

T.C.
KASTAMONU ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI



KÜTÜPHANE ORTAMINDA YÜZ VE PARMAK İZİ TANIMA
SİSTEMİNİN GELİŞTİRİLMESİ

MOHAMMED RIDHA MOHAMMED AHMED ALSARRAR

DOKTORA TEZİ

DOÇ. DR. YASEMİN GÜLTEPE

HAZİRAN - 2023

KASTAMONU

TAAHHÜTNAME

Bu tezin tasarımı, hazırlanması, yürütülmesi, araştırmalarının yapılması ve bulgularının analizlerinde 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, bilimsel etiğe uygun olarak kaynak gösterildiğini bildirir ve taahhüt ederim.

Mohammed Ridha Mohammed Ahmed ALSARRAR

ÖZET

DOKTORA TEZİ

KÜTÜPHANE ORTAMINDA YÜZ VE PARMAK İZİ TANIMA SİSTEMİNİN GELİŞTİRİLMESİ

MOHAMMED RIDHA MOHAMMED AHMED ALSARRAR
KASTAMONU ÜNİVERSİTESİ FEN BİLİMLERİ ENSTİTÜSÜ
MALZEME BİLİMİ VE MÜHENDİSLİĞİ ANA BİLİM DALI
DANIŞMAN: DOÇ. DR. YASEMİN GÜLTEPE

Yönetim bilişim sistemleri, bilgisayar teknolojisinin yaygınlaşmasıyla birlikte çeşitli alanlarda benimsenmiştir. Bir kütüphane yönetim sistemi, bir bilgi sistemine örnektir. Son yıllarda yapay zekâ uygulamaları içeren bilgisayar tabanlı sistemler kütüphane bilgi sistemleri için kullanılmaktadır. Yapay sinir ağları, karmaşık bilgileri analiz etmek için güçlü bir araç olarak kullanılabilen yapay zekânın bir dalı olarak niteliksel bir yöntemdir. Derin öğrenmede, bir Evrişimsel Sinir Ağı görsel görüntüleri analiz etmek için en yaygın olarak uygulanan bir yapay sinir ağı sınıfıdır. Bu tez çalışmasının amacı, Yapay Sinir Ağları ve Evrişimsel Sinir Ağları kullanarak kütüphane ortamında yüz ve parmak izi tanıma sistemini gerçekleştirmektir. Bunun için Parmak izi veri setleri, dört parmak izi grubuna kesin olarak sınıflandırılmış 100 gerçekçi parmak izinden oluşmaktadır. Parmak izi görüntülerinden özneteliklerin çıkarılması için gri seviye oluşum matrisi kullanılmıştır. OpenCV, bilgisayarla görme uygulamaları için çeşitli işlevler sağlar. Bu tez çalışmasında OpenCV'nin yüz algılama ve yüz tanıma için kullanılan OpenCV'deki popüler algoritmaları kullanılmıştır. Yüz tanıma sistemi için ortalama ve weiner filtreleme işlemleri kullanılmıştır. Bu çerçevede parmak tanıma için FVC 2000, FVC 2002 ve FVC 2004 veri setleri; yüz tanıma sistemi için de gerçek zamanlı olarak yüz tanıma yapılarak sistemde oluşturulan veri seti üzerinde deneysel çalışmalar gerçekleştirilmiştir. Sonuç olarak, sınıflandırıcının görüntü işlemlerini hızlandırarak gerçek zamanlı sınıflandırma durumlarına fayda sağlar.

ANAHTAR KELİMELER:Kütüphane bilgi sistemi, yüz tanıma, parmak izi, yapay sinir ağları, evrişimsel sinir ağları

Haziran 2023,70 Sayfa

ABSTRACT

Ph.D THESIS

DEVELOPMENT OF FACE AND FINGERPRINT RECOGNITION SYSTEM IN LIBRARY ENVIRONMENT

MOHAMMED RIDHA MOHAMMED AHMED ALSARRAR

KASTAMONU UNIVERSITY INSTITUTE OF SCIENCE

DEPARTMENT OF MATERIALS SCIENCE AND ENGINEERING

SUPERVISOR: ASSOC. PROF. DR. YASEMİN GÜLTEPE

Management information systems have been adopted in various fields with the spread of computer technology. A library management system is an example of an information system. In recent years, computer-based systems containing artificial intelligence applications have been used for library information systems. Artificial neural networks are a qualitative method as a branch of artificial intelligence that can be used as a powerful tool to analyze complex information. In deep learning, a Convolutional Neural Network is a class of artificial neural networks most widely applied to analyze visual images. The aim of this thesis is to realize the face and fingerprint recognition system in the library environment by using Artificial Neural Networks and Convolutional Neural Networks. For this, Fingerprint datasets consist of 100 realistic fingerprints that are precisely classified into four fingerprint groups. Gray level formation matrix was used to extract features from fingerprint images. OpenCV provides various functions for computer vision applications. In this thesis, OpenCV' popular algorithms in OpenCV, which is used for face detection and face recognition, are used. Average and weiner filtering processes are used for the face recognition system. In this framework, FVC 2000, FVC 2002 and FVC 2004 data sets for finger recognition; For the face recognition system, experimental studies were carried out on the data set created in the system by real-time face recognition. As a result, it benefits real-time classification situations by speeding up the image processing of the classifier.

KEYWORDS:Library information system, face recognition, fingerprint, artificial neural networks, convolutional neural networks

June 2023, 70 Page

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde ve oluşumunda engin bilgi ve birikimlerinden, bilimsel temellerin oluşmasında ilgi ve desteğini esirgemeyen, çalışmalarına yön veren, bilgilendiren değerli hocam Doç. Dr. Yasemin GÜLTEPE'ye teşekkürü bir borç bilirim. Ayrıca tez izleme komitesindeki üyelerime Doç. Dr. Can Doğan VURDU ve Doç. Dr. Nejdet GÜLTEPE'ye teşekkür ederim. Tezin sonuçlandırılmasındaki katkılarından dolayı değerli tez savunma jüri üyeleri Prof. Dr. Erol KURT ve Dr. Öğr. Üyesi Ali Burak ÖNCÜL'e teşekkürlerimi sunarım.

Tezimi gerçekleştirmem ve bu tezi tamamlamam için bana destek ve güven veren Ankara'da yaşayan Libyalı arkadaşlarıma ve manevi hiçbir yardımı esirgemediğim yanımda oldukları için sevgili aileme teşekkür ederim.

Tez çalışmasının sonuçlarının bilişim teknolojileri ve kütüphane bilgi sistemleri alanlarını ilgilendiren faydalı olması ve gelecekte yeni araştırmaları teşvik etmesi için katkı sağlamasını temenni ediyorum.

Kastamonu Üniversitesinde aldığım eğitim için yaptıkları destekten dolayı Libya Hükümetine ve Türkiye'deki Libya Büyükelçiliğine teşekkür ediyorum.

MOHAMMED RIDHA MOHAMMED AHMED ALSARRAR
Kastamonu, 2023

İÇİNDEKİLER

	<u>Sayfa</u>
TEZ ONAYI	ii
TAAHHÜTNAME	iii
ÖZET.....	iv
ABSTRACT	v
TEŞEKKÜR	vi
İÇİNDEKİLER	vii
ŞEKİLLER DİZİNİ	viii
TABLolar DİZİNİ	x
SİMGELER VE KISALTMALAR DİZİNİ	viii
1. GİRİŞ.....	1
2. LİTERATÜR TARAMASI	9
3. YAPAY SİNİR AĞLARI.....	17
4. EVRİŞİMSEL SİNİR AĞLARI.....	21
5. PARMak İZİ VE YÜZ TANIMA.....	24
5.1 Veri Setleri	24
5.2 Gri Düzey Eş Oluşum Matrisi.....	26
5.3 Öznitelik Çıkarma.....	27
5.4 Görüntü İşleme.....	29
5.5 Görüntü İşlemede Kullanılan Temel Filtreler.....	30
6. DENEYSEL ÇALIŞMALAR	33
6.1 Kütüphane Ortamında Parmak İzi Tanıma İçin Yapay Sinir Ağı Yöntemlerinin Uygulanması.....	33
6.2 Evrişimsel Sinir Ağları Kullanılarak Kütüphane Ortamında Parmak İzi Tanıma Sisteminin Geliştirilmesi	43
6.3 Kütüphane Ortamında Yüz Tanıma Sisteminin Geliştirilmesi	49
6.4 Görüntü İşleme Kullanılarak Kütüphane Ortamında Yüz Tanıma Sisteminin Geliştirilmesi.....	62
7. SONUÇLAR VE ÖNERİLER	62
KAYNAKLAR	64
ÖZGEÇMİŞ.....	70

ŞEKİLLER DİZİNİ

Sayfa

Şekil 1.1 Standart bir kütüphanenin genel görünümü.....	1
Şekil 1.2 Bingazi Üniversitesi'nden bir görüntü.....	4
Şekil 1.3 Parmak izi yapısı ve parmak izine ait lokal ve global özellikler	6
Şekil 2.1 Kayıt ve doğrulama içeren parmak izi akış şeması.....	10
Şekil 2.2 Kayıt ve doğrulama içeren parmak izi tanıma algoritması	11
Şekil 2.3 Histogram eşitleme sonucu	15
Şekil 3.1 YSA'da kullanılan yapay nöronların yapısı.....	18
Şekil 3.2 Çok katmanlı algılayıcı model ağ örneği	20
Şekil 4.1 CNN mimarisi.....	22
Şekil 5.1 FVC 2000 her veri setinden bir parmak izi görüntüsü.....	24
Şekil 5.2 FVC 2002 her veri setinden bir parmak izi görüntüsü.....	25
Şekil 5.3 FVC 2004 her veri setinden bir parmak izi görüntüsü.....	26
Şekil 5.4 Ortalama filtreleme yaklaşımı.....	32
Şekil 6.1 Uygulanan çoklu nöronların performansının karşılaştırılması	36
Şekil 6.2 Yapay sinir ağı modeli	38
Şekil 6.3 YSA'nın performans grafiği	39
Şekil 6.4 YSA doğrulama kontrolleri, Mu ve Gradient değerleri	39
Şekil 6.5 Levenberg Marquadt eğitim veri performansı	40
Şekil 6.6 Giriş ekranı.....	40
Şekil 6.7 Veri kümesinin seçilmesi.....	41
Şekil 6.8 Load tuşuna basıldıktan sonraki ekran görüntüsü.....	41
Şekil 6.9 YSA modeli çalıştırma ekranı.....	42
Şekil 6.10 Program çalıştırıldıktan sonra ekran görüntüsü	42
Şekil 6.11 YSA ve CNN verilerinin gösterildiği arayüz.....	43
Şekil 6.12 Model grafiklerinin gösterildiği arayüz	44
Şekil 6.13 Eğitim kaybı.....	45
Şekil 6.14 ROC grafiği.....	46
Şekil 6.15 CNN için eğitim ve deneme başarı grafiği	47
Şekil 6.16 CNN için eğitim kaybı	47
Şekil 6.17 Sonuç ekranı.....	48
Şekil 6.18 (a) Giriş parmak izi, (b) Görüntü zenginleştirme, (c) İkili haldeki görüntü, (d) İnceltilmiş c görüntüsü, (e) İyileştirilmiş d görüntüsü.....	49
Şekil 6.19 Yüz tanıma sisteminin akış şeması	51
Şekil 6.20 Yönetici girişi arayüzü.....	53
Şekil 6.21 Kullanıcı girişi arayüzü.....	53
Şekil 6.22 Kameranın çalışmaya başlaması	54
Şekil 6.23 Tahminlenen görüntü	55
Şekil 6.24 Python kütüphaneleri	55
Şekil 6.25 Yönetici arayüzü, button, textbox ve label kodları	56
Şekil 6.26 Wiener kodları	57
Şekil 6.27 Ortalama kodları	57
Şekil 6.28 Yüz tanımlama kodları.....	58
Şekil 6.29 Benzerlik tanımlama kodları.....	59
Şekil 6.30 Genel dosya yapısı	59

Şekil 6.31 Doğruluk değerleri	61
-------------------------------------	----



TABLÖLAR DİZİNİ

	<u>Sayfa</u>
Tablo 1.1 Dewey ondalık sınıflama sistemi	2
Tablo 1.2 Örnek sosyal bilimler sıralaması	2
Tablo 6.1 FVC 2000 DB1-DB4 veri setlerinin özellikleri	33
Tablo 6.2 FVC 2002 DB1-DB4 veri setlerinin özellikleri	34
Tablo 6.3 FVC 2004 DB1-DB4 veri setlerinin özellikleri	34
Tablo 6.4 ÇKA ağ performansı	36
Tablo 6.5 Önerilen metodun karşılaştırılması	37
Tablo 6.6 YSA parametreleri	37
Tablo 6.7 Görüntülerin ön işleme adımları	60

SİMGELER VE KISALTMALAR DİZİNİ

Simgeler

%	: Yüzde
Ψ	: Transfer fonksiyonu
θ_k	: k sıralı eşik ifadesi
w_{kj}	: k bağlantı ağırlıkları
θ	: Eşik değerini ayarlamak için önyargı
λ	: Öğrenme oranı

Kısaltmalar

AUC	: Eğri Altında Kalan Alan
BMP	: Bitmap Görüntü Dosyası
CD	: Yoğun Disk
CNN	: Evrişimsel Sinir Ağları
ÇKA	: Çok Katmanlı Algılayıcı
ÇKAM	: Çok Katmanlı Algılayıcı Modeli
DDFD	: Çok Katmanlı Algılayıcılar
DDFD	: Derin Yoğun Yüz Dedektörü
DPI	: İnç Başına Düşen Nokta Sayısı
DPM	: Deforme Olabilen Parça Tabanlı Bir Model
DSP	: Dijital Sinyal İşleme
DVM	: Destek Vektör Makineleri
ESA	: Evrişimsel Sinir Ağları
FLOPs	: Saniye Başına Kayan Nokta İşlemi
FVC	: Parmak İzi Doğrulama Yarışması
GIF	: Grafik Değişim Formatı
GLCM	: Gri Düzey Eş Oluşum Matrisi
GUI	: Grafik Kullanıcı Arayüzü
JPEG	: Ortak Fotoğraf Uzmanları Grubu
k-NN	: k-En Yakın Komşu
NIST	: Ulusal Standartlar ve Teknoloji Enstitüsü
NPD	: Normalleştirilmiş Piksel Farkı
MATLAB	: Matris Laboratuvarı
MSE	: Ortalama Kare Hatası
PCA	: Temel Bileşen Analizi
PDBNN	: Sonra Olasılığa Dayalı Karar Tabanlı Sinir Ağı
PPI	: İnç Başına Piksel
PNG	: Taşınabilir Ağ Grafikleri
RBF	: Radyal Temel İşlevi
ReLU	: Rektifiye Doğrusal Birim
RGB	: Kırmızı, Yeşil, Mavi
R-CNN	: Hızlı Bölge Tabanlı Evrişimli Ağlar

ROC	: Alıcı Çalışma Karakteristikleri
ROI	: İlgi Alan
TBA	: Temel Bileşen Analizi
TIFF	: Etiketli Görüntü Dosyası Formatı
TLU	: Eşik Mantık Birimi
YSA	: Yapay Sinir Ağları



1. GİRİŞ

Kütüphaneler içeriğindeki değerli kitap, süreli yayınlar, öğrenci ve araştırmacının ihtiyaç duyduğu tüm bilgilerin öğretim programlarının gereklerini karşılayabilmesi ve en üst düzeye ulaşabilmesi için özellikle üniversitedeki öğrenci ve öğretim üyeleri için bilgi ve veri erişimi için ilk temel başvuru kaynağıdır (Alemna, 2000; Alsarrar ve Gültepe, 2016).



Şekil 1.1 Standart bir kütüphanenin genel görünümü

Üniversite kütüphanesinin misyonu, öğrenciler, akademisyenler ve çalışanlar da dahil olmak üzere üniversiteyi oluşturan tüm insan kaynağının gereksinimlerine cevap veren çeşitli eğitim kaynaklarını belirlemek, düzenlemek ve korumaktır. Bu amaçla, eğitim kaynaklarını bulma, düzenleme ve seçme sürecini kolaylaştırmak ve bunları tüm kullanıcılar için erişilebilir kılmak için çalışırken okuyucular, araştırma ve öğrenme için uygun bir atmosfer oluşturmayı hedefler. Bilgi altyapısının rehabilite edilmesi ve sürdürülmesi ile bilgi ve bilişim hizmetlerine hızlı erişimin sağlanmasına da büyük önem verir.

Kitapları tasnif etmek, araştırmacının aradığı kitabı bulmasını kolaylaştıracak şekilde, konularına en yakın kitapları yanlarındaki raflarda aynı konuda tek bir yerde olacak şekilde tasnif etmek, düzenlemek veya gruplara ayırmak anlamına gelir. Açık raf - açık depo sistemi (open access) sistemi ile hizmet veren kütüphaneler, insan bilgisinin sıfırlarla başlayan ve dokuzlarla biten on ana bölüme ayrıldığı Dewey ondalık

sınıflandırmasını izler (Batty, 1971). Bu amaçla kullanılan sınıflama sistemi, Tablo 1.1’de verilmiştir.

Tablo 1.1 Dewey ondalık sınıflama sistemi

Bilgi Türü	Başlangıç	Bitiş
Genel Konular	001	099
Felsefe ve Psikoloji	100	199
Din	200	299
Toplum Bilimleri	300	399
Dil ve Dil Bilim	400	499
Doğa Bilimleri ve Matematik	500	599
Teknoloji (Uygulamalı Bilimler)	600	699
Sanat (Güzel Sanatlar)	700	799
Edebiyat ve Retorik	800	899
Coğrafya ve Tarih	900	999

Dewey Onlu Sınıflama sistemindeki alanların her biri sırayla on bölüme, her bölüm on alt spesifik kategoriye ayrılır. Bu ayrımlara örnek olarak sosyal bilimler kategorisindeki numaralandırma, Tablo 1.2’de verilmiştir.

Tablo 1.2 Örnek sosyal bilimler sıralaması

No	Bölüm	Sınıflandırma
1	Toplum Bilimleri	300
2	Genel İstatistikler	310
3	Siyasal Bilimler	320
4	Ekonomi	330
5	Hukuk	340
6	Kamu Yönetimi	350
7	Toplumsal Hizmetler ve Dernekler	360
8	Eğitim	370
9	Ticaret, İletişim, Taşımacılık	380
10	Gelenekler, Görgü, Halkbilim	390

Konuyu ifade eden tasnif numarası belirtildikten sonra kitabın sırtına yazar soyadının ilk üç harfinin altına yazılır. Kitabı arayan, gerekli kitabı elde etmek için rütbe, bölüm ve şube ile başlar. Okuyucunun kullandığı kitabı masanın üzerine bırakması, okuyucunun kitabı yanlış yere koyma ihtimali nedeni ile kütüphane çalışanının kitabı yerine koyabilmesi için tercih edilir.

Dünya genelinde olduğu gibi Libya'da da hem akademik hem de halk kütüphanelerinde işlerin aksaksız yürüyebilmesi için birtakım hizmetler sunulmaktadır. Şekil 1.2'de örnek fotoğrafı verilen Libya Bingazi Üniversitesi Kütüphanesi'nde (URL-1, 2022) bu hizmetler sırasıyla:

Kayıt hizmeti: Giriş, çıkış ve görevlendirme formu doldurarak tüm bilgilerini kayıt altına almak için kart kaydı ve düzenleme konusunda uzmanlaşmış bir birim bulunmaktadır.

Görevlendirme hizmetleri: Üniversitede bir öğretim üyesi bir hafta süreyle en fazla 3 kitap ödünç alabilir ve talep olmadığı takdirde tekrar ödünç alma hakkına sahiptir. Bir üniversite öğrencisi 3 gün süre ile en fazla iki kitap ödünç alabilir ve idare öğrenciye uygun göreceği miktardan daha fazla ödünç almasına izin verebilir. Bir üniversite çalışanı 10 gün süreyle en fazla iki kitap ödünç alabilir. Süreli yayınlar, nadir kitapların özel koleksiyonları, el yazmaları, belgeler ve film, CD, teyp, slayt, harita vb. büro malzemeleri ödünç verilmez. Kütüphaneci, kamu yararının gerektirdiğini düşündüğü takdirde herhangi bir kitabı ödünç vermeme veya kütüphane içinde genel kullanımını engelleme hakkına sahiptir. Kütüphaneci, ödünç verilen herhangi bir kitabı ödünç verme süresi sona ermeden önce geri alma veya gerekli görürse herhangi bir kitabı ödünç vermekten kaçınma hakkına sahiptir. Bir veya birden fazla kişi için ihtiyaç halinde ödünç alınan kitabın yenilenmesi kullanım süresi uygun olmadığı durumda bu kitabın ilk öğrenciye ödünç verilmesinde öncelik dikkate alınır ve ilgiliye eğer ellerinde geç kalmış kitaplar varsa yeni kitap ödünç verilmez. Ödünç verme formu doldurulduktan sonra kütüphane tarafından hazırlanan karta göre yapılır.

Kütüphane kullanımına yönelik eğitim ve rehberlik hizmetleri: Yararlanıcıları kütüphaneyi kullanmaları için hazırlamak ve eğitmek için bir program mevcuttur.

Kütüphaneci, özellikle yeni gelen öğrencilerle birlikte oturup kütüphaneyi kullanma adabı konusunda onlara eğitim verip, kütüphaneyi tanıtır ve çeşitli kâğıt ve elektronik ortamdaki kütüphane materyalinden nasıl yararlanacaklarını tanıtmak için günde bir saat yönlendirme yapar. Bu eğitim programının içeriği;

- ✓ Kitaplık listesi görüntüleme,
- ✓ Kütüphane kılavuzu görüntüleme,
- ✓ Kütüphaneden randevu alınması,
- ✓ Kütüphane indeksinde sınıflandırma ve arama sistemi,
- ✓ Kütüphane ödünç verme sistemi,
- ✓ Kütüphane için abonelik kartı nasıl alınır?
- ✓ Kütüphane içinde çalışma sistemi,
- ✓ Elektronik kütüphane ve kullanımı,

şeklindedir.



Şekil 1.2 Bingazi Üniversitesi'nden bir görüntü

Libya'da mevcut kütüphane sistemi bu şekilde devam etmesine rağmen dünyanın çeşitli ülkelerinde bilgi teknolojilerinden faydalanılarak kütüphanelerin hem verdikleri hizmet geliştirilmekte hem de kullanıcılar için daha kullanışlı sistemler oluşturulmaktadır. Bu amaçla biyometrik tanıma sistemlerinin işlevselliği artırarak kullanımı daha da kolaylaştırıp Libya kütüphanecilik sistemine ciddi katkılar sağlayacağı aşikardır. Biyometrik tanıma terimi, bireyleri otomatik olarak tanımak için biyometrik tanımlayıcılar olarak adlandırılan ayırt edici fizyolojik ve davranışsal

özelliklerin kullanılmasını ifade eder. Bir dizi biyometrik teknoloji geliştirilmiştir ve bunların birkaçı farklı alanlarda çeşitli uygulamalarda kullanılmaktadır. Bunlar arasında yüz, iris, parmak izleri, konuşma ve el geometrisi biyometrik sistemlerde en sık kullanılanlardır.

Her biyometriğin güçlü ve zayıf yönleri vardır, bu nedenle belirli bir biyometri seçimi genellikle bir uygulamanın gereksinimlerine bağlıdır. Bir biyometrik sistem, o kişinin sahip olduğu belirli bir fizyolojik ve/veya davranışsal özelliğin gerçekliğini belirleyerek bir kişiyi tanımaktan sorumlu bir örüntü tanıma sistemidir (Belhadj, 2017). Pratik bir biyometrik sistem tasarlarırken en önemli konu, bu biyometrik bireyin nasıl tanındığını belirlemektir.

Parmak izlerinin sınıflandırılması genellikle Henry sınıflandırmasına dayanır (Tan vd., 2005). Henry sınıflandırma şemasının kullanımı, veri tabanındaki belirli bir parmak izi kümesinde arama yapmaya izin vermenin temel avantajını korur. Bu, parmak izi tanımlama durumları için veri tabanı ile karşılaştırma sayısını azaltarak son aşamada işlemi hızlandırır. Parmak izlerinin Henry'nin beş farklı parmak izi çeşitlerine göre sınıflandırılması, çıkıntıların kaba yapısına göre yapılır, esas olarak çekirdek ve delta noktaları, döngü ve sarmal parmak izlerinde görünür (Maltoni vd., 2009).

Yüz tanıma, bir kişinin kimliğini yüzünü kullanarak tanımlama veya doğrulama işlemidir. Kişinin yüz detaylarına göre kalıpları yakalar, analiz eder ve karşılaştırır. Yüz tanıma işlemi, görüntülerde ve videolarda insan yüzlerini tespit etmek ve bulmak için önemli bir adımdır. Yüz yakalama işlemi, kişinin yüz özelliklerine göre analog bilgileri (bir yüz) dijital bilgilere (veriler veya vektörler) dönüştürür. Yüz eşleştirme işlemi, iki yüzün aynı kişiye ait olup olmadığını doğrular.

Biyometrik kimlik doğrulama, kimliğinizi doğrulamak için fiziksel yapınızın bir bölümünü kullanmayı içerir. Bu bir parmak izi, iris taraması, retina taraması veya başka bir fiziksel özellik olabilir. Tek bir özellik veya çoklu özellikler kullanılabilir. Her şey altyapıya ve istenen güvenlik seviyesine bağlıdır. Biyometrik kimlik doğrulama ile incelenen fiziksel özellik genellikle bir kullanıcı adıyla eşleştirilir.

Makine veya bilgisayarlar için otomatik yüz tanıma problemi, resimler üzerinden veya video görüntülerinden bir veya daha fazla kişiyi yüz veritabanını kullanarak tanıma ve doğrulama problemi olarak tanımlanabilir. Bunun için ilk önce yüzü ayırt eden ve karakterize eden öznitelikler çıkartılmalı ve ondan sonra veritabanında eşleştirilmelidir.

Görüntü, çoğu parmak izi kimlik doğrulama yönteminin temeli olan çıkıntılarının akışını kesintiye uğratan süreksizlikleri göstermektedir (Marasco ve Ross, 2014). Sırt sonları, bir sırtın durduğu noktalardır. Çatallanmalar ise bir sırtın ikiye ayrıldığı noktalardır. Noktalar (çok küçük sırtlar), adalar (noktalardan biraz daha uzun sırtlar, geçici olarak farklı iki sırt arasında bir orta boşluk kaplar), göletler veya göller (geçici olarak farklı iki sırt arasındaki boş alanlar), mahmuzlar, köprüler (iki uzun bitişik sırtı birleştiren küçük sırtlar) ve geçitler (birbirini kesen iki sırt) dahil olmak üzere birçok ayrıntı türü mevcuttur. Bunlar, Şekil 1.3’de parmak izi üzerinde verilmektedir (Bhattacharya ve Mali, 2012).



Şekil 1.3 Parmak izi yapısı ve parmak izine ait lokal ve global özellikler

Bütün bu bilgiler doğrultusunda bu tez çalışmasında, yukarıda ifade edildiği üzere konvansiyonel metotlarla çalışan Libya kütüphanelerinin gelişimine katkı sağlamak amacı ile;

- ✓ Kütüphanenin dahili çalışmalarının düzenlemesine yardımcı olan bir sistem sağlamak,
- ✓ İsim, çalışma numarası, araştırmacı durumu ve kişisel fotoğraf gibi bilgiler de dahil olmak üzere kütüphane hizmetlerinden yararlanan tüm kişileri saklama yeteneğine sahip geniş kapsamlı bir veritabanı oluşturmak,
- ✓ Tüm ofis varlıklarını elektronik olarak aboneyi tanımlayarak ödünç alma sırasında muhafaza etmek,
- ✓ Kütüphaneye giriş ve çıkışlarda yönetmelik ve yasalara tam uyum sağlamak,
- ✓ Günlük, haftalık ve aylık olarak kütüphaneye gelen ziyaretçi sayısını bilmek,

gibi amaçlarla, özellik çıkarımı için gri seviye oluşum matrisini kullanarak, nöronların, katmanların ve eğitim algoritmasının en iyi mimarisini seçip sınıflandırma ve tanıma doğruluğunu artırıp halihazırda benzer özellik gösteren algoritmalarla karşılaştırması yapılarak test edilmiştir.

Bu tezin temel amacı, özellik çıkarımı için gri seviye oluşum matrisinin kullanılması, parmak izi tanıma ve yüz tanıma için yapay zeka tabanlı yüksek doğrulukla tahminleme yapılmasıdır. Optimum nöron mimarisini, katmanları ve eğitim algoritmasını seçerek, sınıflandırma ve tanıma gibi karmaşık sorunları daha fazla doğrulukla çözmeye çalışılacaktır. Yaklaşımımızı mevcut algoritmalarla karşılaştırır ve iyi bilinen veri setlerinde test edilmiştir. Yüz tanıma sistemi için ortalama ve wiener olmak üzere iki farkı filtreleme algoritması ile birlikte OpenCV ve Python ile uygulanmıştır. Görüntü ön işleme işleminden sonra Yapay Sinir Ağları (YSA) ve Evrimsel Sinir Ağları (ESA) uygulanmıştır.

Tez çalışmasına konu olan bu uygulama, Libya'daki üniversite kütüphaneleri için eksik olan yapay zekâ temelli akıllı kütüphane bilgi sistem kurulmasında çok büyük destek olacaktır. Akıllı kütüphane sistemi kavramı, minimum insan müdahalesi ile bir kütüphanede çalışmaktır. Deneysel sonuçlara göre yüz ve parmak sistemlerinin kütüphane bilgi sistemlerinde uygulanabileceğini ve hem çevrim içi veri setlerin de hem de gerçek zamanlı veri setlerindeki görüntüleri için yapay sinir ağları ve evrimsel sinir ağlarının kullanıldığı diğer çalışmalara göre oldukça yüksek bir doğruluk değeri gözlemlenmiştir.

Tezin ilk bölümünde tez çalışmasında sunulan problem tanımı ve çözümleri ifade edilmiştir. İkinci bölümünde yüz ve parmak tanıma sistemleri hakkında literatürdeki çalışmalar ile ilgili bilgi sunulmuştur.

Üçüncü, dördüncü ve beşinci bölümlerinde, yapay sinir ağları, evrimsel sinir ağları, parmak izi ve yüz tanıma için önerilen yöntemler anlatılmıştır.

Tezin 6. Bölümünde, kütüphane bilgi sistemi için parmak ve yüz tanıma sisteminin uygulaması sunulmuştur. Grafik kullanıcı arayüzleri ve Python kodları ile desteklenmiştir.

Tezin son bölümünde ise tez çalışmasını ileriye götürebilecek öneriler verilmiştir.

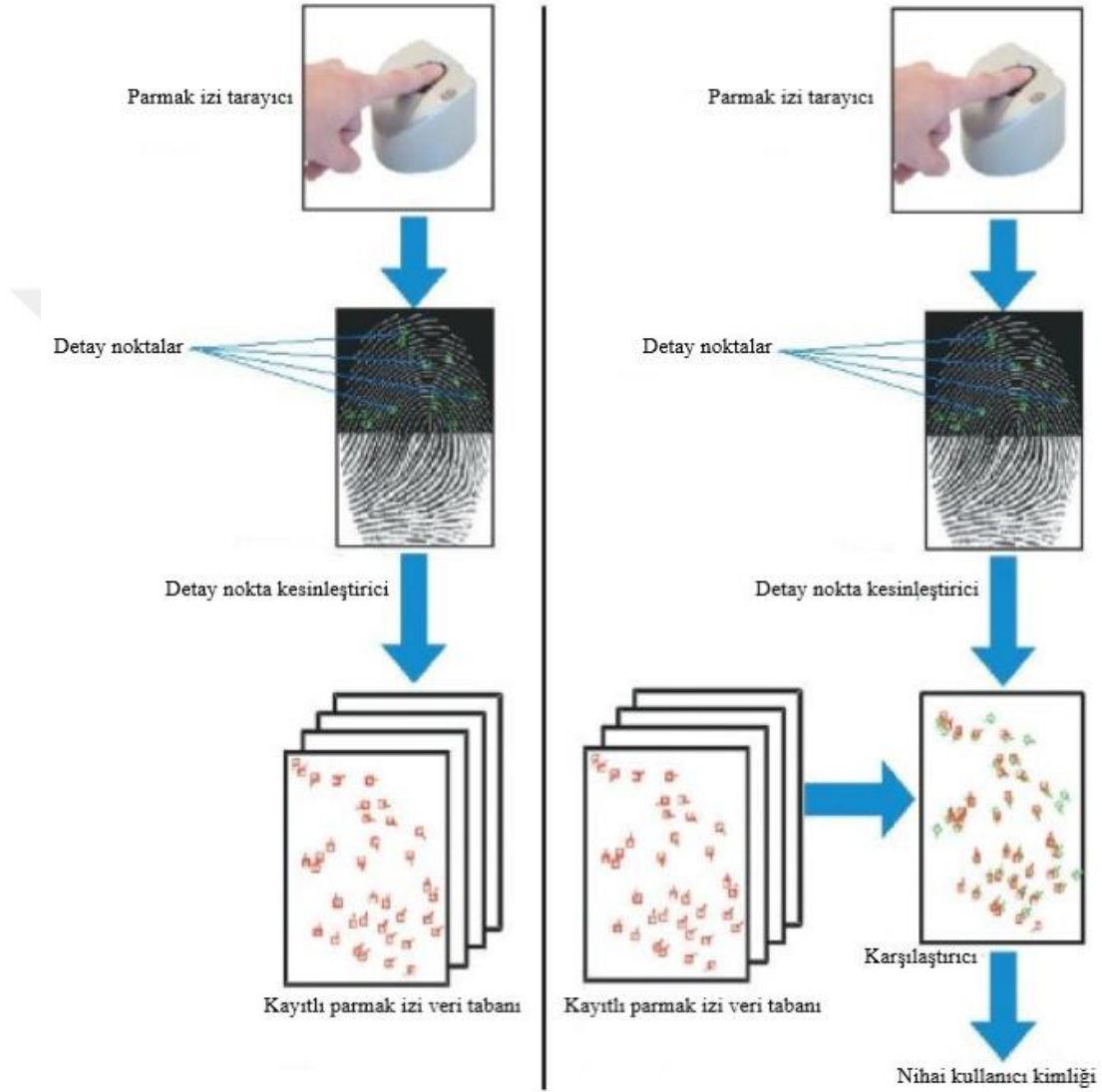
2. LİTERATÜR TARAMASI

Birçok araştırmacı parmak izi sınıflandırma problemini ele almış ve literatürde otomatik parmak izi sınıflandırmasına yönelik birçok yaklaşım sunulmuştur; yine de bu konudaki araştırma hala çok aktiftir. Son yirmi yılda otomatik parmak izi tanıma sistemlerinin tasarımında önemli ilerlemeler kaydedilmiş olmasına rağmen, bir dizi tasarım faktörü (güvenilir ayrıntılar çıkarma algoritmalarının eksikliği, parmak izi görüntüleri arasında güvenilir bir eşleşmeyi nicel olarak tanımlamanın zorluğu, zayıf görüntü elde etme, düşük kontrastlı görüntüler, işçiler için parmak izini okuma zorluğu vb.) istenen performansın elde edilmesinde darboğaz oluşturur. Günümüzde, parmak izi kalitesinin tanıma performansları üzerindeki etkisinin araştırılması da giderek daha fazla ilgi görmektedir.

Parmak izi tanımada, parmak izi eşleştirmenin temeli olarak, parmak izi görüntüsünün ön işlenmesi ve parmak izi görüntüsünün öznelik çıkarımı, tüm parmak izi tanıma sürecinde önemli bir rol oynar. Parmak izi görüntüsü segmentasyonu, parmak izi görüntüsü geliştirme, parmak izi görüntüsü ikilileştirme, parmak izi görüntüsü iyileştirme vb. ikili parmak izi görüntüsü, gri parmak izi görüntüsünü yalnızca 0 (çıkıntı alt grafiğini temsil eder) ve 255 (arka plan alt grafiği) ile temsil edilen ikili görüntüye dönüştürmektir.

Tüm parmak izi özelliğine dayalı sınıflandırma, farklı parmak izlerini bir dereceye kadar ayırt etmek için kullanılabilir. Ancak bu bilgiler, tüm farklı parmak izlerini ayırt etmek için yetersizdir. Bu şekilde, tipik olarak sınıflandırma ve erişimde kullanılırlar. Parmak izinin yerel (local) özellikleri, parmak izindeki düğümleri ifade eder (Mali ve Bhattacharya, 2011). İki parmak izi tipik olarak aynı genel özelliklere sahiptir, ancak yerel özellikleri, düğümleri aynı olamaz. Parmak izi desenleri genellikle süresiz ve düzdür. Bunun yerine sık sık kesintiler, çatallanmalar veya indirimler oluyor. Bu kırılma noktaları, çatallanma noktaları ve dönüm noktaları “düğüm (node)” olarak adlandırılır (Zhao vd., 2021).

Parmak izini tanıma sürecinde iki önemli adım vardır. Birincisi, kaydolma sürecidir ve ikincisi ise başvurusuna göre birini doğrulama veya tanımlama sürecidir. Parmak izini tanımaya yönelik temel algoritmanın taslağı Şekil 2.1’de verilmiştir (Jain vd., 2010).



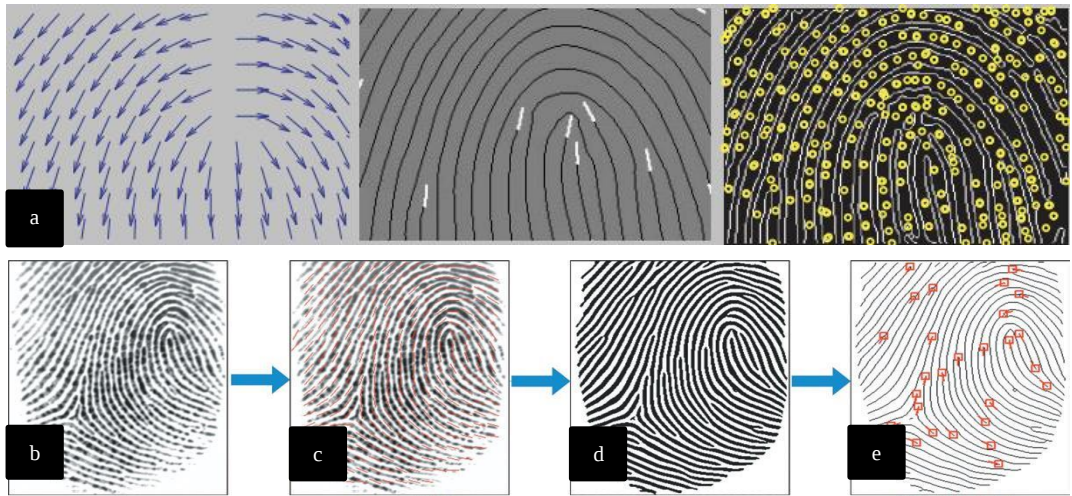
Şekil 2.1 Kayıt ve doğrulama içeren parmak izi akış şeması

Çoğu zaman, bir parmak izi görüntüsü için üç “özellik çıkarma” düzeyi vardır. Özellik çıkarmanın ilk seviyesi, bir görüntüden tek noktalar, sırt akışı ve desen tipi gibi makro ayrıntıları elde etmek için kullanılır. Özellik çıkarmanın ikinci seviyesi, çıkıntılarının nerede başladığı ve bittiği gibi küçük ayrıntılara dayanır. Özellik çıkarmanın üçüncü seviyesi, sırtların şekli, yolu, kenar tasarımı ve yara izleri gibi ayrıntılarıyla ilgili olarak ele alınabilir.

Parmak izlerinde sarmal, kemer veya halka gibi birkaç farklı desen vardır. Seviye 1 özellikleri kalıpları birbirinden ayırmak için kullanılır. Ayrıca ikinci ve üçüncü seviyeler, bir kişinin parmak izini neyin benzersiz kıldığını açıklar. Ve üçüncü seviye çıkarma için, parmak izinin yüksek çözünürlüğü, yani en az 500 inç başına piksel (PPI) olması gerekir.

Şekil 2.2 (a), bir parmak izi görüntüsünden ayrıntıların nasıl alınacağını gösteren kısa bir akış şemasıdır. Algoritmanın ilk adımında, görüntünün sürtünme sırtlarının sayısı ve yönü tahmin edilmektedir. Bundan sonra görüntü, görüntünün kalitesini ve sırt tasarımını iyileştiren “filtreleme” adı verilen bir işlem adımı uygulanır. Algoritmanın bir sonraki adımında, iyileştirilen görüntüden ikili (binary) görüntü oluşturulur. Bu görüntü daha sonra sırtların nerede bittiğini ve nerede ayrıldıklarını bulmak için kullanılır.

Parmak izi görüntüsünden tüm küçük detayların alınması kadar, parmak izi tanıma sisteminin detayları doğru bir şekilde eşleştirmesi de önemlidir. Eşleştirme sürecinde, eşleştirmeyi yapan modül, iki parmak izinin birbiriyle ne kadar iyi eşleştiğini hesaplar. Modül aynı parmak izlerine sahipse eşleştirme puanı yüksek olmalıdır. Değilse, her parmak izinin okunabilmesi için düşük olmalıdır. Eşleştirme sürecini izlemenin dört yolu vardır: ayrıntılara göre eşleştirme, aşamaya göre eşleştirme, iskelete göre eşleştirme ve görüntülerin korelasyonuna göre eşleştirme (Jain vd. 2010).



Şekil 2.2 Kayıt ve doğrulama içeren parmak izi tanıma algoritması (a) parmak izini oluşturan farklı seviyelerde çıkarılan özellikler, (a) gri tonlamalı görüntü, (b) oryantasyon alanı, (c) ikili görüntü, (d) detay noktalar

Nagaty (2001) çalışmasında parmak izi görüntülerinde doğrusal olmayan ayrımları yapabilmek, hatalı ve bozuk örüntüleri sınıflandırmak için Yapay Sinir Ağları (YSA) kullanmıştır. Sonuç olarak hem orijinal bağımsız değişkenlerin hem de iyi bilinen bir istatistiksel aracın tahminlerini birleştirmenin YSA sınıflandırma performansını artırdığı kanıtlanmıştır.

Shah ve Sastry (2004) çalışmasında bir parmak izi sınıflandırma algoritması sunulmuştur. Bu algoritma, parmak izi görüntüsünü kemer, sol döngü, sağ döngü, worl ve tented kemer olmak üzere beş sınıftan birine sınıflandırır. Yeni bir yönelimli çizgi dedektörünün çıktısından elde edilen yeni bir düşük boyutlu özellik vektörü kullanılmıştır. Ayrıca yaygın olarak kullanılan sınıflandırma yöntemlerinden destek vektör makineleri (DVM), k-en yakın komşu (k-NN) algoritmaları ve YSA tercih edilmiştir.

Ulusal Standartlar ve Teknoloji Enstitüsü (NIST) tarafından yapılan testlere göre, Nisan 2020 itibarıyla en iyi yüz tanıma algoritmasının hata oranı 2014 yılında %4,1'e kıyasla yalnızca %0,08'dir (URL-2, 2010).

Prabhakar (2001) çalışmasında bir parmak izinde bulunan hem genel hem de yerel bilgileri birleştiren, parmak izleri için önemli olan yeni bir temsil önerilmiştir. Önerilen temsil, parmak izi görüntülerinin yönlendirilmiş dokular olarak ele alınmasına dayanmaktadır, adli tıp uzmanları tarafından kullanılan temsillerden çok (eşleştirme hızı ve depolama boyutu açısından) farklıdır ve otomatik sistemlere daha uygundur. Bu temsilin performansı hem parmak izi sınıflandırması hem de büyük veritabanlarındaki eşleştirme uygulamaları için değerlendirilmiştir. Sonuç olarak tüm tekniklerin kullanılmasının, büyük bir veri tabanında parmak izi doğrulama sisteminin performansını önemli ölçüde iyileştirdiğini göstermişlerdir.

Parmak izi sınıflandırması, bir parmak izi izlenimini birkaç ana parmak izi türünden biri olarak etiketlemekten oluşur: Kemer, sol döngü, sağ döngü, sarmal, vb. Parmak izi eşleştirmede sorunlar, aynı parmak tarafından iki gösterimin üretilip üretilmediğine karar vermekle aynı anlamdadır. Yao vd. (2003) çalışmasında DVM ve özyinelemeli sinir ağları yaklaşımları kullanılarak yeni parmak izi sınıflandırma algoritmaları

sunulmuştur. Bu yaklaşımın iki ana avantajı vardır: (a) Eğitim setinde belirsiz parmak izi görüntülerinin varlığını tolere edebilir ve (b) test setindeki en zor parmak izi görüntülerini etkili bir şekilde tanımlayabilir. Deneysel sonuçlar, NIST 4 parmak izi veri seti üzerinde denetimli öğrenme yöntemlerinden DVM yöntemi kullanılarak %95,6 doğruluk değeri elde edilmiştir.

Jain vd. (2001) çalışmasında inceltirilmiş sırtların ve önemsiz noktaların önemsiz olmayan hesaplamalarından kaçınarak, yalnızca akış alanını kullanan, parmak izi sınıflandırması için model tabanlı bir yöntem önerilmiştir. Her sınıf için, o sınıftaki parmak izlerinin şeklini modelleyen bir parmak izi çekirdeği tanımlanmıştır. Daha sonra verilen parmak izinin akış alanına en uygun çekirdeğin bulunmasıyla sınıflandırma yapılır. NIST 4 veri setinde %91,25'lik bir sınıflandırma doğruluğu elde edilmiştir. Ayrıca çekirdek yerleştirme prosedürünün parmak izi hizalaması için nasıl kullanılabileceğini de gösterilmiştir.

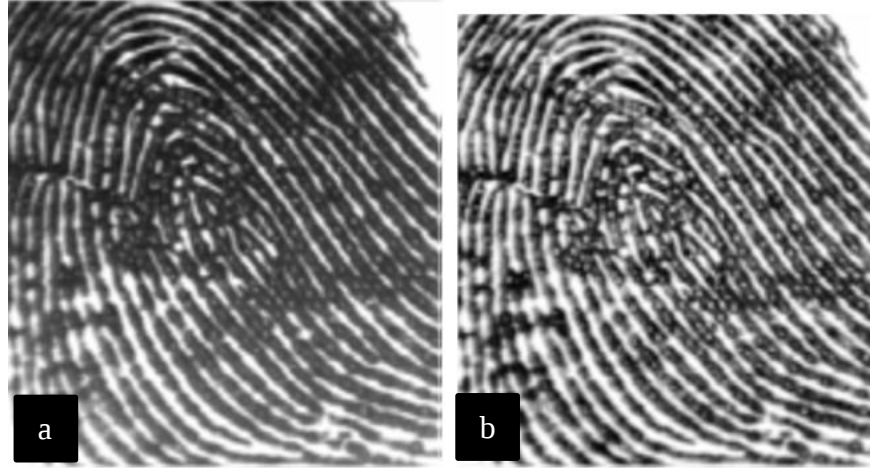
Senior (2001) çalışmasında tekil noktaları tespit etmeye gerek kalmadan baskının sırt yapısını tanımak için Markov zincirindeki bir varyasyon olan bir çeşit istatistiksel model olan gizli Markov modellerini ve karar ağaçlarını kullanan sınıflandırma yöntemleri kullanılmıştır. Bu yöntemler, standart bir parmak izi sınıflandırma algoritması ile karşılaştırılır ve birleştirilir ve kombinasyon için sonuçlar, standart bir parmak izi görüntüleri veritabanı kullanılarak sunulmuştur. Kombinasyon sınıflandırıcısının doğruluğunun, aynı koşullar altında test edilen son teknoloji ürünü iki sisteminkinden daha yüksek olduğu gösterilmiştir.

Jain ve Minut (2002) çalışmasında parmak izi olukları yapısının sınıflandırma şeması, yolla ilgili bilgileri depolar ve sınıflandırmada %94,8 doğruluk değeri elde edilmiştir. Parmakizi yakalamaya alternatif Chang ve Fan (2002) yapısal bilgi sunumu geliştirmiştir. Destek vektör makinesi sınıflandırıcılarının farklı kombinasyonlarını kullanan NIST 4 parmak izi veritabanı üzerinde bazı deneyler yapılmıştır (Yao vd., 2001). Sonuç olarak DVM sınıflandırıcıya ait doğruluk değeri %89,3 elde edilir. Minaee vd. (2019) çalışmasında özellik temsiliyi birlikte öğrenebilen ve tanıma gerçekleştirebilen Evrişimli Sinir Ağlarını (Convolutional Neural Network, CNN) kullanarak parmak izi tanıma için uçtan uca bir derin öğrenme çerçevesi önermiştir.

Önerilen model, iyi bilinen bir parmak izi veri kümesinde çok yüksek bir tanıma doğruluğu sağlayabilir. Bu çerçevenin biyometri tanıma görevleri için yaygın olarak kullanılabileceğine ve daha ölçeklenebilir ve doğru sistemleri mümkün kılmıştır. Ayrıca, çoğunlukla tanıma sonuçlarını etkileyen bir giriş parmak izi görüntüsündeki önemli alanları vurgulamak için bir görselleştirme tekniği kullanılmıştır.

YSA parametrelerinin yapısal olarak budanması, çıkarım sırasında hesaplama, enerji ve bellek aktarım maliyetlerini azaltır. Molchanov vd. (2019)'da bir nöronun (filtre) son kayıba katkısını tahmin eden ve daha küçük puanları yinelemeli olarak kaldıran yeni bir yöntem önerilmiştir. Bir filtrenin katkısına yaklaşmak için birinci ve ikinci dereceden Taylor açılımlarını kullanarak yöntemin iki varyasyonunu açıklanmıştır. ImageNet üzerinde eğitilen modern ağlar için, yöntemlerimiz tarafından hesaplanan katkı ile gerçek öneme ilişkin güvenilir bir tahmin arasında deneysel olarak yüksek ($>93\%$) bir korelasyon ölçülmüştür. Önerilen yöntemlerle budama, doğruluk, saniye başına kayan nokta işlemi (FLOPs) ve parametre azaltma açısından son teknolojiye göre bir gelişme sağlamıştır.

Orijinal parmak izi görüntüsünde birçok gürültü olduğundan, parmak izi görüntüsündeki gürültünün etkisini azaltmak ve parmak izi modellerinin sırt yapısını vurgulamak için genellikle histogram eşitleme gibi bir görüntü geliştirme algoritması uygulanır. Histogram eşitleme, görüntülerin doğrusal olmayan bir şekilde genişletilmesi ve belirli bir gri aralıktaki piksel değerlerinin sayısı kabaca aynı olacak şekilde görüntü piksel değerlerinin yeniden dağıtılmasıdır (Yelmanova ve Romanyshyn, 2017). Şekil 2.3, bir parmak izi görüntü histogramı eşitlemesinin uygulanmasının sonuçlarını göstermektedir (Yazdi ve Gheysari, 2008).



Şekil 2.3 Histogram eşitleme sonucu a) orijinal görüntü b) senkronizasyon sonrası görüntünün histogramı

Sinir ağı tabanlı bir yüz algılama sisteminde geriye dönük olarak bağlı bir sinir ağı, bir görüntünün matrisler kullanılarak küçük çerçevelerini inceler ve her pencerenin bir yüz içerip içermediğine sonucuna varılır. Sistem, tek bir ağ üzerinden performansı artırmak için birkaç ağ arasında arabuluculuk yapar. Bu işlem, yüz dışı görüntülerin tüm alanını kapsayacak şekilde seçilmesi gereken yüz dışı eğitim örneklerini manuel (elle) olarak seçmenin zor görevini ortadan kaldırır (Kumar vd., 2019).

Yüz tanıma için sağlam bir otomatik sistem, güvenli sistemlerde suçlu tanımlama ve kimlik doğrulama gibi birçok uygulama bulacaktır. Bu makale, rastgele bir görüntüden bir yüzü sağlam bir şekilde algılayabilen yapay sinir ağlarına dayalı bir sistemi açıklamaktadır. Bu sistem, iki aşamadan oluşmaktadır. Birinci aşama, girişlerin orijinal bir görüntüden yoğunluk değerlerine göre filtrelendiği iki paralel alt ağı sahip olmaktır. İkinci aşama ağın girdileri, alt ağların çıktılarından ve çıkarılan öznitelik değerlerinden oluşur. İkinci aşamadaki bir çıktı, giriş bölgesinde bir yüzün varlığını gösterir. İlk gizli katmanda 256 birim, ikinci gizli katmanda sekiz birim ve iki çıkış birimi olmak üzere 1024 giriş birimli dört katmandan oluşan yüz tanıma için en erken sinir ağı geliştirilmiştir (Propp ve Samal, 1992). Sistem, yüz görüntülerinin %91'ini ve yüz olmayan görüntülerin %88'ini doğru bir şekilde sınıflandırma yapılmıştır.

Feraud vd. (2001)'de otomatik çağrışımlı sinir ağlarını kullanan bir algılama yöntemi sunulmaktadır. Bu yöntem, beş katmanlı otomatik birleştirici bir ağın doğrusal olmayan bir ana bileşen analizi gerçekleştirebildiğini gösteren temele dayanmaktadır

(Kramer, 1991). Bir otomatik ilişkilendirmeli ağ, önden görünüşlü yüzleri algılamak için kullanılır ve bir diğeri, önden görünümün soluna ve sağına 60 dereceye kadar döndürülmüş yüzleri algılamak için kullanılır. Sonra Olasılığa Dayalı Karar Tabanlı Sinir Ağı (PDBNN) kullanan bir yüz algılama sistemi sunulmuştur (Lin vd., 1997). PDBNN'nin mimarisi, değiştirilmiş öğrenme kuralları ve olasılıksal yorumlama ile bir radyal temel işlev (RBF) ağına benzetilmektedir. Yüz algılama için çok aşamalı bir model, Viola ve Jones algoritması, Gabor filtreleri, Temel Bileşen Analizi (TBA) ve YSA dayalı olarak entegre edilmiştir.

Da'San vd. (2015)'de Viola ve Jones algoritması, Gabor Filtreleri, Temel bileşen Analizi (PCA) ve YSA temelinde yüz algılama için çok aşamalı bir model önerilmiştir. Farfade vd. (2015), Derin Yoğun Yüz Dedektörü (DDFD) olarak adlandırılan, derin öğrenmeye dayalı bir yüz algılama yöntemi önermiştir. Yöntem, poz/yer işareti açıklaması gerektirmez ve tek bir model kullanarak çok çeşitli yönlerdeki yüzleri algılayabilir. Ek olarak DDFD, sınırlayıcı kutu regresyonu, DVM veya görüntü bölümlenme gibi son zamanlardaki derin öğrenme nesne algılama yöntemlerindeki yaygın modüllerden bağımsızdır. Önerilen yöntem, Hızlı Bölge Tabanlı Evrişimli Ağlar (R-CNN) ve özellikle çoklu görüntülü yüz algılama için geliştirilmiş kademeli tabanlı ve Deforme Olabilen Parça Tabanlı Bir Model (DPM) tabanlı diğer yüz algılama yöntemleriyle karşılaştırmıştır.

Liao vd. (2016)'de keyfi poz varyasyonları ve kapatmalar gibi kısıtlamasız yüz algılamadaki zorlukları ele almak için bir yöntem önerdi. Normalleştirilmiş Piksel Farkı (NPD) adı verilen yeni bir görüntü özelliği önerilmiştir. NPD özelliği, deneysel psikolojideki Weber Kesirinden esinlenerek iki piksel değeri arasındaki bazı oranlardaki fark olarak hesaplanır. Yeni özellik, ölçek değişmez, sınırlıdır ve orijinal görüntüyü yeniden oluşturabilir. Ayrıca, NPD özelliklerinin ve bunların kombinasyonlarının optimal alt kümesini öğrenmek için derin bir ikinci dereceden ağaç önerdiler, böylece karmaşık yüz manifoldları öğrenilen kurallarla bölümlenebilir.

3. YAPAY SİNİR AĞLARI

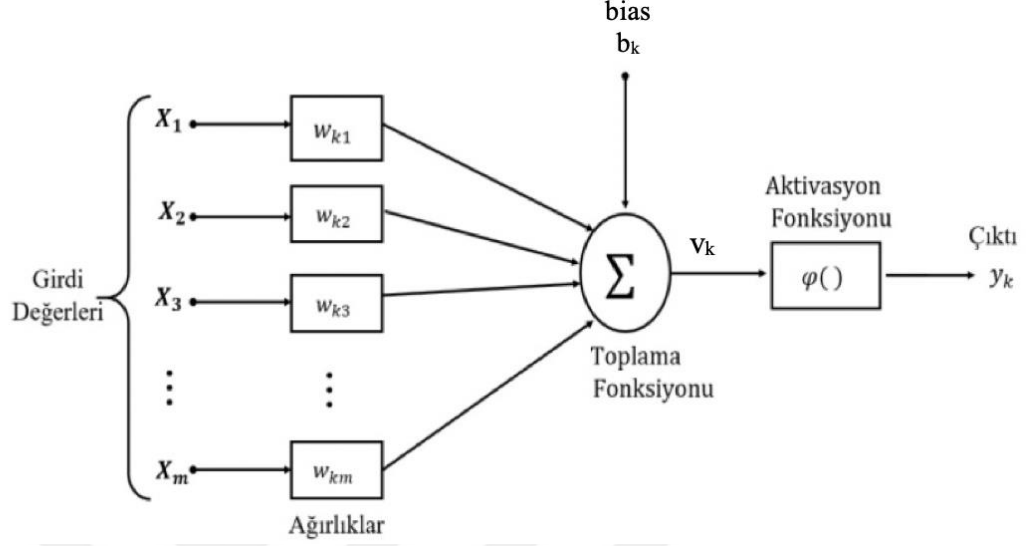
Yapay sinir ağları (YSA), biyolojik nöron ağının (örneğin insan sinir sistemi) simülasyonu için matematiksel bir model tanımlar. YSA, insan beyninin deneyime dayalı olarak öğrenip uyum sağlayabildiği fikrinden ilham alınarak, bu öğrenme ve uyum sürecini taklit edecek şekilde tasarlanmıştır. İnsan beyninde milyonlarca nöron bulunduğundan, yapay sinir ağı düğüm adı verilen birçok nörondan yapılmıştır (Dastres ve Soori, 2021).

Beyindeki temel hesaplama birimi nörondaki dendritler ve akson gibi yapay nöronun da girdi ve çıktısı vardır. Sinaps olduğu gibi yapay nöronun girdilerle özetlenmesi gereken ağırlıkları vardır. Nöron, bir katmanın çıktısı bir sonraki katmana yani diğer katmanlara girdi görevi görecektir şekilde bir katman veya vektör halinde düzenlenir (Walczak ve Cerpa, 2003).

Genelde YSA modelinin en küçük parçası olan perceptron, Eşik Mantık Birimi (Threshold Logic Unit, TLU) olarak da bilinir (Du vd., 2022). Birden fazla girişi ve tek bir çıkışı olan nöron olarak da bilinen tek bir birimden oluşan en basit YSA mimarilerinden biridir. YSA bir giriş katmanı, bir veya daha fazla gizli katman ve bir çıkış katmanı içeren anot katmanlarından oluşur. Her düğüm veya yapay nöron diğerine bağlanır ve ilişkili bir ağırlık ve eşik sahiptir. Herhangi bir düğümün çıktısı belirtilen eşik değerinin üzerindeyse, bu düğüm etkinleştirilir ve ağın bir sonraki katmanına veri gönderilir. Aksi takdirde, ağın bir sonraki katmanına hiçbir veri iletilmez.

Şekil 3.1’da ana bileşenleri içeren genel bir yapay sinir ağı modeli sunulmuştur (Molchanoc vd., 2019). Sinir ağı giriş, çıkış ve birçok gizli katmana sahip çok katmanlı bir algılayıcı geliştirilmiştir. Bir yapay sinir ağında, nöronlar önceki katmanlarından girdiler alır, bu sinyali bir aktivasyon fonksiyonu kullanarak dönüştürür ve bu dönüştürülmüş sinyali ağırlıklar boyunca bir sonraki katmandaki nöronlara gönderir. Ağın daha fazla öğrenmesini sağlamak için konvolüsyon işlemi ile özelliklerin sayısı

artar, böylece konvolüsyonel sinir ağı oluşur. Son yıllarda eğilim normalde derin öğrenme olarak adlandırılan derin evrişimli sinir ağıdır.



Şekil 3.1 YSA’da kullanılan yapay nöronların yapısı

$x_1, x_2, \dots, x_m$ nöronun dış duyu sisteminden veya bağlantı kurduğu diğer nöronlardan aldığı giriş bilgisini veya giriş sinyallerini temsil eder. $w = w_{k1}, w_{k2}, \dots, w_{km}$ biyolojik nöronlar arasındaki sinapsı taklit eden alınan bilgiyi değiştiren sinaptik ağırlıkların vektörüdür. Bunlar, nörona doğru yaymak istedikleri değerleri azaltabilen veya artırabilen kazançlar olarak yorumlanabilir. Toplam işlevi (Σ), doğrusal bir birleştirici oluşturan, giriş sinyallerinin kendi ağırlıkları ile çarpılan toplamını temsil etmektedir. Aktivasyon fonksiyonu (φ), $[0;1]$ veya $[-1,1]$ arasında normalleştirilmiş bir aralıkta nöronun çıkış genliğini kısıtlar. y_k , nöron tarafından üretilen sonuç olan çıkış sinyalidir (Mohammed, 2019).

Her girdi bilgisinin, o nöron için önem derecesini matematiksel olarak temsil eden sinaptik ağırlık olarak da bilinen ilişkili bir ağırlığı vardır. Nöronların giriş sinyalleri, sinaptik ağırlıkları ile çarpılır ve önyargıya eklenen bu sonucun toplamı, nöronun giriş bilgilerini oluşturur (Gültepe ve Duru, 2018; López vd., 2022). Denklem 3.1 ile matematiksel gösterimi verilmiştir.

$$v_k = \sum_{j=1}^m W_{kj} X_j + b_k \quad (3.1)$$

Burada x_j giriş sinyalleridir; w_{kj} nöronun sinaptik ağırlığıdır k ; b_k bias (önyargı)'dır; ve v_k modelin doğrusal birleştirici çıktısıdır. Bias, nöronun dışındaki bir parametredir; ve işlevi, girdi verileri boş olduğunda hataların ortaya çıkmasını önlemektir. Ek olarak, aktivasyon fonksiyonunun net girişini pozitif veya negatif olarak artırır veya azaltır.

Doğrusal birleştiricinin (v_k) çıkış değeri, nöron aktivasyonu için sınır değerle karşılaştırılacaktır. Değer daha yüksekse, nöron aktive olur, aksi takdirde inaktif kalır. Nöronun iç düzeninin aktivasyon fonksiyonu artırıyor, sürekli ve nöronun çıkış genliğine bir sınırlama elemanı olarak çalışıyor. Aktivasyon fonksiyonu tarafından tanımlanan indüklenen lokal alan (veya aktivasyon potansiyeli) açısından bir nöronun çıktısı, $[0; 1]$ veya $[-1.1]$ aralığında normalleştirilmiş bir genliğe sahiptir.

Her nöronun, bitişik katmanların nöronlarına bağlantılar tarafından iletilecek sinyalin yoğunluğundan sorumlu olan, onunla ilişkili bir aktivasyon işlevi vardır (Wan vd., 2022). Aktivasyon fonksiyonları arasında doğrusal ve sigmoid aktivasyon fonksiyonları çok sık kullanılır. Doğrusal etkinleştirme işlevi, ağ çıkışını sınırlamayan ve yalnızca veri giriş ve çıkışını depolamak için kullanılan işlevdir. Doğrusal fonksiyon, Denklem 3.2'de gösterildiği gibi şekilde sunulur:

$$\varphi = k \cdot v \quad (3.2)$$

Burada k bir sabittir. Sigmoid aktivasyon fonksiyonu en popüler olanıdır ve 0 ile 1 arasında sürekli bir değer aralığı varsayar. Ayrıca doğrusal ve doğrusal olmayan davranış arasında yeterli bir dengeye sahiptir.

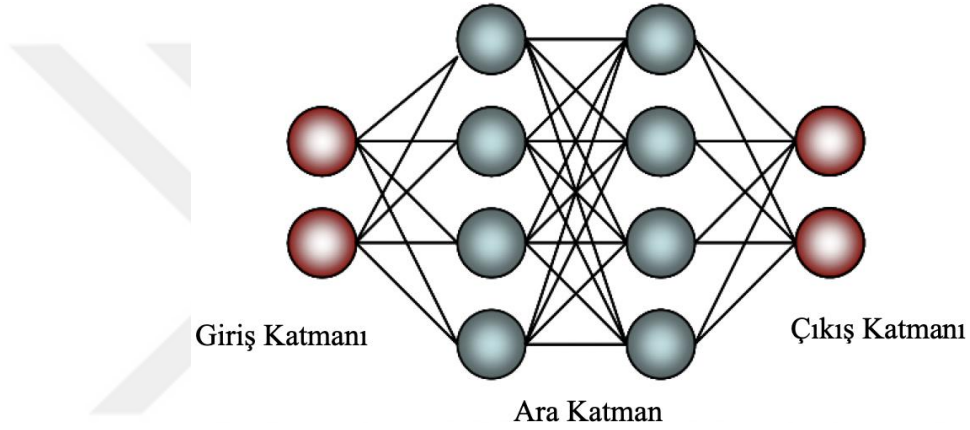
Sigmoid aktivasyon fonksiyonları, Çok Katmanlı Algılayıcı (ÇKA)'da kullanılır. Denklem 3.3 ile tanımlanan lojistik fonksiyon ve Denklem 3.4 ile temsil edilen hiperbolik teğet fonksiyon da sigmoid fonksiyonların örnekleridir. Aralarındaki fark, lojistik fonksiyonun 0 ile 1 arasındaki değerleri benimsemesi, hiperbolik teğet fonksiyonun ise -1 ile +1 arasındaki değerleri benimsemesidir.

$$\varphi(v) = \frac{1}{1+\exp(-a \cdot v)} \quad (3.3)$$

$$\varphi(v) = \tanh(v) \quad (3.4)$$

burada v_k , modelin doğrusal birleştirici çıktısıdır ve a , fonksiyonun eğim parametresidir.

Farklı YSA modelleri arasında Çok Katmanlı Algılayıcı Modeli (ÇKAM) özellikle popülerdir. ÇKAM’de bir giriş katmanı (nöronları harici stimülasyon alır), bir veya daha fazla ara katman ve çıkış katmanı (ağ sonucunu sağlayan) vardır. Böyle bir YSA’nın yapısı Şekil 3.2’de belirtilmiştir.



Şekil 3.2 Çok katmanlı algılayıcı model ağ örneği

Bir ÇKAM ağındaki iç katmanların olmaması, uygulamasını doğrusal olarak ayrılabilir sorunların tedavisine sınırlar. Sadece en az bir ara katman ekleyerek doğrusal olmayan problemleri tedavi etmek mümkün hale gelir. ÇKAM modeline özgü iç katmanlar, ağ giriş verileri kümesi tarafından açıklanan sorunu çıkış katmanına uygun bir gösterime dönüştürür (Veronez vd., 2011).

4. EVRİŞİMSEL SİNİR AĞLARI

Son yıllarda çeşitli sektörlerde derin öğrenme teknolojisi kullanılmaktadır. Derin öğrenme, bilgi öğrenme, problem çözme ve karar verme gibi insan benzeri yetenekler geliştirmiştir (Khan vd., 2022). Büyük şirketler nesnelerin interneti, büyük veri, yapay zekâ ve blockchain dahil olmak üzere en son dijital teknolojileri benimsemeye çalıştılar (Ancin vd., 2022). Derin öğrenme teknolojisi, makine öğrenimi ve yapay zekanın bir gelişimidir.

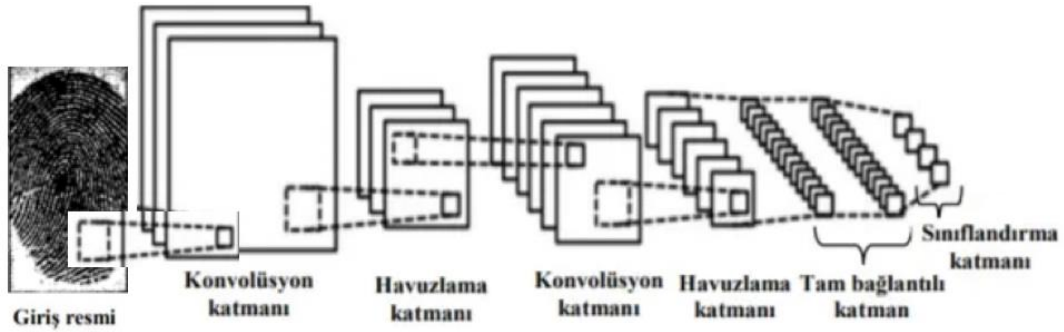
Genel olarak, makine öğrenimi ve derin öğrenme, insanlar tarafından tekrarlı programlama yapılmadan kendi kendine eğitim yapabilir. Derin öğrenme, verilerin sonucunu tahmin etmek için veri kümesi adı verilen ilk veri toplanmasını gerektirir. Derin öğrenme, eğitim ve test verilerine dayalı çıktı verileri üretecektir. Öğrenme değerlendirmesini geçtikten sonra derin öğrenme verileri tahmin edebilir. Derin öğrenme, çeşitli senaryolarda büyük veriler kullanılarak örüntü tanıma veya veri tahmini için kullanılabilir (Zhu vd., 2022). Öğrenme sistemi için kullanılan bazı yöntemler denetimli ve denetimsiz öğrenmedir. Denetimli bir algoritma, girdi ve çıktı verileri arasındaki ilişkiyi tanımlamak için etiketler kullanılır ve elde edilen çıktı ile eşleşen girdiye göre tahmin etmek için öngörücü bir model oluşturur. Buna karşılık, denetimsiz bir algoritma, etiketlenmemiş verileri kullanan bir öğrenme sistemi kullanır. Algoritma, eğitim verilerini, öncelikle boyut küçültme ve gruplama sistemlerine dayalı olarak ayırt edici özelliklerine göre sınıflandırabilir (Zhang vd., 2019).

Derin öğrenme (Fu vd., 2022), ham veriden yüksek ve soyut seviyelere kadar çeşitli temsili öğrenme seviyeleri aracılığıyla ham verilerin otomatik özellik çıkarımına izin veren geleneksel makine öğrenimi sistemlerinden farklıdır. Derin öğrenme, gizli derin ağ katmanlarında sürekli olarak biriken büyük miktarda verinin karmaşık doğrusal olmayan işlevlerini temsil etmenin üstel avantajı ile birlikte, önemli kalıpları güçlendirerek ve girdi verilerindeki alakasız varyasyonları bastırarak öğrenme kapasitesini artırabilir (Shen, 2018).

Derin öğrenmede kullanılan çeşitli teknikler temel olarak evrişimli sinir ağları, tekrarlayan sinir ağları ve derin sinir ağları gibi yeni mimariler temelindedir (Mishra vd., 2021; Qin vd., 2018). Derin öğrenme teknolojisi, yapay sinir ağlarını, özellikle de evrişimsel yöntemi kullanır.

En yaygın kullanılan derin öğrenme algoritmalarından biri konvolüsyonel sinir ağıdır (CNN). CNN, araştırmacılar tarafından dijital görüntülerde nesne tanımlama veya nesne tanıma alanında yaygın olarak kullanılan algoritmalarından biridir (Permanasari vd., 2022). CNN ilk olarak 1960’larda tanıtıldı (Wu vd., 2022) ve bilgisayar görüşünde umut verici performans sonuçları göstermiştir. Bu nedenle CNN, derin öğrenmede en temsili sinir ağı haline gelmiştir (Li vd., 2021; Zarbega ve Gültepe, 2021). CNN, yüksek hesaplamalı karmaşık görsel görevleri çözmek için kullanılmıştır ve esas olarak görüntü sınıflandırmasında, segmentasyon (Zarbega vd Gültepe, 2020), nesne algılama, video işleme, doğal dil işleme ve konuşma tanıma kullanılır.

CNN’nin dört katmanı vardır: Konvolüsyon (evrişimsel) katmanı, havuz katmanı, tam bağlantılı katman ve sınıflandırma katmanı (Bezdan ve Džakula, 2019; Gültepe ve Duru, 2019; Zarbega ve Gültepe, 2021).



Şekil 4.1 CNN mimarisi

Evrişimli katman, temel özellikleri çıkararak girdi görüntülerinin evrişimini hesaplamak için bir çekirdek filtresi kullanır. Filtre çekirdeği aynı boyuta sahiptir, ancak giriş görüntüsünden daha küçük bir sabit parametre değerine sahiptir.

Havuzlama katmanı, ardışık iki konvolüsyon katmanını birleştirecektir. Aşağı (down) örnekleme (sampling) gösterimleri yaparak parametre sayısını ve hesaplama yüklerini

azaltır. Havuzlama katmanındaki işlev, maksimize edilmiş veya ortalama bir değerle sonuçlanabilir. Optimal bir işlev için genellikle maksimize edici bir kombinasyon kullanılır (Ke vd., 2018). Havuzlama katmanı, aşırı uyum veya hesaplama ağırlıklarının azaltılmasında da yardımcı olur.

Üçüncü katman, genellikle evrişimli çıktı katmanı olarak adlandırılan tam bağlantılı katmandır (Bhatt vd., 2021). Tam bağlı katman, ileri beslemeli bir sinir ağına benzer. Bu katman genellikle ağı alt katmanında bulunur. Sonraki katmana gönderilmeden önce düzleştirilen son havuzlamadan veya evrişimli çıktı katmanından girdi alır. Çıktının eşit dağılımı, son havuzlama veya evrişimli katmandan sonra elde edilen sonucun tüm değerlerinin bir vektöre (3B matris) açılması anlamına gelir. Bu yöntem, çıktı konvolüsyonel katmanı tarafından temsil edilen bir özelliğin üst düzey doğrusal olmayan kombinasyonlarını incelemek için basit bir tekniktir.

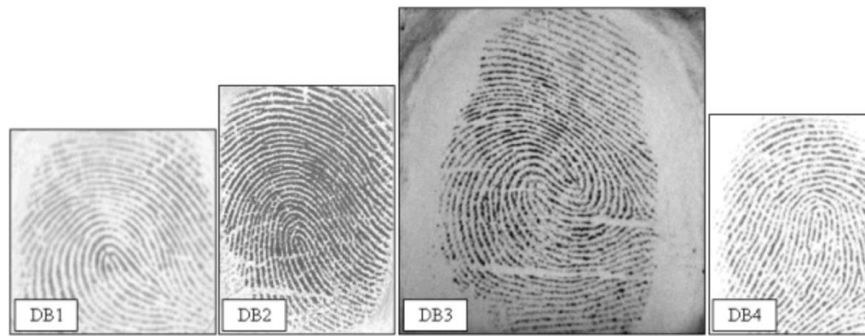
CNN katmanlarında bir etkinleştirme işlevi önemli bir rol oynar. Filtrelenmiş çıktı, aktivasyon adı verilen başka bir matematiksel işlev sağlar. Rektifiye Doğrusal Birimden (ReLU) (Wang vd., 2021), CNN kullanılarak özellik çıkarmada en yaygın etkinleştirme işlevidir. Aktivasyon fonksiyonunun temel amacı, ‘evet’ veya ‘hayır’ gibi bir sinir ağının nihai çıktısına karar vermektir. Etkinleştirme işlevi, çıktı değerlerini -1 ile 1, 1 ile 0 vb. arasında eşler.

5. PARMAK İZİ VE YÜZ TANIMA

5.1 Veri Setleri

Güvenilir ve doğru parmak izi tanıma, birçok bağlamda sağlam algoritmalar gerektiren zorlu bir örüntü tanıma sorunudur. Parmak İzi Doğrulama Yarışması (Fingerprint Verification Competition, FVC) (Maio vd., 2002) ile şirketlerin ve akademik kurumların performanslarının net bir şekilde karşılaştırma yapılmasına ve parmak izi tanıma algoritmalarındaki önemli gelişmeleri yakından izlemesine olanak tanıyan ilk ortak performans ölçütü oluşturmaya çalışılmıştır.

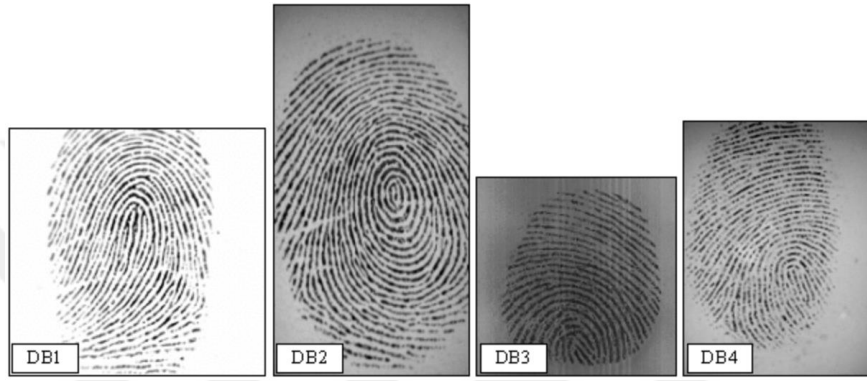
FVC 2000, Parmak İzi Doğrulama Algoritmaları için ilk uluslararası yarışmadır (URL-3, 2010). Geniş bir veri seti olan FVC 2000’de parmak izi algılama tekniklerindeki son gelişmeleri kapsayacak şekilde dört farklı “sensör” kullanılmıştır. Aslında, veri setleri 1 ve 2, iki küçük boyutlu ve düşük maliyetli sensör (sırasıyla optik ve kapasitif) kullanılarak toplanmıştır. Veri seti 3, daha yüksek kaliteli (geniş alanlı) bir optik sensör kullanılarak toplanmıştır. Son olarak, veri seti 4’teki görüntüler sentetik olarak oluşturulmuştur. Her veri seti, 800 görüntüden oluşan sıralı bir “test” kümesine ve algoritma ayarlaması için katılımcılara sunulan 80 görüntüden oluşan açık bir “eğitim” kümesine bölünmüştür. Şekil 5.1’de, FVC 2000’deki her veri setinden örnek görüntüler gösterilmiştir.



Şekil 5.1 FVC 2000 her veri setinden bir parmak izi görüntüsü

FVC 2002, Parmak İzi Doğrulama Algoritmaları için ikinci uluslararası yarışmadır. Değerlendirme Nisan 2002’de yapılmıştır (URL-4, 2010). Veri toplamının sonunda, 30 gönüllü kullanarak her veri seti için toplam 120 parmak izi ve parmak başına 12

adet görüntü (1440 görüntü) toplanmıştır. Bununla birlikte, FVC 2002 testinde kullanılacak her veri setinin boyutu, 110 parmak izi, parmak başına 8 baskı (880 görüntü) olarak belirlenmiştir. Bazı ek verilerin toplanması, toplama hataları durumunda test veritabanlarına dahil edilecekleri sistematik olarak seçilmesine olanak tanımıştır. Elde edilen sonuçlar, 16. Uluslararası Örüntü Tanıma Konferansı'nda sunulmuştur. Şekil 5.2'de FVC 2002 her veri setinden örnek bir parmak izi görüntüsü gösterilmiştir.



Şekil 5.2 FVC 2002 her veri setinden bir parmak izi görüntüsü

FVC 2004, şirketlere, akademik araştırma gruplarına ve bağımsız geliştiricilere açık parmak izi tanıma algoritmalarının bir teknoloji değerlendirmesi olan üçüncü uluslararası Parmak İzi Doğrulama Yarışmasıdır. FVC 2004'ün organizasyonu Nisan 2003'te başlamıştır ve nihai değerlendirmeler Ocak-Mart 2004'te İtalya'nın Bologna Üniversitesinde yapılmıştır. Toplam olarak 67 algoritma değerlendirilmiştir: Kesin giriş/çıkış belirtileriyle uyumlu olan ikili yürütülebilir bir program biçiminde sağlanan her algoritma, daha önce katılımcılar tarafından görülmemiş dört yeni parmak izi veri setinde test edilmiştir. (URL-5, 2010). Her veritabanının temsili bir alt kümesi, görüntü boyutunu ve veritabanlarındaki parmak izlerinin değişkenliğini karşılamak için algoritma ayarlaması için katılımcıların kullanımına sunulmuştur. Şekil 5.3'de FVC 2002'deki her veri setinden örnek bir parmak izi görüntüsü gösterilmiştir.

FVC 2004'te iki farklı kategori (açık ve hafif) kabul edilmiştir. Burada hafif kategori, düşük bilgi işlem gereksinimleri, sınırlı bellek kullanımı ve küçük şablon boyutu ile

karakterize edilen algoritmalar için tasarlanmıştır. Bu tez çalışmasında bu sınırlamalar dikkate alınmamıştır. Bu nedenle algoritmaları açık kategorisinden alınmıştır.



Şekil 5.3 FVC 2004 her veri setinden bir parmak izi görüntüsü

5.2 Gri Düzey Eş Oluşum Matrisi

Görüntüdeki piksellerin odaklama bilgilerini hesaplamak için görüntünün histogram ve yoğunluk gibi niteliksel bilgiler kullanılmaktadır. Hesaplama yalnızca histogramların kullanılması, yalnızca yoğunlukların dağılımı hakkında bilgi taşıyan ancak o dokudaki piksellerin birbirine göre göreceli konumu hakkında bilgi taşımayan doku ölçümleri ile sonuçlar elde edilir. Buna çözüm olarak gri seviyeli eş oluşum matrisine dayalı istatistiksel bir yaklaşım kullanmak, bir görüntüdeki komşu piksellerin göreceli konumu hakkında değerli bilgiler sağlamaya yardımcı olacaktır (Eleyan ve Demirel, 2011).

Gri Seviyeli Birlikte Oluşum Matrisi (GLCM) yöntemi, istatistiksel doku özellikleri için ikinci dereceden bir çıkarımdır. İki piksel arasında istatistiksel bir ilişki gösterir. GLCM, bir görüntüdeki gri düzeylerin (G) sayısı ile orantılı satır ve sütun sayısına sahip bir matristir. GLCM yöntemi gri tonlamalı bir görüntü kullanır (Abregtsen, 2008). GLCM vurgularını hesaplamamanın ilk adımı, bir renkli (Kırmızı, Yeşil, Mavi; RGB) görüntüyü gri tonlamalı bir görüntüye dönüştürmektir. Bir sonraki adım, bir birlikte oluşum ızgarası oluşturmaktır. Ayrıca noktaya ve mesafeye bağlı olarak referans piksel ile bitişik pikseller arasındaki uzamsal ilişkiyi belirlemektir. Bir sonraki adım, oluşturma kafesine bir eş olay çerçevesi ekleyerek simetrik bir ızgara oluşturmaktır. Ardından, çerçevenin her bir bileşeninin olanaklarını sağlayarak simetrik ızgarayı standartlaştırın. Son adım, GLCM vurgularını hesaplamaktır (Jumarlis vd., 2022).

Bir GLCM, (i, j) . ögeyi içerir. Bu ögeler, i olayının sıklığını gösterir ve j ögesiyle birlikte oluşur. Bu, GLCM'nin temelde iki boyutlu bir histogram olduğu anlamına gelir. Bir birlikte oluşum matrisi, $P(i, j, d, \theta)$ olan göreceli frekanslarla tanımlanır. Burada i, j piksel gri seviyelerini gösterir, d mesafeyi gösterir ve θ açı tarafından belirtilen yönü temsil eder. Bir birlikte oluşum matrisi mesafe (r), açı (θ) ve gri seviye ölçekleri i ve j 'den oluşur.

Bir parmak izi görüntüsünün düzgün dokulara sahip bölgelere ayrılabilmesi gözlenir. Bu gözlemlerin bir sonucu olarak, düzenli doku bölgeleri birlikte oluşum matrisleri kullanılarak temsil edilebilir. Bu işlemler 0° , 45° , 90° ve 135° açılardaki eş oluşum matrisleri ile Denklem 5.1, Denklem 5.2, Denklem 5.3 ve Denklem 5.4 kullanılarak uygulanabilir (Topaloğlu, 2013):

$$P(i, j, d, 0) = \# \{((k, l), (m, n)) \in (L_r \times L_c) \times (L_r \times L_c) \\ k - m = 0, |l - n| = d, I(k, l) = i, I(m, n) = j\} \quad (5.1)$$

$$P(i, j, d, 45) = \# \{((k, l), (m, n)) \in (L_r \times L_c) \times (L_r \times L_c) \\ (k - m = d, |l - n| = -d) \text{ or } (k - m = -d, |l - n| = d) \\ I(k, l) = i, I(m, n) = j\} \quad (5.2)$$

$$P(i, j, d, 90) = \# \{((k, l), (m, n)) \in (L_r \times L_c) \times (L_r \times L_c) \\ k - m = d, |l - n| = 0, I(k, l) = i, I(m, n) = j\} \quad (5.3)$$

$$P(i, j, d, 135) = \# \{((k, l), (m, n)) \in (L_r \times L_c) \times (L_r \times L_c) \\ (k - m = d, |l - n| = d) \text{ or } (k - m = -d, |l - n| = -d) \\ I(k, l) = i, I(m, n) = j\} \quad (5.4)$$

5.3 Öznelik Çıkarma

Öznelik çıkarma (feature extraction), ilk ham veri kümesinin bölündüğü ve daha yönetilebilir gruplara indirildiği, boyut indirgeme sürecinin bir parçasıdır. Yani işlem yapmak istediğinizde daha kolay olacaktır. Bu büyük veri setlerinin en önemli özelliği çok sayıdaki değişkene sahip olmalarıdır. Bu değişkenlerin işlenmesi

aşamasında çok fazla bilgi işlem kaynağına ihtiyaç duyulabilir. Bu nedenle, öznitelik çıkarma, değişkenleri seçip özelliklerde birleştirerek bu büyük veri kümelerinden en iyi özelliği elde etmeye yardımcı olur, böylece veri miktarını etkin bir şekilde azaltır. Bu özelliklerin işlenmesi kolaydır, ancak yine de gerçek veri setini doğruluk ve özgünlükle tanımlayabilir.

Öznitelik çıkarma yöntemi, büyük bir veri kümeniz olduğunda ve herhangi bir önemli veya ilgili bilgi kaybı olmadan kaynak sayısını azaltmanız gerektiğinde kullanışlıdır. Öznitelik çıkarma, önemli özellikleri seçerek veri kümesinden gereksiz veri miktarını azaltmaya yardımcı olur.

Sonuç olarak, verilerin azaltılması, modelin daha az makine eforu ile oluşturulmasına yardımcı olur ve ayrıca makine öğrenimi sürecindeki öğrenme ve genelleme işlem adımlarının hızını artırır. Öznitelik çıkarma uygulamalarını şu şekilde sıralayabiliriz (Chatterjee, 2022):

Kelime Çantası (Bag-of-Words), doğal dil işlemede en çok kullanılan tekniktir. Bu süreçte bir cümleden, belgeden, web sitesinden vb. kelimeleri veya özellikleri çıkarırlar ve ardından kullanım sıklıklarına göre sınıflandırırlar. Yani tüm bu süreçte özellik çıkarımı en önemli kısımlardan biridir.

Görüntü işleme, en iyi ve en ilginç alanlardan biridir. Bu alanda temel olarak resimlerinizi anlamak için onlarla oynamaya başlayacaksınız. Bu yüzden burada, özellik çıkarma da dahil olmak üzere birçok teknik ve bunları işlemek için dijital bir görüntü veya videodaki şekil, kenarlar veya hareket gibi özellikleri algılamak için algoritmalar kullanıyoruz.

Otomatik kodlayıcılar: Otomatik kodlayıcıların ana amacı, doğası gereği denetimsiz olan verimli veri kodlamasıdır. Bu süreç denetimsiz öğrenme kapsamındadır. Bu nedenle, yenilerini üretmek için orijinal veri setinin kodlanmasından öğrenerek verilerden koda kadar temel özellikleri belirlemek için Özellik çıkarma prosedürü burada uygulanabilir.

Bir öznitelik çıkarıcı, her bireyde farklı olan parmak izi görüntülerinden çıkıntı uçlarının ve çıkıntı çatallanmalarının bulunmasını sağlar (Yao vd., 2001). Girdi parmak izi görüntüsünde çıkıntıların mükemmel bir şekilde konumlandırılması, ayrıntılar çıkarıldıktan sonra, inceltilmiş bir sırt haritasındaki noktaların çıkarılması gibi önemsiz bir görev olacaktır. Mükemmel bir sırt haritası elde etmek pratikte her zaman mümkün olmayabilir. Detay çıkarma algoritmaları genellikle giriş parmak izi görüntülerinin kalitesine bağlıdır. Bu, veritabanı sınıflandırmasının performansı için önemlidir.

Parmak izinin epidermal sırtların anormal oluşumları, doğum sonrası izler, iş izleri, veri toplama cihazlarındaki sorunlar vb. faktörler nedeniyle, parmak izi görüntüleri iyi tanımlanmış sırt yapıları olabilir. Güvenilir ayrıntılar çıkarma, otomatik parmak izi kimlik doğrulama sisteminin performansı için önemlidir.

5.4 Görüntü İşleme

Görüntü işleme çeşitli alanlarda kullanımı son yıllarda artmıştır. Görüntü işleme, çeşitli kaynaklardan alınan ham (işlenmemiş) görüntüleri iyileştirmek için yaygın olarak kullanılan bir yöntemdir. Görüntü işleme, gürültülerden arındırılmış bir görüntü elde etmek veya ondan bazı yararlı bilgiler çıkarmak amacıyla bir görüntüyü dijital forma dönüştürme ve üzerinde bazı işlemler gerçekleştirme yöntemidir. Görüntünün bir girdi olduğu ve çıkışın da bir görüntü veya görüntü ile alakalı özelliklerin olduğu bir tür sinyal dağıtımıdır (Rani, 2017).

Günümüzde yüz tanıma teknolojileri, yüzdeki benzersiz özellikleri rastgele (özellik tabanlı) ve fotometrik (görünüm tabanlı) özellikler kullanarak çeşitli veri kümeleriyle karşılaştırmak için karmaşık matematiksel temsiller ve eşleştirme süreçlerini kullanmaktadır (Rzemyk, 2017). Eşleştirme işlemi yüzün yapısını, şeklini ve oranlarını karşılaştırarak yapılmaktadır. Örnek olarak; gözler, burun, ağız ve çene arasındaki mesafe; göz yuvalarının üst hatları, ağzın kenarları; burun ve gözlerin yeri; ve elmacık kemiklerini çevreleyen alan verilebilir. Bağımsız bileşen analizi, ön yüz (eigenface) yaklaşımı, spectroface, Gabor dalgacık dönüşümü yöntemi yüz tanıma alanında öncülerden sayılabilir.

5.5 Görüntü İşlemede Kullanılan Temel Filtreler

Renkli görüntüler genellikle, yüz algılama ve yüz tanıma sistemlerinin doğruluğunu azaltan arka plan dağınıklığı içerir. Bunu iyileştirmek için, net gri tonlamalı sürümde tanıma gerçekleştirilmeden önce, renkli giriş görüntüsünden bu gereksiz renk verilerini kaldıran yöntemler bulunmaktadır.

Görüntü kırpma, belirli bir nedenden dolayı görüntüleri çevreleyen gereksiz materyali kaldırır. Görüntü sonradan işleme, ilgili verilerin çıkarılmasına yardımcı olabilir. Örneğin, yüz algılama sistemlerinde, görüntü kırılmasında yüzün en uygun konumda olmasını sağlamak için birçok çıkarma yöntemi kullanılmaktadır (Shorten ve Khosgoftaar, 2019).

Görüntü filtreleme algoritmaları, gürültünün görüntü üzerindeki etkisini azaltır. Sonuç olarak, görüntü filtreleme gri düzey tutarlılığını, arka plan beyaz gürültüsünü ve düzgünlüğü iyileştirir. Ek olarak, düzenli ters otomatik gerileme filtresi de netleştirilmiş bir çıktı görüntüsü sağlar.

Fourier sürekli teorisi, temelinde analiz edilmesi ve onaylanması ile birlikte bir boyuttaki sürekli sinyallere uygulanacak sinyali değerlendirmeye başlayan klasik bir teknolojidir (Weaver, 1989). Piksel başına her komşunun tahmin istatistiklerine bağlı olarak hasarlı görüntüdeki Gauss (Gaussian) gürültüsünü gidermek için Wiener filtresi kullanılabilir (Siddeq ve Yaba, 2009). Bu filtre gürültünün gücüne dayanmaktadır (hasarlı görüntüdeki kontrast gürültüsü). Büyük hacimlerde kontrast olduğunda, filtre çok az pürüzsüzleşir. Kontrast büyükse, filtre daha fazla yumuşatma üzerinde çalışacaktır. Bu filtre, görüntüyü geliştirmek için kullanılan diğer filtrelerden daha iyi sonuçlar verir.

Wiener filtresinin amacı, bir sinyali bozan gürültüyü filtrelemektir. İstatistiksel bir yaklaşıma dayanmaktadır. Tipik filtreler istenen frekans tepkisi için tasarlanmıştır. Wiener filtresi filtrelemeye farklı bir açıdan yaklaşır. Birinin orijinal sinyalin ve gürültünün spektral özellikleri hakkında bilgi sahibi olduğu varsayılır ve biri, çıktısı orijinal sinyale mümkün olduğunca yaklaşıp olan doğrusal zamanda değişmes

filtresini arar (Kazubek, 2003). Wiener filtreleri şu şekilde karakterize edilir (Shetti ve Patil, 2017):

- ✓ Varsayım: Sinyal ve (eklemeli) gürültü, bilinen spektral özelliklere sahip durağan doğrusal rastgele süreçlerdir.
- ✓ Gereklilik: Filtre fiziksel olarak gerçekleştirilebilir, yani nedensel olmalıdır (bu gereklilik bırakılabilir ve nedensel olmayan bir çözümle sonuçlanabilir).
- ✓ Performans kriterleri: Minimum ortalama kare hatası kullanılabilir.

Wiener filtreleme yönteminin temel matematiksel gösterimi, Denklem 5.1 ile ifade edilmektedir.

$$G(u, v) = \frac{H^*(u, v)P_s(u, v)}{|H(u, v)|^2 P_s(u, v) + P_n(u, v)} \quad (5.1)$$

P_s ile bölmek, davranışını açıklamayı kolaylaştırır:

$$G(u, v) = \frac{H^*(u, v)P_s(u, v)}{|H(u, v)|^2 + P_n(u, v)/P_s(u, v)} \quad (5.2)$$

Burada;

$H(u, v)$ = Bozulma fonksiyonu

$H^*(u, v)$ = Bozulma fonksiyonunun karmaşık eşleniği

$P_n(u, v)$ = Gürültünün güç spektral yoğunluğu

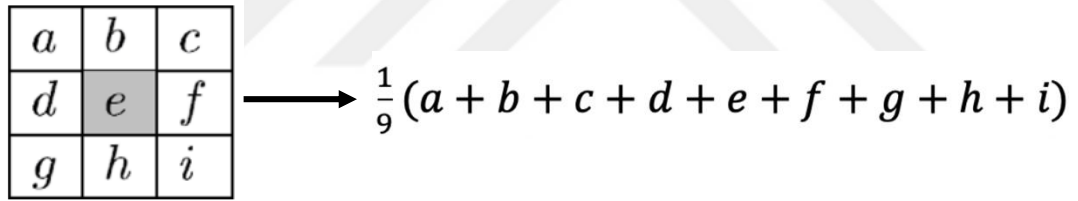
$P_s(u, v)$ = Bozulmamış görüntünün güç spektral yoğunluğu

P_n/P_s terimi, sinyal-gürültünün oranının karşılıklısı olarak yorumlanabilir.

Ortalama (mean) filtresi, görüntülerdeki gürültüyü azaltmak için kullanılan yöntemlerden biridir. Ortalama filtresi, görüntüleri yumuşatmak için kullanılan basit ve kolay uygulanan yöntemlerden birisidir. Ortalama filtre genellikle gürültüyü

azaltmak ve ayrıca bir pikselden diğerine yoğunluk değişimi miktarını azaltmak için kullanılır. Burada önce elemanların toplamı olan bir ortalama alınır ve toplamı eleman sayısına bölünür. Ardından, görüntüde üzerindeki her bir pikselin değeri o pikselin komşuluğundaki piksel değerlerinin ortalaması ile değiştirilmektedir (Thamotharan vd., 2012).

Ortalama filtreleme, yerel bir ortalama alma işlemidir ve en basit doğrusal filtrelerden biridir. Her pikselin değeri, yerel komşuluktaki tüm değerlerin ortalaması ile değiştirilir. Genellikle 3x3 (veya daha büyük) bir kare çekirdek filtresi kullanılır. Şekil 5.1’de tek bir 3x3 örnek gösterilmiştir (Lyra, 2011). Ortalama filtreleme işlemi, bir görüntüdeki her piksel değerini, kendisi de dahil olmak üzere komşularının ortalama değeriyle değiştirmektir. Bu, çevrelerini temsil etmeyen piksel değerlerini ortadan kaldırma etkisine sahiptir. Ortalama filtreleme genellikle bir evrişim filtresi olarak düşünülür. Diğer kıvrımlar gibi, ortalamayı hesaplarken örneklenecek komşunun şeklini ve boyutunu temsil eden bir çekirdeğe dayanır.



Şekil 5.4 Ortalama filtreleme yaklaşımı

Şekil 5.4, ortalama filtreyi kullanarak merkezi piksel değerinin e ’den $(a+b+c+d+e+f+g+h+i)/9$ olarak değiştirileceğini göstermektedir.

$f(i,j)$, gürültülü bir görüntü olsun, o zaman yumaşatılmış görüntü $g(x,y)$ ’dir. S , (x,y) ’nin bir komşuluğu ve n ise S ’deki piksel sayısıdır.

$$g(x, y) = \frac{1}{n} \sum_{(i,j) \in S} f(i, j) \quad (5.3)$$

6. DENEYSEL ÇALIŞMALAR

6.1 Kütüphane Ortamında Parmak İzi Tanıma İçin Yapay Sinir Ağı Yöntemlerinin Uygulanması

Tez çalışması kapsamında yapay sinir ağlarının üstün özellikleri kullanılarak parmak izlerinin sınıflandırılması ve analizi için geliştirilen uygulamada; FVC 2000, FVC 2002 ve FVC 2004 olmak üzere üç kıyaslama veri seti kullanılmıştır. Her veri setinde, yüz adet parmaktan toplanan sekiz yüz adet parmak izi görüntüsü ve parmak başına sekiz görüntü bulunmaktadır.

Tez çalışmasındaki deneylerde kullanılan parmak izi görüntüleri FVC2000, FVC2002 ve FVC2004 veri setlerinden elde edilmiştir. Tez çalışmasında veri setleri üzerinde ileri beslemeli çok katmanlı algılayıcı kullanılmıştır. Yapay sinir ağlarında en fazla kullanılan öğrenme metodu olan danışmanlı öğrenme yöntemlerinden ileri beslemeli çok katmanlı algılayıcının bu çalışmada tercih edilmesinin nedeni, karmaşık doğrusal olmayan ilişkileri çözmesi ve modelin öngörü başarısı iyisi olmasıdır (Alsarrar ve Gültepe, 2018).

FVC 2000 DB1, DB2, DB3 ve DB4 veri seti ile ilgili ayrıntı bilgi Tablo 6.1’de gösterilmiştir. Tablolarda parmak sayısı, parmak başına ait olan görüntü sayısı, görüntülerin boyutu ve inç başına düşen nokta sayısı (DPI) değerleri hakkında bilgiler verilmiştir.

Tablo 6.1 FVC 2000 DB1-DB4 veri setlerinin özellikleri

	Parmak Sayısı	Parmak Başına Görüntüler	Görüntü Boyutu	DPI
FVC2000 DB1	100	8	300x300	500
FVC2000 DB2	100	8	256x364	500
FVC2000 DB3	100	8	448x478	500
FVC2000 DB4	100	8	240x320	500

FVC 2002 DB1, DB2, DB3 ve DB4 veri seti ile ilgili ayrıntı bilgi, Tablo 6.2’de gösterilmiştir.

Tablo 6.2 FVC 2002 DB1-DB4 veri setlerinin özellikleri

	Parmak Sayısı	Parmak Başına Görüntüler	Görüntü Boyutu	DPI
FVC2002 DB1	100	8	388x374	500
FVC2002 DB2	100	8	296x560	569
FVC2002 DB3	100	8	300x300	500
FVC2002 DB4	100	8	288x384	500

FVC 2004 DB1, DB2, DB3 ve DB4 veri seti ile ilgili ayrıntı bilgi Tablo 6.3’de gösterilmiştir.

Tablo 6.3 FVC 2004 DB1-DB4 veri setlerinin özellikleri

	Parmak Sayısı	Parmak Başına Görüntüler	Görüntü Boyutu	DPI
FVC2004 DB1	100	8	640x480	500
FVC2004 DB2	100	8	328x364	500
FVC2004 DB3	100	8	300x480	512
FVC2004 DB4	100	8	288x384	500

Yukarıda Tablo 6.1, Tablo 6.2 ve Tablo 6.3’de gösterilen veri setleri, 730 parmak izi görüntüsü olmak üzere dört sınıfta sınıflandırılmıştır. Bu sınıflar ve toplam parmak izi görüntü sayısı şu şekildedir: Yay sınıfı: 169 görüntü, spiral sınıfı: 140 görüntü, sağ döngü sınıfı: 222 görüntü ve sol döngü: 199 görüntüdür.

Gizli katmandaki nöron sayısının belirlenmesinde deneme-yanılma yöntemi benimsenmiştir. Buradan hareketle herbir seri için en iyi yapay sinir ağı modeli doğrulama kümesinde en iyi sonucu veren ağ olarak seçilmiştir. Ele alınan parmak izleri için toplam dört veri seti için toplam 600 adet parmak izi veri yer almaktadır. Bu parmak izleri verilerin %60’ı eğitim kümesi ve %40’ı ise test kümesi olarak ele

alınmıştır. Bir başka ifade ile 360 tanesi rastgele olarak eğitim kümesine ve 240 tanesi ise test kümesine tahsis edilmiştir.

YSA uygulaması için MATLAB paket program kullanılmıştır. MATLAB, milyonlarca mühendis ve bilim insanı tarafından verileir analiz etmeki algoritmalar geliştirmek ve modeller oluşturmak için kullanılan bir programlama ve sayısal hesaplama ortamıdır. MATLAB, yinelemeli analiz ve tasarım süreçleri için ayarlanmış bir masaüstü ortamını matris ve dizi matematiğini doğrudan ifade eden bir programlama diliyle birleştirir. Yürütülebilir bir not defterinde kodu, çıktıyı ve biçimlendirilmiş metni birleştiren komut dosyaları oluşturmak için Canlı Düzenleyici içerir (URL-6, 2023).

MATLAB programında yapay sinir ağların eğitilmesinde ve test edilmesinde sinir ağı araç kutusu (Neural Network Toolbox) kullanılmaktadır. Sinir ağı araç kutusu kolay bir şekilde kapalı form denklemleri, doğrusal olmayan sistemler için fonksiyonlarının ve uygulamalarının modellenmesini sağlamaktadır. Sinir ağı araç kutusu ileri beslemesi ile danışmanlı öğrenme, radyal temelli ve dinamik sinir ağlarını desteklemektedir.

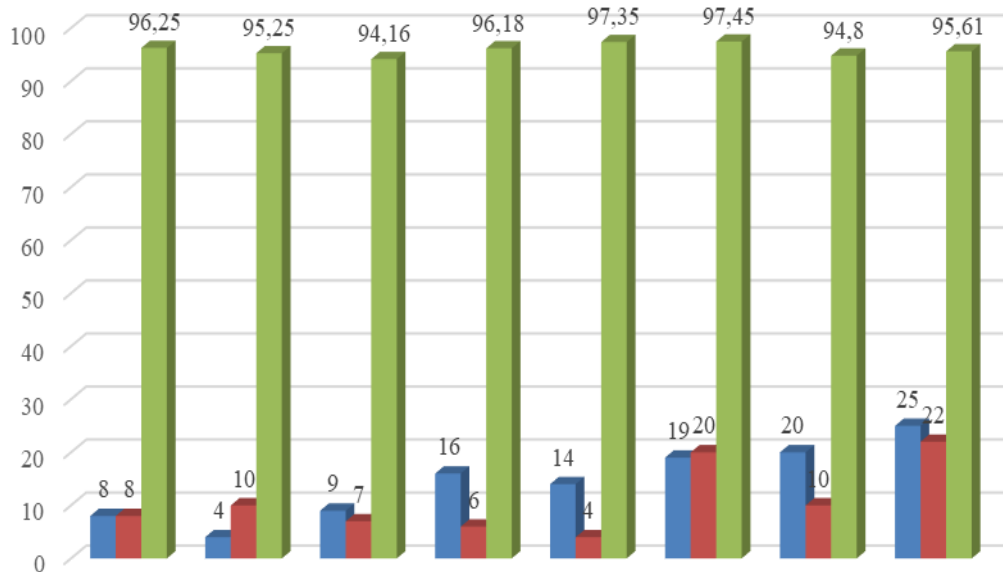
ÇKA, YSA modelleri içerisinde en çok kullanılan ağ tipidir. Avantajlarının yanı sıra YSA'nın bazı dezavantajları da bulunmaktadır. Bunlardan en önemlileri; sinir ağın parametre değerlerinin (öğrenme katsayısı, her katmandaki nöron sayısı, katman sayısı vb.) belirlenmesinde belli kuralların olmaması ve ağın eğitiminin ne zaman bitirileceğine karar vermek için geliştirilmiş belli bir kuralın olmayışıdır. Burada bu sayıları belirlerken deneme yanılma yolu ile belirlenmiştir.

Tez çalışmasında, üç gizli katmanlı YSA için optimal nöron sayısı araştırılmıştır. Bu nedenle Tablo 6.4'de yapılan denemeler sonucunda en yüksek doğruluk oranını elde etmek için gizli katmanlarda farklı nöron sayıları denenmiş ve en yüksek doğruluk oranı elde edilmiştir. Yapılan denemeler sonucunda geliştirilen yapay sinir ağları modelinde ikinci gizli katmandaki nöron sayısı 19, üçüncü gizli katmandaki nöron sayısı 20 olarak belirlendiğinde elde edilen en yüksek doğruluk oranı %97,45 olarak elde edilmiştir.

Tablo 6.4 ÇKA ağ performansı

2. Gizli katmandaki nöron sayısı	3. Gizli katmandaki nöron sayısı	Doğruluk (%)
8	8	96,25
4	10	95,25
9	7	94,16
16	6	96,18
14	4	97,35
19	20	97,45
20	10	94,80
25	22	95,61

Tablo 6.4’de verilen gizli katmanlardaki nöron sayıları arasındaki ilişkiler, görsel olarak Şekil 6.1’de sunulmuştur.



Şekil 6.1 Uygulanan çoklu nöronların performansının karşılaştırılması

Tez çalışmasında önerilen yöntem, literatürde yer alan bazı çalışmalarla karşılaştırılmıştır. Elde edilen deneysel sonuçlar önerilen yöntemin dört sınıflı parmak izi tahminlemede oldukça yararlı olduğunu ve yüksek doğrulukta sınıflandırma yapabildiğini göstermiştir. Tez çalışmasında önerilen metodun literatürde yer alan bazı çalışmalarla karşılaştırma sonuçları, Tablo 6.5’de verilmiştir.

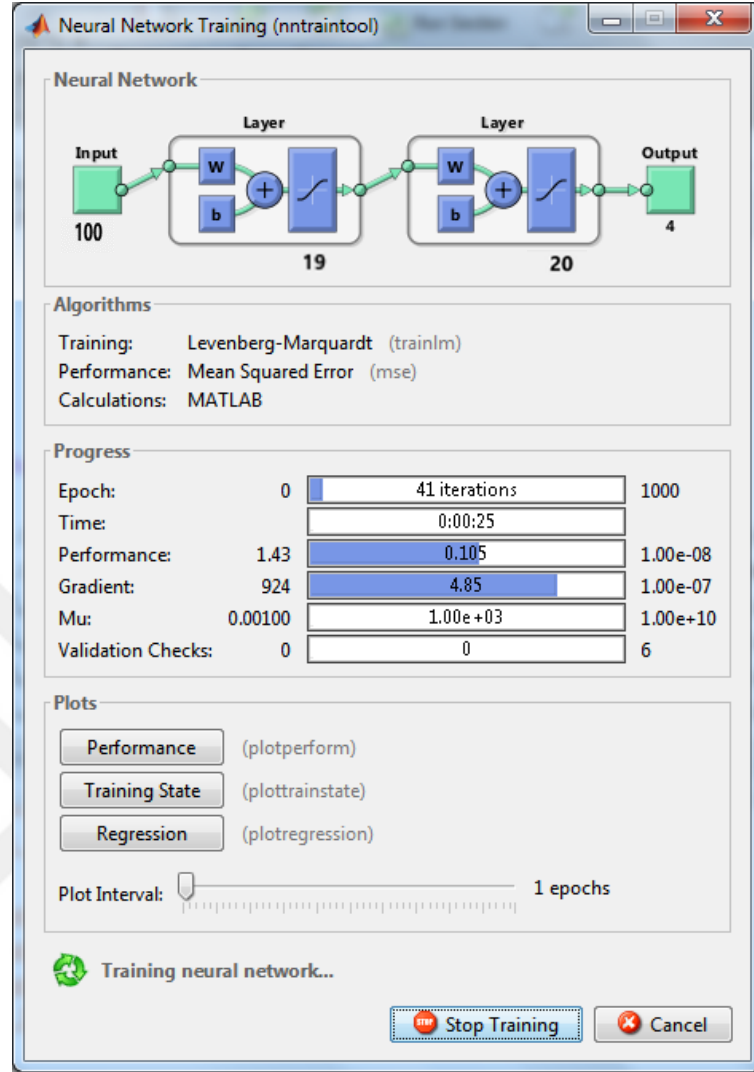
Tablo 6.5 Önerilen metodun karşılaştırılması

Çalışma	Sınıf Sayısı	Doğruluk (%)
Senoir (2001)	4	88,5
Chang ve Fan (2002)	5	94,8
Karu ve Jain (1996)	4	85,4
Karu ve Jain (1996)	5	91,1
Yao vd. (2001)	5	89,3
Mohamed ve Nyongesa (2002)	5	92,4
Önerilen Metot	4	96,42

MATLAB Neural Network toolbox kullanılarak YSA oluşturuldu. Modellenen yapay sinir ağında, Levenberg-Marquardt (Trainlm) geri yayılım algoritması olarak seçilmiştir. Levenberg-Marquardt algoritması az parametrelidir olduğu için problem çözümünde hızlı bir algoritmadır. Bunun yanı sıra bu çalışmada en iyi sonuç için transfer fonksiyonu olarak TANSIG, performans fonksiyonu olarak Ortalama Kare Hatası (Mean Squared Error, MSE), adaptif öğrenme fonksiyonu olarak LEARNGDM tercih edilmiştir. Tablo 6.6'da oluşturulan yapay sinir ağın parametreleri verilmiştir. Verilen YSA parametrelerine göre oluşturulan model, Şekil 6.2'de gösterilmiştir.

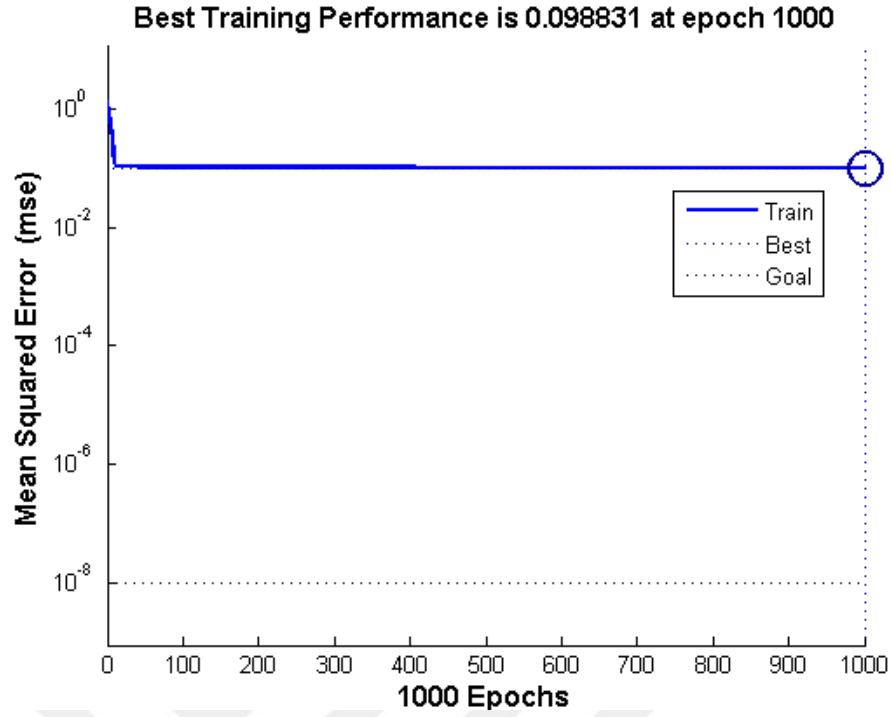
Tablo 6.6 YSA parametreleri

Yapay Sinir Ağı Parametreleri	
Ağ Tipi	İleri beslemeli geri yayımlı
Transfer Fonksiyonu	TANSIG
Performans Fonksiyonu	MSE
Adaptif Öğrenme Fonksiyonu	LEARNGDM
Eğitim Fonksiyonu	TRAINLM
2. Gizli Katman Hücre Sayısı	19
3. Gizli Katman Hücre Sayısı	20
Öğrenme Oranı	%60
Test Oranı	%40



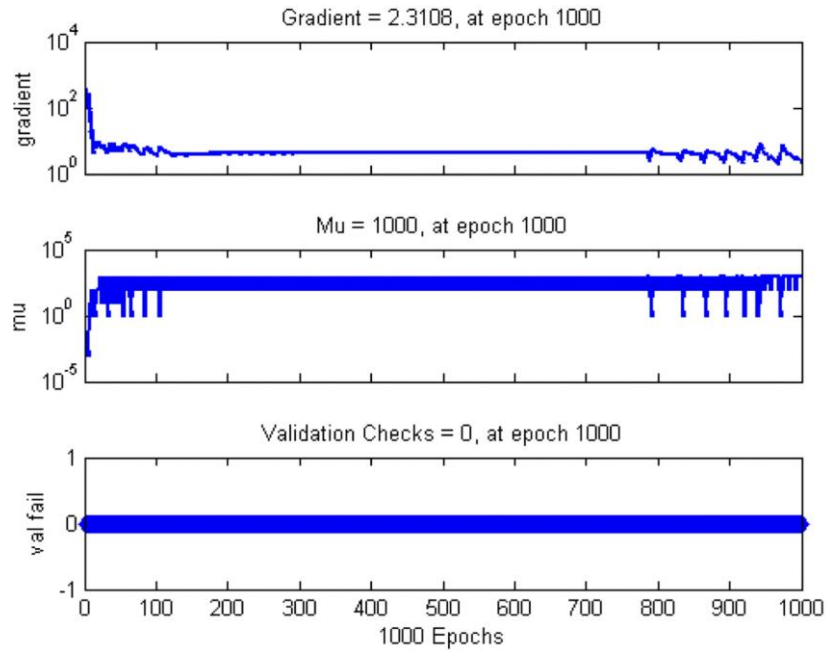
Şekil 6.2 Yapay sinir ağı modeli

Şekil 6.3’de parmak izi veri setinden elde edilen değerler ile YSA’dan elde edilen değerler arasındaki sapma grafiği gösterilmiştir. Şekil 6.3’de görüldüğü gibi YSA’nın eğitim aşaması başladığında yüksek MSE değeri artan periyotlarla (epoch) düşmektedir. Bu, geliştirilen YSA modelinin eğitim aşamasının kabul edilebilir bir şekilde önceden planlandığı anlamına gelir. Grafiğe göre MSE oranı sürekli düşüyor ve 1000 iterasyondan hemen sonra en düşük MSE değeri olan 0,098831 ile en iyi sonuçtur. YSA’nın 1000 gibi iterasyon ile minimum MSE değeri ile ideal sonuca ulaşması deney setinde kullanılan veri optimizasyonunun ideal şekilde yapıldığının bir göstergesidir.



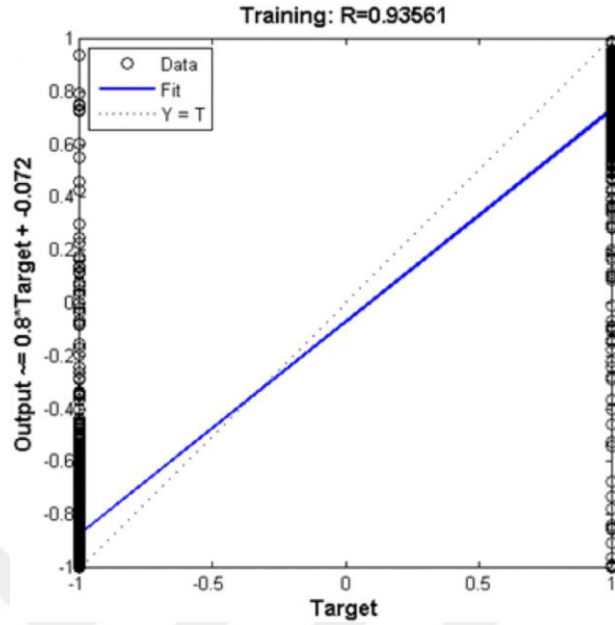
Şekil 6.3 YSA'nın performans grafiği

YSA modeline ait YSA doğrulama kontrolleri, Mu ve Gradient değerleri Şekil 6.4'de gösterilmiştir. Şekil 6.4'de görüldüğü gibi hatalar 300. Epok'tan sonra tekrarlanmış ve 800. Epoch'ta durmuştur. Bu hata 300. Epoch'tan itibaren tekrarlanan verilerin aşırı eşleşmesinin çok iyi olduğunu göstermiştir.



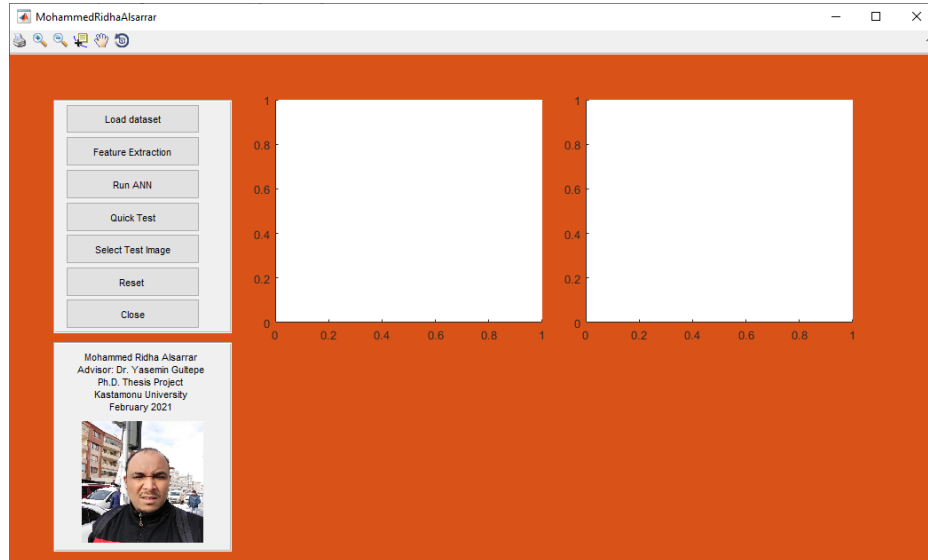
Şekil 6.4 YSA doğrulama kontrolleri, Mu ve Gradient değerleri

Şekil 6.5’de eğitim sonucu önerilen YSA modelinin eğitim oranı gösterilmiştir.



Şekil 6.5 Levenberg Marquadt eğitim veri performansı

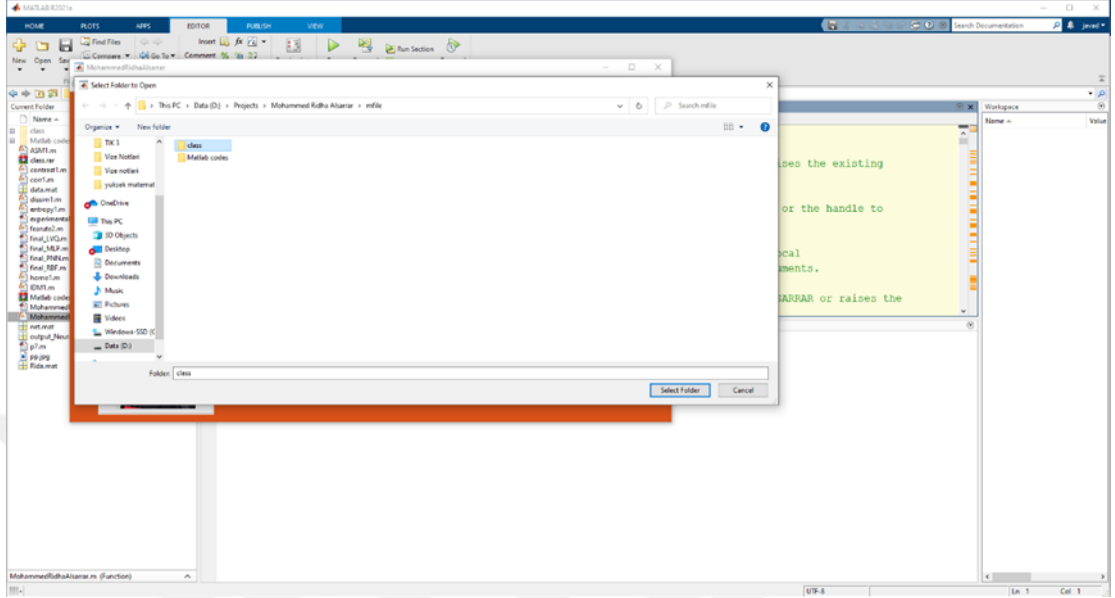
Programın değerlendirilmesi için bu tezde Grafik Kullanıcı Arayüzü (GUI) kullanılmıştır. Bu sonuçlar Şekil 6.6’dan Şekil 6.10’a kadar gösterilmektedir. Şekil 6.6, GUI sonucunun genel gösterimini göstermektedir.



Şekil 6.6 Giriş ekranı

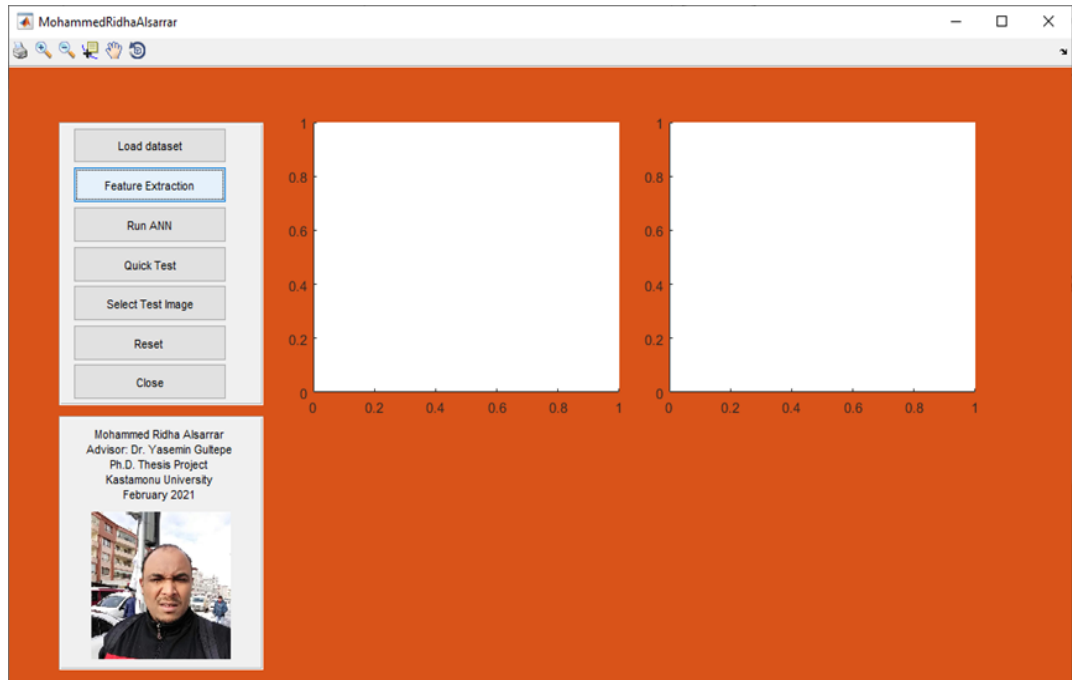
Şekil 6.6’da gösterilen giriş ekranında yedi adet işlemleri teslim eden menüler bulunmaktadır. Birinci sıradaki Load Dataset’dir. Bu menu ile parmak izlerinin

bulunduğu veri kümesi sisteme eklenmektedir. Şekil 6.7’de veri kümesinin seçilme ekranı gösterilmektedir.



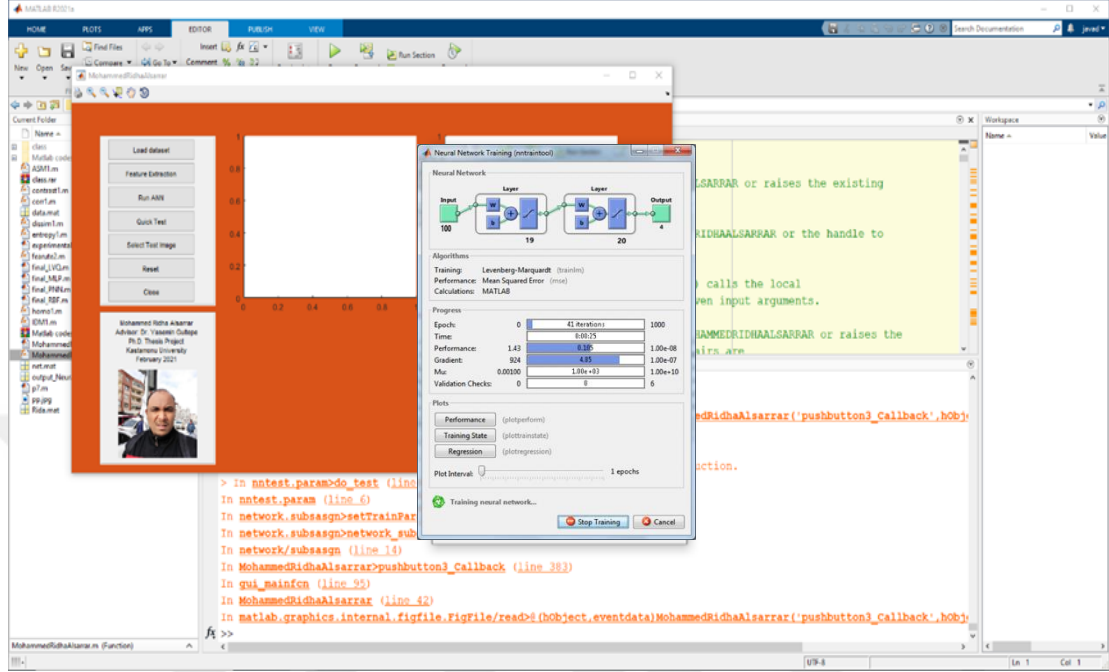
Şekil 6.7 Veri kümesinin seçilmesi

Şekil 6.8’de gösterildiği gibi sisteme aktarılması istenilen veri kümesi, kullanıcı bilgisayar ekranından seçilir ve ardından “Load” tuşuna basılarak veri kümesi sisteme yüklenir.



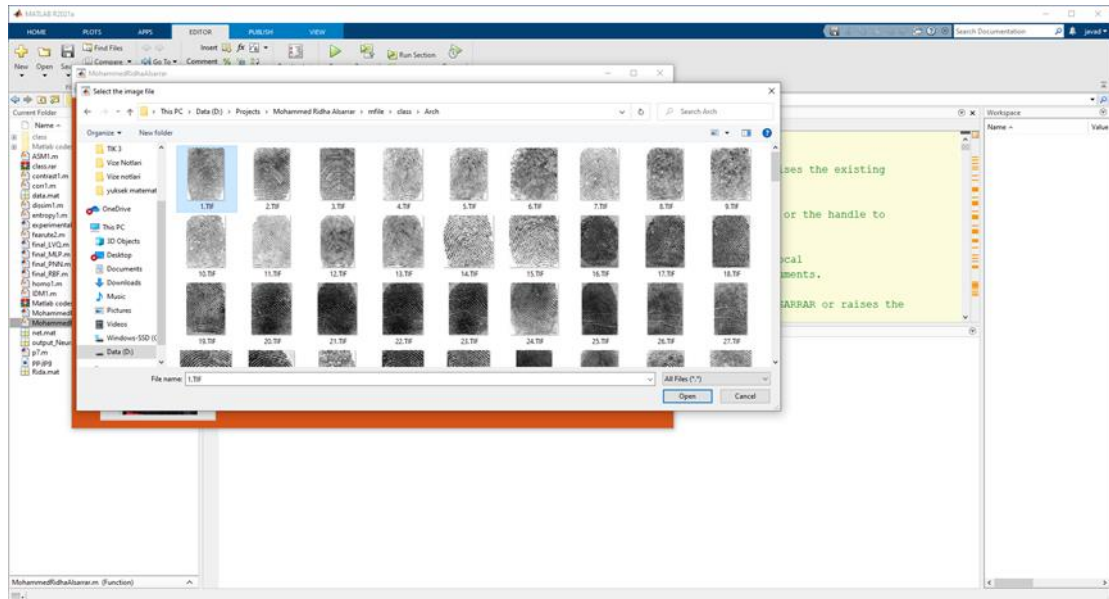
Şekil 6.8 Load tuşuna basıldıktan sonraki ekran görüntüsü

Ekrana gelen formdan kişi görüntüyü seçer ve ardından kullanıcı arayüzünü yükler, “YSA Çalıştır” a tıklayın. Elde edilen sonuçlar, Şekil 6.9’da verilmiştir.



Şekil 6.9 YSA modeli çalıştırma ekranı

Sonraki adım olarak “Test Görüntüsünü Seçin” butonu seçilir. Ardından elde edilen deneysel sonuç görüntüleri gösterilir. Bu işlem adımları Şekil 6.10’da gösterilmiştir.



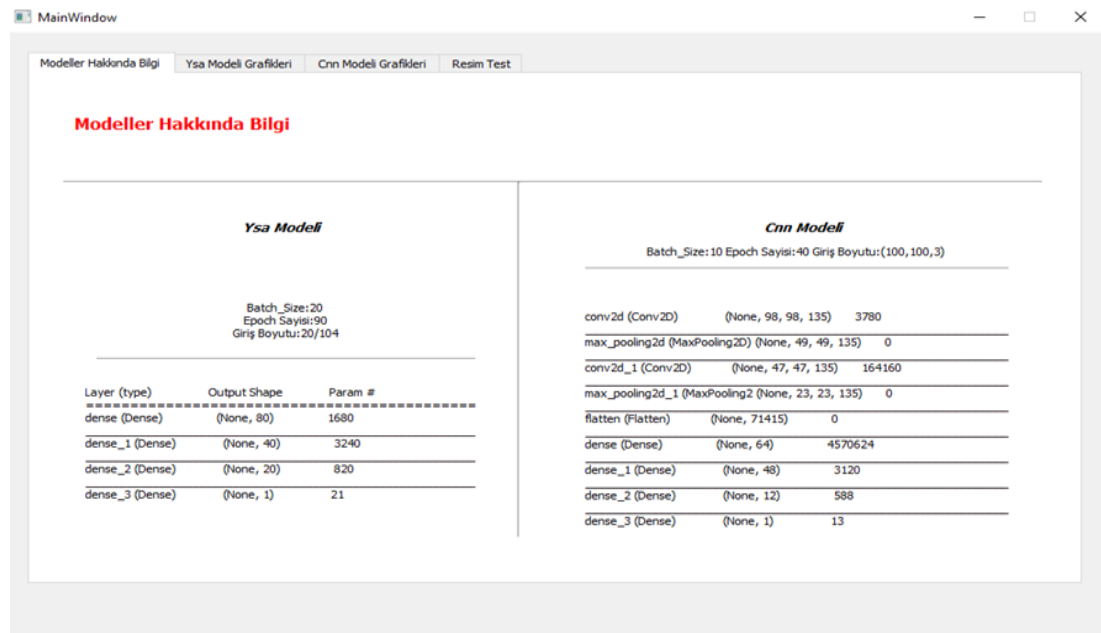
Şekil 6.10 Program çalıştırıldıktan sonra ekran görüntüsü

Ayrıca, kullanıcılar için kullanıcı arayüz (user interface) ortamında işlemleri hızlı bir şekilde sonuçlandırmak ve değerleri başlangıç değerine getirme ve çıkış için kullanılan program formunda "Quick Test", "Reset" ve "Close" isimli çeşitli butonlar (düğmeler) mevcuttur.

6.2 Evrişimsel Sinir Ağları Kullanılarak Kütüphane Ortamında Parmak İzi Tanıma Sisteminin Geliştirilmesi

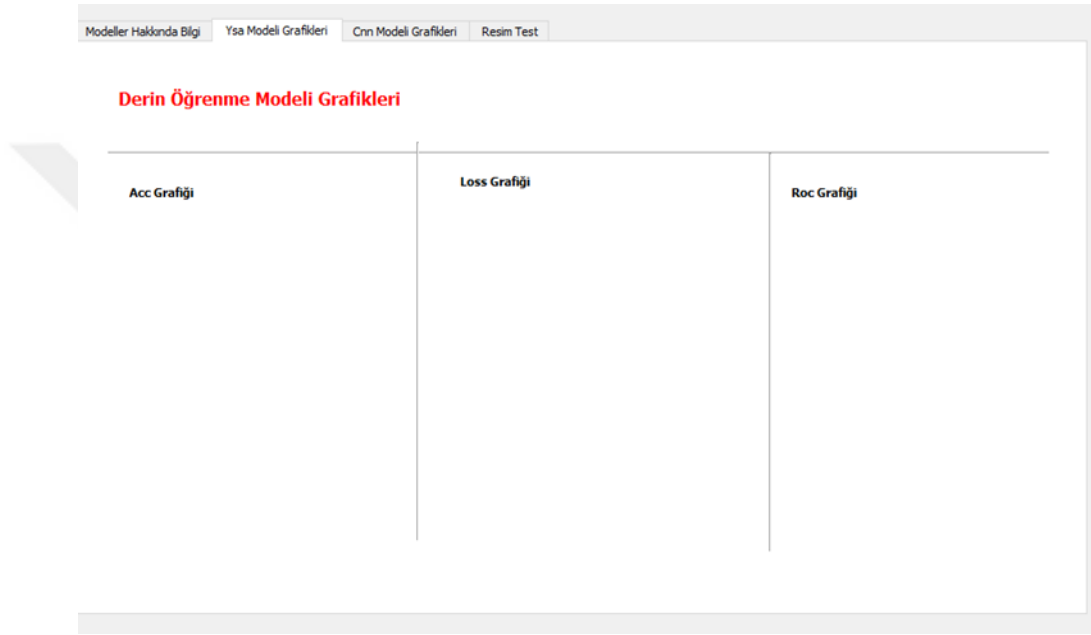
FVC 2000, FVC 2002 ve FVC 2004 veri setleri üzerinde görüntü işleme yapılmıştır. Bunun için hem de Python geniş kütüphaneleri ile bize imkân sunduğu için Python dili kullanıldı. Öncelikle görsel bir arayüz hazırlanmıştır. Python Spider IDE'sinde grafik kullanıcı arayüzleri (GUIs) tasarlamak ve oluşturmak için QT aracı olarak QT Designer ile görsel arayüzler tasarlanmıştır.

Arayüz dört bölümden oluşmaktadır. İlk kısımda YSA ve CNN verileri gösterilmektedir. Şekil 6.11'de YSA ve CNN verilerinin gösterildiği arayüz gösterilmektedir. Test ve eğitim için verisetine göre ne kadar veri alındığı, epoch verisi ve resim boyutu yer alır. Alınan veriler bir sonraki kısımda işleme konulacaktır. Karışıklık matrisinin verileri de CNN içerisinde gösterilmiştir. Farklı veri setine göre veriler de değişecektir.



Şekil 6.11 YSA ve CNN verilerinin gösterildiği arayüz

YSA için; başta toplu (batch) ve epoch sayısı verilmiştir. Tüm veri setinin sinir ağıları boyunca bir kere gidip gelmesine (ağırlıkların güncellenmesi) epoch denir. Epoch'un arttırılması daha iyi verim sağlamayabilir tam tersine eğitim verisini ezberlemeye başlamasına neden olabilir ve yeni bir veride (eğitim verisinde olmayan) sonuçları istenilen performansta olmayabilir. Epoch'un az olması da istenilen performansa ulaştırmayabilir. Şekil 6.12'de modele ait grafiklerin gösterildiği arayüz ekranı gösterilmektedir.



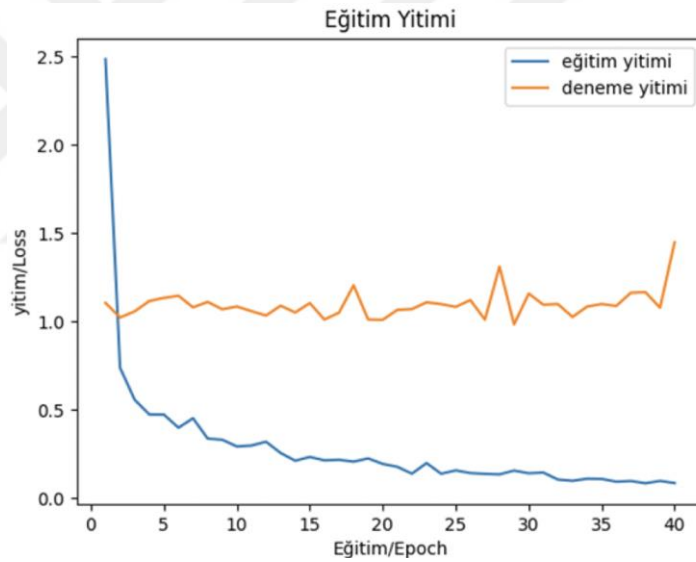
Şekil 6.12 Model grafiklerinin gösterildiği arayüz

Veri setindeki görüntü sayısı optimum değerlere böldüğünde 90 epoch ile overfitting (aşırı öğrenme veya ezberleme) aşamasına gelmeden eğitim verebileceği ön görülmüştür ve bu konuda başarılı da olunmuştur. Bir epoch içerisinde, veri seti içerisindeki bir veri dizisi sinir ağlarında sona kadar gider daha sonra orada bekler batch boyutu kadar veri sona ulaştıktan sonra hata oranı hesaplanır ve optimizasyon işlemi yapılır (yani ağırlıklar güncellenmektedir). Epoch boyutunu az tutarsak, fazla güncelleme meydana gelecek ve model hatalı eğitilecektir. Bunun için de 20 batch verilmiştir.

Tez çalışması kapsamında parmak izi tahminlemesi için dört katmanlı bir yapay sinir ağı modeli oluşturulmuştur. Sonrasında her bir katmandaki shape işlemi yapılmıştır. *Shape*, Python'da bir fonksiyondur. OpenCV kütüphanesi içine dahildir. Resim

boyutunu öğrenmek amacı için kullanılmaktadır. Daha sonrasında NxN ile mükemmel örnek için ne kadar boyuta gelindiği yazılmıştır sonunda da parametre boyutu belirlenmiştir.

CNN için; modelin gerekliliği olarak model üç parametre alır, yani görüntüyü daha sonrasında tanıyabilmek için resmi 3D olarak ele alır. Bunlar; En, boy, yüksekliktir. Tıpkı YSA da olduğu gibi epoch ve batch boyutu belirledikten sonra bu üç parametreyi giriş parametreleri olarak verilmiştir. Sonrasında önce conv2D de resmi iki boyuta indirgenmiştir, maxPooling'de maksimum olabilecek değerleri girilmiştir. Bu işlemleri modelin iki boyutu için de tekrar edilmiş daha sonrasında Flatten ile düzelleştirmek yapılarak YSA ile aynı mantıkta olacak şekilde her bir katmandaki dense değerleri yazılmıştır.

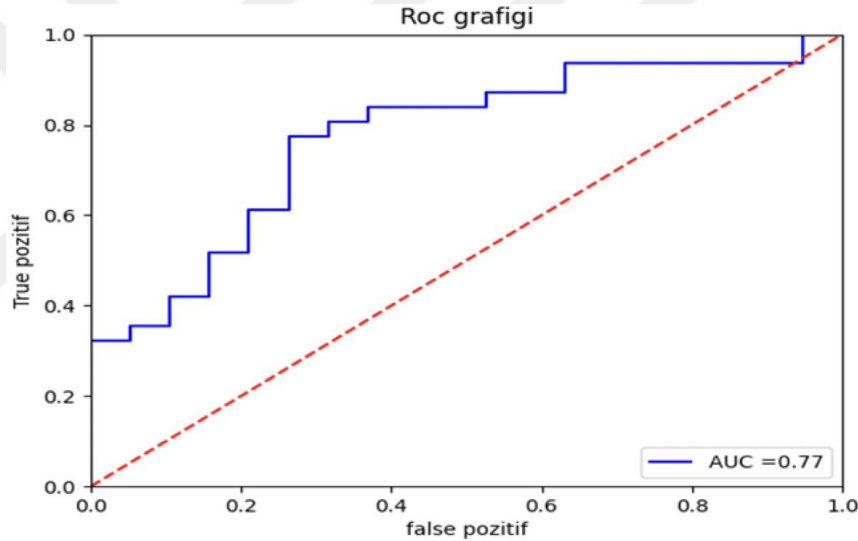


Şekil 6.13 Eğitim kaybı

Görsel gösterim için modellerin eğitim sürecindeki Loss (kayıp/yitim) ve Accuracy (doğruluk) değerleri ve Alıcı Çalışma Karakteristikleri (ROC) eğrileri, işlenen verilerin sonucunu ortaya çıkarmak amacıyla kullanılmıştır. Sistem geliştirilirken ağırlıklı olarak Matplotlib kütüphanesi kullanılmıştır. ROC eğrisi, sınıflandırma problemleri için çok önemli bir performans ölçümüdür. ROC bir olasılık eğrisidir ve altında kalan alan olan Eğri Altında Kalan Alan (AUC) ayrılabilirliğin derecesini veya ölçüsünü temsil eder.

Şekil 6.13’de eğitim kaybı grafiği gösterilmiştir. Grafiği incelediğimizde deneme başarısının yer yer alçaldığını ve yükseldiğini, buna karşın eğitim başarısının sürekli artışta olduğunu gözlemlenmiştir. Deneme içerisinde yükseliş ve alçalışların olması modelin overfitting olmadan eğitildiğinin bir göstergesidir ki düzgün eğitime eğitim verileri sürekli artış gösteremezdi. Ancak epoch boyutu arttıkça düşüşler gözlemlenmiştir. Bu gözlemler epoch-batch sayılarının belirlenmesinde yardımcı olmaktadır.

Loss değerlerine göre epoch arttıkça eğitim yitiminin azaldığını görebilirsiniz. Ancak belirli bir değerden sonrası aşırı öğrenmeye sebep olacaktır. Bu yüzden denemesine göre optimum bir devir sayısı (epoch) na göre sınıflandırma yapılmıştır.

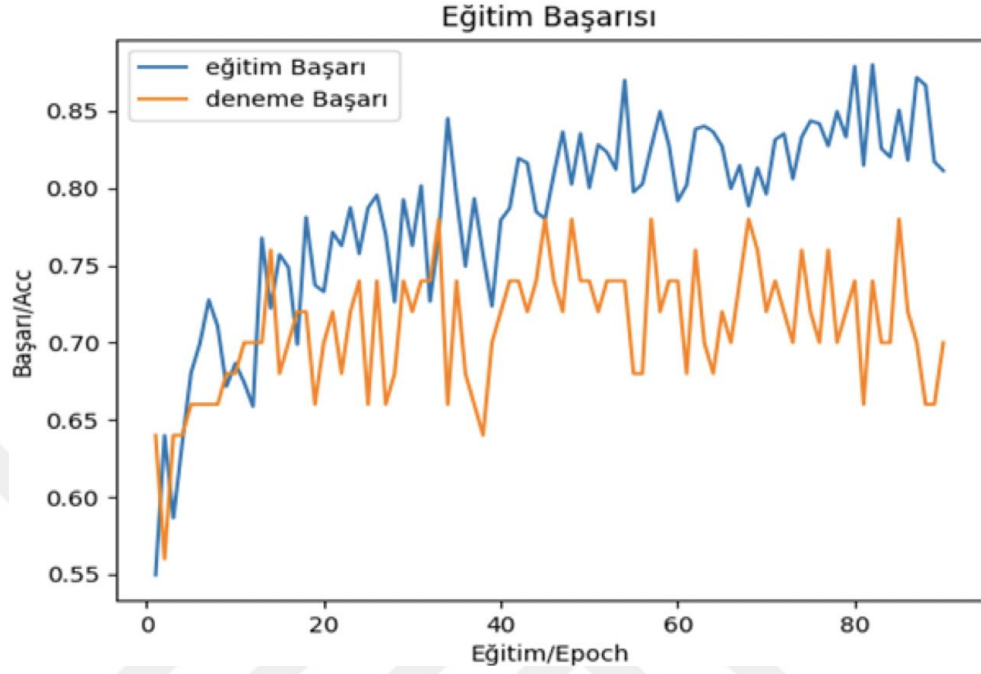


Şekil 6.14 ROC grafiği

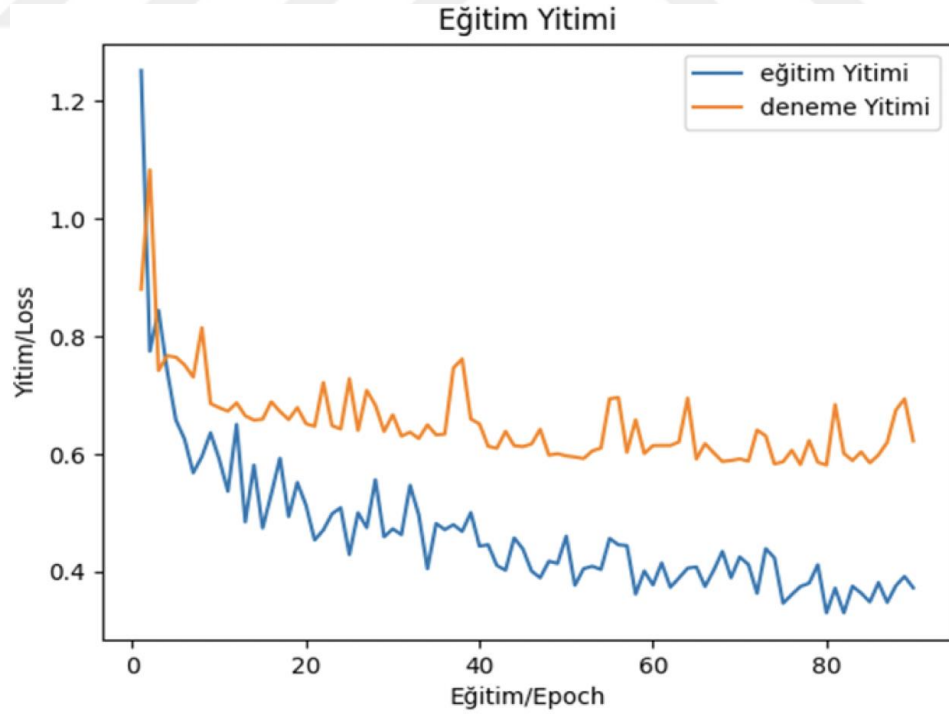
Şekil 6.14’de ROC grafiği gösterilmiştir. Gösterilen ROC eğrisini inceleyecek olursak; AUC’nin değerinin 0,77 olduğu görülmektedir. Optimum AUC değeri 1 olarak kabul edilir. AUC’un altında kalan alan ne kadar büyükse eğitim o kadar başarılı sayılır. Alttaki alan yaklaşık %80 civarındır, bu da eğitimin başarılı bir şekilde ilerlediğini gösterir.

Şekil 6.15’de tez çalışmasında kullanılan parmak izi veri setleri ile CNN modelinin 90 epok boyunca eğitim (train) ve test veri kümesiyle eğitilmesi sırasında elde edilen doğruluk (accuracy) oranı değerleri grafiksel olarak gösterilmiştir. Şekil 6.16’da ise kayıp (loss) değeri grafiksel olarak gösterilmiştir. Grafikler incelendiğinde; ilk

iterasyondan itibaren ağı hızlı bir şekilde öğrendiği, 99 iterasyonun sonunda %77 civarında bir eğitimin gerçekleştiği görülmektedir.

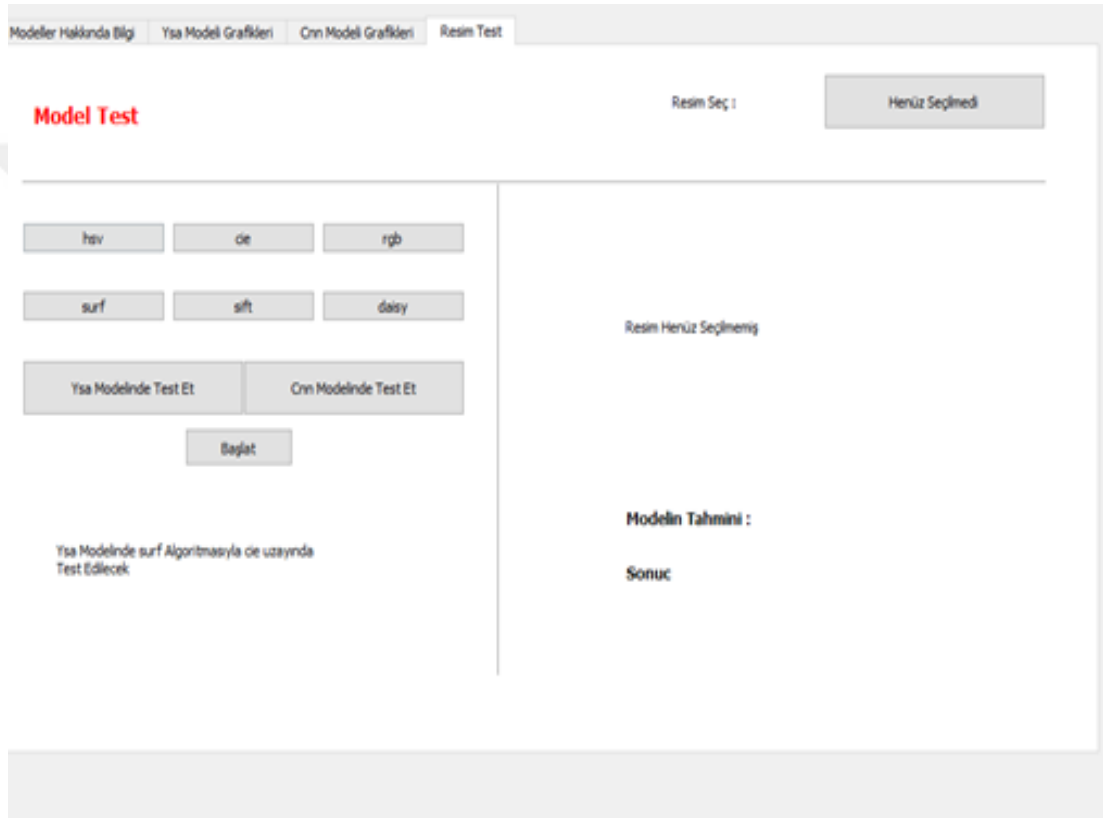


Şekil 6.15 CNN için eğitim ve deneme başarı grafiği



Şekil 6.16 CNN için eğitim kaybı

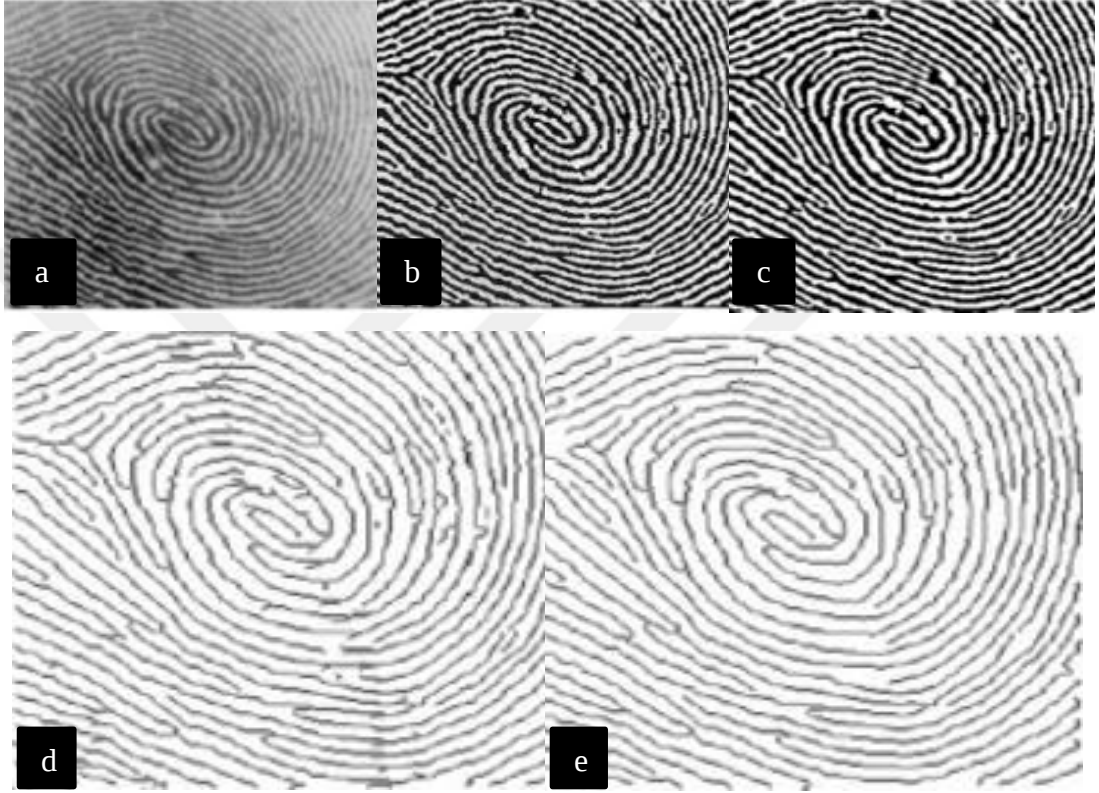
YSA'ya göre CNN'de katman sayının fazla olması nedeniyle işlem adımları biraz daha karmaşık ilerlemektedir. Epoch değeri arttıkça eğitim başarımı da artmıştır. Ancak CNN yapı olarak overfitting olmaya daha müsait bir yapısı vardır. Bu yüzden deneme verisetinde bazen çok kararsız veriler, kayıplar ve başarılar ortaya çıktı. Modelde epoch belli değerlerin üzerine çıktığında kayıp artmaya başladı bu yüzden epoch sayısını belirli sabit değerin üzerine çıkarılmazlar. Eğitim (train) aşamasında testten daha başarılı sonuçlar elde edilmiştir. Şekil 6.17'de sonuç ekranı gösterilmiştir.



Şekil 6.17 Sonuç ekranı

Tez çalışmasında önerilen sisteme literatürde sıklıkla kullanılan yöntemler eklenmiştir. Parmak izi veri setlerindeki görüntüden $N \times N$ boyutundaki alanlar alınıp üzerlerinde görüntü işleme teknikleri gerçekleştirilmiştir. Veri seti içindeki parmak izi görüntüler, yukarıdaki resim seçme alanından seçilir. Şekil 6.17'de gösterilen ekranda sol tarafta bulunan algoritmalar ile hangi model üzerinde işlem yapılmak isteniyorsa ona uygun buton seçilme işlemi yapılır. Başlat butonuna tıklandığında resmin temiz olup olmadığı ve oranı alt kısımda yazmaktadır.

Arka planda incelenen resimin yukarıda gösterilen hali data_ysa dosyasına hangi seçeneklerle işlendiyse kaydedilir. Buna göre parmak izlerinin en temiz hali çıkarılmış olup sonrasında yapılacak başka bir parmak izinden kişi tanıma gibi projelere hazır hale getirilmektedir. Şekil 6.18’de veri seti üzerinde görüntü işleme işlemleri sonucunda elde edilen görüntüler gösterilmiştir.



Şekil 6.18 (a) Giriş parmak izi, (b) Görüntü zenginleştirme, (c) İkili haldeki görüntü, (d) İnceltilmiş c görüntüsü, (e) İyileştirilmiş d görüntüsü

6.3 Kütüphane Ortamında Yüz Tanıma Sisteminin Geliştirilmesi

Genel olarak yüz tanıma, sorgulanan bir yüzün veritabanındaki kayıtlı bilenen yüzlerle karşılaştırılması sonucu sorgulanan yüzün kimliğini doğrulamak veya bu yüzün kimliğini tespit etmek olarak tanımlanabilir. Yüz tanıma genel anlamda, statik bir görüntüden ya da bir video görüntüsünden bir veya daha fazla kişiyi, veritabanında yer alan yüz görüntülerini kullanarak kimliklendirme ya da doğrulamadır. Bir yüz tanıma sisteminin uygulama akış şeması, Şekil 6.19’deki gibi beş adımdan oluşmaktadır. Bu adımlar;

Edinme modülü: Bu, yüz tanıma sürecinin giriş noktasıdır. Yüz görüntüsünün elde edildiği modüldür.

Yüzün tespit edilmesi: Bir görüntünün tamamı yüze ait olmadığından ilk olarak yüzün tespit edilmesi gerekmektedir.

Ön işleme modülü: Yüzün tespit edilmesini etkileyen faktörlerin çıkarılması işleminin yapıldığı adımdır.

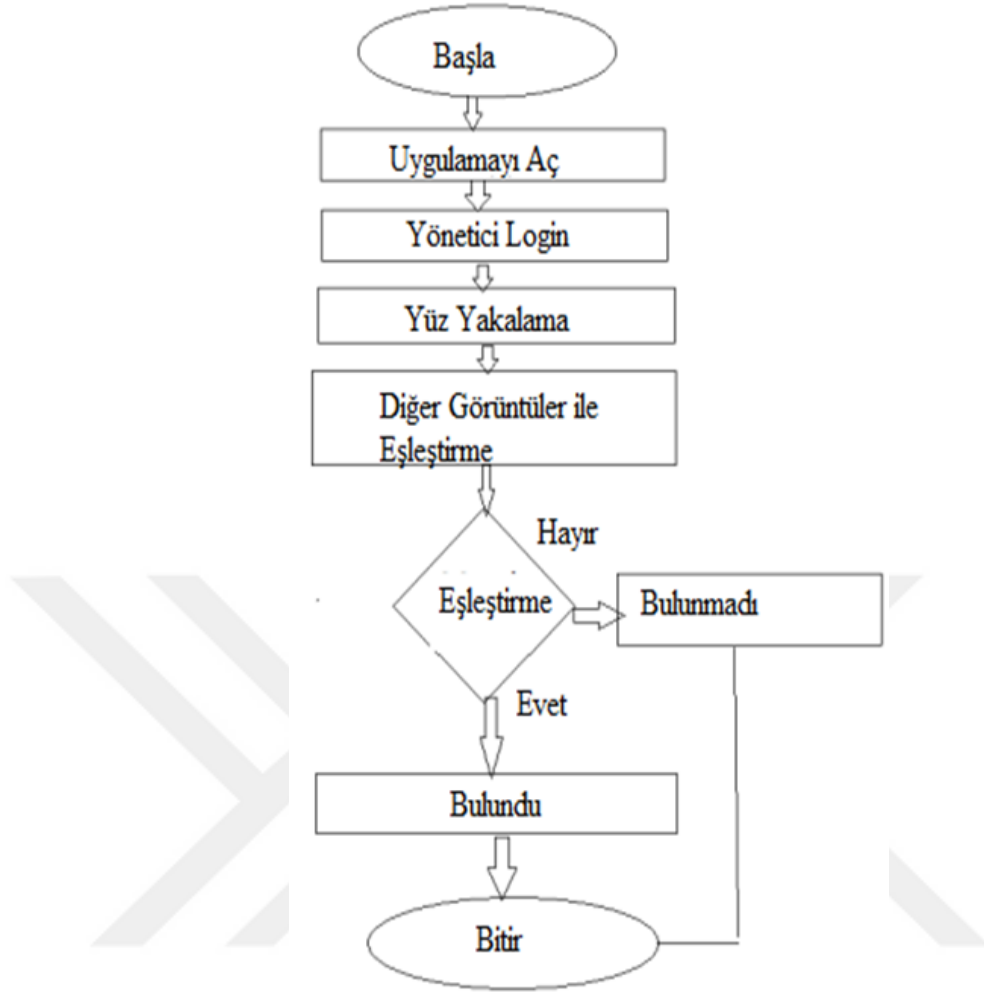
Özellik Çıkarımı: Yüzün ayırt edici özelliklerinin çıkarılması ve tanıma, doğrulama yapılmasının sağlayan adımdır.

Sınıflandırıcı Modülü: Yüz veritabanında bulunan görüntülerle yeni görüntünün karşılaştırıldığı sınıflandırıcı adımı ve görüntünün bilinen veya bilinmeyen olarak sınıflandırıldığı adımdır.

Yüz tespit adımı yüz tanıma işleminin başlangıç noktasıdır. Yüz görüntüsü manyetik bir diskte bulunan bir görüntü olabilir; bir görüntü yakalayıcı tarafından alınabilir veya bir diskte bulunan bir görüntü olabilir; bir görüntü yakalayıcı tarafından alınabilir veya bir tarayıcı yardımıyla kâğıttan taranabilir.

Sınıflandırma modülünde makine öğrenmesi temelinde görüntü sınıflandırıcıların yardımıyla, yüze ait özellikler ile yüz veri setindeki resimlerin özellikleri karşılaştırılır. Karşılaştırma sonunda resim tanınıyor veya tanınmıyor olarak sınıflandırılır.

Eğer istenirse tanınmayana resimde yüz veri setine eklenebilir. Böylece bir sonradaki tanıma işleminde bu resim tanınanlar sınıfında yer almış olur. Özellik çıkarma ve sınıflandırma modülleri, yeni bir yüzün tanınması sürecinde eldeki tanınan sınıf elemanlarının özelliklerini kullanırlar.



Şekil 6.19 Yüz tanıma sisteminin akış şeması

Bu bölümde tez çalışmasında geliştirilecek yazılımın yüz tanıma bölümü ile ilgili tasarıma yer verilmiştir. Geliştirilen uygulama 4 ana bölümden oluşmaktadır:

- ✓ Yüzün tespit edilmesi
- ✓ Görüntüdeki aydınlatma etkisinin azaltılması ve görüntünün iyileştirilmesi, gürültü azaltma, önişleme
- ✓ Poz ve ışıklandırma durumlarına göre en uygun yöntemin belirlenmesi
- ✓ Elde edilen en uygun yönteme göre kişi tanıma

Python dilinde kod yazımında Spyder arayüzü kullanılacaktır. Spyder, Anaconda Navigator'ün geliştirdiği bir ortamdır. Açık Kaynak Bilgisayar Görüşü (OpenCV), açık kaynak kodlu bir görüntü işleme kütüphanesidir. OpenCV, bilgisayarlı görü

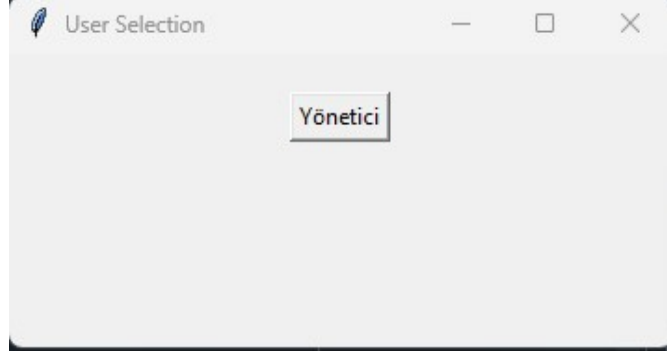
uygulamaları için ortak bir altyapı sağlamak ve ticari ürünlerde makine algısının kullanımını hızlandırmak amacıyla kurulmuştur (URL-7, 2023).

OpenCV kütüphanesi hem klasik hem de son teknoloji bilgisayarla görme ve makine öğrenimi algoritmalarından oluşan kapsamlı bir set içeren 2500'den fazla optimize edilmiş algoritmaya sahiptir. Bu algoritmalar şu şekilde sıralanabilir; Yüzleri algılamak ve tanımak, nesneleri tanımlamak, videolardaki insan eylemlerini sınıflandırmak, kamera hareketlerini izlemek, hareketli nesneleri izlemek, nesnelerin 3B modellerini çıkarmak, stereo kameralardan 3B nokta bulutları oluşturmak, yüksek çözünürlüklü bir görüntü elde etmek için görüntüleri birleştirmek için kullanılabilir. Tüm bir sahnenin görüntüsünü, bir görüntü veri tabanından benzer görüntüleri bulmayı, flaş kullanılarak çekilen görüntülerden kırmızı gözleri gidermeyi, göz hareketlerini takip etmeyi, manzarayı tanımayı ve artırılmış gerçeklikle kaplamak için işaretler oluşturmayı vb.

Kütüphane bilgi sisteminde farklı kullanıcılar (yönetici, öğrenci, katılımcı vs.) olabilir. Tez çalışmasında yönetici ünvanına sahip kullanıcı için tasarlanmıştır. Kullanıcıların sistemi nasıl kullanacaklarına ait akış şeması, Şekil 6.19'de verilmiştir.

6.4 Görüntü İşleme Kullanılarak Kütüphane Ortamında Yüz Tanıma Sisteminin Geliştirilmesi

Genel olarak yüz tanıma, sorgulanan bir yüzün veritabanındaki kayıtlı bilenen yüzlerle karşılaştırılması sonucu sorgulanan yüzün kimliğini doğrulamak veya bu yüzün kimliğini tespit etmek olarak tanımlanabilir. Yüz tanıma genel anlamda, statik bir görüntüden ya da bir video görüntüsünden bir veya daha fazla kişiyi, veri tabanında yer alan yüz görüntülerini kullanarak kimliklendirme ya da doğrulamadır. Yüz tanıma sistemine ait arayüz sırasıyla Şekil 6.20, Şekil 6.21, Şekil 6.22 ve Şekil 6.23'de verilmiştir.



Şekil 6.20 Yönetici girişi arayüzü

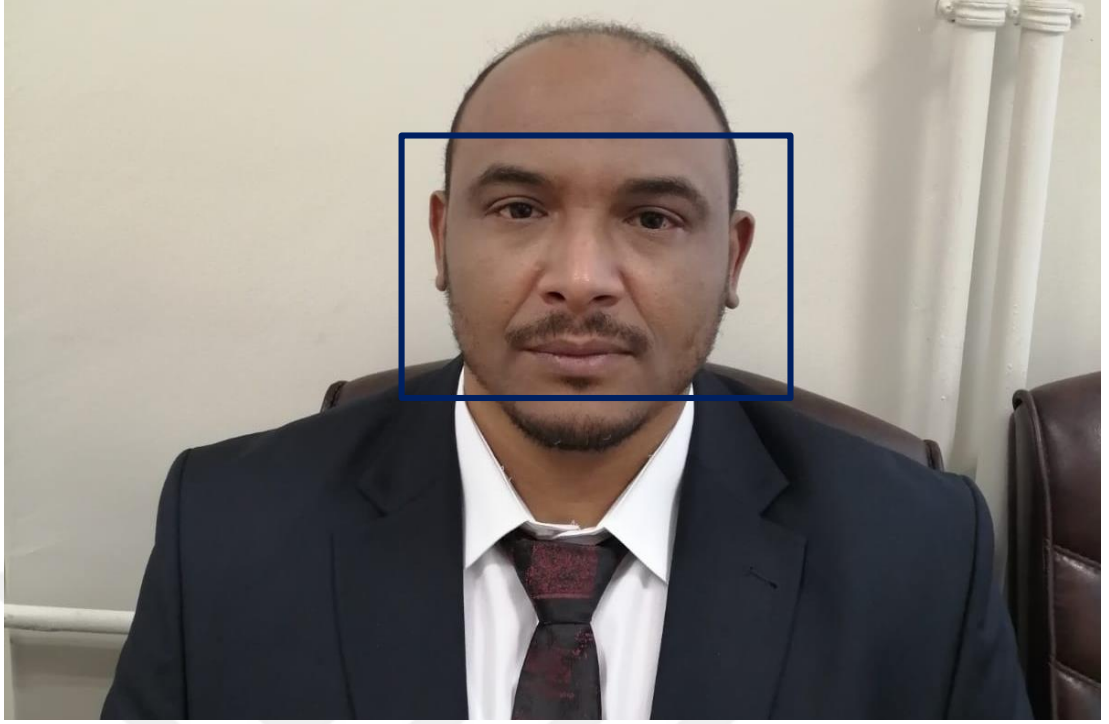
Yüz tanıma sistemine yönetici giriş arayüzü ile başlamaktadır. Tez çalışmasında kullanıcı tanımı olarak *Yönetici* tanımı yapılmıştır. Sistemde *Yönetici* kullanıcısı, sisteme giriş yaptıktan sonra ikinci aşama ise yüz görüntülerinin sisteme kayıt edilmesidir.

Yönetici butonuna basıldığında yönetici arayüzü açılır. Kullanıcı seçme arayüzü kapanır. Yönetici arayüzünde ID ve isim girilecek iki adet textbox bulunmaktadır. Örnek görüntü almak için örnek butonu, filtrelemeler için 2 adet buton ve test amacıyla bir buton bulunmaktadır



Şekil 6.21 Kullanıcı girişi arayüzü

Örnek tuşuna basıldıktan sonra kamera çalışmaya başlar ve gerçek zamanlı görüntülerden video üzerinden fotoğraflar çekilir.



Şekil 6.22 Kameranın çalışmaya başlaması

Bilgisayar kamerası ile görüntü oluşturulurken hem görüntünün alındığı ortamdaki ışıktan kaynaklı hem de kameranın elektronik donanımlarından kaynaklı gürültüler bozucu etki olarak görüntülere eklenirler.

Örnek görüntü alınması işlemi sonunda elde edilen görüntü, korrelasyon fonksiyonlarının özelliklerinden yararlanılarak gerçekleştirilen dijital filtreleme tekniği olan wiener filtreleme kullanarak bozucu etkisi olan gürültülerin giderilmiş görüntünün son hali sisteme kayıt edilir.

İkinci filtreleme adımı olarak örnek görüntü alınması işlemi sonunda elde edilen görüntü bir konvolüsyon filtreleme olan ortalama filtreleme kullanarak gürültüsü giderilmiş görüntünün son hali sisteme kayıt edilir.

Test butonuna basınca kamera açılır ve kişinin yüzünün tanınması için gerekli işlemler arka planda çalışır.



Şekil 6.23 Tahminlenen görüntü

Python, geniş ve kullanışlı birçok kütüphane alt yapısına sahiptir. Tez çalışmasında hedeflenen işlemlerin gerçekleştirilmesi için python ortamında kütüphanelerin import anahtar kelimesi ile eklenmesi gerekmektedir. Sisteme eklenen her bir kütüphane Şekil 6.24’de verilmiştir.

```
import tkinter as tk
import face_recognition
import cv2
import os
import shutil
from scipy.signal import wiener, medfilt
import numpy as np
import math
import sys
```

Şekil 6.24 Python kütüphaneleri

Sistemdeki *Yönetici* kullanıcısının rolü sisteme yeni görüntüler kayıt etmek ve bu görüntüler üzerinde sırasıyla filtreleme işlemleri ve doğruluk tahminlemesi yapmaktır.

Şekil 6.25’de kullanıcı arayüzü tanımlaması ve gerekli işlemler ile ilgili kod blokları verilmiştir.

```
class UserSelectionUI:
    def __init__(self, root):
        self.root = root
        self.root.title("User Selection")
        self.root.geometry("300x150")

        self.admin_button = tk.Button(self.root, text="Yönetici", command=self.open_admin_ui)
        self.admin_button.pack(pady=20)

    def open_admin_ui(self):
        self.root.destroy() # Kullanıcı seçimi arayüzünü kapat

        admin_ui = AdminUI()
        admin_ui.run()
```

```
class AdminUI:
    def __init__(self):
        self.root = tk.Tk()
        self.root.title("Admin Interface")

        self.id_label = tk.Label(self.root, text="ID:")
        self.id_label.pack()

        self.id_entry = tk.Entry(self.root)
        self.id_entry.pack()

        self.name_label = tk.Label(self.root, text="İsim:")
        self.name_label.pack()

        self.name_entry = tk.Entry(self.root)
        self.name_entry.pack()

        self.example_button = tk.Button(self.root, text="Örnek", command=self.take_example)
        self.example_button.pack(pady=10)

        self.wiener_button = tk.Button(self.root, text="Wiener", command=self.apply_wiener_fil)
        self.wiener_button.pack(pady=10)

        self.mean_button = tk.Button(self.root, text="Ortalama", command=self.apply_mean_filter)
        self.mean_button.pack(pady=10)

        self.test_button = tk.Button(self.root, text="Test", command=self.run_face_recognition)
        self.test_button.pack(pady=10)
```

Şekil 6.25 Yönetici arayüzü, button, textbox ve label kodları

Daha sonra wiener filtreleme yönteminin kodlarını tanımlıyoruz daha sonra admin arayüzü içerisinde fonksiyon tanımlanarak kullanılacaktır. Tanımlanan wiener kodları Şekil 6.26’da verilmiştir.

```

def apply_wiener_filter(self):
    # Wiener filtresi uygulama işlemi
    faces_dir = "faces"
    wiener_dir = "wiener"

    if not os.path.exists(wiener_dir):
        os.makedirs(wiener_dir)

    for image in os.listdir(faces_dir):
        face_image = cv2.imread(os.path.join(faces_dir, image))
        face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)

        filtered_image = wiener(face_image, mysized=(155, 155))

        id = image.split(".")[1]
        name = image.split(".")[0]
        filename = f"wiener/{name}.{id}.jpg" # Wiener filtresi uygulanarak kaydedilecek konum
        cv2.imwrite(filename, filtered_image)

    print("Wiener filtresi uygulandı ve kaydedildi.")

```

Şekil 6.26 Wiener kodları

Şekil 6.27’da ortalama filtreleme yönteminin kullanılması için gerekli kodlar gösterilmiştir.

```

def apply_mean_filter(self):
    # Ortalama filtresi uygulama işlemi
    faces_dir = "faces"
    mean_dir = "mean"

    if not os.path.exists(mean_dir):
        os.makedirs(mean_dir)

    for image in os.listdir(faces_dir):
        face_image = cv2.imread(os.path.join(faces_dir, image))
        face_image = cv2.cvtColor(face_image, cv2.COLOR_BGR2GRAY)

        filtered_image = cv2.blur(face_image, (155, 155))

        id = image.split(".")[1]
        name = image.split(".")[0]
        filename = f"mean/{name}.{id}.jpg" # Ortalama filtresi uygulanarak kaydedilecek konum
        cv2.imwrite(filename, filtered_image)

    print("Ortalama filtresi uygulandı ve kaydedildi.")

```

Şekil 6.27 Ortalama kodları

OpenCV kütüphanesi ile Python programımızın insan görüntüleri üzerindeki yüzleri tespit etmesini ve tanımlamasını sağlamak için yazılan kodlar, Şekil 6.28’de verilmiştir. OpenCV, görüntü işleme ve bilgisayarla görme görevlerini gerçekleştirmenize izin veren bir Python kitaplığıdır.

```

class FaceRecognition:
    face_locations = []
    face_encodings = []
    face_names = []
    known_face_encodings = []
    known_face_names = []
    process_current_frame = True

    def __init__(self):
        self.encode_faces()

    def encode_faces(self):
        for image in os.listdir('faces'):
            face_image = face_recognition.load_image_file(f"faces/{image}")
            face_encoding = face_recognition.face_encodings(face_image)[0]

            self.known_face_encodings.append(face_encoding)
            self.known_face_names.append(image)
            print(self.known_face_names)

    def run_recognition(self):
        video_capture = cv2.VideoCapture(0)

        if not video_capture.isOpened():
            sys.exit('Video kaynağı bulunamadı...')

        while True:
            ret, frame = video_capture.read()

            # Only process every other frame of video to save time
            if self.process_current_frame:
                # Resize frame of video to 1/4 size for faster face recognition processing
                small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)

                # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
                rgb_small_frame = small_frame[:, :, ::-1]

                # Find all the faces and face encodings in the current frame of video
                self.face_locations = face_recognition.face_locations(rgb_small_frame)
                self.face_encodings = face_recognition.face_encodings(rgb_small_frame, self.face_locations)

                self.face_names = []
                for face_encoding in self.face_encodings:
                    # See if the face is a match for the known face(s)
                    matches = face_recognition.compare_faces(self.known_face_encodings, face_encoding)
                    name = "Unknown"
                    confidence = '???'

                    # Calculate the shortest distance to face
                    face_distances = face_recognition.face_distance(self.known_face_encodings, face_encoding)

                    best_match_index = np.argmin(face_distances)
                    if matches[best_match_index]:
                        name = self.known_face_names[best_match_index]
                        confidence = face_confidence(face_distances[best_match_index])

                    self.face_names.append(f'{name} ({confidence})')

                self.process_current_frame = not self.process_current_frame

            # Display the results
            for (top, right, bottom, left), name in zip(self.face_locations, self.face_names):
                # Scale back up face locations since the frame we detected in was scaled to 1/4 size
                top *= 4
                right *= 4
                bottom *= 4
                left *= 4

                # Draw a box around the face
                cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

                # Draw a label with a name below the face
                cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
                font = cv2.FONT_HERSHEY_DUPLEX
                cv2.putText(frame, name, (left + 6, bottom - 6), font, 0.8, (255, 255, 255), 1)

            # Display the resulting image
            cv2.imshow('Video', frame)

            # Hit 'q' on the keyboard to quit!
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break

        # Release handle to the webcam
        video_capture.release()
        cv2.destroyAllWindows()

```

Şekil 6.28 Yüz tanımlama kodları

Şekil 6.29’de benzerlik değerinin alınması için gerekli matematiksel hesaplanmaların yapıldığı kodlar gösterilmiştir. Deneysel çalışmada, test doğruluk oranı genel olarak %80’nin üzerinde değer elde edilerek yüz tanıma için ümit verisi sonuçlar elde edilmiştir.

```
def face_confidence(face_distance, face_match_threshold=0.4):  
    range = (1.0 - face_match_threshold)  
    linear_val = (1.0 - face_distance) / (range * 2.0)  
  
    if face_distance > face_match_threshold:  
        return str(round(linear_val * 100, 2)) + '%'  
    else:  
        value = (linear_val + ((1.0 - linear_val) * math.pow((linear_val - 0.5) * 2, 0.2))) * 100  
        return str(round(value, 2)) + '%'
```

Şekil 6.29 Benzerlik tanımlama kodları

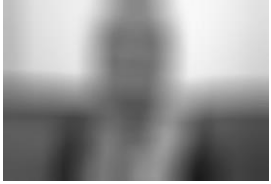
Kodların kayıt edildi, görüntülerin elde edilerek saklandığı ve her bir filtreleme işlemi sonucunda elde edilen görüntülerden ayıklanmış görüntülerin saklandığı vb. klasör yapısı işe Şekil 6.30’de verilmiştir.

data	23.05.2023 14:11	Dosya klasörü	
ImagesUnknown	28.05.2023 13:58	Dosya klasörü	
onisleme1	28.05.2023 13:57	Dosya klasörü	
onisleme2	28.05.2023 13:57	Dosya klasörü	
TrainingImage	28.05.2023 13:57	Dosya klasörü	
TrainingImageLabel	25.12.2022 22:47	Dosya klasörü	
UserDetails	28.05.2023 13:38	Dosya klasörü	
train.py	20.05.2023 17:16	Python File	14 KB

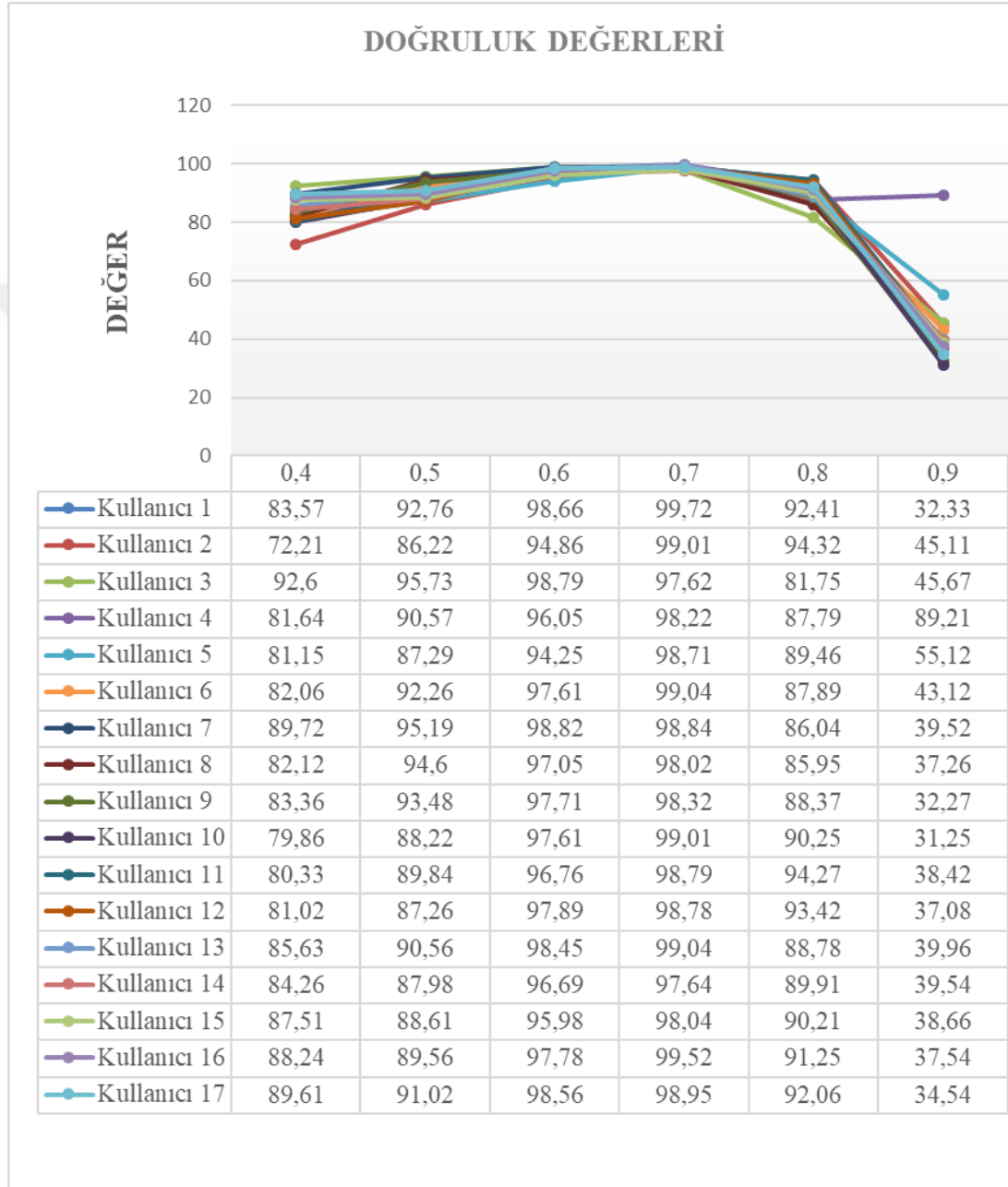
Şekil 6.30 Genel dosya yapısı

Tablo 6.7’de, yüz tanıma sisteminde kullanılan wiener ve ortalama filtreleme görüntülerinin orijinal görüntüler ile karşılaştırılmasını vermiştir.

Tablo 6.7 Görüntülerin ön işleme adımları

Orijinal Görüntü	Wiener Filtreleme	Ortalama Filtreme
		
		
		
		
		
		

Tez çalışmasındaki yüz tanıma sisteminde farklı eşik değerlerine göre doğruluk oranları hesaplanmıştır. Toplam 17 kullanıcı ile sistem test edilmiştir. Şekil 6.31’de gösterilmiştir. En yüksek doğruluk değeri, eşik değeri 0,7 ile %99,72 olarak elde edilmiştir.



Şekil 6.31 Doğruluk değerleri

7. SONUÇLAR VE ÖNERİLER

Sahip olduğumuz manyetik kart, kimlik kartı, çip vb. veya bildiğimiz şifre, kodlama vb. dayalı tanımlama yöntemleri büyük güvenilirlik sorunlarına neden olur. Tüm bu problemler, bir tanımlama ve kimlik doğrulama aracı olarak biyometrinin daha da geliştirilmesini teşvik etmiştir. Parmak izi kullanımı ve yüz tanımlama/doğrulama biyometrisinin en eski ve en başarılı yöntemlerinden biridir. Bu nedenle hem insan gücünden hem de zamandan tasarruf etmek için mevcut kütüphane sistemini otomatikleştirmek gerekir. Akıllı kütüphaneler, teknolojinin en son kullanımını içeren otomatikleştirilmiş ve gelişmiş bir kütüphane sistemleridir. Bu sistemler, kütüphane kullanımını yükseltir ve insanları kütüphaneye daha fazla erişmeye teşvik eder.

Bu tez çalışması, kütüphane kullanıcı girişini otomatikleştirerek kullanıcılar için daha etkileşimli ve kolay hale getirme ihtiyacına dayalı olarak geliştirilmiştir. Geliştirilen sistemler, yüksek doğruluk oranına sahip yüz ve parmak izi tanıma modellerinden oluşmaktadır. Bu nedenle, kullanıcıların kimlik kartı taşımasına gerek yoktur ve her seferinde kullanıcı adı ve şifreyi hatırlamaya gerek yoktur. Sistemler akademik kütüphaneler, okul kütüphaneleri ve halk kütüphaneleri için çok uygun olabilir.

Parmak izi, insan parmağının yüzeyindeki çıkıntıların ve vadilerin grafiksel bir modelidir. Parmak izleri benzersizliği ve kalıcılığı nedeniyle, kişisel tanımlama için kullanılabilecek en güvenilir insan özellikleri arasındadır. Bir otomatik parmak izi tanımlama sisteminin performansı büyük ölçüde, izlerin varlığı, parmak ile edinim sensörü arasındaki basınçtaki değişimler, aşınmış eserler ve edinim işlemi sırasında çevresel koşullar gibi çeşitli faktörlerden etkilenebilen parmak izi görüntü kalitesine dayanır. Etkili bir parmak izi tanımlama sistemi oluşturmak için etkili bir geliştirme algoritması gereklidir.

Tez çalışmasında tanımlama modunda çalışan ve parmak izi algılama, görüntü iyileştirme, özellik çıkarma ve eşleştirme yapabilen eksiksiz bir biyometrik sistemi temsil eden otomatik parmak izi tanıma sistemi gerçekleştirilmiştir. FVC 2000, FVC

2002 ve FVC 2004 veri setleri üzerinde doğru sınıflandırılmış yüksek kaliteli parmak izlerinin %98,83'lük genel tanıma oranı elde edilmiştir.

Tez çalışmasında görsel görüntüleri analiz etmek için en yaygın olarak uygulanan bir yapay sinir ağı sınıfı olan derin öğrenmede CNN uygulaması gerçekleştirilmiştir. Deneysel sonuçlara göre %80 doğruluk elde edilmiştir.

Yüz tanıma sisteminin avantajları arasında daha hızlı işleme, kimlik otomasyonu, mahremiyet ihlali, büyük veri depolama, en iyi sonuçlar, gelişmiş güvenlik, kütüphanelerde gerçek zamanlı yüz tanıma, şirket ofislerindeki çalışanlar, akıllı telefon kilidini açma ve daha birçok günlük yaşamda daha fazla yer alır.

Yüz tanıma sistemi, bir kişiyi dijital bir görüntüden veya bir video kaynağından bir video karesinden tanımlayabilen veya doğrulayabilen bir teknolojidir. Yüz tanıma sistemlerinin çalıştığı birçok yöntem vardır, ancak genel olarak, belirli bir görüntüden seçilen yüz özelliklerini bir veri tabanındaki yüzlerle karşılaştırarak çalışırlar. Ayrıca, kişinin yüz dokularına ve şekline dayalı kalıpları analiz ederek bir kişiyi benzersiz bir şekilde tanımlayabilen biyometrik yapay zeka tabanlı bir uygulama olarak da tanımlanmaktadır.

Bu tez çalışmasında, yüz tanıma için OpenCV uygulamasını ve bu uygulamada kullanılan sınıflandırıcı sunulmuştur. OpenCV'nin yüz tanımadaki önemli rolü, OpenCV'de yüz tanıma için hangi algoritmanın kullanılabileceği üzerinde durulmuştur. Tez çalışmasında yüz tanıma gerçek zamanlı olarak gerçekleştirilmiştir. Sınıflandırma doğruluğu artırmak için iki adet görüntü filtreleme işlemi uygulanmıştır. Gerçek zamanlı görüntüler üzerinde testinde %99 gibi yüksek bir oranda doğruluk ortaya konulmuştur.

Tez çalışmasında geliştirilen parmak izi tanımlama ve yüz tanımlama olmak üzere iki farklı gelişmiş sistem, Libya'daki üniversite kütüphanesinde kullanımı ve kültürel, bilimsel hayata yüksek katkıda bulunulacaktır.

KAYNAKLAR

- Albregtsen F., (2008). Statistical Texture Measures Computed from Gray Level Coocurrence Matrices, Image Processing Laboratory Department of Informatics University of Oslo, 1-14.
- Alemna, A. A., (2000). Libraries, information and society. Accra Ghana Universites Press.
- Alsarrar, M. R. M., & Gültepe, Y. (2016). A comparative study of use of the internet and the library as sources of information by graduate students in the Libyan Academy. *International Journal of Science and Research*, ISSN: 2319-7064.
- Alsarrar, M. R. M., & Gültepe, Y. (2018). A system to identity human entering the place by fingerprint based on artificial neural network. *IJESC*, 11(5), 27920-27924.
- Ancin, M., Pindado, E., & Sanchez, M. (2022). New trends in the global digital transformation process of the agri-food sector: an exploratory study based on twitter. *Agricultural Systems*, 203.
- Batty, C. D. (1971). Introduction to the Dewey Decimal Classification: 18th edition. Bingley Publisher, 108 pages.
- Belhadj, F. (2017). Biometric system for identification and authentication. Computer Vision and Pattern Recognition. Ecole Nationale Supérieure En Informatique Alger, Doctoral Dissertation.
- Bezdan, T., & and Džakula, N. B. (2019). Convolutional Neural Network Layers and Architectures. International Scientific Conference on Information Technology and Data Related Research, 445-451.
- Bhatt, D., Patel, C., Talsania, H., Patel, J., Vaghela, R., Pandya, S., Modi, K., & Ghayvat, H. (2021). Cnn variants for computer vision: History, architecture, application, challenges and future scope, *Electronics*, 10(20).
- Bhattacharya, S., & Mali, K. (2012). Fingerprint recognition using minutiate extraction method. International Conference on Emerging Technologies.
- Chang, J.-H., & Fan, K.-C. (2002). A new model for fingerprint classification by ridge distribution sequences. *Pattern Recognition*, 35(6), 1209-1223.
- Chatterjee, S. (2022). What is feature extraction? Feature extraction in image processing. Great Learning.
- Da'san, M., Alqudah, A., & Debeir, O. (2015) Face detection using Viola and Jones method and neural networks. Proceeding of IEEE International Conference on Information and Communication Technology Research, 40-43.

- Dastres, R., & Soori, M. (2021). Artificial neural networks systems. *International Journal of Imaging and Robotics*, 21(2), 13-25.
- Du, K.-L., Leung, C.-D., Mow, W. H., & Swamy, M. N. S. (2022). Perceptron: Learning, generalization, model selection, fault tolerance, and role in the deep learning era. *Mathematics*, 10, 4730.
- Eleyan, A., & Demirel, H. (2011). Co-occurrence matrix and its statistical features as a new approach for face recognition. *Turkish Journal of Electrical Engineering and Computer Sciences*, 19(1).
- Farfade, S. S., Saberian, M., & Li, L. J. (2015) Multi-view face detection using deep convolutional neural networks. *Proceeding of The International Conference on Multimedia Retrieval*, 1-8.
- Feraud, R., Bernier, O. J., Villet, J. E., & Collobert, M. (2001) A fast and accurate face detector based on neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 42-53.
- Fu, G., Jin, Y., Sun, S., Yuan, Z., & Butler, D. (2022). The role of deep learning in urban water management: A critical review. *Water Research*, 223.
- Gültepe, Y., & Duru, A. E. (2018). Daily SO₂ air pollution prediction with the use of artificial neural network models. *International Journal of computer Applications*, 181(34), 36-40.
- Gültepe, Y., & Duru, A. E. (2019). Using convolutional neural networks for handwritten digit recognition. *International Conference on Data Science, Machine Learning and Statistic*.
- Jain, A., Prabhakar, S., & Pankanti, S. (2001). Matching and classification: a case study in fingerprint domain. *Proceedings-Indian Natl. Sci. Acad. PART A*, 67(2), 223-242.
- Jain, A. K., & Minut, S. (2002). Hierarchical kernel fitting for fingerprint classification and alignment. *Object recognition supported by user interaction for service robots*, 2, 469-473.
- Jain, A. K., Feng, J., & Nandakumar, K. (2010). Fingerprint matching. *Computer* 43(2), 36-44.
- Jumarlis, M., Mirfan, M., & Manga, A. R. (2022). Classification of coffee bean defects using gray-level co-occurrence matrix and k-nearest neighbor. *ILKOM Journal Ilmiah*, 14(1), 1-9.
- Karu, K., & Jain, A. K. (1996). Fingerprint classification. *Pattern Recognition*, 29(3), 389-404. [https://doi.org/10.1016/0031-3203\(95\)00106-9](https://doi.org/10.1016/0031-3203(95)00106-9).
- Kazubek, M. (2003). Wavelet domain image de-noising by thresholding and Wiener filtering. *Signal Letters IEEE*, 10(11).

- Ke, Q., Liu, J., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2018). Computer vision for human-machine interaction. *Computer vision for human-machine interaction, Computer Vision For Assistive Healthcare*, 127-145.
- Khan, M. M., Hossain, S., Mozumdar, P., Akter, S., & Ashique, R. H. (2022). A review on machine learning and deep learning for various antenna design applications. *Heliyon*, 8(4).
- Kramer, M. A. (1991) Nonlinear principal component analysis using auto associative neural networks. *American Institute of Chemical Engineers Journals*, 37(2), 233-243.
- Kumar, A., Kaur, A., & Kumar, M. (2019). Face detection techniques: a review. *Artificial Intelligence Review*, 52, 927-948.
- Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2021). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks Learning Systems*, 1-21.
- Liao, S., Jain, A. K., & Li, S. Z. (2016) A fast and accurate unconstrained face detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2), 211-223.
- Lin, S. H., Kung, S. Y., & Lin, L. J. (1997) Face recognition/detection by probabilistic decision-based neural network. *IEEE Trans Neural Netw* 8(1):114–132.
- López, O. A. M., López, A. M., & Crossa, J. (2022). Fundamentals of artificial neural networks and deep learning. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*, Springer, 379-425.
- Lyra, M. (2011). MATLAB as a tool in nuclear medicine image processing. *MATLAB-A Ubiquitous Tool for the Practical Engineer*.
- Maio, D., Maltoni, D., Cappelli, R., Wayman, J. L., & Jain, A. K. (2002). FVC2000: Fingerprint verification competition. *IEEE Transactions on PAMI*, 24, 402-12.
- Mali, K., & Bhattacharya, S. (2011). Fingerprint recognition using global and local structures. *International Journal of Advanced Trends in Computer Science and Engineering*.
- Maltoni, D., Maio, D., Jain, A. K., & Prabhakar, S. (2009). Handbook of fingerprint Recognition. Springer-Verlag London Limited.
- Marasco, E., & Ross, A. (2014). A survey on antispooofing schemes for fingerprint recognition systems. *ACM Computing Surveys*, 47(2), 1-36.
- Minaee, S., Azimi, E., & Abdolrashidi, A. (2019). Fingernet: Pushing the limits of fingerprint recognition using convolutional neural network. *arXiv Prepr. arXiv1907.12956*.
- Mishra, R. K., Reddy, G. Y. S., & Pathak, H. (2021). The Understanding of Deep

Learning: A Comprehensive Review. Mathematics Problems in Engineering.

- Mohamed, S. M., & Nyongesa, H. (2002). Automatic fingerprint classification system using fuzzy neural techniques. *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02*. 358-362.
- Mohammed, Z. E. (2019). Using the artificial neural networks for prediction and validating solar radiation. *Journal of the Egyptian Mathematical Society*, 27(47).
- Molchanov, P., Mallya, A., Tyree, Frosio, I., & Kautz, J. (2019). Importance estimation for neural network pruning. *IEEE Conference on Computer Vision and Pattern Recognition*, 11264-11272.
- Nagaty, K. A. (2001). Fingerprints classification using artificial neural networks: a combined structural and statistical approach. *Neural Networks*, 14(9), 1293-1305.
- Permanasari, Y., Ruchjana, B. N., Hadi, S., & Rejito, J. (2022). Innovative Region Convolutional Neural Network Algorithm for Object Identification. *Journal Open Innovation: Technology, Market, and Complexity*, 8(182).
- Prabhakar, S. (2001). Fingerprint classification and matching using a filterbank. Michigan State University. Department of Computer Science & Engineering.
- Propp, M., & Samal, A. (1992) Artificial neural network architectures for human face detection. *Proceeding of artificial neural networks in engineering*, 2, 535-540.
- Qin, Z., Yu, F., Liu, C., & Chen, X. (2018) How convolutional neural network see the World-A survey of convolutional neural network visualization methods. *American Institute of Mathematical Sciences*, 1, 149-180.
- Rani, N. (2017). Image processing techniques: A review. *Journal on Today's Ideas-Tomorrow's Technologies*, 5(1), 40-49.
- Rzemyk, T. J. (2017). Biometrics in the criminal justice system and society today. *Effective Physical Security (Fifth Edition)*, Chapter 10, 249-254.
- Senior, A. (2001). A combination fingerprint classifier. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(10), 1165-1174.
- Shah, S., & Sastry, P. S. (2004) "Fingerprint classification using a feedback-based line detector", *IEEE Trans. Syst. Man, Cybern. Part B*, 34(1), 85-94.
- Shen, C. (2018). A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists, *Water Resources Research*, 54(11), 8558-8593.

- Shetti, P. P., & Patil, A. P. (2017). Performance comparison of mean, median and Wiener filter in MRI image de-noising. *International Journal for Research Trends and Innovation*, 2(6).
- Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6.
- Siddeq, M. M., & Yaba, S. P. (2009). Using Discrete Wavelet Transform and Wiener filter for Image. *Wasit Journal for Science & Medicine*, 2(2), 18-30.
- Tan, X., Bhanu, B., & Lin, Y. (2005). Fingerprint classification based on learned features. *IEEE Transactions on Systems Man and Cybernetics Part C (Applications and Reviews)*, 35(3), 287-300.
- Thamotharan, B., Menaka, M., Vaidyanathan, S. & Ravikumar, S. (2012). Survey on image processing in the field of deNoising techniques and edge detection techniques on radiographic images. *Journal of Theoretical and Applied Information Technology*, 41(1).
- Topaloglu, N. (2013). Revised: finger print classification based on gray-level fuzzy clustering co-occurrence matrix. *Energy Education Science and Technology Part A: Energy Science and Research*, 31(3), 1307-1316.
- Veronez, M. R., Souza, S. F., Matsuoka, M. T., Reinhardt, A., & Silva, R. M. (2011). Regional mapping of the geoid using GNSS (GPS) measurements and an artificial neural network. *Remote Sensing*, 3, 668-683.
- Walczak, S., Cerpa, N. (2003). Artificial neural networks. *Encyclopedia of Physical Science and Technology*, 631-645.
- Wan, W., Kubendran, R., Schaefer, C. ... & Cauwenberghs, G. (2022). A compute-in-memory chip based on resistive random-access memory. *Nature*, 608, 504–512.
- Wang, Z. J., Turko, R., Shaikh, O., Park, H., Das, N., Hohman, F., Kahng, M., Chau, D., H. P. (2021). CNN explainer: learning convolutional neural networks with interactive visualization. *IEEE Transactions on Visualization and Computer Graphics*, 27(2), 1396-1406.
- Weaver, H. J. (1989). Theory of discrete and continuous Fourier analysis. *Wiley-Interscience*, 1st Edition.
- Wu, M., Liu, X., Gui, N., Yang, X., Tu, J., Jiang, S., & Zhao, Q. (2022). Prediction of remaining time and time interval of pebbles in pebble bed HTGRs aided by CNN via DEM datasets. *Nuclear Engineering Technology*, 2022.
- Yao, Y., Marcialis, G. L., Pontil, M., Frasconi, P., & Roli, F. (2003). Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines. *Pattern Recognition*, 36(2), 397-406.

- Yao, Y., Frasconi, P., & Pontil, M. (2001). Fingerprint classification with combinations of support vector machines. *International conference on audio- and video-based biometric person authentication*, 253-258.
- Yazdi, M., & Gheysari, K. (2008). A new approach for the fingerprint classification based on gray-level co-occurrence matrix. *International Journal of Computer and Information Science and Engineering*, 2(3).
- Yelmanova, E., & Romanyshyn, R. (2017). Histogram-based method for image contrast enhancement. *14th International Conference The Experience of Designing and Application of CAD Systems in Microelectronics*, 165-169.
- Zarbega, T. S. A., & Gültepe, Y. (2021). Detection and segmentation of nucleus cells in histological images with deep neural networks. *IOSR Journal of computer Engineering*, 23(2), 33-41.
- Zarbega, T. S. A., & Gültepe, Y. (2020). Semantic segmentation of cell nuclei in breast cancer using convolutional neural network. *International Journal of Computer Applications*, 176(22), 1-8.
- Zhao, S., Ge, D., Zhao, J., & Xiang, W. (2021). Fingerprint pre-processing and feature engineering to enhance agricultural products categorization. *Future Generation Computer Systems*, 125, 944-948.
- Zhang, L., Zhang, L., Du, B., You, J., & Tao, D. (2019). Hyperspectral image unsupervised classification by robust manifold matrix factorization. *Information Sciences*, 485, 154-169.
- Zhu, M., Wang, J., Yang, X., ... & Ye, L. (2022). A review of the applicaiton of machine learning in water quality evaluation. *Eco-Environment & Health*, 1(2), 107-116.
- URL-1. University of Benghazi (2022). <https://uob.edu.ly/>, Erişim Tarihi: 15/01/2022.
- URL-2. FRVT Part 2: Identification (2018). <https://nvlpubs.nist.gov/nistpubs/ir/2018/NIST.IR.8238.pdf>, Erişim Tarihi: 15/02/2020.
- URL-3. Fingerprint Verification Competition (2010). <http://bias.csr.unibo.it/fvc2000/>, Erişim Tarihi: 01/03/2020.
- URL-4. Fingerprint Verification Competititon (2010). <http://bias.csr.unibo.it/fvc2002/>, Erişim Tarihi: 01/05/2020.
- URL-5. FVC2004: Third Fingerprint Verification Competition (2010). <http://bias.csr.unibo.it/fvc2004/databases.asp>, Erişim Tarihi: 10/02/2021.
- URL-6. MATLAB (2023). <https://www.mathworks.com/products/matlab.html>, Erişim Tarihi: 12/01/2021.
- URL-7. OpenCV (2023). <https://opencv.org/about/>, Erişim Tarihi: 12/01/2022.