur. Bu çalışmada derin öğrenme teknikleri kullanılarak başarıyı yüksek bir yüz tanıma sistemi geliştirilmiştir. 3D derinlik kamerası ile derinlik bilgisi analizi ve göz kırpma tespiti yapılarak yüz tanıma sistemlerini yanıltmaya yönelik gerçekleştirilen ataklara karşı güvenliği sağlayan bir sistem geliştirilmiştir. 3D derinlik kamerasından alınan derinlik bilgilerinin gradyeni hesaplanarak genlik ve açı histogramları çıkarılıp bu histogramların ortalama, ortanca ve standart sapma gibi istatistiksel analizi yapılarak kamera karşısındaki kişilerin canlılık tespiti yapılmıştır. Derinlik bilgisine ek olarak karar ağacı regresyonu tekniği kullanılarak göz kırpma tespiti yapıp sistemin canlılık tespit başarıyı artırılmıştır.

**Anahtar Kelimeler:** Derin öğrenme, yüz tespiti, yüz tanıma, 3 boyutlu derinlik kamerası, canlı- cansız kişi tespiti, göz kırpma tespiti, evrişimli sinir ağları  
**2021, vii + 94 sayfa**



## **ABSTRACT**

M.Sc. Thesis

### **DEEP LEARNING BASED SECURE FACE RECOGNITION WITH 3D DEPTH CAMERA**

**Sedat YILDIZ**

Bursa Uludağ University  
Graduate School of Natural and Applied Sciences  
Department of Electronic Engineering

**Supervisor:** Assoc. Prof. Dr. Ahmet Emir DİRİK

Deep learning and facial recognition techniques are a research subject that has developed very rapidly in recent years and has applications in many areas in daily life. Facial recognition systems can be used in areas such as tracking criminals, tracking personnel entry and exit from the company. Another important point along with face recognition is the prevention of attacks against face recognition systems. For example, face recognition systems can be misled by methods such as passport photos, printing a person's face photo from the printer, using a face photo from a phone or tablet, video images, and masks. Therefore, preventing the face recognition system from being deceived is as important as developing a successful face recognition system. In this study, a highly successful face recognition system has been developed using deep learning techniques. A system has been developed that provides security against attacks that are made to mislead face recognition systems by analyzing depth information and detecting blinking with a 3D depth camera. By calculating the gradient of the depth information obtained from the 3D depth camera, amplitude and angle histograms were extracted, and the vitality of the people in front of the camera was determined by performing statistical analysis of these histograms such as mean, median and standard deviation. In addition to depth information, blink detection was performed using the decision tree regression technique and the vitality detection performance of the system was increased.

**Keywords:** Deep learning, face detection, face recognition, 3D depth camera, liveness detection, eye blink detection, convolutional neural networks

**2021, vii + 94 pages**

## TEŐEKKÜR

Bu tez alıŐmasının gerekleŐtirilmesinde, deęerli bilgilerini benimle paylaŐan, maddi manevi her trl desteęini benden esirgemeyen, bana her zaman destek olan, iŐ disiplini, bakıŐ aısı, tecrbesi, yol gstericilięi, samimiyeti ile her zaman bana rnek olan, yaŐadığım sıkıntılarını sıkılmadan dinleyip bana yardımcı olmaya alıŐan, dert edinen ok kıymetli saygıdeęer danıŐman hocam Do. Dr. Ahmet Emir DİRİK'e, maddi manevi desteklerini benden esirgemeyen her zaman yanımda olan ok deęerli annem, babam ve kardeŐlerime, bana maddi manevi her zaman destek olan, her zaman yanımda olan deęerli dostum Adem TUNA'ya, alıŐtığım Őirket olan Teracity Yazılım'da bana desteklerini esirgemeyen, tecrbeleriyle bana yol gsteren baŐta deęerli patronum Osman AKIN olmak zere, alıŐmalarımnda bana yardımcı olan, yaptığım demo uygulamaya katkı saęlayan, birlikte yz tanıma, cinsiyet ve yaŐ tespiti konularını kapsayan bir TEYDEB Projesi geliŐtirdiğimiz yazılım mdrmz Engin HAZAR'a, AR-GE mdrmz Nihal BİLAL'e, yazılım Őefimiz Cihan KABRAN'a ve yazılım Őefimiz Erdem ZGÜR'e, deęerli alıŐma arkadaŐlarım İsmail BAYINDIR'a, Dilara SAęER'e, Gufran GÖK'e, Melis SEęMEN'e teŐekkrlerimi sunarım.

Sedat YILDIZ  
11/02/2021

## İÇİNDEKİLER

	<b>Sayfa</b>
ÖZET .....	i
ABSTRACT .....	ii
TEŞEKKÜR .....	iii
SİMGELER VE KISALTMALAR DİZİNİ .....	vi
ŞEKİLLER DİZİNİ .....	ix
ÇİZELGELER DİZİNİ .....	xii
1 GİRİŞ .....	1
2 KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI.....	4
2.1 Yapay Sinir Ağları (Artificial Neural Network) .....	6
2.1.1 Yapay Sinir Hücre Yapısı .....	8
2.1.2 Girdiler (Inputs) .....	9
2.1.3 Ağırlıklar (Weights) .....	9
2.1.4 Toplama Fonksiyonu (Sum Function).....	10
2.1.5 Aktivasyon Fonksiyonu (Activation Function) .....	11
2.1.6 Hücre Çıktısı .....	16
2.2 Yapay Sinir Hücresi Çalışma Prensibi .....	16
2.3 Yapay Sinir Ağı Modelleri.....	17
2.3.1 Tek Katmanlı Algılayıcılar (Single Layer Perceptron).....	17
2.3.2 Çok Katmanlı Algılayıcılar (Multilayer Perceptron) .....	18
2.3.3 İleri Beslemeli Yapay Sinir Ağları .....	19
2.3.4 Geri Beslemeli Yapay Sinir Ağları.....	19
2.4 Derin Öğrenme (Deep Learning).....	22
2.5 Evrişimli Sinir Ağları (Convolutional Neurat Network, CNN) .....	25
2.6 Evrişimli Sinir Ağı Mimarisi .....	26
2.7 Örneklem Katmanı (Pooling Layer) .....	32
2.8 Tam Bağlantılı Katman (Fully Connected Layer).....	34
2.9 Yığın Normalleştirme (Batch Normalization) .....	34
2.10 Seyreltme Katmanı (Dropout) .....	34
2.11 Veri Sınıflandırma.....	36
2.11.1 K-En Yakın Komşuluk Sınıflandırıcı .....	36
2.11.2 Destek Vektör Makineleri (Support Vector Machines).....	39

2.12	Yüz Tespiti.....	40
2.12.1	Bilgi Tabanlı Yöntemler.....	41
2.12.2	Özellik Tabanlı Yöntemler.....	41
2.12.3	Şablon Eşleştirme Tabanlı Yöntemler.....	41
2.12.4	Görünüm Tabanlı Yöntemler.....	41
2.13	3D Derinlik Kameraları .....	45
3	MATERYAL VE YÖNTEM.....	48
3.1	Yüz Tanıma.....	48
3.1.1	Geleneksel Yüz Tanıma Algoritmaları.....	50
3.1.2	Derin Öğrenme Tabanlı Yüz Tanıma: .....	53
3.2	Canlı - Cansız Yüz Tespiti .....	64
4	BULGULAR.....	78
4.1	Yüz Tespiti Analiz Sonuçları.....	78
4.2	Yüz Tanıma Analiz Sonuçları.....	79
4.3	Canlı - Cansız Yüz Tespiti Analiz Sonuçları .....	82
5	TARTIŞMA VE SONUÇ .....	91
	ÖZGEÇMİŞ.....	99

## SİMGELER VE KISALTMALAR DİZİNİ

<b>Semboller</b>	<b>Açıklama</b>
$\mu$	Ortalama
$\sigma$	Varyans

<b>Kısaltmalar</b>	<b>Açıklama</b>
ML	Makine Öğrenmesi
DL	Derin Öğrenme
NN	Sinir Ağı
ESA	Evrişimli Sinir Ağı
YSA	Yapay Sinir Ağı
GPU	Grafiksel İşlem Birimi
CPU	Merkezi İşlem Birimi
PCA	Temel Bileşenler Analizi
LBP	Yerel İkili Örüntüler
K-NN	K-en yakın komşuluk
SVM	Destek Vektör Makineleri
CV	Bilgisayarlı Görü
HOG	Yönlendirilmiş Gradyanların Histogramı
LDA	Doğrusal Ayırma Analizi
EGA	Elastik Grafik Eşleştirme
RELU	Rectified Linear Unit
DNN	Derin Öğrenme Ağı
LFW	Labeled Faces in the Wild Home
ResNet	Residual Network
ILSVRC	Büyük Ölçekli Görsel Tanıma Yarışması
NLP	Natural Language Processing
MS	Milisaniye
FOV	Field of View
EAR	Göz En Boy Oranı
STD	Standart Sapma
3D	3 Boyutlu
2D	2 Boyutlu
YZ	Yapay Zeka

## Çeviriler

## Açıklama

Machine Learning	Makine Öğrenmesi
Deep Learning	Derin Öğrenme
Neural Network	Sinir Ağı
Convolution Neural Networks	Evrişimli Sinir Ağı
Graphical Processing Units	Grafiksel İşlem Birimi
Central Processing Units	Merkezi İşlem Birimi
Principal Component Analysis	Temel Bileşenler Analizi
Eigenfaces	Özyüzler
Local Binary Patterns	Yerel İkili Örüntüler
K-nearest Neighbors	K-en yakın komşuluk
Support Vector Machines	Destek Vektör Makineleri
Computer Vision	Bilgisayarlı Görü
Image Processing	Görüntü İşleme
Histogram of Gradients	Yönlendirilmiş Gradyanların Histogramı
Linear Discriminant Analysis	Doğrusal Ayırma Analizi
Elastic Graph Matching	Elastik Grafik Eşleştirme
Laplacian Faces	Laplasyen Yüzleri
Inputs	Girdiler
Weights	Ağırlıklar
Sum Function	Toplama Fonksiyonu
Activation Function	Aktivasyon Fonksiyonu
Artificial Intelligence	Yapay Zeka
Artificial Neural Network	Yapay Sinir Ağı
Deep Neural Network	Derin Sinir Ağı
Sum	Toplam
Step	Adım
Linear	Doğrusal
Rectified Linear Unit	Doğrultulmuş doğrusal Birim
Vanishing Gradient	Kaybolan Eğim
Leaky	Sızıntı
Perceptron	Algılayıcı
Multilayer Perceptron	Çok Katmanlı Algılayıcılar
Back Propagation	Geriye Yayılım
Supervised Learning	Gözetimli Öğrenme
Unsupervised Learning	Gözetimsiz Öğrenme
Semi Supervised Learning	Yarı Gözetimli Öğrenme
Big Data	Büyük Veri
Pooling Layer	Örnekleme Katmanı
Convolution	Evrişim
Fully Connected Layer	Tam Bağlantılı Katman
Batch	Yığın
Epoch	Döngü

Normalization	Normalleştirme
Dropout	Seyreltme
Stride	Kaydırma
Padding	Dolgulama
Overfitting	Aşırı Uyum
Downsampling	Alt Örnekleme
Flattening	Düzleştirme
Threshold	Eşikleme
Activation	Aktivasyon
Regularization	Düzenleştirme
Binary Classification	İkili Sınıflandırma
Multiclass Classification	Çok Sınıflı Sınıflandırma
Decision Trees	Karar Ağaçları
Random Forest	Rastgele Değişken
Learning Rate	Öğrenme Katsayısı
Architecture	Mimari
Early Stopping	Erken Durdurma
True	Olumlu/Doğru
False	Olumsuz/Yanlış
Epoch	Adım
Large Scale Visual Recognition	Büyük Ölçekli Görsel Tanıma
Challenge	Yarışma
Holistic Features	bütünsel özellikler
Local Feature Approaches	yerel özellik yaklaşımı
Residual Network	Artık Değerli Ağ
Triplet Loss	Üçlü Kayıp
Pos Tagger	Konuşma Bölümü Etiketleme
Natural Language Processing	Doğal Dil İşleme
Siamese Network	Siamese Ağı
Print attack	Baskı Atağı
Video attack/Replay attack	Video Atağı
3D Mask Attack	3D Maske Atağı
Active Flash	Aktif Flaş
Eye Blink Detection	Göz Kırpma Tespiti
Depth Field of View	Derinlik Görüş Alanı
eye landmarks	Göz Karakteristik Noktaları
Eye Aspect Ratio	Göz En Boy Oranı
Accuracy	Başarım
Aligning	Hizalama
Median	Ortanca

## ŞEKİLLER DİZİNİ

	<b>Sayfa</b>
Şekil 2.1 (a) Biyolojik sinir hücresi (b) Yapay sinir hücresi .....	7
Şekil 2.2 Yapay sinir ağı hücre Modeli .....	9
Şekil 2.3 Adım aktivasyon fonksiyon grafiği .....	12
Şekil 2.4 Doğrusal fonksiyon grafiği .....	13
Şekil 2.5 Sigmoid fonksiyon grafiği .....	13
Şekil 2.6 Tanjant hiperbolik fonksiyon grafiği .....	14
Şekil 2.7 RELU fonksiyon grafiği.....	15
Şekil 2.8 Leaky RELU fonksiyon grafiği.....	15
Şekil 2.9 Swish fonksiyon grafiği .....	16
Şekil 2.10 Yapay sinir hücresi hesaplama örneği.....	17
Şekil 2.11 Çok katmanlı algılayıcı örneği.....	18
Şekil 2.12 İleri beslemeli yapay sinir ağı (Öztemel 2006) .....	19
Şekil 2.13 Geri beslemeli yapay sinir ağı .....	20
Şekil 2.14 Yapay sinir ağı öğrenme türleri.....	21
Şekil 2.15 Derin öğrenme, makine öğrenmesi ve yapay zeka.....	23
Şekil 2.16 Evrişimli sinir ağı örneği.....	26
Şekil 2.17 Evrişim katmanı (a) Girdi örneği (b) Filtre örneği .....	27
Şekil 2.18 Evrişim (Konvolüsyon) işlemi örneği.....	28
Şekil 2.19 Evrişim işlemi ile kenar bulma (a) Giriş (b) Filtre (c) Çıktı, sıfırdan büyük değerler beyaz, sıfıra eşit değerler gri, sıfırdan küçük değerler siyah ile gösterilmiştir.....	28
Şekil 2.20 Evrişim işlem basamakları.....	29
Şekil 2.21 3D evrişim örneği.....	30
Şekil 2.22 Kaydırma (Stride) örneği: (a) S = 1 (b) S = 2 .....	31
Şekil 2.23 Dolgulama (Padding) örneği .....	31
Şekil 2.24 Maksimum ve ortalama örnekleme .....	33
Şekil 2.25 3D örnekleme .....	33
Şekil 2.26 Seyreltme (Dropout) örneği .....	35
Şekil 2.27 K-NN sınıflandırıcı ile sınıflandırılmış veri .....	37



Şekil 2.28 K-NN sınıflandırıcı ile veri sınıflandırma .....	38
Şekil 2.29 SVM doğrusal sınıflandırma örneği .....	39
Şekil 2.30 SVM doğrusal olmayan sınıflandırma örneği .....	40
Şekil 2.31 LBP eşikleme örneği .....	44
Şekil 3.1 Yüz tanıma sistemi aşamaları .....	48
Şekil 3.2 (a) Giriş görüntüsü ve yüz tespit sonucu (b) Elde edilen yüz bölgesi (c) Elde edilen yüz özellik vektörü (d) Sınıflandırma yöntemleri ile elde edilen öznitelik vektörlerinin veritabanında saklanması ve giriş ile en eşleşen örneğin işaretlenmesi .....	49
Şekil 3.3 LBP kodunun elde edilmesi .....	52
Şekil 3.4 LBP histogramının elde edilmesi .....	52
Şekil 3.5 (a) Giriş görüntüsü (b) Yüz tespit örneği (c) Kırpılmış yüz görüntüsü (d) ResNet-29 Modeli (e) Yüz öznitelik vektör örneği (128 uzunluklu) (f) Sınıflandırma (K-NN sınıflandırıcı) (g) Yüz tanıma .....	53
Şekil 3.6 ResNet-29 yüz tanıma modeli mimarisi .....	55
Şekil 3.7 CIFAR-10 veri kümesinin 56 katmanlı ve 20 katmanlı bir model ile (a) Eğitim hatası (b) Test hatası .....	56
Şekil 3.8 ResNet blok yapısı .....	57
Şekil 3.9 (a) VGG-19 model mimarisi (b) 34 katmanlı ESA (c) ResNet-34 modeli .....	58
Şekil 3.10 (a) Giriş görüntüsü (b) ResNet-29 modeli ile elde edilen 128 uzun- luklu öznitelik vektör örneği .....	60
Şekil 3.11 Üçlü kayıp örneği .....	61
Şekil 3.12 Siyam ağı ile üretilen 128 uzunluklu öznitelik vektör örnekleri .....	63
Şekil 3.13 Yüz tanıma sistemi saldırı örnekleri .....	65
Şekil 3.14 Aktif flaş tekniği örneği .....	67
Şekil 3.15 Canlı - Cansız Yüz Tespiti Akış Diyagramı .....	69
Şekil 3.16 Yüz tespiti yapılarak yüz bölgesi seçilmiş ve derinlik bilgisi ile RGB piksel koordinatları eşleştirilerek hizalanmış 3 boyutlu (3D) yüz görüntüsü	70
Şekil 3.17 2 boyutlu (2D) yüz örnekleri, z derinlik değerlerinin gradyen genlik ve açı histogramları .....	72
Şekil 3.18 3 boyutlu (3D) yüz örnekleri, z derinlik değerlerinin gradyen genlik ve açı histogramları .....	72
Şekil 3.19 3D yüz görüntüleri derinlik bilgisi gradyen genlik ve açı histogram- larının ortalama, ortanca, standart sapma analiz grafiği .....	73
Şekil 3.20 Intel realsense D435 3D derinlik kamerası .....	74
Şekil 3.21 Göz kırpma analizi işlem adımları .....	75

Şekil 3.22	Sol üst: Göz açıkken göz karakteristik noktalarının görselleştirilmesi, Sağ üst: Göz kapalıyken gözün karakteristik noktaları, Alt: Göz en-boy oranı, göz en-boy oranındaki düşüş bir göz kırpmayı gösterir. ....	76
Şekil 4.1	GeorgiaTech yüz tanıma veriseti .....	80
Şekil 4.2	GeorgiaTech yüz tanıma veriseti sınıf içi sınıf dışı öklid uzaklık analizi	80
Şekil 4.3	GeorgiaTech yüz tanıma örnekleri .....	82
Şekil 4.4	Canlı yüz tespiti .....	83
Şekil 4.5	Canlı yüz tespiti .....	84
Şekil 4.6	Sahte yüz tespiti: yüz çıktısı (A4 Boyutu) .....	85
Şekil 4.7	Sahte yüz tespiti: yüz çıktısı (A5 Boyutu) .....	85
Şekil 4.8	Sahte yüz tespiti: vesikalık fotoğraf .....	86
Şekil 4.9	Sahte yüz tespiti: tablet görüntüsü .....	86
Şekil 4.10	Sahte yüz tespiti: bilgisayar görüntüsü .....	87
Şekil 4.11	Göz en boy oranı grafikleri .....	88
Şekil 4.12	Göz en boy oranı grafikleri .....	89

## ÇİZELGELER DİZİNİ

	<b>Sayfa</b>
Çizelge 2.1 Biyolojik sinir sistemi elemanları ve YSA karşılıkları.....	8
Çizelge 2.2 Toplama fonksiyonu türleri ve matematiksel ifadeleri.....	10
Çizelge 2.3 Aktivasyon fonksiyonu türleri ve matematiksel ifadeleri .....	11
Çizelge 3.1 Intel realsense D435 kamera özellikleri .....	74
Çizelge 4.1 Yüz tespit algoritmaları analizi: GPU üzerinde .....	79
Çizelge 4.2 Yüz tanıma algoritmaları analizi: GPU üzerinde .....	82
Çizelge 4.3 1 dakika süre boyunca göz kırpma sayıları ve ilk göz kırpma için geçen süre .....	90

## 1. GİRİŞ

İnsanın görme, işitme, düşünme gibi vasıflarının makinelere kazandırılması üzerine geçmişten günümüze pek çok çalışma yapılmıştır. Bir makinenin veya dijital bir sistemin insan beyni gibi çok karmaşık yapıya sahip fonksiyonlara sahip olması için yine insan beyninden esinlenilerek yapay sinir ağları geliştirilmiştir. Ancak yapay sinir ağlarının çok fazla işlem gücü gerektirmesi ve donanım yetersizliği çalışmaları yavaşlatmıştır. Özellikle son yıllarda bilgisayar donanımındaki önemli gelişmeler özellikle Grafik işlemci birimi (Graphics Processing Unit, GPU) gibi paralel ve hızlı işlemler yapmaya olanak sağlayan cihazların geliştirilmesi ile makine öğrenmesi, yapay sinir ağlarını daha ileri noktaya taşıyan derin öğrenme gibi alanlarda çok ciddi ilerlemeler meydana gelmiştir. Özellikle görüntünün analiz edilmesi, sınıflandırılması çok daha hızlı ve efektif bir şekilde yapılmaya başlanmıştır. Bilgisayarlara kazandırılmak istenen özelliklerden biri ise kişilerin tanınması işlevidir. Kişi tanıma daha çok yüz tanıma şeklinde karşımıza çıkmaktadır.

Yüz tanıma 1970'li yılların başından günümüze üzerinde pek çok çalışmanın yapıldığı ve çeşitli algoritmaların geliştirildiği biyometrik bir tekniktir. Yüz tanıma insan yüzünün karakteristik özelliklerinin dijital ortamda görüntü üzerinden analiz edilmesiyle bilgisayarlara görme yetisine benzer görme ve tanımlama niteliğinin kazandırılması şeklinde ifade edilebilir.

Özellikle son yıllarda bilgisayar donanım ve yazılımındaki gelişmelerle birlikte yüz tanıma sistemlerinin önemi daha da artmıştır. Yüz tanıma sistemleri pek çok alanda sıklıkla kullanılabilir bir hal almıştır. Güvenlik, sağlık, personel giriş kontrol sistemleri, gazetecilik, suçluların tespit edilmesi ve takibi gibi pek çok alanda yüz tanıma sistemleri kolaylıkla kullanılabilir bir hal almıştır. Kameralar da günlük hayatımızda oldukça yoğun yer almaktadır. Örneğin bir suçlu bir suç işlediğinde suç işlenen bölgedeki kameralar kişinin görüntüsünü alarak yüz tanıma sistemine iletir. Gelen görüntü yüz tanıma sisteminde analiz edilerek kişinin mevcut veritabanındaki kişilerden kim olduğu öğrenilir ve yine kameralarla kişi takip edilerek hızlı bir şekilde yakalanabilir. Bir başka senaryoda şirket personeli veya ziyaretçiler içeriye giriş yaparken kart okuma vb. yöntemlerle içeriye girmektedir. Kart

okuma sistemleri başarılı olsa da kişilerin kartı kaybetme ihtimali, kartın deforme olması vb. durumlar göz önüne alındığında yüz tanıma sistemlerinin bu sistemlere göre hem daha pratik hem de daha kullanışlı olduğu görülmektedir. Cep telefonlarının yüz tanıma ile tuş kilidinin güvenli bir şekilde açılması da kullanıldığı bir başka alandır. Bu şekilde örnekler çoğaltılabilmektedir. Yüz tanımada geçmişten günümüze kadar pek çok farklı yöntem ve teknik kullanılmıştır. Bu yöntemleri kişi görüntüsünü analiz ederek bir takım yüz özelliklerinin elde edildiği geleneksel yöntemler ve özellikle gelişen teknoloji ile yaygın bir şekilde kullanılmaya başlanan derin öğrenme tabanlı modern yöntemler olarak iki gruba ayrılabilir.

Geleneksel yöntemler tanınacak yüz için bir takım görüntü işleme ve analiz teknikleri ile yüzdeki benzersiz özellikleri çıkararak veri tabanındaki örneklerle kıyaslamakta ve nihai kararı vermektedir. Fisherfaces vektör algoritması ve temel bileşenler analizinde (Principal Component Analysis, PCA) kullanılan özyüzler (Eigenfaces), yerel ikili örüntüler (Local Binary Patterns, LBP) geleneksel yöntemlere örnek olarak verilebilir. Bu yöntemler yüz tanıma işlemlerinde belirli bir başarıma ulaşmış olsa da ışık, yüz açısının değişmesi, gözlük vb. aksesuarların kullanılması görüntü analizi ile elde edilen özelliklerin olumsuz etkilenmesine ve veri tabanındaki örneklerden farklılık göstermesine yol açmaktadır. Bu durum yüzlerin yanlış tanıma olasılığını arttırmaktadır. Bu etkiler göz önüne alındığında ve özellikle derin öğrenme, evrişimli sinir ağlarının hızlı bir gelişim göstermesi ile yüz tanımda ciddi bir sıçrayış gerçekleşmiştir.

Büyük miktarlarda verilerin olması ve daha güçlü Merkezi İşlem Birimi (Central Processing Unit, CPU) ve Grafik işlem birimi (Graphics Processing Unit, GPU) ünitelerinin geliştirilmesine bağlı olarak pek çok farklı ortamda ve koşullarda yüksek hassasiyetle çalışabilen daha güçlü yüz tanıma sistemleri geliştirilmiştir. Derin öğrenme tabanlı yüz tanıma sistemleri bunlara örnek verilebilir. Bu sistemler görüntülerden kişilerin yüz kısmına ait bir takım eşsiz özellikleri öğrenerek kişi özellik vektörleri üretmektedirler. Üretilen vektörler kişiye has nitelikleri barındırdıklarından K-en yakın komşuluk (K-Nearest Neighbours, K-NN), destek vektör makineleri (Support Vector Machine, SVM) vb. sınıflandırıcılar ile sınıflandırılarak yüz tanımda etkin rol oynamaktadırlar ve başarılı

sonular vermektedir. Aynı zamanda az veri ile yz tanıma yapabilmek kolaylaşmaktadır. Bu yntemlerin bir dięer avantajı ise ok farklı niteliklerdeki grsellerle eęitildiklerinden ıřık, aksesuar, yz aısı gibi faktrlerden daha az etkilenmesidir. Derin ęrenme modelleri eęitilmesi sonucunda verilerden nemli zellikleri ıkarmayı ğrenecek, daha nce grmedięi yeni bir grnt rneęinin ayırt edici zelliklerini ıkarma yeteneęine sahip olacaktır ve bu zellikleri veri kmesinde saklanan daha nceki rneklerle kıyaslayarak resmin kime ait olduęunu belirleyebilecektir.

Yz tanıma sistemleri pratik ve kullanıřlı olmasının yanında, bu sistemlerin de bir takım zlmesi gereken ynleri bulunmaktadır. Bu sistemler grnty analiz ederek alıřmaktadırlar. Sadece grntnn analiz edilerek sonuca ulařılması bu sistemlerin sahte yz rneklerine karřı zaafiyet gstermelerine yol amaktadır. rneęin bir yz tanıma sisteminde kiřinin normal fotoęrafı, videosu, tablet, cep telefonu, bilgisayar gibi cihazlardan fotoęrafı aynı řekilde video grnts gsterilerek yz tanıma sistemleri kolay bir řekilde yanıtılabilmektedir. Elbette bu problemin zlmesine ynelikte alıřmalar mevcuttur. Bu tez kapsamında hem derin ęrenme tabanlı bir yz tanıma sisteminin geliřtirilmesi hem de 3D gvenlik kamerasından alınan derinlik bilgisi ve kiřilerin gz kırpma hareket analizi kullanılarak yz tanıma sistemlerinin aldatılmasının nne geilmesi zerine yapılan alıřma anlatılacaktır.

Bu tez alıřması 5 blmden oluřmaktadır. Giriř blmnde genel bir bilgilendirme yapılmıřtır. Kuramsal Temeller ve Kaynak Arařtırması blmnde yz tespit ve tanıma iin kullanılan yntemler detaylı bir řekilde incelenmiř, derin ęrenme ve makine ęrenmesi teknikleri ile ilgili yntemler ile 3D kameralar hakkında bilgiler verilmiřtir. Materyal ve Yntem blmnde, derin ęrenme tabanlı yz tanıma sistemi ve canlı-cansız yz tespiti ile ilgili yapılan alıřmalardan bahsedilerek kullanılan yntemler detaylıca aıklanmıřtır. Bulgular blmnde yapılan alıřmalardan ve geliřtirilen gvenli yz tanıma sistemi ile elde edilen sonular ve farklı yz tespit ve tanıma modellerinin kıyaslanması ile canlılık tespiti ile ilgili analizlerden bahsedilmiřtir. Tartıřma ve Sonu blmnde tez alıřmasında kullanılan yntemlerin olumlu ve olumsuz yanları ile tez boyunca yapılan nemli alıřmalar anlatılarak, gelecek alıřmalara rnek olması aısından yorumlanmıřtır.

## 2. KURAMSAL TEMELLER VE KAYNAK ARAŞTIRMASI

Yüz tanıma sistemlerinde birbirinden farklı yöntemler kullanılmıştır. Yüz tanıma bilgisayarlı görü (Computer Vision, CV), görüntü işleme gibi farklı alanları kapsamaktadır. Son yıllarda güvenlik, personel devam kontrol sistemleri, kişi takibi vb. pek çok konuda kullanılmaya başlanmıştır. Bir yüz tanıma sistemi temel olarak yüz tespiti, yüz bölgesini seçme, yüz hizalama son adımında ise birtakım tekniklerle kişinin yüz fotoğrafından benzersiz özellikler çıkarma ve kişi tanıma adımlarından oluşmaktadır. Bir yüz tanıma sistemini daha ileriye taşıyacak önemli etken ise yüz tanıma saldırılarına karşı birtakım önlemler almaktır. Kişi yerine kişinin telefon, tablet veya bilgisayar gibi cihazlardan fotoğraf veya video görüntüsünü gösterme, vesikalık fotoğraf gösterme, kişiye benzer maske yapma gibi örnekler yüz tanıma sistemlerinin açıkları arasında gösterilebilir. Yüz tanımda kullanılan yöntem ve yüz tanıma sistemlerindeki saldırılara karşı uygulanan yöntem ile ilgili materyal ve yöntem bölümünde detaylı bilgi verilecektir.

Yüz tanıma sistemlerinin ilk evresi olan yüz tespiti için literatürde günümüze kadar birbirinden farklı pek çok yöntem sunulmuştur. Örneğin, Viola ve Jones (2001) tarafından önerilen gerçek zamanlı olarak nesne tespiti konusunda başarılı bir nesne tespit algoritmasıdır. Kullandığı Haar öznelikleri ile yüz bölgesi içeren ve içermeyen veriler kullanılarak, önceden eğitilmiş AdaBoost sınıflandırıcı yardımıyla yüz tespiti yapılmaktadır. Chang-yeon (2008) LBP özelliklerini kullanarak yüz tespiti uygulaması geliştirmiştir. LBP her pikselin çevresini merkez pikselin değeriyle eşleştirerek bir resmin piksellerini etiketleyen ve sonucu ikili sayı olarak kabul eden basit ama etkili bir doku operatörüdür. Her bir resim pikseli,  $3 \times 3$ 'lük bir kernel kullanılarak, kullanılan kernelin merkezinin karşılık geldiği merkez pikselin etrafındaki komşu bölgeleriyle kıyaslanıp  $u(x)$  basamak fonksiyonu kullanılarak ikili formatta etiketlenir. Yüz bölgesinin LBP histogramları çıkarılarak kaskat sınıflandırıcılar yardımıyla yüz tespiti yapılmaktadır.

Dalal ve Triggs (2005) tarafından önerilen yönlendirilmiş gradyanların histogramı (Histogram of Gradients, HOG) ile kişi tespiti adlı makaleden yola çıkılarak HOG özelliklerinin SVM vb. sınıflandırıcılar ile nesne tespit ve sınıflandırma, yüz tespiti vb. alanlarda

kullanılabileceği görülmüştür. Said ve ark. (2011) kişi tespiti için HOG ve SVM tabanlı bir yöntem önermiştir. Ranjan ve ark. (2015), Zhang ve ark. (2016) evrişimli sinir ağlarını (CNN) kullanarak yüz tespiti yapan bir model önermiştir. Sun ve ark. (2017) ise Faster R-CNN yaklaşımı ile yüz tespit çalışmalarına katkıda bulunmuştur. Tez kapsamında yüz tespiti için yapılan çalışmalarda hem HOG tabanlı hem de King (2015) tarafından önerilen CNN tabanlı yüz tespit yöntemi üzerinde durulmuştur.

Yüz tanıma için de literatürde pek çok çalışma mevcuttur. Turk ve Pentland (2001) temel bileşenler analizi (Principal Component Analysis, PCA) yaklaşımını kullanarak yüz tanıma yapan bir çalışma önermiştir. Özyüzler (Eigenfaces) olarak bilinen bu yaklaşım yüksek boyutlara sahip verilerin temel bileşenler analizi ile daha düşük boyutlarda temsil edilebilmesi için geliştirilmiş bir yöntemdir. Matris formatındaki görsellerin kovaryansları hesaplanarak öz değer ve öz vektörleri bulunur ve öz değerlerin yüksek olduğu değerlere karşılık gelen öz vektörler veritabanında saklanır. Elde edilen bu öz vektörler öz yüzlere karşılık gelmektedir. Belhumeur ve ark. (1997) yılında eigenfaces yaklaşımına alternatif olarak doğrusal ayırma analizi (Linear Discriminant Analysis, LDA) tabanlı Fisherfaces adlı yöntemi önermiştir. Bu yöntem aynı kişiye ait farklı görüntülerin sınıf içi dağılımının az, farklı kişilere ait görüntülerin sınıflar arası dağılımının ise fazla olması baz alınarak geliştirilmiştir. Lee ve ark. (2001) Fisherfaces algoritması ve elastik grafik eşleştirme (Elastic Graph Matching, EGA) yöntemi ile bir yüz tanıma modeli sunmuştur. Liu ve Wechsler (2002) yüz tanıma için Gabor özelliklerine dayalı Fisher LDA modelini önermiştir. Literatüre katkıda bulunan bir diğer yöntem He ve ark. (2005) tarafından önerilen Laplasyen yüzleri (Laplacian faces) yöntemidir. Yüz tanımada sıklıkla kullanılan bir diğer yöntem yerel ikili örüntüler (Local Binary Patterns, LBP) ise içerik özelliklerini kullanan Ahonen ve ark. (2006) tarafından önerilen yüz tanıma yöntemidir. Yüz bölgesinden LBP histogramları çıkarılarak yüz tanıma yapılan bir yöntemdir. Özellikle 2010'lu yıllardan sonra bilgisayar donanımındaki ve derin öğrenme tarafındaki gelişmelerden sonra yüz tanıma ESA (Evrişimli sinir ağı) tabanlı derin öğrenme kullanılmaya başlanmıştır. Taigman ve ark. (2014), Sun ve ark. (2015), Schroff ve ark. (2015) tarafından önerilen yüz tanıma modelleri derin öğrenme tabanlı modellerdir. Materyal ve Yöntem bölümünde



derin öğrenme tabanlı yüz tanıma sistemlerinden daha detaylı bahsedilecektir.

Kişi canlılık tespiti ile ilgili de göz kırpması, ağız hareketleri takibi, 3D derinlik analizi, görüntü içerik analizi vb. farklı yöntemler kullanılmaktadır. Tez kapsamında canlılık tespiti için 3D derinlik analizi ve göz kırpması özelliklerini kullanarak çalışmalar yapılmıştır. Yapılan çalışmalardan Materyal ve yöntem bölümünde detaylıca bahsedilecektir.

## **2.1. Yapay Sinir Ağları (Artificial Neural Network)**

Biyolojik bir sinir sistemi birbiri ile iletişim halinde olan sinir hücrelerinden oluşmaktadır. Sinir sistemi çok sayıda sinir hücresinden oluşmaktadır. Biyolojik sinir hücreleri birbirlerine bağlanarak farklı işlevleri yerine getirmektedirler. Beynimizde  $10^{10}$  adet sinir hücresi,  $6 \times 10^{10}$ 'dan fazla bağlantı bulunmaktadır. Bu kadar gelişmiş bir yapıya sahip olan insan beyninden çok daha az bir kapasiteye sahip bir makine geliştirerek bile hem bilgiyi işleyebilmek ve kontrol etmek hem de farklı görevlerde çok başarılı sonuçlar elde edilebilmektedir. Yapay sinir ağları insan beyninin öğrenme, karar verme işleyişinin bir kopyasının tasarlanması istenmesi amacıyla ortaya çıkmıştır. Yapay sinir ağları, insana beynine özgü yeni bilgiler üretme, bilgiyi farklı amaçlar için işleyebilme vb. pek çok özelliği otomatik bir şekilde gerçekleştirebilmek için geliştirilen modellerdir.

Yapay sinir ağlarıyla ilgili ilk çalışmalar 1940'lı yıllara dayanmaktadır. Culloch ve Pitts (1943) yapay sinir ağlarına öncülük eden ilk çalışmayı yayınlamışlardır. Bununla birlikte, XOR problemi olarak bilinen yapay sinir ağı türünün başarısızlığı nedeniyle, yapay sinir ağlarına olan ilgi belirli bir süre azalmıştır. 1970'li yıllara kadar XOR probleminin çözülmemesi nedeniyle yapay sinir ağlarının gelişimi durmuştur ancak 1970 yılı ve sonrasında bu problemin çözülmesi ile yapay sinir ağları ile ilgili çalışmalar tekrar ivme kazanmış ve teknikler gelişerek günümüze kadar ulaşmıştır. Özellikle bilgisayar donanımındaki gelişmelerle çok büyük işlem hacmi ve zaman gerektiren yapay sinir ağları kısa sürede ve paralel işlemlerle başarılı bir şekilde yapılmaya başlanmıştır.

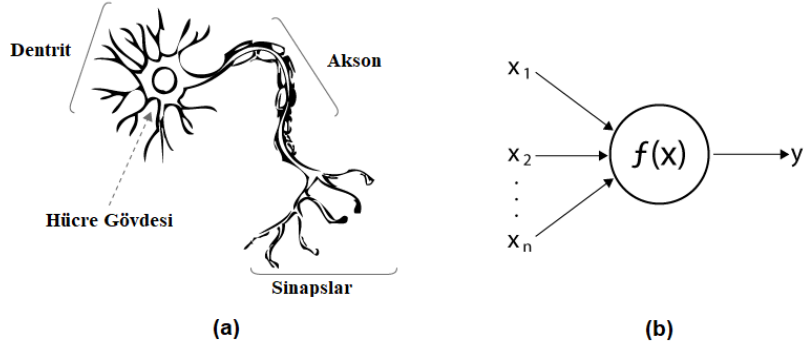
Şekil 2.1’de görüldüğü gibi yapay sinir ağları insanın sinir sistemi baz alınarak geliştirilmiştir. Biyolojik sinir hücresi: Hücre gövdesi, Dentrit, akson ve sinapslardan oluşmaktadır.

**Hücre Gövdesi:** Sinir hücresinin sitoplazma, organeler ve çekirdeğinin yer aldığı kısmıdır.

**Dendrit:** Sinir hücresinin gövdesinden çıkan kısa uzantılardır. Farklı sinir hücrelerinden veya çevreden gelen uyarıları hücre gövdesine iletmektedir.

**Akson:** Sinir hücresinin gövdesinden çıkan uzun bölümdür. Dentritler ile alınan ve hücre gövdesine iletilen uyarıları diğer sinir hücrelerinin dentritlerine taşımaktadır.

**Sinaps:** Bir sinir hücresinin aksonu ile farklı bir sinir hücresinin dentritinin birbirine bağlandığı kısımdır. Bir sinir hücresindeki uyarılar diğer sinir hücrelerine sinapslardan salgılanan hormonlar aracılığıyla iletilmektedir.



**Şekil 2.1.** (a) Biyolojik sinir hücresi (b) Yapay sinir hücresi

Akson uçlarının her biri farklı hücrelerle birleşmektedir. İki hücre arasındaki bilgi alış verişini sinapslar tarafından salgılanan neurotransmitterler yolu ile sağlanmaktadır. Bu şekilde milyarlarca sinir hücresi bir araya gelerek sinir sistemini oluşturmaktadır.

Sinir sistemi hücreleri birbirleriyle sürekli etkileşim halindedirler. Bir sinir hücresine başka

bir hücreden gelen mesajlar dentritler kullanılarak sinir hücresinin gövdesine iletilmektedir. Daha sonra aksonlar ve akson uçlarındaki sinapslar ile diğer hücrelerle bağlantılar kurularak gelen mesaj diğer hücrelere aktarılır. Bir hücrenin akson ucu ile diğer hücrenin dentriti arasında sinaptik boşluklar bulunmaktadır. Sinaptik boşluklarda gelen uyarımların diğer hücrelere geçmesini sağlayan sinaptik kesecikler bulunmaktadır. Sinaptik kesecikler tarafından salgılanan nöroiletken olarak adlandırılan maddeler sinaptik boşluğu doldurarak uyarımların diğer hücrelere geçişini ayarlamaktadır. Gelen bilgilerle bağlantılı olarak sinir hücreleri arasındaki bağlantıların değişmesi veya yeni bağlantıların kurulması insanın öğrenme süreci olarak tanımlanmaktadır.

Biyolojik sinir sistemine benzer şekilde yapay sinir ağı hücresi de girdiler, ağırlıklar, toplama veya birleştirme fonksiyonu, aktivasyon fonksiyonu ve çıkış bölümlerinden oluşmaktadır.

Çizelge 2.1’de sinir sistemi ile yapay sinir ağındaki yapıların birbirlerine karşılık gelen durumları gösterilmiştir.

**Çizelge 2.1.** Biyolojik sinir sistemi elemanları ve YSA karşılıkları

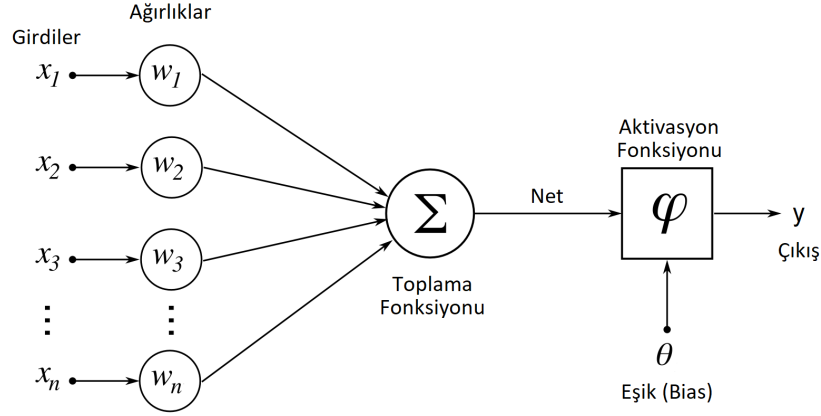
<b>Sinir Sistemi</b>	<b>Yapay Sinir Ağı</b>
Sinir	Yapay Sinir
Sinaps	Ağırlıklar
Dentrit	Toplama Fonksiyonu
Hücre Gövdesi	Aktivasyon Fonksiyonu
Aksonlar	Çıkış

### **2.1.1. Yapay Sinir Hücre Yapısı**

Yapay sinir ağlarının da biyolojik sinir sisteminin sinir hücrelerine benzer yapay sinir hücreleri vardır. Yapay sinir hücreleri işlem elemanı olarak tanımlanmaktadır. Yapay sinir hücrelerinin 5 temel elemanı bulunmaktadır. Bunlar:

- Girdiler
- Ağırlıklar
- Toplama Fonksiyonu

- Aktivasyon Fonksiyonu
- Çıktı



**Şekil 2.2.** Yapay sinir ağı hücre modeli

### 2.1.2. Girdiler (Inputs)

Bir yapay sinir hücresine dışarıdan gelen ve yapay sinir ağının öğrenmesi istenilen bilgilerdir. Çözülmek istenen probleme göre değişkenlik gösterebilir. Örneğin görüntü üzerine yapılan çalışmada girdiler fotoğraflar iken, doğal dil işleme ile ilgili çözülmek istenen bir problemde girdiler metin dizileri olmaktadır. Girdiler alana ve probleme göre farklılık göstermektedir.

### 2.1.3. Ağırlıklar (Weights)

Ağırlıklar yapay sinir hücresine gelen girdilerin önem derecesini ve sinir hücresindeki bağlantıların gücünü temsil etmektedir. Şekil 2.2’de  $w_1$  ağırlığı  $x_1$  girdisinin hücre üzerindeki etkisini göstermektedir. Ağırlıklar giriş değerinin önemini arttırıp azaltabilir. Ağırlığın büyük olması giriş değerinin önemini arttırırken, ağırlık azaldıkça giriş değerinin önemi azalmaktadır. Sıfıra yakın ağırlıklar, bu girdinin değiştirilmesinin çıktıyı değiştirmeyeceği anlamına gelir. Negatif ağırlıklar, bu girdinin artırılmasının çıktıyı azaltacağı anlamına gelir. Ağırlık, girdinin çıktı üzerinde ne kadar etkisi olacağına karar verir.

#### 2.1.4. Toplama Fonksiyonu (Sum Function)

Toplama fonksiyonu hücreye gelen net girdi fonksiyonunun hesaplanmasını sağlamaktadır. Net girdinin hesaplanmasında birbirinden farklı pek çok toplama fonksiyonu kullanılabilir. En yaygın kullanılanı ağırlık toplam fonksiyonudur, bu fonksiyonda her gelen girdi ağırlığıyla çarpılmakta ve sonuçları toplanmaktadır. Böylece ağırlık net girdisi hesaplanır.

**Çizelge 2.2.** Toplama fonksiyonu türleri ve matematiksel ifadeleri

Toplama Fonksiyonu	Denklem	Açıklama
Toplam Fonksiyonu	$Net = \sum_{i=1}^N X_i \cdot W_i$	Ağırlıklar giriş değerleriyle çarpılır ve elde edilen değerler toplanarak <i>Net</i> girdi hesaplanmaktadır.
Çarpım Fonksiyonu	$Net = \prod_{i=1}^N X_i \cdot W_i$	Ağırlıklar giriş değerleriyle çarpılır ve elde edilen değerler çarpılarak <i>Net</i> girdi hesaplanmaktadır.
Maksimum Fonksiyonu	$Net = Max(X_i \cdot W_i)$	N adet giriş içinden ağırlıklar giriş değerleriyle çarpılarak elde edilen sonuçların en büyüğü <i>Net</i> girdi olarak kabul edilmektedir.
Minimum Fonksiyonu	$Net = Min(X_i \cdot W_i)$	N adet giriş içinden ağırlıklar giriş değerleriyle çarpılarak elde edilen sonuçların en küçüğü <i>Net</i> girdi olarak kabul edilmektedir.
Çoğunluk Fonksiyonu	$Net = \sum_{i=1}^N Sgn(X_i \cdot W_i)$	N adet giriş içinden ağırlıklar giriş değerleriyle çarpılarak pozitif ile negatif örneklerin sayısı bulunur. Büyük değer <i>Net</i> girdi olarak kabul edilmektedir.
Kümülatif Toplam Fonksiyonu	$Net = Net_{eski} + \sum_{i=1}^N X_i \cdot W_i$	Girişler ağırlıklı olarak toplanır. Yeni giriş değerleri daha önceki girişlere eklenerek <i>Net</i> girdi hesaplanmaktadır.

Çizelge 2.2’de kullanılan toplama fonksiyonu örnekleri verilmiştir. Uygulamalarda en çok kullanılan girdi ve ağırlıkların çarpılıp toplandığı toplam fonksiyonudur.

### 2.1.5. Aktivasyon Fonksiyonu (Activation Function)

Aktivasyon fonksiyonu yapay sinir hücresine toplama fonksiyonundan gelen net girdiyi işleyerek hücrenin bu girdiden üreteceği çıktıyı belirlemektedir. Genellikle doğrusal olmayan bir fonksiyon türü seçilmektedir. Geri beslemeli ağlarda aktivasyon fonksiyonunun türevi kullanıldığı için bu fonksiyon seçilirken seçilen fonksiyonun türevinin kolay hesaplanabilir olmasına dikkat edilmelidir.

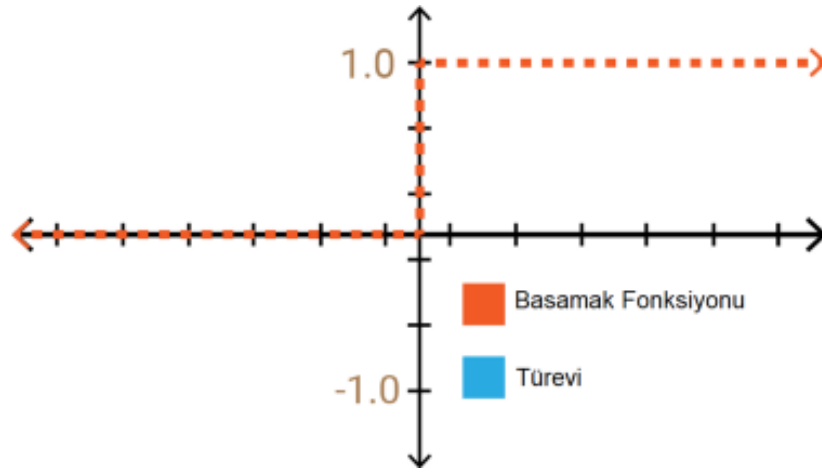
**Çizelge 2.3.** Aktivasyon fonksiyonu türleri ve matematiksel ifadeleri

<b>Aktivasyon Fonksiyonu</b>	<b>Denklem</b>	<b>Aralık</b>
Doğrusal Fonksiyon	$f(x) = x$	$(-\infty, \infty)$
Basamak Fonksiyonu	$f(x) = \begin{cases} x < 0 \text{ için } 0 \\ x \geq 0 \text{ için } 1 \end{cases}$	$\{0, 1\}$
Sigmoid Fonksiyon	$f(x) = \frac{1}{1+e^{-x}}$	$(0, 1)$
Hiperbolik Tanjant Fonksiyonu	$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	$(-1, 1)$
RELU	$f(x) = \begin{cases} x < 0 \text{ için } 0 \\ x \geq 0 \text{ için } x \end{cases}$	$[0, \infty]$
Leaky(Sızıntı) RELU	$f(x) = \begin{cases} x < 0 \text{ için } 0.01x \\ x \geq 0 \text{ için } x \end{cases}$	$(-\infty, \infty)$
Swish Fonksiyonu	$f(x) = x \cdot \text{sigmoid}(\beta x) = \frac{x}{1+e^{-\beta x}}$	$(-\infty, \infty)$

Çizelge 2.3’te aktivasyon fonksiyon türleri verilmiştir. Derin öğrenme uygulamalarında ağırlıklı olarak ara katmanlarda RELU aktivasyon fonksiyonu, ikili sınıflandırma problemlerinde sigmoid fonksiyonu, çok sınıflı sınıflandırma problemlerinde ise sigmoid aktivasyon fonksiyonu tercih edilmektedir.

**Adım (Step) Aktivasyon Fonksiyonu:** Hücre çıktısının girdinin belirli bir eşik değerinin altında olması durumunda 0, eşik değerinin üstünde olması durumunda ise 1 değerini aldığı ikili sınıflandırıcı olarak kullanılan bir fonksiyondur. Bu nedenle genellikle çıktı katmanlarında tercih edilmektedir. Çıkış değeri sabit ve adım aktivasyon fonksiyonunun türevi sıfır olduğu için geri yayılımda öğrenmeye katkısı olmamaktadır bu nedenle gizli katmanlarda bu fonksiyon türü kullanılmamaktadır.

Aşağıda Şekil 2.3'te adım fonksiyonu ve türevi gösterilmiştir. Çıkış değeri sabit olduğu için türevi sıfırdır.

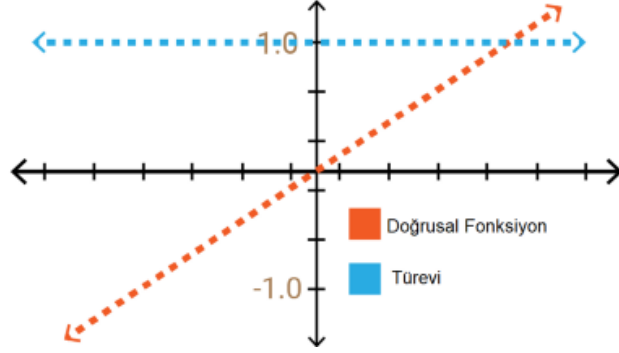


**Şekil 2.3.** Adım aktivasyon fonksiyon grafiği

**Doğrusal (Linear) Fonksiyon:** Genellikle doğrusal problemlerin çözümünde kullanılır. Her bir nöronun ağırlıklarıyla çarpılan girişleri alır ve girişle orantılı bir çıkış sinyali oluşturur. Adım fonksiyonundan daha iyi sonuçlar üretir çünkü birden fazla çıktıya olanak sağlamaktadır. Doğrusal fonksiyonun iki önemli dezavantajı bulunmaktadır. İlk olarak bir model eğitilmek istendiğinde geri yayılımda kullanmak mümkün değildir çünkü doğrusal fonksiyonun türevi sabittir ve giriş değeriyle bir ilişkisi yoktur. Bu nedenle geri gitmek mümkün değildir ve hangi giriş nöronuna bağlı ağırlıkların daha iyi tahmin ürettiği bilinmemektedir. Diğer bir problem ise doğrusal aktivasyon fonksiyonu kullanıldığında sinir ağında kaç katman olursa olsun son katman ilk katmanın bir doğrusal fonksiyonu olacaktır. Bu nedenle doğrusal aktivasyon fonksiyonu ile sinir ağı tek bir katmanlı yapıya

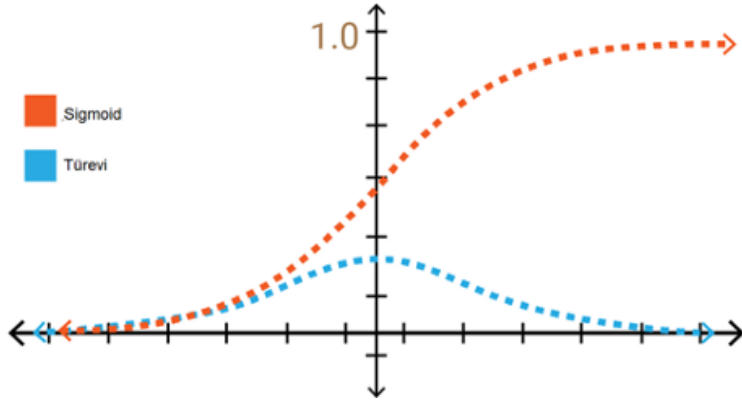
dönüşmektedir. Doğrusal fonksiyon denklemi :

$$A = c \cdot x \quad (2.1)$$



Şekil 2.4. Doğrusal fonksiyon grafiği

**Sigmoid Aktivasyon Fonksiyonu:** Doğadaki karşılaşılan pek çok problem doğrusal değildir. Doğrusal olmayan problemlerin çözümünde doğrusal olmayan fonksiyonlar kullanılmaktadır. Sigmoid aktivasyon fonksiyonu da doğrusal olmayan fonksiyonlardan biridir. Sigmoid fonksiyonu sürekli ve türevi alınabilir bir fonksiyon olduğundan yapay sinir ağı uygulamalarında sıklıkla kullanılabilir. Girdi değerlerine göre 0 ile 1 arasında çıkış değerleri üretmektedir. Bir girdinin olasılığının hesaplanması ile ilgili durumlarda sigmoid fonksiyonu kullanılabilir.



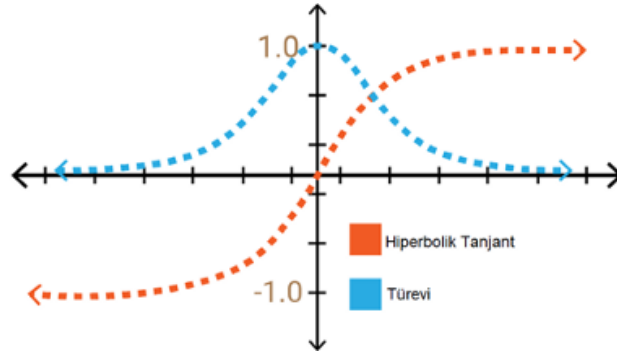
Şekil 2.5. Sigmoid fonksiyon grafiği

Şekil 2.5'te sigmoid aktivasyon fonksiyonu ve türevi gösterilmiştir. Grafik dikkatli incelendiğinde fonksiyonun bitiş uç noktalarında y değerleri sıfıra çok yakındır ve değişim



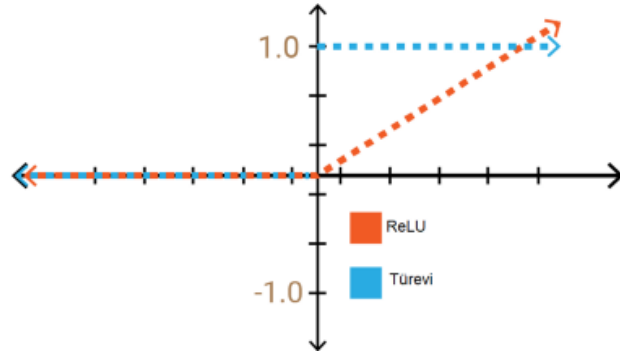
çok küçüktür. Bu bölgelerdeki türev değerleri çok küçüktür ve sifira yaklaşmaktadır. Bu durum kaybolan eğim (vanishing gradient) problemine yol açmaktadır. Bu durumda ilgili sinir hücrelerinde öğrenme yavaşlamakta hatta durmaktadır.

**Tanjant Hiperbolik Aktivasyon Fonksiyonu:** Sigmoid aktivasyon fonksiyonuna benzer bir yapısı vardır ancak fonksiyon  $[-1, +1]$  aralığında değerler üretmektedir. Sigmoid fonksiyonuna göre avantajı, türevinin daha dik olmasıdır, bu da daha fazla değer alabileceği anlamına gelir. Bu hızlı bir öğrenme için daha geniş bir yelpazeye sahip olduğu için daha verimli olacağı anlamına gelir.



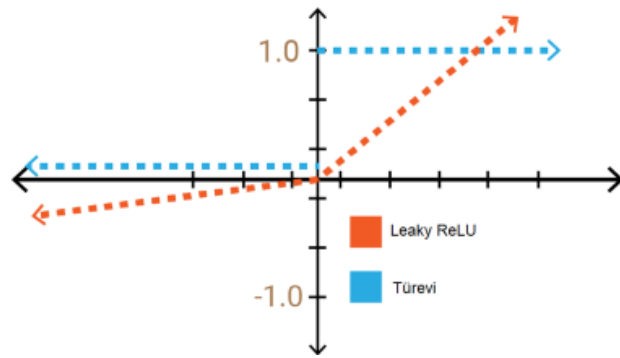
Şekil 2.6. Tanjant hiperbolik fonksiyon grafiği

**RELU (Rectified Linear Unit) Aktivasyon Fonksiyonu:** En yaygın kullanılan aktivasyon fonksiyonlarından biridir. ESA (Evrişimli sinir ağı) veya derin öğrenme modellerinde ara katmanlarda genellikle RELU aktivasyon fonksiyonu kullanılmaktadır. Çıktı olarak  $[0, \infty]$  aralığında değerler üretir. Türevi alınabilir bir fonksiyondur ancak negatif ekseninde tüm değerleri sıfırdır. Bu durum model eğitilirken modelin öğrenme kabiliyetinin azalmasına neden olmaktadır. Giriş değerleri sıfır veya negatif değer aldığı anda gradyen sıfır olur bu nedenle sinir ağı geri yayılım gösterememekte ve öğrenme gerçekleşmemektedir.



Şekil 2.7. ReLU fonksiyon grafiği

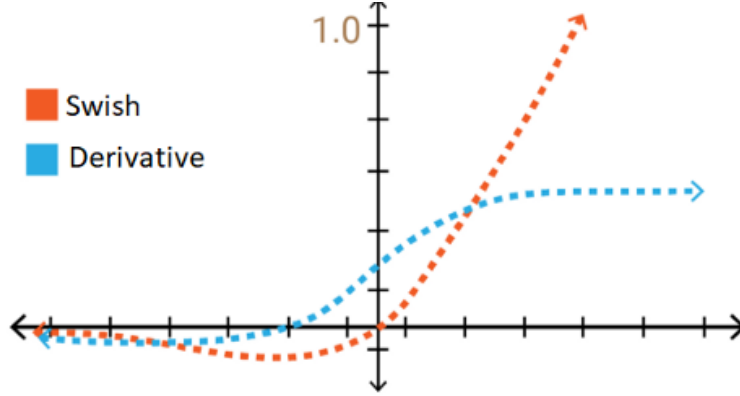
**Leaky (Sızıntı) ReLU Aktivasyon Fonksiyonu:** ReLU aktivasyon fonksiyonu negatif bölgede sıfır değerini almaktadır. Bu durum gradyenin sıfır olmasına ve geri yayılım olmamasına modelin öğrenememesine yol açmaktadır. Problemin çözümüne yönelik olarak Leaky (sızıntı) ReLU aktivasyon fonksiyonu önerilmiştir. Leaky ReLU aktivasyon fonksiyonu negatif bölgede çok küçük bir eğime sahiptir. Böylece gradyen hesaplanabildiği için geri yayılım negatif bölgede de gerçekleşmektedir. Sızıntı değeri olarak 0,01 verilmektedir. Leaky ReLU aktivasyon fonksiyonu  $[-\infty, +\infty]$  aralığında değer almaktadır. Negatif değerler için tutarlı tahminler üretememektedir.



Şekil 2.8. Leaky ReLU fonksiyon grafiği

**Softmax Aktivasyon Fonksiyonu:** Sigmoid fonksiyonuna benzer bir yapıya sahiptir. Sınıflandırma problemlerinde başarılı sonuçlar üretmektedir. En önemli farkı sigmoid fonksiyonunda olduğu gibi çok sınıflı sınıflandırma problemlerinde özellikle modellerin çıkış katmanlarında tercih edilmesidir. Bir girdinin hangi sınıfa ait olduğu ile ilgili  $[0, 1]$  aralığında değerler üreterek sonuç vermektedir.

**Swish Aktivasyon Fonksiyonu:** Swish, Google'daki arařtırmacılar tarafından keřfedilen yeni, kendinden kapılı bir aktivasyon fonksiyonudur. RELU'den en önemli farkı negatif bölgede deęer almasıdır. Őekil 2.9'da SWISH fonksiyonu ve türev grafięi gösterilmiřtir. Grafik incelendięinde giriř deęerleri artsa bile swish fonksiyonu çıkıřı azalabilmektedir. Bu durum swish fonksiyonuna özgüdür ve bu yönü ile dięer aktivasyon fonksiyonlarından ayrılmaktadır.



**Őekil 2.9.** Swish fonksiyon grafięi

$$f(x) = \frac{x}{1 + e^{-x}} \quad (2.2)$$

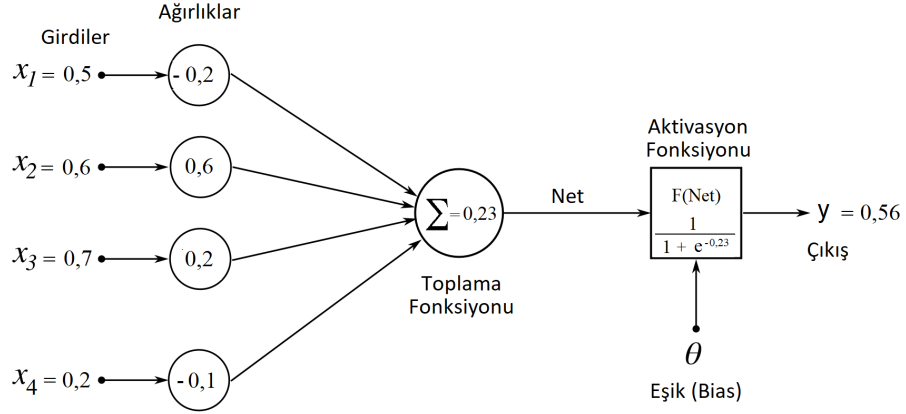
### 2.1.6. Hücre Çıktısı

Bu katman, aędaki son katmandır ve son gizli katmandan girdi alır. Bu katman ile istenilen sayıda ve istenilen aralıkta deęer elde edilebilmektedir. Üretilen çıktı dięer hücelere girdi olarak gönderilebilmektedir.

### 2.2. Yapay Sinir Hücresi Çalışma Prensipleri

Őekil 2.10'da yapay sinir hücresi hesaplama örneęi gösterilmiřtir. Örnekte gösterilen giriř deęerleri ve aęırlıklar rastgele seçilmiřtir. Bu deęerlerin gerçek deęerler olduęunu varsayalım. Örnekteki her bir aęırlık farklı deęere sahiptir. Bunun anlamı her aęırlığın deęerine göre her kanalın daha az ya da daha fazla öneme sahip olmasıdır. Aęırlığın fazla olduęu kanalda önem fazla iken az olduęu yerde kanalın önemi azalmaktadır. Örnekte

incelenirse  $x_2$  girdisinin olduğu kanal en fazla öneme sahip kanaldır çünkü en yüksek ağırlık değerine sahiptir. Ağırlığın negatif değer alması o kanalın sinyal üzerinde ters etki yapacağını göstermektedir.



**Şekil 2.10.** Yapay sinir hücresi hesaplama örneği

### 2.3. Yapay Sinir Ağı Modelleri

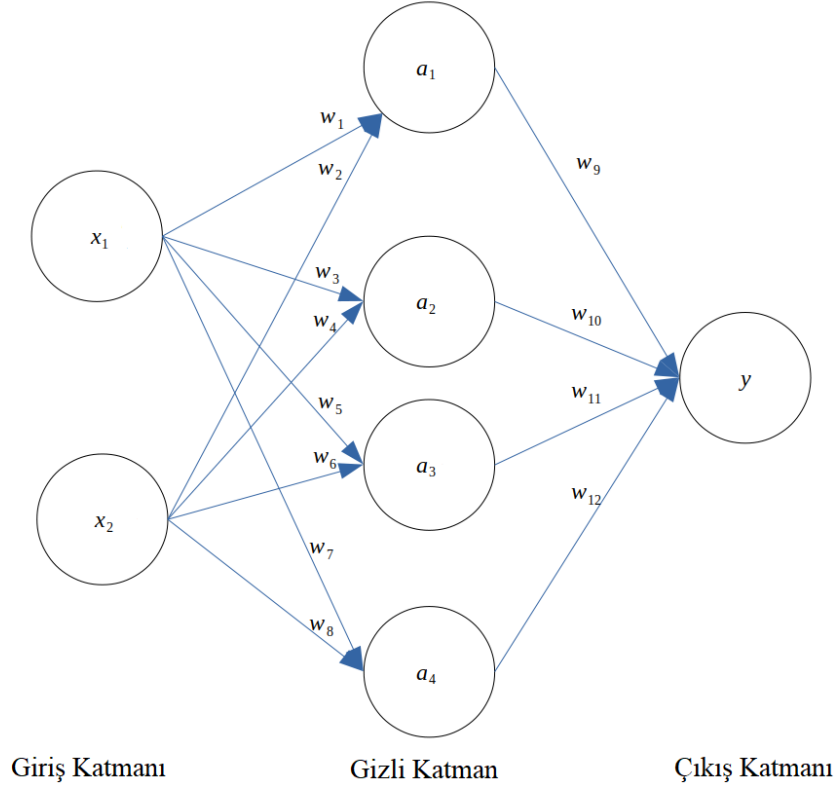
Yapay sinir ağları tek katmanlı algılayıcılar (perceptron), çok katmanlı algılayıcılar, ileri beslemeli yapay sinir ağları ve geri beslemeli yapay sinir ağları şeklinde 4 gruba ayrılabilir.

#### 2.3.1. Tek Katmanlı Algılayıcılar (Single Layer Perceptron)

Adından da anlaşılacağı üzere bu tür ağlar tek bir girdi ve bir çıktı katmanlarından oluşmaktadır. Her bir çıktı ünitesi her bir giriş ünitesine bağlıdır. Her bağlantı bir ağırlığa sahiptir. İlk önerilen sinir ağı modelidir. Tek katmanlı algılayıcılar önsel bilgiye sahip değildir, bu nedenle başlangıç ağırlıkları rastgele atanır. Tüm ağırlıklı girdileri toplar ve eğer toplam eşik üzerindeyse 1, altındaysa 0 değerini üretir. Her bir giriş ilgili ağırlıklarla çarpılıp toplandıktan sonra aktivasyon fonksiyonundan geçirilerek çıktı üretilir. En basit yapay sinir ağı türüdür. Ağın çıktısı bir veya sıfırdan oluşan mantıksal değerdir. İkili sınıflandırma problemleri için kullanılır. Çıktının hesaplanmasında eşik değer fonksiyonu kullanılır. Adaline algoritması bu tür ağlara örnek olarak verilebilir.

### 2.3.2. Çok Katmanlı Algılayıcılar (Multilayer Perceptron)

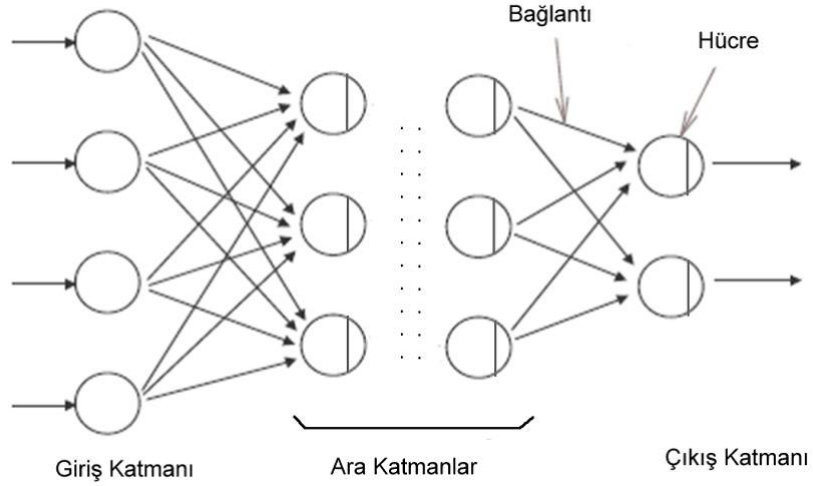
Çok katmanlı algılayıcılar; giriş, bir veya daha fazla gizli katman ve bir çıktıdan oluşan tek katmanlı algılayıcılara benzer yapıya sahip sinir ağı modelleridir. Tek katmanlı algılayıcılardan farklı olarak gizli katmana sahiptirler. Giriş katmanı gelen verileri ara katmana gönderir. Önceki katmanlardan gelen bilgiler bir sonraki katmana aktarılırlar. En az bir ara katman kullanılmaktadır, kullanılan ara katman sayısı probleme veya ihtiyaca göre değişmektedir. Her katmanın çıkışı bir sonraki katmanın girişi olmaktadır. Böylelikle çıkışa ulaşılmaktadır. Genellikle sınıflandırma problemlerinde kullanılır. Şekil 2.11’de çok katmanlı algılayıcı örneği gösterilmiştir. Örnek bir girdi, bir ara katman ve bir çıktı katmanından oluşmaktadır.



Şekil 2.11. Çok katmanlı algılayıcı örneği

### 2.3.3. İleri Beslemeli Yapay Sinir Ağları

Giriş sinyalleri girdi katmanından çıktı katmanına doğru tek yönlü bağlantılarla iletilmektedir yani tek yönlü sinyal akışına izin verilmektedir. Aynı katmanda bulunan nöronlar arasında bağlantı bulunmaz. Yani süreç girişten çıkışa doğru gerçekleşir ve sonlanır. Bu tür ağlarda bir katmanın çıkışı bir sonraki katmana girdi olarak iletilmektedir. Bilgiler ara katmanlar ve çıkış katmanlarında işlenerek sinir ağının çıkışı üretilmektedir.

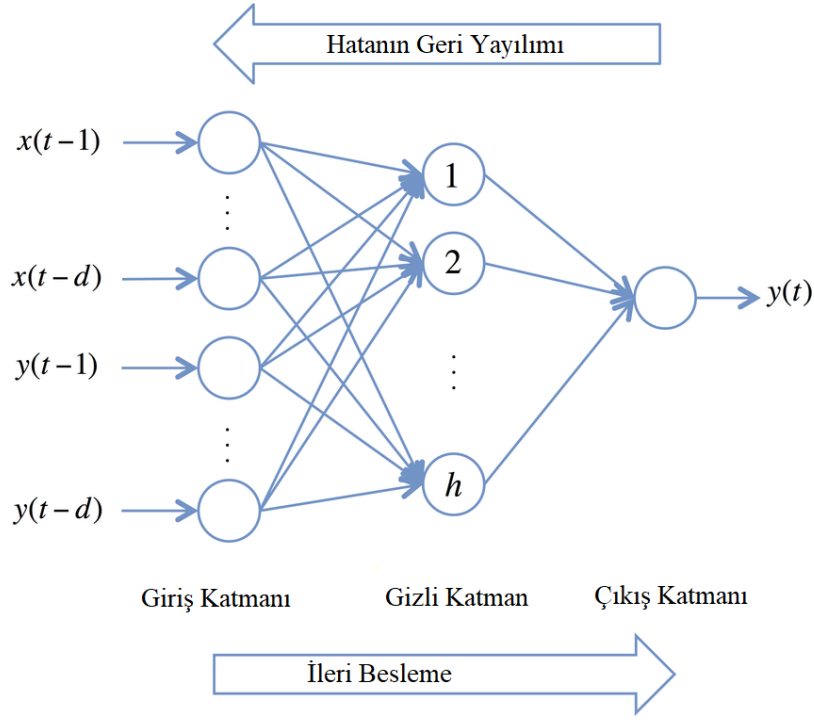


Şekil 2.12. İleri beslemeli yapay sinir ağı (Öztemel 2006)

### 2.3.4. Geri Beslemeli Yapay Sinir Ağları

Geri beslemeli yapay sinir ağı modellerinde girdilere veya önceki ara katmanlara bağlı olarak çıkış ve ara katman çıkışları hesaplandıktan sonra çıkışların girişi veya önceki ara katmanları geri beslediği bir yapıya sahiptir. Böylece hem ileri yönde hem de geri yönde veri aktarımı sağlanmaktadır. Şekil 2.13'te geri beslemeli bir ağ örneği gösterilmiştir.

Yapay sinir ağları eğitilirken 3 farklı yöntem kullanılmaktadır. Bu yöntemler gözetimli öğrenme (Supervised Learning), gözetimsiz öğrenme (Unsupervised Learning) ve yarı gözetimli öğrenme (Semi Supervised Learning) şeklindedir.

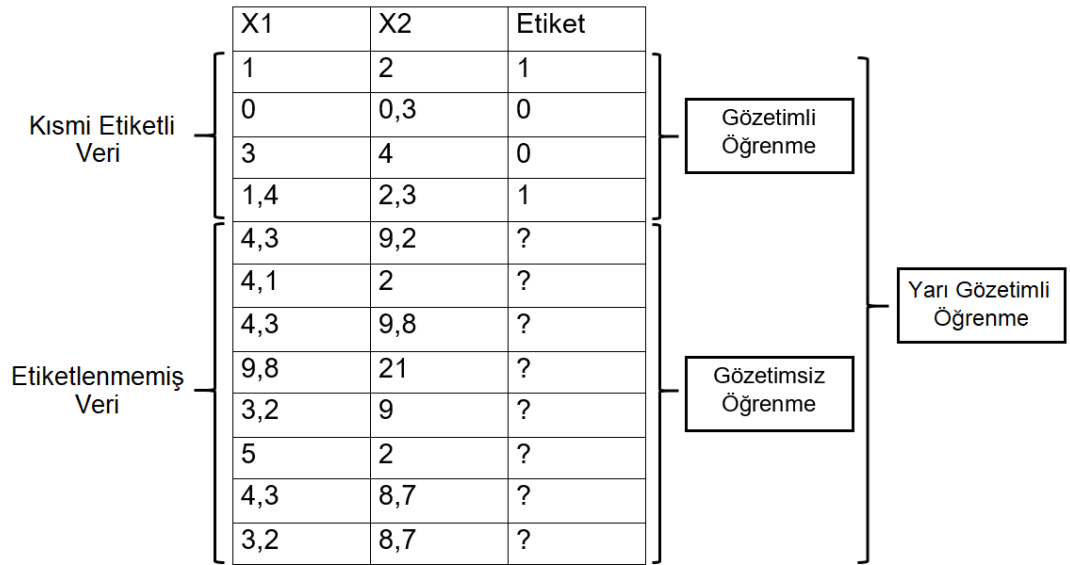


**Şekil 2.13.** Geri beslemeli yapay sinir ağı

**Gözetimli Öğrenme (Supervised Learning):** Gözetimli öğrenme belirli örnekler üzerinden çıktıları öğrenen ve model çıkararak, çıkardığı model üzerinden yeni örnekler hakkında tahminde bulunan bir öğrenme yöntemidir. Bir insanın öğrenme sürecini düşünecek olursak örneğin bir öğretmen öğrencilerine bir konuyu anlatmakta ve sonrasında konu ile ilgili örnek sorular çözmektedir. Öğrenciler örnek soruların çözümlerinden problemin çözümüne yönelik gerekli bilgiyi edinmektedir. Sonrasında öğrenciler karşılıklarına konuyla ilgili yeni bir örnek geldiğinde daha önceki çözdükleri örneklerden yola çıkarak problemin çözümünü bulmaktadır. Gözetimli öğrenmede de buna benzer şekilde mevcut verilerden öğrenme öz konusudur. Yapay sinir ağı eldeki örnekler ve bu örneklerin çıktılarına göre geri yayılım ile sürekli ağırlıklarını güncellemekte ve hatayı en aza indirmektedir. Bir başka gözetimli öğrenme örneği olarak bir sınıflandırma problemi olan fotoğraflardaki nesnelere tanımlanması verilebilir. Mevcut veri setindeki nesne sınıfları ve nesne koordinatları kullanılarak modelin nesnelere tanıması için model eğitilecektir böylece model nesnelere ve koordinatlarını öğrenerek görmediği örnekler için tahmin yapabilecektir.

**Gözetimsiz Öğrenme (Unsupervised Learning):** Gözetimsiz öğrenmede ise gözetimli öğrenmenin aksine sonuçlar yani etiketli veriler ağı verilmemektedir. Ağ giriş bilgilerine göre üretilen çıkış değerlerini birbiriyle kıyaslayarak verileri kümeleyip kendi içinde birtakım sınıflandırma kuralları oluşturmaktadır. Yani bu öğrenme yönteminde ağın öğrenmesi için herhangi bir gözetmene ihtiyaç yoktur. Yapay sinir ağına giriş verileri gösterilir ve daha sonra model örnekleri analiz ederek anlamlı çıktılar üreterek öğrenme işlemi gerçekleştirilir. Sınıflandırma problemleri için tercih edilen bir öğrenme yöntemidir.

**Yarı Gözetimli Öğrenme (Semi Supervised Learning):** Yarı gözetimli öğrenme algoritmaları, hem gözetimli hem de gözetimsiz algoritmaların özelliklerine sahip algoritmalarıdır. Bu algoritmalar gözetimli öğrenme algoritmalarına göre daha az etiketli veriye sahiptir. Öncelikle az etiketli veri üzerinden model birtakım özellikleri öğrenmektedir. Bu yönüyle gözetimli öğrenmeye benzemektedir. Daha sonra etiketlenmemiş veriler üzerinde çalıştırılarak modelin kendi kendine öğrenme süreci devam etmektedir bu yönüyle de gözetimsiz öğrenmeye benzemektedir.



**Şekil 2.14.** Yapay sinir ağı öğrenme türleri



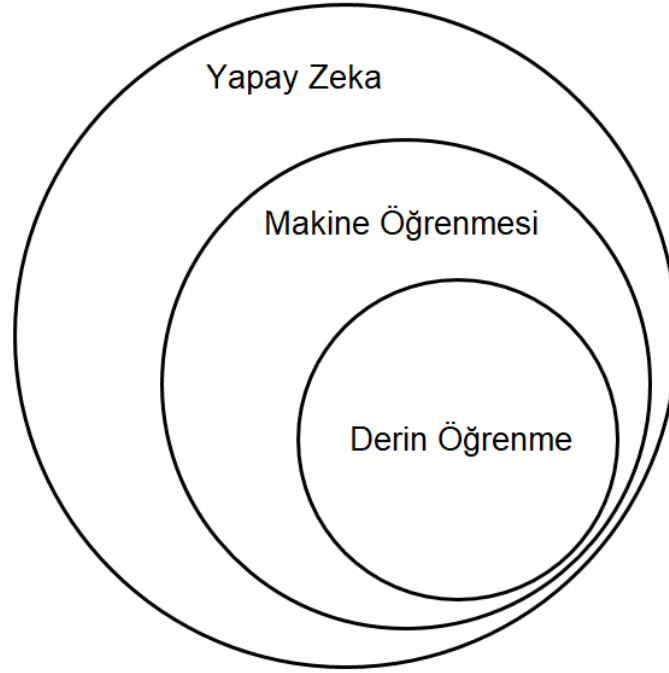
Şekil 2.14'te öğrenme türlerini görselleştiren bir örnek sunulmuştur. Yarı gözetimli öğrenme için model tüm verilere kıyasla az olan etiketli veri ile öğrenme işlemini gerçekleştirmektedir. Daha sonra tüm verilerin çoğunluğunu oluşturan etiketsiz veri ile gözetimsiz öğrenme uygulanmaktadır.

#### **2.4. Derin Öğrenme (Deep Learning)**

Yapay zekâ, çeşitli görevleri yerine getirmek için verilerden yararlanan ve bu verilerden çeşitli özellikleri öğrenerek kendini sürekli geliştiren insan beynine benzeyen sistemlerdir. İlk defa 1950'li yılların başlarında ortaya çıkmıştır. Yapay zeka veriyi işleyerek elde ettiği sonuçlar ile beklenen sonuçlar arasındaki farklılıklardan kaynaklanan hatalardan pek çok önemli görevi doğru ve başarılı bir şekilde öğrenebildiği için kendini sürekli geliştirebilen bir yapıya sahiptir. 1980'li yıllarda ortaya çıkan makine öğrenimi ise mevcut bir veri kümesini birtakım işlemlerden geçirerek verilerden tahminlerde bulunan, veriyi sınıflandıran yöntemleri kapsayan bir yapay zeka alt dalıdır. Yapay zeka da makine öğrenmesi de özellikle çok fazla işlem gücü gerektirmesi, donanım yetersizliği gibi nedenlerle uzun yıllar gelişme gösterememiştir. Ancak 1990'lı yıllara gelindiğinde veri madenciliği ile ilgili çalışmalar popülerliğini arttırmaya başlamış ve 2000'li yıllarda ise derin öğrenme ile ilgili pek çok gelişme yaşanmıştır.

Derin öğrenme bilgisayar algoritmalarını inceleyerek kendi başına öğrenmeye ve geliştirmeye dayalı bir alandır. Makine öğrenimi daha basit kavramlar kullanırken, derin öğrenme, insanların nasıl düşündüğünü ve öğrenme yetilerini taklit etmek amacıyla tasarlanmış yapay sinir ağlarıyla çalışmaktadır. Yakın zamana kadar yapay sinir ağları karmaşık yapısı ve çok fazla işlem gücü gerektirmesi nedeniyle sınırlı gelişim göstermiştir. Bununla birlikte büyük veri (big data) alanındaki ve donanımdaki gelişmeler, bilgisayarların çok karmaşık yapıları insanlardan daha hızlı analiz etmesine, öğrenmesine ve tepki vermesine olanak tanıyan daha derin ve karmaşık sinir ağlarının geliştirilmesine olanak sağlamıştır. Derin öğrenme görüntü sınıflandırma, dil çevirisi ve ses tanıma gibi alanların gelişimine önemli katkılar sağlamıştır. İnsan müdahalesi olmadan herhangi bir örüntü tanıma problemini çözmek için kullanılabilir. Derin öğrenme modellerinin gelişimi temel olarak

yapay sinir ađlarına dayanmaktadır. Yapay sinir ađları çok sayıda parametre içermektedir. Çok sayıda parametre içermesinin yanında çok fazla işlem gücü gerektirmektedir. Özellikle donanımdaki yetersizlikler nedeniyle çok fazla katmana sahip sinir ađları geliştirilememektedir. Derin öğrenme ise özellikle bilgisayar donanımındaki gelişmelerle birlikte yapay sinir ađlarının katman sayılarının arttırılmasıyla çok daha derin mimarilerin elde edilmesiyle ortaya çıkmaktadır. Yapay sinir ađları birkaç katmandan oluşabilirken derin öğrenme ile birlikte 150 katmanlı mimarilerin daha da ötesinde modeller üretilebilmekte ve daha başarılı sonuçlar elde edilebilmektedir. Yapılandırılmamış veya etiketlenmemiş verilerden denetimsiz olarak öğrenebilen ađlara sahip, yapay zekadaki makine öğreniminin bir alt kümesidir. Derin sinir öğrenme veya derin sinir ađı olarak da bilinir. Derin öğrenme, nesne tespiti, ses tanıma, dil çevirileri, karar verme gibi işlevleri yerine getirmek için veriyi işleyen insan beyninin çalışma şeklini taklit eden yapay zekanın onun da daha ötesinde makine öğrenmesinin bir alt dalıdır.



**Şekil 2.15.** Derin öğrenme, makine öğrenmesi ve yapay zeka

Şekil 2.15’ da derin öğrenmenin makine öğrenmesinin, makine öğrenmesinin de yapay zekanın bir alt dalı olduğu gösterilmiştir.

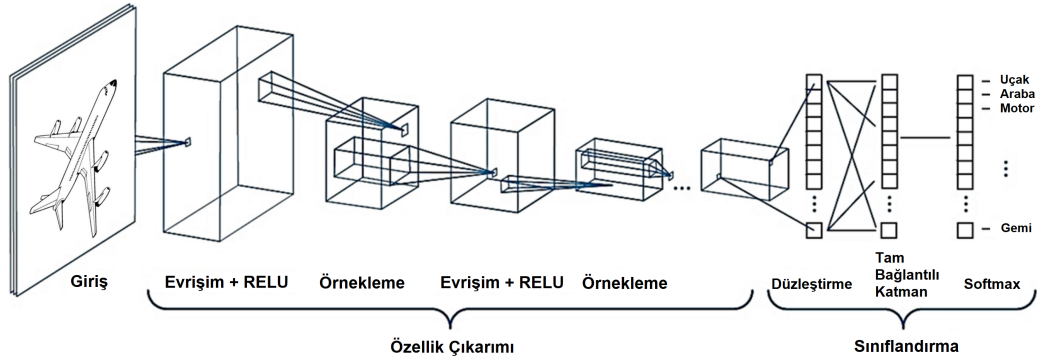
Derin Sinir Ağları (DNN), her katmanın görüntüleri, sesi ve metni anlamlandıran temsil ve soyutlama gibi karmaşık işlemleri gerçekleştirebildiği türden ağlardır. Makine öğrenimi teknikleri içinde en hızlı gelişen alanlardan biri olarak kabul edilen derin öğrenme, ciddi bir şekilde dijital teknolojiyi temsil etmekte ve pek çok kurum veya kişi tarafından yaygın bir şekilde yeni iş modelleri oluşturmak için kullanılmaktadır. Önceki bölümlerde sinir ağlarının, tıpkı insan beyninin nöronlardan oluşması gibi düğümlerden oluştuğundan bahsedilmiştir. Her bir katmandaki düğümler sonraki katmanlara bağlanmaktadır. Ağın, sahip olduğu katman sayısına göre o ağın ne kadar derin olduğu söylenebilmektedir. Genel olarak 10 veya daha fazla katmana sahip modeller derin öğrenme modeli olarak sınıflandırılmaktadır. Derin öğrenme sistemleri, çok fazla veriyi işledikleri için ve karmaşık matematiksel hesaplamalar içerdikleri için güçlü donanıma ihtiyaç duymaktadır. Çok gelişmiş donanımlarda bile bir derin öğrenme modelinin eğitimi veriye de bağlı olarak haftalarca sürebilmektedir. Derin öğrenme sistemleri, doğru sonuçlar elde etmek için büyük miktarda veri gerektirir yani bu sistemler büyük veri setlerinden yararlanarak daha iyi sonuçlar vermektedir. Örneğin, bir yüz tanıma sistemi, yüzün önemli ayırt edici özelliklerini algılamayı ve tanımayı öğrenerek çalışmaktadır. Zamanla sistem mevcut verilerle kendi kendini eğitmekte ve daha başarılı sonuçlar vermektedir. Böylece yüz tanıma sistemi, zamana yüzleri ayırt etmeyi öğrenebilecektir. Bir başka örnekte ise amaç bir sinir ağının köpek içeren fotoğrafları tanımasını sağlamak olsun. Görsellerdeki köpekler birbirinden farklı özelliklere sahip olabilmektedir. Ayrıca görsellerdeki köpeklerin açıları, görselin ışık seviyesi, gölge vb. özellikleri birbirinden farklılık gösterebilmektedir. Bu durum sonuçları doğrudan etkilemektedir. Bu nedenle karşılaşılan bu problemleri çözmek için, farklı ışık, açı, gölge vb. özelliklere sahip görüntü örnekleri toplanmalıdır. Örnek sayısı mümkün olduğunca fazla olmalıdır. Ayrıca "köpek" olarak etiketlenecek birçok köpek yüzü örneği ve köpek olmayan nesnelerin "köpek değil" olarak etiketlenmiş resimleri de dahil olmak üzere bir eğitim resim seti derlenmelidir. Derin öğrenme ya da yapay sinir ağı modelleri veriyi bilgisayarların anlayabileceği formatta yani sayısal formatta işlemektedir. Bu nedenle resim verisi modellerin anlayabileceği şekilde sayısal veriye dönüştürülmektedir. Veriler derin öğrenme ağından geçirilir. Bunun sonucunda modelde ağın öğrendiği farklı özelliklere karşılık gelen değerler ağırlık olarak atanmaktadır. En

son çıkış katmanı elde edilen özellikleri derleyerek çıktı üretmektedir. Böylece giriş görüntüsündeki nesnenin köpek olup olmadığına karar vermektedir. Sonraki adımda ise derin öğrenme modelinden alınan cevap insan tarafından oluşturulan etiketle kıyaslanarak eşleşme olması durumunda çıktı onaylanır, eşleşme olmazsa hata hesaplanır ve ağırlıklar geri besleme ve optimizasyon yöntemleriyle yeniden düzenlenir. Model ağırlıklarını tekrar tekrar ayarlayarak köpek tanıma becerilerini geliştirmeye çalışır. Bu teknik bir gözetimli öğrenme tekniğidir. Son dönemlerde çok sayıda önemli gelişme yaşanmasına rağmen derin öğrenme hala başlangıç aşamasındadır. Ancak buna rağmen pek çok alanda kullanılmaktadır. Örneğin kendi kendine gitme özelliğine sahip otonom araçlar üzerinde çalışmalar yapılmakta ve bu araçlar günlük hayatta yer almaya başlamıştır. Karmaşık sinir ağları tabanlı derin öğrenme modelleri kullanılarak, kaçınılması gereken nesnelere belirlemek, trafik ışıklarını tanımak ve hızı ayarlamak gibi işlevleri yerine getirebilmek için modeller eğitilmektedir. Sinir ağları, hisse senedi fiyatlarından hava durumuna kadar pek çok şeyi tahmin etmede ustalaşmaktadır. Derin öğrenme uygulamaları, hastalar için kanıta dayalı tedavi planları tasarlama ve kanserleri erken tespit etmeye yardımcı olma becerisini geliştirerek tıp alanında da kullanılabilir.

## **2.5. Evrişimli Sinir Ağları (Convolutional Neurat Network, CNN)**

Evrişimli sinir ağları (ESA), bir girdi görüntüsünü alarak, görüntüdeki nesnelere ayırt edici önemli özelliklerini öğrenebilen ve görüntüleri, görüntülerdeki nesnelere birbirinden ayırt edebilen bir derin öğrenme algoritmasıdır. Evrişimli sinir ağları ile ilgili ilk çalışmalar Facebook yapay zeka araştırma grubu direktörü Yann LeCun tarafından 1988 yılında yayınlanmıştır. Yan Lecun LeNet adlı ilk evrişimli sinir ağı modelini tasarlamıştır. LeNet, posta kodları ve rakamları okumak gibi karakter tanıma görevleri için kullanılmıştır. Evrişimli Sinir Ağları günümüzde özellikle görüntü ile ilgili uygulamalarda çok yaygın bir şekilde kullanılmaktadır. Hesaplama maliyetini ve sinir ağlarındaki karmaşıklığı azaltması nedeniyle popülerliği artmıştır. 2012 yılında ESA tabanlı AlexNet derin öğrenme modelinin yayınlanması ile birlikte ESA'lara ilgi daha da artmıştır. Kısa zaman içerisinde 8 ESA katmanına sahip AlexNet mimarisinden 152 ESA katmanlı ResNet mimarisine hızlı gelişmeler yaşanmıştır. Görüntü ile ilgili pek çok problemin çözülmesinde kullanılan

ESA'larda doğruluk açısından çok başarılı modeller geliştirilmiştir. ESA'ların en büyük avantajı görüntülerdeki önemli özellikleri herhangi bir insan denetimi olmadan otomatik olarak algılayabilmesidir. ESA ayrıca hesaplama açısından da oldukça etkilidir. Yapay sinir ağlarına kıyasla hesaplama maliyetini düşürmüştür böylece çok katmanlı mimarilerin de geliştirilmesine olanak sağlamıştır. Şekil 2.16'da ESA tabanlı bir derin öğrenme modeli örneği verilmiştir. Örnekte bir giriş görüntüsü ESA katmanlarından geçirildikten sonra giriş görüntüsüne ait özellikler çıkarılmaktadır. Daha sonra elde edilen özellikler vektörel formatta bir yapay sinir ağına iletilmektedir. Çıkış katmanında ise Softmax sınıflandırıcısı ile görüntünün sınıfı tahmin edilmektedir.



Şekil 2.16. Evrişimli sinir ağı örneği

## 2.6. Evrişimli Sinir Ağı Mimarisi

Şekil 2.16'da görüldüğü gibi tüm ESA modelleri birbirine benzer bir mimaride tasarlanmaktadır. Üzerinde çalışılan bir giriş görüntüsü bulunmaktadır. Giriş görüntüsü birbirini takip eden sıralı evrişim ve örnekleme (pooling) katmanlarından geçirildikten elde edilen özellikler sonra tam bağlantılı katmana (Fully connected layer) iletilmektedir ve çözülmek istenen probleme göre çıktı elde edilmektedir. Eğer problem çok sınıflı bir sınıflandırma örneği ise çıkış katmanında softmax sınıflandırıcı kullanılmaktadır.

ESA'larda kilit rol oynayan katmanlar şu şekildedir:

- Evrişimli katman (Convolutional Layer)
- Aktivasyon katmanı (Activation Layer)
- Örnekleme Katmanı (Pooling Layer)
- Tam bağlantılı katman (Fully Connected Layer)
- Yığın Normalleştirme (Batch Normalization)
- Seyreltme (Dropout)

**Evrişimli Katman (Convolutional Layer):** Evrişim (Convolution) işlemi, iki matrisin eleman bazında çarpımı ve ardından bir toplanmasıdır. ESA'nın ana yapı taşı evrişimli katmandır. Evrişim, iki bilgi kümesini birleştirmeye yönelik matematiksel bir işlemdir.

Şekil 2.17'de sol tarafta evrişim katmanı girdi örneği, sağ tarafta ise evrişim filtresi örneği gösterilmiştir. Şekilde  $5 \times 5$ 'lik bir giriş,  $3 \times 3$ 'lük bir filtre örneği verilmiştir.

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

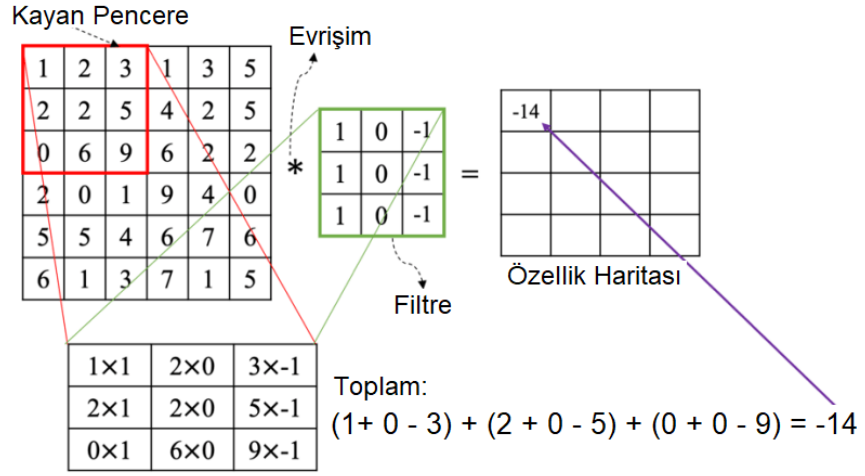
(a) Giriş

1	0	1
0	1	0
1	0	1

(b) Filtre

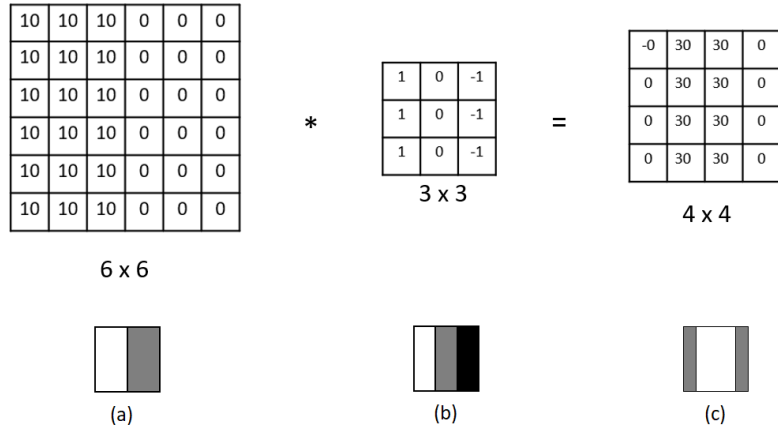
**Şekil 2.17.** Evrişim katmanı (a) Girdi örneği (b) Filtre örneği

Aşağıda Şekil 2.18'de gösterildiği gibi filtre giriş matrisinin üzerinden kaydırılarak evrişim işlemi gerçekleştirilmektedir. Her konumda, eleman bazında matris çarpımı yapıp sonuçlar toplanmaktadır. Elde edilen sonuçlar girişin özellik haritasını temsil etmektedir. Burada filtre giriş matrisinin sol üst tarafındadır, evrişim işleminin çıktısı -14 olarak elde edilmektedir. Bu değer özellik haritasında (1, 1) koordinatındaki değerini temsil etmektedir.



Şekil 2.18. Evrişim (Konvolüsyon) işlemi örneği

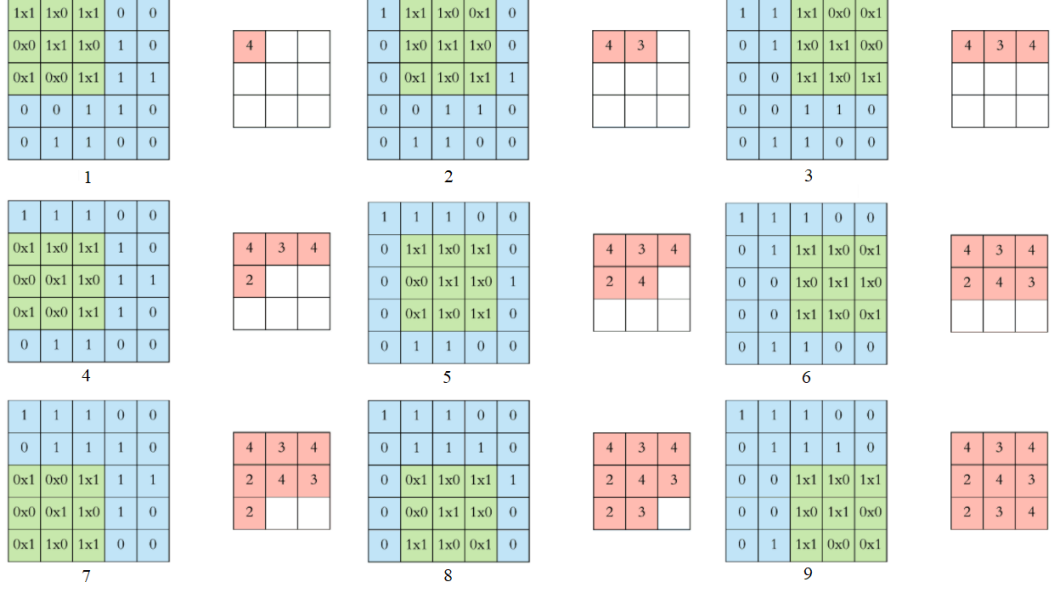
Şekil 2.19’da ise giriş görüntüsünden  $3 \times 3$ ’lük bir filtre kullanılarak evrişim işlemi sonucunda dikey kenarların bulunması örneği gösterilmiştir.



Şekil 2.19. Evrişim işlemi ile kenar bulma (a) Giriş (b) Filtre (c) Çıktı, sıfırdan büyük değerler beyaz, sıfıra eşit değerler gri, sıfırdan küçük değerler siyah ile gösterilmiştir

Burada kullanılan filtre görüntüdeki yoğunluk değerlerindeki ani değişiklikleri tespit etmeye duyarlıdır. Bu özelliği kullanılarak filtrenin dikey kenarları algılamasına olanak

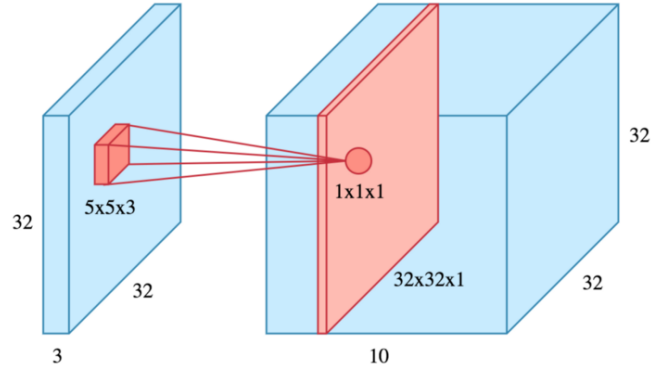
sağlamaktadır, bu nedenle elde edilen  $4 \times 4$ 'lük çıktıda asıl görüntüdeki dikey kenarların olduğu bölgelerde yüksek yoğunluklu değerler elde edilmektedir.



**Şekil 2.20.** Evrişim işlem basamakları

Şekil 2.20'de ise filtre kaydırılarak evrişim işlemi tekrarlanmaktadır. Elde edilen sonuç özellik haritasına eklenmektedir. Bu işlem tüm giriş matrisi/görüntüsü taranana kadar devam etmektedir. Burada  $3 \times 3$  filtre kullanılarak 2 boyutlu bir evrişim işlem örneği gösterilmiştir. Gerçekte ise tüm bu evrişim işlemleri 3 boyutta gerçekleştirilmektedir. Bir görüntü yükseklik, genişlik ve derinlik değerleriyle 3 boyutlu (3D) bir matris olarak temsil edilmektedir. Derinlik ise renk kanallarına (RGB) karşılık gelmektedir. Evrişim filtreleri ise,  $3 \times 3$  veya  $5 \times 5$  gibi belirli yükseklik ve genişliğe sahiptir ve giriş görüntüsünün derinliğini kapsayacak şekilde tasarlanır. Bu nedenle evrişim filtreleri de 3 boyutlu olmalıdır.

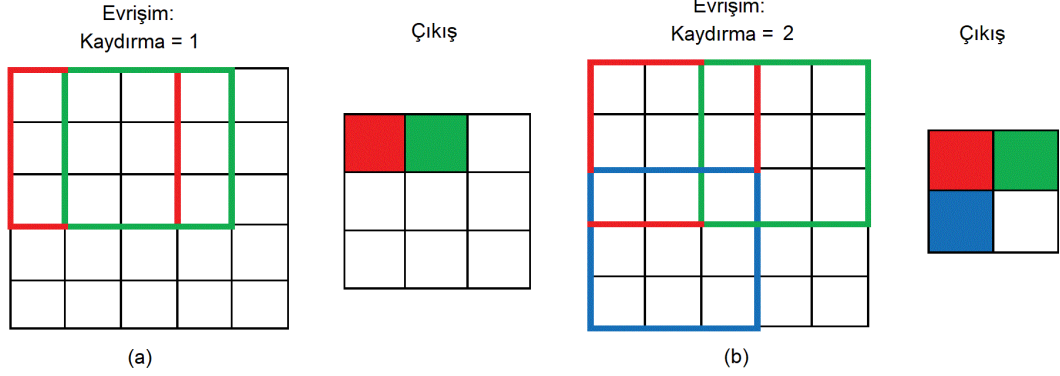




**Şekil 2.21.** 3D evrişim örneği

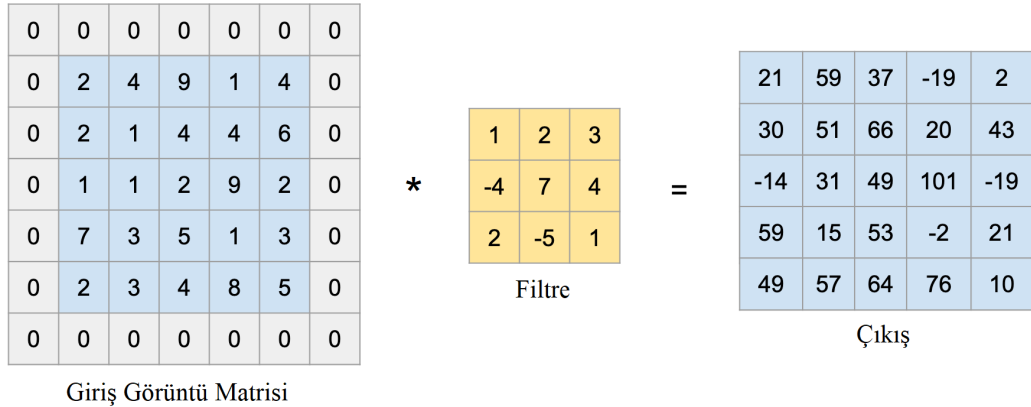
Yukarıda Şekil 2.21’de 3D evrişim örneği gösterilmiştir. Burada  $32 \times 32 \times 3$  boyutlarında bir giriş görüntüsü,  $5 \times 5 \times 3$  boyutlarında bir filtre kullanılmaktadır. Dikkat edilmesi gereken nokta giriş görüntüsünün derinliği ile filtre derinliğinin aynı olması gerektiğidir. Örnekte de derinliklerin eşit ve 3 olduğu görülmektedir. Filtre belirli bir konumdayken, girişin küçük bir hacmini kapsamakta ve yukarıda 2 boyutta açıklanan evrişim işlemi burada 3 boyutta gerçekleştirilmektedir. Filtre giriş görüntüsünün üzerinde kaydırılarak 2 boyuttaki evrişim işlemine benzer şekilde filtre ile giriş görüntüsündeki karşılık gelen değerler eleman bazında çarpılık toplanmakta ve özellik haritası elde edilmektedir. Evrişim işlemi sonucunda elde edilen özellik haritası  $32 \times 32 \times 1$  boyutlarına sahiptir. Şekil 2.21’nin sağ tarafında kırmızı dilimde bu durum gösterilmiştir. Burada 10 farklı filtre kullanılırsa,  $32 \times 32 \times 1$  boyutlarında 10 özellik haritası elde edilecektir. Elde edilen özellik haritaları arka arkaya eklenerek evrişim katmanının çıktısı şekilde de görüldüğü gibi  $32 \times 32 \times 10$  boyutunda elde edilmektedir. evrişimli katmanda kullanılan 2 önemli özellik bulunmaktadır. Bunlar kaydırma (stride) ve dolgulama (padding)’dır.

**Kaydırma (Stride):** Kaydırma, filtrenin giriş örneği boyunca nasıl hareket ettirileceğini gösteren bir değerdir. Örneğin kaydırma yani S değeri 2 olarak seçilirse, bu her evrişim işleminde filtrenin 2 piksel kaydırıldığı anlamına gelmektedir. Kaydırma işlemi sonucunda çıktı boyutlarının Şekil 2.22’de gösterildiği gibi girdi boyutlarına göre küçülmesine yol açmaktadır. Örnekte  $5 \times 5$ ’lik bir giriş matrisinden  $S = 2$  seçildiğinde  $2 \times 2$ ’lik bir çıktı elde edilmektedir.



**Şekil 2.22.** Kaydırma (Stride) örneği: (a) S = 1 (b) S = 2

**Dolgulama (Padding):** Yukarıda kayma anlatılırken evrişim işlemi sonucunda kaydırma değerine göre çıkışta boyutların küçüldüğü anlatılmıştır. Bu durumun önüne geçmek yani elde edilen çıktının boyutlarının küçülmesini engellemek için dolgulama işlemi uygulanmaktadır. Dolgulama işlemi giriş ile çıktının aynı boyutlara sahip olması için giriş matrisinin etrafının sıfır değerleriyle doldurulması şeklinde tanımlanabilir. Şekil 2.23'te  $5 \times 5$  bir giriş görüntüsünün etrafı sıfır değerleri ile doldurularak  $7 \times 7$ 'lik bir matris elde edilmiştir. Evrişim işlemi sonucunda giriş görüntüsüyle aynı yani  $5 \times 5$ 'lik bir çıktı elde edildiği görülmektedir. Padding işlemi uygulanmaması durumunda ise çıkış boyutları küçülmektedir.



**Şekil 2.23.** Dolgulama (Padding) örneği

Dolgulama işlemi ve kaydırma işlemleri sonucu  $n \times n$ 'lik bir giriş matrisinden elde edilen çıkış matris boyutları şu şekilde hesaplanmaktadır:

$$n_{out} = \frac{n_{in} + 2p - k}{s} + 1 \quad (2.3)$$

$n_{out}$  : Çıkış matris boyutu

$n_{in}$ : Giriş matris boyutu

$k$  : Evrişim filtre boyut

$p$  : Dolgulama miktarı

$s$  : Evrişim kaydırma miktarı

## 2.7. Örnekleme Katmanı (Pooling Layer)

ESA'da kullanılan katmanlardan biri de örnekleme katmanıdır. Bu katmanın kullanılmasının sebebi evrişim işleminden sonra çıktı boyutlarını azaltmak ve hesaplamaları hızlandırmaktır. Örnekleme işlemi sonrasında hem eğitim süresi hem de aşırı uyum (overfitting) ile mücadele eden parametre sayısı azaltılmaktadır. Örnekleme katmanı özellik haritalarını bağımsız bir şekilde alt örnekleme (downsampling) yapılarak yükseklik ve genişliği azaltmaktadır. Ancak derinlik sabit kalmaktadır. Evrişim işleminin aksin örnekleme katmanı herhangi bir parametreye sahip değildir. 2 temel örnekleme yöntemi bulunmaktadır: maksimum ve ortalama örnekleme.

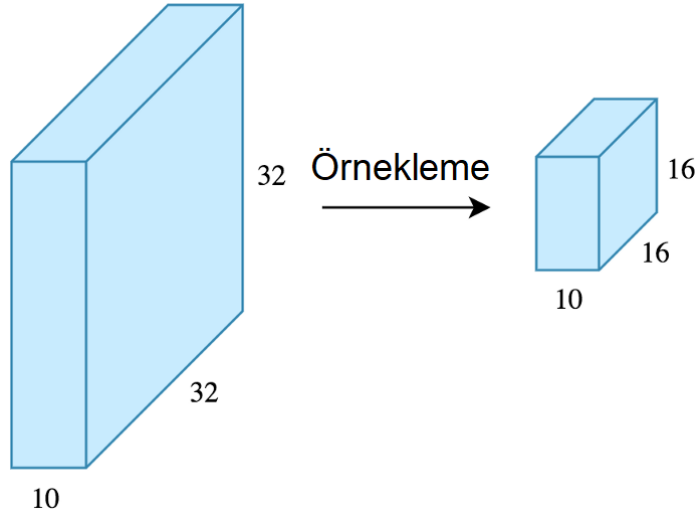
**Maksimum Örnekleme (Max Pooling):** Maksimum örnekleme, her özellik haritasının  $n \times n$ 'lik bir pencereye karşılık gelen her bir bölgesinde en büyük değeri hesaplayan bir örnekleme işlemidir. Maksimum örnekleme işlemi ile giriş örneği aşağı örneklenmekte ve böylece parametre sayısı azaltılarak maliyet düşürülmektedir. Maksimum örnekleme ile özellik haritasındaki baskın özellikler kullanılmaktadır. Daha az öneme sahip özelliklerin etkisi azaltılmaktadır.

**Ortalama Örnekleme (Average Pooling):** Ortalama örnekleme işleminde ise özellik haritalarının  $n \times n$ 'lik bir pencereye karşılık gelen her bir bölgesindeki değerlerin ortalaması hesaplanmaktadır. Özellik haritasındaki tüm değerler ortalama örnekleme işlemi ile hesaba katılarak bir sonraki katmana aktarılmaktadır.

Şekil 2.24'te maksimum ve ortalama örnekleme işlemleri gösterilmiştir. Solda maksimum örnekleme işlemine bakıldığında  $2 \times 2$ 'lik bir pencere, kaydırma (stride) değeri 2 seçilerek pencerenin özellik haritasında eşleştiği bölgenin en büyük değerinin alındığı görülmektedir. Sağ taraftaki ortalama örnekleme işleminde ise  $2 \times 2$ 'lik bir pencere ve kaydırma değeri 2 seçilerek, eşleşen bölgedeki tüm değerlerin ortalamalarının hesaplandığı görülmektedir. Burada dikkat edilmesi gereken bir diğer nokta ise boyutların yarıya inmesidir. Bu durum örnekleme işleminin temel kullanım amacıdır böylece özellik haritasındaki önemli özellikler boyut azaltılmasına rağmen korunmaktadır.



Şekil 2.24. Maksimum ve ortalama örnekleme



Şekil 2.25. 3D örnekleme

Şekil 2.25'de ise 3 boyutlu örnekleme işlemi gösterilmiştir. Burada  $2 \times 2$ 'lik bir filtre kullanılmıştır ve kaydırma değeri 2 alınmıştır.  $32 \times 32 \times 10$  boyutlarındaki giriş, örnekleme

işlemi sonrasında  $16 \times 16 \times 10$  boyutuna indirgenmektedir. Burada genişlik ve yükseklik değerleri kullanılan filtre ve kaydırma değerlerinden dolayı yarıya inerken derinlik aynı kalmaktadır çünkü örnekleme işlemi derinlikten bağımsızdır. Genişlik ve yükseklik yarıya indirilirken ağırlık sayısı girişin  $\frac{1}{4}$ 'üne inmektedir. ESA mimarilerinde genel olarak  $2 \times 2$  filtre ve kaydırma değeri olarak ta 2 değeri kullanılmaktadır.

## 2.8. Tam Bağlantılı Katman (Fully Connected Layer)

Evrışim ve örnekleme katmanlarından sonra, ESA mimarisini tamamlamak için tam bağımlı katman eklenmektedir. Tam bağımlı katman yapay sinir ağı mimarisi ile aynıdır. Yapay sinir ağlarından önceki bölümlerde detaylıca bahsedilmiştir o yüzden burada tekrar bahsedilmeyecektir. Evrişim ve örnekleme katmanlarının çıktıları 3 boyutludur ancak tam bağımlı katmanın girdileri tek boyutlu vektörler şeklindedir. Bu nedenle evrişim ve örnekleme katmanlarının çıktılarını düzleştirme (flattening) işlemi uygulanmaktadır. Düzleştirme 3 boyutlu hacmi 1 boyutlu vektöre çevirme işlemidir. Tam bağımlı bir katman eklemek, evrişimli katmanın çıktısı ile temsil edilen yüksek seviyeli özelliklerin doğrusal olmayan kombinasyonlarını öğrenmenin temel bir yoludur.

## 2.9. Yığın Normalleştirme (Batch Normalization)

Yığın normalleştirme katmanı, aktivasyon katmanından sonra uygulanır ve belirli bir giriş hacminin aktivasyon değerlerini ağıdaki bir sonraki katmana aktarmadan önce normalleştirmek için kullanılır. Girdileri normalleştirmek için kullanılan mini yığının ortalamasını ve varyansını hesaplamak için formüller şu şekildedir.

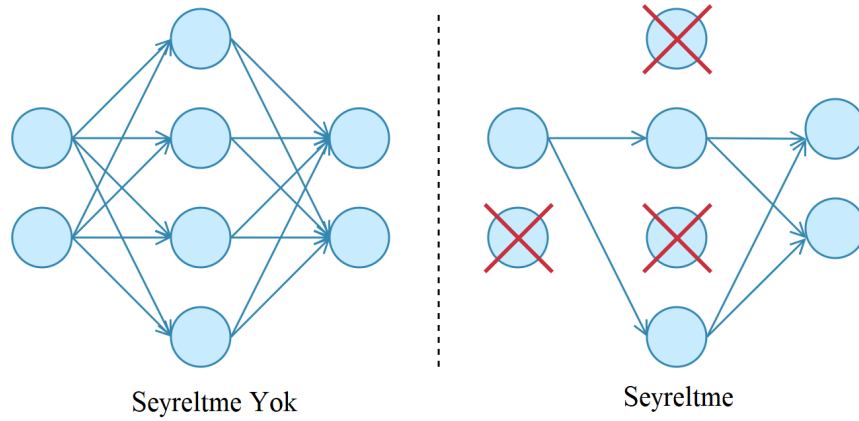
$$\mu_{\beta} = \frac{1}{M} \sum_{i=1}^m x_i \quad \sigma_{\beta}^2 = \frac{1}{m} \sum_{i=1}^m (x_i - \mu_{\beta})^2 \quad (2.4)$$

Yığın normalleştirme katmanı, ağı eğitmek için gereken döngü (epoch) sayısını azaltmada son derece etkilidir, yani daha hızlı ve daha istikrarlı model eğitime katkı sağlamaktadır.

## 2.10. Seyreltme Katmanı (Dropout)

Seyreltme, ESA veya yapay sinir ağlarında rastgele seçilen nöron kümesinin eğitim sırasında eksiltilmesi işlemidir. Seyreltme katmanı, modeli eğitirken başarımını artırarak

aşırı uyumu (overfitting) önlemeyi amaçlayan bir düzenleme (regularization) yöntemidir. Seyreltme, derin sinir ağları için en popüler düzenleme tekniğidir. % 95 doğruluğa sahip son teknoloji modeller bile, sadece seyreltme ekleyerek % 2'lik bir doğruluk artışı elde etmektedir, bu da bu seviyede oldukça önemli bir kazançtır. Seyreltme, overfitting problemini önlemek için kullanılan çok basit ancak etkili bir tekniktir. Model eğitilirken her bir adımda nöronlar geçici olarak belirli bir  $p$  olasılığında devre dışı bırakılır. Devre dışı bırakılan nöronlar her eğitim adımında  $p$  olasılığında örneklenir, böylece bir adımda bırakılan nöron bir sonraki adımda aktif hale gelmektedir. Burada  $p$  bir hiper parametredir ve bırakma oranı olarak adlandırılmaktadır. Örneğin  $p$  değeri 0,5 seçildiğinde her adımda nöronların %50' si devre dışı bırakılmaktadır. Seyreltme tekniğinin başarılı sonuç vermesi şaşırtıcıdır çünkü nöronlar bilerek devre dışı bırakılmaktadır ancak ağ daha iyi performans göstermektedir. Bunun nedeni, seyreltme tekniği ile ağın az sayıda nörona aşırı bağımlı olmasını engellenmektedir ve her nöron bağımsız olarak çalışmaya zorlanmaktadır.



**Şekil 2.26.** Seyreltme (Dropout) örneği

Şekil 2.26' da sol tarafta normal bir sinir ağı örneği gösterilmiştir. Sağ tarafta ise seyreltme uygulanan sinir ağı örneği bulunmaktadır. Görüldüğü gibi seyreltme işlemi sonrası nöronlar ve bağlantılar belirli bir oranda devre dışı bırakılmıştır. Seyreltme işlemi sadece giriş veya gizli katmanlara uygulanabilmektedir. Çıkış katmanında bu yöntem kullanılmamaktadır. Aynı zamanda modelin test aşamasında da seyreltme tekniği uygulanmamalıdır.

## 2.11. Veri Sınıflandırma

Makine öğrenimi ve istatistikte sınıflandırma, bir bilgisayar programının mevcut bir takım veriden önemli özellikleri öğrendiği ve daha sonra öğrenilen özellikleri yeni örnekleri sınıflandırmak için kullandığı bir tür denetimli öğrenme tekniğidir. Öğrenme işleminin yapılacağı ve sınıflandırmada kullanılacak veri basitçe iki sınıftan oluşabildiği gibi çok sınıflı da olabilmektedir. Örneğin bir kişinin kadın mı erkek mi olduğu veya bir e-postanın istenmeyen bir posta olup olmadığı ikili sınıflandırma (binary classification) problemidir. 2'den fazla çiçek türünün türüne göre sınıflandırılması da çok sınıflı sınıflandırmaya (multiclass classification) örnek verilebilir. Sınıflandırma problemlerinin bazı pratik örnekleri şunlardır: konuşma tanıma, el yazısı tanıma, biyometrik tanımlama, belge sınıflandırması vb. biyometrik bir tanıma olan yüz tanıma da kişi tanıma ile ilgili bir problemidir ve burada kişilerin birtakım özelliklere göre sınıflandırılması gerekmektedir. Bu nedenle sınıflandırma teknikleri yüz tanımda da tercih edilmektedir. En sık kullanılan sınıflandırma yöntemleri şunlardır:

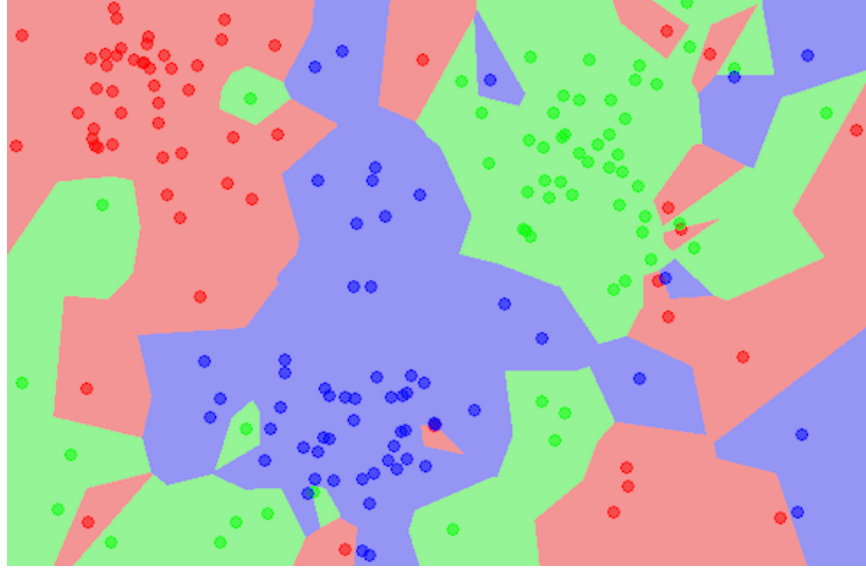
1. Lineer Sınıflandırıcılar (Linear Classifiers) : Logistic Regression, Naive Bayes Classifier
2. K-En Yakın Komşuluk (K-Nearest Neighbor, K-NN)
3. Destek Vektör Makineleri (Support Vector Machine, SVM)
4. Karar Ağaçları (Decision Trees)
5. Rastgele Değişken (Random Forest)

Tez kapsamında da üzerinde çalışılan yüz tanıma sisteminde daha çok destek vektör makineleri ve K-en yakın komşuluk sınıflandırma yöntemleri kullanıldığı için bu iki yöntem hakkında detaylı bilgi verilecektir.

### 2.11.1. K-En Yakın Komşuluk Sınıflandırıcı

K-en yakın komşu algoritması, benzer şeylerin birbirine yakın olduğunu varsayan denetimli bir sınıflandırma tekniğidir. Algoritma, bir grup etiketli örneği alır ve bu örneklerden

önemli özellikleri öğrenerek yeni örnekleri etiketlemeyi öğrenir. Yeni bir örneği etiketlemek için gelen yeni örneğe en yakın etiketli örneklere bakar (bunlar en yakın komşulardır). Yakınlık, tipik olarak bir benzeşmezlik fonksiyonu olarak ifade edilebilir. En yakın komşusu 'K' sayısını kontrol ettikten sonra, komşuların çoğunun sahip olduğu etikete göre bir etiket atar.

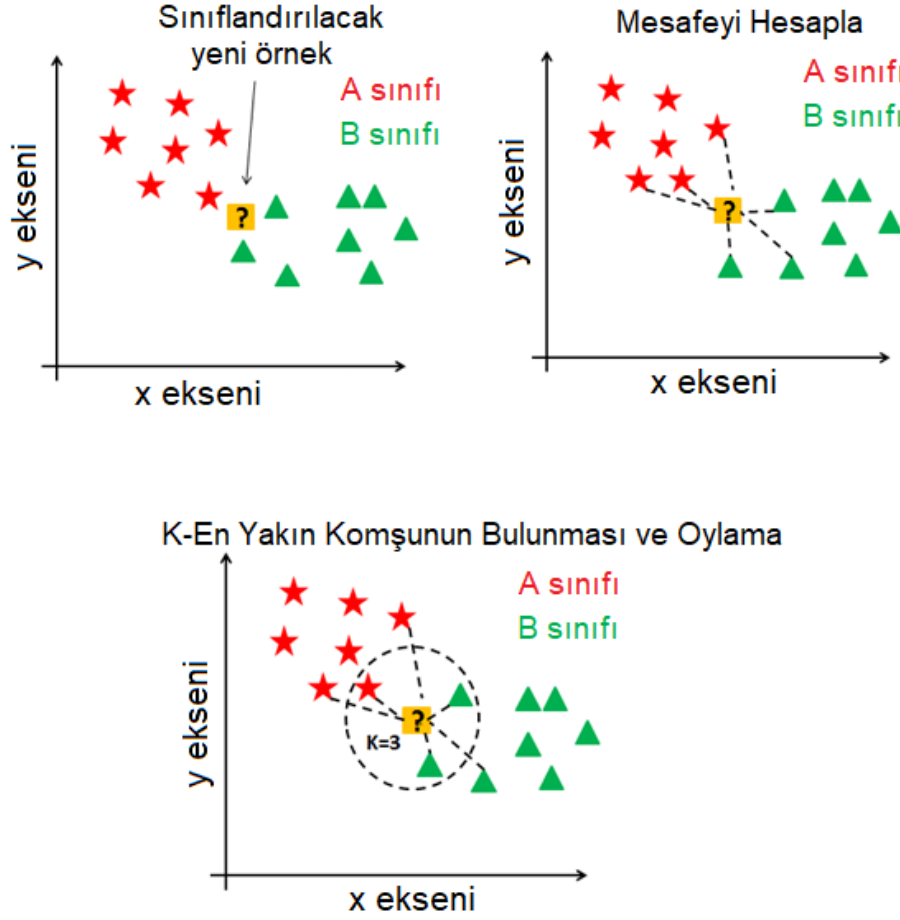


**Şekil 2.27.** K-NN sınıflandırıcısı ile sınıflandırılmış veri

Yukarıda Şekil 2.27'de, genel olarak benzer veri noktalarının birbirine yakın olduğu görülmektedir. K-NN algoritması, bir takım uzaklık hesaplama yöntemlerini kullanarak örnekler arasındaki benzerliği yakalama mantığı ile çalışmaktadır. Öklid uzaklık (Euclidean Distance) tekniği sıklıkla kullanılan uzaklık hesaplama yöntemlerinden biridir.

K-NN sınıflandırıcılar, parametrik olmayan algoritmaların gerekli olduğu gerçek hayat senaryolarında kullanılmaktadır. Bu algoritmalar, verilerin nasıl dağıtıldığı konusunda herhangi bir varsayımda bulunmamaktadır. Burada dikkat edilmesi gereken bir diğer nokta k parametresidir. K değeri sınıflandırılacak veri için gelen örneğin en yakın kaç tanesinin alınacağını temsil etmektedir. K değeri 3 alınırsa, sınıflandırılacak veri için önceden sınıflandırılmış verilerle uzaklığı hesaplanarak yeni gelen veri en yakın 3 tane örneğin olduğu sınıfa atanmaktadır.





Şekil 2.28. K-NN sınıflandırıcı ile veri sınıflandırma

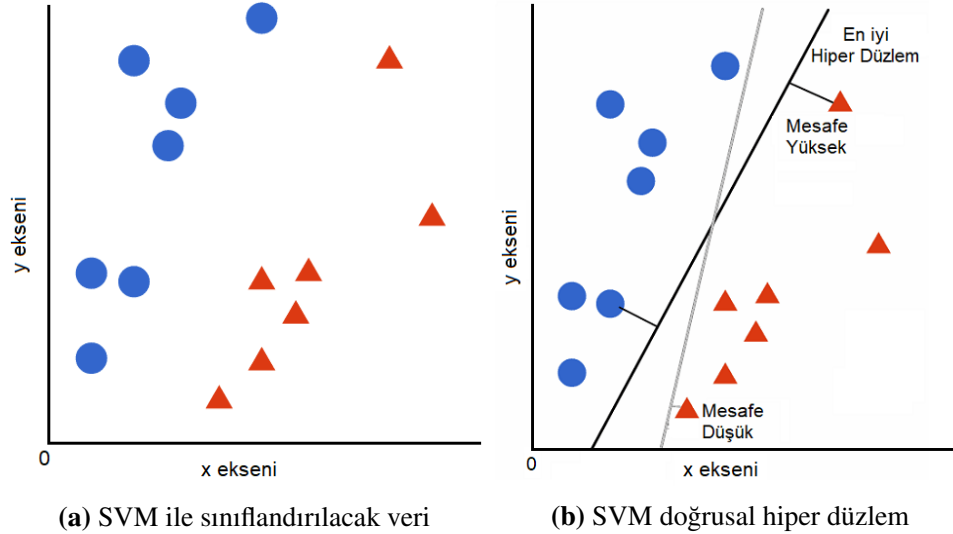
Yukarıdaki Şekil 2.28'de K-NN sınıflandırıcı örneği gösterilmiştir. Algoritmanın çalışmasında bir K parametresi belirlenir. Bu K parametresi kontrol edilecek eleman sayısını temsil etmektedir. Yeni bir örnek geldiğinde bu örneğe en yakın K eleman alınarak birbirlerine göre uzaklıkları hesaplanır. Uzaklık hesaplamak için genellikle öklid uzaklık fonksiyonu kullanılır. Öklid fonksiyonuna alternatif olarak Manhattan, Minkowski ve Hamming fonksiyonları da kullanılabilir. Uzaklığın hesaplanmasının ardından yeni örnek en kısa K uzaklığına göre bir sınıfa atanır.

Öklid uzaklık hesaplama denklemi şu şekildedir:

$$\text{Öklid Uzaklığı} = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (2.5)$$

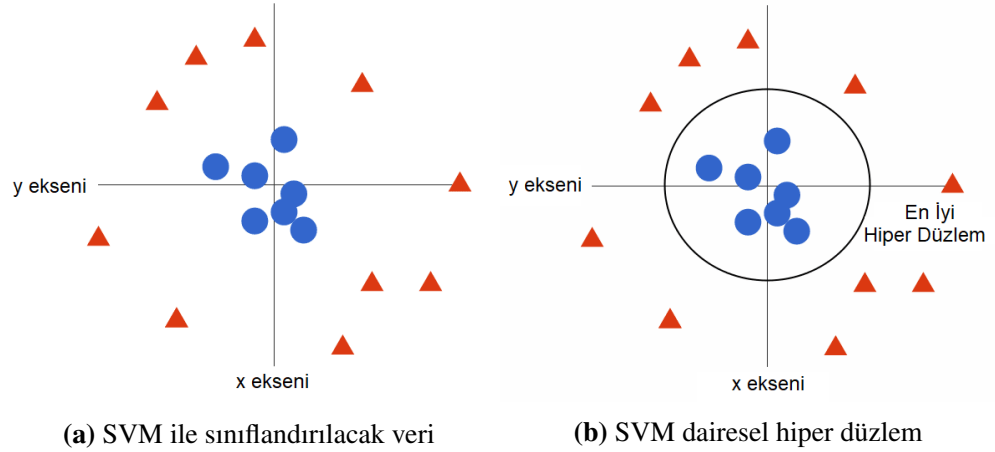
### 2.11.2. Destek Vektör Makineleri (Support Vector Machines)

Destek Vektör Makineleri, sınıflandırma ve regresyon analizi için verilerin analizini sağlayan bir tür denetimli makine öğrenme algoritmasıdır. Regresyon için kullanılabilmesine rağmen, SVM çoğunlukla sınıflandırma için kullanılır. Temel olarak iki sınıfa ait verileri birbirinden en uygun şekilde ayırmak için kullanılır. Bunun için karar sınırları yani hiper düzlemler belirlenmektedir. Destek vektör makinelerini bir örnek ile açıklamak gerekirse: Aşağıdaki Şekil 2.29a'da sol tarafta gösterildiği gibi kırmızı ve mavi iki etiket iki farklı sınıfın özelliklerini temsil etsin. SVM bu iki sınıfı sağ tarafta Şekil 2.29b'de gösterildiği gibi en iyi ayıran bir hiper düzlem atamaktadır. Çizginin bir tarafı kırmızı, diğer tarafı mavi sınıftaki özellikleri temsil edecektir.



Şekil 2.29. SVM doğrusal sınıflandırma örneği

Öğrenme kabiliyetini en üst düzeye çıkarmak için hiper düzlem iki sınıfa en uzak olacak şekilde belirlenecektir. Bununla birlikte, veri kümeleri . Şekil 2.30'da gösterildiği gibi daha karmaşık hale geldikçe, verileri iki grupta sınıflandırmak için tek bir çizgi çizmek mümkün olmayabilir.



**Şekil 2.30.** SVM doğrusal olmayan sınıflandırma örneği

SVM kullanıldığında veri ne kadar karmaşık olursa tahmin sonuçları o kadar başarılı olmaktadır. Şekil 2.30a'daki örnekte eğer z eksenini de eklenerek 3 boyutlu olarak düşünülürse, veriyi en iyi sınıflandıran hiper düzlem görselin 2.30b'deki gibi daire halini almaktadır. SVM çok boyutlu olduğu için veri sınıflandırmada başarılı sonuçların elde edilmesine olanak sağlamaktadır.

## 2.12. Yüz Tespiti

Yüz tespiti yüz hizalama, yüz modelleme, yüz tanıma, kimlik doğrulama, yüz ifadesi izleyip tanıma, cinsiyet tespiti, yaş tahmini gibi birçok işlemi içeren yüz analizi algoritmalarının temelidir. Yüz tanıma sisteminde de ilk yapılan iş yüz tespittir yani bir görsel ya da video da insan yüzü var mı yok mu tespit etmektir. Sisteme herhangi bir görüntü verildiğinde, yüz tespitinin amacı görüntüde herhangi bir yüz olup olmadığını belirlemek ve görüntünün mevcut olup olmadığını görebilmek için görüntünün konumunu ve her yüzün boyutunu döndürmektir. Yüz tespiti için kullanılan teknikler 4 Farklı kategoride incelenebilir:

- Bilgi Tabanlı Yöntemler
- Özellik Tabanlı Yöntemler
- Şablon Eşleştirme Tabanlı Yöntemler
- Görünüm Tabanlı Yöntemler

### **2.12.1. Bilgi Tabanlı Yöntemler**

Bilgiye dayalı yöntem, kurallar kümesine dayanır ve yüzleri tespit etmek için insan bilgisine dayanır. Örneğin bir yüz, belirli mesafeler ve konumlar arasında birbirinin içinde bir burun, gözler ve ağza sahip olmalıdır. Bu yöntemlerle ilgili en büyük sorun, uygun bir kural kümesi oluşturmanın zorluğudur. Kurallar çok genel ya da çok ayrıntılı olursa birçok yanlış pozitif olabilir. Tek başına bu yaklaşım yetersiz ve bir görüntüdeki çok sayıda yüzün tespitinde yetersiz kalmaktadır.

### **2.12.2. Özellik Tabanlı Yöntemler**

Bu yöntem gözler, burun, ağız vs. gibi yüzün yapısal özellikleri çıkarır ve daha sonra bir yüzü tespit etmek için bunları kullanır. Bu algoritmaların sorunu, bu özelliklerin aydınlatma, girişim ve gürültüye bağlı olarak bozulmuş olmasıdır. Yüz bölgelere ayrılır ve bölgenin şekli daha sonra yükseklik, en boy oranı ile parametrelendir, renk ve şekle dayalı bir yüz olarak sınıflandırılır.

### **2.12.3. Şablon Eşleştirme Tabanlı Yöntemler**

Şablon eşleme yöntemi görüntülerdeki yüzleri tespit etmek için kullanılan yöntemlerden biridir. Yüzleri tespit etmek için birtakım yüz şablonları kullanılmaktadır. Şablon eşleme tekniği, bir görüntü üzerinde hazır bir şablon ile eşleşen bölümlerin tespit edilmesi temeline dayanmaktadır. Şablon görüntüsü görüntü üzerinde dolaştırılarak her piksel için şablon ile görüntü üzerindeki eşleşmeler tespit edilerek görüntü ile şablon arasındaki benzerlik ölçülür. Eşleşen piksel koordinatları kaydedilir. Şablonlar yüz tespiti için temel olarak kişilerin ön açıdan çekilmiş fotoğraflarından yüzün genel kısmı, burun ve göz kısımları gibi önemli ayırt edici bölgelerden belirlenmektedir.

### **2.12.4. Görünüm Tabanlı Yöntemler**

Görünüme dayalı yöntemde, şablon eşleştirmenin aksine, modeller veya şablonlar yüz görünümünün temsili değişkenliğini yakalaması gereken bir dizi eğitim görüntüsünden öğrenilir. Görünüme dayalı yaklaşım, diğer yöntemlerden daha iyidir. Genel olarak görünüm temelli yöntem, yüz görüntülerinin ilgili özelliklerini bulmak için istatistiksel

analiz ve makine öğrenmesinden elde edilen tekniklere dayanır. Bu yöntemler, yüz tanıma için özellik çıkarımında da kullanılır. Yüz tespiti için kullanılan birbirinden farklı algoritmalar mevcuttur. En yaygın olarak kullanılan algoritmalar ise şu şekildedir:

- Viola Jones (Haar-Cascade) Yüz Tespiti
- Yerel İkili Örüntüler (Local Binary Pattern, LBP) Yüz Tespiti
- Adaboost Yüz Tespiti
- Gradyen Histogramları (Histogram of Gradients, HOG) ile Yüz Tespiti
- Yapay Sinir Ağı - Derin Öğrenme Tabanlı Yüz Tespiti

**Viola Jones (Haar-Cascade) Yüz Tespiti:** Viola-Jones algoritması, 2001 yılında Paul Viola ve Michael Jones tarafından önerilen gerçek zamanlı olarak nesne tespiti konusunda başarılı ilk nesne tespit çerçevesidir. Çeşitli nesne sınıflarını tespit etmek için eğitilebilse de, esas olarak yüz tanıma sorunu için geliştirilmiştir. Bu yüz algılama çerçevesi, yüksek algılama oranları elde ederken görüntüleri son derece hızlı bir şekilde işleyebilir. Üç temel aşaması vardır:

- Bunlardan ilki, dedektör tarafından kullanılan özelliklerin çok hızlı bir şekilde hesaplanmasına izin veren "İntegral Görüntüsü" adı verilen yeni bir resim gösteriminin tanıtılmasıdır. Yani ilk aşamada resimlerin integralleri alınır. Böylelikle piksel değerlerinin tek tek toplamlarının hesaplanması yerine integralle hesaplanır. Böylelikle bilgisayardan büyük bir işlem gücü kaldırılır.
- İkincisi, çok büyük bir özellik kümesinden az sayıda kritik görsel özellik seçmek için AdaBoost öğrenme algoritması kullanılarak oluşturulan kolay ve etkili bir sınıflandırıcıdır.
- Üçüncü ise yüz benzeri bölgeler üzerinde daha fazla hesaplama yaparken resmin arkaplanını çıkaran sınıflandırıcıların kaskat bir şekilde birleştirilmesidir. (Gupta ve Sharma 2014)

**LBP Yüz Tespiti:** LBP algoritması merkez ve komşu pikseller arasında karşılaştırma yaparak ikili kodlar üreten veriyi ayırt etme kabiliyeti yüksek bir analiz tekniğidir. Her bir

resim pikseli, kendisini çevreleyen  $3 \times 3$  lük komşu bölgeleriyle arasındaki farkın  $u(x)$  basamak fonksiyonu kullanılarak ikilileştirilmesiyle etiketlenir. Bir örüntüye ait LBP histogramı aşağıdaki eşitlikle elde edilir:

$$LBP_{P,R}(x_c, y_c) = \sum_{p=0}^{P-1} s(g_p - g_c) 2^p \quad (2.6)$$

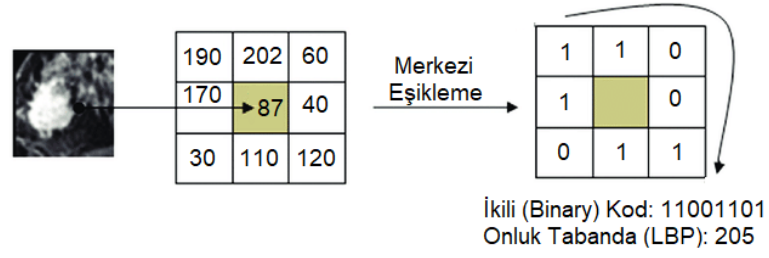
$$s(t) = \begin{cases} t < 0 \text{ için } 0 \\ t \geq 0 \text{ için } 1 \end{cases} \quad (2.7)$$

Eşitlik 2.6'da  $x_c, y_c$  merkez pikselin LBP algoritmasına göre değerini,  $g_p$  merkez pikselin komşularını,  $g_c$  merkez pikselin gri değerini,  $R$  komşuların merkez piksele olan uzaklığını,  $P$  ise işlenen komşu sayısını ifade etmektedir. LBP operatörü ile farklı boyutlardaki görüntü örneklerini analiz etmek mümkün hale gelmiştir. Her bölge için temel LBP değerleri hesaplanmaktadır. Komşu bölgelerin LBP değerleri ard arda eklenerek LBP histogramları çıkarılmakta ve bu histogramlar görüntülerin birbirinden ayrılmasını sağlayan benzersiz tanımlayıcılar olarak kullanılabilir.

Çalışma prensibi şu şekildedir :

$3 \times 3$ 'lük bir pencere kullanılarak görsel üzerinde gezdirilerek merkez piksele göre LBP kodu hesaplanır. Diğer piksel değerleri merkez pikselden büyükse 1, küçükse 0 olarak değerlendirilir. Böylelikle eşikleme işlemi gerçekleştirilir. Eşikleme işleminin ardından sol üst köşedeki ikili sayı ilk basamak kabul edilerek saat yönünde sırasıyla 8 bitlik LBP kodu oluşturulur. Bu kodun onluk sistemdeki değeri hesaplanarak merkez pikselin LBP değeri bulunur. Bu işlem tüm imge boyunca tekrarlanır. Eğitilmiş veri ile olan benzerliğe göre yüz tespiti yapılır.

**Adaboost Yüz Tespiti:** Güçlendirme (Boosting), birçok nispeten zayıf ve yanlış kuralı birleştirerek son derece doğru bir tahmin kuralı oluşturma fikrine dayanarak makine öğrenmesine yönelik bir yaklaşımdır. AdaBoost algoritması, ilk pratik geliştirme algoritmasıdır



**Şekil 2.31.** LBP eşikleme örneği

ve farklı alanlarda en çok kullanılan, çalışılan algoritmalarından biridir. Resimleri hızlı bir şekilde sınıflandırırken yüksek tespit oranlarına sahip bir algoritmadır. Adaboost basit sınıflandırıcılardaki görsel özellikleri seçerek güçlü sınıflandırıcılar üreten bir öğrenme algoritmasıdır. Zayıf sınıflandırıcılardan daha yüksek doğruluk oranlarına sahiptir.

**Gradyen Histogramları ile Yüz Tespiti (Histogram of Gradients, HOG):** İlk defa Shashua ve Dalal tarafından önerilen bu yöntemde imgeyi bir grup lokal histogramlar olarak tanımlamak hedeflenmektedir. Lokal histogramlar görüntülerdeki bölgesel gradyen yönelimlerinin sayılarından oluşmaktadır. HOG yöntemindeki temel düşünce, bir objenin şekil ve görünümünün yön vektörleri ile gösterilebilmesidir. Bu düşüncüyü gerçekleştirmek için görüntüler  $9 \times 9$  veya  $16 \times 16$  piksellik alt bölümlere ayrılmakta ve her bir bölüm için yön vektörleri hesaplanmaktadır. Böylece farklı hücrelerdeki vektörler birleştirilerek görüntüdeki nesnenin önemli özelliklerini ifade eden tanımlayıcılar elde edilmektedir. Görüntülerden birtakım özellikler çıkarılırken ışık, gölge veya arkaplan gibi faktörler sonuçları doğrudan etkilemektedir. Bu etkileri azaltmak için HOG tanımlayıcıları çıkarılırken görüntüler normalize edilmektedir. Normalizasyon sonucu elde edilen HOG histogramlarının aydınlatma, gölgeleme vb. değişimlerden etkilenmesi en aza indirilmektedir. HOG yöntemi ile elde edilen özellik tanımlayıcılar yerel bölgeler üzerinden hesaplandığı için nesnenin geometrik ve fotometrik değişimlere karşı diğer özellik çıkarma yöntemlerine göre daha başarılı olduğu görülmektedir. HOG yöntemi yüzdeki özellikleri çıkarmada başarılı sonuçlar ürettiği için yüz tespitinde de başarılı sonuçlar vermektedir. Ancak bu yöntem kameraya doğrudan bakan yüzler için başarılı olmasına karşın yüz açısının değişmesi durumunda performans olarak zayıf kalmaktadır.

**Derin Öğrenme - Yapay Sinir Ağı Tabanlı Yüz Tespiti:** Yapay sinir ağları, karakter tanıma, nesne tanıma ve otonom robot sürüşü gibi sayısız örüntü tanıma problemine uygulanmaktadır. Yüz tespiti için yıllar boyunca çeşitli ağlar önerilmiştir. Yapay sinir ağlarının kullanmanın avantajı, yüzlerin karmaşıklığını yakalayabilmesidir. Bir sinir ağı görüntünün küçük pencerelerini inceler ve her pencerenin bir yüz içerip içermediğini tespit eder. Sistem, tek bir ağ üzerinden performansı iyileştirmek için birkaç ağ arasında karar verir. Derin öğrenme tabanlı yüz tespit yöntemleri yeterli veri ile yüksek doğruluk oranlarına sahiptir. Farklı açılardan yüzleri başarılı bir şekilde tespit edebilmektedir. Ancak karmaşık ağ yapısına sahip olmaları, fazla veriye ihtiyaç duyulması gibi dezavantajları vardır. Günümüzde sosyal medya platformlarının veya web sitelerinin fazlalığı veri problemini ortadan kaldırmaktadır. Bilgisayar donanımındaki gelişmeler ile birlikte modeller ne kadar karmaşık olursa olsun başarılı ve hızlı sonuç verebilmektedirler.

### **2.13. 3D Derinlik Kameraları**

3D kamera, görüntülerdeki derinlik algısının, insan görüşüyle deneyimlendiği gibi üç boyutu kopyalamasına izin veren bir görüntüleme cihazıdır. Bazı 3D kameralar, birden çok bakış açısını kaydetmek için iki veya daha fazla mercekle kullanırken, bazıları ise konumunu değiştiren tek bir mercek kullanır. İki bakış açısının birleşimi, iki insan gözünün biraz farklı bakış açısıyla olduğu gibi, derinlik algısını mümkün kılmaktadır. 3D efektin arkasındaki ilkeye stereoskopi denir ve bu teknoloji stereoskopik görüntüleme olarak bilinir. Sol ve sağ gözle görülen nesnelere arasındaki fark (binoküler eşitsizlik olarak bilinir) iki perspektifi bütünleştirmek için odaklanma ve görsel merkez yorumlama yoluyla yerleştirmeye ek olarak insan görüşünde perspektif geliştirmeye yardımcı olur.

3D kameraları daha iyi anlayabilmek için öncelikle normal kameranın çalışma mantığını incelemekte fayda vardır. Normal (3D olmayan) bir kamera, görüntüleri ya hareketsiz bir fotoğraf olarak ya da video veya film olarak bilinen hareketli görüntüler olarak kaydeder ve bunlar daha sonra dijital bir sistem veya fotoğraf filmi gibi fiziksel bir ortamda depolanır. Bir kamera, sahneden gelen ışığı odaklayan bir mercek ve görüntü yakalama mekanizmasını tutan bir kamera gövdesi içerir. Kamera kelimesi, karanlık oda anlamına



gelen Latince camera obscura kelimelerinden gelir.

Tüm kameralar aynı temel tasarıma sahiptir: ışık, yakınsak/dışbükey bir mercek aracılığıyla kapalı bir kutuya girer ve bir görüntü, ışığa duyarlı bir ortama (genellikle bir geçiş metal halojenür) kaydedilir. Deklanşör mekanizması, ışığın kameraya girebileceği süreyi kontrol eder. Genellikle bir sıvı kristal ekran (LCD) olan bir ekran, kullanıcının kaydedilecek sahneyi ve ISO hızı, pozlama ve deklanşör hızı gibi ayarları görüntülemesini sağlar. Bir film kamerası veya bir video kamera, bir dizi statik görüntüyü, genellikle saniyede 24 kare hızla art arda kaydetmesi dışında, hareketsiz kameraya benzer bir şekilde çalışır. Görüntüler hızlı bir şekilde arka arkaya koştığında, bize hareket yanılsaması verir. Hareketsiz bir film kamerasında üç temel unsur vardır: optik element, kimyasal element ve mekanik element. Böyle bir kamerada, görüntüleme ve fotoğrafçılık için tek bir lens takımı vardır. Sıçrayan ışık, bir dizi mercek aracılığıyla kameraya ve bir aynaya geçer. Oradan ışık, pentaprizma adı verilen bir cam parçasına yansır. Pentaprizmaya giren ışık üzerine, göz merceğinden geçip göze girene kadar karmaşık bir şekilde zıplar. Kameradaki düğmeye basıldığında, ayna yoldan çekilir ve beşli prizmanın içine sıçramak yerine, nesneden gelen ışık doğrudan kameranın arkasına geçer. Orada fotoğraf filmine çarpar ve bir dizi ışığa duyarlı hücreyi etkilediği için kimyasal bir reaksiyon başlatır. Bu, aktive olan her hücrede küçük bir elektrik yükü açığa çıkarır. Bir kameranın flaşı, filmde veya elektronik olarak iyi görünmek için çok loş olan bir sahneyi aydınlatma girişimidir.

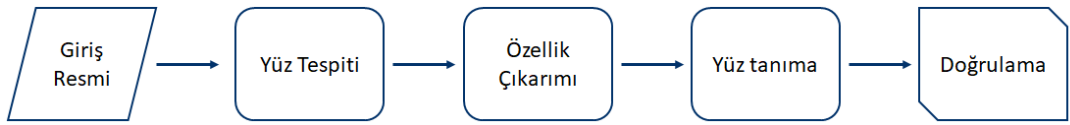
3D kameralar ayrıntılı bir şekilde incelenecek olursa, bunun için stereoskopinin iyi anlaşılması gerekmektedir. Yukarıda bahsedildiği gibi stereoskopi, görüntüleri yakalama ve bunları gerçekçi bir derinlikle üç boyutlu hale getirme tekniğidir. İnsanlar bir görüntüyü yakalamak için iki göze sahiptir. İki göz başın iki yanındadır, bu nedenle görüntü her bir göz retinasına yansıtıldığında aynı nesne iki farklı açıdan görülmektedir. Açı farkı, binoküler eşitsizlikler olarak bilinen sol ve sağ göz tarafından yakalanan iki görüntünün yatay konum farklılıklarına neden olur. Farklılıklar beyin tarafından işlendiğinde derinlik algılanmaktadır.

Temel olarak 3D kameraların çalışma mantığı 3D stereoskopik görüntülerde uygun derinliği oluşturmakla ilgilidir. Burada akılda tutulması gereken üç temel unsur vardır: ön plan, arka plan ve sıfır paralaks. Ön plan, görüntüde izleyiciye en yakın görünen öğedir, arka plan ise en uzaktaki görüntüdür. Sıfır paralaks, ekranı veya varsayılan bölgeyi ifade eder. Görüntüleri post prodüksiyonda işlerken, bir öğenin ne kadar çok farklılığı varsa, izleyiciye o kadar yakın görünecektir.

### 3. MATERYAL VE YÖNTEM

#### 3.1. Yüz Tanıma

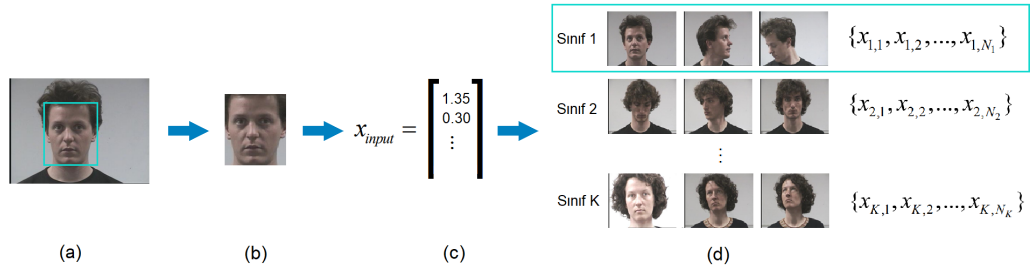
Bir yüz tanıma sistemi için dijital bir kameradan çekilmiş bir resim göz önüne alındığında, içinde yüz olup olmadığı, yüzün nerede olduğu ve yüzün kime ait olduğunu belirlemek temel hedeftir. Bu amaca yönelik olarak, yüz tanıma prosedürü genellikle üç adıma ayrılmaktadır: yüz algılama (Tespit), özellik çıkarma ve yüz tanıma. Şekil 3.1' de bir yüz tanıma sisteminin genel prosedürü verilmiştir. Bunlara ek olarak yüz tanıma yapıldıktan sonra kişi doğrulama adımı da yüz tanıma sistemlerinde bir sonraki adımı oluşturmaktadır.



**Şekil 3.1.** Yüz tanıma sistemi aşamaları

Bir önceki bölümde yüz tespiti ile ilgili detaylı açıklama yapılmıştır. Burada kısaca bahsedilirse yüz tespiti belirli bir görüntüde insan yüzlerinin görünüp görünmediğini ve bu yüzlerin nerede bulunduğunu belirlemektir. Bu adımın beklenen çıktıları, giriş görüntüsündeki her yüzü içeren kısımlardır. Yüz tanıma sistemlerini daha etkili ve tasarımı kolay hale getirmek için yüz tespiti yapıldıktan sonra belirlenen koordinatlara göre yüz bölgesi giriş görüntüsünden kırılmaktadır daha sonra kırılan yüz bölgesini ölçeklendirmek ve düzeltmek için hizalama işlemi uygulanmaktadır. Yüz tespiti ile sadece yüz bölgesi işleneceğinden hem daha az işlem yapılmakta çünkü girdi boyutları küçülmektedir hem de daha başarılı sonuçlar elde edilmektedir. Yüz tespitinden sonra görüntüden yüz bölgesi seçilir. Yüz tanıma için bu yüz bölgelerini doğrudan kullanmanın bazı dezavantajları vardır. İlk olarak yüz bölgeleri çok fazla pikselden oluşmaktadır bu durum etkili bir yüz tanıma sistemi oluşturmak için hesaplama maliyetini arttıracaktır. İkinci olarak ise yüz görüntüleri farklı kameralardan farklı açılarda, farklı ışık koşullarında olmakta, yüz görüntüleri üst üste olup birbirini kaplayabilmekte veya gürültü içerebilmektedir. Bu etkenler yüz tanıma sisteminin başarısını olumsuz etkilemektedir. Bu dezavantajları aşmak için ise yani bilgi toplama, boyut indirgeme, belirgin özellikleri çıkarma ve gürültü giderme için özellik

çıkarma adımı uygulanmaktadır. Özellik çıkarma adımı ESA ile gerçekleştirilebilmektedir. Özellik çıkarımından sonra bir yüz görüntüsü genellikle ESA vb. yapılarla sabit boyutlu bir vektöre dönüştürülür. Her yüzün temsilini vektör vb. bir yapı ile formüle ettikten sonra, son adım bu yüzlerin kimliklerini tanımdır. Otomatik tanıma elde etmek için, bir yüz veritabanı oluşturmak gerekir. Her kişi için birkaç görüntü alınır ve özellikleri çıkarılır ve veri tabanında saklanır. Daha sonra bir giriş yüzü görüntüsü geldiğinde, yüz algılama ve özellik çıkarma işlemi gerçekleştirir ve özelliğini veritabanında depolanan her yüz sınıfıyla karşılaştırılır. Bu sınıflandırma probleminin üstesinden gelmek için önerilen birçok algoritma veya yöntem vardır. K en yakın komşuluk sınıflandırıcı (K-NN), destek vektör makineleri (SVM) bu sınıflandırma algoritmalarına örnek olarak verilebilir. Yüz tanımanın iki genel uygulaması vardır, biri kimlik, diğeri ise doğrulama olarak adlandırılır. Yüz tanıma aşamasında, bir yüz görüntüsü verilerek sistemin o yüz görüntüsünün kim olduğu veya en olası kimliğini söylemesi beklenir; yüz doğrulama adımı ise, bir yüz görüntüsü ve bir yüz tanıma tahmin sonucu sistemin tahmin edilen yüz görüntüsünün doğru tahmin edilip edilmediğini söylemesi beklenmektedir. Yani kişi doğru tahmin edildiyse doğrulama adımı doğru/olumlu (true) sonuç döndürülürken yanlış tahminde ise yanlış/olumsuz (false) sonucu döndürülür.



**Şekil 3.2.** (a) Giriş görüntüsü ve yüz tespit sonucu (b) Elde edilen yüz bölgesi (c) Elde edilen yüz özellik vektörü (d) Sınıflandırma yöntemleri ile elde edilen öznelik vektörlerinin veritabanında saklanması ve giriş ile en eşleşen örneğin işaretlenmesi

Şekil 3.2’de giriş görüntüsü üzerinde yüz tespit, özellik çıkarımı ve sonrasında yüz tanıma işleminin nasıl yapıldığı gösterilmiştir.

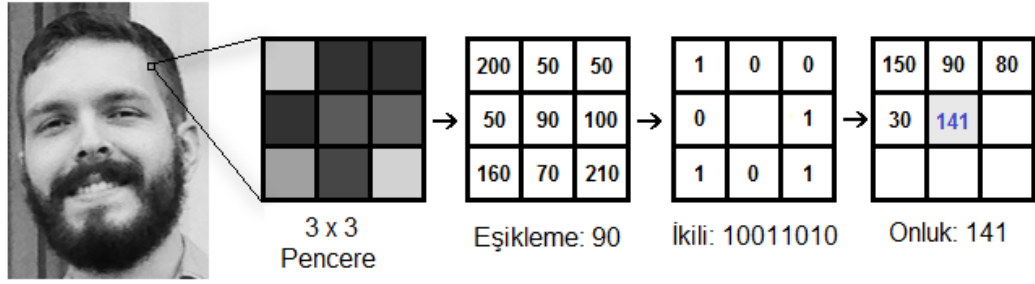
Yüz tanıma için günümüze kadar pek çok farklı yöntem uygulanmıştır. Uygulanan yöntemleri geleneksel yöntemler ve derin öğrenme tabanlı sistemler olarak ikiye ayırmak mümkündür. Klasik yüz tanıma yöntemleri görüntüleri bir takım makine öğrenmesi yöntemleri ile analiz ederek elde ettikleri özellikleri kullanarak yüz tanıma yapmaktadırlar. Bu yöntemlerin birtakım dezavantajları mevcuttur. Aydınlatmanın değişmesi, çözünürlük, mesafe ve kişinin yüz açısının değişmesi vb. durumlar bu yöntemlerin performansını olumsuz etkilemektedir. Ancak derin öğrenme tabanlı yüz tanıma sistemlerinde ise görüntü ESA ile işlenerek birtakım özellikler çıkarılıp sınıflandırma yöntemleri ile sınıflandırılmakta ve yüz tanıma işlemi gerçekleştirilmektedir. Böylece aydınlatma, yüz açısının değişmesi, çözünürlük vb. durumların etkisi ciddi oranda azaltılarak daha başarılı yüz tanıma sistemleri ortaya çıkarılabilmektedir.

### **3.1.1. Geleneksel Yüz Tanıma Algoritmaları**

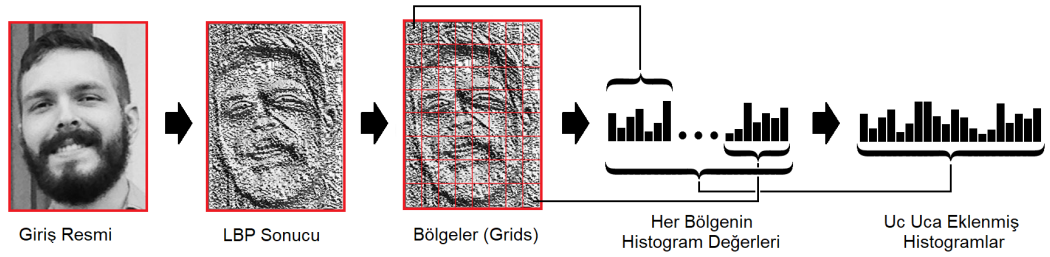
Son yıllarda yüz tanıma algoritmalarında derin öğrenme ve çok hızlı gelişimler yaşanmıştır. Geleneksel yüz tanıma algoritmaları bütünsel özellikler (holistic features) ve yerel özellik yaklaşımı (local feature approaches) kullanan algoritmalar olarak iki kategoride inceleyebilir. Bütünsel özellik yaklaşımını kullanan yöntemler da doğrusal ve doğrusal olmayan gösterim şeklinde ayrılabilir. Temel bileşen analizi (Turk ve Pentland 1991), bağımsız bileşen analizi (Bartlett ve ark. 2002), doğrusal ayırma analizi (Belhumeur ve ark. 1997), 2DPCA (Yang ve ark. 2004) ve doğrusal regresyon sınıflandırıcı (Naseem ve ark. 2010) gibi birçok uygulama, doğrusal projeksiyon görünümü tabanlı yöntemler için iyi sonuç veren uygulamalara örnek verilebilir. Ancak aydınlatma koşullarının değişmesi, yüz ifade ve açılarının değişiklik gösterebilmesi vb. faktörler bu yöntemleri olumsuz etkilemektedir ve bu yöntemler bu etkilerden dolayı yüzleri temsil etmekte başarısız olabilmektedirler. Bu durumun temel nedeni ise yüz modellerinin yüksek boyutlu uzayda karmaşık ve doğrusal olmayan özelliklere sahip olmasıdır. Bu tür durumlarla başa çıkabilmek için, çekirdek PCA, çekirdek LDA (Lu ve ark. 2003) veya yerel doğrusal gömme (He ve ark. 2005) gibi doğrusal olmayan yöntemler önerilmiştir. Çekirdek tekniklerini kullanan doğrusal olmayan bu yöntemler giriş yüz görüntülerini, çok sayıda yüzün doğrusal ve

basitleştirilmiş olduğu daha yüksek boyutlu bir uzaya haritalamaktadırlar. Doğrusal yöntemler, yüzü doğrusal bir alt uzay üzerine yansıtır. Yüz uzayından olan mesafe düzlem dik mesafedir, yüz uzayındaki mesafe ise ortalama görüntüden düzlem boyunca olan mesafedir. Her iki mesafe de mahalanobis mesafelerine dönüştürülebilir ve olasılıklı yorumlar verilebilir (Perlibakas 2004). Çekirdek tabanlı yöntemlerin güçlü teorik temeline rağmen, bu yöntemlerin yüz tanıma problemlerinde pratik uygulaması, doğrusal yöntemlere kıyasla önemli bir gelişme sağlamamaktadır. Yerel özellik yaklaşımını kullanan algoritmalar ise bütünsel özellikler tabanlı algoritmalara göre birtakım avantajlara sahiptir. Bu yöntemler yüz ifadesinin değişmesi gibi yerel değişikliklere, görüntülerin üst üste binmesi, aydınlatma koşullarının değişmesi gibi etkilere karşı daha karardır. Yerel özellik yaklaşımını kullanan algoritmalar içinde en çok bilinen yöntem yerel ikili örüntüler (Local Binary Patterns, LBP) yöntemidir. LBP merkez piksel etrafındaki komşu piksellerdeki değişiklikleri analiz ederek çalışan basit ama etkili bir yöntemdir. Gabor faz desenlerinin histogramı (Zhang ve ark. 2007) ve yerel Gabor ikili desen histogram sekansı (Yang ve Chen 2013) gibi birçok LBP varyantı orijinal LBP'yi iyileştirmek için önerilmiştir. LBP Modeldeki ayırt edici ve önemli bilgileri başarılı bir şekilde tespit etmeye izin verir. Özelliklerin çıkarılmasında etkili bir yöntem olduğu için yüz tanıma vb. pek çok uygulamada, sınıflandırma problemlerinde kullanılabilir.

**LBPH Yüz Tanıma:** Bu yöntem Ojala ve ark. tarafından önerilmiştir. Görüntüler üzerinde  $3 \times 3$ 'lük bir pencere gezdirilerek, pencerenin karşılık geldiği piksel değerleri için komşu pikseller merkez değeri ile kıyaslanmakta ve ikili formata dönüştürülmektedir. Üretilen ikili sayı dizisine LBP kodu denir ve bu kod ile görüntülerdeki nesnelere ait benzersiz özellikler belirlenebilmektedir. Aşağıda Şekil 3.3'de LBP kodunun nasıl elde edildiği görsel olarak da belirtilmiştir. LBP kodu elde etme işlemi tür resim taranarak her nokta için LBP kodları üretilir. Sonrasında bu kodlardan Şekil 3.4'te gösterilen histogram özellikleri elde edilir. Üretilen bu LBP histogramları yüz tanımda kullanılır.



Şekil 3.3. LBP kodunun elde edilmesi



Şekil 3.4. LBP histogramının elde edilmesi

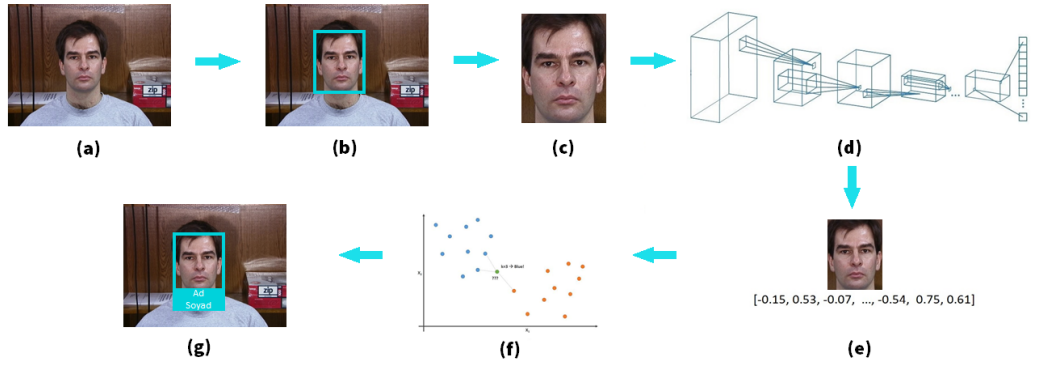
**Fisherfaces Yüz Tanıma:** 1997 yılında P. Belhumeur tarafından Linear Discriminant Analizi (LDA) temel alınarak geliştirilen bir yöntemdir. Bu yöntemde amaç LDA analizi ile bir kişiye ait fotoğraflardan elde edilen özellikleri birbirine yakınlacak şekilde küçültmek yani sınıf içi dağılımı minimuma indirmek, farklı kişilerin fotoğraflarından elde edilen özellikleri yani sınıflar arası dağılımı ise maksimum olacak şekilde ayarlamaktır. Böylece bir kişi için elde edilen vektörler kişinin farklı fotoğrafları için birbirine yakın çıkmakta, farklı kişiler için ise birbirinden uzaklaşmaktadır. Böylece yüz tanıma yapılabilmektedir. Bu yöntemin en önemli dezavantajı ışık yoğunluğu, yüzlerin örtüşmesi gibi durumlardan etkilenebilmesidir.

**Eigenfaces Yüz Tanıma:** Eigenfaces yüz tanıma yöntemi temel bileşenler analizi tekniğini baz alarak geliştirilmiş bir yöntemdir. Boyutu yüksek verilerin boyutlarını azaltarak boyut indirgeme yapmaktadır. Bu yöntemde görüntülerin kovaryans matrisi hesaplanır. Daha sonra hesaplanan kovaryans matrisinin öz değer ve özvektörleri hesaplanır. Görüntüden elde edilen bu özvektörler kişilere özgü özellikleri içeren özyüzler

olarak adlandırılan özelliklere karşılık gelmektedir. Bir veri setindeki her kişinin fotoğraflarının özvektörleri yani özyüzleri çıkartılarak veritabanında saklanmaktadır. Daha sonra yüz tanıma yapılmak istendiğinde kişinin fotoğrafından özvektör analizi yapılarak veritabanında kayıtlı örneklerle kıyaslanır ve en yakın kişi ile eşleşme yapılır. Böylece yüz tanıma yapılır.

### 3.1.2. Derin Öğrenme Tabanlı Yüz Tanıma:

Bilgisayar donanımındaki ve derin öğrenme mimarilerindeki gelişmelerle birlikte görüntülerden birtakım özelliklerin elde edilmesi hem daha hızlı hem de daha başarılı bir şekilde yapılmaya başlanmıştır. Geleneksel yöntemlerdeki aydınlanma, yüz açısının değişmesi gibi yüz tanıma performansını olumsuz etkileyen yöntemler derin öğrenme tabanlı yüz tanıma sistemlerinde problem olmaktan çıkmıştır çünkü görüntüler ESA ile taranarak yüze ait önemli ayırt edici özellikler çıkarılmakta elde edilen bu özellikler sınıflandırma yöntemleri kullanılarak yüz tanıma yapılmaktadır. Derin öğrenme tabanlı modeller farklı ışık koşullarında, farklı açı ve içeriklere sahip yüz görüntülerindeki ayırt edici özellikleri başarılı bir şekilde öğrenebilmektedir.



**Şekil 3.5.** (a) Giriş görüntüsü (b) Yüz tespit örneği (c) Kırpılmış yüz görüntüsü (d) ResNet-29 Modeli (e) Yüz öznitelik vektör örneği (128 uzunluklu) (f) Sınıflandırma (K-NN sınıflandırıcı) (g) Yüz tanıma

Şekil 3.5'te derin öğrenme tabanlı bir yüz tanıma sisteminin genel yapısı gösterilmiştir. Yüz içeren bir giriş görüntüsü yüz tespit işleminden sonra, yüz koordinatlarına göre giriş görüntüsünden seçilmektedir. Yüz bölgesini seçme işleminden sonra yüz hizalama işlemi

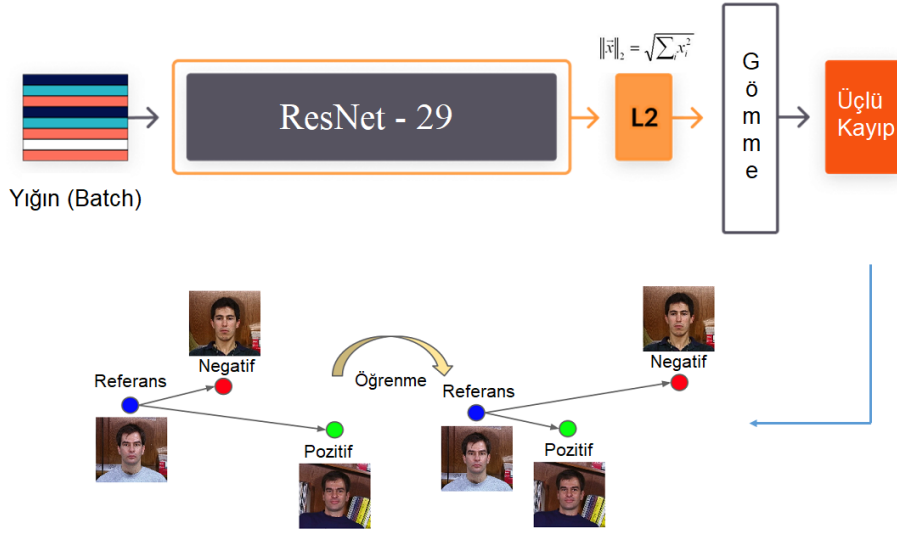


uygulanmakta böylece hem giriş görsel boyutları azaltılmakta ve yüz hizalama ile yüz merkezlenmektedir. Ön işleme adımlarından sonra yüz görüntüsü evrişimli sinir ağından geçirilmekte ve çıktı olarak belirli uzunluğa sahip öznitelik vektörleri elde edilmektedir. Elde edilen öznitelik vektörleri farklı uzunluklarda olabilmektedir. Örneğin tez kapsamında üzerinde çalışılan yüz tanıma sisteminde 128 uzunluklu öznitelik vektörleri elde edilmektedir. Elde edilen öznitelik vektörleri sınıflandırıcılar yardımıyla sınıflandırılmakta ve veritabanına kaydedilmektedir. Böylece yeni bir fotoğraf geldiğinde öznitelik vektörü çıkarılarak veritabanındaki örneklerle karşılaştırıp en yakın sonuca göre kişi tanıma yapılmaktadır.

Şekil 3.6' da ise tez konusu yüz tanıma sisteminde özellik çıkarımı için kullanılan derin öğrenme mimarisi gösterilmiştir. Yığın (Batch) giriş yüz görüntüsü kümesini ifade etmektedir. Model olarak 29 katmanlı bir ResNet ESA mimarisi kullanılmıştır. ResNet-29 modelinin çıkışı L2 normalizasyon yöntemi ile normalize edilmektedir ve sonuçta 128 uzunluklu öznitelik vektörleri elde edilmektedir. Model eğitilirken 3'lü kayıp denemeler triplet loss kayıp fonksiyonu kullanılmaktadır. Bu kayıp fonksiyonu aynı kişiye ait yüz öznitelik vektörlerini birbirine yaklaştırırken farklı kişilere ait öznitelik vektörlerini birbirinden uzaklaştırmaktadır. Yüz tanıma sisteminde kullanılan 29 katmanlı ResNet modeli ResNet-34 modelinin katman sayısının ve her katmandaki filtre sayısının yarıya indirilmesiyle elde edilmiştir. Model yaklaşık 3 milyon yüz fotoğrafından oluşan bir veri kümesiyle eğitilmiştir ve eğitiminde kullanılan eğitim veri kümesi bir dizi farklı veri kümesinden toplanmıştır. VGG veri kümesi, LFW (Labeled Faces in the Wild Home) veri kümesi örnek olarak verilebilir. Ağ eğitilirken ağırlıklar başlangıçta rastgele atanmıştır. Eğitilen model ile test aşamasında LFW veri kümesi üzerinde 0,00272732 standart sapma ile 0,993833 ortalama hata sonucu elde edilmiştir. <sup>1</sup> Klasik yöntemler ve mevcut çalışmalar da düşünüldüğünde bu sonucun çok başarılı olduğu açıktır.

---

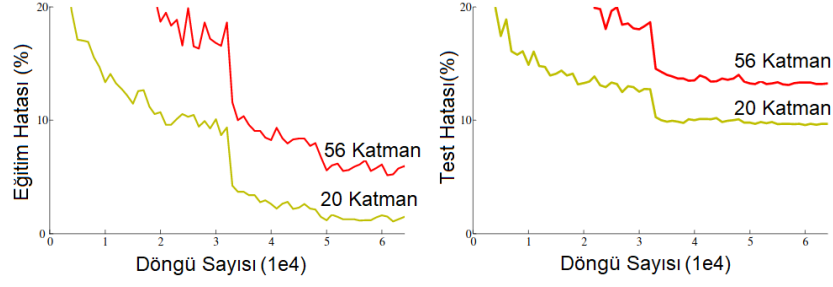
<sup>1</sup><https://github.com/davisking/dlib-models>, 2021



**Şekil 3.6.** ResNet-29 yüz tanıma modeli mimarisi

**ResNet Mimarisi:** 2012 yılında Krizhevsky ve ark. derin evrişimli sinir ağları için bir sıçrama gerçekleştirmiştir. İlk kez bir model ImageNet'te <sup>2</sup> geleneksel özellik çıkarım algoritmalarından daha başarılı olmuştur. Krizhevsky ve ark. tarafından yayınlanan derin evrişimli sinir ağı modeli AlexNet 5 ESA katmanı ve 3 tam bağlantılı katman olmak üzere 8 sinir ağı katmanından oluşmaktadır ve bu model geleneksel ESA'lara öncülük etmiştir. Evrişimli sinir ağlarına ek olarak aktivasyon fonksiyonu ve pooling katmanı da kullanılmıştır. ESA ve ardından pooling katmanlarının kullanılmasıyla birlikte geliştirilen modeller daha karmaşık özellikleri öğrenmeye başlamıştır. Bu da daha başarılı sonuçlar elde etmeye olanak sağlamıştır. ESA'ların kullanılmasıyla birlikte daha derin mimariler üretilmeye başlanmıştır. AlexNet'in ardından pek çok derin öğrenme modeli geliştirilmiştir. Ancak model mimarilerindeki derinlik arttıkça modeller daha karmaşık bir yapıya sahip olduğundan başarımlar düşmekte ve modellerin overfitting problemine daha meyilli olduğu veya öğrenmesinin durduğu görülmüştür. Aşağıda Şekil 3.7'de CIFAR-10 veri kümesi üzerinde 20 katmanlı ve 56 katmanlı normal bir ESA'nın sol tarafta eğitim ve sağ tarafta da test kayıp grafikleri verilmiştir.

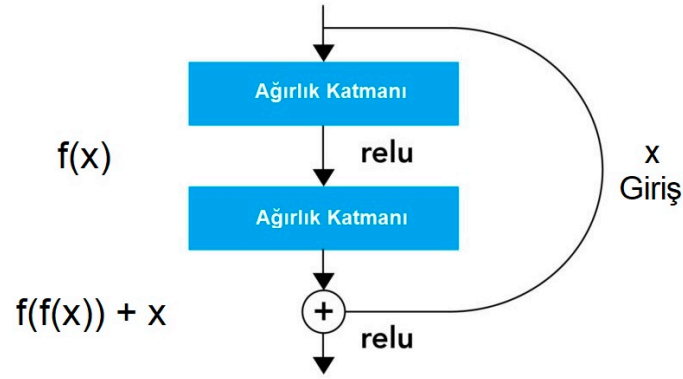
<sup>2</sup><http://www.image-net.org/>, 2021



**Şekil 3.7.** CIFAR-10 veri kümesinin 56 katmanlı ve 20 katmanlı bir model ile (a) Eğitim hatası (b) Test hatası

Şekil 3.7’den görüldüğü gibi daha fazla katman eklemek model karmaşıklığını arttırmakta ve dolayısıyla modelin overfitting problemiyle başarısızlığa meyil gösterdiğini göstermektedir. Grafikler eğitim ve test aşamasında 56 katmanlı modelin 20 katmanlı modele göre daha fazla hata yaptığını göstermektedir. Model derinliği arttıkça girişin etkisi çıkışa doğru azalmakta ve bu durum kaybolan eğim (vanishing gradient) problemine yol açmaktadır. 56 katmanlı modelin başarısız olmasının nedenleri arasında optimizasyon teknikleri, ağır ağırlıklarının rastgele başlatılması gibi durumlar gösterilebilir ancak yapılan pek çok deney ve araştırma derinlik arttıkça model başarımının azalmasının bu nedenlerden olmadığını göstermiştir.

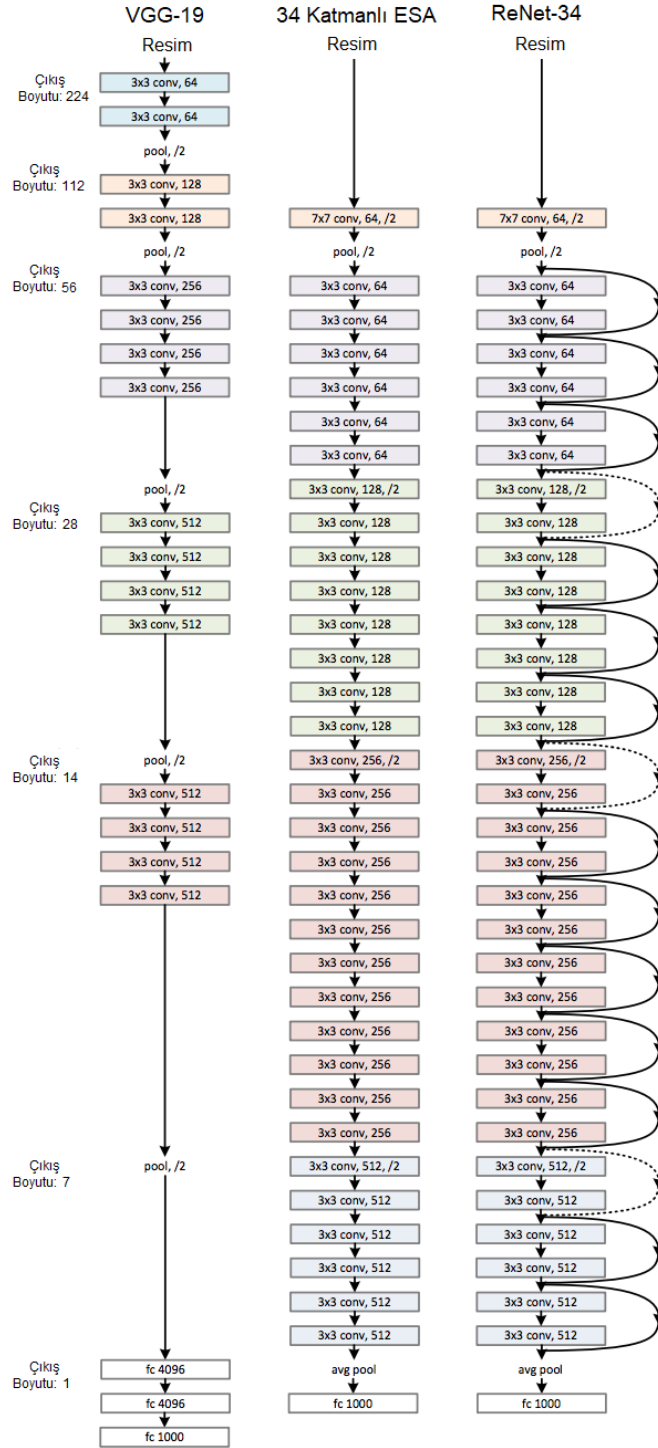
Model derinliği arttıkça yani modellerin katman sayısı arttırıldıkça modellerin eğitilme probleminin artması He ve ark. tarafından artık değerli ağların (Residual Network, ResNet) önerilmesiyle çözülmüştür. ResNet mimarisinde girişin etkisinin sonraki katmanlarda azalmasının önüne geçmek için birtakım kısayol bağlantıları kullanılmaktadır. Böylece ağ ne kadar çok katmana sahip olursa olsun girişin etkisi çıkış katmanlarına iletilebilmekte 152 hatta daha fazla katmana sahip modeller geliştirilebilmekte aynı zamanda kaybolan eğim probleminin de önüne geçilmektedir. Aşağıda Şekil 3.8’de ResNet modellerinde kullanılan kısayol bloğu gösterilmiştir. Görüldüğü gibi giriş değeri çıkışa bir kısayol bağlantısı ile iletilerek toplanmakta ve aktivasyon fonksiyonundan geçirilmektedir.



Şekil 3.8. ResNet blok yapısı

ResNet modellerinin bir diğer avantajı ise parametre sayısının modeldeki katman sayısı çok fazla olmasına rağmen daha az olmasıdır. ResNet modeli büyük ölçekli görsel tanıma yarışmasında (Large Scale Visual Recognition Challenge 2015, ILSVRC 2015) ImageNet'te % 3,57 hata oranı elde ederek 1. olmuştur. (He ve ark. 2015) Şekil 3.9a'da yaygın olarak kullanılan VGG-19 modeli, Şekil 3.9b'de 34 katmanlı normal bir ESA modeli ve Şekil 3.9c'de 34 katmanlı artık evrişimli sinir ağı örneği verilmiştir. ResNet modeli VGG-19'a göre daha fazla katmana sahiptir ancak parametre sayısı daha azdır. Tez kapsamında yapılan çalışmadaki yüz tanıma sisteminde kullanılan model 34 katmanlı ResNet modelinin katman sayısı 29'a indirilerek ve katmanlardaki filtre sayıları da yarıya düşürülerek elde edilmiştir. Böylece daha az parametre ile hesaplama maliyeti düşürülerek daha etkin bir yüz tanıma modeli elde edilmiştir. Burada verilen örnekte çıkışta sınıflandırma yapıldığından Softmax sınıflandırıcı kullanılmıştır. Yüz tanıma sisteminde ise çıkış katmanında 128 uzunluklu vektörler elde edilmektedir.

**L2 Düzenleştirme (L2 Regularization):** Düzenleştirme, eğitim sırasında bir sinir ağı modelinin karmaşıklığını azaltan ve böylece aşırı uyumu önleyen bir dizi farklı tekniği ifade etmektedir. L1, L2 ve Seyreltme (Dropout) olarak adlandırılan yaygın ve etkili 3 düzenleştirme tekniği vardır. Üzerinde çalışılan yüz tanıma sisteminde kullanılan ResNet modelinde L2 düzenleştirme tekniği kullanılmıştır. L2 düzenleştirme, tüm düzenleştirme tekniklerinin en yaygın türüdür ve aynı zamanda ağırlık kaybı (weight



Şekil 3.9. (a) VGG-19 model mimarisi (b) 34 katmanlı ESA (c) ResNet-34 modeli

decay) veya ridge regresyon olarak ta bilinir. L2 düzenleme tekniği aşırı uyum gösterme problemini önlemede veya azaltmada oldukça etkilidir.

$$\Omega(W) = \| W \|_2^2 = \sum_i \sum_j w_{ij}^2 \quad (3.1)$$

Düzenleme terimi  $\Omega$ , ağırlık matrisinin tüm değerlerinin karelerinin toplamı olan ağırlık matrislerinin Öklid Normu (veya L2 normu) olarak tanımlanır. Düzenleme terimi  $\Omega$ , skaler  $\alpha$  değeri ikiye bölünerek ağırlıklandırılır ve normal kayıp fonksiyonuna eklenir. Böylece 3.2 eşitliği elde edilir.

$$\hat{\mathcal{L}}(W) = \frac{\alpha}{2} \| W \|_2^2 + \mathcal{L}(W) = \frac{\alpha}{2} \sum_i \sum_j w_{ij}^2 + \mathcal{L} \quad (3.2)$$

$\alpha$  bazen düzenleme oranı olarak adlandırılır ve sınır ağına eklenen ek bir hiper parametredir. Temel olarak  $\alpha$  modelin ne kadar düzenleneceğini belirlemektedir. Bir sonraki adımda, yeni kayıp fonksiyonunun gradyeni hesaplanır ve ağırlıkların güncellenmesi için önceki ağırlığa eklenir. Yapılan hesaplama aşağıda eşitlik 3.3'de verilmiştir.

$$W_{new} = W_{old} - \varepsilon(\alpha W_{old} + \nabla_W \mathcal{L}(W_{old})) \quad (3.3)$$

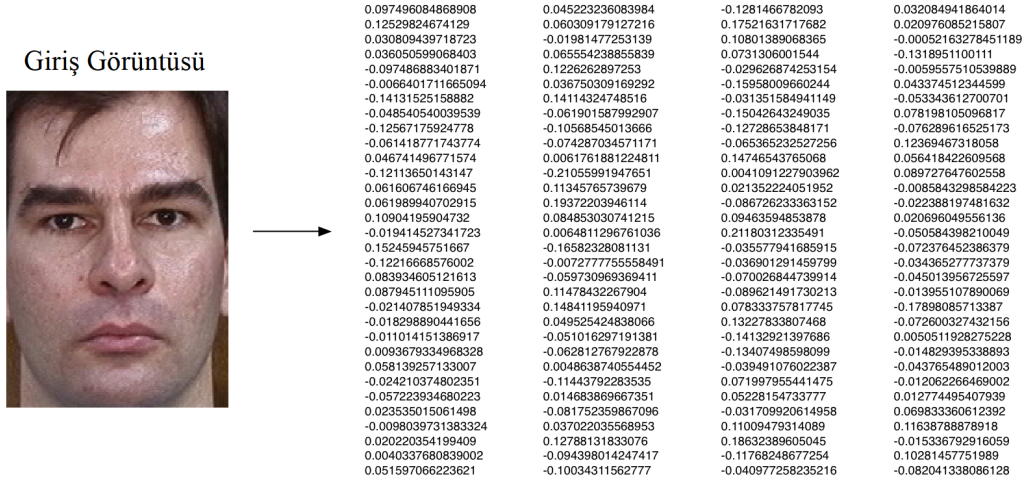
Eşitlik 3.3 düzenlenerek

$$W_{new} = (1 - \varepsilon\alpha)W_{old} - \varepsilon\nabla_W \mathcal{L}(W_{old}) \quad (3.4)$$

3.4 denklemini elde edilir. Düzenleme işlemi ile yapılan en önemli değişiklik bir düzenleme terimi eklenerek kayıp fonksiyonunun gradyeninden bağımsız olarak ağırlıklar her güncellendiğinde ağırlıkların biraz daha azaltılmasıdır. Böylece model eğitilirken daha başarılı sonuçlar elde edilmektedir.

**Öznitelik Vektörleri:** Yüz tanıma sistemlerindeki temel amaç yüz görüntüsünden birtakım benzersiz özellikler çıkartarak yüz tanıma işlemi gerçekleştirmektir. Görüntüden

özelliklerin çıkarılması ile ilgili pek çok yöntem bulunmaktadır. LBP, LDA vb. yöntemler örnek olarak verilebilir. Ancak evrişimli sinir ağları ile derin öğrenme mimarilerinin gelişmesi ile birlikte görüntülerden özellik çıkarımı konusunda önemli gelişmeler yaşanmıştır. Üzerinde çalışılan yüz tanıma sisteminde kullanılan ResNet modeli ile yüz görüntüleri işlenerek 128 uzunluklu kişiye özgü vektörler elde edilmektedir. Bu vektörler model eğitilirken aynı kişiler için birbirine yakın değerlere sahip iken, farklı kişiler için birbirine uzak değerlere sahiptir. Aynı zamanda kişilerin birbirinden ayırt edici özelliklerini temsil etmektedir. Yine bu vektörler K-NN sınıflandırıcı ve benzeri sınıflandırıcılar ile sınıflandırılarak kişi tanıma yapılmaktadır. Uygulamada K-NN sınıflandırıcı tercih edilmiştir.



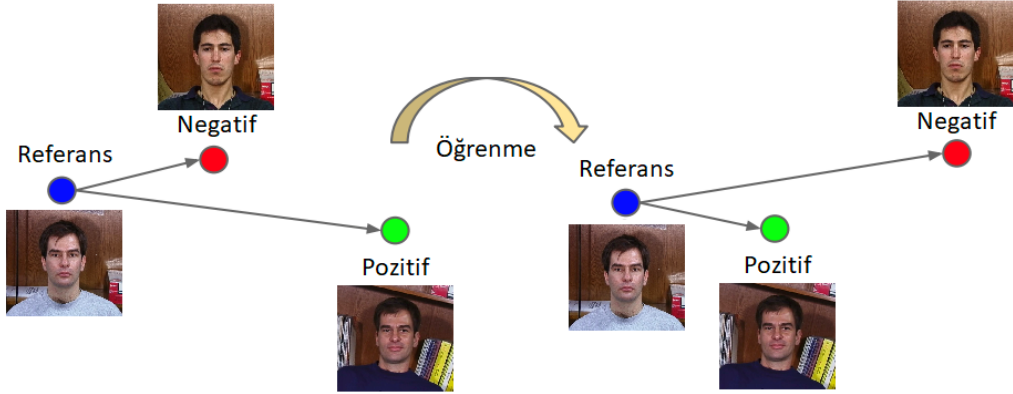
**Şekil 3.10.** (a) Giriş görüntüsü (b) ResNet-29 modeli ile elde edilen 128 uzunluklu öznitelik vektör örneği

Şekil 3.10'da yüz tanıma modeli sonucu elde edilen kişiye özgü 128 uzunluklu öznitelik vektörleri örneği verilmiştir. Buradaki değerler her kişi için birbirinden farklı, aynı kişinin farklı fotoğrafları için ise birbirine yakın olacak şekilde üretilmektedir. Model kişileri ayırt edecek özellikleri çıkarmayı öğrenmektedir.

**Üçlü Kayıp (Triplet Loss):** Yüz tanımadan nesne algılamaya, konuşma bölümü etiketlemeden (Pos Tagger) NLP ile belge sınıflandırmaya kadar birçok uygulama, çok sınıflı bir

sınıflandırma problemi olarak görülmektedir. Tipik softmax tabanlı derin sinir ağının, çıktı katmanındaki sınıf sayısı çok yüksek olduğunda başarısı düşmektedir. Bu problemin çözümüne yönelik olarak giriş verisinden birtakım vektörler çıkarıp yüksek boyutlu vektör uzayında benzer verileri yakın bölgede, benzer olmayan verileri ise birbirinden uzak noktaya yansıtma yöntemi önerilmiştir. Bu bağlamda üçlü kayıp (Triplet Loss) denen kayıp türü kullanılmaktadır. Üçlü kayıp derin öğrenme modeli ile elde edilen çıktının aynı giriş örnekleri için birbirine yakın, farklı giriş örnekleri için birbirinden uzak olmasını sağlayan bir kayıp türüdür.

Üçlü kayıp fonksiyonunun altında yatan temel mantık, bir referans görüntü alınarak, referans görüntünün negatif örneklere yani referans görüntü dışındaki tüm görüntülere uzak, pozitif örneklere yani referans ile aynı görüntülere yakın olmasını sağlamaktır. Şekil 3.11’de bu duruma örnek gösterilmiştir. Yüz tanıma modeli eğitilirken bir referans yüz görüntüsü alınır, referans yüz görüntüsü ile aynı örnekler birbirine yaklaştırılacak ve farklı kişiler birbirinden uzaklaştırılacak şekilde model eğitilir.



**Şekil 3.11.** Üçlü kayıp örneği

Üçlü kayıp şu şekilde hesaplanmaktadır:

1. Referans giriş görüntüsünden öznelik vektörleri çıkarılır (a), üzerinde çalışılan yüz tanıma sisteminde 128 uzunluklu yüz öznelik vektörleri elde edilmektedir.
2. Pozitif giriş görüntüsü örneğinden öznelik vektörleri çıkarılır (p) (Referans giriş



örneği ile aynı sınıf/kişi)

3. Negatif giriş görüntüsü örneğinden öznitelik vektörleri çıkarılır (n) (Referans giriş örneğinden farklı bir sınıf/kişi)
4. Referans örneğinden elde edilen öznitelik vektörü ile pozitif örnekten elde edilen öznitelik vektörünün öklid uzaklığı (Euclidean Distance) hesaplanır,  $d(a, p)$ , aynı zamanda referans örnekten elde edilen öznitelik vektörü ile negatif örnekten elde edilen öznitelik vektörünün öklid uzaklığı hesaplanır,  $d(a, n)$ . İdeal olarak ilk mesafe yani  $d(a, p)$  olabildiğince küçük, ikincisi  $d(a, n)$  ise olabildiğince büyük olmalıdır.
5. Üçlü kayıp şu şekilde hesaplanmaktadır:

$$Loss = \sum_{i=1}^N [\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha]_+ \quad (3.5)$$

$f_i^a$ , referans çıktı öznitelik vektörü ifade eder

$f_i^p$ , pozitif çıktı öznitelik vektörü ifade eder

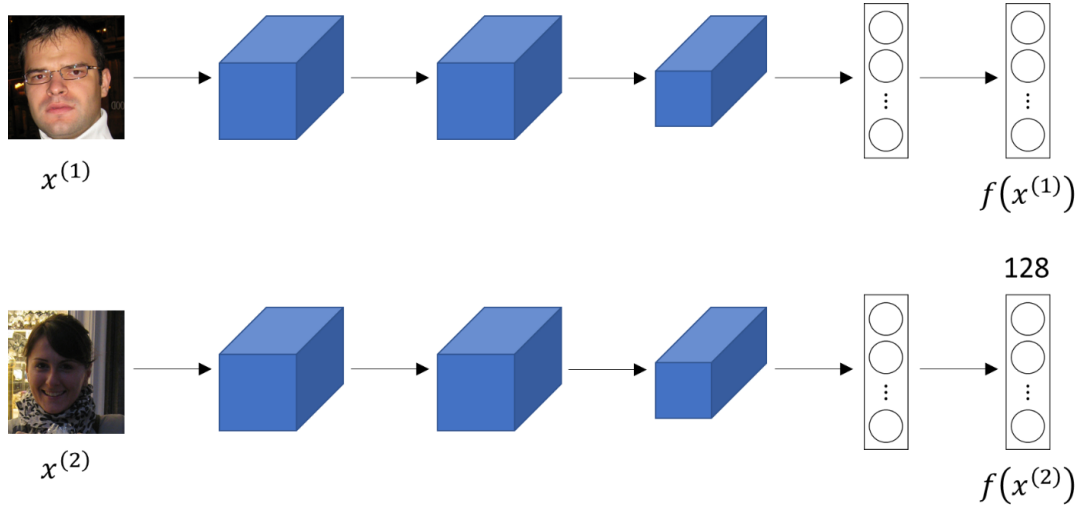
$f_i^n$ , negatif çıktı öznitelik vektörü ifade eder

$\alpha$ , ağın  $f_i^a - f_i^p = f_i^a - f_i^n = 0$ 'a göre optimize etmeye çalışmamasını sağlamak için kullanılan bir sabittir.

$[\dots]_+$ , maksimuma eşittir:  $\max(0, \text{toplam})$

**Siyam Ağı (Siamese Network)** Siyam ağı, iki giriş arasında nasıl ayırım yapılacağını öğrenen bir tür sinir ağı mimarisidir. Hangi görüntülerin benzer hangilerinin olmadığını öğrenmelerini sağlamaktadır. Yüz görüntüleri için aynı kişiye ait görüntülerin benzerliğini, farklı kişiler için ise benzer olmadıklarını öğrenmektedir. Siyam ağları, her biri aynı parametre ve ağırlıklara sahip iki özdeş sinir ağından oluşur. İlk olarak, her ağ giriş olarak iki giriş görüntüsünden birini alır. Ardından, her bir ağın son katmanlarının çıktıları, görüntülerin aynı kimliği içerip içermediğini belirleyen bir çıktı elde edilir.

Yukarıda Şekil 3.12'de verilen örnek, siyam ağını kullanan güzel bir yüz tanıma örneğidir. İlk ağın girdisi bir dizi evrişim ve örnekleme katmanından geçirilir, ardından elde edilen



**Şekil 3.12.** Siyam ağı ile üretilen 128 uzunluklu öznitelik vektör örnekleri

çıktı tam bağımlı katmandan geçirilerek öznitelik vektörleri elde edilmektedir. Son vektör  $f(x^{(1)})$ ,  $x^{(1)}$  giriş görüntü örneğinin öznitelik vektörünü temsil etmektedir. Daha sonra,  $x^{(2)}$  girdisi birinci ağ ile tamamen aynı olan ikinci ağdan geçirilerek  $x^{(2)}$  giriş görüntü örneği için de  $f(x^{(2)})$  öznitelik vektörü elde edilir.

$f(x^{(1)})$  ve  $f(x^{(2)})$  arasındaki mesafe hesaplanır bunun için genellikle Öklid uzaklığı tercih edilir. Elde edilen sonuç belirlenen bir eşik değerinden büyükse iki görüntü aynı, eşik değerinden küçükse iki görüntü farklıdır sonucuna varılır. Hesaplamalar eşitlik 3.6'da gösterilmiştir.

$$d(x^{(1)}, x^{(2)}) = \Gamma \| f(x^{(1)}) - f(x^{(2)}) \|_2^2 \quad (3.6)$$

$x^{(1)}$ ,  $x^{(2)}$  aynı kişiler ise,  $\| f(x^{(1)}) - f(x^{(2)}) \|_2^2$  küçük,

$x^{(1)}$ ,  $x^{(2)}$  farklı kişiler ise,  $\| f(x^{(1)}) - f(x^{(2)}) \|_2^2$  büyük

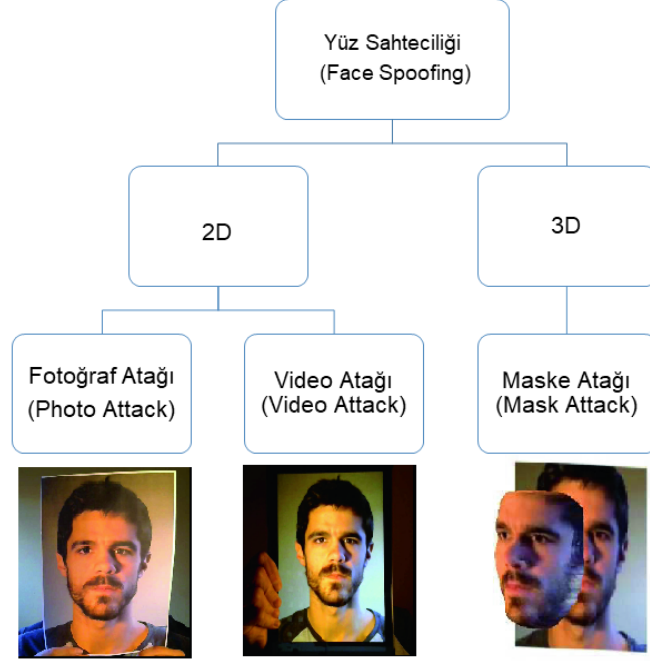
**K-NN Sınıflandırıcı:** K-en yakın komşuluk sınıflandırıcı ile ilgili detaylı bilgi bölüm 2.11.1'de verilmiştir. Eğitilen ResNet-29 modeli ile her yüz görüntüsü ile ilgili 128 uzunluklu öznitelik vektörleri elde edilmektedir. Elde edilen bu vektörler K-NN sınıflandırıcı ile

sınıflandırılarak bir veritabanı oluşturulmaktadır. Veritabanı kişilerin farklı fotoğraflarından elde edilmiş öznitelik vektörlerinin K-NN sınıflandırıcı ile sınıflandırılmış örneklerinden oluşmaktadır. Yüz tanıma yapılmak istendiğinde ResNe-29 modeli ile giriş görüntüsünden öznitelik vektörü çıkarılmaktadır. Daha sonra seçilen değerine göre örneğin k değerinin 3 olduğu varsayılırsa, tanınacak görüntüden elde edilen öznitelik vektörü veritabanında kayıtlı örneklerle kıyaslanarak en yakın uzaklığa sahip 3 örnek bulunmaktadır. Bulunan 3 örnekten en fazla örnek içeren sınıf ile eşleştirilerek yüz tanıma yapılmaktadır. Burada 3 değerine göre bir oylama yapılmaktadır. Örneğin en yakın 3 örnekten 2 tanesi a kişisine, 1 tanesi ise b kişisine ait olsun. Bu durumda yeni kişi a kişisi ile eşleştirilecektir.

### **3.2. Canlı - Cansız Yüz Tespiti**

Yüz tanıma sistemlerinde son yıllarda derin öğrenme yöntemlerindeki gelişmelerle birlikte önemli ilerlemeler olmuştur. Pek çok farklı alanda yüz tanıma sistemleri kullanılmaya başlanmıştır. Örneğin suçluların tespit ve takibinde, personel giriş kontrol sistemlerinde personel takibi, mobil cihazlarda vb. sistemlerde sıklıkla kullanılmaktadır. Yüz tanıma sistemlerinin başarılı olmasının yanında önlem alınması gereken ciddi birtakım ataklar bulunmaktadır. Örneğin bir cep telefonuna yüz tanıma sistemi ile giriş yapıldığı düşünülürse telefonun ait olduğu kişinin fotoğraf, video, maske vb. yöntemlerle telefon kilidi açılabilir ve kişinin kişisel bilgilerinin çalınması başta olmak üzere kişi pek çok sıkıntıya maruz kalabilir. Bu gibi tehditler göz önüne alındığında yüz tanıma sistemi tarafından analiz edilen görüntünün gerçek bir kişiye mi ait olduğu yoksa bir fotoğraf, video veya maske mi olduğunun tespit edilmesi güvenlik açısından ciddi önem teşkil etmektedir. Yani kişinin canlı mı yoksa cansız mı olduğunun tespiti çok önemli bir hal almaktadır.

Yüz tanıma sistemlerini aldatmak için farklı yöntemler mevcuttur. Şekil 3.13'te bu yöntemler gösterilmiştir.



Şekil 3.13. Yüz tanıma sistemi saldırı örnekleri

**Baskı atağı (Print attack) :** Yüz tanıma sistemini aldatmak için kişilerin fotoğrafları kullanılmaktadır. Görüntü bir yazıcıdan çıktı alınır kameraya gösterilerek sistem aldatılır. Günümüzde özellikle sosyal medya vb. paylaşım platformların yaygın kullanılması sonucu kişilerin yüz görüntülerine kolaylıkla ulaşılabilir bu nedenle yaygın bir saldırı türüdür.

**Video atağı (Video Attack/Replay Attack):** Kişilerin video görüntülerini kullanarak yüz tanıma sistemlerinin aldatıldığı bir atak türüdür. Bu yaklaşım kişilerin fotoğraflarını kameraya tutmaya göre davranış ve yüz hareketlerinin daha doğal görünmesini sağlamaktadır. Ağız, göz veya baş hareketleri analiz edilerek görüntülerin kullanıldığı baskı atakları önlenebilir ancak video görüntülerinde ağız hareketi, göz kırpma, başı hareket ettirme gibi durumları da içerdiğinden yüz tanıma sistemlerini aldatmak için etkili bir yöntemdir. Tablet veya akıllı telefonlar kullanılarak kolaylıkla gerçekleştirilebilen bir atak türüdür.

**3D Maske Atak (3D Mask Attack):** Kişilerin yüz maskesi kullanılarak gerçekleştirilen bir atak türüdür. Yüz videosu oynatmaya göre daha önlem alınması daha zor bir saldırdır. Doğal yüz hareketlerine ek olarak derinlik sensörleri gibi bazı ek güvenlik araçlarını aldatmak için de kullanılan bir yöntemdir. Normal yüz maskeleri veya kağıt çıktı maskeleri bu saldırı türünde kullanılabilir.

**Yüz Tanıma Sistemlerine Saldırlara Karşı Kullanılan Güvenlik Yöntemleri:** Yüz tanıma sistemleri normal koşullar göz önüne alındığında özellikle derin öğrenme yöntemleriyle birlikte çok başarılı sonuçlar üretmektedir ancak bu sistemler bazı yöntemlerle kolaylıkla yanıltılabildiği için saldırıya açıktır. Bu nedenle, güvenli bir yüz tanıma sistemi tasarlamak için, yüz tanıma sistemlerini aldatmaya yönelik saldırılara önlemler alınması gerekmektedir. Yüz tanıma sistemlerinin güvenliğini arttırmak ve bu sistemleri saldırılara karşı dirençli hale getirmek için farklı yöntemler bulunmaktadır. Bu yöntemler şu şekilde sıralanabilir:

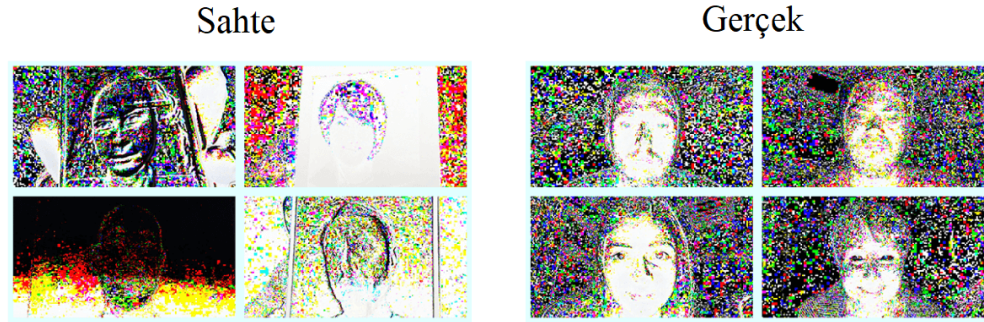
**Yerel İkili Örüntüler (Local Binary Pattern):** Yerel ikili örüntüler (LBP), görüntü piksellerini komşularına göre eşikleyerek, görüntü doku özelliklerini analiz ederek yüz tanıma sistemlerine karşı saldırılara önlem alınmasını sağlayan bir tekniktir. Görüntü  $n \times n$  küçük parçalara bölünür, yerel ikili desen (LBP), görüntü piksellerini komşularına göre eşikleyerek, doku görüntü analizinin yapıldığı yüz tanıma saldırılarını önleme tekniğidir.

Bir görüntü  $n \times n$ 'lik bir pencere ile taranır. Kullanılan  $n \times n$  pencerenin karşılık geldiği pikseller pencerenin merkez konumundaki değerle kıyaslanır ve eğer komşu piksel merkezden büyükse 1, küçükse 0 değeri o komşu piksele atanır. Bu işlem tüm görüntü boyunca uygulanır. Sonuç olarak LBP histogramları elde edilir ve sınıflandırma teknikleri kullanılarak görüntünün sahte bir kişiye mi yoksa gerçek bir kişiye mi ait olduğu tespit edilir.

**Derin Öğrenme Tabanlı Teknikler - Evrişimli Sinir Ağları:** Evrişimli sinir ağları yüz tanımda kullanılmasının yanında yüz tanıma sistemleri aldatan yöntemlere karşı da kullanılabilir. Kamera karşısındaki kişi gerçek bir kişi mi yoksa kişinin video,

fotoğraf vb. görüntüleri mi gösteriliyor bunun tespiti “Kişi gerçek – Onaylandı” veya “Kişi sahte – Onaylanmadı” şeklinde ikili bir sınıflandırma problemi olarak ele alınabilir. Ancak bu problemin çözümünde evrışimli sinir ağlarının göreceği veya anlayacağı belirli ayırt edici özellikler yoktur. Buradaki temel ilke kişilerin anlık olarak çekilmiş canlı fotoğraflarıyla, telefon, tablet, bilgisayar vb. araçlardan elde edilen fotoğrafları kullanarak insan gözünün algılayamadığı ayırt edici özellikleri algılamasıdır. Telefon, tablet, bilgisayar vb. cihazlardaki veya yüz görüntü çıktısı fotoğrafları normal kamera karşısındaki kişinin görüntüsüne göre daha belirsiz, bulanık veya bozulmuş olabilir. Bu farklılıklar ESA ile tespit edilerek karar verilebilir. Ancak kullanılan cihaz özelliği, çevresel koşullar gibi faktörler sonuçları etkilemektedir. ESA ile sonuç alınabilir ancak aşırı uyum gösterme probleminin olması muhtemeldir. Gerçek koşullarda yüz tanıma sisteminin aldatılmasını önlemek performans konusunda daha etkin tekniklere ihtiyaç duyulmaktadır.

**Aktif Flaş (Active Flash) Tekniği:** Yüzdeki ışık yansımalarını kullanarak yüz tanıma sistemlerini aldatan yöntemlere karşı önlemlerin alındığı etkili bir tekniktir. Buradaki fikir, cihaz ekranını ek bir ışık kaynağı olarak kullanarak ışık ortamını değiştirmektir. Ekranı kaplayan beyaz alan, yüzde uygun yansıma üretir.



**Şekil 3.14.** Aktif flaş tekniği örneği

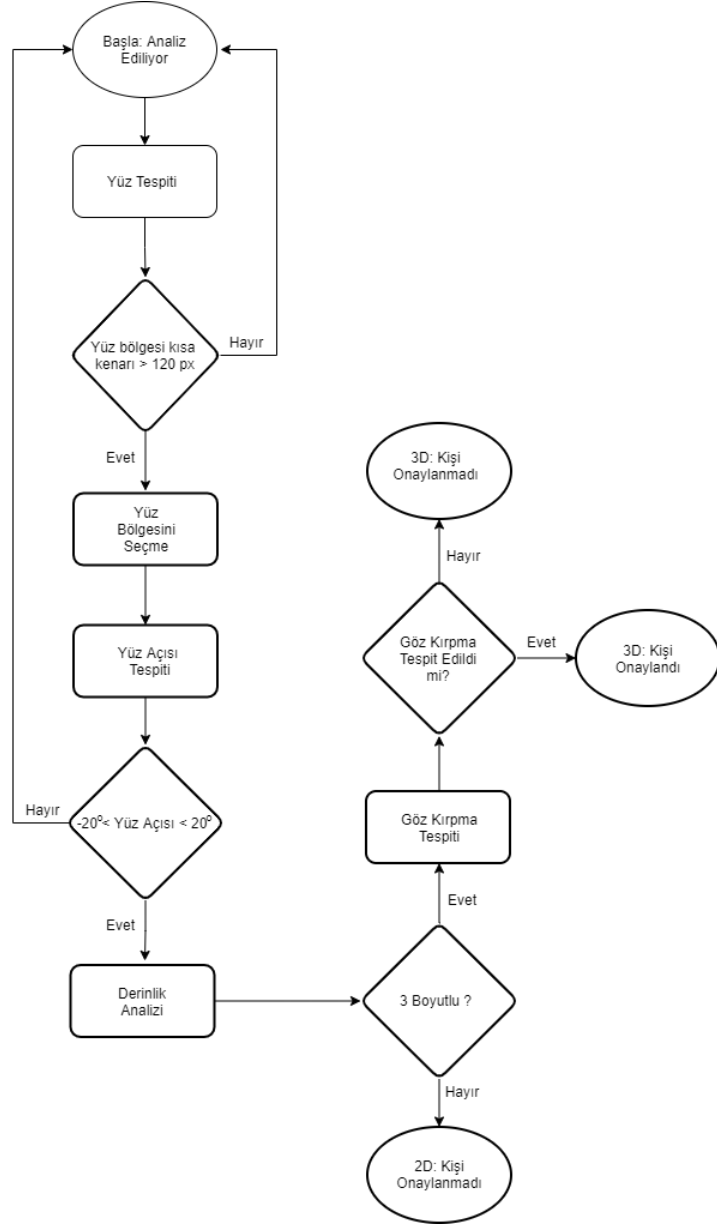
Şekil 3.14’te aktif flaş tekniği ile ilgili örnek gösterilmiştir. Gerçek yüzler, yüzeylerindeki farklılıklar nedeniyle sahte olanlardan ayırt edilmektedir. Canlı yüz örnekleri daha belirgin iken cansız yüz örneklerindeki ayrıntılar ve belirginlik daha azdır. Aktif flaş tekniği

canlı ve cansız kişiler için yüz özelliklerini ayırmaya ve sınıflandırmaya yardımcı olmaktadır.

**3D Derinlik Kamerası:** Yüz tanıma sistemlerini aldatmak için kullanılan yöntemlere karşı kullanılan en güvenli yöntemlerden biridir. Canlı ve cansız yüz tespiti için 3D kamera kullanılmaktadır. Canlı bir insan yüzü derinliğe sahiptir bu nedenle canlı - cansız yüz tespiti ve saldırılara karşı önlem almada yüksek doğruluk sağlamaktadır. Telefon, tablet, bilgisayar, vesikalık fotoğraf veya yüz fotoğraf çıktıları düz bir yüzeye sahip olacaklarından 3D derinlik kamerası tarafından düz yüzeyleri kolaylıkla algılanabilmekte ve gerekli önlem alınabilmektedir. Akıllı telefonlar veya kameralar derinlik sensörleriyle derinliği algılayabilmektedir bu da bu yöntemin uygulanabilirliğini arttırmaktadır. 3D derinlik kameralarının maliyetinin fazla olması bu yöntemin en önemli dezavantajlarından biridir ancak güvenlik ve başarımının yüksek olması bu yöntemi ön plana çıkarmaktadır. Tez konusu derin öğrenme tabanlı yüz tanıma sisteminde güvenliğin sağlanması ve sistemin aldatici yöntemlere karşı çözüm üretmesi için 3D derinlik kamerası ile algılanan derinlik bilgisi kullanılmıştır.

**Göz Kırpma Tespiti (Eye Blink Detection):** Bir insan normal şartlarda dakikada ortalama 15 ila 30 kez göz kırpmaktadır. Göz kırpma sırasında gözler yaklaşık 250 ms kapalı kalmaktadır. Modern kameralar, saniyede 30 kare yani 50 ms gibi çok daha küçük aralıklarla videoları kaydedebilmektedir. Böylece doğal göz kırpma hareketi analiz edilerek canlı yüz görüntüsü ve sahte yani cansız yüz görüntüsü arasındaki fark tespit edilebilmektedir. Çünkü kameranın karşısındaki kişi canlı bir kişi ise göz kırpma hareketi analiz edilerek kişinin canlı olduğu tespit edilebilir ve yüz tanıma sisteminin vesikalık fotoğraf, telefon veya tablet yüz görüntüsü, yüz çıktısı gibi örneklerle sistemin aldatılması önlenir. Ancak video görüntülerinin kullanılması bu tekniğin zayıf noktalarından biridir çünkü video görüntüleri kişinin göz kırpma hareketini içerebilmektedir.

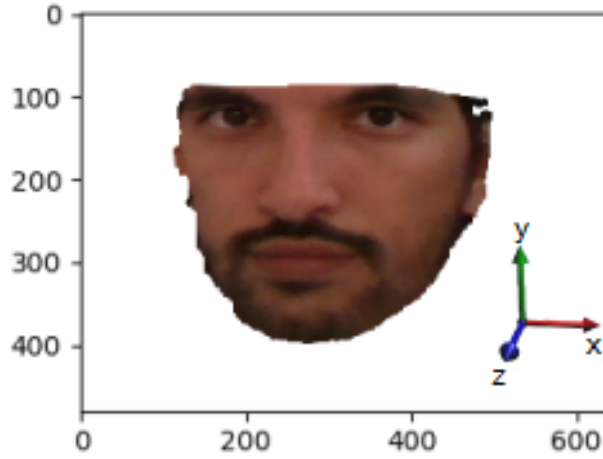
**Canlı – Cansız Yüz Tespiti İçin Kullanılan Yöntem:** Tez kapsamında ise üzerinde çalışılan yüz tanıma sistemini daha güvenli hale getirmek için hem 3D derinlik kamerası ile derinlik analizi hem de göz kırpması tespiti yöntemi birlikte kullanılmıştır. Böylece 2 boyutlu, 3 boyutlu pek çok saldırı yöntemine karşı etkili bir sistem geliştirilmiştir.



**Şekil 3.15.** Canlı - Cansız Yüz Tespiti Akış Diyagramı



Şekil 3.15'te canlı - cansız yüz tespiti için kullanılan yöntemin akış diyagramı gösterilmiştir. Öncelikle derinlik kamerasından alınan görüntü üzerinde yüz tespiti yapılmaktadır. Yüz tespiti yapıldıktan sonra tespit edilen yüz boyutlarının kısa olanı 120 pikselden küçükse işlem yapılmamakta ve sistem analize yeniden başlamaktadır. Yüz bölgesi kısa kenarı 120 pikselden büyük ise tespit edilen koordinatlara göre yüz bölgesi seçilmektedir. Yüz bölgesi seçildikten sonra derinlik bilgisinin karşılık geldiği koordinatlar ile derinlik kamerasından elde edilen RGB görüntünün piksellerinin karşılık geldiği koordinatlar örtüşmemektedir. Bu nedenle derinlik bilgisi ile karşılık geldiği piksel koordinatlarının eşleştirilmesi için hizalama işlemi uygulanmaktadır. Derinlik analizi için kişinin kameraya düz bakması istenmektedir çünkü en iyi derinlik bilgisi kişi kameraya doğrudan baktığında alınmaktadır. Bu nedenle evrişimli sinir ağları kullanılarak kişinin kameraya göre yüz açısı tespit edilmektedir. Yüz açısı -20 ile 20 derece arasında ise derinlik analizi yapılmaktadır. Derinlik analizi sonucunda yüzün 3 boyutlu olup olmadığı kararlaştırılmaktadır. Sistem derinlik algılamazsa alınan görüntünün 2 boyutlu olduğuna karar vererek kişiyi onaylamamaktadır. Eğer derinlik algılanırsa sonrasında göz kırpma tespiti yapılmakta ve göz kırpma tespit edilirse sistem yüzün canlı bir yüz olduğuna karar vermekte ve kişiyi onaylamaktadır. Göz kırpma tespit edilmezse derinlik algılanırsa bile kişi onaylanmamaktadır.



**Şekil 3.16.** Yüz tespiti yapılarak yüz bölgesi seçilmiş ve derinlik bilgisi ile RGB piksel koordinatları eşleştirilerek hizalanmış 3 boyutlu (3D) yüz görüntüsü

**3D Derinlik Kamerası ile Derinlik Analizi:** Derinlik analizi için Şekil 3.16’da gösterildiği gibi yüz bölgesi tespit edilip seçilmiş ve hizalama işlemi uygulanmış yüz görüntüsünün  $x$ ,  $y$  ve  $f_z$  ile temsil edilen özellik matrisi çıkarılmaktadır.  $x$ ,  $x$  yönündeki piksel değerlerini  $y$ ,  $y$  yönündeki piksel değerlerini  $f_z$  ise  $x$  ve  $y$  koordinatlarına karşılık gelen derinlik bilgisini temsil etmektedir. Derinlik ise yüz bölgesinde bir noktanın kameraya uzaklığını temsil etmektedir. Birimi metre (m) ile temsil edilmektedir. Yüz bölgesinin derinlik bilgisini de içerek 3 boyutlu özellik matrisi çıkarıldıktan sonra derinlik bilgisinin  $x$  ve  $y$  noktalarındaki değişimi analiz edilmektedir. Bunun için  $z$  derinlik bilgisinin 3.7’de gösterildiği gibi her nokta için  $x$  ve  $y$  koordinatlarına göre türevi hesaplanmıştır.  $z$  derinlik bilgisinin her nokta için  $x$  ve  $y$  koordinatlarına göre türevi :

$$\nabla f(z) = \frac{\partial f_z}{\partial x} \vec{e}_x + \frac{\partial f_z}{\partial y} \vec{e}_y \quad (3.7)$$

Derinlik bilgisinin türev hesaplandıktan sonra ise Eşitlik 3.8 ve Eşitlik 3.9’da gösterildiği gibi her bir nokta için genlik ve açı hesaplanmıştır:

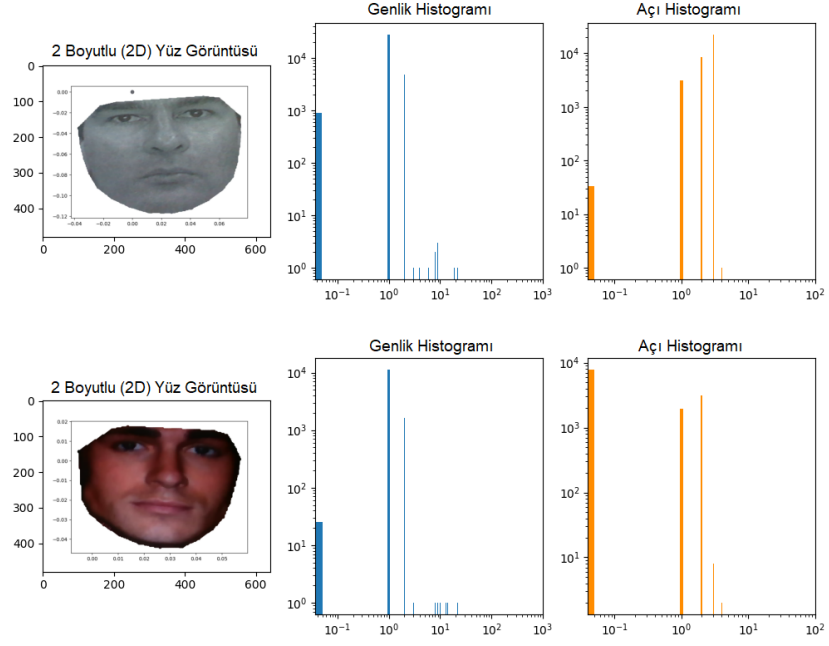
$$\text{Genlik} = |\nabla f| = \sqrt{a^2 + b^2} \quad (3.8)$$

$$\text{Açı} = \tan^{-1}\left(\frac{b}{a}\right) \quad (3.9)$$

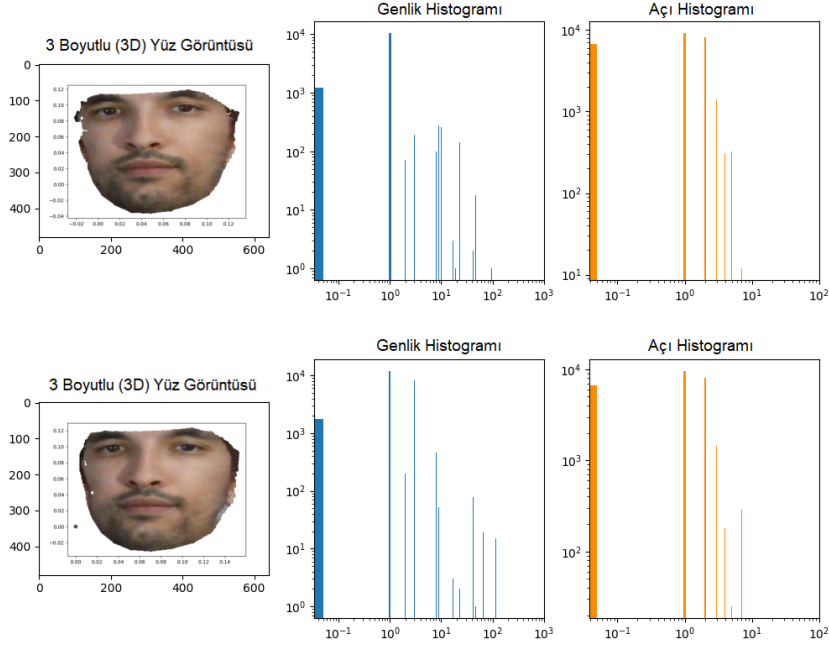
$a$ ,  $z$  derinlik bilgisinin  $x$ ’e göre türevi

$b$ ,  $z$  derinlik bilgisinin  $y$ ’ye göre türevi

Hesaplanan genlik ve açı değerlerinin histogramları çıkarılmıştır. Şekil 3.17’de solda 2 boyutlu yüz örnekleri, ortada ve sağda bu görüntüden elde edilmiş  $z$  derinlik bilgisinin  $x$  ve  $y$  koordinatlarına göre türevinin genlik ve açı histogramları verilmiştir. 2 boyutlu görüntüler için beklendiği gibi genlik histogramları sıfır ve etrafında toplanmıştır. Şekil 3.18’de ise solda 3 boyutlu yüz örnekleri, ortada ve sağda ise bu görüntüden elde edilmiş  $z$  derinlik bilgisinin  $x$  ve  $y$  koordinatlarına göre türevinin genlik ve açı histogramları verilmiştir. 3 boyutlu görüntüler için beklendiği gibi genlik histogramları farklılık göstermektedir.

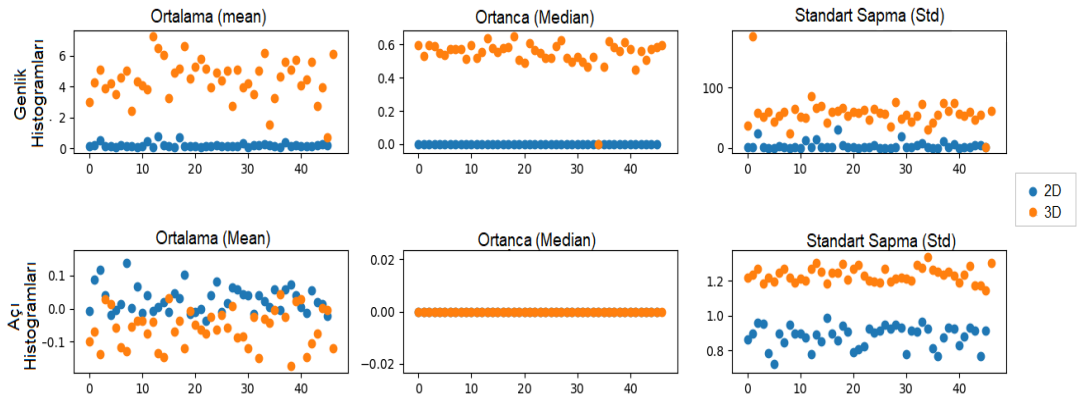


**Şekil 3.17.** 2 boyutlu (2D) yüz örnekleri,  $z$  derinlik değerlerinin gradyen genlik ve açı histogramları



**Şekil 3.18.** 3 boyutlu (3D) yüz örnekleri,  $z$  derinlik değerlerinin gradyen genlik ve açı histogramları

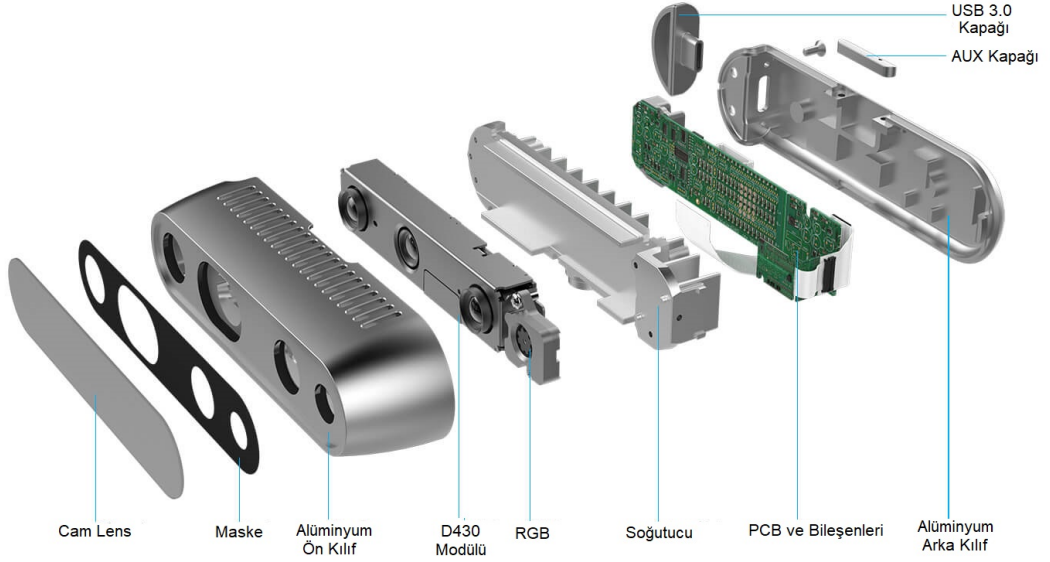
Son adımda ise elde edilen genlik ve açı histogramlarının ortalama (mean), ortanca (median) ve standart sapma gibi istatistiksel analizleri yapılmıştır. Şekil 3.19’da Intel Realsense D435 derinlik kamerasından alınan 2 boyutlu 50 örneğin ve 3 boyutlu 50 örneğin derinlik bilgilerinin türevlerinin genlik ve açı histogramlarından elde edilen ortalama, ortanca ve standart sapma değerleri gösterilmiştir. Yapılan analizler sonucunda ortanca eşik değeri 0,3 olarak seçildiğinde sistemin canlı ve cansız yüzleri ayırt etmede çok başarılı olduğu görülmüştür. Hesaplanan ortanca değeri 0,3’ten büyük ise kişi canlı, 0,3’ten küçük ise kişinin cansız olarak kabul edilmektedir.



**Şekil 3.19.** 3D yüz görüntüleri derinlik bilgisi gradyen genlik ve açı histogramlarının ortalama, ortanca, standart sapma analiz grafiği

3D derinlik bilgisi kullanarak canlılık tespitinin yapılması yüz tanıma sistemlerin aldatılmasına yol açan hemen hemen tüm yöntemler için başarılı sonuç vermektedir. Ancak bir kişinin yüz görüntüsünün yazıcıdan çıktısı alınıp bu görüntü kıvrıldığında bir derinlik oluşturulabilmekte ve derinlik kamerası ile analiz edildiğinde derinlik algılandığı için sistem yanıltılabilmektedir. Bu nedenle çok başarılı bir güvenli yüz tanıma sistemi geliştirmek için derinlik analizine ek olarak göz kırpma tespiti yöntemi kullanılmıştır. Kamera karşısına geçen kişinin derinlik bilgisi algılanırsa ek olarak göz kırpma analizi yapılmakta ve kişinin canlı olup olmadığının kararı verilmektedir. Kamera 3D derinlik bilgisi algılamadığında ise kişinin canlı olmadığı başarılı bir şekilde tespit edilmektedir. Bu durumda göz kırpma analizine ihtiyaç duyulmamaktadır.

**3D Derinlik Kamerası - Intel Realsense D435:** 3D derinlik analizi için Intel Realsense D435 kamera tercih edilmiştir. Yapılan araştırmalar sonucu kameranın derinlik algılamada başarılı olması ve etki mesafesinin uzun olması dikkat çekmiştir. Bu nedenle kameranın geliştirilen sistem için uygun olduğu sonucuna varılmıştır.

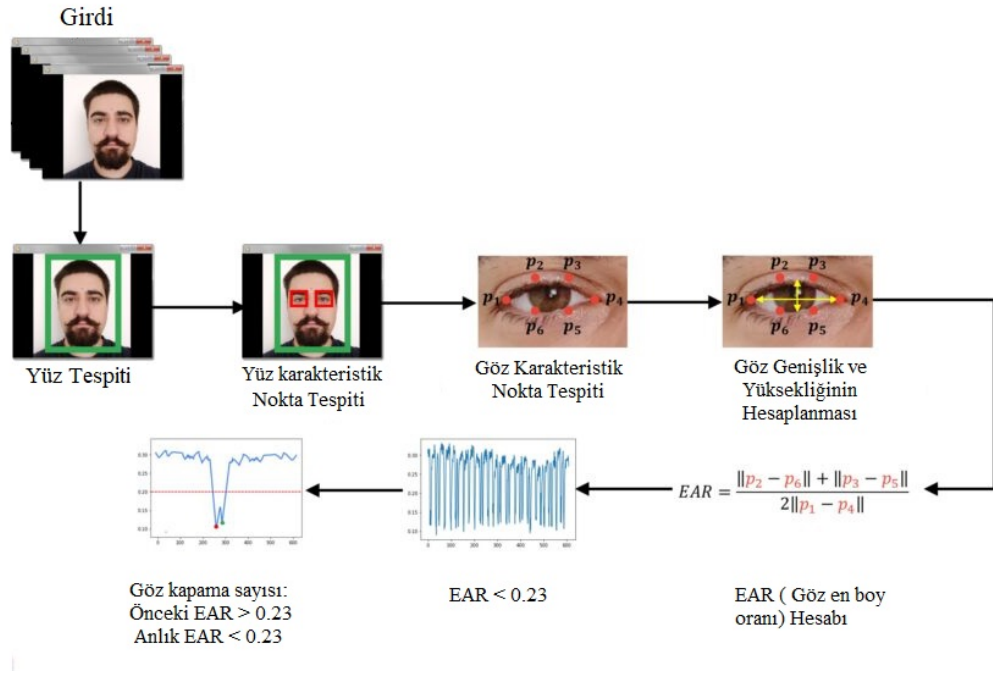


**Şekil 3.20.** Intel realsense D435 3D derinlik kamerası

**Çizelge 3.1.** Intel realsense D435 kamera özellikleri

Özellikler	Açıklama
Kullanım Ortamı	İç ortamlarda veya dış ortamlarda kullanılabilir.
Maksimum Aralık	Yaklaşık 10 metre. Doğruluk, kalibrasyon, ışık vb. koşullara bağlıdır.
Derinlik Teknolojisi	Active IR Stereo
Derinlik Görüş Alanı (Depth Field of View, FOV)	86° × 57° (±3°)
Minimum Derinlik Mesafesi (Min-z)	0,105 m
Derinlik Çözünürlüğü	1280 × 720'e kadar
Derinlik kare hızı	90 fps'e kadar
RGB Sensör Çözünürlüğü	1920 × 1080
RGB Sensör FOV (H × V × D)	69,4° × 42,5° × 77° (± 3°)
RGB kare hızı	30 fps
Kamera Modülü	Intel RealSense Module D430 + RGB Camera
Görüntü İşlemci Kartı	Intel RealSense Vision Processor D4
Uzunluk × Derinlik × Yükseklik	90 mm × 25 mm × 25 mm

**Göz Kırpma ile Canlı – Cansız Yüz Tespiti:** Göz kırpma ile canlılık tespiti için göz hareketleri analiz edilmektedir. Kişinin göz açık kapaması analiz edilerek kişinin canlı bir kişi olduğuna karar verilmektedir. Göz hareketlerini analiz etmek için öncelikle gözlerin karakteristik noktaları (eye landmarks) tespit edilmektedir. Her bir göz, gözün sol köşesinden başlayarak 6 tane  $(x, y)$  koordinatıyla temsil edilmektedir. Aşağıda Şekil 3.21 üzerinde göz karakteristik nokta tespiti ve göz genişlik, yüksekliğinin hesaplanması adımlarında bu noktalar gösterilmiştir.



**Şekil 3.21.** Göz kırpma analizi işlem adımları

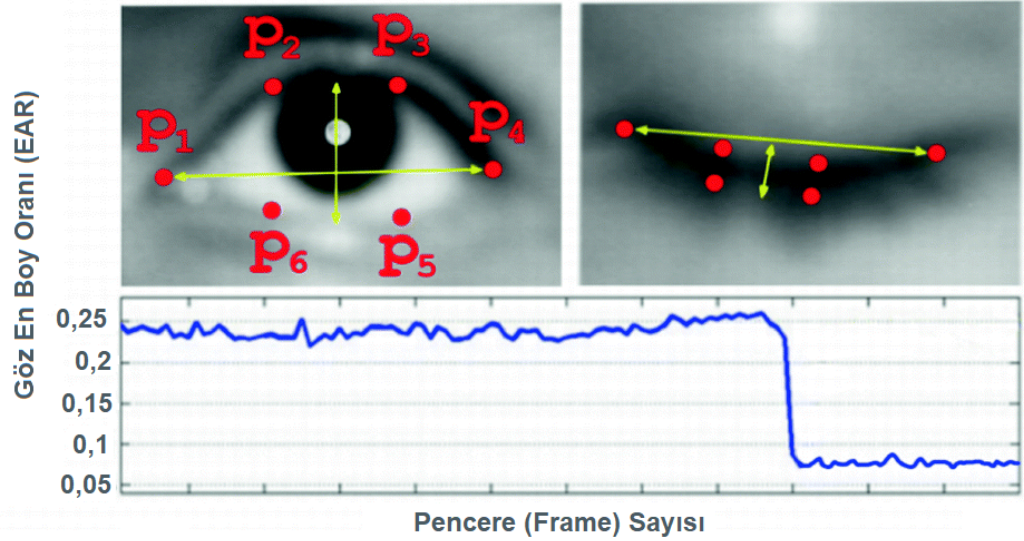
Göz kırpma analizi işlem adımları Şekil 3.21’de gösterilmiştir. Öncelikle kişinin giriş görüntüsü alınarak yüz tespiti yapılmaktadır. Yüz tespiti yapıldıktan sonra derin öğrenme kullanılarak yüzdeki karakteristik noktaları olarak kabul edilen yüz karakteristik noktaları tespit edilmektedir. Bu yüz karakteristik noktaları göz, ağız, burun ve yüz genel hatlarını tanımlamaktadır. Ardından göz karakteristik noktaları tespit edilmekte ve bu noktalar kullanılarak gözün en boy oranı (Eye Aspect Ratio, EAR) hesaplanmaktadır. EAR için eşik değeri olarak belirlenmektedir. Böylece kişinin gözü kapandığında EAR değeri eşik değerinden küçük olacaktır. Böylece göz kırpma tespit edilerek kişinin canlılık tespiti

yapılabilmektedir.

Göz en-boy oranı (EAR) şu şekilde hesaplanmaktadır:

$$\mathbf{EAR} = \frac{\| p_2 - p_6 \| + \| p_3 - p_5 \|}{2 \| p_1 - p_4 \|} \quad (3.10)$$

Burada  $p_1, \dots, p_6$  2 boyutlu göz karakteristik noktalarıdır. Denklemde pay, dikey göz noktaları arasındaki mesafeyi hesaplarken payda ise yatay göz noktaları arasındaki mesafeyi hesaplamaktadır. Burada dikey karakteristik noktaları 2 tane olduğundan payda 2 ile çarpılarak ağırlıklandırılmaktadır. Göz en-boy oranı göz açıkken yaklaşık olarak sabittir ancak göz kırıldığında gözüün en-boy oranı sıfıra düşmektedir. Denklem 3.10 kullanılarak görüntü işleme vb. tekniklerin kullanılmasına gerek kalmadan göz karakteristik noktalarının uzaklık oranına göre kişilerin göz kırıp kırpmadığı anlaşılabilir. Böylece göz kırpmaya algılanarak canlılık analizi yapılabilmekte ve yüz tanıma sisteminin aldatılmasının önüne geçilebilmektedir.



**Şekil 3.22.** Sol üst: Göz açıkken göz karakteristik noktalarının görselleştirilmesi, Sağ üst: Göz kapalıyken gözüün karakteristik noktaları, Alt: Göz en-boy oranındaki düşüş bir göz kırpmayı gösterir.

Şekil 3.22’de T. Soukupova ve J. Cech (2016) tarafından yayınlanan makaleden alınan

görselde gözün kapalı ve açık olması durumunda göz karakteristik noktalarının görselleştirilmesi verilmiştir. Sol üstteki görselde tamamen açık bir göz örneği bulunmaktadır. Buradaki göz en-boy oranı büyük ve yaklaşık olarak sabit bir değere sahiptir. Bununla birlikte kişi göz kırptığında sağ üstteki gibi göz en – boy oranı birden azalmakta ve sıfıra yaklaşmaktadır. Alttaki şekilde bir video görüntüsü için belirli zaman aralığında göz en boy oranının grafiği gösterilmiştir. Grafikten de görüldüğü gibi göz açıkken en boy oranı sabittir ve göz kapalıyken aniden sıfıra düşmektedir. Buradan göz kırpmasının olduğu anlaşılmaktadır. Göz kırpmasının tespit edilmesi vesikalık fotoğraf, yüz görüntü çıktısı, tablet, telefon veya bilgisayar görüntüleri için sahte yani cansız yüzlerin tespiti için başarılı olmaktadır ancak video görüntüleri kullanıldığında göz kırpma analizi canlı-cansız yüz tespiti için başarısız olmaktadır. Bu nedenle üzerinde çalışılan yüz tanıma sisteminin saldırılara karşı daha güvenli hale getirilmesi için öncelikle 3D derinlik kamerası ile derinlik analizi yapılmakta ve eğer derinlik algılanırsa göz kırpma analizi de yapılarak sistemin hemen hemen her saldırı için aldatılmasının önüne geçilmektedir. Derinlik analizi ile normal fotoğraf veya video görüntülerinin sahte olduğu anlaşılabilir ancak yüz fotoğraf çıktısı alınarak derinlik verilebilmektedir. Bu aldatma yöntemini engellemek için de göz kırpma analizi yapılmaktadır.



## 4. BULGULAR

Tez konusu “3 boyutlu derinlik kamerası ile derin öğrenme tabanlı güvenli yüz tanıma” adlı çalışmada ResNet-29 Evrişimli sinir ağı modeli kullanılarak derin öğrenme tabanlı bir yüz tanıma sistemi geliştirilmiştir. Geliştirilen sistemde yüz tanıma için çok başarılı sonuçlar elde edilmesinin yanında 3D derinlik kamerası ve göz kırpması tespiti ile yüz tanıma sistemlerini tehdit eden yöntemlere karşı da başarı sağlanmıştır. Böylece kamera karşısındaki kişinin canlı bir kişi mi yoksa cansız mı olduğu tespit edilerek güvenli yüz tanıma yapılmıştır.

### 4.1. Yüz Tespiti Analiz Sonuçları

Yüz tespiti için 6 farklı yöntem üzerinde çalışılmış ve sonuçları analiz edilmiştir. Bu yöntemler şu şekildedir:

- OpenCV Haar Cascade (Viola Jones)
- OpenCV LBP
- Dlib Convolutional Neural Network (CNN)
- Dlib HOG
- OpenCV Deep Learning
- OpenCV MTCNN

Çizelge 4.1’de yüz tespit modellerinin 475 adet görsel üzerindeki analiz sonuçları verilmiştir. Burada derin öğrenme tabanlı Dlib CNN ve Dlib HOG modellerinin başarımlarından ön plana çıktığı görülmektedir. Modeller GPU üzerinde çalıştırılmıştır. HOG yöntemi kameraya doğrudan bakılan görüntülerde çok başarılı çalışmaktadır ancak yüz açısı kameraya göre değişmeye başladığında tespit başarımları düşmektedir. Ancak Dlib CNN modeli çok farklı açılardan örneğin yüz açısı kameraya göre 90 derece bile olsa tespit yapabilmektedir. GPU ya sahip cihazlar kullanıldığında hem farklı açılardan yüzleri kolaylıkla tespit edebilmesi, hem ışık, gürültü vb. etkilerden etkilenmemesi hem de hız itibarıyla Dlib CNN modeli ön plana çıkmaktadır. Hem Dlib HOG hem de Dlib CNN modelleri yüz olmayan görüntülerde yüz olmadığını başarılı bir şekilde tespit edebilmektedirler. OpenCV LBP modeli GPU üzerinde hız açısından en iyi performansı vermektedir.

**Çizelge 4.1.** Yüz tespit algoritmaları analizi: GPU üzerinde

Yöntem	TP	FP	TN	FN	Ortalama Tespit Süresi (ms)	Başarım (Accuracy)
OpenCV Haar Cascade (Viola Jones)	465	13	92	12	52,5	0,9570
OpenCV LBP	426	32	85	41	17,1	0,8750
Dlib Convolutional Neural Network (CNN)	475	4	97	0	47,2	0,9931
Dlib HOG	475	2	98	0	90,7	0,9965
OpenCV Deep Learning	475	5	97	0	74,4	0,9913
OpenCV MTCNN	471	6	95	4	479,8	0,9826

#### 4.2. Yüz Tanıma Analiz Sonuçları

Yüz tanıma için 4 farklı yöntem üzerinde çalışma yapıldı ve sonuçları analiz edildi. Bu yöntemler şu şekildedir:

- M1:OpenCV Eigenfaces
- M2:OpenCV Fisherfaces
- M3:OpenCV LBPH
- M4:Dlib DNN +K-NN :Resnet-29 Model

Yüz tanıma sonuçlarının analizi için Şekil 4.1'de gösterildiği gibi GeorgiaTech veriseti <sup>3</sup> kullanılmıştır.

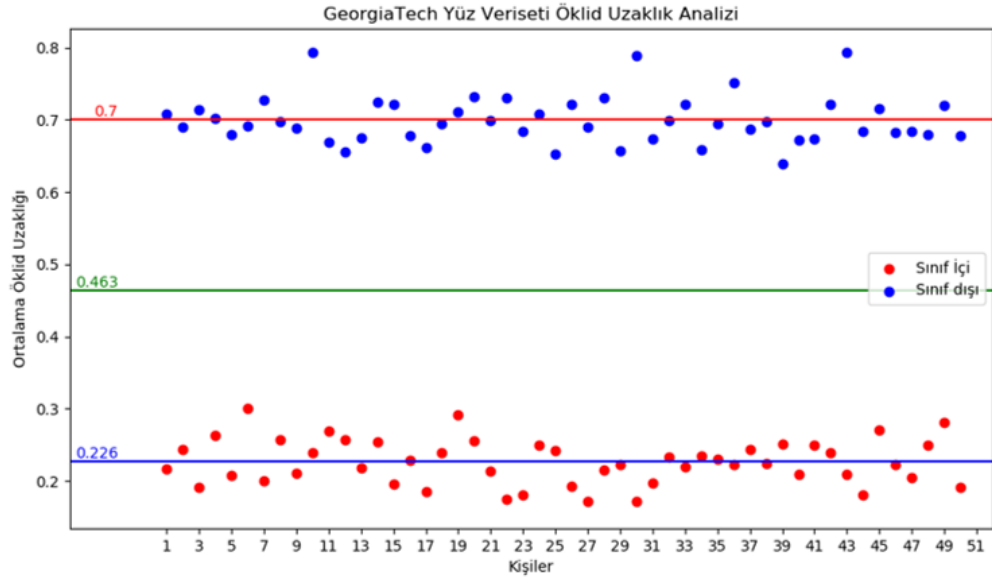
GeorgiaTech Üniversitesinin hazırlanmış olduğu veri setinde model eğitiminde 50 kişiye ait ortalama 14 fotoğraf, test veri setinde 50 kişinin birer fotoğrafı kullanıldı. Görsellere kırpma ve hizalama (aligning) işlemi uygulanarak boyutları  $350 \times 320$  şeklinde eşit olarak ölçeklendirildi.

<sup>3</sup>[http://www.anebian.com/research/face\\_reco.htm](http://www.anebian.com/research/face_reco.htm), 2021



Şekil 4.1. GeorgiaTech yüz tanıma veriseti

#### ResNet-29 Modeli Sınıf İçi Sınıf Dışı Öklid Uzaklık Analizi:



Şekil 4.2. GeorgiaTech yüz tanıma veriseti sınıf içi sınıf dışı öklid uzaklık analizi

Şekil 4.2’de ResNet-29 modelinin sınıf içi ve sınıf dışı Öklid uzaklık analiz grafiği verilmiştir. Burada kırmızı noktalar sınıf içi mavi noktalar ise sınıf dışı uzaklık ortalamalarını

temsil etmektedir. ResNet-29 modeli ile 128 uzunluklu yüz öznitelik vektörleri üretilmektedir. Sınıf içi uzaklık ortalaması ise aynı kişinin farklı yüz fotoğraflarından üretilmiş öznitelik vektörlerinin birbirlerine uzaklıklarının ortalamasını temsil etmektedir. Sınıf dışı uzaklık ortalaması bir kişinin farklı fotoğraflarından elde edilmiş öznitelik vektörlerinin ortalamalarının, diğer kişilerin ortalama öznitelik vektörlerine uzaklığını ifade etmektedir. Şekilde de görüldüğü gibi kırmızı ve mavi noktalar birbirinden net bir şekilde ayrılmıştır. Yani ResNet-29 modelinin farklı kişiler için ürettiği öznitelik vektörleri birbirinden net bir şekilde ayrılmaktadır. Bu da sistemin yüzleri çok başarılı bir şekilde tanıyacağını göstermektedir. Bir diğer önemli nokta ise yüz tanıma sisteminde yüz tanıma yapılıp yapılmayacağına karar vermek için bir eşik değerinin belirlenmesidir. Şekilde üzerinde çalışılan veri kümesi için eşik değeri 0,463 olarak belirlenmiştir. Bu eşik değeri sınıf içi ve sınıf dışı uzaklık ortalamalarının tam orta noktasına denk gelmektedir. Yüz tanıma sisteminde veriye göre sınıf içi sınıf dışı uzaklık analizi yapılarak yüz tanıma yapılacak eşik değeri belirlenebilir.

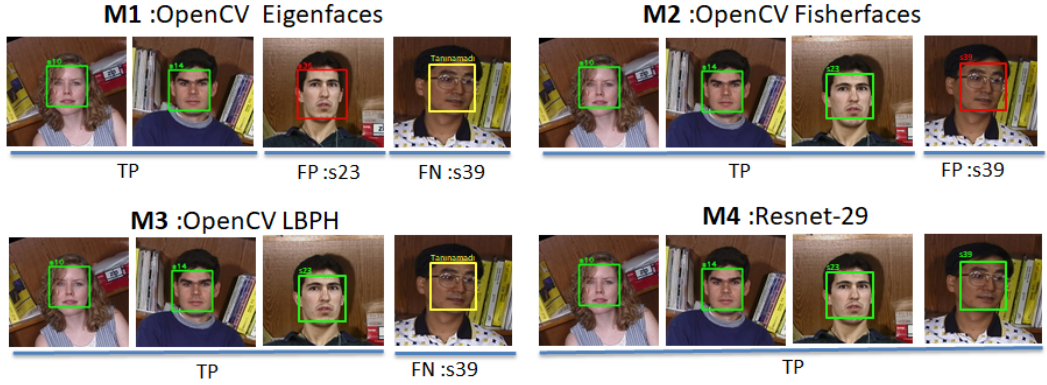
#### **Yüz Tanıma Modelleri Analiz Çizelgesi:**

Çizelge 4.2'de üzerinde çalışılan yüz tanıma modellerinin analizi gösterilmiştir. Yüz tanıma modelleri iki farklı yüz tespiti için gösterilmiştir. Yüz tespiti için yapılan analizler sonucunda Dlib HOG ve Dlib CNN modellerinin çok başarılı olduğu görülmüştür. CPU ve GPU çalışma ortamları için performanslar göz önüne alınarak iki model üzerinde çalışılmıştır. Çizelgeden de görüleceği gibi başarımlar açısından en başarılı model ESA tabanlı ResNet-29 modelidir. Model hem başarımlar hem de hız açısından GPU üzerinde çok başarılı performans sergilemektedir. Sınıflandırma parametresi k 3 olarak alınmıştır. 50 kişi için yapılan yüz tanıma analizinde ResNet-29 modeli %100 doğru tanıma yapmaktadır. Yüz tanıma yapmadan önce görüntüler yüz tespiti ile kırılarak ardından hizalama işlemi uygulanmıştır. Böylece daha başarılı sonuçlar elde edilmiştir.

**Çizelge 4.2.** Yüz tanıma algoritmaları analizi: GPU üzerinde

Yöntem	Yüz Tespit Yöntemi	Eşik Değeri	TP	FP	TN	FN	Ortalama Test Süresi (ms)	Başarım
M1:OpenCV Eigenfaces	Dlib CNN	8600	39	9	10	2	1082,1	0,8167
	Dlib HOG	8000	40	2	9	8	1517,2	0,8305
M2:OpenCV Fisherfaces	Dlib CNN	1350	24	16	9	11	197,1	0,5500
	Dlib HOG	1350	40	3	9	7	593,8	0,8305
M3:OpenCV LBPH	Dlib CNN	60	41	7	10	2	295,2	0,8500
	Dlib HOG	60	46	3	10	1	781,7	0,9333
M4:ResNet-29	Dlib CNN	<b>0,463</b>	<b>50</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>100,2</b>	<b>1,0000</b>
	Dlib HOG	<b>0,463</b>	<b>50</b>	<b>0</b>	<b>10</b>	<b>0</b>	<b>203,0</b>	<b>1,0000</b>

### Yüz Tanıma Modelleri Yüz Tanıma Örnekleri



**Şekil 4.3.** GeorgiaTech yüz tanıma örnekleri

### 4.3. Canlı - Cansız Yüz Tespiti Analiz Sonuçları

Canlı ve cansız kişi tespiti için 3D derinlik kamerasından elde edilen derinlik bilgisi analiz edilmiştir.. Yüz görüntüsünün  $x$ ,  $y$  ve  $f_z$  ile temsil edilen özellik matrisi çıkarılmıştır.  $x$ ,  $x$

yönündeki piksel değerlerini,  $y$ ,  $y$  yönündeki piksel değerlerini  $f_z$  ise  $x$  ve  $y$  koordinatlarına karşılık gelen derinlik bilgisini temsil etmektedir. Derinlik ise yüz bölgesinde bir noktanın kameraya uzaklığını temsil etmektedir. Birimi metre (m) ile temsil edilmektedir. Yüz bölgesinin derinlik bilgisini de içerek 3 boyutlu özellik matrisi çıkarıldıktan sonra derinlik bilgisinin  $x$  ve  $y$  noktalarındaki değişimi analiz edilmiştir. Bunun için  $f_z$  derinlik bilgisinin her nokta için  $x$  ve  $y$  koordinatlarına göre türevi hesaplanmıştır. Derinlik bilgisinin türev hesaplandıktan sonra ise her bir nokta için genlik ve açı hesaplanmıştır. Genlik ve Açı histogramları çıkarılmıştır. 2 boyutlu görüntülerde derinlik az olduğu için veya düz bir yüzey olduğundan dolayı derinlik olmadığı için genlik histogram değeri sıfır ve etrafında bir işaret olarak elde edilmiştir. 3D görüntülerde derinlik bilgisi genlik ve açı histogramları için çok sayıda işaret elde dilmektedir. 3 boyutlu görüntüler bu analizler göz önüne alındığında 2D görüntülere göre çok rahat bir şekilde ayırt edilebilmektedir.



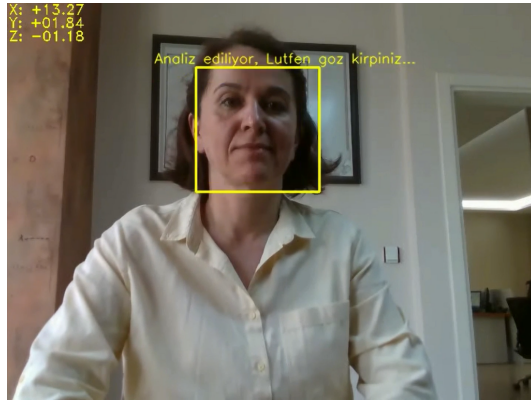
(a) Analiz ediliyor

(b) 3D: kişi onaylandı

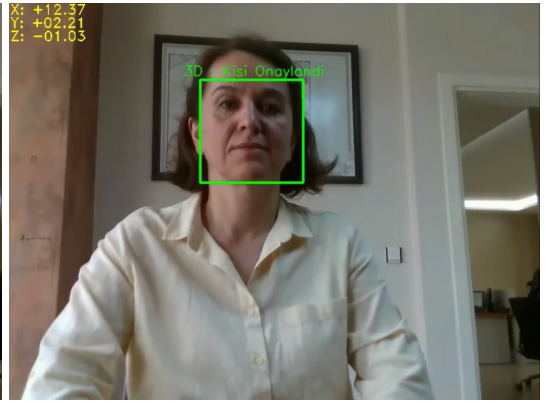
**Şekil 4.4.** Canlı yüz tespiti

Şekil 4.4'te canlı bir kişinin geliştirilen canlı - cansız kişi tespiti sistemi ile analiz sonucu gösterilmiştir. Kamera karşısındaki kişi canlı bir insan olduğu için derinlik bilgisi analiz edildiğinde hesaplanan ortanca değeri 0,3'ten büyük çıkmaktadır. Derinlik onaylandıktan sonra kişinin göz kırpması yapılmıştır. Göz kırpması da tespit edildikten sonra sistem kişinin canlı bir kişi olduğuna karar vererek kişiyi onaylamıştır.

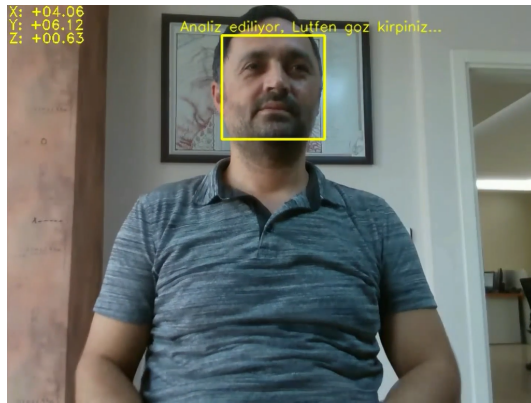




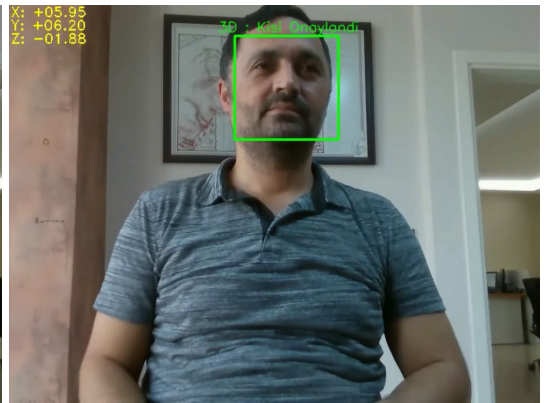
(a) Analiz ediliyor



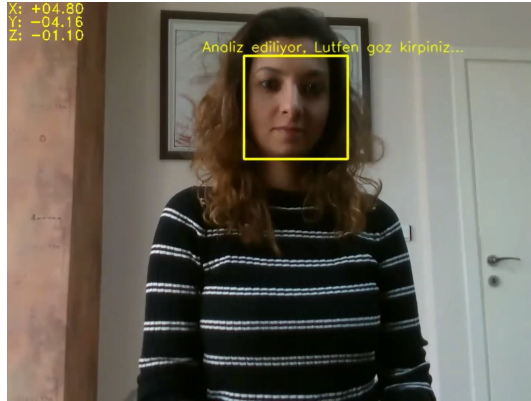
(b) 3D: kişi onaylandı



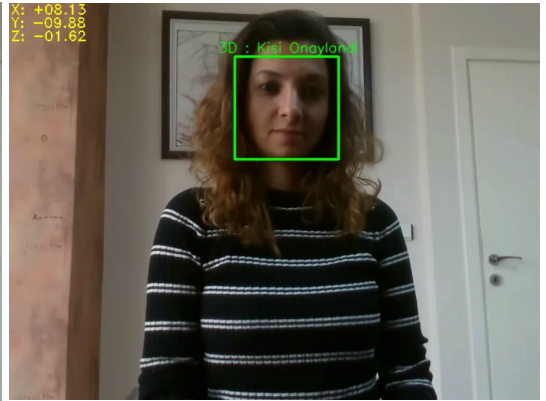
(c) Analiz ediliyor



(d) 3D: kişi onaylandı



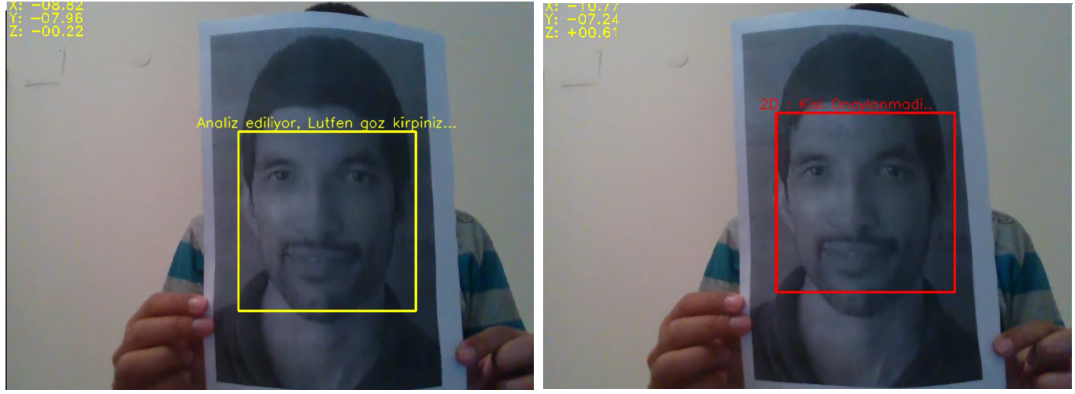
(e) Analiz ediliyor



(f) 3D: kişi onaylandı

**Şekil 4.5.** Canlı yüz tespiti

4.5'te ise 3 farklı ve canlı kişi için, geliştirilen canlı - cansız kişi tespiti sistemi ile analiz sonuçları gösterilmiştir. Sistem 10 farklı canlı kişi üzerinde yapılan analiz sonucu 10 kişi için de doğru karar vermiştir.

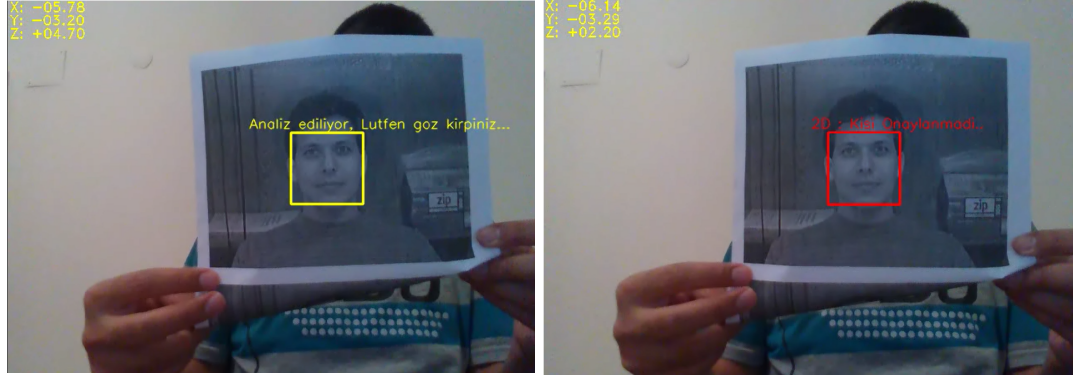


(a) Analiz ediliyor

(b) 2D: kişi onaylanmadı

**Şekil 4.6.** Sahte yüz tespiti: yüz çıktısı (A4 Boyutu)

Şekil 4.6'da yazıcıdan a4 boyutlarında çıktısı alınan bir kişinin canlı - cansız kişi tespiti sistemi ile analiz sonucu gösterilmiştir. Burada görüntü 2 boyutlu olduğu için hesaplanan ortanca değeri 0,3'ten çok küçük olarak hesaplanmıştır. Göz kırpma tespiti de yapılamadığı için sistem kişinin cansız bir kişi olduğuna karar vererek kişiye onay vermemiştir.



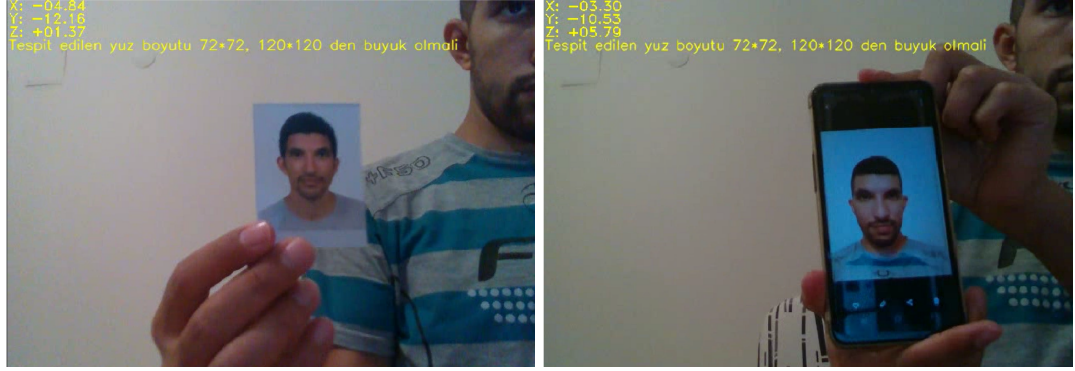
(a) Analiz ediliyor

(b) 2D: kişi onaylanmadı

**Şekil 4.7.** Sahte yüz tespiti: yüz çıktısı (A5 Boyutu)

Şekil 4.7'de yazıcıdan a5 boyutlarında çıktısı alınan bir kişinin canlı - cansız kişi tespiti sistemi ile analiz sonucu gösterilmiştir. Burada görüntü 2 boyutlu olduğu için hesaplanan ortanca değeri 0,3'ten çok küçük olarak hesaplanmıştır. Göz kırpma tespiti de yapılamadığı için sistem kişinin cansız bir kişi olduğuna karar vererek kişiye onay vermemiştir.



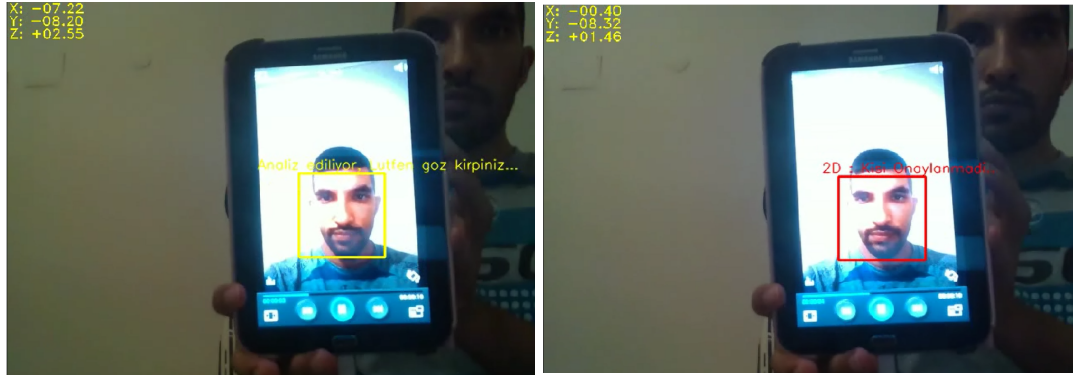


(a) Tespit edilen yüz boyutu  $72 \times 72$ , yüz boyutu  $120 \times 120$ 'den büyük olmalı: işlem yapılamıyor

(b) Tespit edilen yüz boyutu  $72 \times 72$ , yüz boyutu  $120 \times 120$ 'den büyük olmalı: işlem yapılamıyor

**Şekil 4.8.** Sahte yüz tespiti: vesikalik fotoğraf

Şekil 4.8'de solda vesikalik fotoğraf sağda ise cep telefonu görüntüsü olan bir kişinin geliştirilen canlı - cansız kişi tespiti sistemi ile analiz sonucu gösterilmiştir. Burada görüntüler belirlenen yüz boyutlarından ( $120 \times 120$ ) küçük olduğu için sistem herhangi bir işlem yapmamaktadır ve kişiyi tespit edilen yüz boyutunun küçük olduğu ile ilgili uyarılmaktadır.



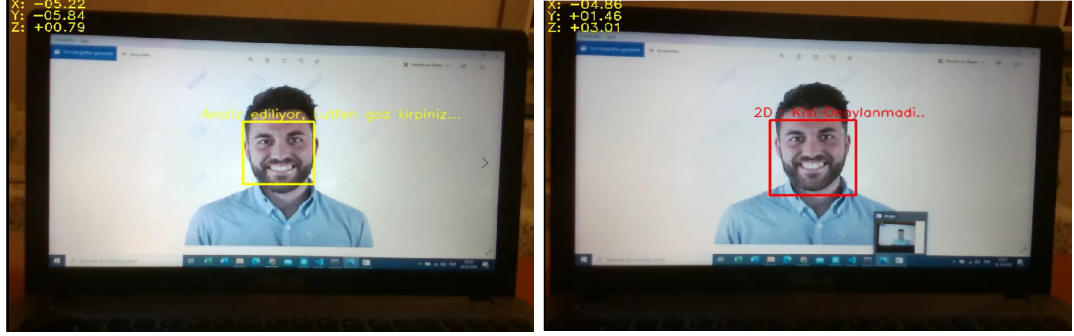
(a) Analiz ediliyor

(b) 2D: kişi onaylanmadı

**Şekil 4.9.** Sahte yüz tespiti: tablet görüntüsü

Şekil 4.9'da tablet video görüntüsü olan bir kişinin canlı - cansız kişi tespiti sistemi ile analizi sonucu gösterilmiştir. Görüntü 2 boyutlu olduğu için hesaplanan ortanca değeri 0,3'ten çok küçük olarak hesaplanmıştır. Burada dikkat çekilecek bir diğer nokta ise videolarda kişilerin göz kırpmasının mümkün olmasıdır. Ancak geliştirilen sistem hem

derinlik hem de göz kırpma tespiti analizi yaptığı için kişinin video görüntüsü bile olsa cansız olduğunu başarılı bir şekilde tespit etmektedir. Derinlik algılamadığı için sistem kişinin cansız bir kişi olduğuna karar vererek kişiye onay vermemiştir.



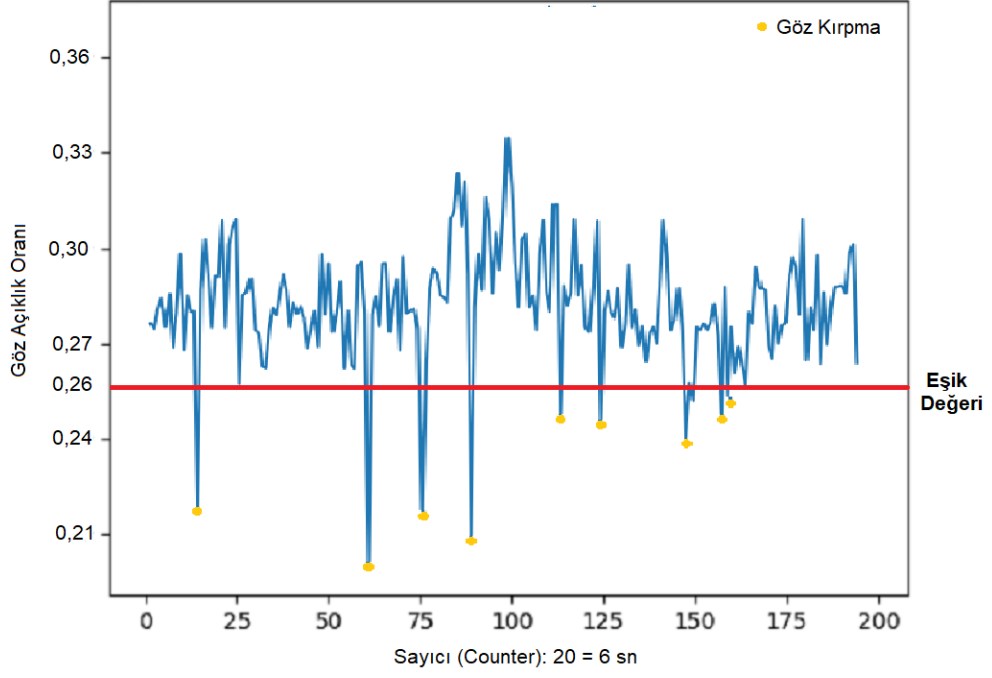
(a) Analiz ediliyor

(b) 2D: kişi onaylanmadı

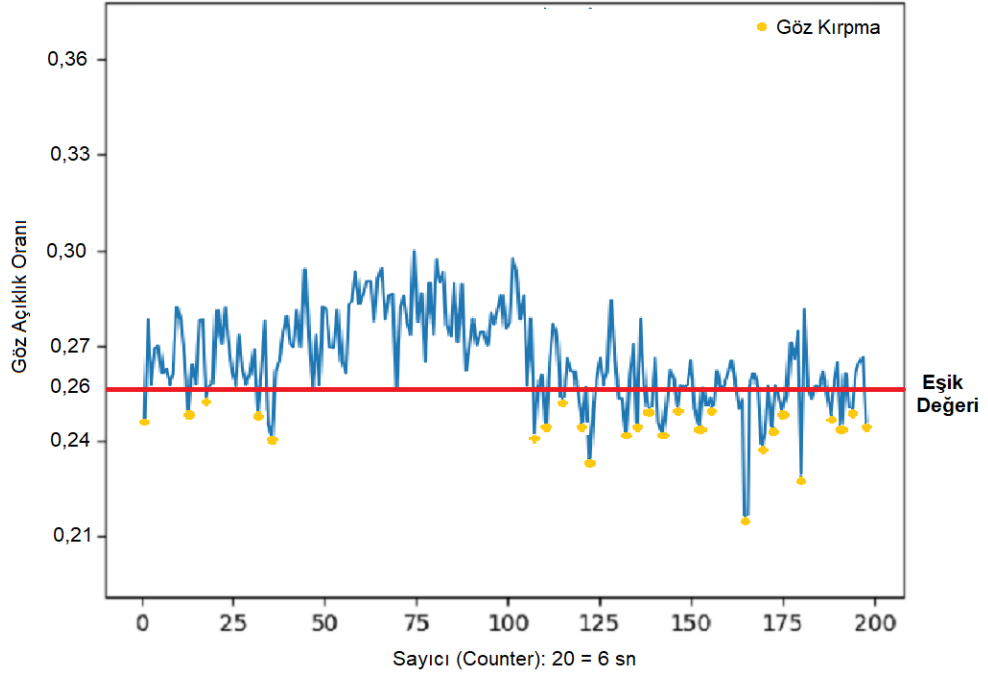
**Şekil 4.10.** Sahte yüz tespiti: bilgisayar görüntüsü

Şekil 4.10'da bilgisayar görüntüsü olan bir kişinin canlı - cansız kişi tespiti sistemi ile analiz sonucu gösterilmiştir. Burada da görüntü 2 boyutlu olduğu için hesaplanan ortanca değeri 0,3'ten çok küçük olarak hesaplanmıştır. Göz kırpma tespiti de yapılamadığı için sistem kişinin cansız bir kişi olduğuna karar vererek kişiye onay vermemiştir.

**Göz Kırpma Tespiti Analiz Sonuçları:** Göz kırpma için göz tespit edildikten sonra canlı kişiler ve fotoğraflar için göz en - boy oranı analizi yapılmıştır. Şekil 4.11'de 2 farklı canlı kişi örneği için 1 dakika boyunca analiz edilen gözlerin en-boy oranı analiz grafikleri verilmiştir. Burada 4.11a ve 4.11b incelendiğinde kamera karşısındaki kişiler canlı kişiler olduğu için göz kırpma anında EAR değeri düşmektedir. Canlı kişiler için eşik değeri 0,26 gibi kabul edilirse, EAR her 0,26 altına indiğinde göz kırpma kabul edilmektedir. Şekil 4.12'de ise farklı boyutlarda cansız iki örnek için göz en - boy oranı analiz grafikleri verilmiştir. 4.12a ve 4.12b incelendiğinde ise 1 dakikalık süre boyunca EAR değerinin 0,26'nın üstünde olduğu görülmektedir. Sistem hem derinlik hem göz kırpma analizi yaptığı için 4 örneğin de başarılı bir şekilde canlı veya cansız olduğuna karar vermiştir.

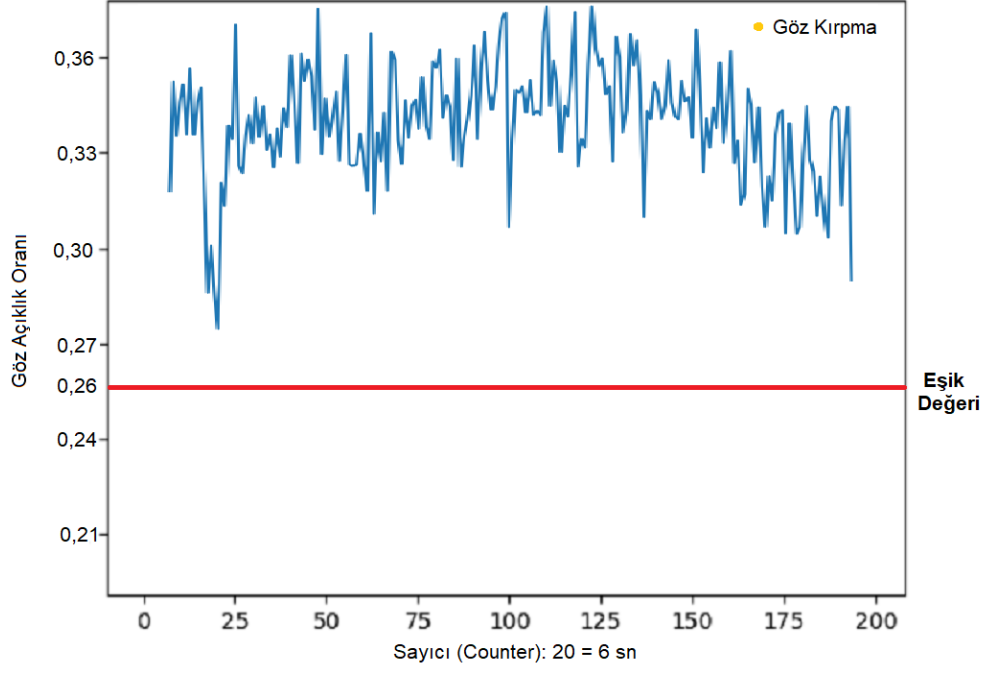


(a) Canlı kişi 1: göz en boy oranı (EAR)

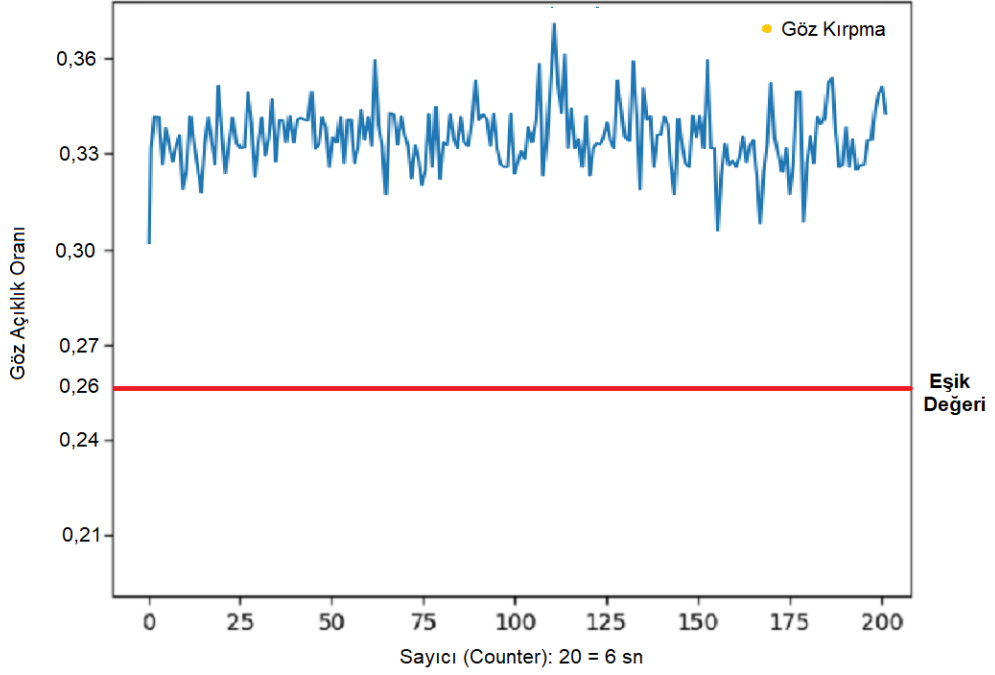


(b) Canlı kişi 2: göz en boy oranı (EAR)

**Şekil 4.11.** Göz en boy oranı grafikleri



(a) Cansız kişi 1 (Yüz Boyutu Büyük): göz en boy oranı (EAR)



(b) Cansız kişi 2 (Yüz Boyutu Küçük): göz en boy oranı (EAR)

**Şekil 4.12.** Göz en boy oranı grafikleri

Göz kırpma tespiti için canlı ve cansız kişiler için 1 dakikalık süreler boyunca göz kırpma analizi yapılmıştır. Burada EAR değeri Şekil 4.11 ve Şekil 4.12 referans alınarak 0,26 olarak seçilmiştir. Analizde 4 farklı canlı kişi için 1 dakika içinde 9 ile 16 arasında göz kırpma tespit edilmiştir. İlk göz kırpma yaklaşık olarak 5,82. saniyede algılanmıştır. Bir fotoğraf analiz edildiğinde ise EAR değeri 0,26 seçildiğinde göz kırpma algılanmamıştır. Sistem burada kullanılan 4 farklı kişinin canlı olduğuna ve fotoğrafın ise cansız olduğuna karar vermiştir.

**Çizelge 4.3.** 1 dakika süre boyunca göz kırpma sayıları ve ilk göz kırpma için geçen süre

<b>Kişi</b>	<b>Analiz Süresi (sn)</b>	<b>Göz Kırpma Tespit Sayısı</b>	<b>İlk Tespit İçin Geçen Süre (sn)</b>
Kişi 1	60 sn	14	6,00 sn
Kişi 2	60 sn	16	6,38 sn
Kişi 3	60 sn	9	6,38 sn
Kişi 4	60 sn	12	5,82 sn
Fotoğraf	60 sn	0	-

## 5. TARTIŞMA VE SONUÇ

Bu çalışmada son yıllarda güvenlik başta olmak üzere pek çok alanda yaygın bir şekilde kullanılabilen derin öğrenme tabanlı bir yüz tanıma sistemi geliştirilmiştir. Ayrıca yüz tanıma sistemlerini aldatmak için kullanılan yanıltıcı tekniklere karşı 3D derinlik bilgisi ve göz kırpması tespiti kullanarak güvenliği sağlayan bir yöntem üzerinde çalışılmıştır. Bilgisayar donanımındaki ve derin öğrenme tekniklerindeki gelişmeler ile birlikte pek çok alanda önemli çalışmalar yapılmıştır. Yüz tanıma da bu alanlardan biridir. Yüz tanımanın ilk aşamasını yüz tespiti oluşturmaktadır. Yüz tespiti için 2 ayrı model kullanılmıştır. Modellerden biri görüntülerden birtakım histogramların elde edildiği ve bu histogramların analiz edilmesiyle görüntüdeki yüzleri bulmaya yarayan HOG yöntemidir. HOG yüz tespit yöntemi özellikle kameraya düz açıyla yani doğrudan bakan kişilerin yüzlerini tespit etmekte çok başarılı olmaktadır. Bu yöntem hem CPU hem de GPU'ya sahip bilgisayarlarda hızlı bir şekilde yüz tespiti yapmaktadır. En önemli dezavantajı ise kişinin kameraya bakış açısı değiştiğinde, yüz tespit başarımı düşmektedir. Üzerinde çalışılan bir diğer yüz tespit yöntemi ise Dlib CNN olarak adlandırılan derin öğrenme veya ESA tabanlı yüz tespit yöntemidir. Derin öğrenme tabanlı bir model olduğu ve çok fazla işlem yapıldığı için CPU üzerinde çok yavaş çalışmaktadır. Ancak bu sorun GPU'ya sahip bilgisayarlar ile rahatlıkla çözülebilmektedir. Bu yöntem farklı açılardan yüzleri tespit etmekte çok başarılıdır. Kişi kameraya tam olarak yan baksa bile yüz tespiti yapabilmektedir. Bir diğer önemli avantajı ise küçük yüzleri tespit etmekte başarılı olmasıdır. Yüz tespit adımından sonra ise yüz tanımadaki daha başarılı sonuçlar elde etmek için görüntülerden yüz bölgeleri seçilmiştir. Seçilen yüz görüntüsüne hizalama işlemi uygulanmıştır. Bu ön işleme adımlarından sonra yüz tanıma yapılmıştır. Üzerinde çalışılan model ESA tabanlı 29 evrişim katmanına sahip ResNet-29 modelidir. ResNet-29 modeli artık katman olarak bilinen katmanlara sahiptir. Normal evrişim katmanına sahip ESA ağlarında ağ derinliği arttıkça parametre sayısı ve karmaşıklık artmaktadır. Bu durum modeller eğitilirken giriş görüntüsünün özellikleri ağ derinleştikçe çıkışa doğru kaybolduğundan aşırı uyum gösterme sorununa yol açmaktadır. Aşırı uyum problemiyle karşılaşılması nedeniyle daha derin ağlar geliştirilememiştir.

ResNet modelinde ise giriş bir sonraki katmanın çıkışına kısayol bağlantılarıyla bağlanmaktadır. Bu kısayollar ile girişin etkileri sonraki katmanlara iletilebilmekte böylece özelliklerin kaybolmasının önüne geçilmektedir. Bunun sonucunda ise çok fazla katmana sahip örneğin 1000 katmanlı modeller geliştirilebilmektedir. ResNet yapılarının bu özelliği yüz tanıma için de çok başarılı sonuçların elde edilmesini sağlamıştır. ResNet-29 modeli ile %99,38'e varan bir başarımla elde edilmiştir. ResNet-29 modeli ile kişilere ait eşsiz, 128 uzunluklu öznitelik vektörleri çıkarılmaktadır. Elde edilen bu vektörler her kişi için birbirinden farklıdır. Böylece sınıflandırma teknikleri ile sınıflandırılarak yüz tanıma yapılabilmektedir. ResNet-29 modeli yaklaşık 3 milyon görsel ile eğitilmiştir. Model eğitilirken 3'lü kayıp denilen kayıp türü kullanılmıştır. 3'lü kayıp hesaplanırken bir kişinin referans görseli, referans ile aynı kişiye ait bir pozitif örnek ve referans görselden farklı bir kişiye ait bir negatif örnek alınmaktadır. Bu kayıp referans görsel için üretilen 128 uzunluklu öznitelik vektörleri ile pozitif örneğe ait öznitelik vektörlerinin birbirine uzaklığı ile referans görselden elde edilen öznitelik vektörleri ile negatif örnekten elde edilen öznitelik vektörlerinin birbirine uzaklığını hesaplamaktadır. Daha sonra referans örnek ile pozitif örneğin öznitelik vektörleri birbirine yaklaştırılmakta, referans ile negatif örneğin öznitelik vektörleri ise birbirinden uzaklaştırılmaktadır. Böylece model kişileri birbirinden ayırt eden özellikleri öğrenmekte ve kişileri ayırt edebilmektedir. Bir kişinin farklı fotoğrafları için üretilen vektörler birbirine yakın, farklı kişilerin fotoğrafları için ise birbirine uzak vektörler üretilmektedir. ResNet-29 modeli ile üretilen vektörler ile sınıf içi sınıf dışı öklid uzaklık analizi ile analiz edilmiştir. Yapılan analizde bir kişinin farklı fotoğrafları için üretilen öznitelik vektörlerinin ortalamasının aynı kişinin tüm vektör değerlerine uzaklıkları hesaplanarak bunların ortalamaları alınmıştır, bu değer sınıf içi uzaklık ortalamasını temsil etmektedir. Yine bir kişinin farklı fotoğraflarından elde edilen öznitelik vektörlerinin ortalaması hesaplanıp bu ortalama vektörünün diğer kişilerin de öznitelik vektörlerinin ortalamalarına uzaklığı hesaplanmıştır. Hesaplanan değerlerin ortalaması alınmıştır. Hesaplanan bu değer ise sınıf dışı uzaklık ortalamasını temsil etmektedir. Sınıf içi ve sınıf dışı uzaklık analizinde elde edilen değerlerin birbirinden net bir şekilde ayrıştığı görülmüştür. Bu değerlerin birbirinden ayrışması modelin ürettiği vektörlerin her kişi için eşsiz olduğunu göstermektedir. Sınıf içi uzaklıkların ortalaması

ile sınıf dışı uzaklıkların ortalaması hesaplanarak bu iki değerin tam orta noktası eşik değeri kabul edilmiştir. Böylece yeni bir kişi için elde edilen vektörün önceden eğitilmiş ve veritabanına kayıt edilmiş öznitelik vektörlerine uzaklığı hesaplanıp en yakın kişi ile eşleştirilmektedir. Eğer hesaplanan uzaklık değeri eşik değerinin üstündeyse yüz tanıma gerçekleşmemektedir. Ancak bu değer eşik değerinin altındaysa yüz tanıma yapılmaktadır. ResNet-29 modeli ile üretilen öznitelik vektörleri  $k$  en yakın komşuluk yani K-NN sınıflandırıcı ile sınıflandırılarak yüz tanıma yapılmaktadır. K-NN sınıflandırıcı uzaklık analizi ve oylama mantığıyla çalışmaktadır. Her kişi için üretilen öznitelik vektörleri her kişi için veritabanına kaydedilmektedir. Daha sonra yüz tanıma yapılmak istendiğinde seçilen  $k$  değeri baz alınarak işlem yapılır. Tanınacak kişinin öznitelik vektörü çıkartılarak, daha sonra bu vektörün veritabanındaki kişi vektörleriyle uzaklıkları hesaplanmaktadır,  $k = 3$  alındığı için geriye en yakın 3 uzaklık değeri ve bu değerlerin ait olduğu kişiler döndürülmektedir. Burada oylama işlemi devreye girmektedir. 3 uzaklık içinden 2 örnek A kişisine, 1 örnek ise B kişisine ait olsun. Bu durumda tanınacak kişi A kişisiyle eşleştirilmektedir. Eğer A kişisinin uzaklık değerlerinin en küçüğü eşik değerini geçerse yüz tanıma yapılmaktadır. Eşik değerinden küçük olması durumunda ise kişi tanınmadı diye sonuç döndürülmektedir. Yüz tanıma işlemi tamamlanmadan önce derinlik ve göz kırpması analizi yapılmaktadır. Çünkü yüz tanıma sistemleri fotoğraf, telefon - tablet görüntüsü, bilgisayar görüntüsü, maske vb. yöntemlerle yanıltılabilmektedir. Bunun önüne geçmek için 3D derinlik kamerası kullanılarak öncelikle kamera karşısındaki kişinin derinlik bilgisi analiz edilmektedir. Derinlik analizinde  $x$ ,  $y$  ve  $z$  bilgileri kullanılmıştır. Kamera karşısındaki kişi için yüz tespiti yapılarak sadece yüz bölgesi için derinlik analizi yapılmıştır. Burada derinliğe sahip bir yüz için derinliğin  $x$  ve  $y$  yönündeki değişimlerinin olması beklenmektedir. Düz bir yüzey için ise derinlik  $x$  ve  $y$  noktalarında değişmeyeceğinden gradyeninin 0 veya 0'a çok yakın olması beklenmektedir. Bu düşünceden yola çıkılarak 3D kameradan alınan yüz görüntüsünün derinlik bilgisinin  $x$  ve  $y$  noktalarına göre türevleri hesaplanmıştır. Hesaplanan bu değerlerin açı ve genlik histogramları çıkarılmıştır. Elde edilen histogramlar canlı bir yüz için farklı noktalarda değerlere sahip iken yani dalgalı bir sonuca sahip iken, cansız yüzler için beklendiği gibi 0 veya çevresinde toplanmıştır. Bu analiz doğrultusunda genlik ve açı histogramlarının ortalama, ortanca ve standart sapma



gibi deęerleri hesaplanmıřtır. Canlı yzler ve cansız yzler iin genlik histogramlarının birbirinden net bir Őekilde ayrıldıęı grlmřtr. Bu ayırımı baz alınarak bir eřik deęeri belirlenmiřtir. Eęer derinlik bilgisi deęiřiminin histogramının ortanca deęeri eřik deęerinin stnde ise kiři canlı, altında ise kiři cansız Őeklinde karar verilmiřtir. Uygulanan canlılık tespit yntemi vesikalık fotoęraflarda, cep telefonu/tablet, bilgisayar grntlerinde, yzlerin yazıcıdan alınan ıktılarında ok bařarılı sonu vermiřtir. Ancak kaęıt ıktı biraz bkldęnde derinlik algılandıęından burada ek bir zme ihtiya duyulmuřtur. Bu probemin zmnde ise gz kırpma analizi uygulanmıřtır. Eęer derinlik bilgisi analizi sonucu eřik deęerinin zerinde bir sonu elde edilirse ek olarak kiřiden gz kırpması istenmiřtir. Gz kırpma da algılanırsa sistem kiřinin canlı olduęuna kesin karar verip yz tanıma sonularını dndrmekte ve kiřiyi onaylamaktadır, gz kırpma algılanmazsa sistem kiřinin canlı olduęunu onaylamamaktadır. Bylece hem yz tanıma hem de yz tanıma sisteminde gvenlięin saęlanması iin bařarılı bir model geliřtirilmiřtir. Tez konusu 3D derinlik kamerası ile daha gvenli yz tanıma yapılarak tm gereklilikler yerine getirilmiřtir. Elbette yz tanıma, canlılık tespiti ve yz tanımda gvenlik gibi konular gnmzde ok poplerdir ve uygulama alanları da gittike yaygınlařmaktadır. Son yıllarda ok nemli geliřmeler yařansa da bu teknoloji hala geliřmeye devam etmektedir. Tez kapsamında yapılan alıřmalarda bařarılı sonular elde edilse de yz tanımda zellikle gvenlik konusunda farklı yaklařımlar da uygulanabilir. rneęin derinlik bilgisi derin ęrenme teknikleri ile yapay zekaya ęretilerek insandaki sisteme benzer bir sistem geliřtirilebilir. Yz tanımda % 99 oranlarına varan bir bařarım elde edilse de bařarımı daha da arttırmak iin farklı derin ęrenme modelleri zerinde alıřılabilir. Bylece yz tanıma ve daha gvenli yz tanıma iin daha bařarılı sonuların elde edilebileceęi ngrlmektedir.

## KAYNAKLAR

- Ahonen, T., Hadid, A., Pietikainen, M. 2006.** Face Description with Local Binary Patterns: Application to Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.12, 2037–2041.
- Arfi, A. M., Bal, D., Hasan, M. A., Islam, N., Arafat, Y. 2020.** Real Time Human Face Detection and Recognition Based on Haar Features, 2020 IEEE Region 10 Symposium (TENSYP), 517–521.
- Baochang Zhang, Yongsheng Gao, Sanqiang Zhao, Jianzhuang Liu 2010.** Local Derivative Pattern Versus Local Binary Pattern: Face Recognition With High-Order Local Pattern Descriptor, *IEEE Transactions on Image Processing* 19.2, 533–544.
- Bartlett, M. S., Movellan, J. R., Sejnowski, T. J. 2002.** Face recognition by independent component analysis, *IEEE Transactions on Neural Networks* 13.6, 1450–1464.
- Belhumeur, P. N., Hespanha, J. P., Kriegman, D. J. 1997.** Eigenfaces vs. Fisherfaces: recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19.7, 711–720.
- Chengjun Liu, Wechsler, H. 2002.** Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition, *IEEE Transactions on Image Processing* 11.4, 467–476.
- Dadi, H., Mohan, P. Apr. 2016.** Improved Face Recognition Rate Using HOG Features and SVM Classifier, *IOSR Journal of Electronics and Communication Engineering(IOSR-JECE)* 11, 34–44.
- Dalal, N., Triggs, B. 2005.** Histograms of oriented gradients for human detection, 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR' 05). Vol. 1, 886–893 vol. 1.
- Database, G. T. F. 2021.** URL: [http://www.anefian.com/face\\_reco.htm](http://www.anefian.com/face_reco.htm).
- Dong, J., Tian, C., Xu, Y. 2017.** Face liveness detection using color gradient features, 2017 International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), 377–382.

- Gunasekar, S., Ghosh, J., Bovik, A. C. 2014.** Face Detection on Distorted Images Augmented by Perceptual Quality-Aware Features, *IEEE Transactions on Information Forensics and Security* 9.12, 2119–2131.
- He, K., Zhang, X., Ren, S., Sun, J. 2015.** Deep Residual Learning for Image Recognition. arXiv: 1512.03385.
- Hyung-Ji Lee, Wan-Su Lee, Jae-Ho Chung 2001.** Face recognition using Fisherface algorithm and elastic graph matching, *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*. Vol. 1. IEEE, 998–1001.
- Juwei Lu, Plataniotis, K. N., Venetsanopoulos, A. N. 2003.** Face recognition using kernel direct discriminant analysis algorithms, *IEEE Transactions on Neural Networks* 14.1, 117–126.
- Kazemi, V., Sullivan, J. 2014.** One millisecond face alignment with an ensemble of regression trees, *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1867–1874.
- King, D. 2021.** Dlib Face Recognition Models, URL: <https://github.com/davisking/dlib-models>.
- King, D. E. 2015.** Max-Margin Object Detection, *CoRR* abs/1502.00046. arXiv: 1502.00046.
- Lagorio, A., Tistarelli, M., Cadoni, M., Fookes, C., Sridharan, S. 2013.** Liveness detection based on 3D face shape analysis, *2013 International Workshop on Biometrics and Forensics (IWBF)*, 1–4.
- Lei, Z., Pietikäinen, M., Li, S. Z. 2014.** Learning Discriminant Face Descriptor, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.2, 289–302.
- Li, J., Zhang, X., Zhang, Y., Wang, H., Yang, F. 2019.** Face Liveness Detection Based On Multiple Feature Descriptors, *2019 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 1–5.
- Low, C. 2015.** Learning compact discriminant local face descriptor with VLAD, *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 825–833.

**Mo, H., Liu, L., Zhu, W., Li, Q., Liu, H., Yin, S., Wei, S. 2020.** A Multi-Task Hardwired Accelerator for Face Detection and Alignment, *IEEE Transactions on Circuits and Systems for Video Technology* 30.11, 4284–4298.

**Naseem, I., Togneri, R., Bennamoun, M. 2010.** Linear Regression for Face Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 2106–2112.

**Padilla, R., Filho, C., Costa, M. Apr. 2012.** Evaluation of Haar Cascade Classifiers for Face Detection,

**Perlibakas, V. 2004.** Distance measures for PCA-based face recognition, *Pattern Recognition Letters* 25.6, 711 –724. ISSN: 0167-8655.

**Ranjan, R., Bansal, A., Zheng, J., Xu, H., Gleason, J., Lu, B., Nanduri, A., Chen, J., Castillo, C. D., Chellappa, R. 2018.** A Fast and Accurate System for Face Detection, Identification, and Verification, *CoRR abs/1809.07586*. arXiv: 1809.07586.

**Robin, M. H., Ur Rahman, M. M., Taief, A. M., Nahar Eity, Q. 2020.** Improvement of Face and Eye Detection Performance by Using Multi-task Cascaded Convolutional Networks, *2020 IEEE Region 10 Symposium (TENSymp)*, 977–980.

**Said, Y., Atri, M., Tourki, R. 2011.** Human detection based on integral Histograms of Oriented Gradients and SVM, *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, 1–5.

**Schroff, F., Kalenichenko, D., Philbin, J. 2015.** FaceNet: A Unified Embedding for Face Recognition and Clustering, *CoRR abs/1503.03832*. arXiv: 1503.03832.

**Soukupová, T., ech, J. 2016.** Real-Time Eye Blink Detection using Facial Landmarks,

**Sripriya, A. V., Geethika, M., Radhesyam, V. 2020.** Real Time Detection and Recognition of Human Faces, *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 703–708.

**Sun, X., Wu, P., Hoi, S. C. H. 2017.** Face Detection using Deep Learning: An Improved Faster RCNN Approach. arXiv: 1701.08289.

**Sun, Y., Liang, D., Wang, X., Tang, X. 2015.** DeepID3: Face Recognition with Very Deep Neural Networks, *CoRR abs/1502.00873*. arXiv: 1502.00873.

**Sun, Y., Wang, X., Tang, X. 2014.** Deep Learning Face Representation by Joint Identification - Verification, *CoRR abs/1406.4773*. arXiv: 1406.4773.

- Taigman, Y., Yang, M., Ranzato, M., Wolf, L. 2014.** DeepFace: Closing the Gap to Human-Level Performance in Face Verification, 2014 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 1701–1708.
- Turk, M., Pentland, A. 1991.** Eigenfaces for Recognition, *Journal of Cognitive Neuroscience* 3.1. PMID: 23964806, 71–86.
- Viola, P., Jones, M. 2001.** Rapid object detection using a boosted cascade of simple features, *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001. Vol. 1, I–I.*
- Viola, P., Jones, M. J. May 2004.** Robust Real-Time Face Detection, *57.2*, 137–154. ISSN: 0920-5691.
- Xiaofei He, Shuicheng Yan, Yuxiao Hu, Niyogi, P., Hong-Jiang Zhang 2005.** Face recognition using Laplacianfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.3, 328–340.
- Yang, B., Chen, S. 2013.** A comparative study on local binary pattern (LBP) based face recognition: LBP histogram versus LBP image, *Neurocomputing* 120, 365–379.
- Yeh, C., Chang, H. 2017.** Face liveness detection with feature discrimination between sharpness and blurriness, 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), 398–401.
- Zhang, B., Shan, S., Chen, X., Gao, W. 2007.** Histogram of Gabor Phase Patterns (HGPP): A Novel Object Representation Approach for Face Recognition, *IEEE Transactions on Image Processing* 16.1, 57–68.
- Zhang, K., Zhang, Z., Li, Z., Qiao, Y. 2016.** Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks, *IEEE Signal Processing Letters* 23.10, 1499–1503. ISSN: 1558-2361.
- Zhang, N., Luo, J., Gao, W. 2020.** Research on Face Detection Technology Based on MTCNN, 2020 International Conference on Computer Network, Electronic and Automation (ICCNEA), 154–158.
- Zuo, Y., Gao, W., Wang, J. 2020.** Face Liveness Detection Algorithm based on Livenesslight Network, 2020 International Conference on High Performance Big Data and Intelligent Systems (HPBD IS). IEEE, 1–5.

## ÖZGEÇMİŞ

Adı Soyadı : Sedat YILDIZ

Doğum Yeri ve Tarihi : Elazığ, 1994

Yabancı Dil : İngilizce

Eğitim Durumu (Kurum ve Yıl)

Lise : Bursa Atatürk Anadolu Lisesi, 2012

Lisans : Uludağ Üniversitesi Elektrik Elektronik Mühendisliği, 2018

İletişim : sedatyildiz\_1623@hotmail.com

Çalıştığı Kurum : Bursa Teracity Yazılım

### Tezden Yapılan Yayınlar

**Yıldız, S., Özgür, E. 2020.** Derin Öğrenme Tabanlı Yüz Tanıma Sisteminin İkizler ve Yaşlanma Üzerindeki Başarımının Ölçülmesi. *Elektrik-Elektronik ve Biyomedikal Mühendisliği Konferansı (ELECO20)*, Bildiri No: 4

**Yıldız, S., Özgür, E., Bilal, N. 2019.** Yapay Zeka Tabanlı Yüz Tanıma Sisteminin Geliştirilmesi Ve Optimizasyonu *Ulusal Yazılım Mühendisliği Sempozyumu (UYMS19)*