

T.C.  
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
İSTATİSTİK ANABİLİM DALI

**İÇERİK TABANLI GÖRÜNTÜ ERİŞİM YÖNTEMLERİYLE AİLE  
BİREYLERİNDE YÜZ TANIMA SİSTEMİ**

DOKTORA TEZİ

HAZIRLAYAN: Faruk AYATA  
DANIŞMAN: Doç. Dr. Hayati ÇAVUŞ

VAN-2020



T.C.  
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ  
İSTATİSTİK ANABİLİM DALI

**İÇERİK TABANLI GÖRÜNTÜ ERİŞİM YÖNTEMLERİYLE AİLE  
BİREYLERİNDE YÜZ TANIMA SİSTEMİ**

DOKTORA TEZİ

HAZIRLAYAN: Faruk AYATA

VAN-2020



## KABUL VE ONAY SAYFASI

İstatistik Anabilim Dalı'nda Doç. Dr. Hayati ÇAVUŞ danışmanlığında, Faruk AYATA tarafından sunulan “İçerik Tabanlı Görüntü Erişim Yöntemleriyle Aile Bireylerinde Yüz Tanıma Sistemi” isimli bu çalışma Lisansüstü Eğitim ve Öğretim Yönetmeliği'nin ilgili hükümleri gereğince ....../...../..... tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile başarılı bulunmuş ve ..... tezi olarak kabul edilmiştir.

Başkan:.....

İmza:

Üye:.....

İmza:

Üye:.....

İmza:

Üye:.....

İmza:

Üye:.....

İmza:

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun ....../...../..... tarih ve ..... sayılı kararı ile onaylanmıştır.

İmza

.....

Enstitü Müdürü

Prof. Dr. Suat ŞENSOY



## TEZ BİLDİRİMİ

Tez içindeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edilerek sunulduğunu, ayrıca tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

*(İmza)*

Faruk AYATA





## ÖZET

### İÇERİK TABANLI GÖRÜNTÜ ERİŞİM YÖNTEMLERİYLE AİLE BİREYLERİNDE YÜZ TANIMA SİSTEMİ

AYATA, Faruk

Doktora Tezi, İstatistik Anabilim Dalı

Tez Danışmanı: Doç. Dr. Hayati ÇAVUŞ

Aralık 2020, 149 sayfa

Yüz tanıma sistemleri, kriminoloji, güvenlik sistemleri gibi görüntü içeren alanlarda veriyi işlemek için görüntü işleme, makine öğrenmesi ve derin öğrenme algoritmaları kullanılmaktadır. Yüz tanıma sistemleri kayıp bireylerin bulunması için umut verici biyometrik teknolojilerden biri olabilir; çünkü ebeveynlerin ve akrabalarının kaybolan bir bireyin yüz fotoğrafına sahip olma olasılığı, parmak izi veya iris gibi diğer biyometrik yöntemlerden daha yüksek olabilmektedir. Bu tez çalışmasında, kaybolan, kaçırılan ya da aranan bireylerin bulunması temeline dayanan içerik tabanlı bir görüntü erişim sistemi tasarlanıp, tasarlanan bu sistem ile yüz tahmini yapmak amaçlanmaktadır. Bu kapsamda sistemde kullanılmak üzere uygun makine öğrenmesi tekniğinin bulunabilmesi için FEI ve CelebA veri seti temin edilmiştir. FEI, CelebA ve aile yüz veri seti üzerinde Makine öğrenmesi tekniklerinden; ESA (Evrişimsel Sinir Ağları), YGH-DVM (Yönelimli Gradyan Histogramı-Destek Vektör Makineleri) ve DSA (Derin Sinir Ağları) yöntemleri kullanılmıştır. Bu yöntemlerin analiz sonuçları değerlendirilerek en iyi yüz tanıma özelliğine sahip olan yöntem olarak %98,86 başarı oranı ile ESA belirlenmiştir. Sonrasında, tarafımızca oluşturulan ve hiçbir çalışmada henüz kullanılmamış olan, aile yüz veri setine ESA tekniği uygulanarak aile yüz referans modeli ve bu model baz alınarak robot resim oluşturulmuştur. Aile yüz veri seti içerisinde rastgele seçilen 10 ailenin aile referans modeline ait robot resmi ile aileden seçilen bir birey kıyaslanarak %88,77 ile %93,90 arasında başarı oranı elde edilmiştir. Ayrıca web tabanlı bir robot resim oluşturma programı hazırlanmıştır.

**Anahtar kelimeler:** Aile yüz veri seti, Evrişimsel sinir ağı, Makine öğrenmesi, Robot resim oluşturma, Yüz tanıma.



## ABSTRACT

### FACE RECOGNITION SYSTEM IN FAMILY INDIVIDUALS WITH CONTENT-BASED IMAGE ACCESS METHODS

AYATA, Faruk

Ph.D. Thesis Department of Statistics

Supervisor: Assoc. Prof. Dr. Hayati ÇAVUŞ

December 2020, 149 page

Image processing, machine learning and deep learning algorithms are used to process data in fields such as facial recognition systems, criminology, security systems. Facial recognition systems can be one of the promising biometric technologies for finding missing individuals, as parents and relatives are more likely to have a face photo of missing individual rather than other biometric methods such as fingerprints or iris. In this thesis, a content-based image retrieval system based on finding missing, kidnapped or wanted individuals is designed and it is aimed to make face estimation with this system. In this context, FEI data set has been provided to find the appropriate machine learning technique to be used in the system. Among Machine Learning techniques on FEI, CelebA and family face dataset; ESA (Evrişimsel sinir Ağları), YGH-DVM (Yönelimli Gradyan Histogramı - Destek Vektör Makineleri) and DSA (Derin Sinir Ağları) methods were used. Evaluating the analysis results of these methods, ESA was determined as the method with the best facial recognition feature, with a 98,86% success rate. Afterwards, the family face reference model was applied to the family face data set, which was created by us and has not been used in any study, and a police sketch was created based on this model. A success rate between 88,77% and 93,90% was obtained by comparing the police sketch of the family reference model of 10 families randomly selected from the family face data set and an individual selected from the family. In addition, a web-based robot image creation program has been prepared.

**Keywords:** Convolutional neural network, Create a sketch, Face recognition, Family face data set, Machine learning.



## ÖN SÖZ

Bu tez çalışmasının her noktasında bana öneri ve destekleriyle yol gösteren, değerli katkılarını esirgemeyen danışman hocam Sayın Doç. Dr. Hayati ÇAVUŞ'a, çalışmalarım süresince bilgi ve tecrübesinden faydalandığım Sayın Prof. Dr. Eray ÇELİK'e ve Sayın Prof. Dr. Murat KAYRI'ye, tez çalışmam boyunca desteklerini hiçbir zaman esirgemeyen aileme ve eşime, ayrıca katkılarından dolayı değerli çalışma arkadaşlarım Öğr. Gör. Ebubekir SEYYARER'e, Öğr. Gör. Dr. Taner UÇKAN'a ve Öğr. Gör. Mevlüt İNAN'a sonsuz teşekkürlerimi sunarım.

2020

Faruk AYATA



## İÇİNDEKİLER

	Sayfa
ÖZET .....	i
ABSTRACT .....	iii
ÖN SÖZ.....	v
İÇİNDEKİLER.....	viii
ÇİZELGELER LİSTESİ .....	xii
ŞEKİLLER LİSTESİ.....	xiv
SİMGELER VE KISALTMALAR .....	xvii
EKLER DİZİNİ.....	xx
1. GİRİŞ.....	1
2. KAYNAK BİLDİRİŞLERİ .....	5
3. MATERYAL VE YÖNTEM.....	13
3.1. Materyal.....	13
3.1.1. Görüntü işleme.....	13
3.1.1.1. Görüntü işleme teknikleri .....	14
3.1.1.2. Görüntü işlemenin uygulama alanları .....	15
3.1.2. İTGE sistemleri.....	15
3.1.3. Makine öğrenmesi .....	18
3.1.3.1. Makine öğrenmesi algoritmaları .....	19
3.1.4. DSA .....	20
3.1.5. ESA.....	24
3.1.5.1. Giriş katmanı.....	25
3.1.5.2. Evrişim katmanı .....	26
3.1.5.3. Düzleştirilmiş doğrusal birim katmanı.....	27
3.1.5.4. Örnekleme katmanı .....	28
3.1.5.5. Düzleştirme katmanı .....	39
3.1.5.6. Tamamen bağlı katmanı.....	30
3.1.6. YGH-DVM .....	31
3.1.7. Uzaklık ölçüm yöntemleri .....	34

	<b>Sayfa</b>
3.1.7.1. Öklid uzaklık ölçüsü .....	34
3.1.7.2. Chebychev uzaklık ölçüsü .....	35
3.1.7.3. Manhattan (City-Block) uzaklık ölçüsü.....	35
3.1.7.4. Minkowski uzaklık ölçüsü .....	36
3.1.7.5. Karl-Pearson uzaklık ölçüsü .....	36
3.1.8. Yüz bulma yöntemleri .....	37
3.1.8.1. Bilgi tabanlı yöntemler .....	37
3.1.8.2. Özellik tabanlı yöntemler.....	37
3.1.8.3. Şablon tabanlı yöntemler .....	37
3.1.8.4. Görünüm tabanlı yöntemler .....	38
3.1.9. Yüz nirengi noktalarının bulunması .....	40
3.1.10. Veri seti.....	43
3.1.10.1. FEI veri seti.....	43
3.1.10.2. CelebA veri seti.....	43
3.1.10.3. Aile yüz veri seti .....	44
3.2. Yöntem .....	45
3.2.1. Veri setlerinin düzenlenmesi .....	45
3.2.1.1. FEI veri setinin düzenlenmesi.....	45
3.2.1.2. CelebA veri setinin düzenlenmesi .....	46
3.2.1.3. Aile yüz veri setinin düzenlenmesi .....	46
3.2.2. Yöntemin belirlenmesi ve testi .....	46
3.2.2.1. YGH-DVM algoritması .....	47
3.2.2.2. DSA yöntemi .....	48
3.2.2.3. ESA yöntemi.....	50
3.2.3. Belirlenen yöntemin veri setine uygulanması ve yüz nirengi noktalarının tespiti .....	52
3.2.4. Yüz nirengi noktalarının düzenlenmesi ve kaydedilmesi.....	58
3.2.5. Yüz nirengi noktaları arasındaki mesafenin hesaplanması ve kaydedilmesi .....	63
3.2.6. Aile yüz referans modelinin oluşturulması.....	67
3.2.7. Yüz giydirme işlemi .....	74



	<b>Sayfa</b>
4. BULGULAR .....	86
5. TARTIŞMA VE SONUÇ .....	97
KAYNAKLAR .....	100
EKLER .....	108
ÖZ GEÇMİŞ .....	147



## ÇİZELGELER LİSTESİ

Çizelge	Sayfa
Çizelge 3.1. Aktivasyon fonksiyonları ve türevleri.....	28
Çizelge 3.2. Yüz bulma yöntemlerinin karşılaştırılması .....	39
Çizelge 3.3. Yüz organ kütüphanesi bölümlerinin bir kısmı.....	78
Çizelge 3.4. Yüz organ kütüphanesi bölümlerinin devamı .....	79
Çizelge 4.1. ESA yöntemiyle veri setleri üzerinde yapılan test sonuçları.....	86
Çizelge 4.2. DSA yöntemiyle veri setleri üzerinde yapılan test sonuçları .....	86
Çizelge 4.3. YGH-DVM yöntemiyle veri setleri üzerinde yapılan test sonuçları.	86
Çizelge 4.4. Uygulanan yöntemlerin veri setlerine göre başarı oranları .....	87
Çizelge 4.5. Aile yüz organlarının benzerlik oranları .....	88
Çizelge 4.6. Tahmin performansı hata ölçüm değerleri .....	90
Çizelge 4.7. Rastgele seçilen 10 aile üzerinde SSIM yöntemiyle yapılan test sonuçları.....	91
Çizelge 4.8. Rastgele seçilen 10 aile ve o ailelerden seçilen bireylerin robot resimleri .....	92
Çizelge 4.9. Rastgele seçilen 10 aile ve o ailelerden seçilen bireylerin robot resimleri (devamı).....	93
Çizelge 4.10. Dört aile benzerlik sorgulama tahmin performansı hata değerleri ..	96



## ŞEKİLLER LİSTESİ

Şekil	Sayfa
Şekil 3.1. Görüntü İşleme Teknikleri .....	14
Şekil 3.2. Temel bir İTGE sistemi.....	16
Şekil 3.3. İTGE sistemi blok diyagramı .....	16
Şekil 3.4. İTGE sistemlerde benzerlik karşılaştırması ve sınıflandırma .....	18
Şekil 3.5. Nöron yapısı .....	20
Şekil 3.6. İnsandaki bir sinir hücresinin matematiksel modeli ise şu şekilde gösterilebilir .....	21
Şekil 3.7. Yapay sinir ağı .....	22
Şekil 3.8. Üç gizli katmandan oluşan derin sinir ağı örneği.....	23
Şekil 3.9. ESA (Evrşimsel Sinir Ağı).....	25
Şekil 3.10. 3x4 boyutta bir giriş görüntüsüne 2x2'lik filtrenin uygulandığı konvolüsyon işlemi.....	27
Şekil 3.11. Örnekleme katmanı .....	28
Şekil 3.12. 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı maksimum havuzlama işlemi .....	29
Şekil 3.13. Düzleştirme katmanı.....	30
Şekil 3.14. Tamamen bağlı katman .....	30
Şekil 3.15. YGH blok diyagramı .....	32
Şekil 3.16. DVM ile sınıflandırma işlemi .....	33
Şekil 3.17. Herhangi iki gözlem arasındaki uzaklık.....	34
Şekil 3.18. Yüz nirengi noktaları.....	40
Şekil 3.19. Eşleştirilmiş iki şekil .....	42
Şekil 3.20. FEI veri setinden bir kesit .....	43
Şekil 3.21. CelebA veri setinden bir kesit .....	44

<b>Şekil</b>	<b>Sayfa</b>
Şekil 3.22. Aile yüz veri setinden örnek bir aile .....	44
Şekil 3.23. FEI yüz veri seti klasörleri .....	46
Şekil 3.24. Aile yüz veri seti klasörleri .....	46
Şekil 3.25. YGH-DVM yöntemi için kullanılan algoritma (a) ve akış diyagramı (b) .....	47
Şekil 3.25. YGH-DVM yöntemi ile yüzün tespiti .....	48
Şekil 3.25. DSA yöntemi için kullanılan algoritma (a) ve akış diyagramı (b) .....	49
Şekil 3.26. DSA yöntemi ile yüzün tespiti .....	49
Şekil 3.25. ESA yöntemi için kullanılan algoritma (a) ve akış diyagramı (b) .....	50
Şekil 3.27. ESA yöntemi ile yüzün tespiti .....	51
Şekil 3.28. Yüz tanımlama blok diyagramı .....	53
Şekil 3.29. Crop_function ile 300x400 olarak boyutlandırılmış görüntü .....	54
Şekil 3.30. Görüntü üzerindeki yüzün tespiti, yüz noktalarının bulunması ve kaydedilmesi .....	56
Şekil 3.31. İstenmeyen kısımları atılan ve yeniden boyutlandırılan görüntü .....	57
Şekil 3.32. Main_function adımları .....	58
Şekil 3.33. Yüz nirengi nokta koordinat değerlerinin tutulduğu dosya içeriğinin bir kısmı .....	59
Şekil 3.34. Birey yüz nirengi nokta koordinat değerlerinin veri tabanına eklenmesi .....	60
Şekil 3.35. Birey .csv dosyası seçim diyalog penceresi .....	61
Şekil 3.36. Birey index ve koordinat değerlerinin görüntülenmesi .....	62
Şekil 3.37. Birey index ve koordinat değerlerinin veri tabanına kaydedilmesi .....	63
Şekil 3.38. Öklid uzaklık hesaplama ve kayıt ara yüzü .....	64
Şekil 3.39. Öklid uzaklık kontrol program ara yüzü .....	65
Şekil 3.40. Aile birey Öklid uzaklık listeleme ara yüzü .....	66

<b>Şekil</b>	<b>Sayfa</b>
Şekil 3.41. Öklid uzaklık değerlerinin listeleme bölümlerine aktarılması .....	67
Şekil 3.42. Referans yüz modeli oluşturma ara yüzü .....	68
Şekil 3.43. Aile adına göre listeleme ara yüzü .....	69
Şekil 3.44. Aile ya da birey silme ara yüzü .....	70
Şekil 3.45. Aile referans yüz koordinat değerlerinin hesaplanması ve listelenmesi .....	71
Şekil 3.46. Referans yüz koordinat değerlerinin kaydedildiği dosya içeriğinin bir kısmı .....	72
Şekil 3.47. Referans yüz modeli .....	73
Şekil 3.48. Referans yüz koordinat değerlerinin ve resminin kaydedilmesi .....	73
Şekil 3.49. Robot resim çizme program ara yüzü .....	75
Şekil 3.50. Aile referans yüz modelinin çalışma alanına eklenmesi .....	77
Şekil 3.51. Referans aile görüntüleme alanı .....	81
Şekil 3.52. Referans aile birey büyütme alanı .....	82
Şekil 3.53. Program kullanılarak oluşturulan örnek robot resimler .....	83
Şekil 3.54. Sistemin çalışmasını gösteren genel blok diyagramı .....	84
Şekil 4.1. Uygulanan yöntemlerin veri setlerine göre başarı oran grafiği .....	87
Şekil 4.2. Aile yüz organlarının benzerlik oranlarını gösteren grafik .....	89
Şekil 4.3. 1- 4 Aile tahmin/gerçek değer grafikleri .....	94
Şekil 4.4. 5 – 8 Aile tahmin/gerçek değer grafikleri .....	95
Şekil 4.5. 9. ve 10. aile tahmin/gerçek değer grafikleri .....	96





## SİMGELER VE KISALTMALAR

Bu çalışmada kullanılmış bazı simgeler ve kısaltmalar, açıklamaları ile birlikte aşağıda sunulmuştur.

Kısaltmalar	Açıklama
<b>YSA</b>	Yapay Sinir Ağları
<b>ESA</b>	Evrişimsel Sinir Ağları
<b>DVM</b>	Destek Vektör Makineleri
<b>B-ESA</b>	Bölgesel Evrişimsel Sinir Ağları
<b>YGH</b>	Yönelimli Gradyan Histogramı
<b>KYK</b>	K- Yakın Komşu
<b>RO</b>	Rastgele Orman
<b>TSA</b>	Tekrar Sinir Ağları
<b>DSA</b>	Derin Sinir Ağları
<b>İTGE</b>	İçerik Tabanlı Görüntü Erişim
<b>TBA</b>	Temel Bileşenler Analizi



## EKLER DİZİNİ

<b>Ek</b>	<b>Sayfa</b>
Ek 1. Etik Kurul Raporu .....	108
Ek 2. Web tabanlı robot resim oluşturma uygulamasına ait görüntüsü ve java script kodları .....	110



## 1. GİRİŞ

İlerleyen teknolojik gelişmelerle beraber büyük veri günlük yaşantımızın her alanına girmektedir. CNC (Computer Numerical Control) cihazları, üretim makineleri, ses kayıt cihazları, uçak motorları, sensörler, cep telefonları, web kameraları, güvenlik kameraları gibi aygıtlarla bir saniyede bile milyonlarca veri kaydedilmektedir. Kaydedilen bu veriler içerisinden görüntü olanlar; yüz tanıma sistemleri, kriminoloji, güvenlik sistemleri gibi alanlarda kullanılarak insanların manuel olarak kontrolünü gerçekleştirdiği kimlik kontrolü, güvenli giriş-çıkış kontrolü ve personel takibi gibi işlerde üzerine düşen iş yükünü azaltmaktadır. Bu alanlarda veriyi işlemek için görüntü işleme, makine öğrenmesi ve derin öğrenme algoritmaları kullanılmaktadır. Elde edilen analiz sonuçları ilgili işletmelerin daha iyi hizmet verebilmeleri ve kar marjlarını arttırmaları için kullanıma sunulmaktadır.

Ses, görüntü ve belge inceleme alanlarındaki ilerleme başta makine öğrenmesi olmak üzere derin öğrenme ve YSA (Yapay Sinir Ağları) gibi alanların da etkisiyle mümkün olmaktadır. Görüntü inceleme alanı; dijital görüntü işleme, yüz tanıma, nesne tanıma, desen tanıma, kod tanıma ve optik karakter tanıma gibi teknolojileri barındırmaktadır (Anonim, 2020a). Bu teknolojilerden yüz tanıma teknolojisi kayıp bireylerin bulunması için umut verici biyometrik teknolojilerden biri olabilir. Çünkü ebeveynlerinin ve akrabalarının kaybolan bir bireyin yüz fotoğrafına sahip olma olasılığı, parmak izi veya iris gibi diğer biyometrik yöntemlerden daha yüksek olabilmektedir.

Yüz tanıma sistemlerinin kullanımı diğer biyometrik tanıma sistemlerine göre daha kolay olabilmektedir. İris, retina ve parmak izi tanıma sistemlerinde kişinin göz tarayıcısına bakması ya da parmak izini belirlenen cihazın üzerine tutması gerekmektedir. Oysaki yüz tanıma sisteminde böyle bir gereklilik olmayabilir, çünkü kişiyi rahatsız etmeden de görüntüsü alınabilir.

Yüz tanıma sistemleri dört ana bölümden oluşmaktadır. Bunlar; Algılama, Hizalama, Öznetelik çıkarma ve Eşleştirme'dir (Holat, 2016). Çalışma prensibi olarak ise iki temel yönetime dayanmaktadır; bunlardan birincisi TBA (Temel Bileşenler Analizi) diğeri ise bir yüz üzerindeki organlardan öz nitelik vektörleri çıkarmaya dayalı

yöntemdir (Erdoğan, 2010). Yüz öznitelik çıkarımı, yüz üzerindeki ölçülebilir ya da gözlenebilir bilgilerden ilgisiz ve fazla bilginin elenerek örüntüye ait karakteristik özelliklerin elde edilmesini sağlamaktadır (Kurt, 2019).

Görüntülerin işlenmesi ve sınıflandırılması için birçok teknik (Bkz. Şekil 3.1) ve bu tekniklere ait algoritmalar kullanılmaktadır. Bu algoritmaların uygulamadaki başarı oranları çok yüksek değildir, çünkü kameranın açısı, aydınlatma, poz ve yüz ifadesi gibi değişkenler yüzün tanınmasında zorluklar çıkarabilmektedir (Varol ve Cebe, 2011). Yüksek başarı oranı elde edilmek istenildiği durumlarda YSA tabanlı sınıflandırma algoritmalarının tercih edilmesi daha doğru olmaktadır (Pişkin, 2020). YSA, insan beyninin çalışma sisteminin taklit edilmesiyle ortaya çıkmış ve 1980-2000 yılları arasında büyük başarı sağlamış bir yöntemdir (Doğan, 2010). YSA kullanarak geliştirilen yüz tanıma sistemlerinde kişilerin yüz bilgileri sinir ağına eğitim verisi olarak verilmektedir. Daha sonra sinir ağı bu yüzleri baskın özelliklere (öz nitelik çıkarımı) göre sınıflandırma yapar. Tanıma için verilen yüz sinir ağına geldiğinde ise öz nitelikler daha önceden çıkartıldığı için hızlı bir şekilde gelen yüzün hangi sınıfa ait olduğu belirlenmektedir.

YSA'ların geliştirilmesiyle elde edilen ve daha iyi beyin modellemesine sahip mimarilere Derin Öğrenme adı verilmektedir. Derin öğrenmenin özelleştirilmiş bir mimarisi olan ESA yöntemleri görüntü işleme ve yüz tanıma sistemlerinde oldukça başarılıdır (Cengil ve Çınar, 2016).

Yukarıda belirtilen alanlarda yapılan literatür incelemeleri sonucunda günümüz yüksek teknolojisi olan makine öğrenmesi ve yapay zekâdan en yüksek verimde faydalandığı görülmektedir. Yapay zekâ; bilgisayar ve bilgisayar destekli makinalara, insanlara özgü olan karar verme, akıl yürütme, varsayımda bulunma gibi yeteneklerin kazandırılması işlemidir. Kısacası, algoritma üretebilen otomatik sistemlerin genel adıdır (Kılıç, 2019; Süer, 2019). Makine öğrenmesi; programlanmadan büyük miktardaki veriler üzerinden kendi kendine öğrenmeyi, sonuç çıkarmayı ve sınıflandırmayı yapabilmektedir. Ayrıca yapay zekâ algoritmalarını içeren bir teknolojidir (Razikou Dilwani, 2019; Sakarya, 2019).

İnsan hakları; tüm insanlara ayırım gözetmeksizin eşit, özgür ve onurlu yaşama hakkı tanımaktadır. Türkiye tarafından 1954'te onaylanmış olan ve iç mevzuatın bir parçasını oluşturan Avrupa İnsan Hakları Sözleşmesi, temel hak ve özgürlükleri:

Yaşama hakkı, işkence, insan ticareti yasağı, insanlık dışı veya küçültücü muamele yasağı, kölelik ve zorla çalıştırma yasağı, kişi özgürlüğü ve güvenliği olarak ifade etmektedir (Anonim, 2020b).

İnsan haklarının ihlali olarak bilinen insan ticareti; çocukların, erkeklerin ve kadınların çıkar elde etmek amacıyla kaçırılması, alıkonulması ve çalıştırılması anlamına gelmektedir. Ayrıca iyi bir yaşam isteği, eğlence sektörüne katkı, ekonomik ve politik sebepler, caydırıcı cezaların yetersiz olması gibi sebeplerden ötürü tüm dünyada yaşanmaktadır. Bu durum dünya ülkeleri tarafından karşılaşılan en olumsuz sosyal sorunlardan biridir. Birleşmiş Milletler Çocuklara Yardım Fonu'na (UNICEF) ve İnsan Ticaretiyle Mücadele Ajanslar Arası Koordinasyon Grubuna (ICAT) göre küresel insan ticareti mağduru olanların % 28'ini çocuklar oluşturmaktadır (Anonim, 2020c). 2008-2016 yılları arasındaki Türkiye İstatistik Kurumunun (TÜİK) verilerine göre kaybolan çocukların sayısı 104,531'e ulaşmıştır. Kaçırılma vakalarının dışında kendi isteğiyle evden kaçanların sayısı 26,168 olarak tespit edilmiştir (Anonim, 2020d). İnsan ticaretini engelleme adına Türkiye'de birçok yasal ve idari düzenleme yapılmıştır. 1 Haziran 2005'te yürürlüğe giren Türk Ceza Kanunu'nun 80. maddesine göre 8 yıldan 12 yıla kadar hapis cezası getirilmiştir.

Her ne sebeple olursa olsun insan ticareti suçunu işleyip bireyleri hürriyetinden yoksun bırakan suçluların ve bu suça maruz kalan aile bireylerinin bulunması noktasında emniyet güçleri; olay yeri inceleme, iz inceleme, balistik inceleme, belge inceleme, antropolojik inceleme, ses ve görüntü kayıtlarının incelenmesi gibi alanlarda büyük bir çaba sarf etmektedir.

Bu tez çalışmasının temel amacı; kaybolan, kaçırılan ya da aranan bireylerin bulunması temeline dayanan içerik tabanlı bir görüntü erişim sistemini tasarlamak ve yüz tahmini yapmaktır. Bu kapsamda; ilk adımda yüz tahmininde kullanılmak üzere 39 aileden yüz resimleri toplanmış ve bu resimlerin düzenlenmesi (3. Bölümde bu düzenlemeler hakkında bilgi verilmiştir) yapılmıştır. İkinci adımda uygun makine öğrenme tekniğinin bulunabilmesi için FEI ve CelebA veri setleri temin edilmiştir. Üçüncü adımda FEI, CelebA ve Aile yüz veri seti üzerinde Makine Öğrenme tekniklerinden ESA, DVM ve DSA yöntemleri kullanılmıştır. Bu yöntemlerin analiz sonuçları değerlendirilerek en iyi yüz tanıma özelliğine sahip olan yöntem belirlenmiştir. Dördüncü adımda analizler sonucunda bulunun en uygun yöntem, Aile

yüz veri setine uygulanıp yüz nirengi noktaları tespit edilmiştir. Beşinci adımda, bulunan yüz nirengi noktalarının koordinat düzlemindeki temsili değerleri Öklid uzaklık yöntemi (bu yöntemin seçilme nedeni 3.Bölümde açıklanmıştır) kullanılarak aralarındaki mesafe hesaplanmıştır. Altıncı adımda bu mesafe değerlerini kullanarak aile yüz referans modeli oluşturulmuştur. Yedinci adımda aile yüz referans modeli baz alınarak robot resim oluşturulmuştur.



## 2. KAYNAK BİLDİRİŞLERİ

Görüntü işleme, dijital görüntüler üzerinde gelişmiş resim işleme tekniklerini uygulamak için bilgisayar hesaplamalarının kullanılmasıdır. Bilgisayarlar daha karmaşık verilerin işlenmesine imkân tanımaktadır. Görüntüler üzerindeki gürültüleri minimum seviyeye indirerek iyileştirmeler gerçekleştirebilir.

Görüntülerin dijital olarak işlenmesi büyük ölçüde öngörülemeyen hesaplamaların kullanılmasına imkân tanır. Bu sayede hem zaman kaybı yaşanmaz hem de süreçte yaşanabilecek hatalar azaltılmış olur (Sıddık, 2019). Görüntü işleme teknikleri günümüzde birçok alana uygulanabilmektedir ve bu alanlardan birkaçı aşağıda listelenmiştir.

- Sağlık sektöründe yapılan çalışmalar; cilt kanserinin teşhisinde (Kassem, 2016), *Pseudomonas Aeruginosa* hastalığının teşhisinde (Amon, 2017), meme kanserinin teşhisinde (Taşdemir, 2018), göz damarlarında gelişen hastalıkların teşhisinde (Nabiyev, 2007; Tükel, 2019), radyolojik görüntülerin sınıflandırılmasında (Karhan, 2014).
- Uydu görüntülerinden yararlanarak bölgesel su kaynaklarının tespiti, kara bölgelerinin tespitinde kullanılmaktadır (Şenel, 2018).
- Desen, doku ve şekil bilgisinden yararlanarak hayvan, bitki, yiyecek ve nesnelerin sınıflandırılmasında kullanılmaktadır (Kaya, 2014; Karhan, 2014; Horosan, 2016; Nawaf, 2019; Rashid, 2019; Seyyarer ve Aydın, 2017).
- Veri madenciliğinde kullanılmaktadır (Jahangiri, 2015).
- Güvenlik sistemlerinde kullanılmaktadır (Alhamrouni, 2017; Ayata, 2019).

Görüntü işlemenin tıp, bilişim, savunma ve güvenlik gibi birçok alanda kullanılması sayısallaştırılmış görüntülerin miktarını ve kalitesini oldukça arttırmıştır. Bu artış büyük veri yığınlarına sebep olmuştur. Bu yığınlar içerisinde istenilen görüntünün aranması oldukça zor ve zaman alan bir süreçtir. 1990 yılında İçerik Tabanlı Görüntü Erişim (İTGE) sistemlerinin geliştirilmesiyle büyük veri yığınları içerisinde istenilen görüntülerin bulunması süreci kolaylaşmıştır (Kılıçaslan ve ark, 2020). İTGE

sistemi iki aşama olarak tanımlanabilir: İndeksleme ve Arama. İndeksleme aşamasında, veri tabanındaki her bir görüntü renk, doku ve şekil özelliklerinden oluşan bir dizi özellik ile temsil edilir. Arama aşamasında, kullanıcı bir sorgu görüntüsü seçtiğinde, bu sorgunun bir dizi özelliği çıkartılır. Benzerlik değeri hesaplanır, bu değer veri tabanındaki özellik vektörleriyle karşılaştırılır ve kullanıcıya sorgu görüntüsüne en yakın veya benzer görüntüleri getirmektedir (Mohamed ve ark, 2008).

Jahangiri (2015), yapmış olduğu çalışmada İTGE sistemini farklı bir alan olan veri madenciliği ile harmanlayarak bir sistem mimarisi önermişlerdir. Bu mimaride 35 farklı kategorideki resimler üzerinde testler yaparak, görüntü sayısı artırılırsa verimlilik olasılığının da bir o kadar arttığını belirtmişlerdir. Ayrıca İTGE sistemlerini sınıflandırmada kullanan Kaya ve Kayci (2014), 380 kelebek görüntüsü kullanarak, geliştirdikleri sistemle kelebeklerin sınıflandırılmasında büyük bir başarı elde ettiklerini belirtmişlerdir. İTGE sistemlerini sağlık alanında kullanan Karhan (2014) ve Ergen (2010) geliştirdikleri sistemlerle radyolojik görüntülerin sınıflandırılmasına katkı sağlamışlardır.

Görüntü işleme alanı; dijital görüntü işleme, yüz tanıma, nesne tanıma, desen tanıma, kod tanıma ve optik karakter tanıma gibi teknolojileri barındırmaktadır (Anonim, 2020a). Bu teknolojilerden yüz tanıma; güvenlik sistemlerinde, personel, üye ve öğrenci takip sistemlerinde, geçiş kontrollerinde, kayıp veya aranan bireylerin bulunmasında kullanılabilmektedir.

Yüz tanıma sistemleri, dört ana bölümden oluşmaktadır: Algılama, Hizalama, Öznitelik çıkarma ve Eşleştirme (Holat, 2016). Çalışma prensibi olarak ise iki temel yönteme dayanmaktadır: Temel Bileşenler Analizi (TBA), Bir yüz üzerindeki organlardan öz nitelik vektörleri çıkarmaya dayalı yöntemdir (Erdoğan, 2010). Yüz öznitelik çıkarımı, yüz üzerindeki ölçülebilir ya da gözlenebilir bilgilerden ilgisiz ve fazla bilginin elenerek örüntüye ait karakteristik özelliklerin elde edilmesini sağlamaktadır (Kurt, 2019).

Görüntüleri bilgisayarda işleyebilen birçok algoritma mevcuttur. Bu algoritmalar; makine öğrenmesi, derin öğrenme ve YSA başlıkları altında toplanmaktadır. Bu algoritmaların uygulamadaki başarı oranları çok yüksek değildir; çünkü kameranın açısı, aydınlatma, poz ve yüz ifadesi gibi değişkenler yüzün tanınmasında zorluklar çıkarabilir (Varol ve Cebe, 2011). Yüksek başarı oranının elde

edilmek istenildiği durumlarda YSA tabanlı sınıflandırma algoritmalarının tercih edilmesi daha doğru olacaktır (Pişkin, 2020). YSA'ların geliştirilerek daha iyi bir beyin modellenme amacına sahip mimarileri derin öğrenme olarak adlandırılmaktadır. Derin öğrenmenin özelleştirilmiş bir mimarisi olan ESA görüntü işleme ve yüz tanıma sistemlerinde oldukça başarılıdır (Cengil ve Çınar, 2016). Mevcut literatürün taranması sonucunda yüz tanıma sistemlerinde ESA yöntemi kullanarak başarı sağlamış birkaç çalışma aşağıda verilmiştir.

Zhou ve ark. (2020); yüz tanıma sistemlerinde aydınlatma, poz ve yüz ifadesi gibi hataya sebep olan durumları en aza indirmek için ESA tabanlı bir sistem önermişlerdir. Yüz tanıma ve hizalama için kıvrımlı sinir ağı modeli tasarlamışlardır. Analiz sonuçlarında %95,3 başarılı tanıma gerçekleştirmişlerdir. Önerdikleri sistemle farklı açılarda çekilen görüntüler üzerinde başarılı sonuçlar elde etmişlerdir.

Ayata ve ark. (2020), ESA kullanarak yüz tanıma işlemini gerçekleştirip istemediğiniz kişilerin kişisel bilgisayarınızı kullanmasını kısıtlayan ve bu kısıtlamaya ek olarak kişisel bilgisayarınızı kullanmak isteyen kişinin fotoğrafını çekerek bu fotoğrafı sistemde daha önce tanımlanmış olan ve bilgisayar sahibine ait cep telefonuna mesaj olarak gönderen bir uygulama geliştirmişlerdir. Testler sonucunda en kötü açı ve ışık değerinde %76,31 ve en iyi açı ve ışık değerinde de %99,15 başarı sağlanmışlardır. Geliştirdikleri sistem ile ESA yöntemini yüz tanımada kullanarak kişisel bilgisayarların güvenliğini arttırmışlardır.

Kim ve ark. (2019), iki farklı kümeleme algoritmasına sahip düşük güç tüketimli bir yüz tanıma işlemcisi önermişlerdir. Kullandıkları iki kümeleme algoritmasıyla hesaplama yükünü azaltmışlardır ve mobil cihazlarda her zaman aktif olan 0,26mW güç tüketen yüz tanıma işlemcisi tasarlamışlardır.

Liu ve Wang (2019), geliştirdikleri sistem ağırlıklı ortalama yöntemini kullanarak ard arda çekilen resimlerden çıkartılan yüz ifadelerden ortalama değere göre uygun ifadeyi bulan sistemi ESA ile gerçekleştirmişlerdir. Hızlı çekimlerde karşılaşılan poz açısı ve ışık değerinden kaynaklanan farklı yüz ifadelerini süzerek tek bir ifadeye indirmişlerdir.

Godbole ve Burad (2019), farklı açılarda çekilen yüz görüntülerinden yüzü tanımak için yeni bir algoritma önermişlerdir. Ayrıca yüz görüntüsü tanıma işlemi yapıldıktan sonra yüz ifade tanıma işlemini de gerçekleştirmişlerdir.

Vinay ve ark. (2017), yaptıkları çalışma iki basit ama etkili ESA mimarisi kullanarak yüz tanıma işlemi yapmaktadır. Her iki mimaride de Gabor ve Frangi filtrelerini, veri seti olarak CMU, Grimace, Yale, Face 95 ve FEI kullanmışlardır. Analiz sonucunda veri setlerine göre başarı oranının %90 ile %100 arasında değiştiğini belirtmişlerdir. Görüntülerden yüzleri tanıma noktasında ESA yönteminin ne kadar etkili olduğunu ortaya koymuşlardır.

Coşkun ve ark. (2017), geliştirdikleri sistemde ESA katmanlarından ikisine iki normalleştirme işlemi ekleyerek yeni bir mimari önermişlerdir. Bu mimari sayesinde ağın hızlanmasını sağlamışlardır. Ayrıca yüzleri sınıflandırmak için Softmax sınıflandırıcı kullanmışlar ve test için Georgia veri seti kullanarak analiz sonuçlarında yüksek başarı oranı elde ettiklerini belirtmişlerdir. Geliştirdikleri sistemle yüz tanıma işlemlerindeki zamansal probleme çözüm üretmişlerdir.

Yüz tanıma, yüz izleme, duygu analizi ve yüz modelleme alanlarında sıklıkla kullanılan yüz nirengileme, insan yüzündeki ağız, burun, kaş, göz ve yüz çehresi gibi ortak bulunan bazı referans noktaların saptanmasıdır (Çeliktutan, 2013; Wang, 2014; Cao, 2012).

İrk, cinsiyet, poz değişikliği, ışık, yüz ifade farklılıkları ve kullanılan aksesuarlar yüz nirengilemeyi oldukça zor bir hale sokmaktadır (Çeliktutan, 2013). İlgili mevcut literatür tarandığında yüz nirengi noktalarının tespiti; yüz tanıma sistemlerinde görüntüyü tanımda zorluklara sebep olan görünüm varyasyonlarını, poz ve oklüzyon problemlerini (diş ve çene bozuklukları) düzeltmede (Feng ve ark., 2017; Zhanpeng Zhang ve ark., 2014), yüz ifadelerinin tespitinde (Ayvaz ve Gürüler, 2017) ve yüz modelleme (Whitmarsh, 2008) alanlarında sıklıkla kullanılmaktadır. Yüz nirengi noktalarının tespitinde farklı yöntemler, algoritmalar kullanılabilmektedir. İlgili literatür tarandığında bu yöntemlerle ilgili birkaç çalışma aşağıda verilmiştir.

Fan ve Zhou (2016), Derin öğrenme teknikleri kullanılarak yüz üzerindeki noktaların bulunmasındaki sınırlılıkları araştırmışlardır. Bu çalışmanın testlerini 300W veri seti ve yüz noktalarının tahmini için de ESA kullanmışlardır. Çalışma sonucunda kaş ve göz yapılarının tanınmasında bazı sınırlılıkların olabileceğini ve bundan ötürü yüz noktalarının çıkartılmasında hataya sebep olabileceğini belirtmişlerdir.

Zhou (2014), Derin Evrişimli Sinir Ağları (Deep Convolutional Neural Networks -DCNN) ile geliştirdikleri sistemden yüz noktalarını bulmada oldukça başarılı

olmuşlardır. Geliştirdikleri sistemde yüz hatlarının kıvrımlı olmasından kaynaklanan hatayı, 4 katlı ESA ile ard arda tüm katların bir önceki kattan gelen bilgiyi düzenleme görevi vererek eğitmişlerdir. Testlerini 300W veri seti üzerinde yapıp yüz noktalarını bulmada başarılı sonuçlar elde ettiklerini belirtmişlerdir.

Cao ve ark. (2012) yüz noktalarının bulunmasında var olan sistemlerden daha hassas ve verimli olan açık şekil regresyonu yaklaşımını önermişlerdir. Önceki regresyon tabanlı yaklaşımlardan farklı olarak görüntüdeki yüz resmini çıkartmak için vektörel bir regresyon fonksiyonunu kullanmışlardır. Ayrıca önerdikleri sistemin zamansal olarak daha iyi çalıştığını belirtmişlerdir.

Le ve ark. (2011), Aktif Şekil Modeli (Active Shape Model -ASM) yönteminin performansını arttırmak için bazı iyileştirmeler önermişlerdir. Sistemin testlerini Caltech ve imm\_face yüz veri setlerinde yapıp en iyi sonucu ASM-DVM ile elde ettiklerini belirtmişlerdir.

Cootes ve ark. (1998), önerdikleri sistemde Aktif Görünüm Modelini (Active Appearance models-AAM) kullanarak görüntüleri yorumlayan bir yöntem sunmuşlardır. Geliştirdikleri sistem eğitim verisiyle görüntü arasındaki hataları tespit edip ilişkilendirir; bu ilişkiler sayesinde öğrenme işlemini gerçekleştirmişlerdir. Ayrıca AAM algoritması deforme nesneleri bulmak içinde kullanılabileceğini belirtmişlerdir.

Son zamanlarda yüz tanıma sistemleri, adli bilişim ve güvenlik sistemlerinde çok fazla ilgi görmeye başlayan konulardan birisidir. Bu sistemin genel amacı farklı iki yüz görüntüsünün aynı kişiyi temsil edip etmediğini anlamaktır. Sınıflandırılmamış görüntüler üzerinde işlem yapıldığında bu süreç zorlaşmaktadır (Cao ve ark., 2013). Karmaşık arka planlar, ışık değeri, poz açısı ve ifade farklılıkları gibi tanıma zorlukları yaratan durumlarda farklı algoritmalar kullanılarak süreç kolaylaştırılmıştır. Ayrıca aynı aile üyeleri arasındaki benzerlikten kaynaklanan hatalar da olabilmektedir. Bu sorunun çözümü için pek çok çalışma mevcut olup, ilgili literatür tarandığında bu zorlukları aşma adına yapılan çalışmaların bazıları aşağıda verilmiştir.

Kute ve ark. (2020), yaptıkları çalışmada yüz ile yüz organları arasındaki ilişkiyi incelemişlerdir. Face94 veri seti üzerinde dudaktan yüze ve burundan yüze ulaşmaya çalışmışlardır. Testlerde ESA, YGH ve Gabor yöntemlerini kullanmışlardır. Testler sonucunda ESA'nın, %91 başarı oranı ile diğer yöntemlere göre daha üstün başarı elde

ettiklerini ayrıca dudak ve burun yapısının kişileri ayırt etmede kullanılabileceğini belirtmişlerdir.

Talandova ve ark. (2016), 3D yüz okuyucusu kullanarak üç aileden elde ettikleri görüntüleri Biometric Device Broadway 3D aygıtı kullanarak aile içi benzerlik bulmaya çalışmışlardır. Aygıtın kullandığı Turnstile Enrolment Application (TEA) yazılımı ile bu benzerlikleri elde edip karşılaştırmışlardır. Test sonuçlarına göre birinci derece akrabalıklarda benzerlik oranının yüksek, birinci dereceden daha düşük akrabalık derecelerinde ise benzerlik oranının az olduğunu belirtmişlerdir.

Dandekar ve Nimbarte (2014), akrabalık derecelerini belirlemek için yaptıkları çalışmada, Familiy 101, Kinface ve kendi oluşturdukları veri setlerini ve yöntem olarak da Yerel İkili Örüntüler yöntemini kullanmışlardır. Testler sonucunda kendi veri setleriyle %92 benzerlik oranını elde ettiklerini belirtmişler ve birinci derece yakınları başarılı bir şekilde sınıflandırabilmişlerdir.

Cao ve ark. (2013), benzer yüzleri bulma adına yaptıkları çalışmada, Labeled Faces in the Wild (LFW) veri seti üzerinde kendi geliştirdikleri algoritma ile testler gerçekleştirmişlerdir. Bu testler sonucunda kendi geliştirdikleri algoritma ile %89,73 başarı elde etmişler ve geliştirdikleri sistem ile aranan yüze yakın yüzleri listeleyebilmişlerdir.

Ghahramani ve ark. (2014), yaptıkları çalışmada görüntülerden yola çıkarak aile sınıflandırması yapmışlardır. Testler için kendi oluşturdukları veri setini kullanmışlardır. Kendi algoritmaları ile sınıflamada %77, Liner SVM ile de %76 başarıya ulaştıklarını belirtmişlerdir.

Dehshibi ve ark. (2012), yaptıkları çalışmada dört aileden oluşan ve bu ailelerdeki bireylerin farklı açılardan çekilen görüntüleri üzerinde Yerel Gabor İkili Desen Histogram Dizisi yöntemi kullanarak aile içi yüz benzerliklerini incelemişlerdir. Testler sonucunda aile için yüz benzerliğinin %40,54 olduğunu belirtmişlerdir. En çok benzerlik gösteren organın burun, en az benzerlik gösteren organın ise gözler ve kaşlar olduğunu belirtmişlerdir.

Günümüz yüz tanıma sistemleri, güvenlik güçlerinin videolardaki ya da görüntülerdeki yüzleri tespit ederek şüphelilere ait birçok verinin elde edilmesini sağlamaktadır. Güvenlik birimleri bazı durumlarda şüphelilere ulaşmakta zorluk yaşayabilir bu gibi durumlarda şüpheliyi gören kişilerden destek alarak adli tıp

ressamları ya da bazı yazılımlar ile robot resimler oluştururlar. Bu robot resimleri kendi veri tabanındaki görüntüler içerisinde arayarak suçlulara ulaşmaya çalışırlar (Liu ve ark, 2018). Jain ve ark. (2012), robot resimleri eşleştirmede temelde yaşanan üç sorunun olduğunu bunlardan ilkinin yüz yaşlanmasından dolayı yaşanan sıkıntılar, ikincisinin; yüzdeki izlerden kaynaklı sorunlar, sonuncusunu ise robot resimlerin gerçek yüzlerle eşleştirme zorlukları olarak belirtmişlerdir. Bu sorunların çözümü için de kaliteli veri tabanlarının oluşturulması ve gerçek görüntülere mutlaka bir ön işlem uygulanarak iyileştirme yapılması gerektiğini belirtmişlerdir. Bu bilgiler ışığında, ilgili literatür taraması yapılarak robot resim oluşturma ve eşleştirme adına yapılan birkaç çalışma aşağıda listelenmiştir.

Liu ve ark. (2018), geliştirdikleri algoritma ile yüz görüntüleri ve bu görüntülere ait robot resimlerin eşleştirilmesini sağlamışlardır. Çalışmalarını üç farklı veri setinde (E-PRIP, PRIP-VSGC ve UoM-SGFS) test edip, geliştirdikleri algoritma ile, sırasıyla %70,1, %61,6 ve %86,9 başarılı eşleştirme değerleri elde etmişlerdir.

Iranmanesh ve ark. (2018), geliştirdikleri sistemde yüz resimleri ile robot resimleri eşleştirmek için Öznitelik Destekli Derin Evrişimli Sinir Ağı kullanmışlardır. Sistemin testlerini E-PRIP ve IIIT-D veri setleri üzerinde gerçekleştirerek %76,4 başarılı eşleştirme değeri elde etmişlerdir.

Wan ve Panetta (2016), yaptıkları çalışmada yüz resimleri ile robot resimleri eşleştirmek için bir sistem geliştirmişlerdir. Bu sistemde özellik çıkarıcı olarak Yerel İkili Model, sınıflandırıcı olarak da ki-kare uzaklığını kullanmışlardır. CUHCK, FERET ve TUPIS veri setlerinde testlerini gerçekleştirmişlerdir. Test sonuçlarında en yüksek başarıyı TUPIS veri setinde %88,5 olarak elde etmişlerdir.

Mittal ve ark. (2014), dijital resimlerle robot resimleri eşleştirme üzerine bir algoritma önermişlerdir. Önerilen algoritmada, CMU Multi-PIE veri tabanından 50.000 görüntü aracılığıyla oluşturulan SSD tabanlı bir sözlük kullanmışlardır. PRIP veri setini üzerinde önerdikleri algoritmayı kullanarak yaptıkları testler sonucunda %60 başarı sağladıklarını belirtmişlerdir.

Han ve ark. (2013), yüz görüntüleri ile robot resmin eşleştirilebilmesi adına önerdikleri sistemde çok ölçekli yerel ikili desen yöntemini kullanmışlardır. Veri seti olarak PCSO ve MEDS-II veri setlerini kullanmışlardır. Bireylerin yüz görüntülerindeki

yüz organlarının benzerliğinden yararlanarak doğru eşleştirme başarısının PCSO veri setinde %73,2 ve MED-II veri setinde ise %92,7 olduğunu belirtmişlerdir.

Klum ve ark. (2013), yaptıkları çalışmada yüz görüntüsü ile robot resmi eşleştirme yöntemlerinde adli eskizlerin ve kompozit eskizlerin etkilerini karşılaştırmışlardır. Kendi veri setleri üzerinde yaptıkları testler sonucunda kompozit eskizlerin Holistik yöntemlerde (bilgisayar yazılımları) %94,67 başarı sağladığını belirtmişlerdir.

Zhang ve ark. (2011), yaptıkları çalışmada bireylerin yüz görüntülerinden robot resimlerini eşleştirebilen bir sistem önermişlerdir. Sistemin testlerini FERET veri setinde yapmışlardır. Özellik çıkarıcı olarak kendi algoritmalarını, sınıflandırıcı olarak da PCA kullanarak %98,70 başarı oranı ile sınıflandırma yapmışlardır.



### **3. MATERYAL VE YÖNTEM**

#### **3.1. Materyal**

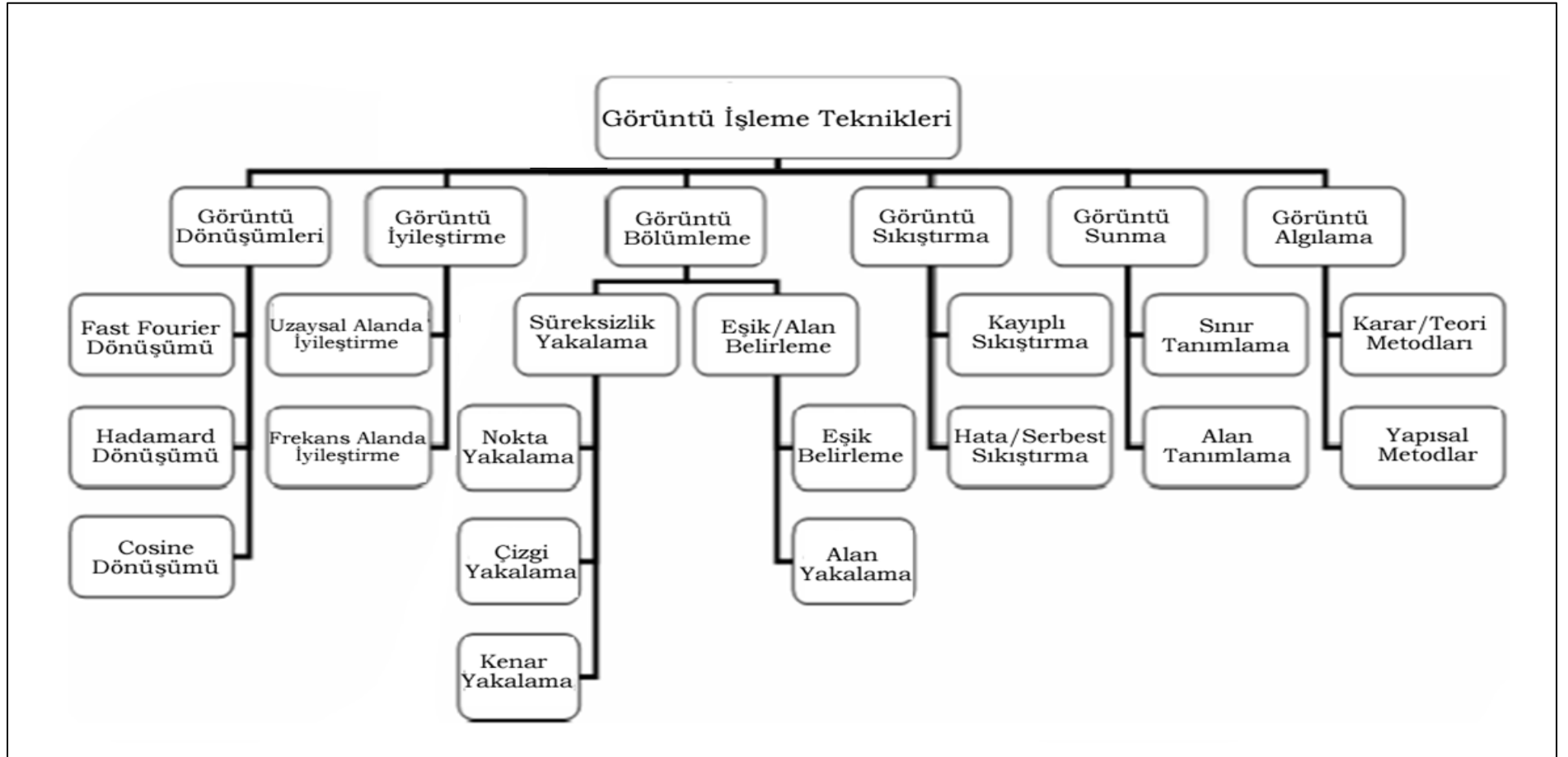
##### **3.1.1. Görüntü işleme**

Görüntü işleme, daha önceden ölçülüp elektronik ortama kaydedilmiş verilerin bilgisayar veya bazı yazılımlarla çalışmanın amacına göre değiştirilmesine yönelik yapılan bilgisayar çalışmasıdır. Bir başka deyişle kaydedilmiş görüntüleri işleyerek (çeşitli görüntü filtreleri uygulayarak) değiştirmek veya iyileştirmek anlamına gelir (Yılmaz, 2016; Karakoç, 2012).

##### **3.1.1.1. Görüntü işleme teknikleri**

Görüntüler üzerinde işlem yapılabilmesi için görüntülerin sayısal görüntü biçimine dönüştürülmesi gerekir. Bu sayısal görüntü biçimi, görüntünün satır ve sütunlara dönüştürülmüş halidir. Satır ve sütunların kesiştiği noktaların her birine piksel denir. Şekil 3.1’de görüntü işleme teknikleri gösterilmektedir.

- Görüntü dönüşümleri; farklı açılardan çekilmiş görüntülerin işlenerek önden çekilmiş gibi gösterme tekniğidir.
- Görüntü iyileştirme; görüntünün niteliğini optimum düzeyde iyileştirme sürecidir.
- Görüntü onarma; bozulmuş görüntünün orijinal haline getirme işlemidir.
- Görüntü bölümleme; görüntü içindeki nesnelere göre görüntünün bölünmesi işlemidir. Bir başka deyişle görüntüdeki nesnelerin sınıflandırılması işlemidir.
- Görüntü sıkıştırma; sıkıştırılan görüntünün hafızada kapladığı alanın yönetilebilmesi için yapılan işlemdir.
- Görüntü sunma; görüntüdeki nesnelerin daha belirgin bir hale getirme işlemidir.
- Görüntü algılama; herhangi bir fiziksel temas olmaksızın cisimlerin gözlenmesi ve bu gözleme dayalı verilerin kaydedilmesi işlemidir.



Şekil 3.1 Görüntü İşleme Teknikleri (Onat, 2008).

### 3.1.1.2. Görüntü işleminin uygulama alanları

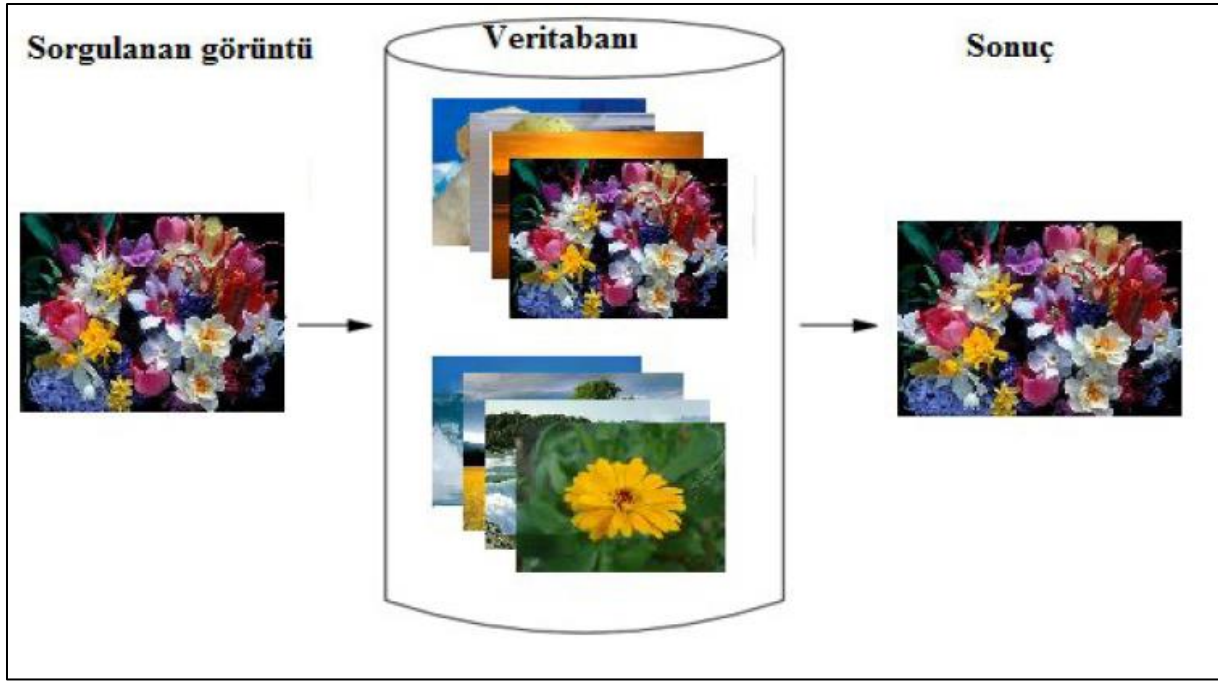
Görüntü işleme uygulamaları günlük yaşantımızın birçok alanında kullanılmakta ve hayatı kolaylaştırma adına çok sayıda imkân sunmaktadır. Bu alanlardan bazıları:

- Tıbbi görüntü analizi
- Astronomi
- Endüstriyel uygulamalar ve mühendislik
- Belge işleme
- Güvenlik ve savunma sanayi
- Bankacılık ve ticaret
- Uzaktan algılama sistemleri
- Gözetim ve izleme
- Robotik sistemler
- Otomatik navigasyon sistemleri
- Spor müsabakaları
- Sanatsal faaliyetler

### 3.1.2. İTGE (İçerik Tabanlı Görüntü Erişim) sistemleri

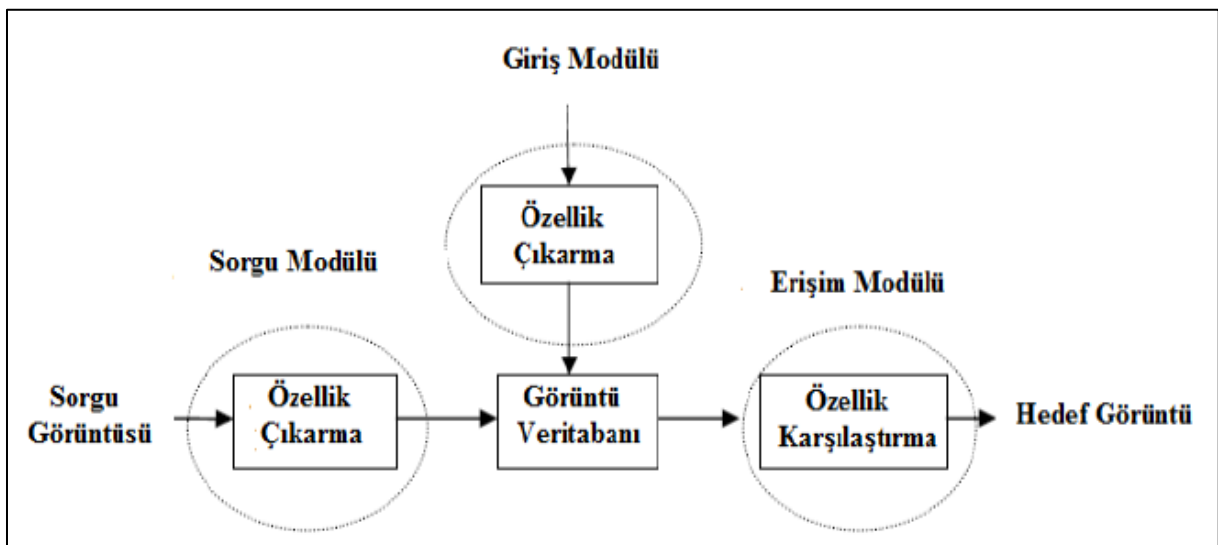
İTGE sistemlerin tanıtımı 1990 yılında Niblack ve arkadaşları tarafından yapılmıştır. İTGE sistemlerindeki temel amaç bir imgeyi, imge veri tabanında arayarak ona en yakın (benzer) imgeleri bulmaktır. Bu işlemde herhangi bir metinsel bilgi kullanılmayıp, arama işlemi tamamen resim içeriğiyle yapılmaktadır. İTGE sistemleri; görüntü işleme, yapay görme, istatistik, sinyal işleme, desen tanıma gibi süreçleri kapsamaktadır. Bu sistemler imgenin düşük düzey özelliklerini kullanılarak arama yapmaktadır (Karhan, 2014; Jahangiri, 2015).

İTGE sistemlerindeki imgeye göre arama kavramı ilk olarak IBM QBIC tarafından ortaya atılmıştır. Ayrıca İTGE sistemleri alt yapısını kullanan Virage sistemleri ESA tarafından kullanılmaktadır. Bunun dışında Candid, Photobook ve Netra da bu sistemleri kullanmaktadır (Alpkoçak, 2009).



Şekil 3.2 Temel bir İTGE sistemi (Karhan, 2015).

İTGE sistemlerinin çoğu Şekil 3.2 gösterildiği gibi benzer (temel) bir yapıya sahiptir. Bu sistemler imgenin içeriğini görsel tanımlayıcılar kullanarak açıklamaya çalışmaktadırlar. Bu görsel tanımlayıcılar renk, doku ve şekildir. Tanımlanan bu özellikler ile imge veri tabanında kayıtlı olan özellikler çeşitli algoritmalar ile kıyaslanarak aralarında ilişki olup olmadığı belirlenmektedir ve sonuç bilgisi geri gönderilmektedir (Şekil 3.3) (Karhan, 2015; Alpkoçak, 2009).

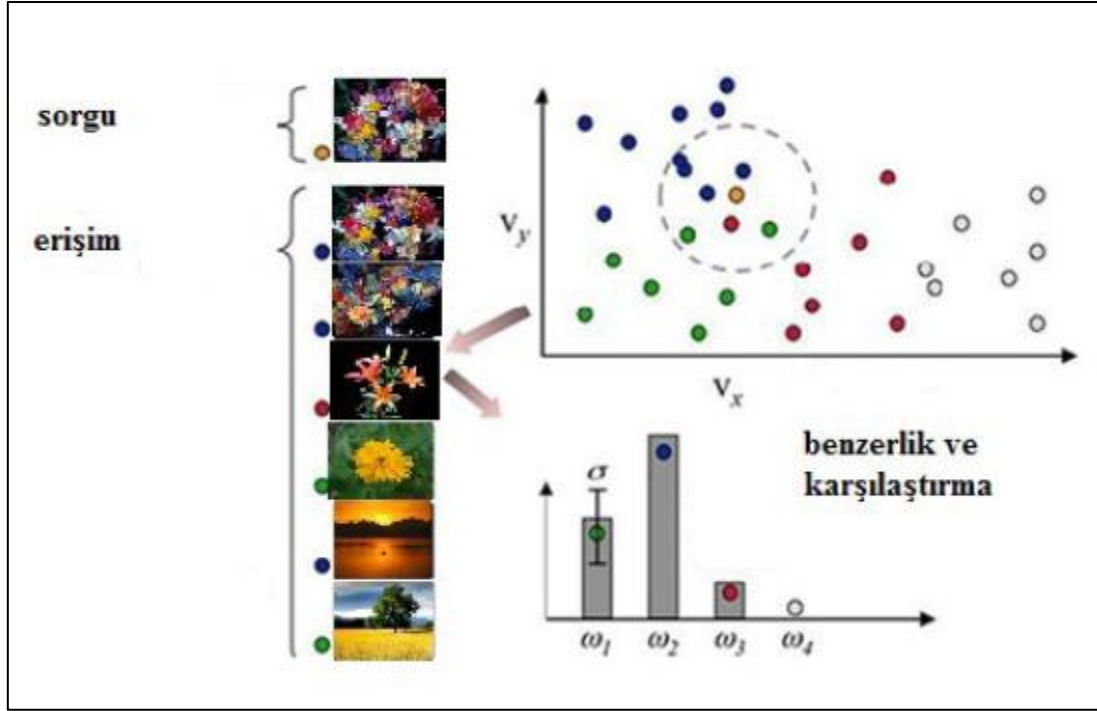


Şekil 3.3 İTGE sistemi blok diyagramı (Khokher, 2011).

İTGE sistemlerinin temelinde özellik çıkartma vardır. Özellik çıkarımı, imgenin renk, doku ve şekil gibi özelliklerinin sisteme aktarılıp sayısallaştırılmasıyla başlar. Bu sistemlerde algoritmaların çıkardığı özelliklerin yanında kullanıcılarında yorumları da sisteme dahil edilebilir (Karhan, 2015).

- *Doku tabanlı:* Doku; bir imgenin düzgünlüğü, düzenliliği ve kalitesi gibi özelliklerini tanımlamak için kullanılmaktadır. Ağaç, bulut, saç ve gökyüzü gibi nesneleri ayırt etmek için doku analizi kullanılabilir. Doku bir yüzey özelliğidir ve bu dokunun tanımı onun tüm komşu noktalarının da aynı renk düzeylerini içermesi gerekmektedir. Doku analizi için eş oluşum matrisi, ayık kosinüs dönüşümü, ayrık dalgacık dönüşümü ve iki boyutlu histogram uygun yöntemlerdir (Karhan, 2014; Jahangırı, 2015).
- *Renk tabanlı:* Renk, İTGE sistemlerinde en çok kullanılan özellik tanımlama seçeneğidir. Renklerin temsili için histogram kullanılmaktadır. Histogram basit bir şekilde imgedeki her rengin piksel oranını açıklamaktadır. İmgeden çıkartılan görüntü histogramı, sistemde kayıtlı olan histogramlar ile karşılaştırılarak seçim yapılmaktadır (Jahangırı, 2015).
- *Şekil tabanlı:* Şekil, imge sınıflamada ve indekslemede kullanılan önemli bir özellik tanımlayıcısıdır. Şekil, özellik çıkarımı için temel tanımlayıcılar, fourier tanımlayıcı, şablon eşleştirme gibi farklı algoritmalar kullanılmaktadır. Bu algoritmalar görüntü segmentasyonu, görüntü sınır belirlemesi ve farklı filtreler uygulayarak özellik çıkarımı yapmaktadır (Jahangırı, 2015).

İTGE sistemlerinde benzerlik ölçümü, sistemin sınıflandırma yapabilmesi için önemli bir kavramdır. Bu benzerlik ölçümünü de elde ettiği renk, doku ve şekil özellik bilgilerine göre yapmaktadır. İTGE sistemlerinde sınıflandırma öznitelikler arasındaki benzerlik ölçüsüne göre yapılmaktadır. Öznitelikler arasındaki benzerliği hesaplamak için uzaklık ölçüm yöntemleri kullanılmaktadır. Şekil 3.4'te görüldüğü gibi imgeler özellikleri ile temsil edilir ve aranan özelliğe en yakın özelliği olan imgeler geri döndürülmektedir (Karhan, 2014).



Şekil 3.4 İTGE sistemlerde benzerlik karşılaştırması ve sınıflandırması (Karhan, 2014).

### 3.1.3. Makine öğrenmesi

İnsanların teknolojiye olan ilgisinin ve erişiminin tavan yaptığı 1990'lı yıllarda işlenmesi gereken çok miktarda ham veri oluşmuştur. Bu ham verilerin toplanması, saklanması ve işlenmesi noktasında veri madenciliği ve programlanmış makineler kullanılmaktadır. Makine öğrenmesi, elde edilen ham veriler üzerinde matematiksel ve istatistiksel işlemler yaparak gerçekleşmemiş olaylar hakkında tahminleme ve modelleme yapmasını sağlayan yapay zekânın bir alanıdır (Şahinarslan, 2019; Türker, 2019).

Makine öğrenmesi öğrenme yöntemine göre iki başlıkta incelenir ancak bazı kaynaklarda bu sayının arttığı görülebilmektedir (Mohri ve Rostamizadeh, 2012). Bunlar;

- *Denetimli öğrenme (supervised learning)*: Bu öğrenmede denetmen yardımıyla veriler ve çıktıların bulunduğu eğitim veri seti makineye verilerek makinenin öğrenmesi sağlanmaktadır. Makine bu veriler arasındaki ilişkiyi çözdükten sonra bu ilişkiyi kullanarak hiç tanımadığı bir örnek için varsayımda bulunmaktadır (Şahinarslan, 2019).
- *Denetimsiz öğrenme (unsupervised learning)*: Bu öğrenmede denetçi bulunmamaktadır. Verilerin tamamı makineye girdi olarak verilmektedir. Makine veriler arasındaki ilişkiyi çözerek kümeleme ya da sınıflandırma yapmaktadır (Mohri ve Rostamizadeh, 2012).

- *Yarı Denetimli Öğrenme (Semi-Supervised Learning)*: Bu öğrenmede verinin az bir kısmı girdi ve çıktı değerlerinden oluşan etiketlenmiş veridir. Makine hem etiketlenmiş veriyi hem de etiketlenmemiş veriyi inceleyerek öğrenme sürecini gerçekleştirmektedir. Regrasyon, sınıflandırma ve sıralama gibi farklı birçok problemi çözmek için bu öğrenme kullanılabilir (Mohri ve Rostamizadeh, 2012).
- *Transdüktif Akıl Yürütme (Transductive Inference)*: Bu öğrenmede de makineye etiketlenmiş ve etiketlenmemiş veri verilmektedir. Makine etiketlenmiş verideki çıktı değerlerinden yararlanarak etiketlenmemiş verinin çıktı değerlerini tahmin etmektedir. Bu öğrenmede sisteme sonradan veri eklenmez (Şahinarslan, 2019).
- *Pekiştirmeli Öğrenme (Reinforcement Learning)*: Bu öğrenmede eğitim ve test aşaması birlikte yürütülmektedir. Veri içerisinde çıktı değerleri bulunmaz. Denetçi çıktı değerlerini, girdi değerlerine göre iyi veya kötü şeklinde değerlendirerek modeli makineye öğretmeye çalışır. Bu öğrenmede süreç daha önemlidir (Şahinarslan, 2019).
- *Aktif Öğrenme (Active Learning)*: Bu öğrenmede oldukça zor veya zaman gerektiren etiketsiz verinin etiketlenmesi sürecini ortadan kaldırıp, etiketlenmemiş veri içerisinde kendisine en fazla bilgiyi verecek olanı seçerek öğrenme sürecine başlamaktadır (Koçyiğit, 2016).

### 3.1.3.1. Makine öğrenmesi algoritmaları

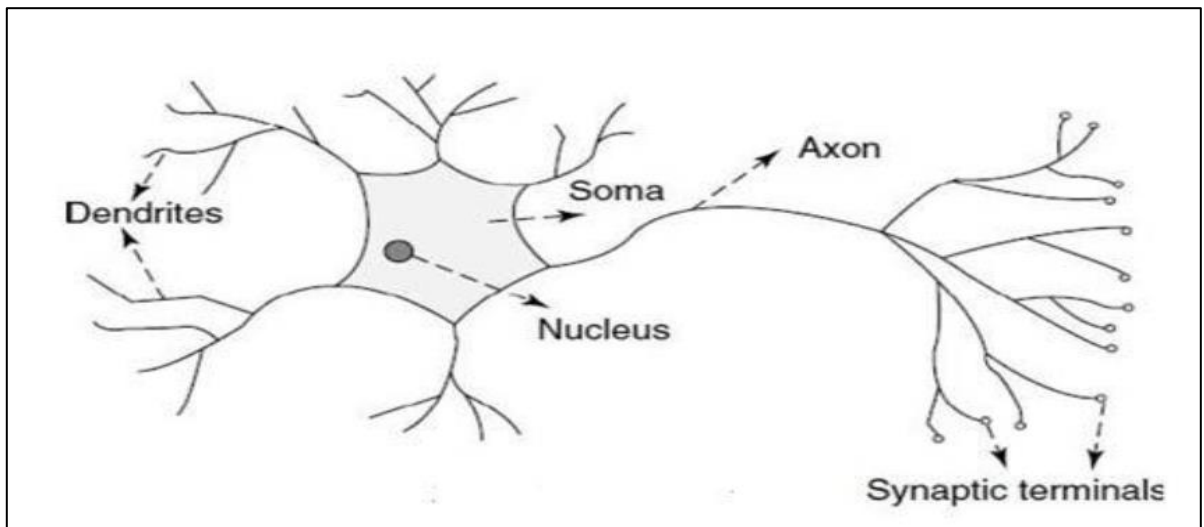
Makine öğrenmesi algoritmaları, toplanan verilerle ne yapılacağına karar vermeye yardımcı olmaktadır. En uygun makine öğrenmesi algoritmasının seçilmesi verinin boyutuna, kalitesine ve çeşitliliğine bağlıdır. Birçok amaç için geliştirilmiş algoritmalar vardır bunlardan en yaygın olanları (Khorram, 2019):

- *Doğrusal Regrasyon (Linear Regression)*: En basit makine öğrenmesi algoritmasıdır. İki sürekli değişken arasındaki ilişkinin anlaşılması için kullanılır. Temel amaç neden ve sonuç arasında lineer bir ilişkinin olup olmadığını belirlemektir (Anonim, 2020d).
- *DVM*: Sınıfları farklı olan verileri ayırmak için kullanılan bir algoritmadır. Bu sınıflandırma işlemini için her grubun üyelerine eşit uzaklıkta çizgiler çizerek yapmaktadır (Şahinarslan, 2019).
- *Karar Ağaçları (Decision Trees)*: Basit kararlar ile sınıfları belli olan verinin parçalara ayrılması sürecidir. Her parçalama işleminde birbirine benzeyen veriler gruplanır ve böylece tümevarım yöntemiyle veriler sınıflandırılmış olur (Türker, 2019).

- **YSA:** İnsan beyninin biyolojik sinir yapısını taklit ederek, kendi kendine öğrenme, tahminleme yapabilme ve hatırlama yoluyla yeni bilgiler üretebilen yazılımların genel adıdır. Bir başka deyişle biyolojik sinir ağlarının yapısından ve işlevselliğinden ilham alarak çalışan bir algoritmadır. Sorunları çözmek için birbirine bağlı çok sayıda işlem ögesi içermektedir (Tükel, 2019; Demir ve ark., 2019).
- **KYK:** Bu algoritma sınıfı belirlenecek olan örneğin verilen K değerince en yakın komşuları seçilir ve bu komşuların çoğunluğu hangi sınıftaysa bu örnek de o sınıfa dahil edilmektedir (Türker, 2019).
- **Naive Bayes Algoritması:** Bu algoritmada makineye sunulan etiketlenmiş veriler üzerinde olasılık işlemleri yürütür ve sisteme sunulan yeni verileri bu olasılık değerlerine göre sınıflandırır (Anonim, 2020e).
- **RO:** Bu algoritmada birden fazla karar ağacının kullanılmasıyla sınıflandırma işlemini gerçekleştirilmektedir. Veri kümesinden rastgele örnekler seçilerek N adet karar ağacı üretilir ve bu karar ağaçlarından yapılan tahminlerin ortalaması ya da çoğunluğuyla daha doğru tahmine ulaşılır (Türker, 2019).

### 3.1.4. DSA (Derin Sinir Ağları)

Sinir ağları birbiriyle iletişim halinde olan çok sayıda nöronlardan oluşmaktadır. Nöronlarda bulunan denritritler elektrik sinyalinin alma görevindedir. Alınan bu sinyaller somada işlenir. Üretilen çıktı akson yardımıyla denritritteki terminale iletilir (Tiryaki, 2019). Bu terminal diğer bir nöron ile bağlantıyı sağlar. Şekil 3.5'te nöron yapısı gösterilmektedir.



Şekil 3.5 Nöron yapısı.



1957 yılında ilk kez Frank Rosenblatt tarafından tanıtılan, YSA'nın en küçük parçası olarak bilinen perceptron, "Eş. 3.1"deki gibi lineer bir fonksiyonla ifade edilmektedir (Şekil 3.6) ( Ayyüce, 2020).

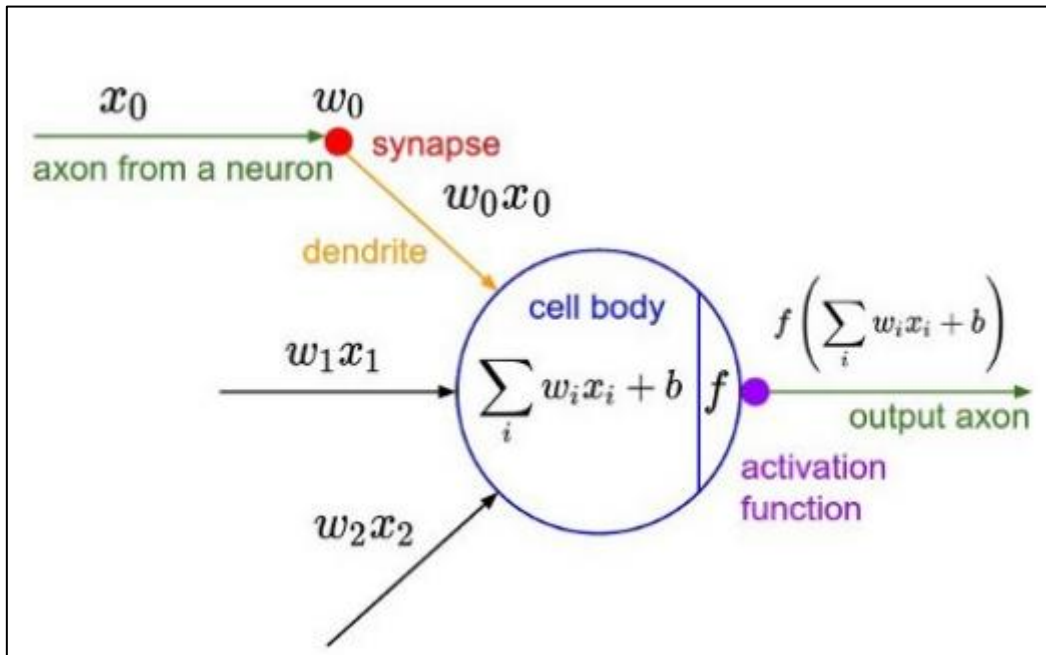
$$y = w * x + b \quad (3.1)$$

y: girdiye ait skoru değeri

x: bağımsız değişken, girdi

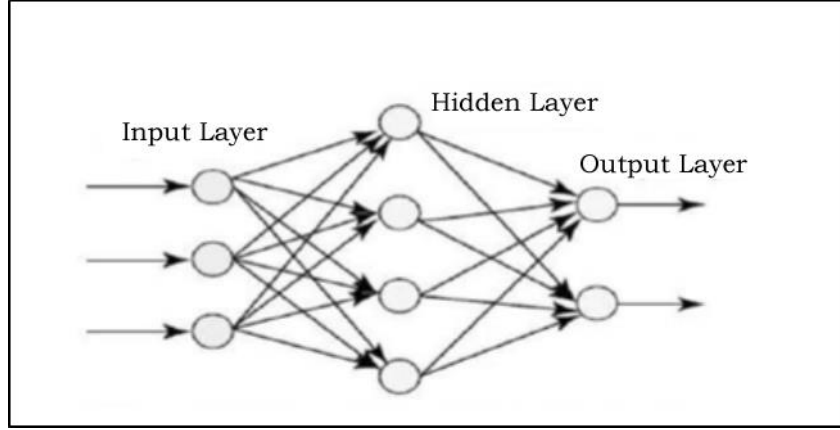
w: ağırlık parametresi

b: bias değeri



Şekil 3.6 İnsandaki bir sinir hücresinin matematiksel modeli ise şu şekilde gösterilebilir (Ayyüce, 2020).

YSA insan beyninin sinir yapısına benzerdir. Burada da yapay nöronlar vardır. Yapay nöronlar üzerinde temel olan üç işlev gerçekleştirilir: çarpma, toplama ve aktivasyon. YSA'lar birden fazla yapay nöronun birleşimiyle oluşmaktadır. Nöronlara gelen bilgiler sinir ağında işlenip verilen problem çözülmektedir (Tiryaki, 2019). YSA'da veriler giriş katmanıyla alınır, gizli katmanda işlenir ve çıktı katmanına gönderilir (Şekil 3.7).

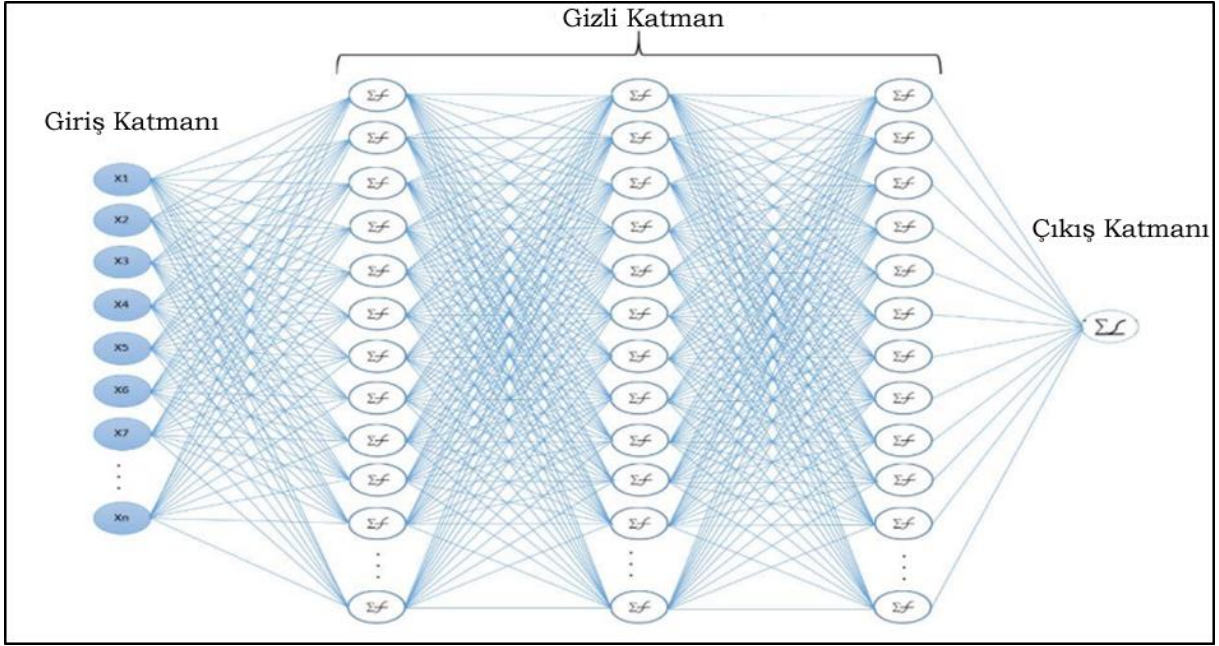


Şekil 3.7 Yapay sinir ağı (Tiryaki, 2019).

Veri hacminin, değişken sayısının ve değişkenler arasındaki ilişkinin artması YSA'ya olan ilgiyi de arttırmıştır. YSA'nın yapısal olarak daha geniş ve karmaşık haline DSA denilmektedir. DSA'da gizli katman sayısı ikiden fazladır ve bu katmanlardaki nöron sayısı da YSA'ya oranla daha fazladır (Anonim, 2020g).

Özellik çıkarımı ve algılama işlemleri için geliştirilmiş olan DSA, Makine öğrenmesinin bir alt dalıdır.

DSA'da doğrusal olmayan çoklu katmanlar vardır. Her katman birbiriyle ilişkilidir. DSA'ya bir görüntü verildiğinde köşeler veya kenarlar gibi özellikleri kullanarak sınıflandırma yapıp öğrenebilir. Bu öğrenme sürecinde öğrenilecek nesnelerin karmaşıklığını artıran katman yapısını kullanılmaktadır (Kurbanov, 2018). Şekil 3.8'de görüldüğü üzere katmanlar kendisinden önce gelen katmandan çıktı verisini alır, dönüşüm uygular ve bir sonuç üreterek bir sonraki katmana iletir (Tiryaki, 2019).



Şekil 3.8 Üç gizli katmandan oluşan derin sinir ağı örneği iletir (Tiryaki, 2019).

Video analizi, doğal dil işleme, el yazısı tanıma, ses işleme, gen analizi, konuşma tanıma, biyoenformatik, bilgisayarlı görme, tıbbi alanlar gibi birçok alanda derin öğrenme mimarileri kullanılmaktadır (Kurbanov, 2018;Tiryaki, 2019).

DSA algoritmaları veri setindeki tüm veriyi eğitim ve test veri setleri olmak üzere ikiye ayırır. Eğitim veri seti, sinir ağındaki ağırlıkları tahmin etmekle görevlidir. Test veri setinin görevi ise bu ağırlık değerlerinin doğruluğunu test etmektir. DSA ileri besleme yöntemi ve geri yayılım algoritması ile çalışmaktadır. Geri yayılım algoritması ilk çalıştırıldığında her nörona bir ağırlık değeri atanır. Daha sonra değerler atanan ağırlıklarla çarpılıp toplanır ve sonuca bias değeri eklenerek aktivasyon fonksiyonu aracılığıyla sonraki katmanın nöronlarına giriş değeri olarak gönderilmektedir (Eş. 3.1). Çıkış işlemine kadar benzer işlemler sürer. Son aşamada çıktı değeri ile gerçek değer kıyaslanarak hata değeri bulunur (Eş. 3.3) (Ser, 2019).

$$\hat{y}_i = f_2 \left\{ \sum_{j=1}^R w_{kj}^{(2)} \left[ f_1 \left( \sum_{j=1}^R w_{kj}^{(1)} x_j + b_k^{(1)} \right) \right] + b_k^{(2)} \right\} \quad (3.2)$$

$f_1$  ve  $f_2$  : Aktivasyon fonksiyon

$w_{kj}$  : Ağırlıklar

$x_j$  : Girdiler

$b_k$  : Bias

$$E_0 = \frac{1}{2} \sum_{i=1}^p (y_i + \hat{y}_i)^2 \quad (3.3)$$

$\hat{y}_i$  : Tahmini değerler

$y_i$  : Gerçek değerler

Sonraki aşamada bulunan hata değeri geriye doğru tüm katmanlara ağıdaki ağırlıklara göre dağıtılmaktadır. Ağırlıkların dağıtımı için optimizasyon yöntemi kullanılmaktadır (Eş. 3.4).

$$\frac{\partial E_p(t)}{\partial w_{kj}^2(t)} = \frac{\partial E_p(t)}{\partial n_k^2(t)} \frac{\partial n_k^2(t)}{\partial w_{kj}^2(t)} \quad (3.4)$$

$$\frac{\partial E_p(t)}{\partial b_k^2(t)} = \frac{\partial E_p(t)}{\partial n_k^2(t)} \frac{\partial n_k^2(t)}{\partial b_k^2(t)} \quad (3.5)$$

“Eş. 3.5” hatanın ağırlıklara göre kısmi türevlerini ifade etmektedir. Güncellenmiş ağırlık değeri “Eş. 3.6” ile hesaplanır.

$$w^{l+1} = w^l - \alpha \frac{\partial E_p}{\partial w^l} \quad (3.6)$$

$w^{l+1}$  : Güncellenmiş ağırlık değeri

$w^l$  : Bir önceki iterasyon ağırlığı

$\alpha$  : Öğrenme oranı

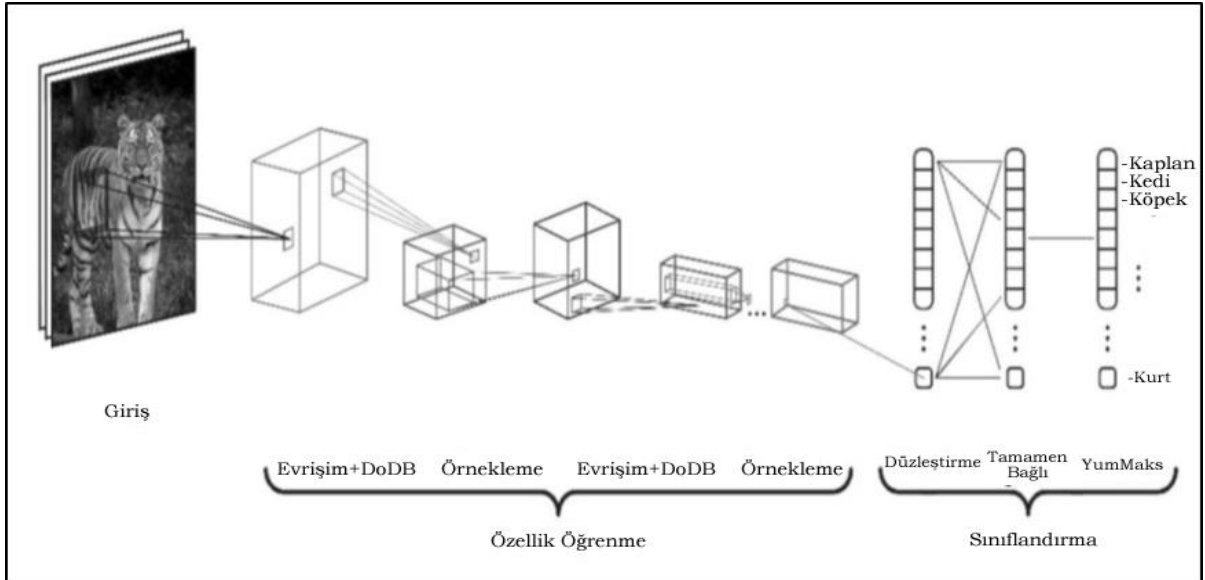
### 3.1.5. ESA (Evrişimsel Sinir Ağı)

ESA daha çok görüntü içeren verilerin sınıflandırılmasında kullanılan YSA tabanlı bir yöntemdir. ESA temelde, ham olarak verilen görüntünün hangi görüntü sınıfına ait olduğu ya da o görüntüyü tanımlayan en iyi sınıf olasılığının çıkartılması işlemini yapmaktadır. Bu işlev insanlar için doğumdan itibaren süre gelen bir öğrenmedir. İnsan bir görüntüye baktığında

farkında olmadan görüntünün ayırt edici özelliklerini çıkarır ve o nesneyi sınıflandırır (Akpınar, 2019).

ESA; gelen ham görüntünün, görüntünün boyutlarını ve 3 ana rengini ifade eden RGB (Red Green Blue) sayı dizisini oluşturmaktadır. Örneğin; 300x300x3 sayı dizisi, 300x300 boyutundaki renkli bir görüntünün temsili piksel değeri olmaktadır. Piksel değerleri ise 0 ile 255 arasında değer alarak o pikselin yoğunluğunu ifade etmektedir. Bu değerler ESA'nın görüntü sınıflandırma için kullanacağı girdi değerleridir (Akpınar, 2019).

ESA görüntüyü parçalara ayırarak her parçaya bir filtre uygulamaktadır. Her filtreleme sonrasında görüntü bir parça küçülmektedir. Sonra bu parçalar arasında ilişki kurarak tanımayla çalışmaktadır (Şekil 3.9).



Şekil 3.9 Evrişimsel sinir ağı (Sewak ve ark., 2018).

ESA; giriş, evrişim, düzleştirilmiş doğrusal birim, örnekleme, düzleştirme, tamamen bağlı katmanlarından oluşur.

### 3.1.5.1. Giriş katmanı

Ham olan görüntünün sisteme girdiği katmandır. Bu mimaride giriş verisinin boyutu önemlidir. Çünkü giriş verisinin boyutu ne kadar büyük olursa ağıın başarısı o derece artar ama buna bağlı olarak eğitim süresi de artar. Aynı şekilde düşük boyutlu giriş verisi ağıın başarısını ve eğitim süresini azaltmaktadır (İnik ve Ülker, 2017).

### 3.1.5.2. Evrişim katmanı

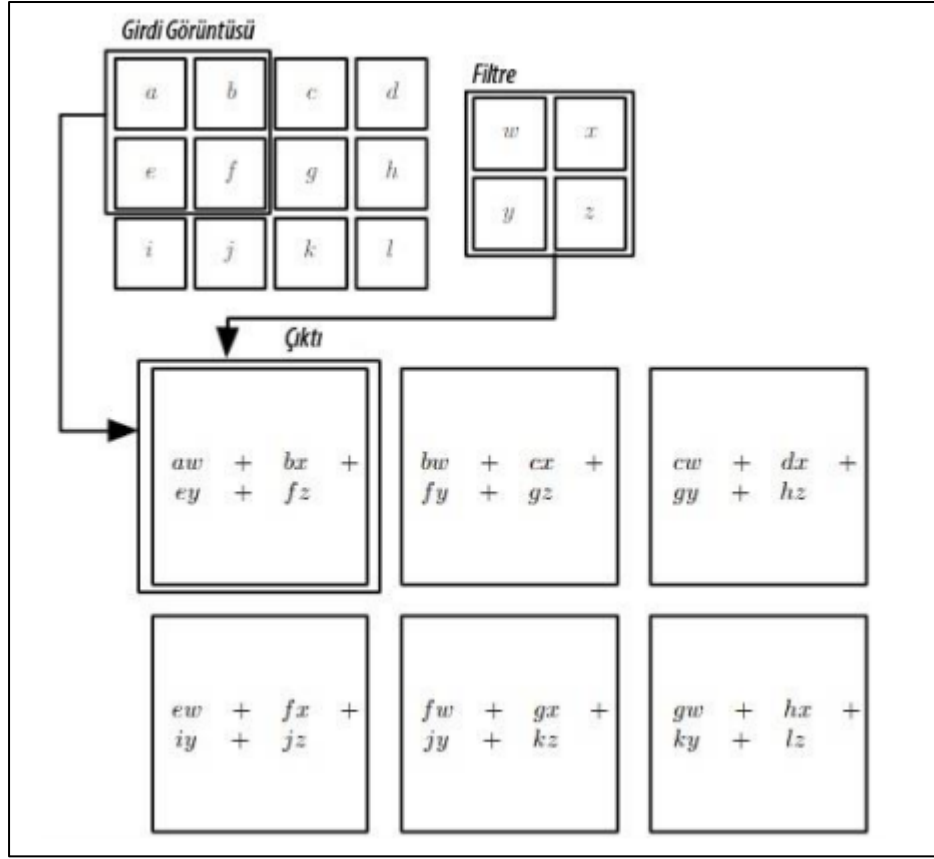
Bu katman konvolüsyon ya da dönüşüm katmanı olarak da adlandırılır. Bu katmanda veriden özellik çıkarma ve hesaplama işlemleri yapılır. Eğer ağa verilecek görüntü renkli ise evrişim katmanı; genişlik, yükseklik ve derinlik olmak üzere 3 kanaldan oluşur. Derinlik RGB değerlerinden meydana gelmektedir. Ayrık zamanlı evrişim işlemi “Eş. 3.7” ile (filtre  $w$ , girdi  $x$ , zaman  $t$ , ve  $s$  sonuç) ifade edilmektedir (Goodfellow ve ark., 2016).

$$s(t) = (x * w)(t) = \sum_{a=-\infty}^{\infty} x(a)w(t - a) \quad (3.7)$$

Eğer girdi olarak resim kullanılırsa evrişim işlemi “Eş. 3.8” ile ifade edilir.  $i$  ve  $j$  terimleri evrişim sonucunda elde edilecek olan yeni matrisin konumlarını göstermektedir (Goodfellow ve ark., 2016).

$$s(i,j) = (I * K)(i,j) = \sum_m \sum_n I(i,j)K(i - m, j - n) \quad (3.8)$$

Bu katmanda giriş görüntüsüne filtreler uygulanır. Örneğin filtre  $3 \times 3 \times 3$ 'lük bir filtre ise görüntünün sol üstünden başlayıp sağa doğru adım adım ilerleyerek tüm görüntüyü tarar ve bir özellik haritası oluşturur (Şekil 3.10). Bu harita orijinal boyuttan daha küçüktür ama mimari dolgu kullanarak görüntü boyutu eşitlenmektedir (Tiryaki, 2019; Akpınar, 2019).



Şekil 3.10 3x4 boyutta bir giriş görüntüsüne 2x2’lik filtrenin uygulandığı konvolüsyon işlemi (Goodfellow ve ark., 2016).

Görüntü üzerinde filtre kaydırma işlemi: örneğin  $n \times n$  boyutundaki bir görüntüye  $f \times f$  boyutunda bir filtre uygulanır ise ( $p$ ; dolgu miktarının değeri ve  $s$ ; kaydırma değeri) yeni görüntünün genişlik ve yükseklik değerleri (YGD) “Eş. 3.9”a göre hesaplanır. Yeni görüntünün derinlik değeri ise filtre sayısı olur.

$$YGD = \frac{n+2p-f}{s} + 1 \quad (3.9)$$

### 3.1.5.3. Düzleştirilmiş doğrusal birim katmanı

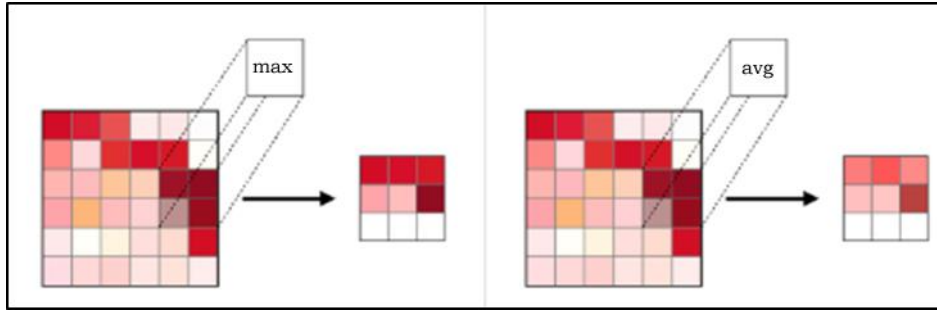
Bu katman evrişim katmanlarından sonra gelir ve nöronlara gelen değerlere karşılık üretilecek olan çıktıları belirler ve doğrusal olmayan bir fonksiyondur. Çeşitli aktivasyon fonksiyonları vardır. Bunlar Çizelge 3.1’de gösterilmektedir.

Çizelge 3.1 Aktivasyon fonksiyonları ve türevleri (Salman, 2018)

Aktivasyon Fonksiyonu	Matematiksel Denklemleri
RELU	$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases}, f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases}$
Sigmoid	$f(x) = \frac{1}{1 + e^x}, f'(x) = f(x)(1 - f(x))$
Tanh	$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}}, f'(x) = 1 - f(x)^2$

#### 3.1.5.4. Örnekleme katmanı

Bu katman evrişim katmanlarının arasına eklenerek görüntünün kayma boyutunu azaltır. Böylece hesaplamalar daha kısa sürede gerçekleşir (Akpınar, 2019). Ortalama ve maximum örnekleme türleri vardır (Şekil 3.11).



Şekil 3.11 Örnekleme katmanı (Amidi, 2020).

Görüntü üzerinde örnekleme işlemi: örneğin  $n \times n$  boyutundaki bir görüntüye örnekleme uygulanır ise ( $G_1$ ; Giriş görüntü boyutunun genişlik değeri,  $Y_1$ ; Giriş görüntü boyutunun yükseklik değeri,  $F$ ; filtrenin mekânsal boyut değeri ve  $s$ ; kaydırma değeri) yeni görüntünün genişlik ve yükseklik değerleri (YGD) aşağıdaki eşitliklere (Eş. 3.10, Eş. 3.11, Eş. 3.12, Eş. 3.13'e) göre hesaplanır. Yeni görüntünün derinlik değeri ise önceki derinlik ile aynı olur (İnik, 2017).

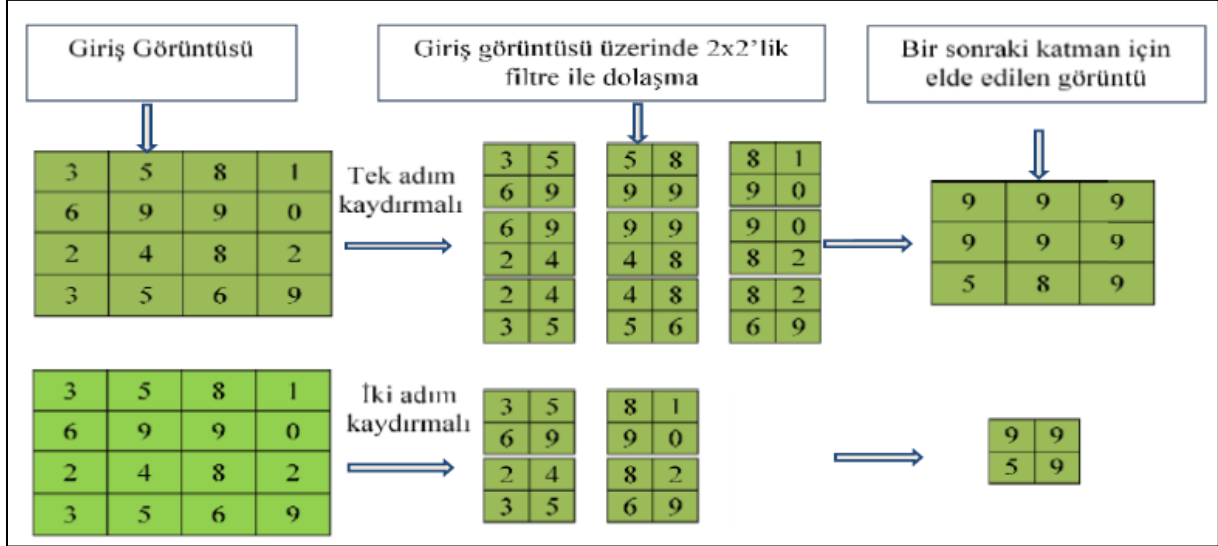
$$YGD = G_2 \times Y_2 \times D_2 \quad (3.10)$$

$$G_2 = (G_1 - F) / s + 1 \quad (3.11)$$



$$Y_2 = (Y_1 - F)/s + 1 \quad (3.12)$$

$$D_2 = D_1 \quad (3.13)$$



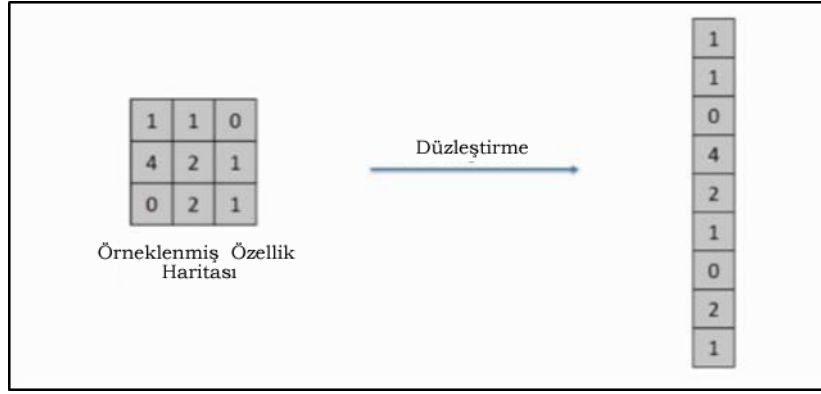
Şekil 3.12 5x5'lik giriş görüntüsüne 2x2 filtre ile bir ve iki adım kaymalı maksimum havuzlama işlemi.

Şekil 3.12'de gösterildiği gibi 2x2'lik bir filtre ile maximum havuzlama işlemi yapıldığında her 2x2'lik matris içinde "Eş. 3.14" kullanılarak yeni matris değerleri elde edilir.

$$\text{Maksimum} = (x_{0,0}, x_{0,1}, x_{1,0}, x_{1,1}) \quad (3.14)$$

### 3.1.5.5. Düzleştirme katmanı

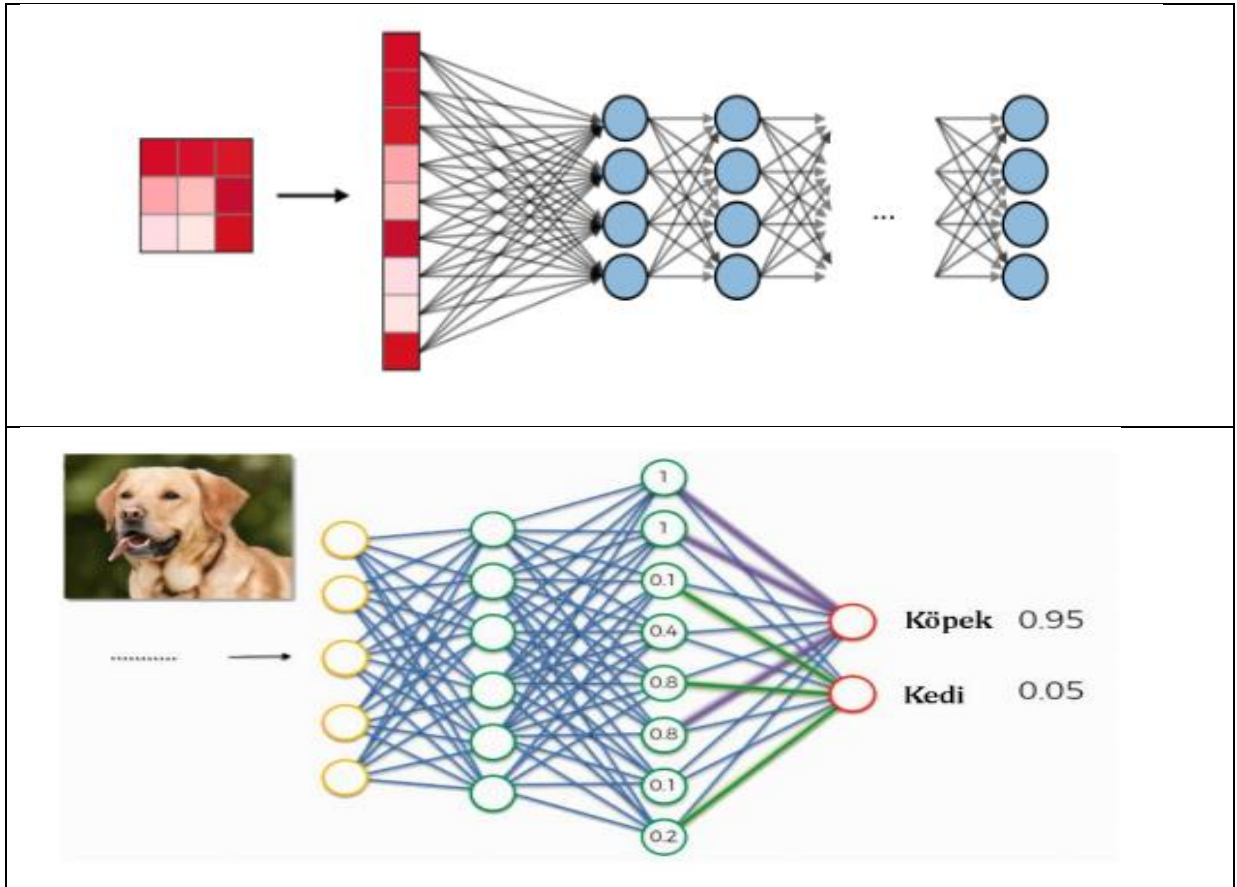
Bir sonraki katmana veri hazırlama işlemini yapmaktadır. Üst katmandan gelen matrisleri tek boyutlu diziye dönüştürme işlemini gerçekleştirir (Şekil 3.13).



Şekil 3.13 Düzleştirme katmanı (Akpınar, 2019).

### 3.1.5.6. Tamamen bağlı katman

Çıktı üretilmeden önceki katman olan tamamen bağlı katmanda sınıf belirleme yapılmaktadır. Hücreler üzerindeki ağırlık değerlerine göre görüntüyü en iyi tanımlayan ya da en yüksek değere sahip ağırlık, o görüntünün sınıfını belirlemektedir (Şekil 3.14).



Şekil 3.14 Tamamen bağlı katman (Amidi, 2020; Kadiroğlu, 2019).

### 3.1.6. YGH – DVM (Yönelimli Gradyan Histogramı-Destek Vektör Makineleri)

İmgelerden öznitelik çıkarmak için sıklıkla kullanılan bir özellik tanımlayıcısıdır. YGH tanımlayıcı tekniği, bir görüntü algılama penceresinin veya ilgilenilen bölgenin lokalize bölümlerinde gradyan oryantasyonunun oluşumlarını saymaktadır. YGH tanımlayıcısını diğer yöntemlerden farklı kılan en önemli özellik nesne veya şekil odaklı olmasıdır. YGH özellik çıkarıcısının çalışma adımları aşağıdaki gibidir (Lai, 2016; Singh, 2019):

#### 1. Adım: Ön işleme

İmgenin genişlik/yükseklik oranının 1:2 olarak ayarlanması gereklidir. Tercih edilen görüntü en/boy oranı genelde 64x128'dir. Bunun nedeni ise imgeden özellik çıkarmak için imgenin 8x8'lik ve 16x16'lık yamalara bölüneceğidir (Lai, 2016; Singh, 2019).

#### 2. Adım: Gradyanların Hesabı

Gradyan, x ve yönlerinde meydana gelen küçük değişimlerdir. Gradyan hesabı için aşağıdaki eşitlikler (Eş. 3.15 ve Eş. 3.16) kullanılır (Lai, 2016; Singh, 2019).

$$d_x = I(x+1, y) - I(x, y) \quad (3.15)$$

$$d_y = I(x, y+1) - I(x, y) \quad (3.16)$$

$d_x$  ve  $d_y$  sırasıyla yatay ve dikey gradyan,  $I(x, y)$   $(x, y)$  konumundaki piksel değeridir. İmgedeki tüm pikseller için bu işlem tekrarlanır.

#### 3. Adım: Gradyanın büyüklüğünün ve yönünün hesaplanması

Önceki adımda hesaplanan gradyan değerleri kullanılarak her pikselin değeri ve büyüklüğü hesaplanır. Gradyanın büyüklüğü ( $g$ ) ve yönü ( $\theta$ ) için aşağıdaki denklemler (Eş. 3.17 ve Eş. 3.18) kullanılır (Lai, 2016; Singh, 2019).

$$g = \sqrt{d_x^2 + d_y^2} \quad (3.17)$$

$$\theta = \tan^{-1}\left(\frac{d_y}{d_x}\right) \quad (3.18)$$

#### 4. Adım: Gradyanların histogramının bloklara bölünmesi

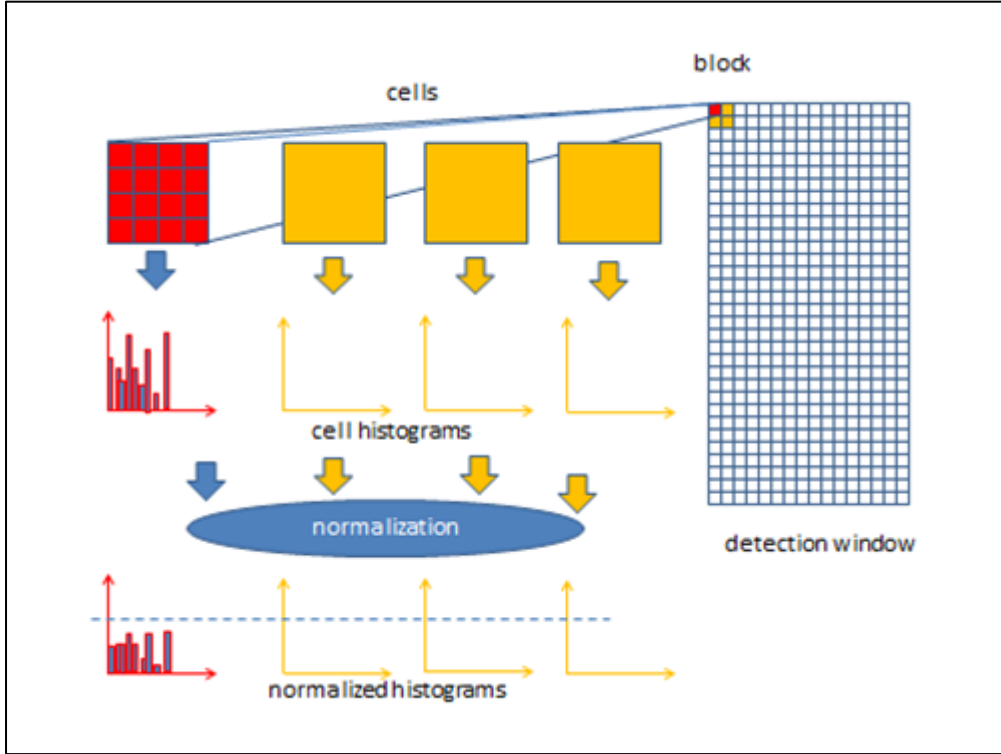
Gradyanlar histogramlarının bölünmesiyle görüntüyü temsil edecek özellikler çıkartılır. Daha fazla bölme kullanmak, görüntüden daha ayrıntılı yönlendirme bilgileri çıkarır, ancak daha fazla sayıda özellik üretir. Genelde tercih edilen 8x8 ya da 16x16'dır (Lai, 2016; Singh, 2019).

##### 5. Adım: Gradyanların normalleştirilmesi

İmgenin gradyanları aydınlatmaya duyarlıdır. Eğer bu değerler normalleştirilmezse imgede belirli bölümler diğer bölümlere göre daha parlak olacaktır. Normalleştirme için aşağıdaki denklemler (Eş. 3.19 ve Eş. 3.20) kullanılır (Lai, 2016; Singh, 2019).

$$V_k = \sqrt{\sum_{i=1}^n v_i^2} \quad (3.19)$$

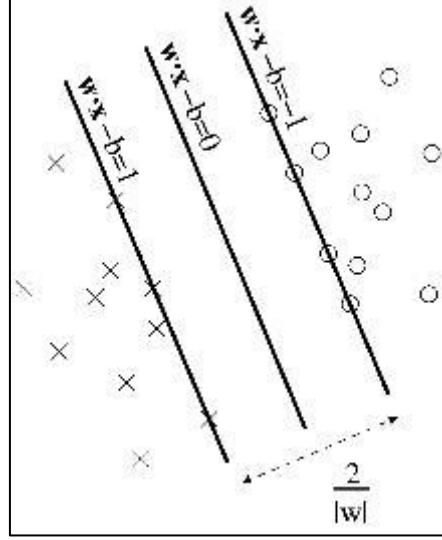
$$V_{\text{norm}} = \left[ \frac{v_1}{v_k}, \frac{v_2}{v_k}, \dots, \frac{v_n}{v_k} \right] \quad (3.20)$$



Şekil 3.15 YGH blok diyagramı (Singh, 2019).

Şekil 3.15 Yukarıda belirtilen YGH yönteminin çalışma adımlarını gösteren blok diyagram verilmiştir.

DVM'ler iki sınıfa ait verileri en uygun şekilde birbirinden ayırmak için kullanılan bir makine öğrenimi algoritmasıdır. Bu ayırma işlemini hiper düzlemler ile yapmaktadır. İki gruba da en yakın noktalar üzerinden paralel çizgiler çizilir ve bu çizgileri birbirine yaklaştırarak ortak tek bir sınır çizgisi çizilir (Ulgen, 2017) (Şekil 3.16).



Şekil 3.16 DVM ile sınıflandırma işlemi (Şeker, 2010)

N adet elemandan oluşan bir veri kümesinin her bir noktasının tanımı “Eş. 3.21” ile gösterilir.

$$D=\{(x_j, y_j), j = 1 \dots N, y_j \in \{-1, 1\}, x_j \in R^n\} \quad (3.21)$$

$$f(x)=wx+b \quad (3.22)$$

Şekil 3.16 gösterildiği gibi her nokta “Eş. 3.22” ile ifade edilebilir. Bu denklemde w ifadesi karar fonksiyonun normalini, x ifadesi doğru üzerindeki noktaları ve b ise eğilim değerini gösterir. Burada amaç w ve b değerlerini eğitim verileri ile bulmaktır. Doğrular arasındaki mesafe ise “Eş. 3.23” ile elde edilir (Şeker, 2010; Ateş ve Küçüksille, 2013).

$$\text{Mesafe} = \frac{2}{|w|} \quad (3.23)$$

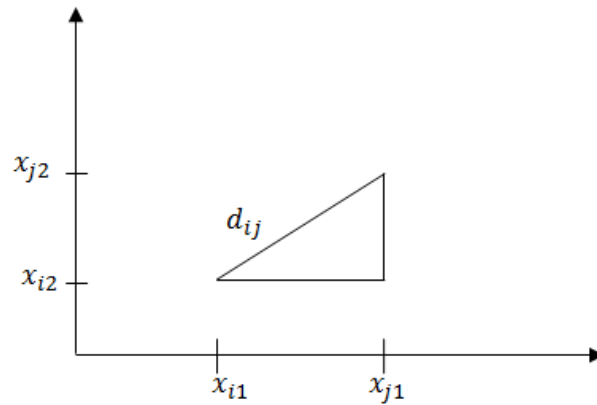
### 3.1.7. Uzaklık ölçüm yöntemleri

Bir birimin bir başka birime ya da kümeye olan yakınlığının veya uzaklığının sayısal değerlerle ifade edilmesinde uzaklık ölçüleri kullanılmaktadır. Aşağıda gösterildiği üzere çok farklı uzaklık ölçüleri vardır.

- Öklid Uzaklık Ölçüsü
- Manhattan City-Block Uzaklık Ölçüsü
- Chebyshev Uzaklık Ölçüsü
- Minkowski Uzaklık Ölçüsü
- Karl-Pearson Uzaklık Ölçüsü/ Standartlaştırılmış Öklid Uzaklık Ölçüsü
- Kolerasyon Uzaklığı Ölçüleri

#### 3.1.7.1. Öklid uzaklık ölçüsü

Uzaklık ölçüleri arasında en çok kullanılanlardan biridir. Çok boyutlu uzayda Pisagor bağıntısı yardımıyla bulunan uzaklıktır. Herhangi iki gözlem arasındaki uzaklığın x-y düzlemindeki hali Şekil 3.17'deki gibidir.



Şekil 3.17 Herhangi iki gözlem arasındaki uzaklık

Öklid uzaklığı;

$$d_{ij} = \sqrt{\sum_{k=1}^p (x_{ik} - x_{jk})^2} \quad (3.24)$$

Şeklindedir (Tatlıldil, 1996). Burada;

$d_{ij}$  ; i. ve j. birimin birbirine olan uzaklığı

$x_{ik}$  ; i. birimin k. değişken değeri

$x_{jk}$  ; j. birimin k. değişken değeri

$i = 1, \dots, n$  ;

$j = 1, \dots, n$

$k = 1, \dots, p$  'dir.  $n$  birim ve  $p$  değişken sayısıdır.

Karesel Öklid uzaklığı ise;

$$d_{ij}^2 = \sum_{k=1}^p (x_{ik} - x_{jk})^2 \quad (3.25)$$

Biçimindedir (Tatlıldil, 1996). Aykırı (uç) değerlere daha fazla ağırlık verir.

### 3.1.7.2. Chebychev uzaklık ölçüsü

İki gözlem arasındaki farkların mutlak değerinin en büyüğü olarak tanımlanır ve

$$d_{ij} = \max_k^p |x_{ik} - x_{jk}| \quad (3.26)$$

Biçiminde hesaplanır (Alpar, 2011).

### 3.1.7.3. Manhattan (City-Block) uzaklık ölçüsü

Bu uzaklık ölçüsü, birimlerin arasındaki uzaklık farklarının mutlak değerlerinin toplamı ile bulunan bir ölçüdür ve

$$d_{ij} = \sum_{k=1}^p |x_{ik} - x_{jk}| \quad (3.27)$$

Biçiminde ifade edilir.

Kesikli sayısal veriler için kullanılması tavsiye edilir (Johnson ve Wichern, 1988).

#### 3.1.7.4. Minkowski uzaklık ölçüsü

Nicel veriler için kullanılan uzaklık ölçümlerinden biri de Minkowski uzaklığıdır. Minkowski uzaklık ölçüsü

$$d_{ij} = \left[ \sum_{k=1}^p |x_{ik} - x_{jk}|^q \right]^{1/q} \quad (3.28)$$

Şeklindedir. “Eş. 3.28”  $q = 1$  için City-Block uzaklık ölçüsünü,  $q = 2$  için Öklid uzaklık ölçüsünü verir (Gündüz, 2011).  $q$  boyut değerini belirler.

#### 3.1.7.5. Karl-Pearson uzaklık ölçüsü/ standartlaştırılmış Öklid uzaklık ölçüsü

Öklid uzaklığındaki farkların  $1/s_k^2$  ile düzeltilmesi/ standartlaştırılması ile elde edilir ve

$$d_{ij} = \sqrt{\sum_{k=1}^p \frac{1}{s_k^2} (x_{ik} - x_{jk})^2} \quad (3.29)$$

Biçiminde ifade edilir. Genellikler farklı ölçü birimlerimde kullanılır (Alpar, 2011).

#### 3.1.8. Yüz bulma yöntemleri

Yüz bulma ile ilgili birçok algoritma önerilmiştir. Bu algoritmalar incelenerek sunulan yöntemler dört kategoriye ayrılmıştır. Bunlar: (Yang ark., 2002; Hjelmås ve Low, 2001; Zhang ve Zhang, 2010).



### 3.1.8.1. Bilgi tabanlı yöntemler

Bu yöntemde yüz bulmanın temel dayanak noktası insan yüz bilgisine dayanmaktadır. Yüz özelliklerini ve ilişkilerini tanımlayan basit kurallar koyulmuştur. Örneğin; bir görüntüde genellikle birbirine simetrik duran iki göz, iki gözün arasında bir burun ve burunun aşağısında ağzın olduğu bir yüz görülmektedir. Özellikler arasındaki mesafe ve konum bilgileri aralarındaki ilişkiyi göstermektedir (Karaali, 2012; Kaplan, 2018).

Yang ve Huang (1994), yüz tespiti için üç seviyeden oluşan hiyerarşik bilgiye dayalı bir algoritma üzerinde çalışmışlardır. Bu algoritma görüntü üzerinde her seviyede farklı özellikleri tarayarak son aşamaya kadar tüm seviyeleri geçen görüntü, yüz olarak kabul edilmektedir.

### 3.1.8.2. Özellik tabanlı yöntemler

İnsan gözü, farklı ışıklarda ve pozlarda yüzleri rahatlıkla tanımlayabilmektedir. Kullanılan bazı yöntemler yüzü tespit etmekte başarılı olmaktadır. Örneğin; kenar algılayıcılar ile kaş, burun, göz, ağız ve saç çizgisi gibi belirgin özellikleri tanır ve daha sonra yüzü bulup teyit eder. Böylece bu özniteliklere göre yüzün varlığını kanıtlayacak bir istatistiksel model oluşturulur. Bu yöntemde kullanılan algoritmaların dezavantajı aydınlatma, gürültü ve pozlara karşı duyarlı olmasıdır (Karaali, 2012; Kaplan, 2018).

Bu yöntemi kullanarak yüz tespiti yapan Leung ve ark. (1995), yüzü tanımlamak için burun deliği, göz ve dudak ilişkisini kullanmıştır. Ayrıca bir başka çalışmada ten rengi de bu yöntemde yüzü tanımlamak için kullanılmıştır (Burl ve ark., 1995).

### 3.1.8.3. Şablon eşleştirme yöntemleri

Bu yöntemde, önceden tanımlanmış ve ön cepheden çekilmiş bir yüz görüntü şablonu ya da yüz şablonunun bir fonksiyonu yüzü temsil etmek için kullanılmaktadır. Yüzün tespiti için giriş görüntüsü ile şablon arasındaki korelasyon hesaplanmakta ve bu değere göre yüz olup olmadığı belirlenmektedir (Bayrakdar ve ark., 2016; Karaali, 2012).

### 3.1.8.4. Görünüm tabanlı yöntemler

Bu yöntemde yüze ait özellikleri çıkartmak için çeşitli istatistiksel sınıflandırma ve makine öğrenmesi yöntemleri kullanılmaktadır. Makine öğrenmesi tabanlı olanlar görüntüdeki yüzlerin tespiti için eğitim verilerini kullanırlar. Eğitim verilerinin doğruluğu yüz tespit oranını da arttırmaktadır (Çetinkaya, 2012). Bu yöntemde yüzün tespiti için aşağıdaki adımlar uygulanır (Kaplan, 2018);

1. Görüntülere ön işlem uygulayarak, yüz bulmaya uygun hale getirilmesi.
2. Test ve eğitim verilerinin tek standarda getirilmesi.
3. Algoritmanın hem pozitif hem de negatif verilerle eğitilmesi.
4. Yüzün algılanması için bir arama izleminin uygulanması.

Görünüm tabanlı yöntemlerinin kullandığı, iyi bilinen sınıflandırma algoritmaları şunlardır (Kaplan, 2018);

- Temel Bileşen Analizi
- DVM
- YSA
- Naive Bayes sınıflandırıcısı
- Gizli Markov Modeli
- Seyrek Ağ Analizi (SNOW-Sparse Network of Winnows)
- Adaboost

Yüz bulma yöntemlerinin avantajları ve dezavantajları Çizelge 3.2’de gösterilmektedir.

Çizelge 3.2 Yüz bulma yöntemlerinin karşılaştırılması (Kaplan, 2018)

Yaklaşım	Özellik Tabanlı	Görünüm Tabanlı	Şablon Eşleme	Bilgi Tabanlı
Özellikler	Bölgesel	Bütünsel	Şablon	Tanımlanmamış
Karmaşıklık	Kolay	Zor, Uzun eğitim süreci	Kolay	Kodlanması zordur
Görüntü adedi	Tek resim olsa dahi yüksek başarı	Resim sayısı fazla olmak zorunda	Tek resim olsa dahi yüksek başarı	Görüntüye göre değişkenlik gösterir
Boyut	Düşük boyutlu	Çeşitli yöntemlerle boyut azaltılmalıdır	Fark etmez	Fark etmez
Bellek Kullanımı	Minimum	Maksimum	Kalıp boyutu kadar	Kurallar doğrultusunda
Çevresel Duyarlılık	Az	Çok	Orta	Kurallara bağlı
Başarım	Düşük başarı	Yüksek başarı	Orta	İyi
Avantajlar	Yüze ait poz ve doğrultuları aydınlanma koşullarından bağımsızdır	1.İşbatlanmış başarılı sonuçlar üretirler. 2.Güçlü ve başarılı makine öğrenmesi algoritmalarını kullanırlar. 3.Farklı boyutlardaki ve doğrultulardaki yüz resimleri için de başarılı çalışırlar.	Hesaplama açısından oldukça basit yöntemlerdir.	1.Karmaşık olmayan arka planlarda oldukça başarılı sonuçlar üreten yöntemlerdir. 2.kodlanmış kurallar baz alınarak, bir test resmi içindeki yüze ait özellikleri çıkarmak ve sonra aday yüzleri onaylamak basit yöntemlerdir. 3.Yüze ait özellikleri ve bu özelliklere ait ilişkileri tanımlayacak kurallar bulmak çok kolaydır.
Dezavantajlar	1.Karmaşık arka planlarda yüzlerin bulunması zordur. 2.Aydınlanma ve resimlerdeki diğer benzeri gürültüler, yüzlerdeki özelliklerin bulunmasını zorlaştırır.	1.Sınıflandırma algoritmalarını eğitmek için birçok pozitif ve negatif örnek gerektirirler. 2.Genellikle resim üstünde tarama işlemi gerektirirler.	1.Farklı poz doğrultuları için şablon yüzlerin bulunması ve doğru olarak uygulamak zordur. 2.Şablonlar eğer yüze yakın bölgelerden itibaren test resimleri üzerinde taranmazlarsa yüzün bulunması çok maliyetli olabilir.	1.Farklı pozlardaki resimler için bu yöntemlerin uygulanması oldukça güçtür. 2.İnsan bilgisini kurallara dönüştürmek her zaman kolay olmayabilir. Çok detaylı kurallar kullanıldığında yüzleri bulmak mümkün olmayabilir, daha az detaylı kurallar kullanıldığında da hatalı pozitif sonuçlar bulunabilir.

### 3.1.9. Yüz nirengi noktalarının bulunması

Yüz tanıma, yüz izleme, duygu analizi ve yüz modelleme alanlarında sıklıkla kullanılan yüz nirengileme, insan yüzündeki ağız, burun, kaş, göz ve yüz çehresi gibi ortak bulunan bazı referans noktaların saptanmasıdır (Çeliktutan, 2013.; Wang, 2014; Cao, 2012). Oldukça zor bir süreç olan yüz nirengi noktalarının tespiti, insanların ırk, cinsiyet, yüz ifade farklılıkları, poz değişiklikleri ve gözlük gibi aksesuarların kullanımıyla biraz daha güçleşmektedir (Çeliktutan, 2013). Yüz nirengi noktalarının tespitinde sıklıkla kullanılan yöntem ASM'dir.

ASM, 1995'te Tim Cootes ve Chris-Taylor tarafından geliştirilmiştir. Görüntüdeki nesnenin bir örneğine uyması için yenilemeli olarak çalışabilen, esneyebilen bir nesne şeklinin istatistiksel modelidir. Bu yöntem şekilleri, bir dizi noktalarla ifade eder. Oluşturulan bu şekillerin hizalanması için Procrustes Analiz yaklaşımı kullanılır. Hizalanmış verilerin ortak bir ortalamaya sahip olmasıyla iki şekil hizalanmış (eşleştirilmiş) olur (Sung, 2007; Shbib, 2015; Martin, 2008).

Bir şekil,  $n$  adet noktadan oluşan ve genellikle iki veya üç boyutlu bir yer işareti kümesini temsil etmektedir. Şekilleri tanımlayan noktalar kenarlıklar, tepe noktaları, profil noktaları, köşeler veya diğer özelliklerle eşleşecek şekilde seçilmektedir (Martin, 2008).

Matematiksel olarak  $k$  boyutlu uzayda  $n$  dönüm noktası ile tanımlanan bir şekil, bir  $n_k$  vektörü ile temsil edilir. İki boyutlu bir resimde ( $k = 2$ ),  $n$  nirengi noktası  $\{(x_i, y_i) : i = 1, \dots, n\}$ ,  $X = (x_1, y_1, x_2, y_2, \dots, x_{n-1}, y_{n-1}, x_n, y_n)^T$  vektörüyle tanımlanmaktadır (Martin, 2008). Şekil 3.18 bir insan yüzünden elde edilen nirengi noktalarını göstermektedir.



Şekil 3.18 Yüz nirengi noktaları(Martin, 2008)

Şekillerin hizalanması ve Procrustes Analizi: İki şeklin hizalanması, ölçek, döndürme ve çevirme gibi benzerlik parametrelerine dayanarak iki şekli bir birine uyduracak minimum ölçüyü bulmasıdır. İki şekli hizalamanın klasik çözümü Procrustes Analiz yöntemidir (Martin, 2008).

İki şekli  $x_1$  den  $x_2$  'ye hizalamak için T dönüşüm parametrelerini  $x_1$  'e uygulandığında onu en iyi şekilde hizalayan  $x_2$  'yi bulmak için, minimum procrustes mesafesinin bulunması gereklidir (Eş. 3.30) (Martin, 2008).

$$D_{\text{Procrustes}}(x_1, x_2) = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2 + (y_1 - y_2)^2} \quad (3.30)$$

İlk adım, şekillerin ağırlık merkezlerini  $\bar{x}$  ve  $\bar{y}$  olarak hesaplamaktır (Eş. 3.31).

$$(\bar{x}, \bar{y}) = \left( \frac{1}{n} \sum_{i=1}^n x_i, \frac{1}{n} \sum_{i=1}^n y_i \right) \quad (3.31)$$

İkinci adım, şekilleri başlangıç noktasına kaydırarak her iki şekil ağırlık merkezlerini  $\bar{x}_1$  ve  $\bar{x}_2$  olarak hizalamaktır (Eş. 3.32).

$$x_{1c} = x_1 - \bar{x}_1 \quad \text{ve} \quad x_{2c} = x_2 - \bar{x}_2 \quad (3.32)$$

Üçüncü adım, her bir şekil İzomorfik dönüşüm uygulanarak normalleştirilmektedir (Eş. 3.34).

$$\widetilde{x}_1 = \frac{x_{1c}}{\|x_{1c}\|} \quad \text{ve} \quad \widetilde{x}_2 = \frac{x_{2c}}{\|x_{2c}\|} \quad (3.34)$$

Dördüncü adım, şekil matrislerinin düzenlenmesi işlemidir (Eş. 3.35).

$$\acute{X}_1 = (\widetilde{x}_1 | \widetilde{y}_1)_{nx2} \quad \text{ve} \quad \acute{X}_2 = (\widetilde{x}_2 | \widetilde{y}_2)_{nx2} \quad (3.35)$$

Beşinci adım,  $\acute{X}_1$  ve  $\acute{X}_2$  değerlerine tekil değer ayrışımı uygulanır. Yani, bir matrisi 3 parçaya ayırarak tutar ve bu üç parçayı kullanarak aynı matrisin yeniden elde edilmesini

sağlamaktadır (Eşitlik 3.36). Bu değer de en uygun dönüşüm matrisini vermektedir (Martin, 2008).

$$USV^T = TDA(\hat{X}_2 \hat{X}_1^T) \quad (3.36)$$



Şekil 3.19 Eşleştirilmiş iki şekil.

İki şeklin en uygun değerde eşlenmesiyle Şekil 3.19’da gösterilen bir görüntü ortaya çıkar.

### 3.1.10. Veri seti

#### 3.1.10.1. FEI yüz veri seti

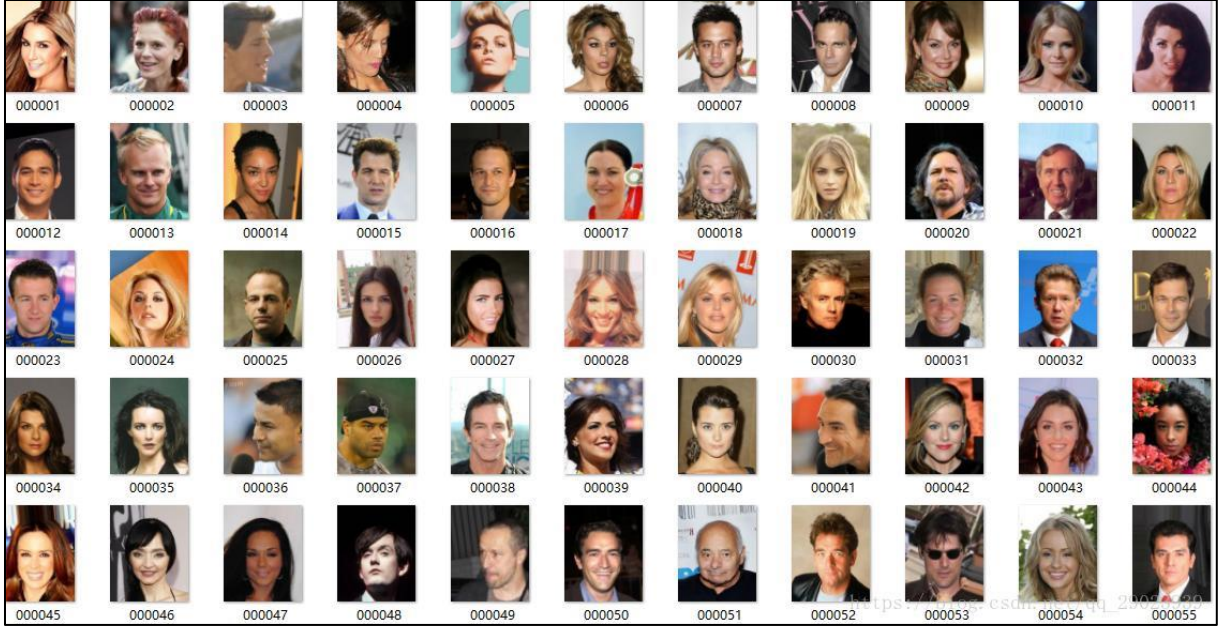
Brezilya'daki FEI Yapay Zekâ Laboratuvarı'nda Haziran 2005 ile Mart 2006 arasında 19-40 yaşlar aralığındaki öğrenciler ve personel tarafından farklı görünüm, saç modeli ve aksesuarlar tercih edilerek çekilmiş bir dizi yüz görüntüsü içeren Brezilyalı yüz veri tabanıdır. FEI yüz veri tabanında 200 kişinin 14 farklı poz ve ışık değerinden oluşan toplamda 2800 yüz görüntüsü bulunmaktadır. Tüm görüntüler homojen arka plana karşı dik konumda önden 180 dereceye kadar döndürülerek çekilmiştir. Toplamda 200 kişi olan deneklerin her birinin Şekil 3.20'de gösterildiği gibi (biri nötr, biri gülümseyen, biri gülümsemeyen ve diğerleri de farklı açılarda olan) 14 farklı pozunu bulunmaktadır. Her poz arasında yaklaşık %10'luk bir değişim vardır. Her görüntünün orijinal boyutu 640x480 pikseldir (Thomaz, 2012).



Şekil 3.20 FEI veri setinden bir kesit (Thomaz, 2012).

#### 3.1.10.2. CelebA veri seti

Bu veri seti 40 öznitelik ek açıklaması olan 200.000'den fazla ünlü resmine sahip büyük ölçekli bir yüz veri setidir. 10.177 bireyin sarı saç, gülümseme, keçi sakalı ve gözlük gibi çeşitli özellikleri vardır (Liu, 2012). Her görüntünün boyutu 178x218 olarak belirlenmiştir. Tez çalışmasında bu veri setinin bir kısmı kullanılmıştır. Şekil 3.21 bu veri setine ait görüntülerden oluşan bir kesit paylaşılmıştır.



Şekil 3.21 CelebA veri setinden bir kesit (Liu, 2012).

### 3.1.10.3. Aile yüz veri seti

Aile yüz veri seti; 39 çekirdek ailedeki bireylerin, çoğunluğu biyometrik resim (kimlik, pasaport ve ehliyet gibi belgelerde kullanılan, en az 600dpi olacak şekilde yüksek kaliteye sahip, arka fonu beyaz ve başka objelerden arındırılmış vesikalık fotoğraf türü) olan yüz görüntülerinden oluşmaktadır (Şekil 3.22). Bu veri setinde her ailenin 4-9 arasında değişen bireylerinin yüz görüntüleri bulunmaktadır. Toplam yüz görüntü sayısı 208'dur. Bu veri seti ilk kez tarafımızca oluşturulmuştur ve herhangi bir başka çalışmada kullanılmamıştır.



Şekil 3.22 Aile yüz veri setinden örnek bir aile.



### 3.2. Yöntem

İlgili literatür ve yapılan çalışmalar incelendiğinde, doğrudan görüntü üzerinde işlem yapan uygulamalarda ESA yönteminin kullanıldığını görülmektedir. Bu yöntem kullanılarak yapılan uygulamaların başarı oranı diğer yöntemlere göre biraz daha yüksek olduğu görülmektedir.

İTGE sistemlerde özellik çıkarımı için imgenin renk, doku ve şekil gibi özelliklerinin sisteme aktarılıp sayısallaştırılmasıyla yapılmaktadır. Bireylerin yüz organ özelliklerini İTGE sistemleriyle elde etmenin güç olacağı düşünüldüğünden tercih edilmemiştir. Onun yerine özellik çıkarımı için ESA ve ASM yöntemleri tercih edilmiştir ve bireylerin yüz nirengi noktaları arasındaki Öklid uzaklığını kullanarak değerlendirme yapılmıştır.

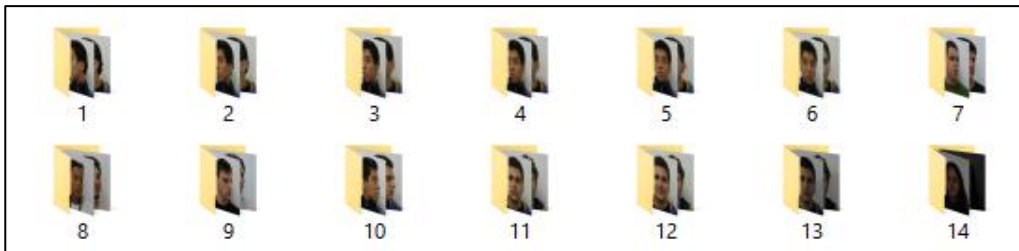
#### 3.2.1. Veri setlerinin düzenlenmesi

Tez çalışmasında üç farklı veri seti üzerinde çalışılmıştır. Bunlar:

- FEI veri seti
- CelebA veri seti
- Aile yüz veri seti

##### 3.2.1.1. FEI veri setinin düzenlenmesi

Ham olarak indirilen veri setinde 200 kişinin 14 farklı pozdaki görüntüleri bulunmaktadır. Bu görüntüler hatalı ya da bozuk olma ihtimaline karşın tek tek incelenmiştir. Görüntüler Şekil 3.23'te gösterildiği gibi birey ve poz bazında etiketlenerek klasörleştirilmiştir.



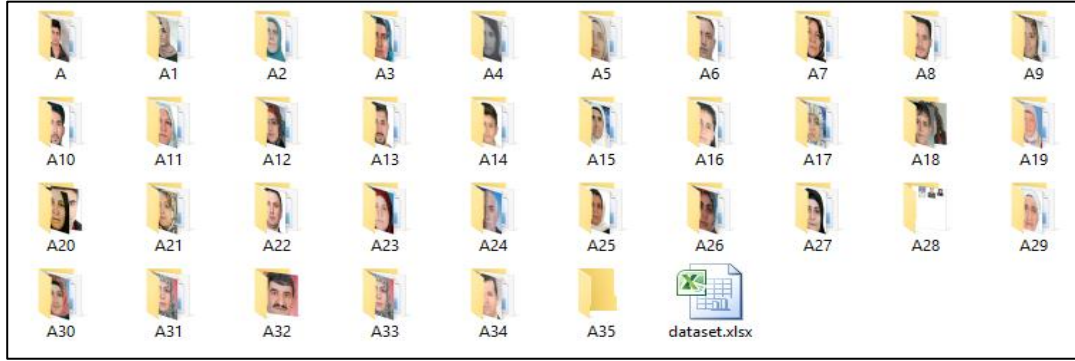
Şekil 3.23 FEI yüz veri seti.

### 3.2.1.2. CelebA veri setinin düzenlenmesi

Bu veri setinde yaklaşık 200,000 görüntü bulunmaktadır. Çalışmada kullanılmak üzere bu veri setinden 2800 adet görüntü alınmıştır.

### 3.2.1.3. Aile yüz veri setinin düzenlenmesi

39 aileden birey görüntüleri talep edilerek toplanan görüntülerin herhangi bir platformda paylaşılmayacağı belirtilmiştir. Bu kapsamda alınan etik raporu da tezin ekler kısmında sunulmuştur. Toplanan resimler tarayıcı aracılığıyla dijital ortama atılmıştır. Dijital ortamdaki bu görüntüler Şekil 3.24'te gösterildiği gibi aile bazında etiketlenerek klasörleştirilmiştir.



Şekil 3.24 Aile yüz veri seti.

### 3.2.2. Yöntemin belirlenmesi ve testi

Literatürde yapılan incelemeler sonucunda görüntü üzerindeki yüzlerin tespiti noktasında birçok yöntem (Dlib ESA, Dlib YGH, Dlib DNN, Dlib DNN + k-NN, FaceNet 20180402-114759, OpenCV, OpenCV Fisherfaces, OpenCV EigenFaces vb.) bulunmaktadır. Bu bağlamda çalışmamızda kullanmak üzere, yüz tanıma sistemlerinde popüler olan ve diğerlerine göre daha iyi sonuçlar üreten, makine öğrenmesi algoritmalarından biri olan YGH-DVM, DSA ve ESA olmak üzere üç farklı yöntem seçilmiştir. Bu yöntemler arasından performans olarak en iyi sonucu veren yöntemin çalışmada kullanılmasına karar verilmiştir.

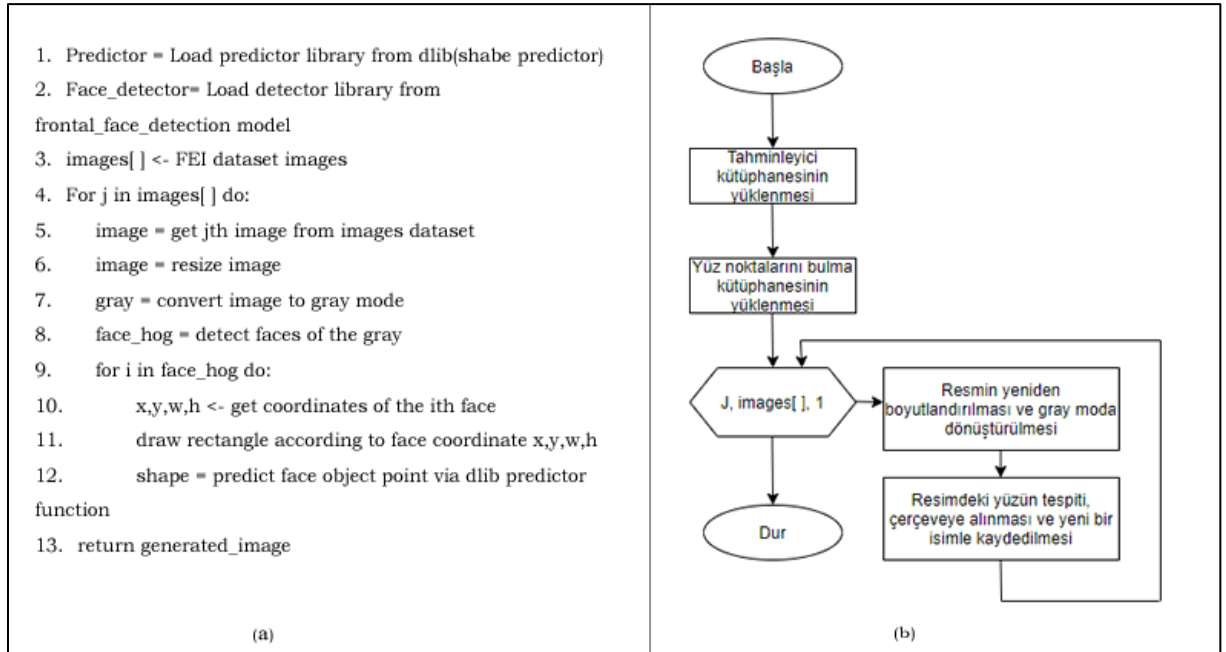
### 3.2.2.1. YGH-DVM algoritması

Bu algoritmada özellik çıkarıcı olarak YGH, sınıflandırıcı olarak da DVM kullanılmıştır. YGH özellik çıkarıcısında degradelerin (x ve y türevleri) yönlerinin dağılımı özellik olarak kullanılır. Degradelerin büyüklüğü kenarlarda ve köşelerde (ani yoğunluk değişimlerinin olduğu bölgeler) büyük olduğundan nesnenin şekli hakkında daha çok bilgi içermektedir (Mallick, 2016). YGH'ların, bloklar ve hücrelerden oluşan resimlerin sıkıştırılmış ve kodlanmış halleri olduğu söylenebilir. Her pikselin parlaklık yoğunluğu ve yönleri hakkında bilgi içermektedirler.

DVM'ler iki sınıfa ait verileri en uygun şekilde birbirinden ayırmak için kullanılan bir makine öğrenimi algoritmasıdır. Bu ayırma işlemini hiper düzlemler ile yapar. İki gruba da en yakın noktalar üzerinden paralel çizgiler çizilir ve bu çizgileri birbirine yaklaştırarak ortak tek bir sınır çizgisi çizilerek ayrılması sürecidir (Ulgen, 2017).

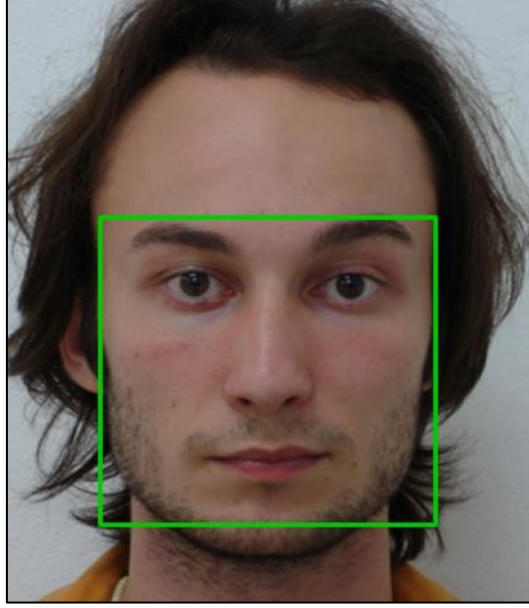
YGH algoritmasından gelen özellik vektörü, DVM modelinde her bir etiketle giriş vektörü için bir eşleşme puanı belirlemek için kullanılır. DVM, eğitim verisi içerisinde en yakın eşleşmeye ait maksimum puanlı etiketi döndürmektedir (Thaware, 2018; Soydaş, 2017).

YGH-DVM yöntemi kullanılarak görüntüdeki yüzün tespiti için hazırlanan programın algoritması ve akış diyagramı Şekil 3.25'te verilmiştir.



Şekil 3.25 YGH-DVM yöntemi için kullanılan algoritma (a) ve akış diyagramı (b).

YGH-DVM yöntemi uygulanarak görüntüdeki yüz tespit edilmiştir. Tespit edilen yüz çerçeve içine alınmıştır (Şekil 3.26).

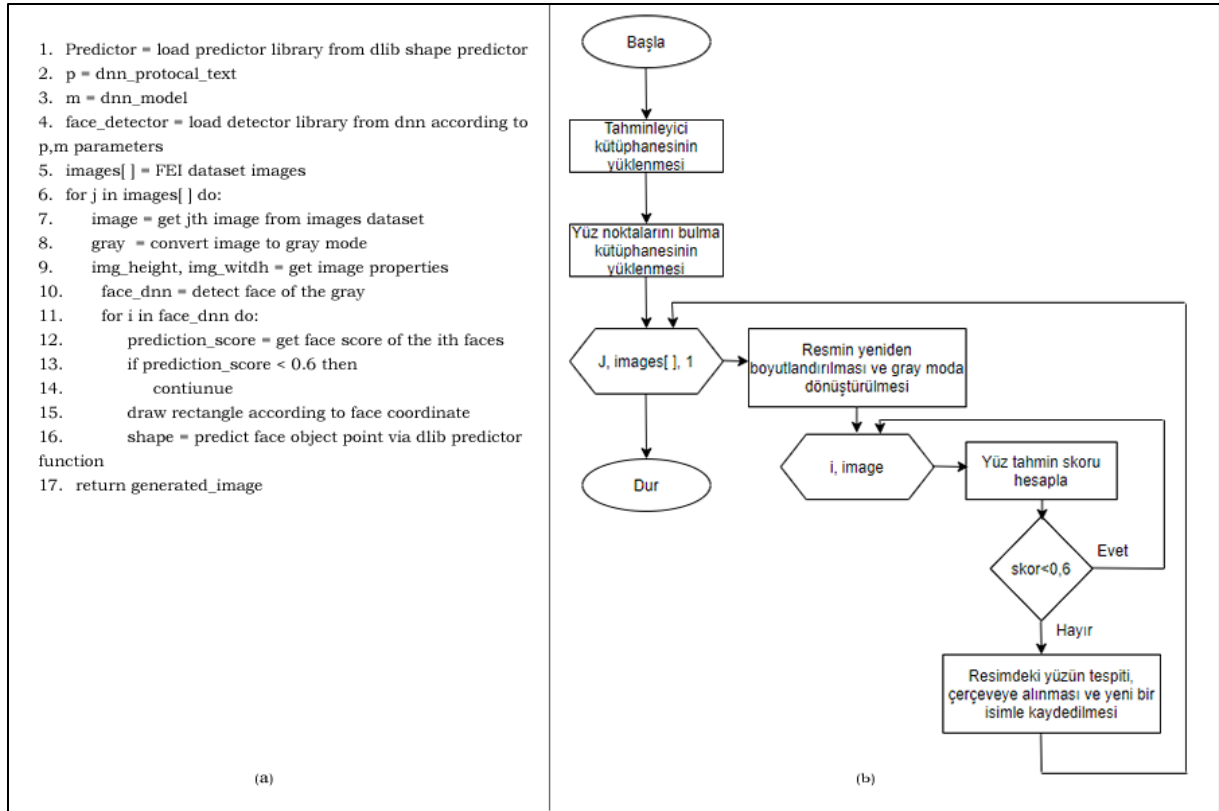


Şekil 3.26 YGH-DVM yöntemi ile yüzün tespiti.

### 3.2.2.2. DSA yöntemi

DSA, büyük eğitim verilerini işleyebilen ve çok sayıda gizli katmana sahip olan makine öğrenme modelleri ile çalıştığı için çok daha faydalı özellikleri öğrenebilir, sınıflandırabilir ve tahmin doğruluğunu arttırabilir.

DSA yöntemi kullanılarak görüntüdeki yüzün tespiti için hazırlanan programın algoritması ve akış diyagramı Şekil 3.27’de verilmiştir.



Şekil 3.27 DSA yöntemi için kullanılan algoritma (a) ve akış diyagramı (b).

DSA yöntemi uygulanarak görüntüdeki yüz tespit edilmiştir. Tespit edilen yüz çerçeve içine alınmıştır (Şekil 3.28).

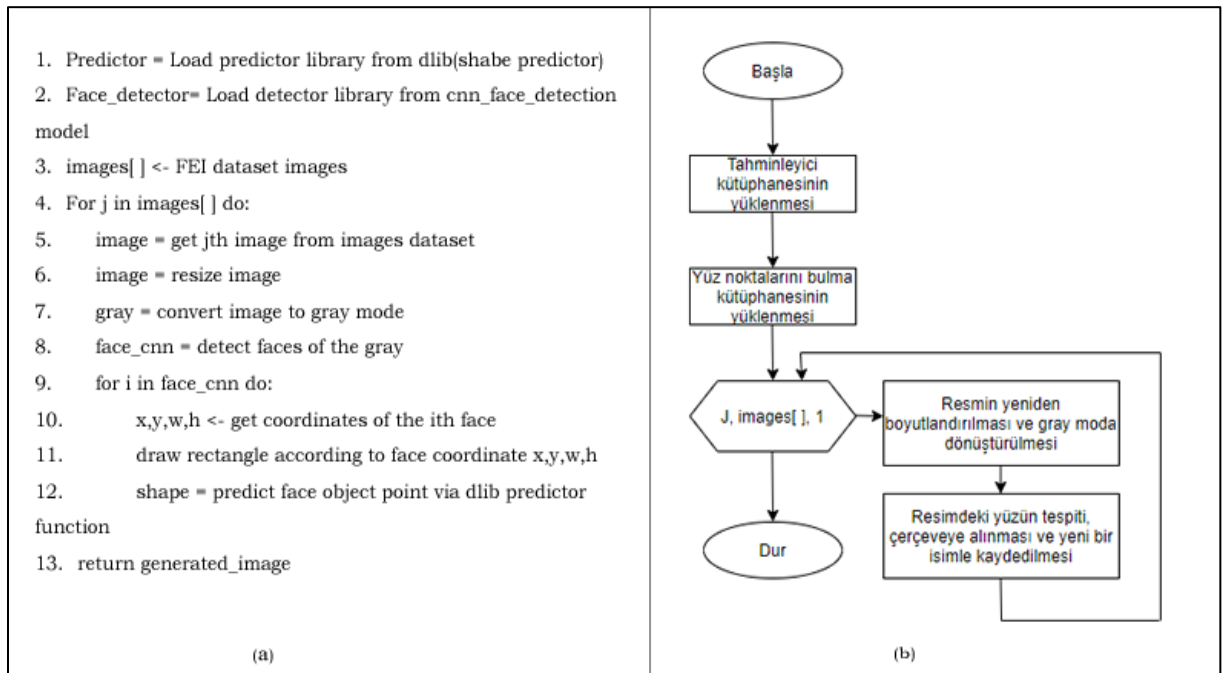


Şekil 3.28 DSA yöntemi ile yüzün tespiti.

### 3.2.2.3. ESA yöntemi

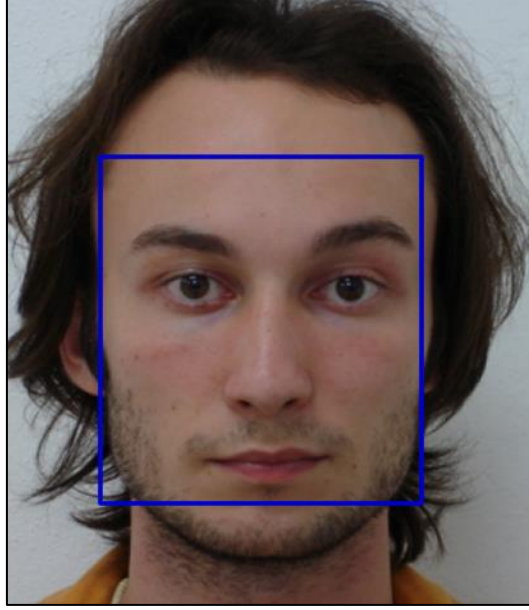
Eğitilebilen birçok katmandan oluşan ve bu sayede yüksek miktarda öznetelik üretebilen, ileri beslemeli, ESA mimarisi, DSA'ya göre daha hızlı öğrenir. Çünkü DSA'da girdi boyutu nedeniyle bağlantı büyüklüğü artar ve bu da yavaş öğrenmeye sebep olur ama ESA yapısında eğitim verisinin az tutulması ile bu dezavantaj ortadan kaldırılır (Akay, 2018; Hanılçı, 2019; Coşkun ve ark., 2017).

ESA yöntemi kullanılarak görüntüdeki yüzün tespiti için hazırlanan programın algoritması ve akış diyagramı Şekil 3.29'da verilmiştir.



Şekil 3.29 ESA yöntemi için kullanılan algoritma (a) ve akış diyagramı (b).

ESA yöntemi uygulanarak görüntüdeki yüz tespit edilmiştir. Tespit edilen yüz çerçeve içine alınmıştır (Şekil 3.30).



Şekil 3.30 ESA yöntemi ile yüzün tespiti.

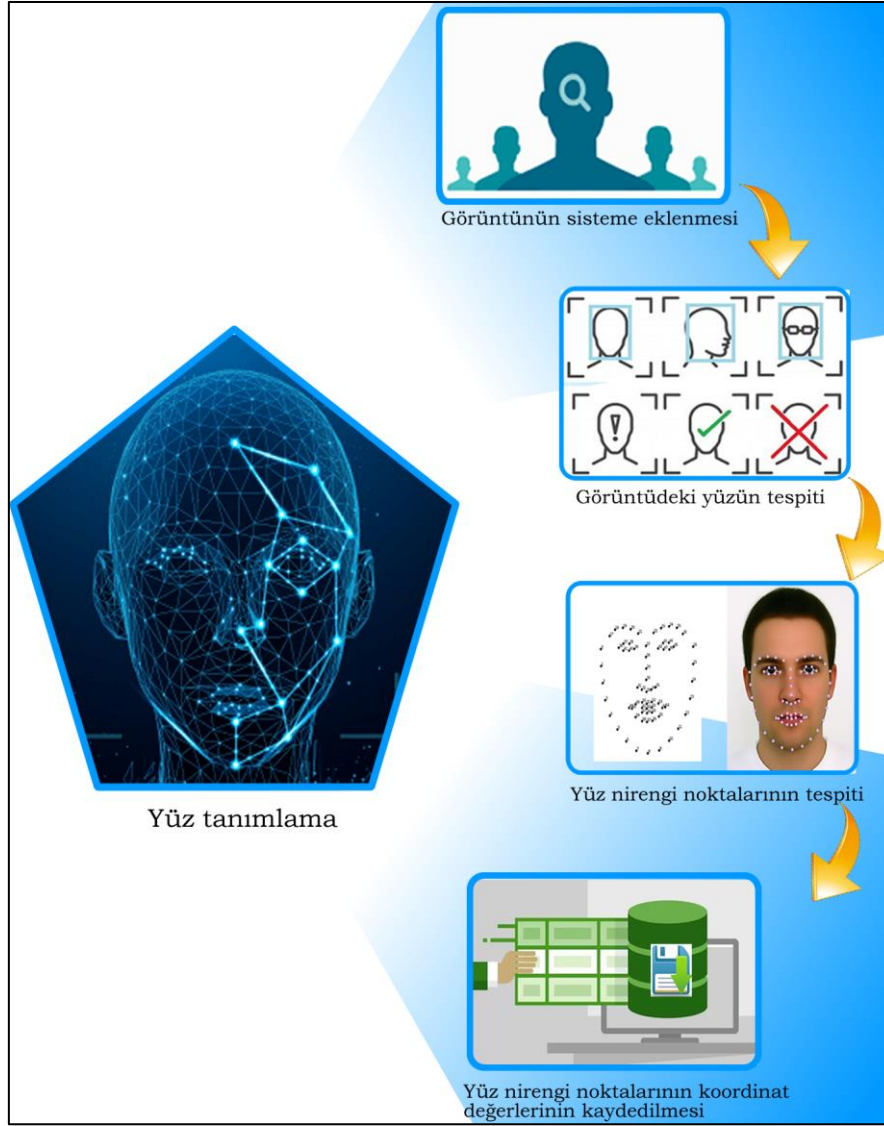
Yukarıda belirtilen üç algoritmanın da FEI, CelebA ve Aile yüz veri setleri üzerinde yapılan testler sonucunda en iyi performansı ESA yönteminin sergilediği görülmüştür ve bu doğrultuda hazırlanan sistemde ESA yönteminin kullanılmasına karar verilmiştir.

### 3.2.3. Belirlenen yöntemin veri setine uygulanması ve yüz nirengi noktalarının tespiti

Bir önceki bölümde FEI, CelebA ve Aile yüz veri setleri üzerinde yapılan testler sonucunda ESA yöntemi diğer yöntemlere nazaran görüntüdeki yüzü tespit noktasında daha iyi performans gösterdiği ve literatür incelendiğinde en popüler yaklaşım olarak vurgulandığı için çalışmada kullanılması sistemin kararlılığını arttıracaktır.

Gerçekleştirilen çalışmada yüz nirengi noktalarının tespiti için önceden eğitilmiş bir model dosyasına ihtiyaç vardır. Eğitilmiş bir model dosyasının temini için, C++ programlama dilinde platform-bağımsız olarak geliştirilmiş Dlib kütüphanesi kullanılmıştır. Dlib kütüphanesi 2002 yılından beri geliştirilmeye devam eden içerisinde derin öğrenme, makine öğrenimi ve bilgisayarlı görü algoritmalarını içeren, C++ ve Python API'sı sayesinde geniş bir kitleye hizmet sunan bir kütüphanedir.





Şekil 3.31 Yüz tanımlama blok diyagramı.

Şekil 3.31’de yüz nirengi noktalarının tespitinde kullanılan sistemin blok diyagramı gösterilmektedir. Aile yüz veri setindeki her bir ailedeki bireyin görüntüsü, hazırlanan programda *images* klasörü içerisine kopyalanır. Program çalıştırıldığında 1. adımda görüntüdeki yüz tespit edilir. 2. Adımda, yüz üzerindeki nirengi noktaları tespit edilir ve 3. adımda ise yüz nirengi noktalarının koordinat değerleri .csv formatında bilgisayara kaydedilir. Böylece sistemin ilk aşaması olan yüz tanımlama gerçekleştirilmiş olur.

Hazırlanan programda 4 adet fonksiyon kullanılmıştır. Bu fonksiyonlar;



- **crop\_function**

**Sözde Kod**

1. pic\_num = 1
2. files[ ]  $\leftarrow$  images aile dataset
3. for i in files[ ] do:
4.   frame = read i<sup>th</sup> image
5.   resize\_image = resize frame abouth 300 x 400
6.   save\_image  $\leftarrow$  save resized image according to pic\_num value
7.   Pic\_num++

Yukarıda sözde kodu verilen fonksiyon, veri setindeki tüm resimler aynı boyuta getirmek için hazırlanmıştır. Tüm program için belirtilen görüntü boyutu 300 x 400'dür (Şekil 3.32).



Şekil 3.32 Crop\_function ile 300x400 olarak boyutlandırılmış görüntü.

- **ffl\_function**

**Sözde Kod**

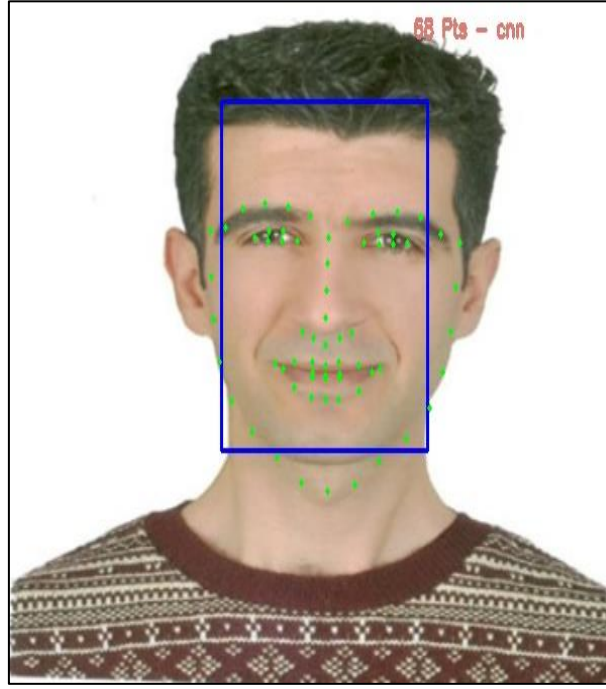
1. predictor = Load predictor library from dlib(shabe predictor)
2. face\_detector= Load detector library from cnn\_face\_detection model
3. pic\_num = 1

```

4. files[ ] ← images dataset
5. for i in files[ ] do:
6.   frame = read ith image from files
7.   gray = convert frame to gray mode
8.   clahe_img = convert gray to clahe mode
9.   detection = detect the faces in the clahe images
10.  for k,d in detection do:
11.    shape = predictor(clahe_img, d) get coordinate
12.    for (name, (i,j) ) in facial_landmarks_items do:
13.      for (x,y) in shape[i:j] do:
14.        if (i,j) == (48,68) then
15.          index_array[0] ← special_coordinate of the lips
16.          save_files ← special coordinates value of lips to files
17.        if (i,j) == (17,22) then
18.          index_array[1] ← special_coordinate of the left ibrows
19.          save_files ← special coordinates value of left ibrows to files
20.        if (i,j) == (22,27) then
21.          index_array[2] ← special_coordinate of the right ibrows
22.          save_files ← special coordinates value of right ibrows to files
23.        if (i,j) == (36,42) then
24.          index_array[3] ← special_coordinate of the left eye
25.          save_files ← special coordinates value of left eye to files
26.        if (i,j) == (42,48) then
27.          index_array[4] ← special_coordinate of the right eye
28.          save_files ← special coordinates value of right eye to files
29.        if (i,j) == (27,36) then
30.          index_array[5] ← special_coordinate of the nose
31.          save_files ← special coordinates value of nose to files
32.        if (i,j) == (0,17) then
33.          index_array[6] ← special_coordinate of the face side
34.          save_files ← special coordinates value of face side to files
35. pic_num++

```

Yukarıda sözde kodu verilen fonksiyon, ilk adım olarak; boyutlandırılmış olan görüntüdeki yüzü bulmaktadır. İkinci adımda; bulduğu yüz üzerinde yüz organlarını (kaş, göz, burun, ağız, yüz hattı) tespit etmektedir. Üçüncü adımda; yüz organlarının konumlarını (x,y koordinat değerlerini) tespit etmektedir. Dördüncü adımda; bu koordinat değerlerini .csv formatında kaydetmektedir (Şekil 3.33).



Şekil 3.33 Görüntü üzerindeki yüzün tespiti, yüz noktalarının bulunması ve kaydedilmesi.

#### • face\_crop\_function

##### Sözde Kod

1. lines[ ]  $\leftarrow$  read lines of the coordinate files
2. corner\_right\_up  $\leftarrow$  find from lines
3. corner\_left\_up  $\leftarrow$  find from lines
4. corner\_right\_down  $\leftarrow$  find from lines
5. corner\_left\_down  $\leftarrow$  find from lines
6. frame = read images
7. crop\_frame = frame [corner\_right\_up, corner\_left\_up, corner\_right\_down, corner\_left\_down]
8. generated\_image = resize crop\_frame 300 x 400
9. save genareted\_image

Yukarıda sözde kodu verilen fonksiyon, koordinat değerleri kaydedilmiş olan görüntü üzerindeki yüzün ayırt edilmesinde bir yararı olmayan, Şekil 3.33’de noktalarla gösterilen kısımlar dışında kalan bölgelerin, atılmasını sağlar. Bu durum programın tepki süresini hızlandırır (Şekil 3.34).



Şekil 3.34 İstenmeyen kısımları atılan ve yeniden boyutlandırılan görüntü.

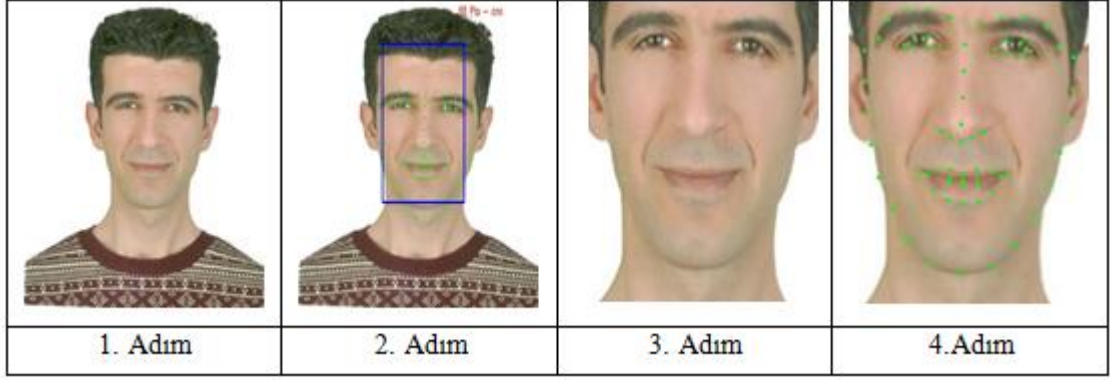
- **main\_function**

**Sözde Kod**

**Giriş** ← image dataset

**Çıkış** ← new\_images, image\_coordinate\_files

1. crop\_function( )
2. ffl\_function( ) #Face Find and Landmarks function
3. face\_crop\_function( )
4. ffl\_function( )



Şekil 3.35 Main\_function adımları.

Yukarıda sözde kodu verilen main\_function fonksiyonun izlediği adımlar görsel olarak Şekil 3.35’de gösterilmiştir.

#### 3.2.4. Yüz nirengi noktalarının düzenlenmesi ve kaydedilmesi

Her aile bireyinin görüntüsü üzerinde, yüz tespiti yapıp yüz nirengi nokta koordinat değerlerinin kaydedildiği .csv formatındaki dosyaların tamamı C# programlama dilinde, bu çalışmayı yapan kişi tarafından, hazırlanmış olan program ile düzenlenip veri tabanına kaydedilir. C# programlama dili, 1990’lı yılların sonlarında, yazılım sektöründe sıklıkla kullanılan C ve C++ dillerinden türetilmiş olup, .net framework platformu için hazırlanmış nesne yönelimli bir yazılım dilidir.

Şekil 3.36’da aile yüz veri setinde rastgele seçilmiş olan bir bireyin yüz nirengi noktaların kaydedildiği .csv dosyasının içeriğinin bir kısmı gösterilmektedir.

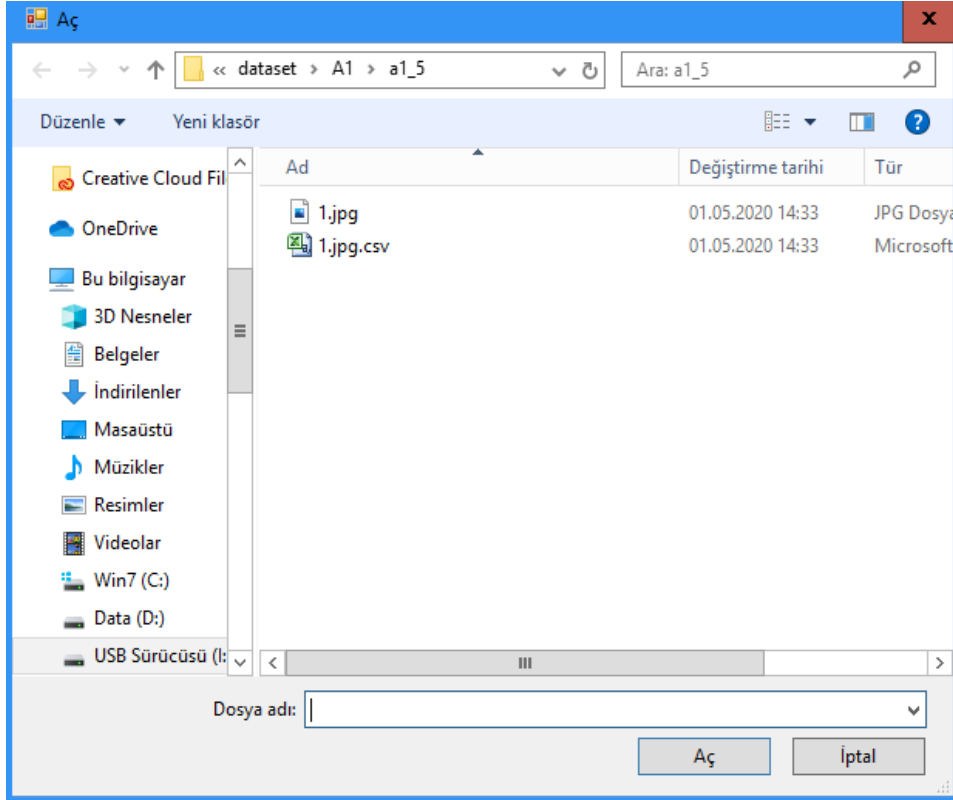
	A	B	C	D	E
1	index ,right_eye_degree , co_ordinate				
2	42	0	(200, 66)		
3	43	0	(216, 55)		
4	44	0	(233, 56)		
5	45	0	(244, 68)		
6	46	0	(232, 75)		
7	47	0	(216, 73)		
8	index ,nose_degree , co_ordinate				
9	27	0	(164, 61)		
10	28	0	(164, 95)		
11	29	0	(165, 128)		
12	30	0	(165, 163)		
13	31	45	(132, 175)		
14	32	45	(146, 183)		
15	33	45	(161, 191)		
16	34	45	(177, 185)		
17	35	0	(189, 178)		
18	index ,side_portion_degree , co_ordinate				
19	0	71	(16, 62)		
20	1	82	(12, 104)		
21	2	85	(12, 149)		
22	3	85	(14, 193)		
23	4	83	(25, 239)		
24	5	80	(42, 285)		
25	6	75	(67, 326)		
26	7	71	(103, 357)		
27	8	63	(147, 367)		

Şekil 3.36 Yüz nirengi nokta koordinat değerlerinin tutulduğu dosya içeriğinin bir kısmı.

Bu dosyada yer alan index ve koordinat değerleri bilgilerinin dosyadan çekilerek aile isimleriyle birlikte veri tabanına kaydedilmesi için C# programlama dilinde hazırlanan programın ara yüzü Şekil 3.37’de gösterilmektedir.

Şekil 3.37 Birey yüz nirengi nokta koordinat değerlerinin veri tabanına eklenmesi.

Aile adı kısmına kaydedilecek ailenin adı yazılır. Daha sonra *Dosya Seç* butonuna tıklanarak Şekil 3.38’de gösterilen diyalog penceresinden belirtilen aileye ait birey seçilir.



Şekil 3.38 Birey .csv dosyası seçim diyalog penceresi.

Seçilen bireye ait .csv formatındaki dosyanın adı, *Seçilen Dosya* etiketi kısmında görüntülenir. Ayrıca seçilen bireye ait index ve koordinat değerleri bilgileri *Birey İndex* ve *Koordinat Değerleri* kısmında görüntülenir (Şekil 3.39).



Aile Kayıt / Referans Yüz Modeli Oluşturma

Kayıt Ekle Listele/Referans Yüz

Aile Adı

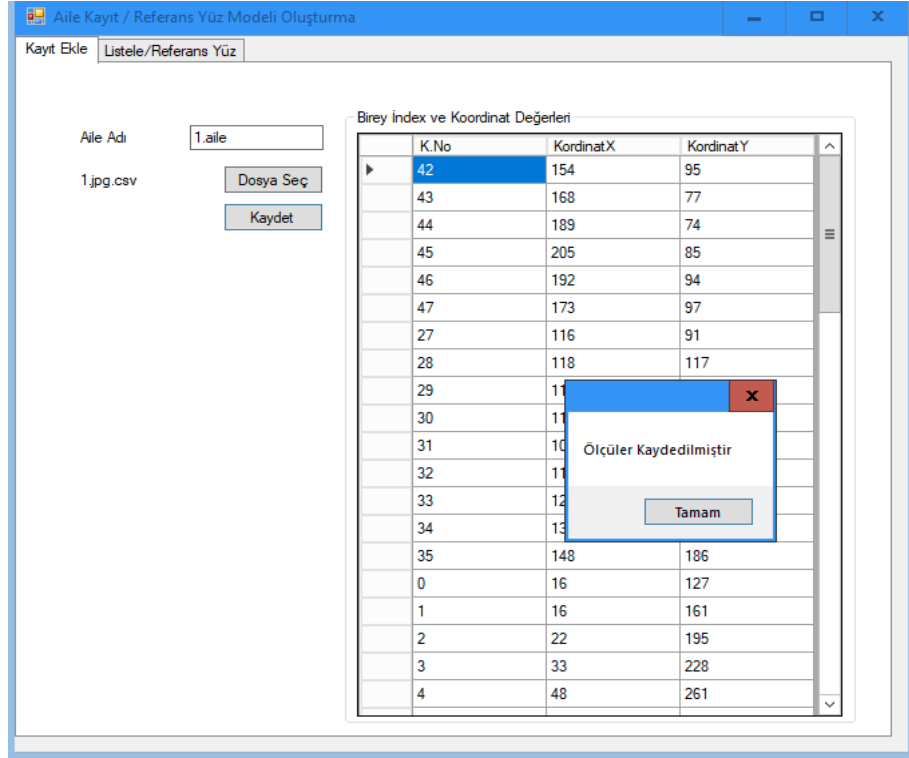
1.jpg.csv Dosya Seç Kaydet

Birey İndex ve Koordinat Değerleri

K.No	KordinatX	KordinatY
42	154	95
43	168	77
44	189	74
45	205	85
46	192	94
47	173	97
27	116	91
28	118	117
29	118	142
30	118	169
31	103	192
32	113	195
33	125	197
34	137	191
35	148	186
0	16	127
1	16	161
2	22	195
3	33	228
4	48	261

Şekil 3.39 Birey index ve koordinat değerlerinin görüntülenmesi.

Aile adı kısmı doldurulduktan sonra *Kaydet* butonuna tıklanarak seçilen birey, belirtilen aile adıyla veri tabanına kaydedilir (Şekil 3.40).



Şekil 3.40 Birey index ve koordinat değerlerinin veri tabanına kaydedilmesi.

### 3.2.5. Yüz nirengi noktaları arasındaki mesafenin hesaplanması ve kaydedilmesi

Yüz nirengi noktaları arasındaki uzaklıkları hesaplamak için, uzaklık ölçü türleri arasında en yaygın kullanılan Öklid uzaklık yöntemi tercih edilmiştir. Uzaklıklar hesaplanırken yüz üzerindeki organlar (kaşlar, gözler, ağız, burun ve yüz bölgesi sınırı) bireysel olarak dikkate alınıp uzaklık hesabı yapılmıştır. Bu değerlerin hesaplanıp veri tabanına kaydedilmesi için C# programlama dilinde hazırlanan programdaki Öklid Uzaklık sekmesi kullanılmıştır. Programın ara yüzü Şekil 3.41’de gösterilmiştir.

K.No	KordinatX	KordinatY	Öklid
42	154	95	22,8
43	168	77	21,21
44	189	74	19,42
45	205	85	15,81
46	192	94	19,24
47	173	97	19,1
27	116	91	26,08
28	118	117	25
29	118	142	27
30	118	169	27,46
31	103	192	10,44
32	113	195	12,17
33	125	197	13,42
34	137	191	12,08
35	148	186	0
0	16	127	34
1	16	161	34,53
2	22	195	34,79
3	33	228	36,25
4	48	261	36,72

Şekil 3.41 Öklid uzaklık hesaplama ve kayıt ara yüzü.

Aile adı kısmına kaydedilecek ailenin adı yazılır. Daha sonra *Dosya Seç* butonuna tıklanarak Şekil 3.38’de gösterilen diyalog penceresinden belirtilen aileye ait birey seçilir.

Seçilen bireye ait .csv formatındaki dosyanın adı, *Seçilen Dosya* etiketi kısmında görüntülenir. Ayrıca seçilen bireye ait index, koordinat ve Öklid uzaklık değerleri *Birey Index, Koordinat ve Öklid uzaklık Değerleri* kısmında görüntülenir (Şekil 3.42).

Aile adı kısmı doldurulduktan sonra *Kaydet* butonuna tıklanarak seçilen birey, belirtilen aile adıyla veri tabanına kaydedilir.

Yüz nirengi noktaları arasındaki Öklid uzaklıkları veri tabanına kaydedildikten sonra hazırlanan programla bu ölçüler kontrol edilebilir.

Şekil 3.42 Öklid uzaklık kontrol program ara yüzü.

Şekil 3.42’de gösterilen programda listeleme sekmesi ile veri tabanında kayıtlı olan tüm aile ve aile bireyleri görüntülenebilir. Hem aile adına göre hem de tüm kayıtlar listelenebilir. Tüm kayıtlar seçilip listele butonuna tıklandığında Şekil 3.43’te gösterildiği gibi veri tabanındaki tüm aile kayıtları listelenir.

Aile Kayıt / Referans Yüz Modeli Oluşturma

Kayıt Ekle Öklid Uzaklık **Listele/Referans Yüz**

Listeleme

☐ Aile Adına Göre

☒ Tümüü

AileAdi	bireyID
a	37
a	38
a	39
a	40
a10	54
a11	55
a12	56
a13	57
a14	58
a15	59
a16	60
a5	41
a5	42
a5	43
a5	44
a5	45
a8	51
a9	53

Sol Kaş Uzunluk Değerleri

Sağ Kaş Uzunluk Değerleri

Sol Göz Uzunluk Değerleri

Sağ Göz Uzunluk Değerleri

Burun Uzunluk Değerleri

Yüz Uzunluk Değerleri

Ağız Uzunluk Değerleri

Şekil 3.43 Aile birey Öklid uzaklık listeleme ara yüzü.

Listelenen tüm kayıtlar içinden Öklid uzaklıkları kontrol edilecek kayda çift tıklanarak, o kaydın tüm Öklid uzaklık değerleri listeleme bölümlerine dolar (Şekil 3.44).

Şekil 3.44 Öklid uzaklık değerlerinin listeleme bölümlerine aktarılması.

### 3.2.6. Aile yüz referans modelinin oluşturulması

Veri tabanına kaydedilen her bir aileyi temsil edecek referans yüz modellerinin oluşturulması işlemi bu bölümde anlatılmıştır. Referans yüz modelinin oluşturulması için önceki bölümde hesaplanıp veri tabanına kaydedilen Öklid uzaklıkları ve merkezi eğilim ölçülerinden, koordinat değerleri arasında uç ve tekrarlanan değerlerin olmamasından dolayı, aritmetik ortalama kullanılmıştır.

Aritmetik ortalama merkezi eğilim ölçüleri arasında en çok kullanılanıdır. “Eş. 3.37” gösterildiği gibi birimlerin belirli bir değişkene göre aldıkları değerlerin birimdeki elemanların sayısına bölünerek hesaplanır (Anonim, 2020f).

$$\bar{X} = \frac{x_1 + x_2 + \dots + x_n}{n} \quad ya \ da \quad \bar{X} = \frac{\sum x}{n} \quad (3.37)$$

Referans yüz modelinin oluşturulması için hazırlanan programın ara yüzü Şekil 3.45’te gösterilmiştir.

Şekil 3.45 Referans yüz modeli oluşturma ara yüzü.

Referans yüz modeli oluşturma sekmesinde üç bölüm bulunmaktadır:

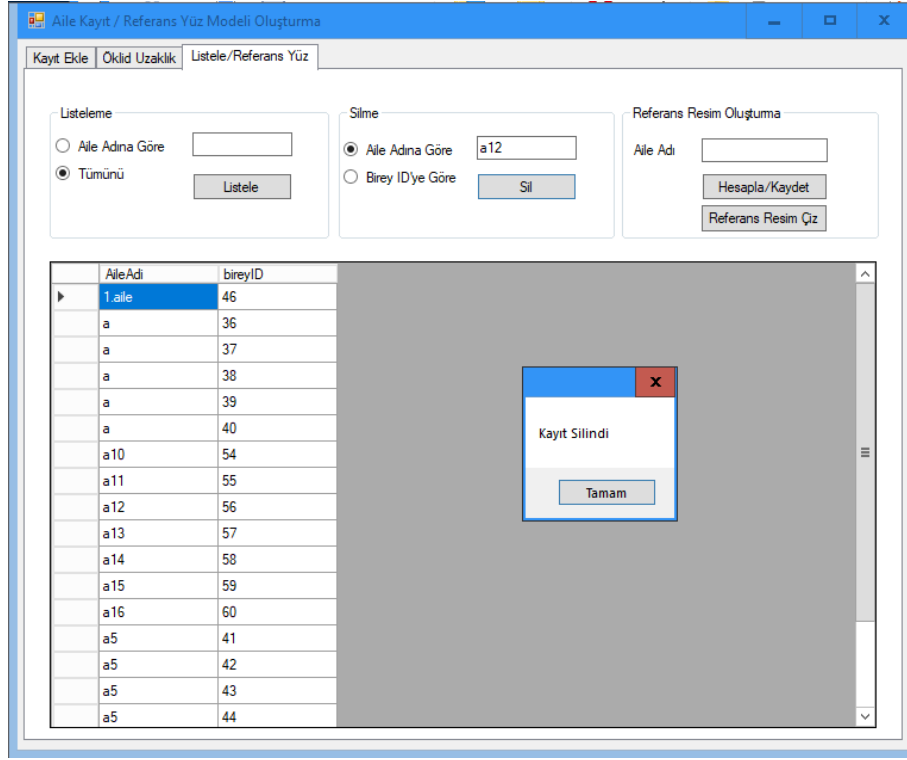
- **Listeleme bölümü:** Veri tabanında kayıtlı olan ailelerin listelenmesi sağlanır. Bu listeleme işlemi, ister aile adına göre istenirse tüm kayıtlar olarak yapılabilir. Listeleme bölümündeki *Aile Adına Göre* seçeneği seçilip bir aile adı yazıldıktan sonra *listele* butonuna tıklanarak o ailedeki tüm bireylerin listelenmesi sağlanır (Şekil 3.46).

AileAdı	bireyID
a5	41
a5	42
a5	43
a5	44
a5	45

Şekil 3.46 Aile adına göre listeleme ara yüzü.

- Silme bölümü: Veri tabanında kayıtlı olan ailelerin veya bireylerin silinmesi için kullanılır. Bu silme işlemi, ister aile adına göre (tüm ailenin silinmesi) istenirse birey ID'ye göre yapılabilir. Silme bölümündeki *Aile Adına Göre* seçeneği seçilip bir aile adı yazıldıktan sonra *Sil* butonuna tıklanarak o ailedeki tüm bireylerin silinmesi sağlanır ya da *Birey ID'ye Göre* seçeneği seçilip silinecek bireyin ID'si yazıldıktan sonra *Sil* butonuna tıklanarak bireyin silinmesi sağlanır (Şekil 3.47).





Şekil 3.47 Aile ya da birey silme ara yüzü.

- Referans Resim Oluşturma bölümü: Referans yüzü modeli oluşturulacak ailenin adı *Aile Adı* kısmına yazılır. Daha sonra Hesapla/Kaydet butonuna tıklanarak Öklid uzaklık ve aritmetik ortalama değerleri kullanılarak aileye özgü referans yüz modelinin koordinat değerleri Şekil 3.48’de görüldüğü gibi listelenir.

Aile Kayıt / Referans Yüz Modeli Oluşturma

Kayıt Ekle Öklid Uzaklık Listele/Referans Yüz

Listeleme

☐ Aile Adına Göre

☒ Tümünü

Silme

☒ Aile Adına Göre

☐ Birey ID'ye Göre

Referans Resim Oluşturma

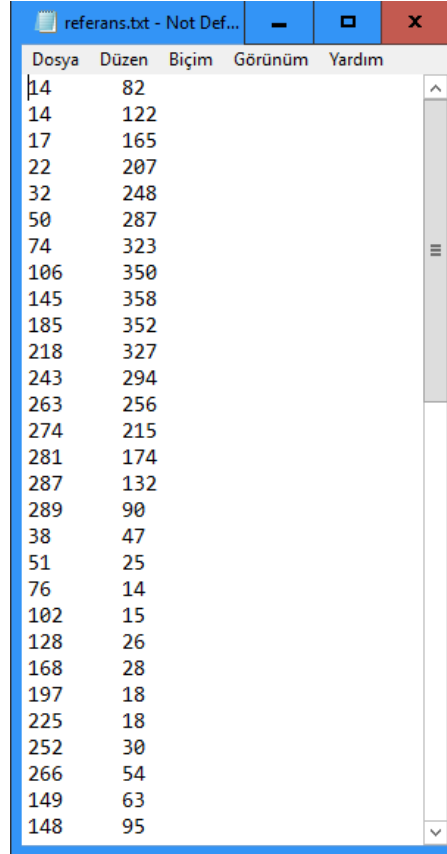
Aile Adı

	AileAdı	KNO	ortX	ortY
▶	a5	0	14	82
	a5	1	14	122
	a5	2	17	165
	a5	3	22	207
	a5	4	32	248
	a5	5	50	287
	a5	6	74	323
	a5	7	106	350
	a5	8	145	358
	a5	9	185	352
	a5	10	218	327
	a5	11	243	294
	a5	12	263	256
	a5	13	274	215
	a5	14	281	174
	a5	15	287	132
	a5	16	289	90

Referans Ölçüler Kaydedildi

Şekil 3.48 Aile referans yüz koordinat değerlerinin hesaplanması ve listelenmesi.

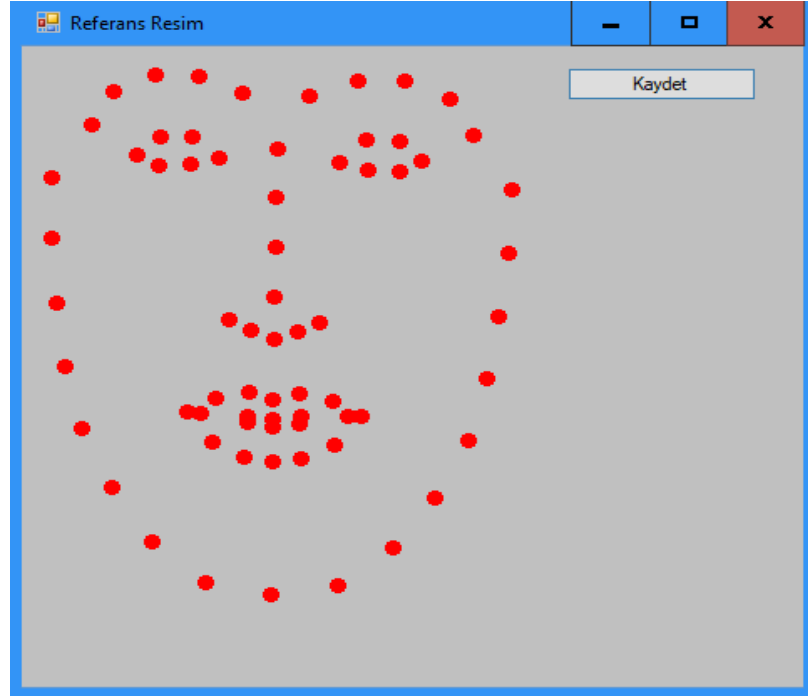
Listelenen bu koordinat değerleri, *referans.txt* adında bir metin dosyası oluşturularak içine kaydedilir (Şekil 3.49).



Dosya	Düzen	Biçim	Görünüm	Yardım
14	82			
14	122			
17	165			
22	207			
32	248			
50	287			
74	323			
106	350			
145	358			
185	352			
218	327			
243	294			
263	256			
274	215			
281	174			
287	132			
289	90			
38	47			
51	25			
76	14			
102	15			
128	26			
168	28			
197	18			
225	18			
252	30			
266	54			
149	63			
148	95			

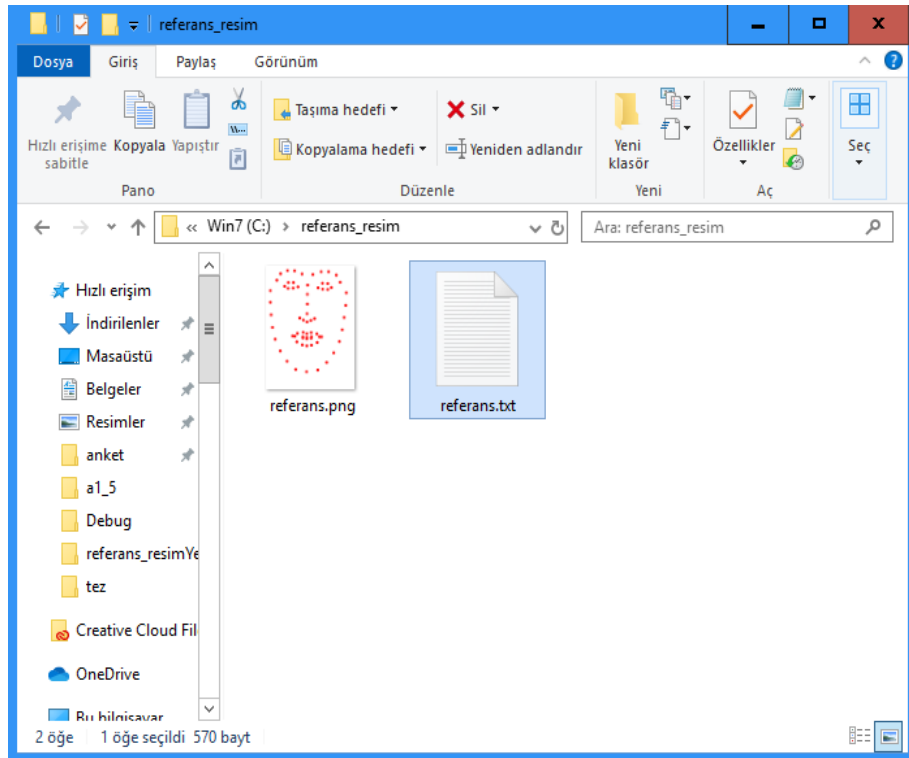
Şekil 3.49 Referans yüz koordinat değerlerinin kaydedildiği dosya içeriğinin bir kısmı.

Kaydetme işlemi tamamlandıktan sonra bu koordinat değerlerine göre referans yüz modeli çizdirilebilir. Referans Resim Oluşturma bölümündeki *Referans Resim Çiz* butonu tıklanarak referans yüz modeli çıkartılır (Şekil 3.50).



Şekil 3.50 Referans yüz modeli.

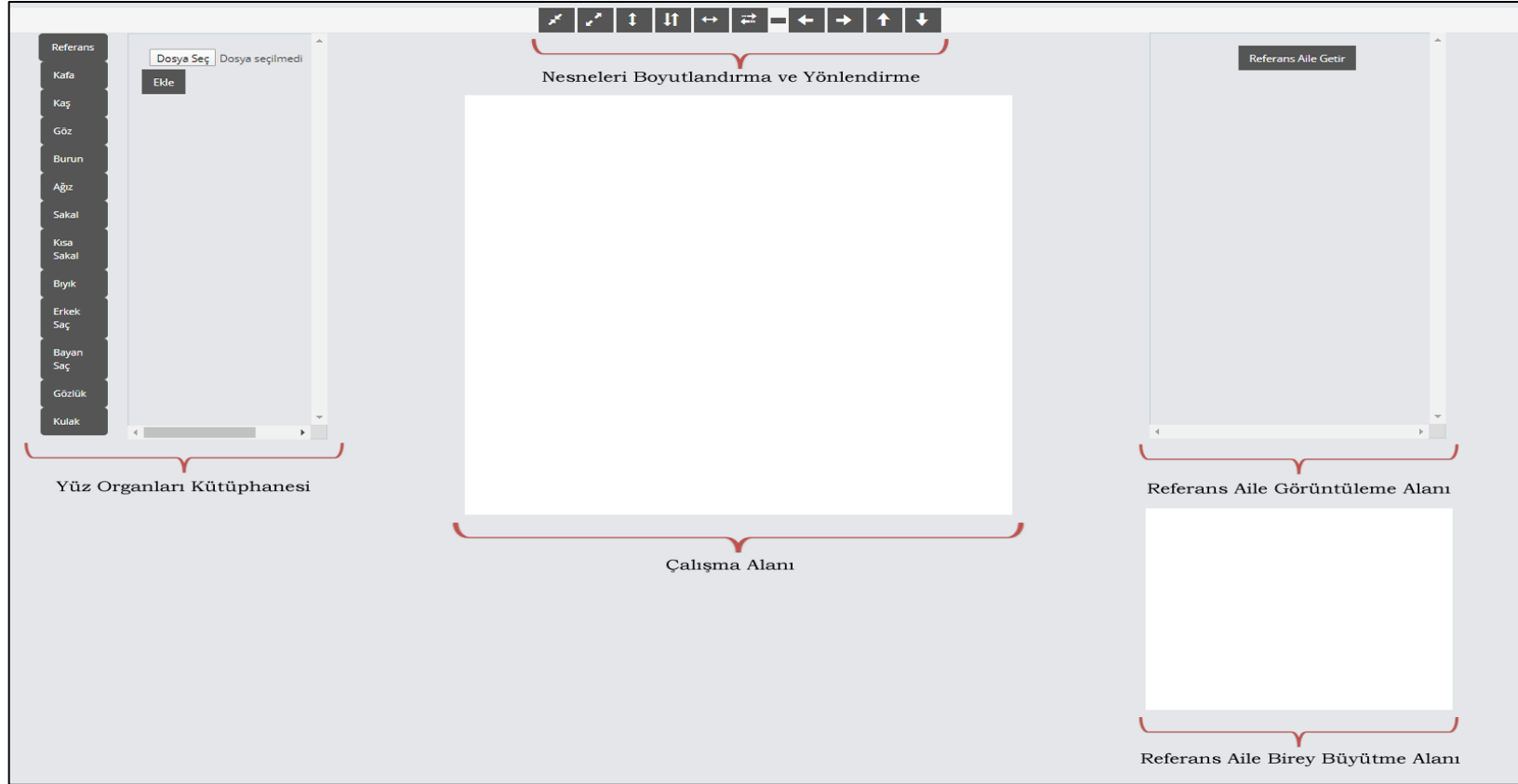
Şekil 3.50’de görüldüğü gibi referans resim oluşturulduktan sonra kaydet butonuna tıklanarak seçilen aileye özgü oluşturulan yüz referans modeli *.png* formatında C:\referans\_resim klasörüne kaydedilir (Şekil 3.51).



Şekil 3.51 Referans yüz koordinat değerlerinin ve resminin kaydedilmesi.

### **3.2.7. Yüz giydirme işlemi**

Hazırlanan sistemin son adımı, yüz giydirme işlemi yani robot resim çizimidir. Robot resmin çizdirilebilmesi için html tabanlı bir betik dil olan JavaScript kullanılmıştır. JavaScript, web sayfalarının iyileştirilmesi, işlevselliğinin arttırılması ve dinamik sayfaların hazırlanması amacıyla kullanılan bir betik dildir. Robot resmin çizimi için hazırlanan programın ara yüzü Şekil 3.52’de gösterilmektedir.

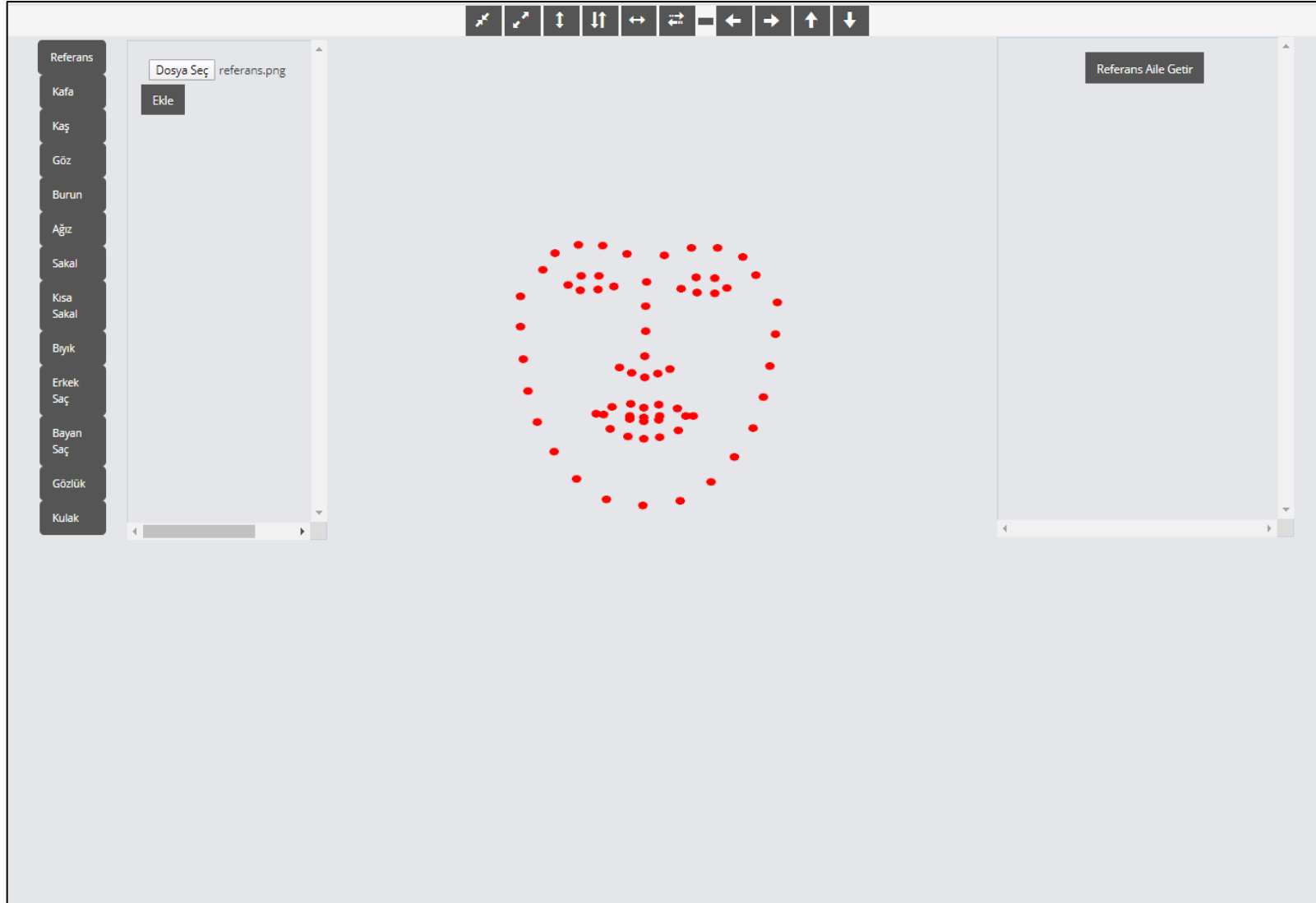


Şekil 3.52 Robot resim çizme program ara yüzü.

- *Yüz organları kütüphanesi:* Robot resmin çizilebilmesi için yüz organlarının görüntülerini içeren bir yüz organı kütüphanesi oluşturulmuştur (Çizelge 3.3 ve Çizelge 3.4). Bu kütüphane içinde yüzü oluşturan organlar, yüzün ve kafanın şekillenmesini sağlayan nitelikler ve yüz üzerindeki organlarda kullanılan bazı aksesuarlar bulunmaktadır. Toplam 12 farklı bölümden oluşan bu kütüphanede 118 adet görüntü bulunmaktadır. Bu bölümler:

- Kafa modelleri
- Kaş modelleri
- Göz modelleri
- Burun modelleri
- Ağız modelleri
- Sakal modelleri
- Kısa sakal modelleri
- Bıyık modelleri
- Erkek saç modelleri
- Bayan saç modelleri
- Gözlük modelleri
- Kulak modelleri





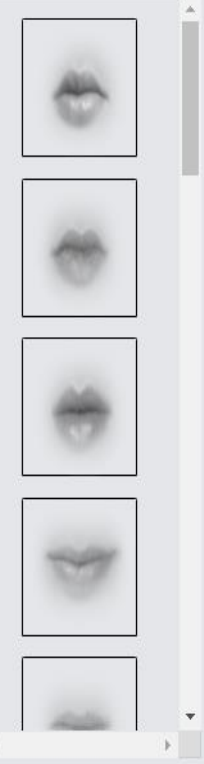

Yüz organları kütüphanesi bölümünde bulunan *Referans* bölümü ile; bir önceki bölümde anlatılan, (C# programlama dilinde hazırlanan aile yüz referans modeli oluşturma programı ile) ailedeki bireylere göre oluşturulan (Şekil 3.39) modelin çalışma alanına eklenmesi sağlanmaktadır. *Dosya Seç* butonuna tıklanarak açılan pencereden referans modelinin kayıtlı olduğu klasöre gidilip referans resim seçildikten sonra *Ekle* butonuna tıklanarak referans resim çalışma alanına eklenir (Şekil 3.53).







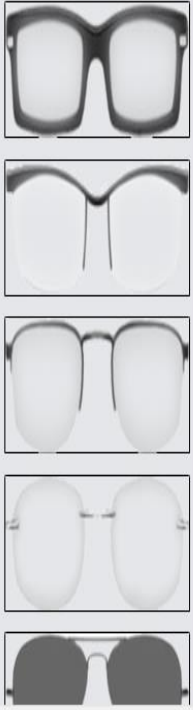

Şekil 3.53 Aile referans yüz modelinin çalışma alanına eklenmesi.



Çizelge 3.3 Yüz organ kütüphanesi bölümlerinin bir kısmı

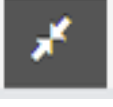




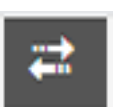


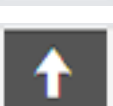

Kafa Modelleri	Kaş Modelleri	Göz Modelleri	Burun Modelleri	Ağız Modelleri	Sakal Modelleri
					

Çizelge 3.4 Yüz organ kütüphanesi bölümlerinin devamı

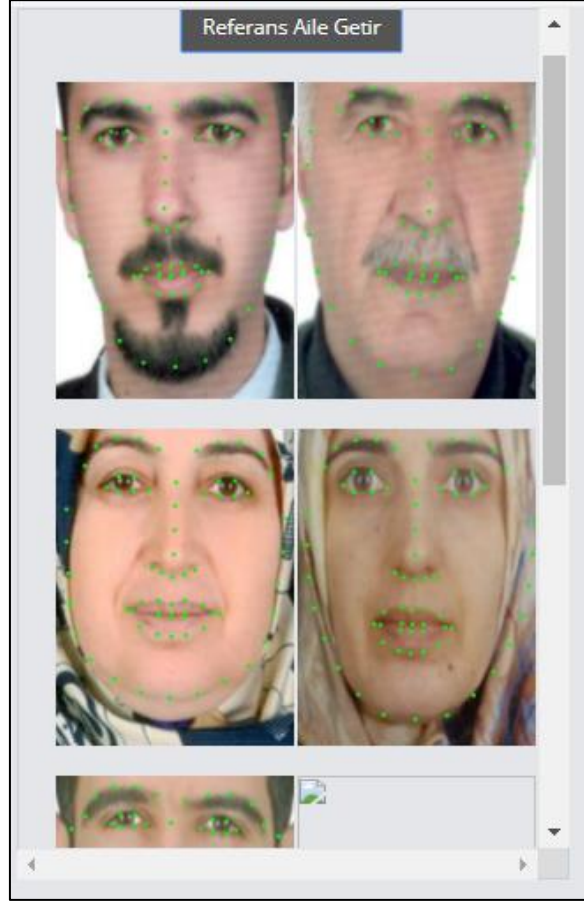
Kısa Sakal Modelleri	Bıyık Modelleri	Erkek Saç Modelleri	Kadın Saç Modelleri	Gözlük Modelleri	Kulak Modelleri
					

- *Nesneleri boyutlandırma ve yönlendirme:* Bu alanda bulunan araç butonları ile çalışma alanına eklenen nesnelerin boyutları değiştirilebilir aynı zamanda çalışma alanı üzerinde yönlendirme yapılabilir. Çizelge 3.4’te araç butonları ve görevleri verilmiştir.

Çizelge 3.5 Nesneleri boyutlandırma ve yönlendirme araç butonları

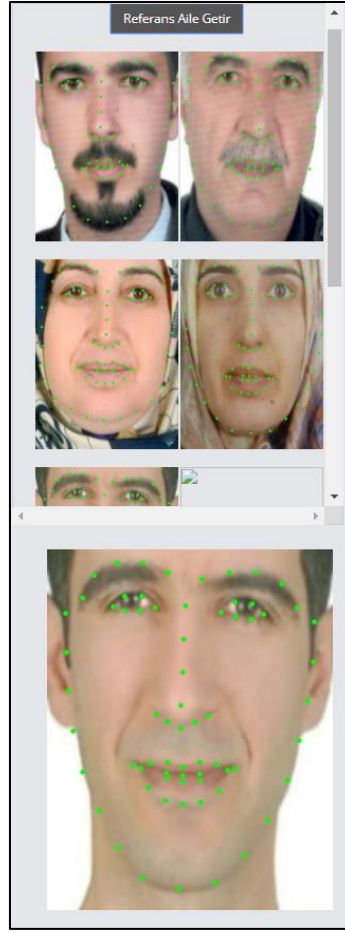
Buton	Görevi
	Seçili olan nesneye ait hem genişlik hem de yükseklik değerlerini azaltır.
	Seçili olan nesneye ait hem genişlik hem de yükseklik değerlerini artırır.
	Seçili olan nesneye ait yükseklik değerini artırır.
	Seçili olan nesneye ait yükseklik değerini azaltır.
	Seçili olan nesneye ait genişlik değerini artırır.
	Seçili olan nesneye ait genişlik değerini azaltır.
	Seçili olan nesneyi çalışma alanında sola doğru kaydırır.
	Seçili olan nesneyi çalışma alanında sağa doğru kaydırır.
	Seçili olan nesneyi çalışma alanında yukarı doğru kaydırır.
	Seçili olan nesneyi çalışma alanında aşağı doğru kaydırır.

- *Referans aile görüntüleme alanı:* Yüz giydirme işlemi yapılırken aile bireylerinin incelenmesi için aile yüz referans modeli çıkartılan aile üyelerinin resimleri bu kısımda görüntülenir (Şekil 3.54).



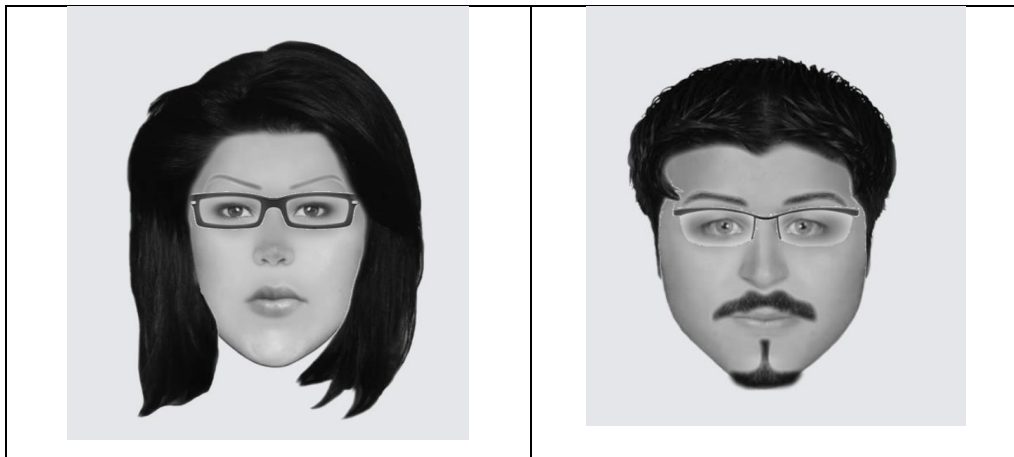
Şekil 3.54 Referans aile görüntüleme alanı

- *Referans aile birey büyütmeye alanı:* Yüz giydirme işlemi yapılırken referans aile görüntüleme alanından seçilen aile bireyi görüntüsünün büyük hali bu alanda görüntülenir (Şekil 3.55).

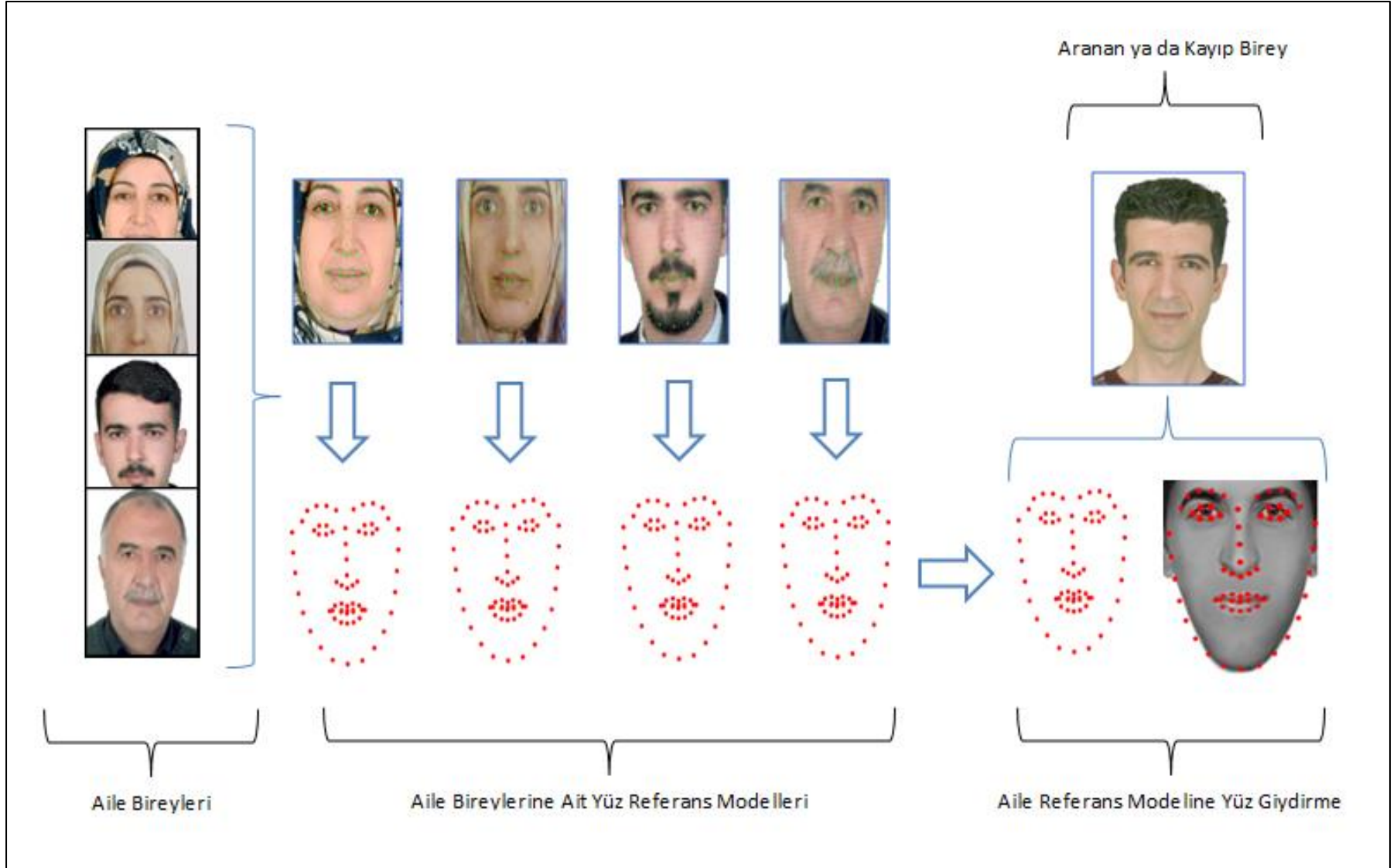


Şekil 3.55 Referans aile birey büyütme alanı

- *Çalışma alanı:* Yüz giydirme işlemi için kullanılacak nesnelerin yerleştirildiği alandır. Bu alan üzerinde her türlü taşıma, boyutlandırma ve silme işlemleri gerçekleştirilebilir. Şekil 3.56’da robot resmi çizilen iki aileye ait kadın ve erkek örnek modeller gösterilmektedir.



Şekil 3.56 Program kullanılarak oluşturulan örnek robot resimler.



Şekil 3.57 Sistemin çalışmasını gösteren genel blok diyagramı.

Şekil 3.57’de sistemin genel yapısı gösterilmektedir. Aile bireylerinden alınan resimler görüntü işleme teknikleri kullanılarak düzenlenir. Makine öğrenmesi yöntemlerinden ESA kullanılarak görüntülerdeki yüzler ve yüz üzerindeki organları gösteren yüz nirengi noktaları belirlenir. Bu nirengi noktalarının temsili koordinat değerleri veri tabanına kaydedilir. Geliştirilen yazılım ile her ailedeki bireylerden oluşan, aileyi en uygun şekilde temsil edecek, aile referans modeli oluşturulur. Web tabanlı hazırlanan robot resim oluşturma programında aile referans modeli baz alınarak, aranan/kayıp bireye göre, yüz giydirme işlemi yapılmaktadır.





#### 4. BULGULAR

Yapılan tez çalışması kapsamında farklı veri setleri üzerinde yüzü ve yüz organlarını bulmak için makine öğrenmesi teknikleri kullanılmıştır. Görüntüler üzerindeki yüzün tespiti için birçok yöntem bulunmaktadır. En iyi yöntemin seçilmesi için adına yapılan testlerin sonuçları Çizelge 4.4’de gösterilmektedir.

Çizelge 4.1 ESA yöntemiyle veri setleri üzerinde yapılan test sonuçları

	ESA B.O
FEI Veri Seti	98,86
CelebA Veri Seti	99,89
Aile Veri Seti	100,00

ESA yöntemi; FEI, CelebA ve Aile veri setlerine uygulanarak Çizelge 4.1 gösterilen sonuçlar elde edilmiştir. En yüksek başarının Aile veri seti üzerinde elde edildiği görülmektedir.

Çizelge 4.2 DSA yöntemiyle veri setleri üzerinde yapılan test sonuçları

	DSA B.O.
FEI Veri Seti	97,61
CelebA Veri Seti	95,79
Aile Veri Seti	91,83

DSA yöntemi; FEI, CelebA ve Aile veri setlerine uygulanarak Çizelge 4.2 gösterilen sonuçlar elde edilmiştir. En yüksek başarının FEI veri seti üzerinde elde edildiği görülmektedir.

Çizelge 4.3 YGH-DVM yöntemiyle veri setleri üzerinde yapılan test sonuçları

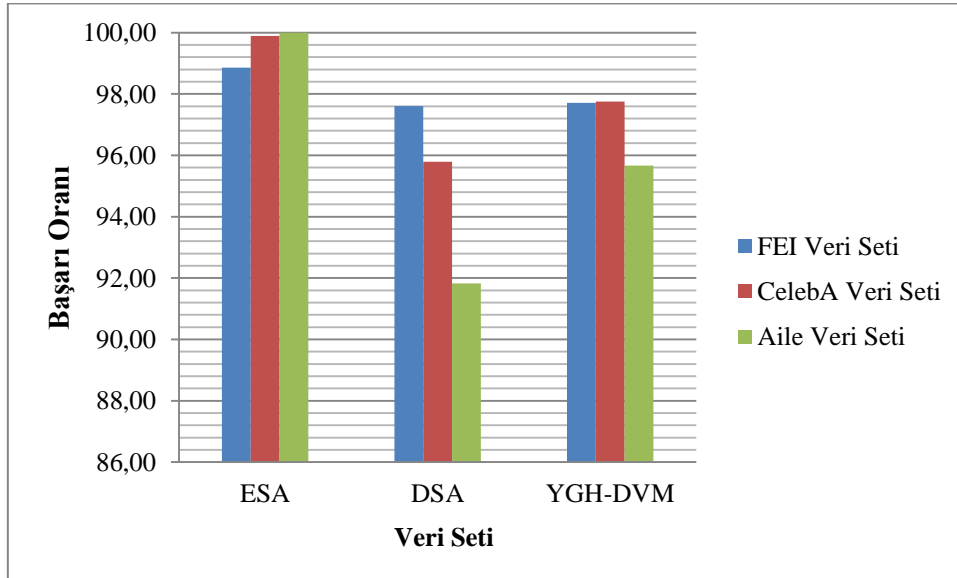
	YGH-DVM B.O.
FEI Veri Seti	97,71
CelebA Veri Seti	97,75
Aile Veri Seti	95,67

YGH-DVM yöntemi; FEI, CelebA ve Aile veri setlerine uygulanarak Çizelge 4.3 gösterilen sonuçlar elde edilmiştir. En yüksek başarının CelebA veri seti üzerinde elde edildiği görülmektedir.

Çizelge 4.4 Uygulanan yöntemlerin veri setlerine göre başarı oranları

		ESA	DSA	YGH-DVM
FEI Veri Seti	Toplam Başarı	2768	2733	2736
	Başarı Oranı	98,86	97,61	97,71
CelebA Veri Seti	Toplam Başarı	2797	2682	2737
	Başarı Oranı	99,89	95,79	97,75
Aile Veri Seti	Toplam Başarı	208	191	199
	Başarı Oranı	100,00	91,83	95,67

En uygun yöntemi seçme adına belirlenen yöntemler ve veri setleri üzerinde yapılan testler Çizelge 4.4'te gösterilmiştir. Bu sonuçlara dayanarak hazırlanan başarı oran grafiği de Şekil 4.1'de gösterilmiştir. Çalışmamızda yüzün tespiti için; ESA, DSA ve YGH-DVM yöntemleri arasından en iyi performansı sergileyen ESA tercih edilmiştir.



Şekil 4.1 Uygulanan yöntemlerin veri setlerine göre başarı oran grafiği.

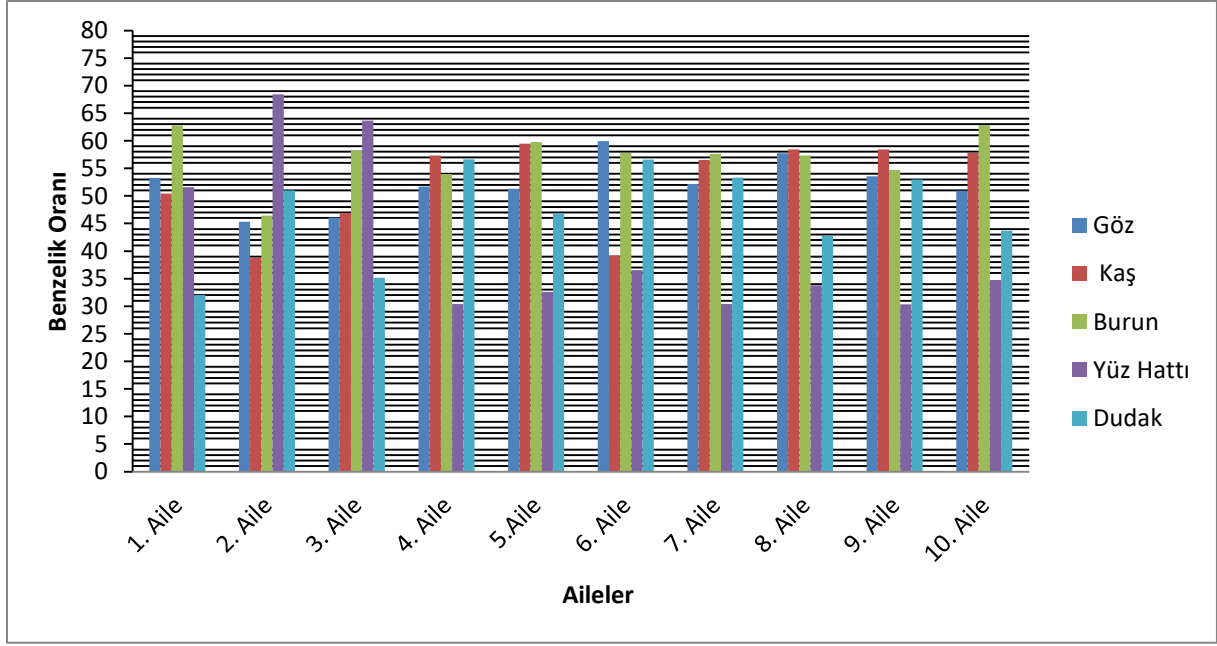
Ailelerdeki bireylerin birbirine ne kadar benzediğini ölçme adına yapılan testlerin sonuçları Çizelge 4.5’te gösterilmiştir.

Çizelge 4.5 Aile yüz organlarının benzerlik oranları

	<b>Göz</b>	<b>Kaş</b>	<b>Burun</b>	<b>Yüz Hattı</b>	<b>Dudak</b>
<b>1. Aile</b>	53,25	50,41	62,77	51,56	32,01
<b>2. Aile</b>	45,32	38,84	46,39	68,42	51,02
<b>3. Aile</b>	46,06	46,89	58,21	63,68	35,16
<b>4. Aile</b>	51,69	57,34	53,95	30,41	56,62
<b>5. Aile</b>	51,31	59,50	59,76	32,60	46,83
<b>6. Aile</b>	59,92	39,19	57,79	36,54	56,55
<b>7. Aile</b>	52,17	56,50	57,61	30,40	53,31
<b>8. Aile</b>	57,72	58,44	57,32	33,78	42,75
<b>9. Aile</b>	53,55	58,41	54,71	30,36	52,97
<b>10. Aile</b>	50,93	57,79	62,86	34,72	43,71

Çizelge 4.5’te ailelerdeki bireylerin yüz organlarının benzerlik oranları verilmektedir.

1. Aile bireyleri arası en benzer organın burun olduğu, 2. Aile bireyleri arası en benzer organın yüz hattı olduğu, 3. Aile bireyleri arası en benzer organın yüz hattı olduğu, 4. Aile bireyleri arası en benzer organın kaş olduğu, 5. Aile bireyleri arası en benzer organın burun olduğu, 6. Aile bireyleri arası en benzer organın göz olduğu, 7. Aile bireyleri arası en benzer organın burun olduğu, 8. Aile bireyleri arası en benzer organın kaş olduğu, 9. Aile bireyleri arası en benzer organın kaş olduğu, 10. Aile bireyleri arası en benzer organın burun olduğu görülmektedir. Benzerlik oranlarını gösteren grafik Şekil 4.2’de verilmiştir.



Şekil 4.2 Aile yüz organlarının benzerlik oranlarını gösteren grafik.

Rastgele seçilen 10 aileden oluşturulan referans modele göre yüz giydirme (robot resim oluşturma) işlemi yapılmıştır. Bu robot resimler çalışmada kullanılan uygulama ile yüz ve yüz nirengi noktaları tespit edilip veri tabanına kaydedilmiştir. Daha sonra her aileden seçilen bireye ait robot resim ile gerçek resmin ne kadar benzediğini test etme adına yüz nirengi noktaları arasındaki mesafe değerleri kıyaslanmıştır. Testlerde tahmin performansının ölçümünde kullanılan istatistiklerden; Hata kareler ortalaması (Mean Squared Error - MSE), Hata kareler ortalamasının karekökü (Root Mean Squared Error - RMSE), Ortalama Mutlak Yüzde Hata (Mean Absolute Percentage Error - MAPE) kullanılmıştır (Çizelge 4.6). Tahmin performanslarını ölçmede kullanılan istatistikler aşağıdaki eşitlikler (Eş. 4.1, Eş. 4.2, Eş. 4.3, Eş. 4.4 ve Eş. 4.5) ile elde edilmektedir (Vandeput, 2019).

$$e_t = f_t - d_t \quad (4.1)$$

$$\text{Bias} = \frac{1}{n} \sum_n e_t \quad (4.2)$$

$$\text{MAPE} = \frac{1}{n} \sum \frac{|e_t|}{d_t} \quad (4.3)$$

$$RMSE = \sqrt{\frac{1}{n} \sum e_t^2} \quad (4.4)$$

$$MSE = \frac{1}{n} \sum e_t^2 \quad (4.5)$$

$e_t$  : Hata değeri

$n$  : Tahmin adeti

Çizelge 4.6 Tahmin performansı hata ölçüm değerleri

	MAPE	RMSE	MSE
1. Aile	10.34	3.11	9.66
2. Aile	7.14	2.28	5.19
3. Aile	7.41	2.37	5.60
4. Aile	6.10	1.86	3.45
5. Aile	10.20	3.31	10.95
6. Aile	11.23	3.62	13.12
7. Aile	10.64	3.52	12.37
8. Aile	7.41	2.10	4.42
9. Aile	9.74	3.02	9.10
10. Aile	6.88	1.98	3.93

Tahmin doğruluğunu ölçmek için en sık kullanılan istatistiksel yöntem MAPE'dir. MAPE, bireysel mutlak hataların talebe (her döneme ayrı ayrı) bölünmesiyle elde edilir. Aslında, yüzde hatalarının ortalamasıdır (Vandeput, 2019).

SSIM (Structural Similarity - Yapısal Benzerlik) iki görüntü arasındaki yapısal benzerliği yani algısal kaliteyi ölçmede kullanılan bir yöntemdir. Bu yöntem görüntünün algısal varyasyonundaki değişimleri tespit etmektedir. "Eş. 4.6" ile ifade edilmektedir (Kaş, 2017; Dhall, 2011).

$$SSIM(w_1, w_2) = \frac{[2\mu_{w_1}\mu_{w_2} + C_1][2\sigma_{w_1w_2} + C_2]}{[\mu_{w_1}^2 + \mu_{w_2}^2 + C_1][\sigma_{w_1}^2 + \sigma_{w_2}^2 + C_2]} \quad (4.6)$$

$\mu_{w_1}$  :  $w_1$ 'in ortalaması

$\mu_{w_2}$  :  $w_2$ 'in ortalaması

$\sigma_{w_1}^2$  :  $w_1$ 'in varyansı

$\sigma_{w_2}^2$  :  $w_2$ 'in varyansı

$\sigma_{w_1w_2}$  :  $w_1$  ve  $w_2$ 'nin kovaryansı





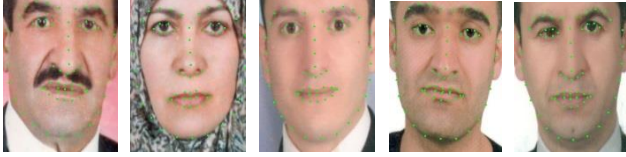















Çizelge 4.7 Rastgele seçilen 10 aile üzerinde SSIM yöntemiyle yapılan test sonuçları

<b>Aileler</b>	<b>SSIM</b>
1. Aile	0,805
2. Aile	0,705
3. Aile	0,511
4. Aile	0,594
5. Aile	0,629
6. Aile	0,585
7. Aile	0,664
8. Aile	0,659
9. Aile	0,501
10. Aile	0,598




















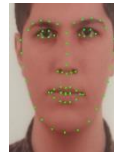
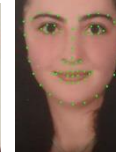






















Çizelge 4.7 İncelendiğinde SSIM yöntemi uygulanan 10 ailenin test sonuçlarında en yüksek değer sahip aile 1. Ailedir. SSIM yöntemi yapısal benzerliği dikkate aldığı için benzerlik değerleri düşük çıkmıştır. Tezde amaçlanan, Öklid uzaklığı kullanılarak benzerlik değerlerinin hesaplanmasıdır. Çizelge 4.6'da Öklid uzaklığı kullanılarak hesaplanan benzerlik değerleri gösterilmektedir.

Yüz giydirme işlemi için rastgele seçilen 10 ailenin ve o ailelerden seçilen bireylerin robot resimleri Çizelge 4.8'de gösterilmektedir.

Çizelge 4.8 Rastgele seçilen 10 aile ve o ailelerden seçilen bireylerin robot resimleri

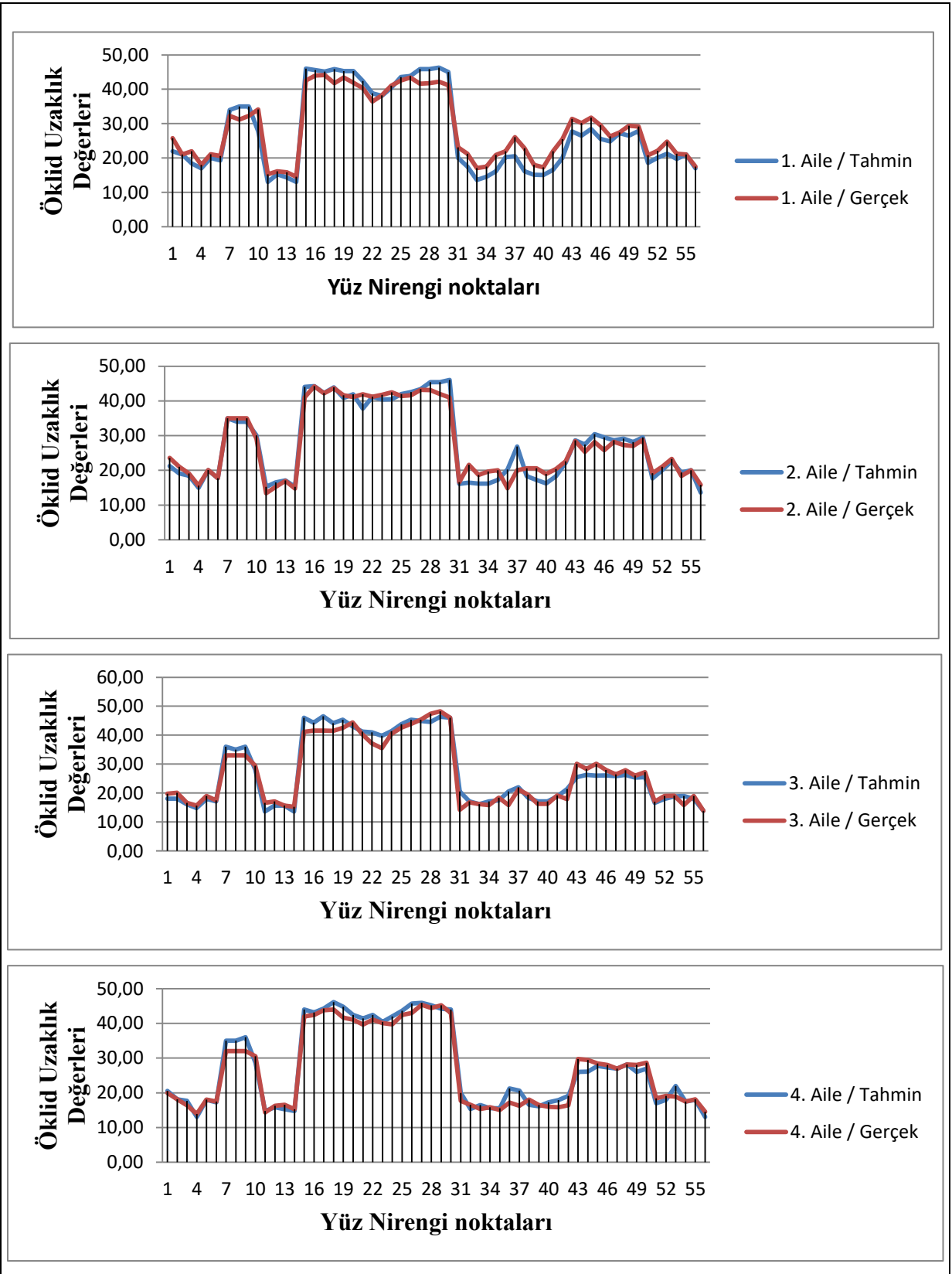
Aileler	Aile Resimleri					Aranan – Kayıp Birey		Robot Resmi
1. Aile								
2. Aile								
3. Aile								
4. Aile								
5. Aile								

Çizelge 4.9 Rastgele seçilen 10 ailenin ve o ailelerden seçilen bireylerin robot resimleri (devamı)

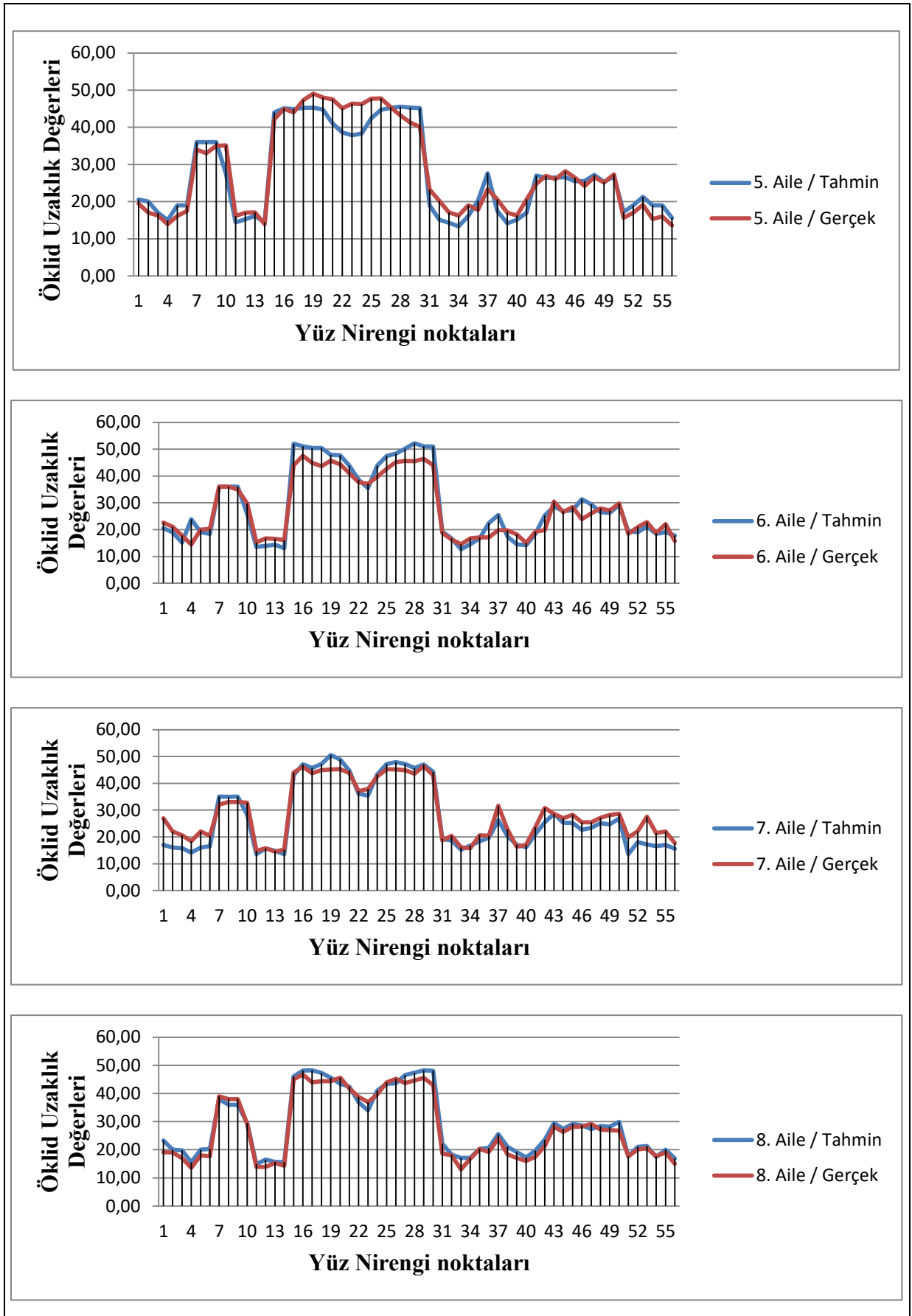
Aileler	Aile Resimleri						Aranan – Kayıp Birey		Robot Resmi		
6. Aile											
7. Aile											
8. Aile											
9. Aile											
10. Aile											



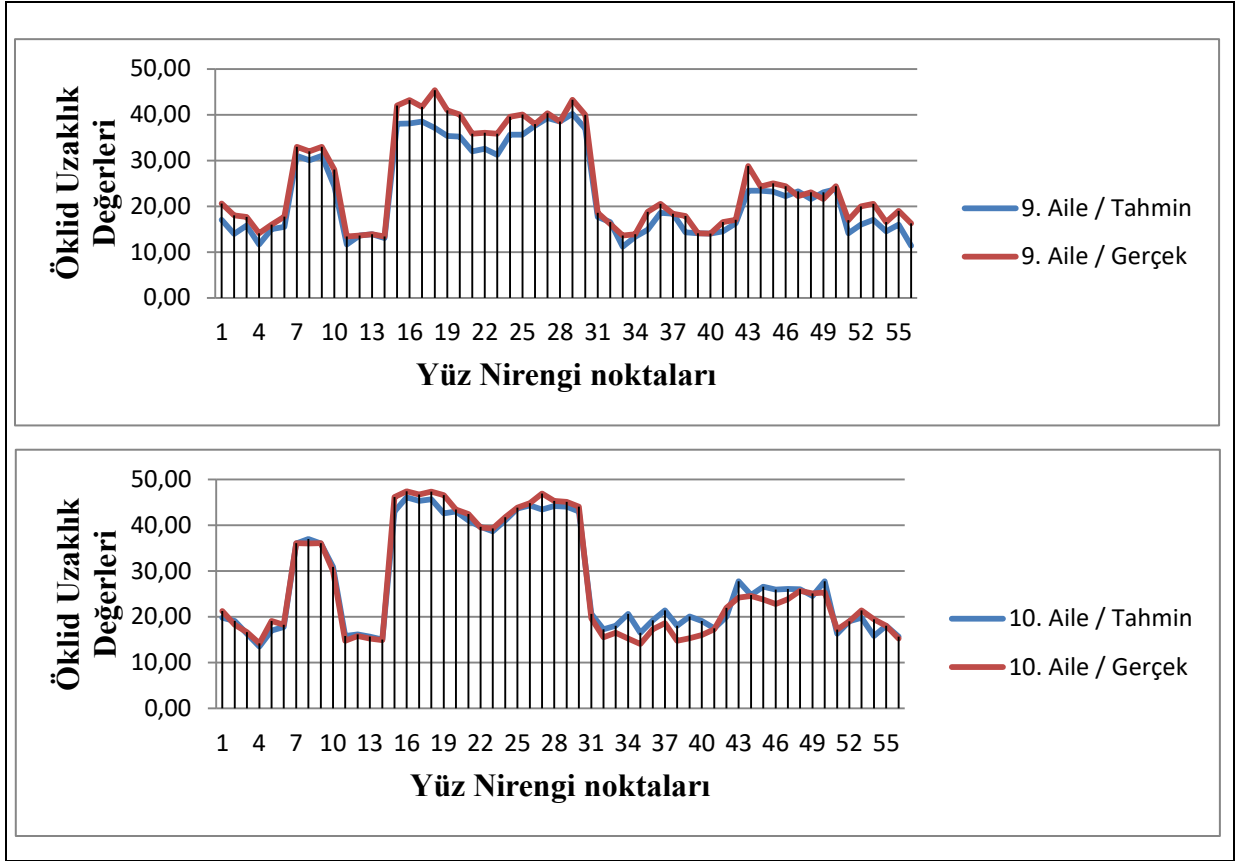
Şekil 4.3, Şekil 4.4 ve Şekil 4.5’de tüm ailelerin tahmin/gerçek değer grafikleri verilmiştir.



Şekil 4.3 1- 4 Aile tahmin/gerçek değer grafikleri.



Şekil 4.4 5 – 8 Aile tahmin/gerçek değer grafikleri.



Şekil 4.5 9. ve 10. Aile tahmin/gerçek değer grafikleri.

Aileler arasından rasgele seçilen 4 aile için sorgulama yapılarak tahmin doğruluk değerleri hesaplanmıştır. Bir aileden elde edilen tahmin değerleri ile başka bir aileden bir bireyin gerçek değerleri karşılaştırılmıştır. Elde edilen analiz sonuçları Çizelge 4.10'da verilmiştir.

Çizelge 4.10 Dört aile benzerlik sorgulama tahmin performansı hata değerleri

	MAPE	RMSE	MSE
1. Aile - 8. Aile Bireyi	16.84	5.16	26.67
4. Aile - 10. Aile Bireyi	22.75	6.97	48.61
9. Aile - 1. Aile Bireyi	20.56	6.21	38.61
6. Aile - 9. Aile Bireyi	21.51	6.81	46.47



## 5. TARTIŞMA VE SONUÇ

Bu tez çalışmasında; kaybolan, kaçırılan ya da aranan bireylerin bulunması temeline dayanan içerik tabanlı bir görüntü erişim sistemi tasarlanarak yüz tahmini yapılmıştır. Makine öğrenme teknikleri kullanarak görüntüler üzerinde yüz ve yüz organları tespit edilmiştir. En iyi performansı elde etmek için makine öğrenme yöntemlerinden olan ESA, DSA ve YGH-DVM yöntemleri FEI, CelebA ve Aile veri setlerine uygulanmıştır. Elde edilen sonuçlara göre en iyi performansı sergileyen yöntem tezde kullanılmıştır. Çizelge 4.4 incelendiğinde Aile veri seti üzerinde en yüksek başarıya ulaşan yöntemin %100 başarı oranı ile ESA olduğu görülmektedir. Bundan dolayı geliştirilen sistemde ESA yöntemi kullanılmıştır.

Bu tez çalışmasında kullanılmak üzere 39 aileden birey görüntüleri toplanmıştır. Literatürde bu tarz bir veri seti bulunmadığı için oluşturulan veri seti ve yapılan çalışma alanında bir ilki temsil etmektedir. Yapılan çalışma ile aile yüz veri setindeki bireylerin görüntülerine ESA yöntemi uygulanarak yüz ve yüz organlarının temsili koordinat değerleri bulunmuştur. Daha sonra Öklid uzaklık yöntemi ile bu koordinat değerlerinin arasındaki mesafeler tesbit edilerek yüz organlarının ölçüleri hesaplanmıştır. Sonraki adımda bu ölçüm değerleri kullanılarak aileye özgü referans bir yüz modeli çıkartılmıştır. Son adımda ise bu modele göre yüz giydirme işlemi yapılarak robot resim oluşturulmuştur.

Rastgele seçilen 10 aileden oluşturulan referans modele göre yüz giydirme (robot resim oluşturma) işlemi yapılmıştır. Bu robot resimlerin her aileden rastgele seçilen bir bireye ne kadar benzediği test edilmiştir. Tahmin doğruluğunu ölçmek için en sık kullanılan istatistiksel yöntem olan MAPE tercih edilmiştir. MAPE, bireysel mutlak hataların talebe (her döneme ayrı ayrı) bölünmesiyle elde edilmektedir. Aslında, hata yüzdelerinin ortalamasıdır (Vandeput, 2019). Çizelge 4.6 incelendiğinde en düşük hata orana sahip aile 6,10 ile dördüncü ailedir. En yüksek hata değerine sahip aile ise 11,23 ile altıncı ailedir. MAPE değeri %10'un altında olan modeller “çok iyi” olarak sınıflandırılmaktadır (Lewis, 1982;Turhan ve ark., 2013).

Şekil 4.3, Şekil 4.4 ve Şekil 4.5'te gösterilen grafikler incelendiğinde seçili ailenin yüz referans modeli ve aile bireylerinden biri kıyaslanmaktadır ve yüz nirengi noktaları arasındaki mesafelerin ne kadar yakın olduğu görülmektedir. Bu durum tahmin model ile gerçek modelin birbirine benzediğini ve aynı aileler arasında yüz benzerliğinin yüksek olduğunu göstermektedir.

Çizelge 4.7’de gösterildiği gibi aileler arasından rasgele seçilen 4 aile için sorgulama yapılarak tahmin doğruluk değerleri hesaplanmıştır. Bir aileden elde edilen tahmin değerleri ile başka bir aileden bir bireyin gerçek değerleri kıyaslanmaktadır. Test sonuçları incelendiğinde hata oranlarının yüksek olduğu görülmektedir. Yani, rastgele seçilen bir ailenin yüz referans modeli ile bir başka aileden seçilen bir bireyin kıyaslanmasında yüz nirengi noktaları arasındaki mesafelerin uzak olduğu anlaşılmaktadır. Bu durum tahmin modeli ile gerçek modelin birbirine benzemediğini ve farklı aileler arasında yüz benzerliğinin düşük olduğunu göstermektedir.

Aile içi benzerliği ölçme adına yapılan test sonuçları Çizelge 4.5’te gösterilmektedir. Bu çizelge incelendiğinde, aileler arasında en çok benzerlik gösteren organın burun, en az benzerlik gösteren organın ise dudak olduğu görülmektedir. Ayrıca aileler arasında en çok benzerlik oranını %68,42 ile 2. ailedeki yüz hattı oluşturmaktadır. Şekil 4.2’de bu ailelerin benzerlik oran grafiği verilmektedir.

Talandova ve ark. (2016), aile içindeki bireylerin yüz benzerliklerini ölçmek için Biometric Device Broadway 3d aygıtının Turnstile Enrolment Application (TEA) yazılımını kullanmışlardır. Testler için iki aileye ait veri seti kullanmışlardır. Birinci derece akrabalarda benzerlik oranının yüksek olduğunu belirtmişlerdir. Yapılan çalışmada 39 aileden oluşan aile yüz veri seti üzerinde, yüz benzerliğini ölçmek için yüz nirengi noktaları tespit edilip bu nirengi noktaları arasındaki Öklid uzaklık mesafesi dikkate alınarak benzerlik oranı çıkartılmıştır. Birinci derece akrabaların yüz benzerliğinin yüksek olduğu görülmektedir.

Cao ve ark. (2013), Labeled Faces in the Wild (LFW) veri seti üzerinde aynı bireye ait farklı görüntülerdeki yüzlerini eşleştirmeye çalışmışlardır. Kendi geliştirdikleri algoritma ile %89,73 başarı elde etmişlerdir. Ayrıca Dehshibi ve ark. (2012), 4 aileden oluşan veri setleri üzerinde Yerel Gabor İkili Desen Histogram Dizisi yöntemi kullanarak aile içi yüz benzerliklerini incelemişlerdir. Testler sonucunda aile içi yüz benzerliğinin %40,54 olduğunu belirtmişlerdir. Bu tezde yapılan çalışmada 39 aileden oluşan aile veri seti üzerinde ESA yöntemi ve Öklid uzaklığı kullanılarak çekirdek ailedeki bireyler arasında benzerliği bulmak için yapılan testler sonucunda en yüksek benzerlik oranı %68,42 ile yüz hattı olmuştur (bkz. Çizelge 4.5).

Zhang ve ark. (2011), Han ve ark.(2013), Klum ve ark. (2013) ve Liu ve ark. (2018) farklı veri setleri üzerinde bireylerin görüntüleri ile robot resimlerini eşleştirmek için birçok sistem geliştirmişlerdir. Bu sistemlerde kullanılan algoritmaların kendilerine özgü ama veri setlerinin benzer olabileceğini belirtmişlerdir. Bireylerin robot resimlerini oluşturmak için Holistik yöntemlerin (bilgisayar yazılımları) daha başarılı olduğunu ve veri setleri

oluşturulurken genelde bu yöntemin kullanılması gerektiğini belirtmişlerdir. Literatürde belirtilen çalışmaların test sonuçları dikkate alındığında en yüksek başarı oranının %98,70 olduğu görülmektedir (Zhang ve ark.(2011)).

Bu tez kapsamında yapılan çalışmada;

- Çalışmaya özgü bir aile yüz veri seti oluşturulmuştur. Aile yüz veri seti; 39 çekirdek ailedeki bireylerin, çoğunluğu biyometrik resim olan, yüz görüntülerinden oluşmaktadır. Bu veri setinde her aileden sayıları 4-9 arasında değişen bireylerin yüz görüntüleri bulunmaktadır. Yüz görüntülerinin toplam sayısı 208'dir. Bu veri seti ilk kez bu tez çalışmasına özgü oluşturulmuştur ve herhangi bir başka çalışmada kullanılmamıştır.
- Makine öğrenme tekniklerinden olan ESA, DSA ve YGH-DVM yöntemleri python programlama dili ve ilgili kütüphaneleri kullanılarak uygulamalar geliştirilmiştir.
- Visual Studio .Net platformu kullanılarak hazırlanan uygulama ile aile bireylerinin yüz nirengi noktaları işlenerek aile yüz referans modeli oluşturulmuştur.
- Bir önceki adımdan elde edilen aile yüz referans modeli baz alınarak web tabanlı yüz giydirmeye (robot resim oluşturma) uygulaması geliştirilmiştir.
- Yapılan analizler sonucunda (bkz. Çizelge 4.6) geliştirilen sistemin robot resim ile gerçek resmi eşleştirmede %93,90 oranında bir başarıya sahip olduğu gözlemlenmiştir.

Çalışmanın dezavantajlarından biri ailelerden toplanan görüntülerin kalitesidir. Toplanan görüntülerin biyometrik resim olması tercih edilmektedir. Fakat bazı aile bireylerinin biyometrik resimlere sahip olmamasından kaynaklı test değerlerinde düşme görülmektedir.

İleriye yönelik çalışmalarda;

- Çizelge 3.3 ve Çizelge 3.4'te gösterilen yüz organ kütüphanesine boyun, alın kırışıklığı, gözaltı kırışıklığı ve renk bileşenleri gibi yeni bölümler eklenerek kütüphane genişletilebilir. Bu yeni bölümlerin eklenmesi kalıtsal bir benzerlik yaratmasa da robot resim oluştururken benzerlik yüzdesini artırabilir.
- Bir ressam tarafından 12 bölümden oluşan kütüphaneye her bölüm için yeni yüz organları (göz, burun, kaş, dudak, kulak ve kafa modeli), yüz nitelikleri (saç, sakal ve bıyık) ve yüz organ aksesuarları (gözlük, hırma ve piercing) çizilerek kütüphanedeki organların çeşitliliği artırılabilir. Bu sayede robot resim oluştururken benzerlik yüzdesi artırılabilir.

- Yukarıda bahsedilen ve çalışmada dezavantaj olarak görülen görüntü kalite değeri (uygun keskinlik, doğru renk seçimi ve gürültüsüz bir görüntü) ailelerin tamamından biyometrik görüntü toplanarak arttırılabilir. Ayrıca aile bireylerinden farklı açılardan çekilmiş biyometrik görüntüler toplanarak ya da video görüntüleri sisteme dâhil edilip başarı yüzdesi arttırılabilir.
- Geliştirilen sistemde yüz tanıma için kullanılan ESA, DSA ve YGH-DVM algoritmaları geliştirilerek veya farklı algoritmalar kullanılarak başarı oranları arttırılabilir.
- Uygulamada kullanılan bilgisayar özellikleri (ekran kartı, ram ve işlemci) arttırılarak zamansal anlamda başarı oranı arttırılabilir.



## KAYNAKLAR

- Akay, S., 2018. *Facial Action Unit Detection in Videos Using Deep Neural Networks* (Yüksek Lisans Tezi). BÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Akpınar, B., 2019. *Görüntü Sınıflandırma İçin Derin Öğrenme İle Bayesçi Derin Öğrenme Yöntemlerinin Karşılaştırılması* (Yüksek Lisans Tezi). AKÜ, Fen Bilimleri Enstitüsü. Afyon.
- Alhamrouni, M., 2017. *Iris Recognition By Using Image Processing Techniques* (Yüksek Lisans Tezi). AÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Alpar, R., 2011. *Çok Değişkenli İstatistiksel Yöntemler*. Detay Yayıncılık. İstanbul, Türkiye.
- Alpkoçak, A. ve T. Berber, 2009. Tıbbi Görüntü Arşivlerinden İçerik Tabanlı Görüntü Erişim Yöntemleri, *Sağlık ve Hastane Yönetimi 2. Kongresi Bildiriler Kitabı*, Ankara.
- Amidi, A., 2020. Evrişisel ağ katmanları. <https://stanford.edu/~shervine/l/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks>. Erişim tarihi: 14.04.2020.
- Amon, T., 2017. *Diagnosis Of The Pseudomonas Aeruginosa Disease On Human And Plants With Image Processing And Pattern Recognition Techniques* (Yüksek Lisans Tezi). EÜ, Fen Bilimleri Enstitüsü. İzmir.
- Anonim, 2020a. <http://www.prowmes.com/blog/makine-ogrenmesi/>. Erişim tarihi: 09.04.2020
- Anonim, 2020b. <http://www.aksaray.gov.tr/insan-haklari-nedir>. Erişim Tarihi: 14.07.2020.
- Anonim, 2020c. <https://www.unicef.org/turkey/basin-bultenleri/tum-dunyada-belirlenmis-insan-ticareti-magdurlarının-nerede-yse-ucte-birini>. Erişim tarihi: 14.04.2020.
- Anonim, 2020d. <https://biruni.tuik.gov.tr/medas/?kn=204&locale=tr>. Erişim tarihi: 02.03.2020.
- Anonim, 2020e. <https://kodedu.com/2014/05/naive-bayes-siniflandirma-algoritmasi/>. Erişim tarihi: 09.04.2020.
- Anonim, 2020f. [https://acikders.ankara.edu.tr/pluginfile.php/792/mod\\_resource/content/2/Merkezi\\_Eglim\\_ve\\_Dagilim\\_Olculeri.pdf](https://acikders.ankara.edu.tr/pluginfile.php/792/mod_resource/content/2/Merkezi_Eglim_ve_Dagilim_Olculeri.pdf). Erişim tarihi: 14.05.2020.
- Anonim, 2020g. Deep Neural Network. <https://www.techopedia.com/definition/32902/deep-neuralnetwork>. Erişim tarihi: 22.10.2020
- Ateş, N. ve Küçüksille, E. U., 2013. Destek vektör makineleri ile yaramaz elektronik postaların filtrelenmesi, *Türkiye Bilişim Vakfı Bilgisayar Bilimleri ve Mühendisliği Dergisi*, 6: 81-87.
- Ayata, F., Çavuş, H., İnan, M., Seyyarer, E., Biçek, E., ve Kına, E., 2020. Dostroajan: Facial recognition based system input control agent. *AJIT-e: Online Academic Journal of Information Technology*, 11(40): 82–96.
- Ayvaz, U., ve Gürüler, H., 2017. Bilgisayar kullanıcılarına yönelik duygusal ifade tespiti. *Bilişim Teknolojileri Dergisi*, 231–231.
- Bayrakdar, S., Akgün, D., ve Yücedağ, İ., 2016. Yüz ifadelerinin otomatik analizi üzerine bir literatür çalışması A survey on automatic analysis of facial expressions. *SAÜ Fen Bilimleri Dergisi*, 20(2): 383–398.
- Burl, M. C., Leung, T. K., ve Perona, P., 1995. Face localization via shape statistics. *Workshop on Automatic Face and Gesture Recognition*.
- Cao, X., Wei, Y., ve Wen, F., 2012. Face alignment by explicit shape regression. *2012 IEEE Conference on Computer Vision and Pattern Recognition*.
- Cao, Q., Ying, Y. ve Li, P., 2013. Similarity Metric Learning for Face Recognition, *2013 IEEE International Conference on Computer Vision*, Sydney, NSW, pp. 2408-2415, doi: 10.1109/ICCV.2013.299.
- Çeliktutan, O., Akakın, Ç. H., ve Sankur, B., 2013. İnsan yüzlerinde 2b nirengi noktalarının

- otomatik saptanması. *IEEE 16th Signal Processing, Communication and Applications Conference*.
- Cengil, E., ve Çınar, A., 2016. A new approach for image classification: Convolutional neural network. *European Journal of Technic EJT*, 6(2): 96–103.
- Çetinkaya, H. H., 2012. *Ten Rengi Ve Yüz Bulma Teknolojisine Dayalı İnsan Sayma Sistemi* (Yüksek Lisans Tezi). BŞEÜ, Fen Bilimleri Enstitüsü. Bilecik.
- Cootes, T. F., Edwards, G. J., ve Taylor, C. J., 1998. Active appearance models. *Eropean Conferance on Computer Vision*, 2: 484–498.
- Coşkun, M., Uçar, A., Yıldırım, Ö., ve Demir, Y., 2017. Face recognition based on convolutional neural network. *2017 International Conference on Modern Electrical and Energy Systems (MEES)*.
- Dhall, A., Asthana A., ve Goecke, R., 2011. A SSIM-based approach for finding similar facial expressions. *2011 IEEE International Conference on Automatic Face ve Gesture Recognition (FG)*, Santa Barbara, CA. 815-820.
- Dandekar, A. R. ve Nimbarte, M. S., 2014. Verification of family relation from parents and child facial images, *2014 International Conference on Power, Automation and Communication (INPAC)*, Amravati, 157-162.
- Dehshibi, M. M., Shanbezadeh, J. ve Alavi, M., 2012. Facial family similarity recognition using local gabor binary pattern histogram sequence, *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, 219-224.
- Demir, A., Akılotu, B., Kadiroğlu, Z., ve Şengür, A., 2019. Makine öğrenmesi yöntemleri kullanılarak bitcoin tahmini. *2019 1st International Informatics and Software Engineering Conference*, 1–4.
- Doğan, G., 2010. *Yapay Sinir Ağları Kullanarak Türkiye’deki Özel Bir sigorta şirketinde portföy Değerlendirmesi* (Yüksek Lisans Tezi). HÜ, Fen Bilimleri Enstitüsü. Ankara.
- Erdoğan, A. Y., 2010. *Yüz Tanımada Özyüz Ve Fisher Yüz Algoritmalarının İncelenmesi*. (Yüksek Lisans Tezi). AÜ. Fen Bilimleri Enstitüsü. Ankara.
- Erge, B., ve Baykara, M., 2010. Content based medical image retrieval feature extraction of using statistical spatial methods for content based medical image retrieval. *2010 IEEE 18th Signal Processing and Communications Applications Conference*, Diyarbakir, 692-695.
- Fan, H., ve Zhou, E., 2016. Approaching human level facial landmark localization by deep learning, 47: 27–35.
- Feng, Z., Kittler, J., Member, L., Christmas, W., ve Wu, X., 2017. A unified tensor-based active appearance face model. *ACM Transactions on Multimedia Computing Communications and Applications*, 1–12.
- Ghahramani, M., Yau, W. ve Teoh, E.K., 2014. Family verification based on similarity of individual family member’s facial segments. *Machine Vision and Applications* 25: 919–930.
- Godbole, R., ve Burad, S., 2019. Face expression detection using CNN. *International Journal Of Innovative Research in Technolgy*, 5(12), 16–18.
- Goodfellow, I., Bengio, Y. ve Courville, A., 2016. Deep Learning, MIT Press. <https://www.deeplearningbook.org/contents/convnets.html>. Erişim Tarihi: 14.10.2020.
- Gündüz, S., 2011. *Uzaklık Fonksiyonlarının Çok Boyutlu Ölçekleme Algoritmalarındaki Etkinliğinin İncelenmesi Ve Uygulamalar* (Yüksek Lisans Tezi). ÇÜ, Fen Bilimleri Enstitüsü. Adana.
- Han, H., Klare, B., Bonnen, K. ve Jain, A., 2013. Matching composite sketches to face photos: a component based approach. *IEEE Transactions on Information Forensics and Security*, 8, 1: 191-204.
- Hanılçı, A., 2019. *Evrişimsel Sinir Ağları Kullanılarak Ekg Ve Yüz Tabanlı Biyometrik*

- Tanuma** (Yüksek Lisans Tezi). BTÜ, Fen Bilimleri Enstitüsü. Bursa.
- Hjelmås, E., ve Low, B. K., 2001. Face detection: A survey. Computer vision and image understanding, *Computer Vision And Image Understanding*, **83**(3): 236–274.
- Holat, R., 2016. **Yüz Bulma Ve Tanıma Sistemleri Kullanarak Kimlik Tespitinin Yapılması** (Yüksek Lisans Tezi). DÜ, Fen Bilimleri Enstitüsü. Düzce.
- İnik, Ö., ve Ülker, E., 2017. Derin öğrenme ve görüntü analizinde kullanılan derin öğrenme modelleri. *Gaziosmanpaşa Bilimsel Araştırma Dergisi*, **6**: 85–104.
- Iranmanesh, S. M., Kazemi, H., Soleymani, S., Dabouei, A. ve Nasrabadi, N.M., 2018. Deep sketch-photo face recognition assisted by facial attributes. *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, Redondo Beach, CA, USA, 2018, 1-10, doi: 10.1109/BTAS.2018.8698564.
- Jahangiri, N., 2015. **Veri Madenciliği Teknikleriyle İçerik Tabanlı Görüntü Erişimi** (Yüksek Lisans Tezi). GÜ, Fen Bilimleri Enstitüsü. Ankara.
- Jain, A. K., Klare, B. ve Park, U., 2012. Face Matching and retrieval in forensics applications, *IEEE MultiMedia*, **19**, **1**: 20-20, doi: 10.1109/MMUL.2012.4.
- Johnson, R. A., ve Wichern, D. W., 1988. *Applied Multivariate Statistical Analysis*. Prentice-Hall International Editions.
- Kadiroğlu, Z., 2019. **Histopatolojik Meme Kanseri Görüntülerinin Evrişimsel Sinir Ağları Kullanarak Sınıflandırılması** (Yüksek Lisans Tezi). F.Ü., Fen Bilimleri Üniversitesi, 48s, Elazığ.
- Kaplan, A., 2018. **Gerçek Ve Yarı Gerçek Zamanlı Yüz Tespit Etme** (Yüksek Lisans Tezi). FÜ, Fen Bilimleri Enstitüsü. Elazığ.
- Karaali, A., 2012. **Face Detection And Facial Expression Recognition Using Moment Face Detection And Facial Expression Recognition**. (Yüksek Lisans Tezi) Fen Bilimleri Enstitüsü. İstanbul.
- Karakoç, M., 2012. **Görüntü İşleme, Teknolojiler ve Uygulamaları** (Yüksek Lisans Tezi). EÜ, Fen Bilimleri Enstitüsü. İzmir.
- Karhan, Z., 2014. **Radyolojik Görüntüler İçin İçerik Tabanlı Görüntü Erişim Sistemi** (Yüksek Lisans Tezi). FÜ, Fen Bilimleri Enstitüsü. Elazığ.
- Kassem, A., 2012. **Segmentation Of Skin Cancer By Using Image Processing Techniques** (Yüksek Lisans Tezi). YTÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Kaş, Ü., Tanyıldızı, E., 2017. Euler Renk ve Hareket Büyütme Yöntemlerinin Performans Analizi. *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimleri Dergisi* **17**: 506-515. Afyon.
- Kaya, Y., ve Kayci, L., 2014. Kelebek görüntülerin sınıflandırılması için bir içerik bazlı görüntü erişim sistemi. *XVI. Akademik Bilişim Konferansı*. Mersin.
- Khokher, A., Talwar, R., 2011. Image Retrieval: A State of the art approach for CBIR. *International Journal of Engineering Science and Technology (IJEST)*, **3**(8):45-49.
- Khorram, T., 2019. **Network Anomaly Detection Using Optimized Machine Learning Algorithms** (Yüksek Lisans Tezi). SÜ, Fen Bilimleri Enstitüsü. Konya.
- Klum, S., Han, H., Jain, A. ve Klare, B., 2013. Sketch based face recognition: forensic vs. composite sketches. *International Conference on Biometrics (ICB)*. 1-8.
- Kılıç, Ö., 2019. **Arşivlerde yapay zekâ uygulamalar : GEODI - Atılım Üniversitesi Kadriye Zaim Kütüphanesi Ankara Dijital Kent Arşivi Örneği**. Bilgi Yönetimi Ve Bilgi Güvenliği Sempozyumu.
- Kılıçaslan, M , Tanyeri, U , Demirci, R., 2020. Tekrarlı ortalama yardımıyla renk indirgeme ve görüntü erişimi . *Düzce Üniversitesi Bilim ve Teknoloji Dergisi* , **8** (1): 1042-1057.
- Kızrak, A., 2020. Evrişim katmanı. <https://medium.com/@ayyucekizrak/derine-daha-derine-evrisimli-sinir-aglari-2813a2c8b2a9>. Erişim tarihi: 14.04.2020.
- Koçyiğit, G., 2016. **Dictionary Ensemble Based Active Learning For Multiple Instance**

- Image Classification** (Yüksek Lisans Tezi). İTÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Kurbanov, O., 2018. **Derin Sinir Ağları Kullanarak Parmak İzi Tanımda Yeni Yaklaşımlar** (Yüksek Lisans Tezi). EÜ, Fen Bilimleri Enstitüsü. Kayseri.
- Kurt, Z., 2019. **Temel Bileşen Analiziyle Öznelik Seçimi ve Görsel Nesne Sınıflandırma** (Yüksek Lisans Tezi). EOÜ, Fen Bilimleri Enstitüsü. Eskişehir.
- Kute, R.S., Vyas, V. ve Anuse, A., 2020. Association of face and facial components based on cnn and transfer subspace learning for forensics applications. *SN COMPUT. SCI.* **1**: 287. <https://doi.org/10.1007/s42979-020-00280-2>.
- Lai, C. Q. ve Teoh, S. S., 2016. An Efficient method of hog feature extraction using selective histogram bin and pca feature reduction. *Advances in Electrical and Computer Engineering*. **16**: 101-108. 10.4316/AECE.2016.04016.
- Leung, T. K., Burl, M. C., ve Perona, P., 1995. Finding faces in cluttered scenes using random labeled graph matching. *IEEE International Conference on Computer Vision*, 637–644. <https://doi.org/10.1109/iccv.1995.466878>.
- Lewis, C.D., 1982. **Industrial and Business Forecasting Methods**. Londra: Butterworths Publishing.
- Liu, K., ve Wang, W., 2019. Real-Time facial expression recognition based on CNN. *2019 International Conference on System Science and Engineering (ICSSE)*, 120–123.
- Liu, D., Li, J., Wang, N., Peng, C., ve Gao, X., 2018. Composite Components-Based Face Sketch Recognition. *Neurocomputing*. **302**. 10.1016/j.neucom.2018.03.042.
- Liu, S., Song, Z., Liu, G., Xu, C., Lu, H., ve Yan, S., 2012. *Cross-Scenario Clothing retrieval via parts alignment and auxiliary set*. In *Computer Vision and Pattern Recognition (CVPR)*, 2012 IEEE Conference on, 3330–3337. IEEE, 2012.
- Mallick, S., 2016. Anonim 2020f. <https://www.learnopencv.com/histogram-of-oriented-gradients/>. Erişim tarihi: 29.04.2020.
- Martins, P., 2008. **Active Appearance Models for Facial Expression Recognition and Monocular Head Pose Estimation** (Doktora Tezi). Department of Electrical and Computer Engineering, University of Coimbra.
- Mittal, P., Jain, A., Goswami, G., Singh, R. ve Vatsa, M., 2014. Recognizing composite sketches with digital face images via ssd dictionary. *IJCB 2014 - 2014 IEEE/IAPR International Joint Conference on Biometrics*. 10.1109/BTAS.2014.6996265.
- Milborrow S., Nicolls F., 2008. Locating Facial Features with an Extended Active Shape Model. In: *Forsyth D., Torr P., Zisserman A. (eds) Computer Vision – ECCV. ECCV 2008. Lecture Notes in Computer Science, 5305. Springer, Berlin, Heidelberg*.
- Mohamed, S., Weng, Y., Jiang, J. ve Ipson, S., 2008. An Efficient face image retrieval through DCT features. *Proceedings of the 10th IASTED International Conference August 18-20, Kailua-Kona, HI, USA*.
- Mohri, M., ve Rostamizadeh, A., 2012. **Foundations of Machine Learning**. Cambridge. MIT Press.
- Nabiyev, V. ve Bahçekapili, S., 2007. Diabetik retinopatinin otomatik algılanması amacıyla göz görüntüsünden kan damarlarının eşiklenmesi. *Elektrik, Elektronik, Bilgisayar ve Biyomedikal Mühendisliği 12. Ulusal kongresi*. Eskişehir.
- Nawaf, A. M., 2019. **Development Of A New Software For Fabric Defect Detection and Classification Using Image Processing and Machine Learning Methods** (Yüksek Lisans Tezi). Ç.Ü. Fen Bilimleri Enstitüsü. Adana.
- Onat, M., 2008, **Dijital Görüntü İşleme Yöntemi ile Liçli Beton Numunelerindeki Çatlakların Tesbit Edilmesi** (Yüksek Lisans Tezi), FÜ, Fen Bilimleri Enstitüsü, Elazığ.
- Pişkin, M., 2020. Yüz Tanıma | Mesut Pişkin. <http://mesutpiskin.com/blog/yuz-tanima.html>. Erişim tarihi: 03.03.2020.

- Rashid, M. K. (2019). *Feature Detection And Classification Of Pistachio By Using Image Processing* (Yüksek Lisans Tezi). GÜ, Fen Bilimleri Enstitüsü. Gaziantep.
- Razikou Dilwani, A. A., 2019. *Makine Öğrenmesi İle Acil Triyaj Ve Hastane Yatış Tahmini* (Yüksek Lisans Tezi). AÜ, Fen Bilimleri Enstitüsü. Ankara.
- Salman, M., 2018. *Hiperspektral ve Lidar Verilerinin Öz nitelik Ve Karar Seviyelerinde Tümüleştirilmesi ve Derin Evrişimli Sinir Ağlarıyla Sınıflandırılması* (Yüksek Lisans Tezi). HÜ, Fen Bilimleri Enstitüsü, Ankara.
- Sakarya, M., 2019. *Seçilen Siğil Tedavi Yönteminin Farklı Makine Öğrenmesi Algoritmaları İle Başarımının Tahmin Edilmesi* (Yüksek Lisans Tezi). SÜ, Fen Bilimleri Enstitüsü. Sakarya.
- Ser, G. ve Bati, C. T., 2019. Derin sinir ağları ile en iyi modelin belirlenmesi: mantar verileri üzerine keras uygulaması. *Yüzüncü Yıl Üniversitesi Tarım Bilimleri Dergisi*. 406-417. 10.29133/yyutbd.505086.
- Sewak, M., Karim, R., ve Pujari, P., 2018. *Practical Convolutional Neural Networks: Implement Advanced Deep Learning Models Using Python*. Packt Publishing Ltd.
- Seyyarer, B., ve Aydın, T., 2017. Değişmez momentler kullanarak içerik tabanlı görüntü erişim sistemi ve imge sınıflandırma yöntemlerinin karşılaştırılması. *Anattolliian Journall off Computer Sciencies*, 2(1): 1–9.
- Sıddık, O., 2019. *Automatic Spirulina Detection Using Image Processing Techniques* (Doktora Tezi). AÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Singh, A., 2019. HOG Özellik Tanımlayıcısına Değerli Bir Giriş. <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/>. Erişim Tarihi: 14.10.2020.
- Shbib, R. ve Zhou, S., 2015. Facial Expression Analysis using Active Shape Model. *International Journal of Signal Processing, Image Processing and Pattern Recognition*. 8: 9-22.
- Soydaş, M., 2017. Anonim-8. <http://mehmetsoydas.com/2017/05/27/real-time-face-detection-recognition-YGH-features-svm/>. Erişim tarihi: 29.04.2020.
- Sung, J. ve Kim, D., 2007. A Background Robust Active Appearance Model Using Active Contour Technique. *Pattern Recognition*. 40. 108-120. 10.1016/j.patcog.2006.06.017.
- Süer, K., 2019. *Yapay Zeka İle Meme Kanseri Lenf Nodu Analizi* (Yüksek Lisans Tezi). BÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Şahinarslan, F. V., 2019. *Makine Öğrenmesi Algoritmaları İle Nüfus Tahmini: Türkiye Örneği* (Yüksek Lisans Tezi). İTÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Şeker, S. E., 2010. Anonim-9. <http://bilgisayarkavramlari.sadievrenseker.com /2008/12/01/svm-support-vector-machine-destekci-vektor-makinesi/>. Erişim tarihi: 29.04.2020.
- Şenel, G., 2018. *Delineation Of Water Bodies With Landsat 8 And Sentinel 2 Satellite Imagery Using Different Image Processing Algorithms* (Yüksek Lisans Tezi). Fen Bilimleri Enstitüsü. İstanbul.
- Talandova, H., Kralik, L. ve Adamek, M., 2016. Determination of The Uncertainties and the Physiological Similarities of Family Members by Using the Biometric Device the Broadway 3D. *International Journal of Applied Engineering Research*. 11: 6373-6375.
- Tatlıdil, H., 1996. *Uygulamalı Çok Değişkenli Analiz*. Akademi Mat-1996. Ankara.
- Thaware, R., 2018. Anonim-7. <https://www.eeweb.com/profile/rajeevthaware/articles/real-time-face-detection-and-recognition-with-svm-and-YGH-features>. Erişim tarihi: 29.04.2020.
- Thomaz, C. E., 2012. FEI yüz veri seti. <https://fei.edu.br/~cet/facedatabase.html>. Erişim tarihi: 14.04.2020.
- Tiryaki, B., 2019. *Trafik İşaretlerinin Derin Sinir Ağları İle Sınıflandırılması* (Yüksek

- Lisans Tezi). AÜ, Fen Bilimleri Enstitüsü. Erzurum.
- Turhan, C., Gökçen, G., ve Kazanasmaz, T. 2013. *Yapay sinir ağları ile İzmir'deki çok katlı binaların toplam enerji tüketimlerinin tahmin edilmesi*. 11. Ulusal Tesisat Mühendisliği Kongresi. İzmir.
- Tükel, T., 2019. *Görüntü İşleme Ve Evrimsel Sinir Ağları Kullanılarak Diyabetik Retinopati Hastalığının Tespiti* (Yüksek Lisans Tezi). SDÜ, Fen Bilimleri Enstitüsü. Isparta.
- Türker, S., 2019. *Zararlı Android Yazılımlarının Makine Öğrenmesi İle Ailelerine Göre Sınıflandırılması* (Yüksek Lisans Tezi). HÜ, Fen Bilimleri Enstitüsü. Ankara.
- Ulgen, K., 2017. Anonim-10. <https://medium.com/@k.ulgen90/makine-ogrenimi-bolum-4-destek-vektor-makinelere-2f8010824054>. Erişim tarihi: 29.04.2020.
- Vandeput, N., 2019. Tahmin KPI: RMSE, MAE, MAPE. <https://medium.com/analytics-vidhya/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d>. Erişim tarihi: 02.06.2020.
- Vandeput, N., 2020. <https://towardsdatascience.com/forecast-kpi-rmse-mae-mape-bias-cdc5703d242d>. Erişim Tarihi: 28.10.2020.
- Varol, A., ve Cebe, B., 2011. Algorithms of Face Recognition. *5th International Computer ve Instructional Technologies Symposium*, September, 22–24.
- Vinay, A., Reddy, D. N., Sharma, A. C., Daksha, S., Bhargav, N. S., Kiran, M. K., Murthy, K. N. B., ve Natrajan, S., 2017. G-CNN and F-CNN: Two CNN Based Architectures For Face Recognition. *Proceedings of the 2017 International Conference On Big Data Analytics and Computational Intelligence, ICBDAI 2017*, 6, 23–28. <https://doi.org/10.1109/ICBDACI.2017.8070803>.
- Wan, Q. ve Panetta, K., 2016. A Facial Recognition System For Matching Computerized Composite Sketches to Facial Photos Using Human Visual System Algorithms. *2016 IEEE Symposium on Technologies for Homeland Security (HST)*, Waltham, MA, 2016, 1-6, doi: 10.1109/THS.2016.7568945.
- Wang, N., Gao, X., Tao, D., ve Liu, W., 2014. Facial Feature Point Detection: A Comprehensive Survey. *Neurocomputing*. 275. 10.1016/j.neucom.2017.05.013.
- Whitmarsh, T., Velkamp, R., Spagnuolo, M., Marini, S., ve Haar, F., 2008. Landmark Detection on 3D Face Scans by Facial Model Registration. *1st international symposium on shapes and semantics*. 71–5.
- Yang, G., ve Huang, T., 1994. Human Face Detection In A Complex Background. *Pattern Recognition*, 27(1): 53–63. [https://doi.org/10.1016/0031-3203\(94\)90017-5](https://doi.org/10.1016/0031-3203(94)90017-5).
- Yang, M. H., Kriegman, D. J., ve Ahuja, N., 2002. Detecting Faces In Images: A Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1): 34–58. <https://doi.org/10.1109/34.982883>.
- Yılmaz, M., 2016. *Görüntü İşleme Teknikleri ile Zirai Tuzaklardaki Böcek Adedi Tespiti* (Yüksek Lisans Tezi). MÜ, Fen Bilimleri Enstitüsü. İstanbul.
- Zhang, C., ve Zhang, Z., 2010. A Survey of Recent Advances in Face Detection. *Learning*, June, 1,17. <https://doi.org/10.1.1.167.5270>.
- Zhou, A., Chen, J., Ding, J., & Pan, Z., 2020. Face Recognition Based on Two-stage CNN Combined with Transfer Learning. **2019 IEEE 4th International Conference on Image, Vision and Computing (ICIVC)**, Xiamen, China, pp. 401-406, doi: 10.1109/ICIVC47709.2019.8980761.
- Zhou, E., 2014. Extensive Facial Landmark Localization with Coarse-to-fine Convolutional Network Cascade. **2013 IEEE International Conference on Computer Vision Workshops**. <https://doi.org/10.1109/ICCVW.2013.58>
- Zhang, Zhanpeng, Luo, P., Loy, C. C., & Tang, X., 2014. Facial Landmark Detection by Deep Multi-task Learning. *European Conference on Computer Vision*.

**EK 1.**

Evrak Tarih ve Sayısı: 15/05/2020-32729



T.C.  
VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ REKTÖRLÜĞÜ  
Sosyal ve Beşeri Bilimleri Etik Kurulu



Sayı : 85157263-604.01.02-E.32729  
Konu : 21. Doktora Öğrenci Faruk  
AYATA'ya ait tez çalışmasında  
alınan kurul kararı

15/05/2020

Sayın Doç. Dr. Hayati ÇAVUŞ

Sosyal ve Beşeri Bilimleri Yayın Etik Kurulu'nun 13.05.2020 tarih ve 2020/03-21 sayılı kararı gereği; Danışmanlığını yapmış olduğunuz doktora öğrencisi Faruk AYATA'nın yaptığı, "İçerik Tabanlı Görüntü Erişim Sistemiyle Yüz Tahmini" adlı tez çalışmasıyla ilgili kurulumuzun aldığı karar ekte sunulmuştur.

Gereğini bilgilerinize rica ederim.

e-imzalıdır  
Prof. Dr. Orhan DENİZ  
Etik Kurulu Başkanı

Ek: 21. Doktora Öğrenci Faruk AYATA'ya ait tez çalışmasında alınan kurul kararı


Adres: Van Yüzüncü Yıl Üniversitesi Rektörlüğü Sosyal ve Beşeri Bilimleri Etik  
Kurulu Zeve Kampüsü 65080 Tuşba/VAN  
Telefon: +90 432 2251701-04 / +90 4445065 Faks: +90 432 4865413  
e-Posta: rektorluk@yyu.edu.tr Elektronik Ağ: http://www.yyu.edu.tr

Ayrıntılı bilgi için imzalat: Mehmet Şah OĞUZ  
Unvanı: Şef

Bu belge 5070 sayılı Elektronik İmza Kanununun 5. Maddesi gereğince güvenli elektronik imza ile imzalanmıştır.



Evrak Tarih ve Sayısı: 14/05/2020-32528

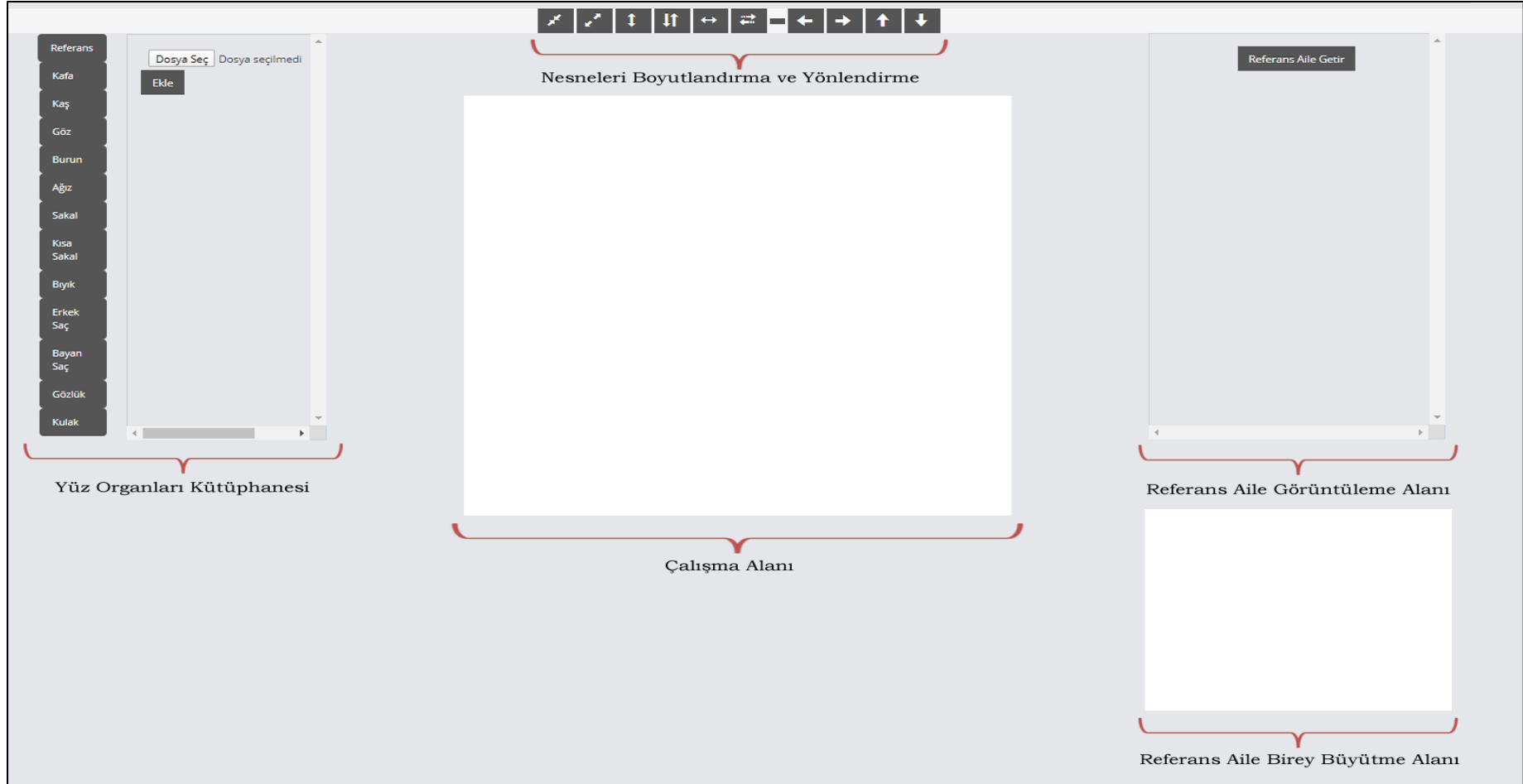
	<p style="text-align: center;"><b>T.C.</b>  <b>VAN YÜZÜNCÜ YIL ÜNİVERSİTESİ</b>  <b>SOSYAL VE BEŞERİ BİLİMLERİ YAYIN</b>  <b>ETİK KURUL BAŞKANLIĞI</b>  <b>ETİK KURUL KARARLARI</b></p>
<b>TOPLANTI TARİHİ: 13.05.2020</b> <b>OTURUM SAYISI: 2020/03</b> <b>TOPLANTIDA ALINAN KARAR SAYISI: 28</b>	<p style="text-align: right;"><b>Sayfa: 21/28</b></p>

Van Yüzüncü Yıl Üniversitesi Sosyal ve Beşeri Bilimleri Yayın Etik Kurulu'nun 13/05/2020 tarihinde saat 10.00' da Van Yüzüncü Yıl Üniversitesi Rektörlük toplantı salonunda Prof. Dr. Orhan DENİZ başkanlığında yapmış olduğu toplantıda aşağıdaki karar/kararları almıştır:

**KARAR NO 2020/03-21.** Danışmanlığını, Eğitim Fakültesi, Bilgisayar ve Öğretim Teknolojileri Eğitimi Anabilim Dalı öğretim üyesi Doç. Dr. Hayati ÇAVUŞ'un yaptığı, doktora öğrencisi Faruk AYATA'nın yaptığı, "İçerik Tabanlı Görüntü Erişim Sistemiyle Yüz Tahmini" adlı tez çalışmasında kullanılacak olan araçlar incelenmiş olup, söz konusu araçların ilgili kişilere uygulanmasında Sosyal ve Beşeri Etik Kuralları ve İlkeleri çerçevesinde herhangi bir sakınca olmadığına karar verilmiştir.

	<b>BAŞKAN</b> Prof. Dr. Orhan DENİZ Edebiyat Fakültesi	
<b>ÜYE</b> Prof. Dr. Mehmet Şirin ÇIKAR İlahiyat Fakültesi	<b>ÜYE</b> Prof. Dr. Hayati AYDIN İlahiyat Fakültesi (Katılmadı)	<b>ÜYE</b> Prof. Dr. Reha SAYDAN İktisadi ve İd. Bil. Fakültesi
<b>ÜYE</b> Prof. Dr. Metin AYIŞIĞI Edebiyat Fakültesi	<b>ÜYE</b> Prof. Dr. Hasan ÇİÇEK Eğitim Fakültesi	<b>ÜYE</b> Prof. Dr. Zihni MEREY Eğitim Fakültesi

## EK 2. Web tabanlı robot resim oluşturma uygulamasına ait görüntüsü ve java script kodları



**Java Script Kodları**

```

<html lang="en">
    <head>
        <meta charset="utf-8" />
        <title>TEZ</title>
<style type="text/css" media="print">
@media print {
    body {-webkit-print-color-adjust: exact;}
}
@page {
    size:A4 portrait;
    margin-left: 0px;
    margin-right: 0px;
    margin-top: 0px;
    margin-bottom: 0px;
    margin: 0;
    -webkit-print-color-adjust: exact;
}
</style><style type="text/css" media="print">
@media print {
    body {-webkit-print-color-adjust: exact;}
}
@page {
    size:A4 portrait;
    margin-left: 0px;
    margin-right: 0px;
    margin-top: 0px;
    margin-bottom: 0px;
    margin: 0;
    -webkit-print-color-adjust: exact;
}
.img-hover-zoom {
    height: 300px; /* Modify this according to your need */
    overflow: hidden; /* Removing this will break the effects */
}
/* Point-zoom Container */
.img-hover-zoom--point-zoom img {
    transform-origin: 65% 75%;
    transition: transform 1s, filter .5s ease-out;
}
/* The Transformation */
.imageX{
    width:600px;
    height:auto;
}
.imageX:hover{
    transform:scale(1.2);
}
</style>
    </head>
    <body style="overflow: hidden;">
        <div class="row" style="margin-right: 0px; margin-left: 0px;">

```

```

    <div class="clearfix form-actions" style="padding: inherit; margin-bottom: 1px; margin-top:
6px; width:100%">
        <div class="col-md-12" style="text-align:center" >
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;"
id="small">

                <span class="glyphicon glyphicon-resize-small" ></span>
            </button>

            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;">
                <span class="glyphicon glyphicon-resize-full" id="full"></span>
            </button>

            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;">
                <span class="glyphicon glyphicon-resize-vertical"
id="vertical"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;">
                <span class="glyphicon glyphicon-glyphicon-sort"
id="vertical_small"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;">
                <span class="glyphicon glyphicon-resize-horizontal"
id="horizontal"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;">
                <span class="glyphicon glyphicon-glyphicon-transfer"
id="horizontal_small"></span>
            </button>
            <button class="btn btn-sm btn-inverse" style="padding: 0px 5px;">
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;"
id="left">

                <span class="glyphicon glyphicon-arrow-left"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;"
id="right">

                <span class="glyphicon glyphicon-arrow-right"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;"
id="up">

                <span class="glyphicon glyphicon-arrow-up"></span>
            </button>
            <button class="btn btn-lg btn-inverse" style="padding: 0px 5px;"
id="down">

                <span class="glyphicon glyphicon-arrow-down"></span>
            </button>

        </div>
    </div>
</div>
<div class="container" style="margin-left: inherit;">
    <div class="row">
        <div class="col-md-4" id="deneme">
            <div class="row">

```

```

<div class="col-md-4" style="margin-left: 0px;">
  <div class="tabbable">
    <ul class="nav flex-column nav-pills " style="font-
size:12px; " id="myTab4" >
      <li class="active ">
        <a data-toggle="tab" data-faruk="resim"
style="background-color: #555;color: white;" href="#Referans">Referans</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="kafa"
style="background-color: #555;color: white;" href="#Kafa">Kafa</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="kas"
style="background-color: #555;color: white;" href="#Kaş">Kaş</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="goz"
style="background-color: #555;color: white;" href="#Göz">Göz</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="burun"
style="background-color: #555;color: white;" href="#Burun">Burun</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="agiz"
style="background-color: #555;color: white;" href="#Ağız">Ağız</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="sakal"
style="background-color: #555;color: white;" href="#Sakal">Sakal</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="ksakal"
style="background-color: #555;color: white;" href="#ksakal">Kısa Sakal</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="biyik"
style="background-color: #555;color: white;" href="#Bıyık">Bıyık</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="sac"
style="background-color: #555;color: white;" href="#Saç">Erkek Saç</a>
      </li>
      <li >
        <a data-toggle="tab" data-
faruk="Bayan_Sc" style="background-color: #555;color: white;" href="#Bayan_Sc">Bayan Saç</a>
      </li>
      <li >
        <a data-toggle="tab" data-faruk="gzluk"
style="background-color: #555;color: white;" href="#Gözlük">Gözlük</a>
      </li>
    </ul>
  </div>
</div>

```

```

<a data-toggle="tab" data-faruk="kulak"
style="background-color: #555;color: white;" href="#kulak">Kulak</a>
</li>
</ul>
</div>
</div>
<div class="col-md-8">
<div class="tab-content " style="overflow: scroll;height:550px">
<div id="Referans" class="tab-pane in active" >
<div style="auto; margin-left: 2%; ">
<input class="btn-sm" type="file"
id="referans_resim">
<input class="btn btn-sm btn-inverse"
type="submit" id="ekle" value="Ekle">
</div>
</div>
<div id="Kafa" class="tab-pane">
<div class="row justify-content-md-center">
<div class="card2 mb-4" >
<a href="#" id="kafa_1">

</a>
</div>
&nbsp;
<div class="card2 mb-4">
<a href="#" id="kafa_2">

</a>
</div>
&nbsp;
<div class="card2 mb-4">
<a href="#" id="kafa_3">

</a>
</div>
&nbsp;
<div class="card2 mb-4">
<a href="#" id="kafa_4">

</a>
</div>
&nbsp;
<div class="card2 mb-4">
<a href="#" id="kafa_5">

```

```



</a>
</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="kafa_6">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="kafa_7">
      

      </a>
    </div>
  </div>
</div>
<div id="Saç" class="tab-pane ">
  <div class="row justify-content-md-center">
    <div class="card2 mb-4" >
      <a href="#"
id="sac_1">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="sac_2">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="sac_3">
            

            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="sac_4">

```

```



</a>
</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="sac_5">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="sac_6">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="sac_7">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4" >
        <a href="#"
id="sac_8">

        </a>
      </div>
    </div>
  </div>
  <div id="Bayan_Sc" class="tab-pane ">
    <div class="row justify-content-md-center">
      <div class="card2 mb-4">
        <a href="#" id="Bayan_Sc_1">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="Bayan_Sc_2">

```



```



</a>
</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="Bayan_Sc_3">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="Bayan_Sc_4">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="Bayan_Sc_5">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="Bayan_Sc_6">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="Bayan_Sc_7">
            

            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="Bayan_Sc_8">
              

              </a>
            </div>

```

```

&nbsp;
<div class="card2 mb-4">
  <a href="#" id="Bayan_Sc_9">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="Bayan_Sc_10">
    
    </a>
  </div>
</div>
<div id="Kaş" class="tab-pane">
  <div class="row justify-content-md-center">
    <div class="card2 mb-4" >
      <a href="#"
id="kas_1">
        
        </a>
      </div>
      <div class="card2 mb-4">
        <a href="#" id="kas_2">
          
          </a>
        </div>
        <div class="card2 mb-4">
          <a href="#" id="kas_3">
            
            </a>
          </div>
          <div class="card2 mb-4">
            <a href="#" id="kas_4">
              
              </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">
              <a href="#" id="kas_5">

```

```
  
  
</div>  
&nbsp;  
<div class="card2 mb-4">  
    <a href="#" id="kas_6">  
          
  
        </a>  
    </div>  
    &nbsp;  
    <div class="card2 mb-4">  
        <a href="#" id="kas_7">  
              
  
            </a>  
        </div>  
        &nbsp;  
        <div class="card2 mb-4" >  
            <a href="#"  
id="kas_8">  
  
                  
  
                </a>  
            </div>  
            &nbsp;  
            <div class="card2 mb-4">  
                <a href="#" id="kas_9">  
                      
  
                    </a>  
                </div>  
                &nbsp;  
                <div class="card2 mb-4">  
                    <a href="#" id="kas_10">  
                          
  
                        </a>  
                    </div>  
                    &nbsp;  
                    <div class="card2 mb-4">  
                        <a href="#" id="kas_11">  
                              
  
                            </a>
```

```

</div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="kas_12">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="kas_13">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="kas_14">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="kas_15">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="kas_16">
        
        </a>
    </div>
</div>
</div>
<div id="Göz" class="tab-pane">
    <div class="row justify-content-md-center">
        <div class="card2 mb-4">
            <a href="#"
id="goz_1">
                
            </a>
        </div>
    </div>

```

```

&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_2">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_3">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_26">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_4">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_5">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="goz_6">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#"
id="goz_7">

```

```



</a>

</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_8">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="goz_9">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="goz_10">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="goz_11">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="goz_12">
            

            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="goz_13">
              

              </a>
            </div>
          &nbsp;
        </div>
      &nbsp;
    </div>
  &nbsp;
</div>

```

```

&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_14">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_15">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_16">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_17">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_18">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_19">
    
    </a>
  </div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_20">

```

```



</a>

</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="goz_21">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="goz_22">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="goz_23">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="goz_24">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="goz_25">
            

            </a>
          </div>
        </div>
      </div>
    <div id="Burun" class="tab-pane">
      <div class="row justify-content-md-center">
        <div class="card2 mb-4" >
          <a href="#"
id="burun_1">

```



```



</a>

</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="burun_2">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="burun_3">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="burun_4">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="burun_5">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="burun_6">
            

            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="burun_7">
              

              </a>
            </div>
          &nbsp;
        </div>
      &nbsp;
    </div>
  &nbsp;
</div>

```

```

&nbsp;
<div class="card2 mb-4" >
    <a href="#"
id="burun_8">
        
    </a>
</div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="burun_9">
        
    </a>
</div>
</div>
<div id="Ağız" class="tab-pane">
    <div class="row justify-content-md-center">
        <div class="card2 mb-4" >
            <a href="#"
id="agiz_1">
                
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_2">
                
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_3">
                
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_4">
                
            </a>
        </div>
    </div>

```

```

&nbsp;
<div class="card2 mb-4">
    <a href="#" id="agiz_5">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="agiz_6">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="agiz_7">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4" >
    <a href="#"
id="agiz_8">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4" >
    <a href="#"
id="agiz_8">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="agiz_10">
        
        </a>
    </div>
&nbsp;
<div class="card2 mb-4">
    <a href="#" id="agiz_11">

```

```



</a>
</div>
&nbsp;
<div class="card2 mb-4">
  <a href="#" id="agiz_12">
    

    </a>
  </div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="agiz_13">
      

      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="agiz_14">
        

        </a>
      </div>
      &nbsp;
      <div class="card2 mb-4">
        <a href="#" id="agiz_15">
          

          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="agiz_16">
            

            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="agiz_17">
              

              </a>
            </div>
          &nbsp;
        </div>
      &nbsp;
    </div>
  &nbsp;
</div>

```

```

        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_18">
                
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_19">
                
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="agiz_20">
                
            </a>
        </div>
    </div>
    <div id="Sakal" class="tab-pane">
        <div class="row justify-content-md-center">
            <div class="card2 mb-4">
                <a href="#" id="sakal_1">
                    
                </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">
                <a href="#" id="sakal_2">
                    
                </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">
                <a href="#" id="sakal_3">
                    
                </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">

```

```

<a href="#" id="sakal_4">
  
  </a>
</div>
  &nbsp;
  <div class="card2 mb-4">
    <a href="#" id="sakal_5">
      
      </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
      <a href="#" id="sakal_6">
        
        </a>
      </div>
    </div>
  </div>
  <div id="ksakal" class="tab-pane">
    <div class="row justify-content-md-center">
      <div class="card2 mb-4">
        <a href="#" id="ksakal_1">
          
          </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
          <a href="#" id="ksakal_2">
            
            </a>
          </div>
          &nbsp;
          <div class="card2 mb-4">
            <a href="#" id="ksakal_3">
              
              </a>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>
  <div id="Bıyık" class="tab-pane">
    <div class="row justify-content-md-center">

```

```

<div class="card2 mb-4" >
    <a href="#"
id="biyik_1">
        
        </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
        <a href="#" id="biyik_2">
            
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="biyik_3">
                
                </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">
                <a href="#" id="biyik_4">
                    
                    </a>
                </div>
                &nbsp;
                <div class="card2 mb-4">
                    <a href="#" id="biyik_5">
                        
                        </a>
                    </div>
                </div>
            </div>
            <div id="Gözlük" class="tab-pane">
                <div class="row justify-content-md-center">
                    <div class="card2 mb-4" >
                        <a href="#"
id="gzluk_1">
                            
                            </a>
                        </div>
                    &nbsp;

```

```

<div class="card2 mb-4">
    <a href="#" id="gzluk_2">
        
        </a>
    </div>
    &nbsp;
    <div class="card2 mb-4">
        <a href="#" id="gzluk_3">
            
            </a>
        </div>
        &nbsp;
        <div class="card2 mb-4">
            <a href="#" id="gzluk_4">
                
                </a>
            </div>
            &nbsp;
            <div class="card2 mb-4">
                <a href="#" id="gzluk_5">
                    
                    </a>
                </div>
                &nbsp;
                <div class="card2 mb-4">
                    <a href="#" id="gzluk_5">
                        
                        </a>
                    </div>
                </div>
            </div>
            <div id="kulak" class="tab-pane">
                <div class="row justify-content-md-center">
                    <div class="card2 mb-4">
                        <a href="#"
id="kulak_1">
                            
                            </a>
                        </div>
                    <div class="card2 mb-4">

```



```

<a href="#" id="kulak_2">
  
</a>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="col-md-6" id="deneme1" style="left: 73%;">
  <div class="col-md-6">
    <div class="tab-content " style="overflow: scroll;height:550px;
padding-left: initial;">
      <div >
        <div>
          <div style="margin-left:20%">
            <input class="btn btn-
sm btn-inverse" type="submit" id="Rekle" value="Referans Aile Getir">
          </div>
          &nbsp;
          <div >
            <a href="#"
id="zoomT">
              <div
id="Rresim_1" class="col-md-6"></div>
            </a>
          </div>
          <div >
            <a href="#"
id="zoomT">
              <div
id="Rresim_2" class="col-md-6" ></div>
            </a>
          </div>
          &nbsp;
          <div >
            <a href="#"
id="zoomT">
              <div
id="Rresim_3" class="col-md-6"></div>
            </a>
          </div>
          &nbsp;
          <div >
            <a href="#"
id="zoomT">
              <div
id="Rresim_4" class="col-md-6" ></div>
            </a>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>

```

```

id="zoomT">
id="Rresim_5" class="col-md-6"></div>
id="zoomT">
id="Rresim_6" class="col-md-6" ></div>
id="zoomT">
id="Rresim_7" class="col-md-6"></div>
id="zoomT">
id="Rresim_8" class="col-md-6" ></div>
</div>
</div>
</div>
<div id="yazdiralan" class="col-md-8"
style="width:650px;height:800px;position:relative;top:10px;left:15%;">
  <div id="img_resim" ></div>
  <div id="img_kafa"></div>
  <div id="img_sac"></div>
  <div id="img_Bayan_Sc"></div>
  <div id="img_kas"></div>
  <div id="img_goz"></div>
  <div id="img_burun"></div>
  <div id="img_agiz"></div>
  <div id="img_biyik"></div>
  <div id="img_sakal"></div>
  <div id="img_ksakal"></div>
  <div id="img_kulak"></div>
  <div id="img_gzluk"></div>
  <div id="zoom"></div>

```

```

        </div> <!-- /col-md-4 -->
    </div>
</div>
<div>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js"></script>
    <script type="text/javascript">
        if("ontouchend" in document) document.write("<script
src='assets/js/jquery.mobile.custom.min.js'>"+"<"+"/script>");
    </script>
    <script src="assets/js/bootstrap.min.js"></script>
    <script src="assets/js/typeahead-bs2.min.js"></script>
    <script src="assets/js/excanvas.min.js"></script>
    <script src="assets/js/jquery-ui-1.10.3.custom.min.js"></script>
    <script src="assets/js/jquery.ui.touch-punch.min.js"></script>
    <script src="assets/js/chosen.jquery.min.js"></script>
    <script src="assets/js/fuelux/fuelux.spinner.min.js"></script>
    <script src="assets/js/date-time/bootstrap-datepicker.min.js"></script>
    <script src="assets/js/date-time/bootstrap-timepicker.min.js"></script>
    <script src="assets/js/date-time/moment.min.js"></script>
    <script src="assets/js/date-time/daterangepicker.min.js"></script>
    <script src="assets/js/bootstrap-colorpicker.min.js"></script>
    <script src="assets/js/jquery.knob.min.js"></script>
    <script src="assets/js/jquery.autosize.min.js"></script>
    <script src="assets/js/jquery.inputlimiter.1.3.1.min.js"></script>
    <script src="assets/js/jquery.maskedinput.min.js"></script>
    <script src="assets/js/bootstrap-tag.min.js"></script>
    <script>
        $('#deneme')
        .draggable()
        $('#deneme1')
        .draggable()
        $('#img_sac')
        .draggable()
        $('#img_Bayan_Sc')
        .draggable()
        $('#img_kafa')
        .draggable()
        $('#img_kas')
        .draggable()
        $('#img_goz')
        .draggable()
        $('#img_gzluk')
        .draggable()
        $('#img_kulak')
        .draggable()
        $('#img_burun')
        .draggable()
        $('#img_biyik')
        .draggable()
        $('#img_agiz')
        .draggable()
    </script>

```

```

$('#img_sakal')
.draggable();
$('#img_ksakal')
.draggable();
$('#img_resim')
.draggable();
$('#zoom')
.draggable();
$('.genel')
.resizable();
$(document).ready(function(){
    $(function(){
        $("h1").draggable();
        $("h1,img,p,span,div").dblclick(function(){
        });
    });
});
window.oncontextmenu = function () {
}
$(window).on('mousedown', function(e) {
    if( e.which == 2 ) {
        e.preventDefault();
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        $('#img_'+aktif_tab).attr('style',"background:none");
    }
});
$(document).ready(function(){
    }).keydown(function(event)
    {
        var tus = new Array(37, 38, 39, 40);
        var kod = (event.which) ? event.which : event.keyCode;
        if(event.keyCode==37){ var aktif_tab = $('#myTab4 li.active a').data('faruk');
            $('#img_'+aktif_tab).css('left',$('#img_'+aktif_tab).offset().left-artis);
        }

        if(event.keyCode==38){ var aktif_tab = $('#myTab4 li.active a').data('faruk');
            $('#img_'+aktif_tab).css('top',$('#img_'+aktif_tab).offset().top-artis);
        }

        if(event.keyCode==39){ var aktif_tab = $('#myTab4 li.active a').data('faruk');

        $('#img_'+aktif_tab).css('left',$('#img_'+aktif_tab).offset().left+artis); }
        if(event.keyCode==40){ var aktif_tab = $('#myTab4 li.active
a').data('faruk');
        $('#img_'+aktif_tab).css('top',$('#img_'+aktif_tab).offset().top+artis); }
    });
</script>
<!-- ace scripts -->
<script src="assets/js/ace-elements.min.js"></script>
<script src="assets/js/ace.min.js"></script>
<!-- inline scripts related to this page -->
<script type="text/javascript">
$('#yazdirbtn').on('click', function(){
    var yazdirilacakAlan= document.getElementById('divyazdir').innerHTML;
    var orjinalSayfa= document.body.innerHTML;

```

```

document.body.innerHTML = yazdirilacakAlan;
window.print();
document.body.innerHTML = orjinalSayfa;
});
</script>
<script type="text/javascript">
    function print_(){
        var yazdirilacakAlan= document.getElementById('yazdiralan').innerHTML;
        var orjinalSayfa= document.body.innerHTML;
        document.body.innerHTML = yazdirilacakAlan;
        window.print();
        document.body.innerHTML = orjinalSayfa;
    }
    function printPage(elementId) {
        var printContent = document.getElementById(elementId);
        var windowUrl = 'about:blank';
        var uniqueName = new Date();
        var windowName = 'Print' + uniqueName.getTime();
        var printWindow = window.open(windowUrl, windowName,
'left=50000,top=50000,width=0,height=0');

        printWindow.document.write(printContent.innerHTML);
        printWindow.document.close();
        printWindow.focus();
        printWindow.print();
        printWindow.close();
    }
    function makeScreenshot() {
        html2canvas(window.parent.document.body, {
            onrendered: function(canvas) {
                var cand = document.getElementsByTagName('canvas');
                if(cand[0] === undefined || cand[0] === null){

                }else{
                    //cand[0].remove();
                    document.body.removeChild(cand[0]);
                }
                document.body.appendChild(canvas);
            }
        });
    }

    function prnt() {
        html2canvas(document.getElementById("yazdiralan"), { scale:
1 }).then(canvas => {

            document.body.appendChild(canvas);

        });
    }
    var artis = 3;
    jQuery(function($) {
        $('#a[id^="kafa"]').click(function(){
            $('#img_kafa').attr('style','background:url("'" + $(this).find('img').attr('src') + "'") no-repeat
;width:300px;height:450px;position:fixed;z-index:1;top:25%;left:42%");

```

```

    });
    $('a[id^="kulak"]').click(function(){
        $('#img_kulak').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:350px;height:120px;position:fixed;z-index:1;top:44%;left:40%");
    });
    $('a[id^="sac"]').click(function(){
        $('#img_sac').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:330px;height:300px;position:fixed;z-index:6;top:18%;left:41%");
    });
    $('a[id^="Bayan_Sc"]').click(function(){
        $('#img_Bayan_Sc').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:450px;height:600px;position:fixed;z-index:6;top:17%;left:38%");
    });
    $('a[id^="kas"]').click(function(){
        $('#img_kas').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:235px;height:40px;position:fixed;z-index:3;top:40%;left:44%");
    });
    $('a[id^="goz"]').click(function(){
        $('#img_goz').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:240px;height:110px;position:fixed;z-index:2;top:42%;left:44%");
    });
    $('a[id^="burun"]').click(function(){
        $('#img_burun').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:140px;height:130px;position:fixed;z-index:3;top:49%;left:46%");
    });
    $('a[id^="agiz"]').click(function(){
        $('#img_agiz').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:170px;height:90px;position:fixed;z-index:2;top:60%;left:45%");
    });
    $('a[id^="biyik"]').click(function(){
        $('#img_biyik').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:150px;height:60px;position:fixed;z-index:2;top:62%;left:47%");
    });
    $('a[id^="sakal"]').click(function(){
        $('#img_sakal').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-
repeat;width:250px;height:240px;position:fixed;z-index:2;top:52%;left:43%");
    });
    $('a[id^="ksakal"]').click(function(){
        $('#img_ksakal').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:150px;height:90px;position:fixed;z-index:2;top:66%;left:47%");
    });
    $('a[id^="gzluk"]').click(function(){
        $('#img_gzluk').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:300px;height:70px;position:fixed;z-index:5;top:42%;left:44%");
    });
    $('a[id^="zoomT"]').click(function(){
        $('#zoom').attr('style',"background:url('"+$(this).find('img').attr('src')+ "') no-repeat
;width:300px;height:400px;position:fixed;z-index:5;top:60%;left:70%");
        $('#zoom').css('background-size','200+px'+ ' '+280+ 'px');
    });
    $('#ekle').on('click', function(){
        $('#img_resim').attr('style',"background:url('c:/referans_resim/'+$( '#referans_resim').prop('files

```

```

')[0].name+"") 0% 0% / 350px 350px no-repeat !important;width:350px;height:350px;position:fixed;z-
index:9999;top:10px;left:42%");
});
$('#Rekle').on('click', function(){
    document.getElementById("Rresim_1").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/1.jpg' >" ;
    document.getElementById("Rresim_2").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/2.jpg' >" ;
    document.getElementById("Rresim_3").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/3.jpg' >" ;
    document.getElementById("Rresim_4").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/4.jpg' >" ;
    document.getElementById("Rresim_5").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/5.jpg' >" ;
    document.getElementById("Rresim_6").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/6.jpg' >" ;
    document.getElementById("Rresim_7").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/7.jpg' >" ;
    document.getElementById("Rresim_8").innerHTML ="<img
width=100px height=150px src='c:/referans_resim/8.jpg' >" ;
});
$('#left').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    $('#img_'+aktif_tab).css('left',$('#img_'+aktif_tab).offset().left-artis);
});
$('#right').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    $('#img_'+aktif_tab).css('left',$('#img_'+aktif_tab).offset().left+artis);
});
$('#up').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    $('#img_'+aktif_tab).css('top',$('#img_'+aktif_tab).offset().top-artis);
});
$('#down').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    $('#img_'+aktif_tab).css('top',$('#img_'+aktif_tab).offset().top+artis);
});
$('#horizontal_').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    var oran =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))+artis;
    $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
    $('#img_'+aktif_tab).css('background-size',oran+'px
'+($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))+px');
    $('#img_'+aktif_tab).css('width',oran+'px');
    $('#img_'+aktif_tab).css('height',($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace
(' ','').substring(0,3))+px');
});
$('#horizontal_small_').on('click', function(){
    var aktif_tab = $('#myTab4 li.active a').data('faruk');
    var oran =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))-artis;
    $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');

```

```

        $('#img_'+aktif_tab).css('background-size',oran+'px
'+($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))+px');
        $('#img_'+aktif_tab).css('width',oran+'px');
        $('#img_'+aktif_tab).css('height',($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(
(' ','').substring(0,3))+px');
    });
    $('#vertical_').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))+artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-
size',($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))+px+'
'+oran+'px');
        $('#img_'+aktif_tab).css('height',oran+'px');
        $('#img_'+aktif_tab).css('width',($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(
' ','').substring(0,3))+px');
    });
    $('#vertical_small_').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))-artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-
size',($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))+px+'
'+oran+'px');
        $('#img_'+aktif_tab).css('height',oran+'px');
        $('#img_'+aktif_tab).css('width',($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(
' ','').substring(0,3))+px');
    });
    $('#small').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))-artis;
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))-artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#horizontal_small').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))-artis;
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3));
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#vertical_small').on('click', function(){

```



```

        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3));
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))-artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#horizontal').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))+artis;
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3));
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#vertical').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3));
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))+artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#full').on('click', function(){
        var aktif_tab = $('#myTab4 li.active a').data('faruk');
        var oran1 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[1].split(':')[1].replace(' ','').substring(0,3))+artis;
        var oran2 =
parseInt($('#img_'+aktif_tab).attr('style').split(';')[2].split(':')[1].replace(' ','').substring(0,3))+artis;
        $('#img_'+aktif_tab).css('background-repeat','no-repeat !important');
        $('#img_'+aktif_tab).css('background-size',oran1+'px '+oran2+'px');
        $('#img_'+aktif_tab).css('width',oran1+'px');
        $('#img_'+aktif_tab).css('height',oran2+'px');
    });
    $('#select').chosen();
    $('#id-disable-check').on('click', function() {
        var inp = $('#form-input-readonly').get(0);
        if(inp.hasAttribute('disabled')) {
            inp.setAttribute('readonly', 'true');
            inp.removeAttribute('disabled');
            inp.value="This text field is readonly!";
        }
        else {
            inp.setAttribute('disabled', 'disabled');

```

```

        inp.removeAttribute('readonly');
        inp.value="This text field is disabled!";
    }
});
$('[data-rel=tooltip]').tooltip({ container:'body' });
$('[data-rel=popover]').popover({ container:'body' });

$('textarea[class*=autosize]').autosize({ append: "\n" });
$('textarea.limited').inputlimiter({
    remText: '%n character%s remaining...',
    limitText: 'max allowed : %n.'
});
$.mask.definitions['~']='[+-]';
$('.input-mask-date').mask('99/99/9999');
$('.input-mask-phone').mask('(999) 999-9999');
$('.input-mask-eyescript').mask('~9.99 ~9.99 999');
$(".input-mask-product").mask("a*-999-a999",{placeholder:"
",completed:function(){alert("You typed the following: "+this.val());}});
$( "#input-size-slider" ).css('width','200px').slider({
    value:1,
    range: "min",
    min: 1,
    max: 8,
    step: 1,
    slide: function( event, ui ) {
        var sizing = [ "input-sm", 'input-lg', 'input-mini', 'input-
small', 'input-medium', 'input-large', 'input-xlarge', 'input-xxlarge'];
        var val = parseInt(ui.value);
        $('#form-field-4').attr('class',
sizing[val]).val(''+sizing[val]);
    }
});
$( "#input-span-slider" ).slider({
    value:1,
    range: "min",
    min: 1,
    max: 12,
    step: 1,
    slide: function( event, ui ) {
        var val = parseInt(ui.value);
        $('#form-field-5').attr('class', 'col-xs-'+val).val('col-xs-
'+val);
    }
});
$( "#slider-range" ).css('height','200px').slider({
    orientation: "vertical",
    range: true,
    min: 0,
    max: 100,
    values: [ 17, 67 ],
    slide: function( event, ui ) {
        var val = ui.values[$(ui.handle).index()-1]+"";

```

```

        if(! ui.handle.firstChild ) {
            $(ui.handle).append("<div class='tooltip right in'
style='display:none;left:16px;top:-6px;'><div class='tooltip-arrow'></div><div class='tooltip-
inner'></div></div>");
        }
        $(ui.handle.firstChild).show().children().eq(1).text(val);
    }
}).find('a').on('blur', function(){
    $(this.firstChild).hide();
});
$( "#slider-range-max" ).slider({
    range: "max",
    min: 1,
    max: 10,
    value: 2
});
$( "#eq > span" ).css({ width:'90%', 'float':'left',
margin:'15px'}).each(function() {
    // read initial values from markup and remove that
    var value = parseInt( $( this ).text(), 10 );
    $( this ).empty().slider({
        value: value,
        range: "min",
        animate: true
    });
});
$('#id-input-file-1 , #id-input-file-2').ace_file_input({
    no_file:'No File ...',
    btn_choose:'Choose',
    btn_change:'Change',
    droppable:false,
    onchange:null,
    thumbnail:false //| true | large
    //whitelist:'gif|png|jpg|jpeg'
    //blacklist:'exe|php'
    //onchange:"
    //
});
$('#id-input-file-3').ace_file_input({
    style:'well',
    btn_choose:'Drop files here or click to choose',
    btn_change:null,
    no_icon:'icon-cloud-upload',
    droppable:true,
    thumbnail:'small'//large | fit
    //,icon_remove:null//set null, to hide remove/reset button
    /**,before_change:function(files, dropped) {
        //Check an example below
        //or examples/file-upload.html
        return true;
    }*/
    /**,before_remove : function() {
        return true;
    }

```

```

    }*/
    preview_error : function(filename, error_code) {
        //name of the file that failed
        //error_code values
        //1 = 'FILE_LOAD_FAILED',
        //2 = 'IMAGE_LOAD_FAILED',
        //3 = 'THUMBNAIL_FAILED'
        //alert(error_code);
    }
}).on('change', function(){
    //console.log($(this).data('ace_input_files'));
    //console.log($(this).data('ace_input_method'));
});
//dynamically change allowed formats by changing before_change callback
function
$( '#id-file-format').removeAttr('checked').on('change', function() {
    var before_change
    var btn_choose
    var no_icon
    if(this.checked) {
        btn_choose = "Drop images here or click to choose";
        no_icon = "icon-picture";
        before_change = function(files, dropped) {
            var allowed_files = [];
            for(var i = 0 ; i < files.length; i++) {
                var file = files[i];
                if(typeof file === "string") {
                    //IE8 and browsers that don't
                    if(!
                    (/\.(jpe?g|png|gif|bmp)$/i).test(file) ) return false;
                }
                else {
                    var type = $.trim(file.type);
                    if( ( type.length > 0 && !
                    (/^image\/(jpe?g|png|gif|bmp)$/i).test(type) )
                    || ( type.length
                    === 0 && ! (/\.(jpe?g|png|gif|bmp)$/i).test(file.name) )//for android's default browser which gives an
                    empty string for file.type
                    ) continue;//not an
                    image so don't keep this file
                }
                allowed_files.push(file);
            }
            if(allowed_files.length === 0) return false;
            return allowed_files;
        }
    }
    else {
        btn_choose = "Drop files here or click to choose";
        no_icon = "icon-cloud-upload";
    }
}

```

```

        before_change = function(files, dropped) {
            return files;
        }
    }
    var file_input = $('#id-input-file-3');
    file_input.ace_file_input('update_settings',
    { 'before_change':before_change, 'btn_choose': btn_choose, 'no_icon':no_icon })
    file_input.ace_file_input('reset_input');
    });
    $('#spinner1').ace_spinner({ value:0,min:0,max:200,step:10,
btn_up_class:'btn-info' , btn_down_class:'btn-info'})
    .on('change', function(){
        //alert(this.value)
    });
    $('#spinner2').ace_spinner({ value:0,min:0,max:10000,step:100,
touch_spinner: true, icon_up:'icon-caret-up', icon_down:'icon-caret-down'});
    $('#spinner3').ace_spinner({ value:0,min:-100,max:100,step:10, on_sides:
true, icon_up:'icon-plus smaller-75', icon_down:'icon-minus smaller-75', btn_up_class:'btn-success',
btn_down_class:'btn-danger'});

    $('.date-picker').datepicker({ autoclose:true}).next().on(ace.click_event,
function(){
        $(this).prev().focus();
    });
    $('input[name=date-range-
picker]').daterangepicker().prev().on(ace.click_event, function(){
        $(this).next().focus();
    });

    $('#timepicker1').timepicker({
        minuteStep: 1,
        showSeconds: true,
        showMeridian: false
    }).next().on(ace.click_event, function(){
        $(this).prev().focus();
    });

    $('#colorpicker1').colorpicker();
    $('#simple-colorpicker-1').ace_colorpicker();

    $(".knob").knob();
    //we could just set the data-provide="tag" of the element inside HTML, but
    IE8 fails!

    var tag_input = $('#form-field-tags');
    if(! (/msie\s*(8|7|6)/.test(navigator.userAgent.toLowerCase()))) )
    {
        tag_input.tag(
        {
            placeholder:tag_input.attr('placeholder'),
            //enable typeahead by specifying the source array
            source: ace.variable_US_STATES, //defined in ace.js >>
        }
    )
    }
    ace.enable_search_ahead

```

```

        }
    );
}
else {
    //display a textarea for old IE, because it doesn't support this plugin
    or another one I tried!
    tag_input.after('<textarea id="'+tag_input.attr('id')+'"'
name="'+tag_input.attr('name')+'"' rows="3">'+tag_input.val()+'</textarea>').remove();
    //$('#form-field-tags').autosize({ append: "\n" });
}
$('#modal-form input[type=file]').ace_file_input({
    style:'well',
    btn_choose:'Drop files here or click to choose',
    btn_change:null,
    no_icon:'icon-cloud-upload',
    droppable:true,
    thumbnail:'large'
}))

//chosen plugin inside a modal will have a zero width because the select
element is originally hidden
//and its width cannot be determined.
//so we set the width after modal is show
$('#modal-form').on('shown.bs.modal', function () {
    $(this).find('.chosen-container').each(function(){
        $(this).find('a:first-child').css('width' , '210px');
        $(this).find('.chosen-drop').css('width' , '210px');
        $(this).find('.chosen-search input').css('width' , '200px');
    });
});
});
</script>
</body>
</html>

```



## **ÖZ GEÇMİŞ**

Van'ın Merkez ilçesine bağlı İskele mahallesinde 1983 yılında doğdu. İlköğretim ve liseyi Van'da tamamladı. Süleyman Demirel Üniversitesi, Bilgisayar Sistemleri Öğretmenliği bölümünden 2008 yılında mezun oldu. Yüksek lisans eğitimini 2013 yılında Süleyman Demirel Üniversitesi Elektronik Bilgisayar Ana Bilim Dalında tamamladı. Halen Van Yüzüncü Yıl Üniversitesi Başkale Meslek Yüksek Okulu'nda Öğretim Görevlisi olarak görev yapmaktadır.