T.C.
HALİÇ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI
BİLGİSAYAR MÜHENDİSLİĞİ PROGRAMI

MOBİL ANDROID ORTAMINDA PARMAK İZİ TANIMA VE KİMLİK DOĞRULAMA SİSTEMİNİN GELİŞTİRİLMESİ

YÜKSEK LİSANS TEZİ

Hazırlayan
Süleyman ÇINAR

Danışman
Prof. Dr. Muhammet KÖKSAL

İstanbul - 2014

T.C.

HALİÇ ÜNİVERSİTESİ

FEN BİLİMLERİ ENSTİTÜSÜ MÜDÜRLÜĞÜNE

Bilgisayar Mühendisliği Anabilim Dalı Bilgisayar Mühendisliği Programı Tezli Yüksek Lisans öğrencisi **Süleyman ÇINAR** tarafından hazırlanan **"Mobil Android Ortamında Parmak İzi Tanıma ve Kimlik Doğrulama Sisteminin Geliştirilmesi"** adlı bu çalışma jürimizce Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Tarihi : 05.05.2014

(Jüri Üyesinin Ünvanı , Adı , Soyadı ve Kurumu) :	İmzası :
Jüri Üyesi: Prof.Dr.Muhammet KÖKSAL **Dan.-HAL.Üniv. Elektrik-Elektronik Müh.ABD Öğr.Üyesi**	.....
Jüri Üyesi :Prof.Dr.Mübariz EMİNLİ **HAL.Üniv. Bilgisayar Müh. ABD Öğr.Üyesi**	.....
Jüri Üyesi : Yrd.Doç.Dr.Soner ÖZGÜNEL **HAL.Üniv. Elektrik-Elektronik Müh.ABD Öğr.Üyesi**	.....
Jüri Üyesi : Prof.Dr.Nariman ŞERİFOĞLU **HAL.Üniv. Elektrik-Elektronik Müh.ABD Öğr.Üyesi (Yedek)**	.....
Jüri Üyesi : Yrd.Doç.Dr.Ulviye HACIZADE **HAL.Üniv. Bilgisayar Müh.ABD Öğr.Üyesi (Yedek)**	.....

ÖNSÖZ

Yüksek lisans eğitimim ve tez çalışmamın tamamlanması süresince büyük bir gayret ve özveriyle çalışmamı takip eden, gösterdiği sabır ve hoşgörüyle bana destek olan tez danışmanım Sayın Prof. Dr. Muhammet KÖKSAL'a çok teşekkür ederim.

Bu tez çalışması süresince yardımlarını esirgemeyen Sayın Yrd. Doç. Dr. Oğuz KARAN'a ve yüksek lisans eğitimim kapsamında destekleriyle yanımda olan Sayın Yrd. Doç. Dr. Sabri Serkan GÜLLÜOĞLU'na teşekkürlerimi sunarım.

Son olarak eğitim hayatım boyunca bana destek olan ve verdiğim her kararın arkasında durarak beni bu günlere getiren sevgili anne ve babama sonsuz teşekkür ederim.

İstanbul, 2014 Süleyman ÇINAR

İÇİNDEKİLER

Sayfa No.

KISALTMALAR ..... **III**
ŞEKİLLER ..... **V**
TABLOLAR ..... **VI**
ÖZET ..... **VII**
SUMMARY ..... **VIII**
1. GİRİŞ **1**
2. BİYOMETRİK SİSTEMLER **5**
2.1. Biyometri **5**
2.2. Biyometrik Sistemlerde Kullanılan Yöntemler **6**
2.2.1. Parmak İzi Tanıma 7
2.2.1.1. Parmak İzinin Özellikleri 7
2.2.3. Yüz Tanıma 9
2.2.4. İris Tanıma 10
2.2.5. Retina Tanıma 11
2.2.6. Damar Tanıma 11
2.2.7. El Yazısı Tanıma 12
2.2.8. DNA Tanıma 12
2.2.9. El Geometrisi Tanıma 12
2.2.10. İmza Tanıma 13
2.3. Biyometrik Sistemlerin Uygulama Alanları **13**
2.4. Biyometrik Sistemlerin Standartları **14**
2.5. Biyometrik Sistemlerin Performansı **15**
3. ANDROID PLATFORM **17**
3.1. Android İşletim Sistemi **17**
3.2. Android Kavramları **18**
3.2.1. Çalışma Durumu 19
3.2.2. Duraklatma Durumu 19
3.2.3. Durdurma Durumu 20
3.2.4. İmha Durumu 20
3.3. Android Yaşam Döngüsü Aşamaları **20**
3.3.1. Android Toplam Yaşam Süresi 20
3.3.2. Görülebilen Yaşam Süresi 21
3.3.3. Ön Plandaki Yaşam Süresi 21
3.4. Android İşletim Sistemi Mimarisi **21**
3.4.1. Aktiviteler 21
3.4.2. Hizmetler 22
3.4.3. Radyo Alıcıları 22
3.4.4. İçerik Sağlayıcısı 22

4. KULLANILAN TEKNOLOJİ VE YÖNTEMLER **24**
4.1. Minutiae Özellik Noktalarının Elde Edilmesi (Minutiae Extraction) **28**
4.2. Yönelim Alanlarının Tahmin Edilmesi **29**
4.3. İz Algılama (Ridge Detection) **31**
4.4. Özellik Noktalarının Algılanması (Minutiae Detection) **33**
4.5. Minutiae Özelliklerinin Eşleştirilmesi (Minutiae Matching) **36**
4.6. Nokta Desenlerinin Hizalanması **37**
4.7. Sıraya Dizilen Nokta Desenlerinin Eşleştirilmesi **39**
4.8. Parmak İzi Okuma Teknolojileri **45**
4.9. Parmak İzi Okuma Yöntemi **45**
4.10. Optik Sensörler **46**
4.11. Java Nesne Serileştirme (Java Object Serialization) **47**
4.11.1. Nesne Serileştirmenin Temelleri 48
5. MOBİL ANDROID ORTAMINDA GELİŞTİRİLEN PARMAK İZİ TANIMA VE KİMLİK DOĞRULMA SİSTEMİ **49**
5.1. Geliştirilen Kullanıcı Arayüzü Uygulaması **49**
5.2. Sistemin Kullanım Durumu **51**
5.3. Sistemin Sınıf Yapısı **56**
5.4. Mobil Android Parmak İzi Tanıma Sisteminin Arayüz Mimarisi **58**
5.4.1. Sistemde Kullanılan Secugen Fonksiyonları 58
5.4.1.1. Sistemin Eşleşme Başarı Puanı 61
5.5. Mobil Android Parmak İzi Tanıma Sisteminin Kullanıcı Arayüzü **63**
5.5.1. Sisteme Parmak İzinin Tanımlanması 65
5.5.1.1. Parmak İzi Tanımlama Aşamasında Parmak İzinin Elde Edilmesi 67
5.5.1.2. Parmak İzi Tanımlama Aşamasında Kimlik Bilgilerinin Girilmesi 68
5.5.1.3. Kimlik Bilgilerinin ve Parmak İzi Özelliklerinin Kaydedilmesi 69
5.5.1.4. Parmak İzi Tanımlama Aşamasının UML Aktivite Şeması 73
5.5.2. Sistemde Kimlik Doğrulamanın Yapılması 74
5.5.2.1. Kimlik Doğrulama Aşamasında Parmak İzinin Elde Edilmesi 76
5.5.2.2. Kimlik Doğrulama Aşamasında Minutiae Algoritmasının Kullanılması 82
5.5.2.3. Kimlik Doğrulama Aşamasının UML Aktivite Şeması 83
5.6. Sistemin Performans Ölçütlerinin Değerlendirilmesi **84**
5.6.1. Sonuçların Analizi 85
6. SONUÇLAR VE ÖNERİLER **87**
7. KAYNAKLAR **89**
8. ÖZGEÇMİŞ **92**

KISALTMALAR

AFIS	: Automatic Fingerprint Identification Systems (Otomatik Parmakizi Tanıma Sistemleri)
ANSI	: American National Standards Institute (Amerikan Ulusal Standartlar Enstitüsü)
API	: Application Programming Interfaces (Uygulama Programlama Arayüzleri)
ATM	: Automatic Teller Machine (Otomatik Banka Veznesi)
CCD	: Charge Coupled Device (Yüklenme İliştirilmiş Araç)
CJIS	: Criminal Justice Information Services (Ceza Hukuku Bilgi Hizmetleri)
CMOS	: Complementary Metal Oxide Semiconductor (Bütünleyici Metal Oksit Yarıiletken)
DNA	: Deoksiribo Nükleik Asit
DPI	: Dots Per Inch (İnç Başına Nokta Sayısı)
DVM	: Dalvik Virtual Machine (Dalvik Sanal Makinesi)
EER	: Equal Error Rate (Eşit Hata Oranı)
FAR	: False Accept Rate (Yanlış Kabul Oranı)
FBI	: Federal Bureau of Investigation (Federal Soruşturma Bürosu)
FRR	: False Reject Rate (Yanlış Red Oranı)
GUI	: Graphical User Interfaces (Grafiksel Kullanıcı Arayüzleri)
IAFIS	: Integrated Automated Fingerprint Identification System (Tümleşik Otomatik Parmakizi Tanıma Sistemi)
ID	: Identity (Kimlik)

INCITS	: International Committee for Information Technology Standarts (Uluslararası Bilgi Teknolojileri Standartları Komitesi)
INCITS M1	: International Committee for Information Technology Standarts M1 (Uluslararası Bilgi Teknolojileri Standartları Komitesi M1)
JTC1	: Joint Technical Committee 1 (Ortak Teknoloji Teknik Komite 1)
JPG	: Joint Photographic Group (Birleşik Fotoğraf Grubu)
JVM	: Java Virtual Machine (Java Sanal Makinesi)
LED	: Light Emission Diode (Işık Saçan Diyot)
NIST	: National Institute of Standards and Technology (Ulusal Standartlar ve Teknoloji Enstitüsü)
OASIS	: Organization for the Advancement of Structured Information Standards (Yapılandırılmış Bilgi Standartlarının İlerlemesi için Organizasyon)
OPTS	: Otomatik Parmakizi Tanıma Sistemi
PDA	: Personal Digital Assistant (Kişisel Sayısal Yardımcı)
SD	: Secure Digital (Güvenli Sayısal)
SDK	: Software Development Kit (Yazılım Geliştirme Araçları)
SDMC	: Secure Digital Memory Card (Güvenli Sayısal Hafıza Kartı)
SRL	: Serialization (Serileştirme)
UML	: Unified Modeling Language (Birleşik Modelleme Dili)
USB	: Universal Serial Bus (Evrensel Seri Veriyolu)
Wi-Fi	: Wireless Fidelity (Kablosuz Bağlantı Alanı)

ŞEKİLLER

Sayfa No.

Şekil 2.1. Sınıf tipleri yay, döngü ve helezon 7
Şekil 2.2. Giriş parmak izi görüntüsünden ayrıntı noktalarının çıkarılması 9
Şekil 3.1. Android yazılım yığını katmanları 17
Şekil 3.2. Donanım cihazıyla birleşmiş android ve java akış şeması 18
Şekil 3.3. Android sisteminin aktivite yaşam döngüsü akış şeması 19
Şekil 3.4. Android işletim sistemi mimarisinin temel bileşenleri 23
Şekil 4.1. Galton-Henry Parmak izi sınıflandırması 25
Şekil 4.2. Minutiae özellik noktaları; iz sonları ve çatallanmalar 26
Şekil 4.3. Aynı parmağa ait iki farklı parmak izi görüntüsü 26
Şekil 4.4. Bir parmak izi tanıma sisteminin genel blok diyagramı 27
Şekil 4.5. Minutiae özelliklerinin çıkarılma akış şeması 29
Şekil 4.6. Sobel operatöründeki pikselin değeri 30
Şekil 4.7. Parmak izi görüntülerinden yönelim alanlarının tahmin edilmesi 31
Şekil 4.8. İz sonu, çatallanma ve iz pikseli 33
Şekil 4.9. Giriş parmak izi ve yönelim alanları bulunmuş parmak izi 34
Şekil 4.10. Yerelleştirilmiş parmak izi bölgesi ve çıkarılmış iz haritası 35
Şekil 4.11. İnceltilmiş iz haritası ve çıkarılmış Minutiae özellikleri 35
Şekil 4.12. Giriş ve taslak izlerin hizalanması 37
Şekil 4.13. Sınırlayıcı kutu 43
Şekil 4.14. Giriş ve taslak Minutiae özellik noktaları 44
Şekil 4.15. Minutiae özelliklerine dayalı hizalama ve eşleştirme sonuçları 45
Şekil 4.16. Secugen parmak izi okuyucu cihazı 46
Şekil 4.17. Toplam iç yansımalı optik parmak izi sensörü 47
Şekil 5.1. Sitemin bileşenlerinin genel bir görüntüsü 50
Şekil 5.2. Sistemin kullanım durumu diyagramı 52
Şekil 5.3. Sitemin sınıf diyagramı 57
Şekil 5.4. Geliştirilen sistemin arayüz görüntüleri 64
Şekil 5.5. Kimlik bilgilerinin tanımlanması 66
Şekil 5.6. Taranarak alınan parmak izi görüntüsü 68
Şekil 5.7. Alınan kimlik bilgileri ve parmak izi görüntüsü 69
Şekil 5.8. Kimlik bilgilerinin kaydedilmesi 72
Şekil 5.9. SD karta kaydedilen JPG ve SRL dosyaları 73
Şekil 5.10. Parmak izi tanımlama aşamasının UML aktivite şeması 74
Şekil 5.11. Kimlik bilgilerinin doğrulanması 75
Şekil 5.12. Güvenlik seviyesi ayarı ile sınır değerinin değiştirilmesi 81
Şekil 5.13. Kişinin sistemde bulunması ve kimlik doğrulamanın gerçekleşmesi 81
Şekil 5.14. Eşleşmenin gerçekleşmemesi sonucu verilen bilgi 82
Şekil 5.15. Kimlik Doğrulama Aşamasının UML aktivite şeması 83
Şekil 5.16. Sistemin FAR ve FRR hata oranlarının değişimi 86

TABLOLAR

Sayfa No.

Tablo 5.1. Eşleşme puanı ile ilgili güvenlik seviyeleri 62
Tablo 5.2. Güvenlik seviyesi ayarı kullanılarak yapılan uygulama sonuçları 86

GENEL BİLGİLER

Adı ve Soyadı : Süleyman ÇINAR

Anabilim Dalı : Bilgisayar Mühendisliği

Programı : Bilgisayar Mühendisliği

Tez Danışmanı : Prof. Dr. Muhammet KÖKSAL

Tez Türü ve Tarihi : Yüksek Lisans – Mayıs 2014

ÖZET

MOBİL ANDROID ORTAMINDA PARMAK İZİ TANIMA VE KİMLİK DOĞRULAMA SİSTEMİNİN GELİŞTİRİLMESİ

Biyometrik sistemler, kullanım alanının yaygınlaşamaması ve yüksek maliyetlere mal olmasından dolayı pahalı bir teknoloji olarak görülmektedir. Bununla birlikte bu sistemler, donanımsal olarak fazla yer kaplamakta ve taşınabilir olmaktan uzaktırlar. Bu nedenlerle, biyometrik sistemlerin incelenmesi ve kullanım alanları göz önüne alındığında düşük maliyetli ve taşınabilir bir biyometrik sistem tasarımının yapılmasının faydalı olacağı düşünülmüştür.

Bu tez çalışmasında, Android mobil cihazı ve evrensel seri veriyolu (USB) bağlantılı parmak izi okuyucusu kullanılarak parmak izi tanıma ve kimlik doğrulama sistemi geliştirilmiştir. Parmak izi doğrulama, parmak yapısında bulunan deri özelliklerinin meydana getirdiği çizgi ve aralıkların farklılaşmasını kullanan karşılaştırma metodudur. Bu çizgilerin USB parmak izi okuyucusu kullanımı ile USB arayüzünden Android platformun sayısal ortamına aktarımı sağlanmıştır. Yapılan literatür taramasında, Özellikli Nokta Eşleştirme (Minutiae Matching) metodunun kullanılmasına karar verilmiş ve Android mobil platformda parmak izi eşleştirmeye dayalı kimlik doğrulama sistemi gerçekleştirilmiştir. Ayrıca biyometri ve kullanıcı arayüzü çözümlerini birleştiren bir kullanıcı arayüzü programı oluşturmaya ve biyometrik kimlik bilgilerini analiz etmek için güvenli bir kullanıcı arayüzü geliştirmeye odaklanılmıştır. Burada kullanıcı arayüzünün tasarımında ve uygulama aşamasında yine Android platformu temel alınmıştır.

Bu tezin ana hedefi, farklı biyometrik tanıma tekniklerini incelemek, bu tekniklerden Android mobil arayüz için yeni gereksinimler oluşturmak ve bu doğrultuda bir arayüz prototipi geliştirmektir.

Anahtar Kelimeler: Android, özellikli nokta eşleştirme, mobil kimlik doğrulama, parmak izi tanıma, parmak izi tanımlama.

GENERAL INFORMATION

Name and Surname : Süleyman ÇINAR

Field : Computer Engineering

Program : Computer Engineering

Supervisor : Prof. Dr. Muhammet KÖKSAL

Degree Awarded and Date : Master of Science – May 2014

SUMMARY

DEVELOPMENT OF IDENTITY AUTHENTICATION SYSTEM AND FINGERPRINT RECOGNITION ON MOBILE ANDROID ENVIRONMENT

Biometric systems are regarded as an expensive technology due to their usage areas' inability to proliferate and their high costs. However, these systems occupy very much space in terms of hardware and they are far from being mobile. Therefore, building of a low-cost and portable biometric system design is thought to be beneficial when biometric systems are examined and usage areas are taken into account.

Fingerprint recognition and identity authentication system are developed by using a fingerprint reader on an Android platform and universal serial bus (USB) in this study. Fingerprint authentication is a comparison method that uses differentiation of lines and range formed by the skin features that exist in finger structure. These lines are transferred to the platform of Android numerical environment by the usage of USB fingerprint reader and USB interface. In the compiled literature, specialty point matching method (Minutiae Matching) is decided to be utilized and fingerprint matching system on Android mobile platform is realized. Moreover, formation of a user interface program that combines the solutions of biometry and user interface, and the development of a safe user interface to analyze biometric identity information are focused on. Here, Android platform is once again based on in the designing and application of user interface.

The main goal of this dissertation is to analyze different biometric recognition techniques, to create new necessities for the Android mobile interface from these techniques and to develop an interface prototype in this perspective.

Keywords: Android, minutiae matching, mobile identity authentication, fingerprint recognition, fingerprint identification.

1. GİRİŞ

Bugünün dünyasında kullanılan bilgiler farklı özellikleriyle birlikte çok önemli bir role sahiptirler. Bilgi yalnızca ticaretin temelinde değil aynı zamanda bir bütün olarak toplumun da temelinde yer almaktadır. Bilgiyi kullanan ve ileten tüm teknolojiler arasında mobil iletişim, dünya çapındaki ağ tarafından güçlendirilen ve önde gelen hizmetler sunduğu ve sabit iletişim sağladığı için büyük bir önem taşımaktadır. Günümüzde mobil iletişim yaygın olarak tüm dünyada kentsel alanlar ve uzak yerlerde teknoloji lideri ve çözüm sağlayıcısı olarak bilinmektedir. Evrensel mobil bağlantılar 2011 yılı sonu itibariyle dünya nüfusunun % 87'si ile kıyaslanabilen bir rakam olan 6 milyar değerine ulaşmıştır. Diğer yandan 2010 yılında 5.4 ve 2009 yılında 4.7 milyar olan bu değerler mobil abonelikte oldukça büyük bir artışın olduğunu göstermektedir (Mobithinking, 2013). On yıl önce cep telefonu kullanımında böyle büyük bir artışın olacağı ve günümüzdeki uygulamalarla rekabet edebilecek derece çeşitliliğin oluşacağı öngörülmemiştir.

Yetişkinlerin çoğu mobil uygulamalar olmadan yaşayamamaktalar; bu durum cep telefonlarının insanların yaşamlarına ne derece nüfuz ettiğini ve günümüz toplumunun nasıl da ayrılmaz bir parçası haline geldiğini açıkça göstermektedir. Akıllı telefonların geliştirilmesinden bu yana, insanlar her geçen gün daha yeni uygulamalar ve oyunlar edinmeye her zamankinden daha fazla ilgi göstermektedirler. Bu alanda araştırma yapan çeşitli şirketler arasında olan Google bu insanların ilgilerinden faydalanmayı bilen en iyi çözümü geliştirmiştir. Google mobil cihazlar için Android denen açık bir platform geliştirmiştir. Bu işletim sistemi; uygulama sunucuları üzerine geliştirilen yazılım (middleware), kullanıcı arayüzü ve uygulamalar içeren yaygın bir platformdur. Bu sistem birçok cep telefonuyla uyumludur. Google'un Android'i, Microsoft'un Windows Mobile'i ve Apple'ın Iphone'nu cep telefonu işletim sistemleri arasında rekabet eden en önemli üç işletim sistemidir (Developer, 2013).

Günümüzde çeşitli kurumsal ve kamu işlemleri uygulamalarında, kişisel bilgilerin saklanması ve korunumu gibi alanlar da güvenlik çok önemli bir konuma gelmiştir. Kişisel verilerin kayıt altına alınması, çalışma ortamlarında personel takibi, güvenlik kuvvetlerinde kişi takibi ve bilgilerine erişebilmek, sanal kimlik uygulamalarında en önemli güvenlik nedeni haline gelmiştir. Bu bilgilerin saklanmasında çeşitli yöntemler kullanılmaktadır. Biyometrik özellikler olarak adlandırılan; gözün özellikleri (retina ve iris gibi), mimik ve yüz özellikleri, el geometrisi, bilek ve el damarları, yürüyüş biçimi, imza tanıma, konuşma doğrulama ve parmak izi tanıma gibi alanlar bu tekniklerden bazılarıdır. Bu biyometrik özellikler, fiziksel ve davranışsal karakteristiklere bağlı bireyleri birbirinden farklı kılacak özelliklerin sisteme kayıt edilmesini ve kimlik tespiti sürecinde kayıtlı verilere ulaşımı sağlamaktadır. Gerçekleştirilen herhangi bir biyometrik sistemde dikkat edilmesi gereken nokta, dış ortamdan alınarak çıkarılan özelliğin doğru bir şekilde tam olarak alınması ve kayıt için dış etkilere karşı güvenilir bir veri tabanının oluşturulmasıdır. Daha sonra kimliklendirme yapmak için sisteme giriş yapılarak kullanılan biyometrik yöntemlerden bir veya bir kaçı ile kimlik tespiti işleminin gerçekleştirilmesidir. Bu çalışmada günümüzde geçerli olan ve gelişen biyometrik yöntemler incelenmiş olup Android tabanlı Mobil platform kullanılarak biyometrik parmak izi tanıma teknolojisine dayalı yeni bir sistem gerçekleştirilmiştir.

Geçmişin sloganı en hızlı iletişimdi. Günümüzde mobil iletişim alanında güvenlik ve çok yönlülüğe artan ilgi sebebiyle bu slogan en güvenli iletişim olarak değişmiştir. Bugün Iphone telefonlarda uygulanabilen biyometrik tanıma kavramı yakın geçmişte insanlar için bir hayaldi. Akıllı telefon sayısındaki artış bu durumu haklı göstermektedir. Mobil ortamların ortaya çıkışı ve koruyucu güvenlik donanımlı uygulamalar gelecekte akıllı telefonların fark yaratacak olan en temel özelliği olacaktır. Dolayısıyla sensörler, güvenli kullanıcı etkileşimi için biyometri uygulamaları gibi çeşitli uygulamaları bir araya getirerek kullanıcı için güvenli etkileşim sağlayan uygulamalar cep telefonlarının geleceğinde başarının örneği olacaklardır. Günümüzde Asya'daki çoğu insan internet bağlantılı kişisel bilgisayarlar yerine cep telefonları aracılığıyla internete bağlanmaktadırlar. Biyometri gibi şirketler tüm dünyada büyük bir pazar potansiyeli kazanarak bu büyük sektörden faydalanmak için üstün özellikli, güvenli mobil uygulamaların geliştirilmesine odaklanmaktadırlar.

Mobil endüstri dünyasında bir devrim gerçekleştirmiştir. Bilgi teknolojisinde akıllı telefonun kullanımı, iletişim, taşımacılık, bankacılıkta olduğu gibi günlük iş dünyasının da alt yapısını oluşturmakta ve neredeyse bütün sektörler her geçen gün mobil iletişim sistemlerine duyulan bağımlılığın biraz daha arttığını göstermektedir. Bu artan bağımlılıklarla beraber bir sürü uygulamayla bağlantılı mobil ağ, tek bir çatı altındaki dünyaya benzetilmektedir. Bilgilerin çoğu yüksek öneme sahip, özel ve olağanüstü değerlidir. Günümüzde artış göstermekte olan bilgi hırsızlığı amaçlı siber saldırılar, bilgi sahibi için bilginin güvenliği konusunda bir endişe kaynağı oluşturmaktadır. Bu durum, biyometrik kimlikler gibi tehlikeleri belirleyen ve gerektiğinde mümkün olduğunca bu tehditlere karşı önlem alan çeşitli güvenlik araçlarının ortaya çıkmasına neden olmuştur. Bu saldırıların kaynağını ortaya çıkarmak için birçok çalışma yapılmıştır. Yapılan açıklamaya göre 2011 yılında araştırmaya katılanların yaklaşık yarısının bir önceki sene en az bir güvenlik sorunu oluşmuş, % 45,6'sı en az bir kasıtlı saldırıya maruz kaldıklarını bildirmişlerdir. Dolayısıyla çözülmesi gereken bir güvenlik ve kimlik hırsızlığı kaygıları olduğu ortaya çıkmıştır. Bu durum aynı oranda devam ederse, gerçekleşen kayıplar daha da artacaktır.

Çalışmanın ikinci konu başlığı altında biyometrinin tanımı ve biyometrik yöntemlerden bahsedilmekle birlikte özellikleri detaylı bir şekilde incelenmiş olup takip eden üçüncü konu başlığında arayüzün geliştirilmesinin arka planındaki Android platform hakkında bilgi verilmiştir. Bu bölümde soyut sistem, sistem bileşenleri, modelleme dili, araçları örneklerle gösterilmiş ve analizin teorisi vurgulanmıştır. Dördüncü kısımda parmak izi tanıma teknolojileri üzerinde durulmuş ve parmak izi eşleştirilmesinde kullanılan yöntem ve araçlar incelenmiştir. Burada arayüz kısmında kullanılan Minutiae Matching (Özellikli Nokta Eşleştirme) algoritmasının matematiksel modeli anlatılmıştır. Beşinci bölümde geliştirilen Android tabanlı mobil biyometrik kullanıcı arayüzü uygulaması analiz edilerek detayları ve tasarım örnekleri sunulmuştur. Son bölümde yer alan altıncı konu başlığında ise uygulama sonuçları ve ileride uygulamanın geliştirilebileceğini düşündüğümüz aşamaları kısaca tanımlanmıştır.

Biyometrik sistemlerin fazla yer kaplamaları, yüksek maliyetli oluşları ve taşınabilir olmamaları gibi nedenlerle kullanım alanları fazla yaygınlaşamamıştır. Bilgi kullanımında mobil iletişimin büyük bir payının bulunduğu, güvenliğin her yerde çok önemli olduğu ve parmak izinin en fazla kullanılan, taklit edilemez bir

biyometrik bilgi olduğu görülmektedir. Mobil platformda biyometri ve kullanıcı ara yüzü çözümlerini birleştiren bir kullanıcı arayüzü programına ihtiyacın olduğu görülmüştür. Güvenliğin her yerde sağlanması için mobil aygıtlar ile biyometrik kimlik bilgilerini analiz eden bir kullanıcı arayüzü programına gereksinim duyulmaktadır.

Bu tez çalışması kapsamında biyometri ve mobil Android tabanlı çözümleri birleştiren kullanıcı arayüzü sistemi oluşturmaya ve biyometri kimliklerini (ID) analiz etmek için güvenli bir parmak izi tanıma ve kimlik doğrulama sistemini geliştirmeye odaklanılmıştır. Kullanıcı arayüzünün tasarım ve uygulanması Android platform temel alınarak yapılmıştır. Mobil ortamlar incelendiğinde Android işletim sisteminin en yaygın kullanılan bir işletim sistemi olması ve yazılım geliştiricileri için çözümler sunması gibi nedenlerden dolayı tercih edilmiştir. Parmak izi bilgileri için gerekli olan veriler biyometrik arayüz den alınmıştır. Parmak izi görüntülerinden biyometrik özelliklerin elde edilmesi için noktasal ayrıntı tabanlı özellik çıkarma (Minutiae Extraction) ve noktasal ayrıntı tabanlı özelliklerin eşleştirilmesi (Minutiae Matching) algoritmaları kullanılmıştır (Secugen, 2013). Bu çalışmada farklı biyometrik tanıma teknikleri incelenmiş, bu tekniklerden parmak izi tanıma yönteminin diğer yöntemlere göre başarı ve performansının daha yüksek ve çok daha güvenilir olması gibi nedenlerden dolayı tercih edilmesine karar verilmiştir. Parmak izi tanıma işleminde, Minutiae tabanlı parmak izi tanıma yöntemi kullanılmış ve parmak izi görüntüsü üzerinden noktasal özellikler elde edilerek eşleştirildiğinde diğer yöntemlere göre daha başarılı ve performanslı olduğu için tercih edilerek kullanılması uygun görülmüştür.

Bu nedenlerle, biyometrik sistemlerin incelenmesi ve kullanım alanları göz önüne alındığında düşük maliyetli ve taşınabilir bir biyometrik sistem tasarımının yapılmasının faydalı olacağı düşünülmüştür.

2. BİYOMETRİK SİSTEMLER

Çalışmanın bu kısmında biyometri, biyometrik sistemler, biyometrik standartlar ve performans ölçütleri hakkında bilgiler verilmiştir. Gerçekleştirilen mobil Android parmak izi tanıma ve kimlik doğrulama sisteminde kullanılan biyometrik parmak izi tanıma yöntemi anlatılmıştır. Bu bilgiler doğrultusunda tez kapsamında gerçekleştirilen Android tabanlı mobil sistemde, biyometrik parmak izi tanıma yönteminin kullanılarak bir arayüz uygulamasının gerçekleştirilmesine karar verilmiştir.

2.1. Biyometri

Günümüzde uygulama alanları çok çeşitlenen biyometrik sistemler çok eski zamanlardan beri kullanılmaktadır. Farklı alanlarda çok çeşitli biyometrik doğrulama sistemleri geliştirilmiştir. Biyometrik sistemlerin kullanılmasıyla kimlik doğrulama işlemlerinde büyük kolaylıklar sağlanmıştır. Kişiler herhangi bir bilgi ezberlemeden veya bir kart taşımadan kimlik doğrulayabilmektedirler (Şamlı ve Yüksel, 2009). Biyometrik sistemlerde kimlik doğrulama işlemleri kişinin biyometrik özellikleri kullanılarak yapılmaktadır. Bu nedenle başkasına devredilmesi veya kaybolması gibi olaylar gerçekleşmemektedir.

Biyometrik tanıma sistemleri bireyleri birbirinden ayırabilme olanağı sunarak bireyin kim olduğunu kanıtlamasına olanak sağlamaktadırlar. Biyometrik sistemlerle daha güvenli sistemler geliştirilmektedir. Biyometrik kimlik doğrulamada sistemin başarısı ve güvenliği için birden fazla yöntem geliştirilmiştir. Genel olarak biyometrik sistemlerin çalışma prensibi; her yöntemin kendine özel girdi cihazıyla alınan verilerin incelenip daha önceden kaydedilmiş değerlerle karşılaştırılıp eşleştirilmesinden oluşmaktadır (Kakıcı, 2013). Biyometrik sistemler güvenliğin çok önemli olması nedeniyle geliştirilmiştir. Örneğin bir biyometrik tanıma sistemiyle parmak izinden suçlu tespiti yapılabilmektedir.

Biyoloji bilimlerinde biyometri ve biyometrik sistemler veri analizi ve sorunları için istatistiksel ve matematiksel yöntemlerin geliştirilmesinde referans olarak kullanılmaktadır. Bugünün dünyasında bir bireyin kimliğini doğrulamak için güvenilir kimlik doğrulama yöntemleri kullanılmaktadır. Biyometrik sistemlerle binalara, bilgisayar sistemlerine, dizüstü bilgisayarlara, cep telefonlarına, USB hafiza belleklerine güvenli bir erişim sağlanabilmektedir (Femila ve Irudhayaraj, 2011).

2.2. Biyometrik Sistemlerde Kullanılan Yöntemler

Biyometrik sistemlerin yöntemleri iki aşamadan oluşmaktadır. İlk aşamada bireyin kimliğini oluşturan bilgiler sisteme ait araçlar kullanılarak alınmaktadır. Alınan bu bilgiler, yönteme ait kullanılan algoritmalarla incelenmekte ve kişiyi tanımlayacak özellikler çıkarılarak veritabanına kaydedilmektedir. Diğer kısım da ise bireyin sisteme ait aynı cihazla alınan bilgileri, sistemde kullanılan yönteme ait aynı algoritmalarla incelenmekte ve kimliğinin özellikleri elde edildikten sonra veritabanında ki bilgilerle karşılaştırılmaktadır. Eğer eşleşme oluşursa bireyin kimliği doğrulanarak kimliği tespit edilmektedir. Biyometrik sistemler, kullanılan çeşitli yöntemler, algoritmalar ve cihazlar nedeniyle birbirinden çok farklılıklar göstermektedir (Kakıcı, 2013).

Biyometrik sistemler, bireyin fiziksel veya davranışsal özelliklerini (örneğin yüz, parmak izi, ses, imza, tuş darbesi ritimleri) analiz ederek kimliğinin benzersizliğini doğrulamaktadırlar. Bu sistemler bireyi kayıtlı özellikleri ile her defasında karşılaştırmaktadırlar. Biyometrik sistemler, biyometrik tanıma algoritmalarını uygulayan bir bilgisayar sistemini kullanmaktadır. Bu sistemler genel olarak algılama, özellik çıkartma ve eşleştirme aşamalarından oluşmaktadır (Mudholkar ve diğ., 2012).

Günümüzde geçerli olan biyometrik tanıma sistemleri temel olarak iki grupta incelenmektedirler. Bunlardan ilki; Fizyolojik Özellikler (Parmak İzi, Retina, DNA, Damar, Yüz, El Geometrisi, Ses, Yüz Termogramı) olarak bilinmektedir. Diğer grupta ise; Davranışsal Özellikler (İmza Atımı, Yürüyüş, Tuş Vuruşu, Konuşma) yer almaktadır (Şamlı ve Yüksel, 2009).

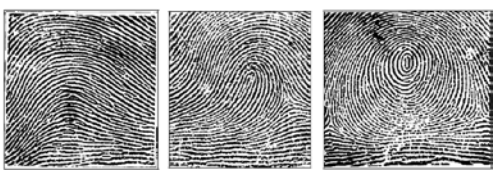
2.2.1. Parmak İzi Tanıma

Parmak izi en fazla kullanılan taklit edilemez bir bilgidir. Parmak izi tanıma sistemlerinin yazılım ve donanım alanlarında çok büyük ilerlemelerin olması, bu sistemlerin hızlı ve kolay bir biçimce kullanılmasına yol açmıştır. En temel parmak izi tanıma algoritmaları korelasyon, ayrıntı (Minutiae) ve çizgi (ridge) tabanlı eşleştirme yöntemleridir. Korelasyon tabanlı eşleştirme yöntemleri, iki farklı çizgi modeli karşılaştırıldığından dolayı kayıt noktalarının kesin yer bilgisini gerektirirler ve resmin çevrilmesinden etkilenmektedirler. Ayrıntı (Minutiae) tabanlı eşleştirme tekniğinde ise parmak izinin ayrıntı noktaları belirlenmekte ve bu ayrıntı noktaları oluş sırasına göre karşılaştırılmaktadır. Çizgi tabanlı eşleştirme yönteminde ise çizgiye ait yön ve şekil özellikleri kullanılmaktadır. Ayrıntı ve çizgi temelli teknikler düşük çözünürlükteki parmak izi görüntülerinden ayrıntı ve çizgi özelliklerini çıkaramamaktadırlar. Ayrıntı ve çizgi tabanlı eşleştirme yöntemlerinde görüntü iyileştirme ve temizleme yöntemleri kullanılmaktadır (Sönmez ve diğ., 2007).

Galton, parmak izinin kalıtımsal olmadığını ve her insanın parmak izinin birbirinden farklı olduğunu yapmış olduğu çalışmalarla ifade etmiştir. Henry Faulds ise parmak izinin sınıflandırılmasına kesin olarak açıklık getirmiştir. Farklı sınıflandırmalar olsa bile Galton ve Henry'nin yapmış olduğu çalışmaların ürünü olan sınıflandırma sistemi yaygın olarak kullanılmaktadır (Kakıcı, 2013). Bir otomatik parmak izi tanıma sisteminde (OPTS) parmak izi tanıma işlemi, parmak izinde bulunan özellik noktalarının ve bunlara ait bilgilerin karşılaştırılması ile gerçekleştirilmektedir (Şamlı ve Yüksel, 2009).

2.2.1.1. Parmak İzinin Özellikleri

Galton ve Henry'nin çalışmaları sonucunda bulunan sınıflar çizgi biçimlerine göre ayrılmakta ve bu sınıflar Şekil 2.1'de gösterilmektedir (Henry, 1900).



Şekil 2.1. Sınıf tipleri yay, döngü ve helezon

Daha detaylı çalışmalar için bu sınıflarda alt sınıflara ayrılmaktadır. Bu sınıflandırmalarla birlikte parmak izleri çizgi formatında incelendiğinde belirli özelliklere sahip noktalar ortaya çıkmaktadır. Parmak izinden gerekli özellikleri elde etmek için aşağıdaki görüntü işleme teknikleri kullanılmaktadır (Kakıcı, 2013).

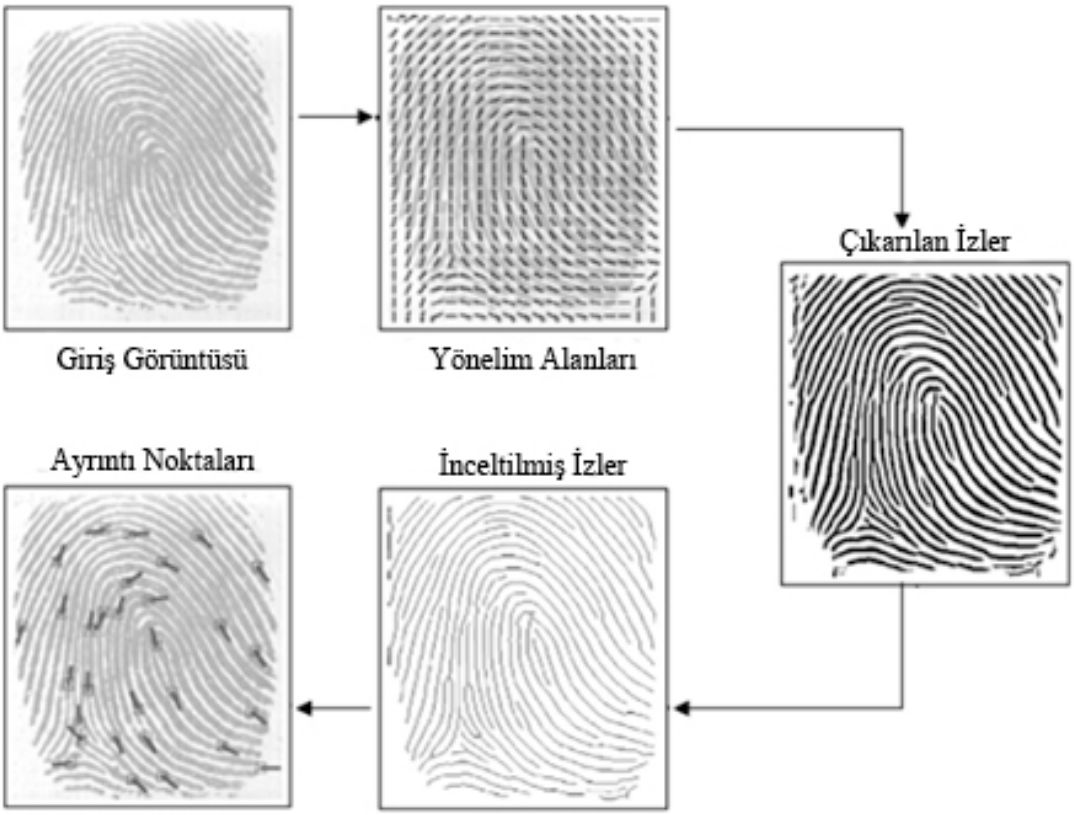
• Görüntünün gürültülerden temizlenmesi

• Kenarların tespiti

• Özelliklerin çıkarılması

Parmak izinin gürültülerden temizlenmesi için filtreler kullanılmaktadır. Median, mean ve gaussian gibi filtreleme işlemleri yapılmaktadır. Kenar tespiti için sobel, gradient, prewitt filtreleri kullanılmaktadır. Özelliklerin tespit edilmesi için kenar algılama işleminin sonucunda oluşan görüntüde inceltme algoritmaları uygulanarak parmak izi çizgileri uygun hale getirilmektedir. Bu işlemin sonunda çizgilerin bitiş ve çatal noktaları elde edilmektedir (Kakıcı, 2013).

Parmak izi en yaygın olarak kullanılan biyometrik bir özelliktir. Parmak izi temel olarak parmak yüzeyindeki sırt ve vadilerin birleşiminden oluşmaktadır. Parmak izi desenini oluşturan hatlara sırtlar ve sırtlar arasındaki boşluklara ise vadiler veya oluklar denmektedir. Yüksek kaliteli bir parmak izi görüntüsü elde edildiğinde, bir taslak içine parmak izinin ayırt edici özellikleri dönüştürülmektedir. Bu süreç özellik çıkartma olarak adlandırılmaktadır. Giriş görüntüsü alındıktan sonra ayrıntı tabanlı (Minutiae) tekniğinin kullanılma aşamaları Şekil 2.2'de gösterilmiştir (Chaudhary ve Nath, 2009). Görüntü üzerinde parmak izi tanıma işlemleri olarak görüntü iyileştirme, ayrıntıların çıkarılması (Minutiae) ve eşleştirme adımları uygulanmaktadır. Parmak izi resminde görüntü iyileştirmeleri yapılarak sırt yapısı açıklıkları artırılmakta böylelikle özellik noktaları daha kolay bir biçimde çıkarılmaktadır. İyileştirilen parmak izi görüntüsü ikili hale dönüştürülmektedir. Daha sonra çatallanmaların kesin yeri için sırt kalınlığının piksel değeri azaltılarak görüntü üzerinde inceltme algoritması kullanılmaktadır. Görüntü iyileştirilerek hat sonları ve çatallanma noktaları tespit edilmektedir. Ayrıntıların yerleri işaretlenerek yönleriyle birlikte çıkartılmakta ve bir kimlik özelliği oluşturmak için saklanmaktadır. Daha sonra ayrıntı (Minutiae) noktaları eşleştirilmekte, taslak ve giriş ayrıntılar kendi aralarında hizalanarak en fazla sayıda eşleşmelerin bulunması sağlanmaktadır (Chaudhary ve Nath, 2009).



Şekil 2.2. Giriş parmak izi görüntüsünden ayrıntı noktalarının çıkarılması

2.2.3. Yüz Tanıma

Yüz tanıma sistemi devrim niteliğinde olan biyometrik buluşlardan biridir. İlk olarak askeriye alanlarında kullanılarak geliştirilmiştir. İleri teknoloji silahlarında, caddelere yerleştirilen güvenlik kameralarıyla aranmakta olan bir suçlunun tespiti ve yakalanması gibi uygulamalarda kullanılmaktadır. Güvenlik amaçlı olarak yüz görüntülerinin otomatik olarak tanınması yaygın bir biçimde kullanılmaktadır (Şamlı ve Yüksel, 2009).

Yüz tanıma kullanımı kolay olan bir yöntemdir. Yüz nitelikleri insanların birbirlerini tanımak için insanlar tarafından kullanılan en yaygın biyometrik özelliklerdendir. Kimlik doğrulama işlemlerinde vesikalık fotoğraflar kontrol edilmekte ve yüz tanıma yöntemiyle kimlik tespiti gerçekleştirilmektedir. Yüz tanıma işlemleri iki yöntemden birine göre yapılmaktadır. İlk yöntemde gözler kaşlar, burun, dudak, çene ve yüz boşluklarının ilişkileri, ya da yüz niteliklerinin yeri ve şekline göre gerçekleştirilmektedir. İkinci yöntem ise yüzü temsil eden yüz niteliklerinin analiz edilmesiyle yapılmaktadır (Jain ve diğ., 2007). Otomatik bir yüz

tanıma sisteminin iyi çalışabilmesi için, bir yüzün alınan görüntüde mevcut olup olmadığının tespit edilmesi, eğer varsa yüzün bulunması ve farklı çevre koşulları altında (alınan herhangi bir resimden) genel bir bakışla yüzün tanınması gerekmektedir.

2.2.4. İris Tanıma

İris tanıma sistemlerinin kullanım amacı, iris şeklinin bir ömür boyunca değişmemesi ve diğer biyometrik sistemlere göre daha az bozulacak ve daha az zarar görecek bir yapıya sahip olmasıdır. İris tanıma sistemi gözleri olmayan, gözleri görmeyen, Nistagmus hastalığına sahip (gözleri titreyen) veya irisleri olmayan kişilerde uygulanamamaktadır. Bu kişiler haricinde havaalanları gibi kimlik doğrulamanın çok önemli olduğu yerlerde çok yüksek bir başarı ile uygulanmaktadır (Şamlı ve Yüksel, 2009).

İris tarama biyometrik sistemleri bir bireyin kimliğini doğrulmak amacıyla, bireyin yaşamı boyunca değişmeden kalan insan irisinin karakteristik özelliklerini kullanmaktadır. İris gözün pigmentli veya renkli dairesinin alanıdır, genellikle kahverengi, yeşil, gri veya mavi halkalar irisi oluşturmaktadır. İrisin insan bünyesinde iyi korunan ve yaralanmalardan en az etkilenen bir yapısı vardır. Genellikle iris tarama işlemi irise yakın özel bir kamerayla çekilen bir fotoğrafla başlamaktadır. Kullanıcı iris okuma cihazına yaklaşık bir metre mesafede durmak zorundadır. Kamera gözü aydınlatarak çok yüksek çözünürlükle bir fotoğraf çekmek için kızılötesi bir yansıtıcı kullanmaktadır. İrisin iç yüzeyindeki desenlerin karakteristik özelliklerinin haritasını çıkaran bir iris tarama algoritması kullanılmaktadır. İris desen bilgileri karmaşık ve birbirinden farklı 200'ün üzerinde özellik noktalarından oluşmaktadır (Femila ve Irudhayaraj, 2011). Bu eşsiz noktaları oluşturan halkalar radyal bir modda iris bölgesinin görüntüsünü veren doku içinde sınıflandırılmaktadır. Bir bireyin sağ ve sol göz yapıları da farklı olduğundan, iris tanıma teknolojileri bu özelliği de kullanarak yanlış biyometrik tanımaların önüne geçmektedirler.

2.2.5. Retina Tanıma

Retina tanıma işlemi insanın göz bebeği arkasındaki damar tabakanın doğrulanarak kimliğinin tanınmasıdır. İnsanların damar yapıları birbirlerinden farklı olmasına rağmen çeşitli hastalıklar sonucu damar ve göz yapısı değişerek damarları etkilediğinden çok fazla yaygınlaşmış bir yöntem değildir. Retina tanıma işleminde özelliklerin belirlenmesi sırasında kişinin belirli bir noktaya bakması bu yöntemin az kullanılmasına neden olmaktadır (Kakıcı, 2013).

İris tanıma teknolojisi ile birlikte, retina tarama sistemi çok güvenilir ve doğru bir biyometrik teknolojiyi oluşturmaktadır. Kullanımı en zor olan biyometrik tanıma sistemlerinden biridir. Kullanıcıdan uygun bir başarı elde edilmesi için sabırlı olması gerekmektedir. Biyometrik kimliklendirme için kullanılan retina damarlarının desenleri kişiden kişiye ve tek yumurta ikizleri arasında bile benzersiz olduğu kanıtlanmıştır. Retina desen yapısı bir ömür boyunca değişmemektedir (Femila ve Irudhayaraj, 2011). Retina tarayıcı cihazı ile net bir görüntünün alınması için bir noktaya sürekli bakılması gerekmektedir. Bu teknolojiyle eşsiz retina desenlerinin taranması için optik yansıtıcı yoluyla düşük yoğunluklu kızılötesi ışık kaynağı kullanılmaktadır. Damarlarla ilgili çıkarılan bilgiler kaydedilmektedir. Retina tanıma sistemi, kimlik tespiti ve doğrulaması işlemlerinde başarıyla uygulanmaktadır.

2.2.6. Damar Tanıma

Damar tanıma retina tanımaya benzemektedir. Damar tanımada göz arkasındaki damarlar yerine el üzerindeki damarların yapısına bakılmaktadır. Bu tanıma sisteminde el görüntüsü alındıktan sonra damarların yapısı belirlenmektedir. El üzerindeki damarların yapısı da kişiden kişiye farklılıklar göstermektedir. Retina tanımaya göre el damarlarından kimlik tespit işlemleri daha kolay gerçekleştirilmektedir (Kakıcı, 2013). El üzerinde oluşabilecek çeşitli nedenlerden dolayı damar yapısı başarıyla elde edilemediğinden tanıma işlemleri olumsuz sonuçlar vermektedir.

2.2.7. El Yazısı Tanıma

El yazısını oluşturan harflerin şekil ve konumlarının biçimleri birbirinden farklılıklar göstermektedir. Harflerin oluşturulma sırası, noktaların ve çizgilerin oluşturulma şekli de belirlenecek özellikleri oluşturmaktadır. Daha önceden yazılmış yazılardan kimlik tanıma işlemi yapıldığında bazı özellikler değiştiğinden, yazı yazılırken yapılan tespit işlemi daha doğru sonuçlar vermektedir. Diğer tanıma yönteminde kullanılan cihazlar farklı işler içinde kullanılabildiği halde el yazısı tanıma için üretilen cihazlar sadece bu amaçla kullanılmaktadırlar. Bu nedenle diğer cihazlara göre maliyetleri daha fazla olmaktadır (Kakıcı, 2013).

2.2.8. DNA Tanıma

Bu tanıma sisteminde bireyin saç, tırnak, deri parçası, kan, sperm veya herhangi bir biyolojik özelliği alınarak hücre içerisindeki DNA molekülleri incelenmektedir. Güvenlik güçleri tarafından suçlu tespiti işlemlerinde kullanılmaktadır. Doğruluğu çok yüksek ve geçerli bir yöntemdir. Bununla birlikte kişinin biyolojik yapısında değişikliğin olması sonucu alınan DNA örneğinin kalitesi azalacağından inceleme yapılması zorlaşmaktadır (Şamlı ve Yüksel, 2009).

2.2.9. El Geometrisi Tanıma

El geometrisi tanıma sistemleri elin şekli, avuç içi boyutu, parmakların uzunlukları ve genişlikleri gibi insan elinden alınan ölçümlere göre yapılmaktadır. El geometrisi tabanlı kimlik doğrulama sistemleri çok yaygın olarak kullanılmaktadır. Bu yöntemin kullanımı basit ve ucuzdur. Elleri etkileyen kuru havalar gibi çevresel faktörlerden el geometrisi tabanlı kimlik doğrulama sistemleri etkilenmemektedirler. Fakat el geometrisi çok fazla ayırt edici olmadığı için büyük nüfuslu kimlik tespiti sistemlerinde tercih edilmemektedir. Çocukların büyüme çağında el geometrisi bilgileri değişebilmektedir. Bununla birlikte, bireylerin ellerine yüzük veya metal eşyalar takması sonucu el geometrisinin özelliklerinin çıkarılması zorlaşmaktadır. El geometrisi tanıma sistemleri, büyük bir nüfusa sahip olan yerlerde, bireyin kimliğinin belirlenmesi gereken yerlerde çok fazla kullanılmamaktadır (Jain ve diğ., 2007). El geometrisi tabanlı sistemin fiziksel boyutu büyüktür. Sadece birkaç parmak yerine

tüm elin (genellikle işaret ve orta parmak) ölçümlerine dayanan kimlik doğrulama sistemleri mevcuttur. Bu sistemlerde kullanılan cihazlar el geometrisi için kullanılan cihazlardan daha küçük, ancak diğer bazı özellikleri (örneğin, parmak izi, yüz, ses gibi) çıkarmak için kullanılan cihazlardan daha büyüktür.

2.2.10. İmza Tanıma

Bir kişinin, herhangi bir yazının altına bu yazıyı yazdığını veya onayladığını ifade etmek için her zaman aynı biçimde yazdığı işaretler veya adlardan oluşmaktadır. İmzalar kişiler tarafından hayatları boyunca çok sık kullanılmaktadır. Hukuksal açıdan çok büyük yaptırımları bulunmaktadır. Sahte imzalar atılması sonucu kişi borç altına girebilmekte ve tüm mal varlığının başkalarına devredilmesi gibi olaylar gerçekleşebilmektedir. Kimlik doğrulamada imzanın kim tarafından atıldığının tespit edilmesi gerekmektedir. İmzayı tespit etmek için iki tip bilgi kullanılmaktadır. İlki imza atarken geçen süre, kalemin basım şiddeti, ivmesi gibi taklit edilmesi zor olan özelliklerdir. Diğer yöntem ise bir imzanın kolay taklit edilebilen desen biçiminden oluşmaktadır (Şamlı ve Yüksel, 2009).

2.3. Biyometrik Sistemlerin Uygulama Alanları

Yüksek güvenlik gerektiren sistemlerde bir kişinin kimliğinin tespit edilmesi çok önemli bir hale gelmiştir. Kullanılan sistemde, "Kimliği iddia edilen kişi gerçekten var mı? Bu tesisi kullanmaya yetkili bu kişi mi? Hükümet tarafından belirlenmiş izin listesinde ismi var mı?" biçiminde değişik güvenlik olayları görülmektedir. Günlük hayatta iletişim ve ulaşımda oluşan hızlı gelişmeler, güvenlik konusunda artan kaygılar nedeniyle güvenilir kullanıcı kimlik doğrulama tekniklerine duyulan ihtiyacı artırmıştır. Bu nedenlerle biyometrik sistemler geliştirilerek çok fraklı uygulamalarda kullanılmaktadırlar. Bu uygulamalar şu üç ana grup altında kategorize edilmektedir (Jain ve diğ., 2007).

- Bilgisayar sistemlerine giriş, elektronik veri güvenliği, e-ticaret, internet erişimi, ATM veya kredi kartı kullanımı, fiziksel erişim kontrolü, cep telefonu, PDA, tıbbi kayıt yönetimi, uzaktan eğitim gibi ticari uygulamalar.
- Ulusal kimlik kartı, ceza evleri, ehliyet, sosyal güvenlik, yardım ödemeleri, sınır kontrolü, pasaport kontrolü gibi devlet uygulamaları.

- Suçluya ait araştırma, ceset kimlik tespiti, anne ve babalık tespiti gibi adli uygulamalar.

2.4. Biyometrik Sistemlerin Standartları

Biyometrik sistemlerden en verimli şekilde yaralanmak için biyometrik tanıma yöntemlerine standartlar getirilmiştir. Standartlaştırma kamu kuruluşları ve diğer organizasyonlar şeklinde iki gruba ayrılmaktadır. Kamu kuruluşları ISO standartlarını kapsar (ANSI, INCITS, CJIS/FBI), resmi olmayan kurumlar ise tanımlı şirketlerden oluşmaktadır (Tilton, 2009).

Biyometrik sistem ve araçlar ile işlemlerin yapılabilmesi için standartlar geliştirilmiştir. Bu standartlar (Varlık, 2008):

• Sağlam çözümlerin gelişmesini sağlar.

• Maliyeti azaltır ve süreklilik sağlar.

• Çalışmalar için yol gösterici niteliğindedir.

• Veri iletimini standartlaştırır.

• Sistemlerin birbiri ile uyumunu sağlar.

• Biyometrik sistemlerin uyumunun test edilmesini sağlarlar.

Biyometrik standartlar aşağıdaki alanlara ayrılmıştır:

• Adli ve tanıma standartları

• Veri standartları

• API (Application Programming Interfaces) program ara yüzleri standartları

• Güvenlik standartları

• Test ve Sertifikasyon standartları ve temel çerçevesi

• Diğer standartlar

Biyometrik güvenlik sistemleri konusunda uluslararası bir standart oluşturulmuştur. Uluslararası Bilgi Teknolojileri Standartları Komitesi (INCITS-International Committee for Information Technology Standards) tarafından, parmak izi, iris, retina ve ses tanıma gibi biyometrik tanıma sistemlerinde kullanılacak işlemlere uluslararası bir standart getirmek için oluşturulmuştur.

Aşağıda standart geliştirme ajans ve kuruluşları gösterilmiştir (Varlık, 2008).

ANSI (Amerikan Ulusal Standart Enstitüsü)

CJIS/FBI IAFIS (FBI için parmak izi sıkıştırma ve açma standartları)

INCITS M1(Bilgi teknoloji standartları için Uluslararası Komite)

NIST (Ulusal standartlar ve Teknoloji Enstitüsü)

JTC1 (Ortak Teknoloji Teknik komite)

OASIS (Yapılandırılmış bilgi standartlarının ilerlemesi için organizasyon)

Standart bir parmak izi incelemesinde 30-40 kadar özellik noktası tespit edilebilmektedir. Yapılan araştırmalar sonucunda iki farklı kişide aynı konumlu özellik nokta sayısının 8'i aşamayacağı kanıtlanmıştır (Maltoni ve diğ., 2009).

İki parmak izinin aynı kimliğe ait olduğunun tespit edilebilmesi için en az 11-12 özellik noktasının benzer olması gerekmektedir. Benzer özellik noktalarının sayısı ülkeler arasında farklılıklar göstermektedir. FBI sayılara bağlı kalmadan araştırma yapmaktadır (Akın ve diğ., 2002).

Bu tez çalışmasında geliştirilen sistemde biyometrik standartlardan tanıma ve doğrulama standartlarında faydalanılmıştır.

2.5. Biyometrik Sistemlerin Performansı

Parola tabanlı sistemlerde, farklı iki biyometrik özellik arasında doğru bir karşılaştırmanın yapılabilmesi için, bir kullanıcının kimliğinin tespit edilmesi gerekmektedir. Bir biyometrik sistemde, iki biyometrik özelliğin karşılaştırılması sonucu bir kullanıcının biyometrik özelliği, azda olsa benzerleriyle eşleştirilebilmektedir. Bunlar; kusurlu algılama sonuçları (örneğin, arızalı sensörler nedeniyle parmak izinin gürültülü olması), kullanıcının biyometrik karakteristiğinde oluşan değişiklikler (örneğin, konuşma tanımayı etkileyen solunum rahatsızlıkları), çevre koşullarında oluşan değişiklikler (örneğin, yüz tanımada tutarsız aydınlatma seviyeleri) ve algılayıcı sensörler ile kullanıcı etkileşimindeki değişiklikler (örneğin, kapanmış iris veya kısmi parmak izi) sonucu oluşmaktadır (Jain ve diğ., 2007). Bir bireyin biyometrik özellikli veri setinde gözlenen değişiklikler sınıf-içi değişim olarak adlandırılmakta ve iki farklı bireyden alınan özellik veri setleri arasında oluşan değişim ise sınıflar arası değişim olarak ifade edilmektedir. Biyometrik özellikli verilerde, sınıf içi küçük ve sınıflar arası büyük değişimler görülebilmektedir.

İki biyometrik özellik arasındaki benzerlik derecesi, bir benzerlik puanı ile belirtilmektedir. Benzerlik karşılaştırma puanı; bir kullanıcıya ait alınan iki biyometrik özellikli örneğin karşılaştırılıp aynı gerçek puan sonuçlarının elde edilmesiyle oluşmaktadır. Sahte bir puan; sınır değerini aştığında yanlış kabul (false

accept) veya yanlış karşılaştırma, gerçek bir puan; sınır değerinin altına düştüğünde yanlış reddetme (false reject) veya yanlış olmayan karşılaştırma olarak ifade edilmektedir. Yanlış kabul oranı; (False Accept Rate - FAR) bir biyometrik sistemde, sınır değerin aşılması sonucu oluşan sahte puan olarak tanımlanmaktadır. Benzer şekilde, Yanlış reddetme oranı; (False Reject Rate - FRR) bir biyometrik sistemde, sınır değerin altına düşülmesi sonucu oluşan gerçek puan olarak tanımlanmaktadır.

FAR ve FRR oranları biyometrik sistemlerin yaptığı karşılaştırmaların ne kadar doğru olduğunu ölçmek için kullanılmaktadırlar. Yanlış kabul oranı, yetkisiz kişilerin sisteme giriş izni, yanlış reddetme oranı ise yetkili bir kişiye sisteme giriş izni verilmemesi olasılığıdır. Biyometrik sistemlerde, karşılaştırılan özelliklerin sayısı bir sınır (threshold) değeriyle kıyaslanmaktadır. FAR ve FRR ölçüm değerleri, bir biyometrik sistemin güvenirliliği hakkındaki en doğru bilgiyi vermektedir (Grother, 2006).

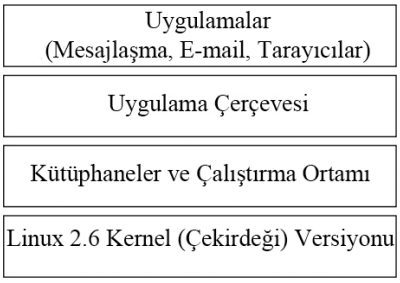
FAR ve FRR değerleri ters orantılı olarak çalışmaktadır. FAR değeri artarken FRR değeri ise azalmaktadır. Sınır değeri ayarlanarak FAR ve FRR değerleri değiştirilebilmektedir.

3. ANDROID PLATFORM

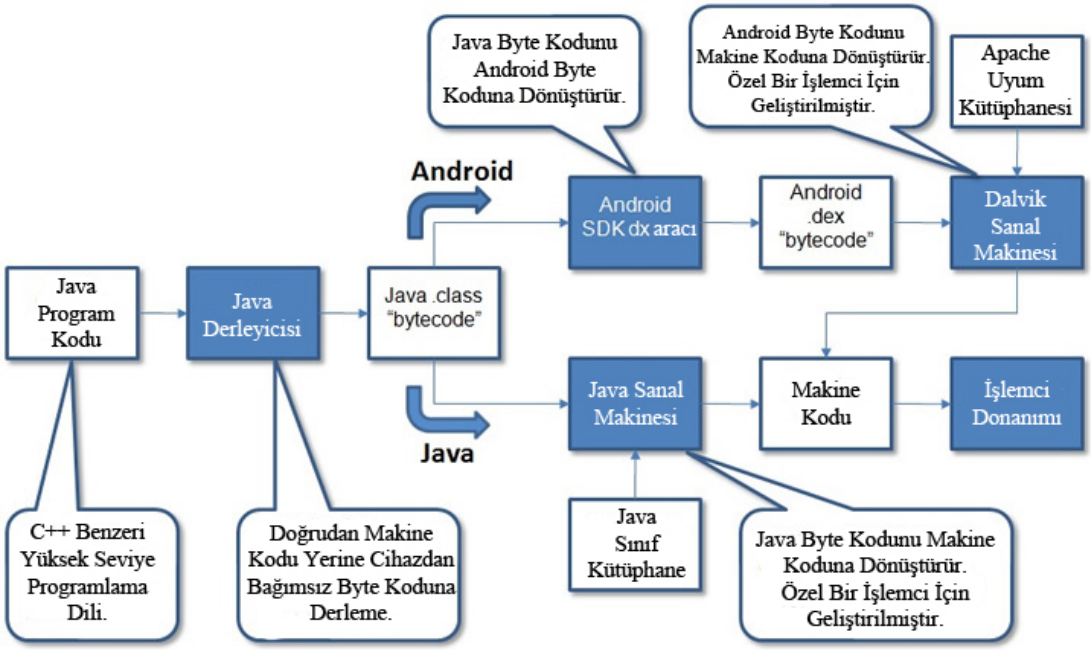
Bu bölümde gerçekleştirilen mobil parmak izi tanıma ve kimlik doğrulama sisteminde kullanılan Android işletim sisteminin yapısı hakkında bilgiler verilmiştir. Bu kısımda Android işletim sisteminin üstünlükleri vurgulanmış ve gerçekleştirilen Mobil parmak izi tanıma ve kimlik doğrulama sisteminde, Android tabanlı işletim sisteminin tercih edilme nedenleri açıklanmıştır.

3.1. Android İşletim Sistemi

Android işletim sistemi, mobil cihazlar için, uygulamalar ve uygulama sunucuları üzerine geliştirilen yazılım (middleware) içeren Linux tabanlı Open Source (açık kaynak kodlu) bir yazılımdır (Fergytech, 2013). Eşi benzeri yoktur çünkü Google aktif bir şekilde Android platformunu geliştirmektedir. Ücretsiz olarak Android işletim sistemini cihazlarında kullanmak isteyen telefon kullanıcılarının, donanım üreticilerinin ve yazılım geliştiricilerinin hizmetine sunmaktadır. Android işletim sisteminin yazılım geliştirme kiti (SDK) Java programlama dilini kullanarak Android işletim sistemindeki uygulamaları geliştirmeye başlamak için gereken araç ve uygulama programlama arayüzlerini (API) sağlamaktadır. Şekil 3.1 (Emeraldinsight, 2013) ve 3.2'de (Technomicon, 2013) Android platformun mobil cihazlar için bir işletim sistemi, middleware ve bazı önemli uygulamalar içeren açık bir yazılım çatısı olduğu gösterilmiştir (Developer, 2013).



Şekil 3.1. Android yazılım yığını katmanları



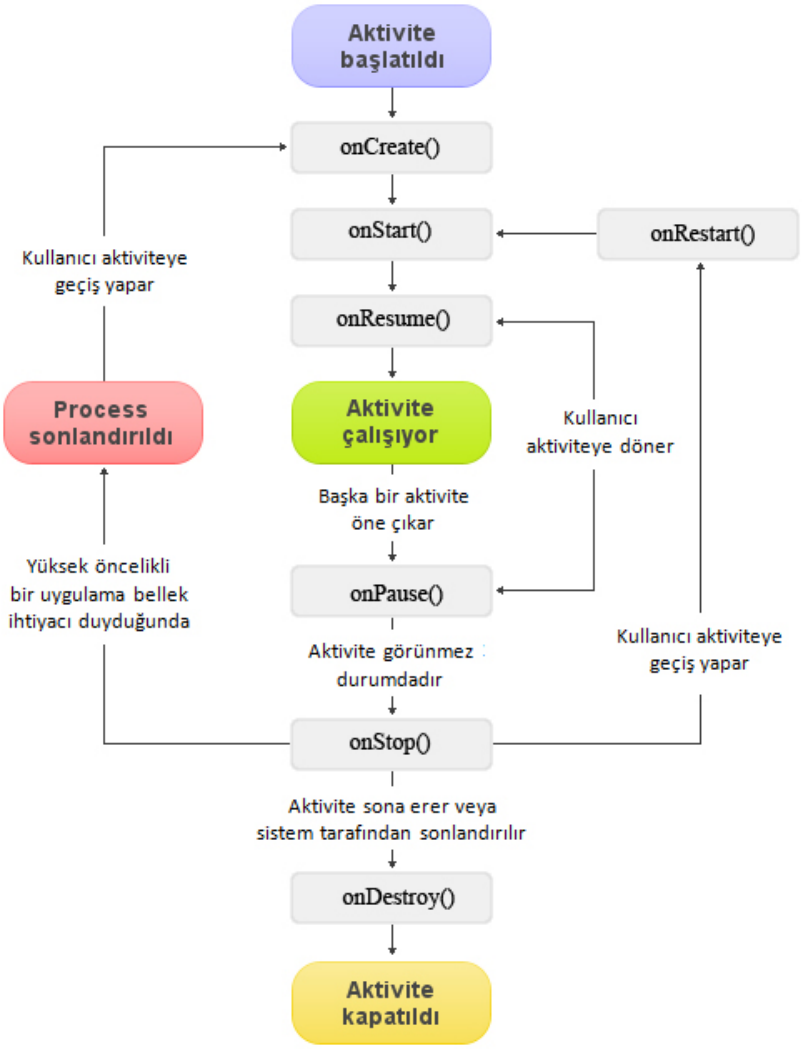
Şekil 3.2. Donanım cihazıyla birleşmiş android ve java akış şeması

Şekil 3.2'de söz konusu bir yazılım cihazında kullanılabilir bir program üretmek için Java ve Android işletim sisteminde, çeşitli parçaların nasıl bir arada çalışabileceği gösterilmiştir. Biri Android işletim sisteminden giden ve diğeri daha klasik ve sade olan Java'dan geçen iki yol gösterilmektedir. Her ikisinde de bir işlemcide çalışabilen makine kodları üretilmekte fakat işlemciye ulaşmak için farklı yollar izlenmektedir (Hibben, 2010).

3.2. Android Kavramları

Android aktivite yaşam döngüsü ve her bir evre ile ilişkilendirilmiş çeşitli olay gidericiler Android işletim sisteminde bir uygulamanın nasıl çalıştığını anlamak için anahtar kavramlardır. Dolayısıyla yazılım geliştiricisi olarak, geliştirmeye başlamadan önce, Android aktivite yaşam döngüsü hakkında derin bir bilgi birikimine sahip olunması gerekmektedir. Android işletim sisteminde bir aktivitenin Şekil 3.3'de (Bilgigunlugum, 2014) gösterildiği gibi dört durumu vardır (Android-apps, 2013).

Dikdörtgen biçimli kutuların her biri durum akışı süresince uygulamanızda çağrılan metotları temsil etmektedir. Aynı şekilde, oval kutuların her biri de aktivitenin içinde olabileceği ana durumları temsil etmektedir (developer, 2013).



Şekil 3.3. Android sisteminin aktivite yaşam döngüsü akış şeması

3.2.1. Çalışma Durumu

onStart(), çalışma konumunda Aktivite ekranın ön planındadır. Tamamen görülebilir ve kullanıcı için aktif haldedir (Android-apps, 2013).

3.2.2. Duraklatma Durumu

onPause(), duraklatma durumunda, aktivite odağını kaybetmiş fakat hala görülebilmektedir. Bu durum, herhangi bir başka aktivitenin ekranın tamamını kaplamaması ya da arka plandaki aktivitenin kısmen görülebilmesi için biraz şeffaflığı olan bu aktivitenin üstüne gelmesiyle gerçekleşmektedir. Duraklatışmış bir

aktivite tamamen canlıdır. Aktivitenin durumunu tamamen koruyabildiği, bilgiyi tutabildiği ve Android işletim sisteminde ki pencereyi kontrol eden pencere yöneticisine bağlı kaldığı anlamına gelmektedir.

3.2.3. Durdurma Durumu

onStop() konumu, Aktivitenin artık kullanıcıya görülür olmadığı durumdur. Bir aktivite başka bir aktivite tarafından görünmez hale getirildiğinde, durdurulmuş denmektedir. Bu durumda aktivite yine canlı ve durumunu korumakta, fakat başka bir yerden hafızaya ihtiyaç duyulduğunda, Android sistemi tarafından aktivitenin yaşam süreci sonlandırılıp imha edilmektedir (developer, 2013).

3.2.4. İmha Durumu

onDestroy() konumunda, Aktivite hafızada yer almıyorsa, imha edilmiş ya da ölmüş denmektedir. Aktivite kullanıcıya göründüğünde, yeniden başlatılarak ya da eski durumuna geri yüklenerek kalınan yerden devam edilmektedir (Android-apps, 2013).

3.3. Android Yaşam Döngüsü Aşamaları

Android işletim sisteminin aktivite yaşam döngüsü grafiği analiz edildiğinde her bir aktivitenin üç döngüsünün olduğu ve bunların geri arama metotları tarafından tanımlandığı görülmektedir. Android işletim sisteminin yaşam döngüsünün üç aşaması bulunmaktadır (developer, 2013).

3.3.1. Android Toplam Yaşam Süresi

onCreate() metodu ile onDestroy() metodu arasındaki geçen arama süresidir. Bu aynı zamanda uygulama için onCreate() metodundaki ekran tasarımı, genel değişken gibi evrensel kaynaklar ve onDestroy() metodundaki uygulama ile ilişkili tüm kaynakların sürümlerini oluşturma süresidir.

3.3.2. Görülebilen Yaşam Süresi

Bu süre, onStart() metoduna yapılan bir aramadan onStop() metoduna yapılan karşılıklı bir aramaya kadar geçen zamandır. Burada aktivite kullanıcıya görünür durumdadır ve durumunu değiştirmeksizin sürdürür.

3.3.3. Ön Plandaki Yaşam Süresi

Bu süre, onResume() metodu ile başlar ve bir karşılığı olan onPause() metoduna kadar sürer. Bu yaşam süresi boyunca aktivite kullanıcıya tamamen görünür durumdadır. Diğer bütün aktivitelerin önüne geçmiştir ve kullanıcıyla etkileşim içindedir.

3.4. Android İşletim Sistemi Mimarisi

Mobil Android parmak izi tanıma sisteminin geliştirilme mimarisi, Java'nın kullanılmasının yanı sıra C/C++ kütüphanesin bulunduğu Dalvik sanal makinesine (DVM - Dalvik Virtual Machine) bağlıdır. Uygulama çerçevesi, donanım, yazılım ve dış kaynaklarla etkileşim gibi cihaz kaynaklarının yönetimi için önceden belirlenmiş hizmetleri içermektedir. Geliştirilen sistem, Java ve Android SDK (yazılım geliştirme kiti) platformu çerçevesinde biyometrik doğrulama alanında gerçek zamanlı hizmet sunan bir dizi Secugen biyometrik kütüphaneleri (Secugen, 2013) uygulanarak gerçekleştirilmiştir. Linux kernel (çekirdek) güvenlik modelleri, hafiza yönetimi ve ağ oluşturma için yeterli bir sürücü temin edilmektedir. Bütün Android işletim sistemli uygulamaların dört ana bileşeni bulunmaktadır.

3.4.1. Aktiviteler

Web sayfalarında bulunan formlar gibi, aktivitelerde tek bir görevi yerine getirmek için bir kullanıcı arayüzü gösterirler. Kullanıcıya bir giriş ekranının gösterilmesi aktivite sınıfına bir örnek olarak verilmektedir.

3.4.2. Hizmetler

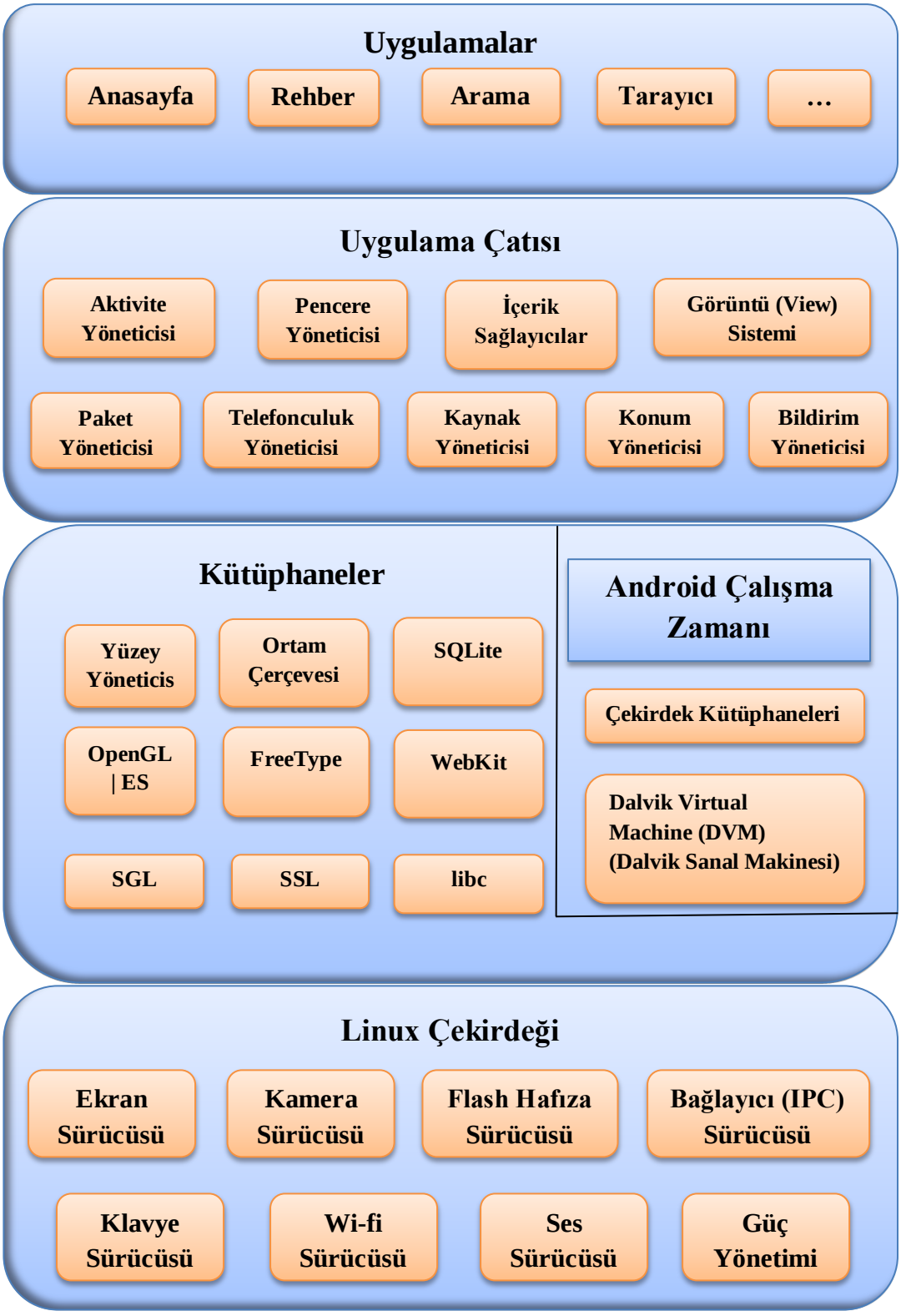
Hizmetler, kullanıcı arayüzüne sahip olmamaları yönüyle Aktivitelerden ayrılmaktadırlar. Hizmetler bazı görevleri yerine getirmek için arka planda çalışmaktadırlar. Hizmet sınıfına, bir elektronik postanın bir internet sunucusundan alınıp getirilmesi örnek verilmektedir.

3.4.3. Radyo Alıcıları

Bu tür bileşenlerin genel amacı ya sistem kodları ya da diğer uygulamalar tarafından yapılan radyo duyurularını almak ve tekrarlamaktır. Örneğin, bir radyo alıcısı bir kablosuz bağlantı alanının (Wi-Fi) kurulduğu bilgisini vermek için yapılmış olabilmektedir.

3.4.4. İçerik Sağlayıcısı

Bu tür bileşenler kendi uygulamalarından diğer uygulamalara veri temin etme görevini yerine getirmektedirler. Bu bileşenler, elektronik posta adresi aramak ve bulduğunda almak için elektronik posta uygulamasına telefonun mevcut kişi listesi uygulamasını kullanmasına izin vermektedirler. Şekil 3.4'de (android-app-market, 2013) Android işletim sistemini oluşturan temel mimarinin bileşenleri gösterilmiştir.



Şekil 3.4. Android işletim sistemi mimarisinin temel bileşenleri

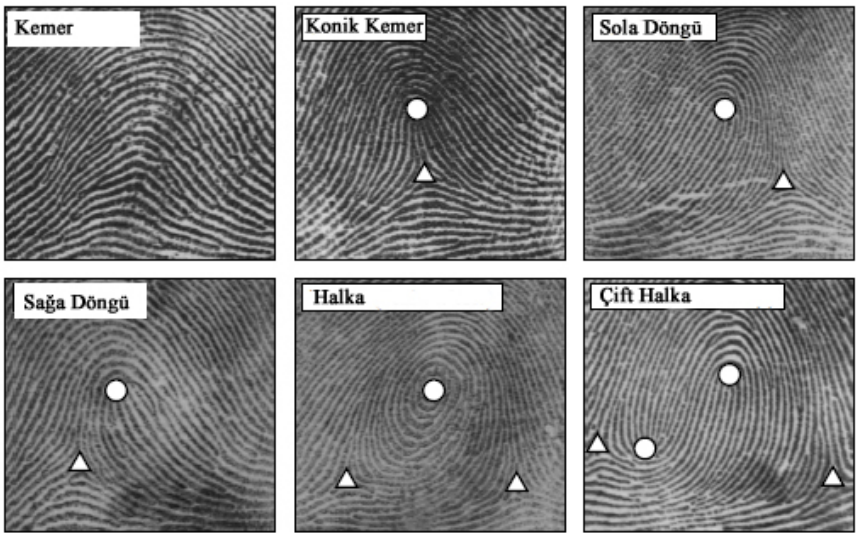
4. KULLANILAN TEKNOLOJİ VE YÖNTEMLER

Parmak izleri, insan parmaklarında var olan grafiksel akış benzeri çıkıntılardan oluşmaktadır. Bu çıkıntılar birkaç yüzyıldan beri kişisel kimlik tanımlamada yaygın olarak kullanılmaktadır. Parmak izlerinin kullanım geçerlilikleri iyi derecede benimsenmiştir. Doğası gereği, mevcut teknolojinin parmak izinden kimlik tanımlamada kullanımı, popüler kişisel kimlik tanımlama yöntemleri olan imza, yüz ve konuşmaya dayalı tanımlama yöntemlerinden çok daha güvenilirdir. Parmak izi doğrulama, genellikle suçlu tespiti gibi adli işlemlerde kullanılmasına rağmen, erişim denetimi, finansal güvenlik, silah satıcıları ve sürücü ehliyeti gibi sivil uygulamalarda da kullanılmaya başlanmıştır. Parmak izi doğrulama işlemleri, profesyonel parmak izi uzmanları tarafından elle yapılmaktadır. Ancak elle parmak izi doğrulama yorucu, zaman alıcı ve pahalı olduğundan, yeni uygulamaların performans gereksinimlerini karşılamaya uygun olmamaktadır. Sonuç olarak otomatik parmak izi tanıma sistemleri (Automatic Fingerprint Identification Systems - AFIS) günümüzde büyük bir talep görmektedir. Otomatik parmak izi tanıma sistemlerinin tasarlanması konusunda, önemli gelişmeler kaydedilmesine rağmen, bir takım tasarım faktörleri (güvenilir Minutiae çıkarma algoritmalarının eksikliği, güvenilir bir niceliksel tanımlama, parmak izi görüntüleri ve parmak izi sınıflandırması esnasındaki karşılaştırma zorlukları) istenilen performansa ulaşmadaki eksiklikleri oluşturmaktadır. Otomatik parmak izi tanıma sistemi aşağıdaki konuların hepsi veya birkaçı ile ilgilenmektedir (Jain ve diğ., 1997).

- Parmak izi toplama: Parmak izi görüntülerinin nasıl elde edileceği ve uygun bir biçimde nasıl temsil edileceği
- Parmak izi doğrulama: İki parmak izinin de aynı parmağa ait olup olmadığının belirlenmesi
- Parmak izi tanıma: Veritabanında parmak izinin sorgulanarak aranması

- Parmak izi sınıflandırma: Verilen bir parmak izinin geometrik görünümlerine göre önceden belirlenmiş olan kategorilerden birine göre atanması

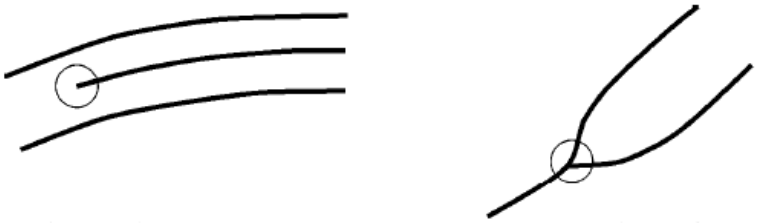
Parmak izini sınıflandırmanın amacı, verilen parmak izlerinin geometrik özelliklerine göre belirli bir kategoriye atanmasıdır. Galton-Henry sınıflandırma şeması birçok ülkede kabul edilmektedir. Dünya çapında kullanılan sınıflama şemaları Galton-Henry sınıflandırma şemasının çeşitleridir. Şekil 4.1'de Henry'nin tanımladığı sınıflar kemer, konik kemer, sola döngü, sağa döngü, halka ve çift halka gösterilmektedir (Maltoni ve diğ., 2009).



Şekil 4.1. Galton-Henry Parmak izi sınıflandırması

Parmak izini sınıflandırmanın temel amacı; büyük parmak izi veritabanlarının yönetimini kolaylaştırmak ve parmak izi eşleştirme sürecini hızlandırmaktır. Genel olarak iyi bilinen Henry sistemi ile belirli bir çerçeve içinde elle parmak izi sınıflandırması yapılmaktadır. Her bir ayrı çerçeve için farklı özellik ayarları kullanılmaktadır. Bununla birlikte, kullanılan çerçeve ne tür olursa olsun, sınıflandırma; iz desenleri, yerel iz yönelimleri ve Minutiae özelliklerine dayanmaktadır. Eğer bu özellikler parmak izi görüntüsünden sayısal ortama aktarılabilirse, parmak izi sınıflandırması çok daha kolay bir şekilde otomatik olarak yapılabilmektedir. Geliştirilen algoritmaların, araştırmalarda parmak izlerini içeren beş ya da altı kategoride orta ölçekli test kümeleri üzerinde yüzde doksan doğrulukla sınıflandırıldığı belirtilmektedir (Karu ve Jain, 1996).

Parmak izi doğrulama, iki parmak izinin aynı parmağa ait olup olmadığının belirlenmesidir. Eğer iki parmak izi de aynı kaynaktansa, yerel iz yapılarının (Minutiae detaylarının) birbiriyle topolojik eşleşmesi beklenmektedir. On sekiz farklı çeşit yerel iz yapıları tespit edilmiştir (Lee ve Gaensslen, 1991). En önemli iki yapı ise Minutiae özellikleri olarak adlandırılan iz sonları ve iz çatallanmalarıdır. İz sonları ve çatallanma özellikleri Şekil 4.2'de gösterilmektedir (Jain ve diğ., 1997).



Şekil 4.2. Minutiae özellik noktaları; iz sonları ve çatallanmalar

Ancak uygulamada;

- Hiçbir uygunluk önceden bilinemez.
- Taslak ve giriş Minutiae özellikleri arasında (dönme ve doğrusal olmayan bozulmalar arasında) birbiriyle bağıntılı bir çeviri vardır.
- Taslaklar ve girişler için gerçek olmayan Minutiae özellikleri mevcuttur.
- Bazı Minutiae özellikleri tespit edilememektedir.

Bu nedenlerle, parmak izi görüntülerinden gelen gerçek olmayan Minutiae özelliklerini algılamak ve oluşan bozulmalardan kurtulmak için Minutiae benzerliklerini otomatik olarak elde edecek ve bu şartlar altında faaliyet gösterecek bir parmak izi doğrulama algoritması kullanılmaktadır. Bu hedefe oldukça zor bir şekilde ulaşılabilmektedir. Aynı parmağa ait iki parmak izi görüntüsünün örnekleriyle bu zorluk Şekil 4.3'de gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.3. Aynı parmağa ait iki farklı parmak izi görüntüsü

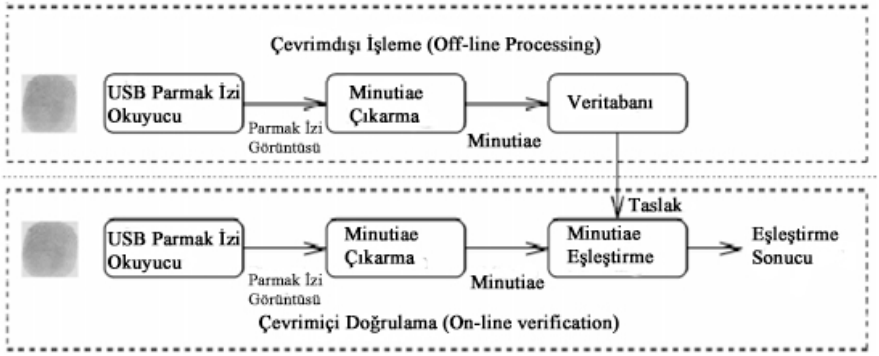
Parmak izi tanıma; bireyin kimliğini belirlemek için alınan parmak izinin, veritabanında bir parmak izi sorgulaması yapılarak eşleştirilmesi sürecini ifade etmektedir. Parmak izi tanımanın amacı, veritabanında yapılan sorgulama sonucu aranan parmak izinin hızlı bir şekilde var olup olmadığının belirlenmesidir. Buradaki kritik konuyu ise erişim hızı ve doğruluğu oluşturmaktadır. Bu problem, bilgisayar görüşü (computer vision) altında çalışan desen tanıma (pattern recognition), veritabanı (database), paralel işleme (parallel processing) gibi birtakım tekniklerle yapılmaktadır. Operasyonel parmak izi erişim sistemleri, çeşitli kolluk kuvvetleri tarafından kullanılmaktadır (Lee ve Gaensslen, 1991).

Bu tez kapsamında, Android tabanlı canlı bir parmak izi tanıma ve kimlik doğrulama sistemi, USB parmak izi okuyucu cihazı kullanımı ile parmak izi görüntüsünün yakalanması ve gerçek zamanlı veritabanında saklanan bilgiler ile karşılaştırılması bir arayüz programı geliştirilerek yapılmıştır.

Böyle bir sistemin kimlik tanıma ve erişim kontrol uygulamaları gibi çok çeşitli kullanım alanları bulunmaktadır. Geliştirilen sistemin genel blok diyagramı Şekil 4.4'de gösterilmektedir (Jain ve diğ., 1997). Sistemin çalışması iki aşamadan oluşmaktadır (Jain ve diğ., 1997).

1- Çevrimdışı işleme (Off-line processing): Kimlik doğrulaması yapılacak kişinin parmak izinden çeşitli izlenimleri yakalanır ve bir özellik çıkarma modülü tarafından işlenerek alınan özellikler daha sonra kullanılmak üzere veritabanı taslağı olarak depolanmaktadır.

2- Çevrimiçi doğrulama (On-line verification): USB parmak izi tarayıcısına bireyin parmağı yerleştirilir. Minutiae noktaları, yakalanan parmak izi görüntüsünden çıkarılır. Daha sonra, Minutiae noktaları bir eşleme modülüyle (veritabanında kendi taslağına karşılık eşleşen sonuçlarla) beslenir ve girilen bireyin kimliği doğrulanır.



Şekil 4.4. Bir parmak izi tanıma sisteminin genel blok diyagramı

Çalışma kapsamında geliştirilen Android tabanlı çevrimiçi parmak izi tanıma sisteminin iki ana bileşeni aşağıdaki gibidir.

Minutiae Özelliklerinin Elde Edilmesi (Minutiae Extraction): Minutiae özellikleri iz sonları ya da iz çatallanmalarından oluşmaktadır. İyi bir bölümlendirme yapılarak inceltilmiş iz haritasındaki tekil noktaların çıkarımı yapılabilmektedir. İyi bir iz haritasının elde edilebilmesi için uluslararası yöntemlerin kullanılması gerekmektedir.

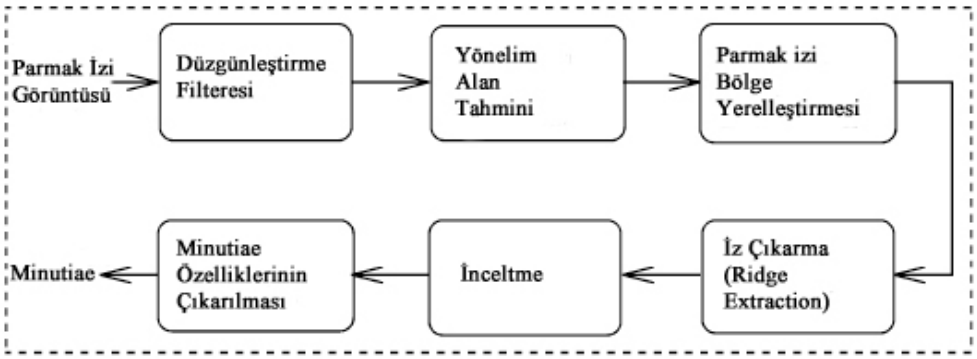
Minutiae Özelliklerinin Eşleştirilmesi (Minutiae Matching): Elde edilen parmak izlerinin nokta desenlerindeki bozulmalar nedeniyle, benzerlikleri önceden bilinmeksizin, yapılan esnek eşlemelerdir. İki nokta deseni arasında hiçbir bozulma yoksa ve nokta desenlerinin bulunduğu yerler tam olarak tespit edilmiş olsa bile, iki nokta deseni arasındaki en uygun eşlemeyi yakalamak oldukça zor olmaktadır (Jain ve diğ., 1997).

Parmak izi görüntüsü verilerinden pürüzsüz bir yönelim alanının elde edilebilmesi için, Minutiae çıkarma performansını artıran bir yöntem kullanılmaktadır (Ratha ve diğ., 1995). Minutiae eşleştirmesi için, hizalama tabanlı esnek bir eşleme algoritması kullanılmaktadır. Bu algoritma, Minutiae özellik noktaları arasında kapsamlı bir araştırma yapmadan, doğrusal olmayan bozulmalar ile farklı parmak izlerinin hatalı görüntü dönüşümleri arasındaki benzerlikleri bulma yeteneğine sahiptir.

4.1. Minutiae Özellik Noktalarının Elde Edilmesi (Minutiae Extraction)

Parmak izi kontrol işlemleri, parmak izi tanıma yapılarak iz yapılarının Minutiae özelliklerinin belirlenmesine dayanmaktadır. Parmak izlerindeki Minutiae özellik noktalarının, bozulma ve yaşlanmayla değişmeyen kendisine özgü bir topolojik yapısı bulunmaktadır (Lee ve Gaensslen, 1991). Bunun sonucunda parmak izi tanıma, Minutiae özellik noktalarının topolojik yapısına dayalı bir benzeşim sunmaktadır. Bu durum, Minutiae özellik noktalarını eşleştirme işlemi için, parmak izi doğrulama sırasında oluşan karışıklığı azaltan sınırlı ölçüdeki giriş noktası desen bozulmalarını tolere edebilecek düzeydeki bir tür nokta deseni eşleşmesinden oluşmaktadır. Bu bağlamda, otomatik parmak izi doğrulama işlemindeki ilk aşama, parmak izlerinden Minutiae özellik noktalarının elde edilmesi işlemidir. Parmak izi doğrulama sisteminde, Ratha ve diğerlerinin önerdiği yöntemin geliştirildiği bir

Minutiae çıkarma algoritması kullanılmaktadır (Ratha ve diğ., 1995). Minutiae özellik noktalarını elde eden algoritmanın genel bir akış şeması Şekil 4.5'de gösterilmektedir (Jain ve diğ., 1997). Burada giriş parmak izi görüntülerinin çözünürlüğü 500 dpi olarak kabul edilmektedir (Jain ve diğ., 1997).



Şekil 4.5. Minutiae özelliklerinin çıkarılma akış şeması

4.2. Yönelim Alanlarının Tahmin Edilmesi

Akış benzeri desenlerin yönelim alanlarının tahmin edilmesinde bir dizi yöntemler kullanılmaktadır. Bu işlemler için Rao algoritmasının yeni bir Hiyerarşik yöntemi kullanılmaktadır. Rao algoritması aşağıdaki ana aşamalardan oluşmaktadır (Rao, 1990).

- Giriş parmak izi görüntüsü WxW büyüklükteki bloklar içine bölünmektedir.
- Her bir bloğun içindeki her bir pikselde G_x ve G_y gradyanları hesaplanmaktadır.
- Her bir bloğun yerel yönelim tahmini (4.3)'deki denklem kullanılarak bulunmaktadır.

Görüntünün her bir pikseli için $G_x(i, j)$ ve $G_y(i, j)$ gradyanları, Sobel filtresi uygulanarak hesaplanabilmektedir. (Dadlani ve diğ., 2006).

$$G_x(i,j) = ((A_2 + KA_3 + A_4) - (A_{0+}KA_7 + A_6))/(K+2) \quad \textbf{(4.1)}$$

$$G_y(i,j) = ((A_0 + KA_1 + A_2) - (A_{6+}KA_5 + A_4))/(K+2) \quad \textbf{(4.2)}$$

Sobel operatöründeki K=2 alındığında, A_i pikselinin anlamı Şekil 4.6'da gösterilmiştir (Dadlani ve diğ., 2006).

A_0	A_1	A_2
A_7	$F(j,k)$	A_3
A_6	A_5	A_4

Şekil 4.6. Sobel operatöründeki pikselin değeri

İz yönelimindeki bir sonraki adımda, WxW büyüklüğünde blok için tüm piksellerin gradyan değerleri kullanılarak iz açısı hesaplanmaktadır (Dadlani ve diğ., 2006).

$$\theta_0 = \frac{1}{2}\tan^{-1}\left(\frac{\sum_{i=1}^{W}\sum_{j=1}^{W} 2\, G_x(i,j)G_y(i,j)}{\sum_{i=1}^{W}\sum_{j=1}^{W}(G_x^2(i,j)-G_y^2(i,j))}\right) \quad \textbf{(4.3)}$$

Bu denklemde: W, bloğun boyutudur; G_x ve G_y sırasıyla x ve y yönlerindeki gradyan büyüklükleridir.

Bununla birlikte, yüksek eğrili izler, parazitler, leke ve iz başlarında kesilmelerin bulunması yerel yönelim alanının hatalı tahminine neden olmaktadır. Bu sınırlamaları aşmak için bir son işlem (postprocessing) sürecinin uygulanması gerekmektedir. Buradaki, tutarsız yönelim alanlarını geliştirmek için aşağıdaki tekrar eden adımlar kullanılmaktadır (Jain ve diğ., 1997).

(i, j) bloklarının yerel bölgelerindeki yönelim alanlarının tutarlılık düzeyleri aşağıdaki formülle hesaplanmaktadır.

$$C_0 = \frac{1}{N}\sqrt{\sum_{(i',j')\in D}|\theta\,(i',j') - \theta(i',j')|^2} \quad \textbf{(4.4)}$$

$$|\theta' - \theta| = \begin{cases} d & \text{eğer } (d = (\theta' - \theta + 360) \bmod 360) < 180 \\ d - 180 & \text{aksi halde} \end{cases} \quad \textbf{(4.5)}$$

Bu denklemlerde: D, Yerel bölge etrafındaki (i, j) bloklarını simgelemektedir. Bu algoritmada D operatörünün boyutu 5x5 olarak kabul edilmektedir. N, D operatörünün içindeki blokların sayısıdır. $\theta(i', j')$ ve $\theta(i, j)$; bloklardaki sırasıyla (i', j') ve (i, j) değerlerinin yerel iz yönelimleridir.

Eğer tutarlılık seviyesi (4.4), belirli bir T_c eşiğinin üzerinde ise, bu bölgedeki yerel yönelimler, belirli bir seviyenin altına kadar düşük çözünürlükte yeniden tahmin edilmektedir. Burada, şema düzenleme işlemi ile daha düzgün bir yönelim alan tahmini elde edilebilmektedir. Solda Rao ve sağda Hiyerarşik yöntemle parmak izi görüntülerinden yönelim alanlarının tahmin edilmesi Şekil 4.7'de gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.7. Parmak izi görüntülerinden yönelim alanlarının tahmin edilmesi

Rao algoritması ile yönelim alanları karşılaştırılarak hiyerarşik bir yöntem kullanılmaktadır. Blok boyutu (WxW), 16x16 ve D'nin boyutu 5x5 alınmaktadır. Parmak izi görüntülerindeki yönelim alanlarının tahmin edilmesinden sonra, gri seviye yerel varyansları temel alan parmak izi görüntüleriyle ilgili bölgeyi bulmak için gruplara ayırma algoritması kullanılmaktadır. Gruplara ayırma algoritması ise, görüntüde sadece bir parmak izinin mevcut olup olmadığı varsayımına dayanmaktadır (Jain ve diğ., 1997).

4.3. İz Algılama (Ridge Detection)

Giriş görüntülerinin yönelim alanları tahmin edilerek parmak izi bölgelerinin yerleri belirlendikten sonra, Minutiae algoritmasında ki bir sonraki işlem basamağı ise iz algılamadır. Parmak izi görüntülerinde bulunan izlere karşılık gelen en belirgin özelliği, izler üzerindeki normal yönelimler boyunca kendi yerel maksimumlarına ulaşmasıdır. Bunun için piksellerin, bu özelliklere dayalı iz pikselleri olduğu tespit edilmiştir (Jain ve diğ., 1997).

Parmak izlerinin önemli bir özelliği, iz üzerindeki gri-seviyeli değerlerin (iz yoğunluğu) yerel iz yönüne çapraz yönde yerel maksimuma ve çatalların (izin ikiye bölünmesi) gri-seviyeli değerlerinin aynı yönde yerel minimuma gitmesidir (Jain ve diğ., 1997). Bu özellik kullanılarak görüntü üzerindeki noktanın iz olduğu tespit edilmektedir.

Görüntüye $h_t(x, y; u, v)$ ve $h_b(x, y; u, v)$ maskeleri L x H (11 x 7) değerleriyle uygulanmaktadır. Bu iki maske, yerel iz yönünde maksimum yerel gri seviyeli değerlerin tespitini sağlamaktadır (Jain ve diğ., 1997).

$$h_t(x,y;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}},\ u = \left(v.\tan\left(\theta(x,y)\right) - \frac{H}{2.\cos\left(\theta(x,y)\right)}\right), v \in \Omega \text{ ise} \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}},\ u = \left(v.\tan\left(\theta(x,y)\right)\right), v \in \Omega \text{ ise} \\ 0\ ,\quad \text{diğer durumlarda} \end{cases} \quad \textbf{(4.6)}$$

$$h_b(x,y;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}},\ u = \left(v.\tan\left(\theta(x,y)\right) + \frac{H}{2.\cos\left(\theta(x,y)\right)}\right), v \in \Omega \text{ ise} \\ \frac{1}{\sqrt{2\pi}\delta} e^{-\frac{u}{\delta^2}},\ u = \left(v.\tan\left(\theta(x,y)\right)\right), v \in \Omega \text{ ise} \\ 0\ ,\quad \text{diğer durumlarda} \end{cases} \quad \textbf{(4.7)}$$

Denklemde yer alan Ω değeri aşağıda (4.8)'de verilen denklemdeki aralığı kapsamaktadır (Jain ve diğ., 1997).

$$\Omega = \left[-\left|\frac{L.\sin\left(\theta(x,y)\right)}{2}\right|, \left|\frac{L.\sin\left(\theta(x,y)\right)}{2}\right|\right] \quad \textbf{(4.8)}$$

Burada: $\theta(x, y)$; (x, y) Noktasındaki yerel iz yönüdür.

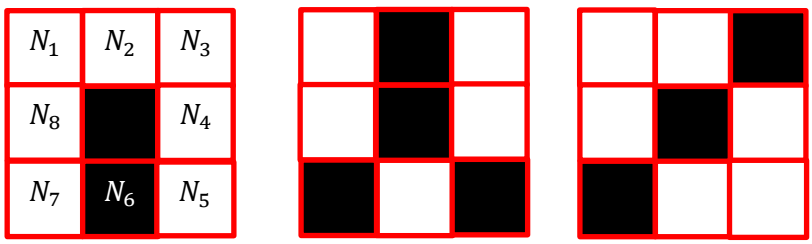
Eğer (x, y) noktasındaki gri-seviye değeri belirli bir T_{iz} eşik değerinden yüksek ise, (x, y) noktası iz olarak işaretlenmektedir (Jain ve diğ., 1997).

Bu algoritma kullanılarak, yerel iz genişliğinin, maske genişliğine uyarlanmasıyla, parmak izi görüntüsündeki izlerin yeri saptanmaktadır. Bununla birlikte oluşan ikili iz haritasında, görüntü girdisindeki iz lekeleri, kırılmalar ve parazitler nedeniyle, sık sık çukurlar ve benekler oluşmaktadır. İz iskeletlerinin özellik noktalarının tespitinde kullanılmasından beri, bu tür çukur ve beneklerin varlığı, özellik çıkarma algoritmasında ciddi performans eksikliğine neden olmaktadır. Bu benek ve çukurlar iz iskeletlerini büyük ölçüde değiştirebilmektedirler. Bu nedenle, benek ve çukurları ortadan kaldırma yönteminin iz inceltmeden önce uygulanması gerekmektedir. Bu aşamalardan sonra girilen bir

parmak izi görüntüsü üzerinde, parmak izinden elde edilen nispeten pürüzsüz bir iz haritası elde edilmektedir. Özellik noktalarını algılama algoritmasındaki bir sonraki adım ise özellik noktalarının bulunması ve iz haritalarının inceltilmesi işlemidir (Jain ve diğ., 1997).

4.4. Özellik Noktalarının Algılanması (Minutiae Detection)

İnceltilmiş iz haritası elde edildiğinde, özellik noktalarının algılanması, basit bir konu haline gelmektedir. Bir noktanın (sekiz ilişkili), inceltilmiş iz üzerinde olduğu varsayılırsa konum bilgisi 1 değerini, aksi durumda 0 değerini almaktadır. İnceltilmiş izler üzerindeki noktalarını (x, y) değerleri belirtmektedir. N_0, N_1, ..., N_7 değerleri ise bu noktanın çevresindeki 8 komşuluk noktasını göstermektedir. Eğer bu toplam = 1 ise $(\sum_{i=0}^{8} N_i) = 1$, (x, y) noktası iz sonu olarak ifade edilmektedir. Eğer bu komşuluk toplamı > 2 ise $(\sum_{i=0}^{8} N_i > 2)$, (x, y) noktasında iz çatallanması olduğu kabul edilmektedir. Fakat bunların dışında, bu komşuluk toplamı = 2 ise $(\sum_{i=0}^{8} N_i = 2)$, (x, y) noktasının iz pikseli üzerinde olduğu tespit edilmektedir. Şekil 4.8'de (Dadlani ve diğ., 2006) iz sonu, çatallanma ve iz pikseli tespitine bir örnek verilmektedir. Bununla birlikte, inceltilmiş iz haritasındaki istenmeyen sivriliklerin ve kırılmaların varlığı, birçok sahte özellik noktası tespitine sebep olabilmektedir. Bu yüzden özellik noktası tespitinden önce, sivri ve kırılan izlerin eklenmesini ortadan kaldırmak için düzeltme işlemleri yapılmaktadır (Jain ve diğ., 1997).



Şekil 4.8. İz sonu, çatallanma ve iz pikseli

İz düzeltme algoritmasında aşağıdaki işlem kullanılmaktadır.

- Eğer iz haritalarının bir kolu dikeyse, yerel iz yönleri ve boylarını belirtilen T_b eşiğinden daha kısa olması için ortadan kaldırılmaktadır.
- Eğer izdeki bir kırılma yeterince kısa veya diğer izler bu izin içerisinden geçiyorsa bu kırılma diğer izlere bağlanmaktadır.

Yukarıdaki yöntemler ile sahte Minutiae özelliklerinin büyük bir kısmı silinmesine rağmen, birçok sahte Minutiae özellikleri halen bulunabilmektedir. Bunun nedeni yukarıdaki işlemlerin yerel iz bilgilerine dayanmasıdır. Eğer bu bilgilere güvenilmezse, yukarıdaki yöntemlerle gerçek Minutiae özelliklerinden sahte Minutiae özelliklerini ayırt etmenin yolu bulunmamaktadır. Bu yüzden yapısal bilgilere dayanan bir filtreleme işlemi gerekmektedir. Sahte Minutiae özelliklerini ortadan kaldırmak için aşağıdaki kurallara dayanan düzeltme algoritması kullanılabilmektedir (Jain ve diğ., 1997).

- Eğer küçük bir bölgede kümelenmiş birkaç Minutiae özellikleri varsa, bunlardan küme merkezine en yakınları hariç geri kalanların hepsi ortadan kaldırılmaktadır.
- Eğer iki Minutiae özelliği birbirine yeterince yakın yerleştirilmişse, her ikisi de çıkarılmaktadır.

Aşağıdaki resimlerde, mürekkepsiz bir tarayıcı ile (512x512) çözünürlükte yakalanmış bir parmak izi görüntüsünden, Minutiae özelliklerinin çıkarılma sonuçları gösterilmektedir. Şekil 4.9'da giriş parmak izi görüntüsünden (solda), yönelim alanlarının elde edilmesi (sağda) gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.9. Giriş parmak izi ve yönelim alanları bulunmuş parmak izi

Şekil 4.10'da yerelleştirilmiş parmak izi bölgesinden (solda), elde edilmiş iz haritası (sağda) gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.10. Yerelleştirilmiş parmak izi bölgesi ve çıkarılmış iz haritası

Şekil 4.11'de giriş parmak izi görüntüsünden (solda), üst üste bindirilmiş yönelimler ve Minutiae özelliklerinin çıkarılması (sağda) gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.11. İnceltilmiş iz haritası ve çıkarılmış Minutiae özellikleri

Yukarıdaki görüntü düzeltme işlemleri gerçekleştirildikten sonra, geriye kalan Minutiae özellikleri doğru Minutiae özellikleri olarak işlenmektedir. Bu uygulanan yöntemlere rağmen, her bir Minutiae özelliği için kesin bir yer garanti edilememekle birlikte birkaç sahte Minutiae özelliği silinebilmektedir. Geriye kalan her bir Minutiae özelliği için aşağıdaki parametreler kaydedilmektedir (Jain ve diğ., 1997).

- X-Koordinatı,

- Y-Koordinatı,
- İz ilişkili yerel izin yönü gibi tanımlanan yönelimler,
- İlişkilendirilmiş iz.

Kaydedilen izler, iz içi uzaklık ortalaması tarafından normalleştirilen, tek boyutlu küçük sinyaller gibi temsil edilmektedir. Kaydedilen bu izler, Minutiae eşleştirme aşamasında hizalama işlemi için kullanılmaktadır (Jain ve diğ., 1997).

4.5. Minutiae Özelliklerinin Eşleştirilmesi (Minutiae Matching)

Otomatik parmak izi doğrulama ve tanıma sistemlerinde, akıllı piksel eşleştirme ve parmak izi görüntülerindeki iz deseni eşleştirme işlemleri yerine, nokta deseni eşleştirme (Minutiae Matching) algoritması kullanılmaktadır (Ton ve Jain, 1989). Genel bir nokta eşleştirme işleminde algoritma, arama yolları üstel sayısını azaltmak için kullanılan noktalar arasındaki bağıl mesafeler gibi her nokta ve noktanın bölgesel özellikleri ile ilişkilendirilmiş özellikleri oldukça dirençli olmaktadır. Esneklik yöntemiyle her bir çifte karşılık gelen güven seviyesi (diğer çiftlerin belirli bir ölçüte kadar kabul edilmesine dayanan güven seviyesi) tekrarlanarak ayarlanmaktadır (Ranade ve Rosenfeld, 1993). Bu algoritma, eşleme karışıklığını azaltmak için önerilmiş bir dizi güncel sürümüne rağmen tekrarlanan yapıları gereği oldukça yavaş kalmaktadır (Ton ve Jain, 1989).

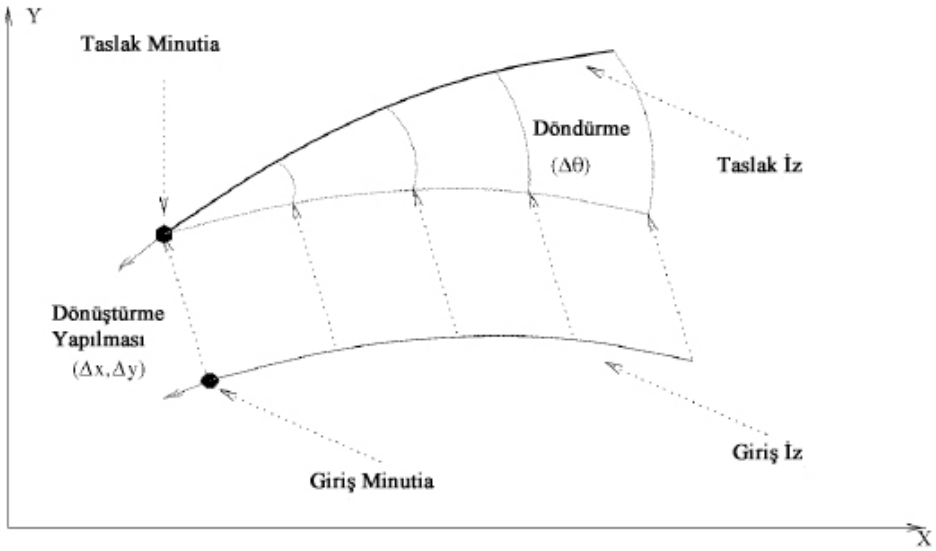
Bu nedenlerle hizalama tabanlı bir eşleme algoritması kullanılmaktadır. Çünkü bu algoritma, hızlı çalışmakta ve veri ayrışımını çok kolay yapılabilmektedir. Hizalama tabanlı eşleme algoritması, Minutiae eşlemeyi iki aşamaya ayırmaktadır (Jain ve diğ., 1997).

- Hizalama aşaması; Veritabanındaki taslak ile giriş arasındaki ölçekleme, rotasyon, çevirim gibi dönüşümleri tahmin etmekte ve giriş Minutiae özelliklerini, taslak Minutiae özelliklerine göre tahmini parametrelerle hizalamaktadır.
- Eşleme aşaması; Kutupsal koordinat sistemindeki çokgenlere dönüştürülerek, bir dizi esnek eşleme algoritması elde edilen çokgenlerin eşlenmesinde kullanılmaktadır.

4.6. Nokta Desenlerinin Hizalanması

Düzlemsel nokta desenlerinin iki kümesine karşılık gelen iki nokta çifti tam olarak hizalanabilmektedir. İki nokta deseni arasında doğru bir hizalama işlemi, olası tüm ilgili nokta çiftlerinin test edilmesi ve uygun bir seçme işleminin elde edilmesiyle mümkün olmaktadır. Bununla birlikte, parazit ve bozulmaların varlığı sebebiyle, bu taslaklar her zaman tam uyumlu olarak hizalanamamaktadır. İki nokta deseni arasındaki görüntü dönüşümlerini doğru ve tam olarak değerlendirmek amacıyla, karşılık gelen nokta çiftlerine nispeten büyük numaralar verilmesi gerekmektedir. Bu ise çok sayıda yanlış eşleşmelerin test edilmesine sebep olmaktadır. Bu yüzden nokta çiftlerine karşılık gelen bir hizalama, mümkün olmasına rağmen pratik bulunmamaktadır (Jain ve diğ., 1997).

Eğri parçalarına karşılık gelen iki nokta deseni, parazit ve bozulmaların olmasına rağmen yüksek bir doğrulukla hizalanma yeteneğine sahiptir. Parmak izindeki her bir Minutiae bir iz ile bağlantılıdır. İzlere karşılık gelen hizalamayla gerçek bir hizalanmanın elde dilebilmesi Şekil 4.12'de gösterilmektedir (Jain ve diğ., 1997).



Şekil 4.12. Giriş ve taslak izlerin hizalanması

Minutiae algılama aşamasında, bir Minutiae çıkarıldığında ve kaydedildiğinde daima bulunduğu iz üzerinde tutulmaktadır. Bu iz, Minutiae ile çakışan kaynağı düzlemsel bir eğri olarak temsil etmektedir. X-koordinatı Minutiae yönü ile aynı yönlü olmalıdır. Ayrıca bu düzlemsel eğri, ara iz mesafesi

ortalamasıyla denkleştirilmektedir. Bu izlerin eşleşmesiyle, taslak ve giriş parmak izi arasındaki bağıl dönüşüm doğru bir şekilde tahmin edilebilmektedir. Kendisine özgü olan R^d ve R^D simgeleri, sırasıyla Minutiae özelliklerindeki izle ilişkili olan kümelerdeki giriş görüntülerini ve taslaklarını göstermektedir (Jain ve diğ., 1997).

İki nokta kümesini hizalamak için genelde aşağıdaki algoritma kullanılmaktadır (Jain ve diğ., 1997).

1- Her iz $d \in R^d$ şeklinde tek boyutlu ayrık sinyal olarak temsil edilmektedir. Her bir ize karşılık $D \in R^D$ değeri aşağıdaki formüle göre eşleşmektedir.

$$S = \frac{\sum_{i=0}^{L} d_i D_i}{\sqrt{\sum_{i=0}^{L} d_i^2 D_i^2}} \quad \textbf{(4.9)}$$

Burada, L, iki izin arasındaki minimum uzunluktur; D_i, d_i, mesafeleri, sırasıyla x ekseni için d ve D izlerindeki i ninci noktaları temsil etmektedir.

İz üzerindeki örnekleme aralığı, ortalama izler arası mesafeyi ayarlamaktadır. Eğer eşleme puanı $S\ (0 \leq S \leq 1)$ arasında belirli bir T_r eşik değerinden daha büyük ise, ikinci aşamaya geçilmekte, aksi halde bir sonraki iz çifti eşlemesine devam edilmektedir.

2- İki iz arasındaki görüntü dönüşümünün tahmin edilmesi Şekil 4.12'de gösterilmektedir (Jain ve diğ., 1997). Genel olarak en küçük kareler yöntemi, görüntü dönüşümünün tahmin edilmesi için kullanılmaktadır.

İki iz arasına karşılık gelen çeviri vektörü $(\Delta x, \Delta y)^T$, aşağıdaki denklemle hesaplanmaktadır (Jain ve diğ., 1997).

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^d \\ y^d \end{pmatrix} - \begin{pmatrix} X^D \\ Y^D \end{pmatrix} \quad \textbf{(4.10)}$$

Buradaki $(x^d, y^d)^T$ ve $(x^D, y^D)^T$ vektörleri, sırasıyla d ve D izleriyle ilişkilendirilmiş ve Referans Minutiae değeri olarak bilinen iki Minutiae özelliğinden gelen x ve y koordinatlarını oluşturmaktadır.

İki iz arasındaki dönme açısı $\Delta\theta$, şu formülle hesaplanmaktadır (Jain ve diğ., 1997).

$$\Delta\theta = \frac{1}{L}\sum_{i=0}^{L}(\gamma_i - \Gamma_i) \quad \textbf{(4.11)}$$

Burada, L, d ve D izleri arasındaki en kısa mesafeyi göstermektedir; $\gamma_i - \Gamma_i$, sırasıyla d ve D izleriyle, Referans Minutiae değerine göre ilişkilendirilmiş izler üzerindeki i ninci noktanın radyal açısıdır. Bu açı, giriş ve taslak görüntüler arasında ölçeklendirme faktörü olarak kabul edilmektedir. Çünkü parmak izi görüntüleri, çevrimiçi ve çevrimdışı çalışma aşamalarında aynı cihazla yakalanmaktadır.

3- Minutiae özellik değerleri $(x^d, y^d, \theta^d)^T$, Referans Minutiae değerleri gibi tahmin edilen görüntü dönüşüm parametrelerine dayalı olarak ifade edilmektedir. Bütün N giriş Minutiae değerlerinin çevirimi ve dönüştürülmesi bu Referans Minutiae değerlerine göre aşağıdaki formülle yapılmaktadır (Jain ve diğ., 1997).

$$\begin{pmatrix} x_i^A \\ y_i^A \\ \theta_i^A \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix} + \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ \sin\Delta\theta & -\cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i - x^d \\ y_i - y^d \\ \theta_i - \theta^d \end{pmatrix} \quad \textbf{(4.12)}$$

Burada, $(x_i, y_i, \theta_i)^T$, (i=1, 2, …, N), giriş Minutiae değerini temsil etmektedir; $(x_i^A, y_i^A, \theta_i^A)^T$, hizalanmış Minutiae değerine karşılık gelmektedir.

4.7. Sıraya Dizilen Nokta Desenlerinin Eşleştirilmesi

Eğer iki nokta deseni, birbirleriyle tam olarak hizalanmışsa, karşılık gelen noktaların her bir çifti tamamen birleşmektedir. Böyle bir durumda nokta desen eşlemesi, örtüşen çift sayısını sayarak yapılmaktadır. Fakat uygulamada böyle bir durumla karşılaşılmamaktadır. Minutiae özelliklerinin yerleştirilmesi ve belirlenmesi sırasında oluşan hata, bağıl görüntü dönüşümünü kurtarmak için hizalama algoritmasını engellemektedir. Bununla birlikte, hizalama taslağında parmak izi gösterimindeki doğal bir özellik olan, parmak izlerindeki bozulmalar doğrusal modellerden değildir. Bu tür doğrusal olmayan bozulmaların varlığı, her bir giriş Minutiae özelliğinin taslakta onun karşılığına gelen Minutiae özelliğine göre

konumunun korunmasını engellemektedir. Bu nedenlerle, Minutiae görüntü istasyonlarının ve doğrusal olmayan bozulmaların tolere edilebilir olması için nokta desen eşleme algoritmasının hizalanması esnek olmalıdır. Genelde bu tür esnek eşlemeler, her bir taslak Minutiae çerçevesinin özelliğine göre giriş Minutiae özelliğinde ona karşılık gelen bütün olası özellik noktalarına sınırlayıcı kutu (bounding box) yerleştirilmesiyle elde edilmektedir. Bu kutu, taslak özellikli noktaya göre giriş özellikli noktanın alabileceği yerleri belirtmektedir. Giriş özellikli noktaya karşılık gelen Minutiae özellik noktaları bu kutu içinde kalacak şekilde sınırlandırılmaktadır (Ratha ve diğ., 1995).

Bu yöntem, uygulamada iyi bir sonuç sağlamamaktadır. Çünkü genel biçimsel bozukluklar oldukça büyük olurken, yerel biçimsel bozukluklar küçük olabilmektedir. Özellikli nokta yer belirleme hataları ve doğrusal olmayan biçim bozukluklarını giderme özelliği olan bir uyarlamalı esnek eşleştirme algoritması uygulanmalıdır (Jain ve diğ., 1997).

Taslak özellikli noktaların Minutiae kümesi

$$P = \left((x_1^P, y_1^P, \theta_1^P)^T, \dots, (x_M^P, y_M^P, \theta_M^P)^T\right)$$

ile, verilen Minutiae Referans noktalarına göre yukarıdaki taslağın hizalanmasını sağlayan giriş görüntülerindeki N Minutiae kümeleri ise

$$Q = \left((x_1^Q, y_1^Q, \theta_1^Q)^T, \dots, (x_N^Q, y_N^Q, \theta_N^Q)^T\right)$$

ile ifade edilebilmektedir.

Esnek eşleme algoritması aşağıdaki aşamalardan oluşmaktadır (Jain ve diğ., 1997).

1- Her bir özellik noktası kutupsal koordinat sistemine çevrilmektedir.

$$\begin{pmatrix} r_i \\ e_i \\ \theta_i \end{pmatrix} = \begin{pmatrix} \sqrt{(x_i^* - x^r)^2 + (y_i^* - y^r)^2} \\ \tan^{-1}\left(\frac{y_i^* - y^r}{x_i^* - x^r}\right) \\ \theta_i^* - \theta^r \end{pmatrix} \quad \textbf{(4.13)}$$

Burada, $(x_i^*, y_i^*, \theta_i^*)$, Minutiae özelliklerinin koordinatlardır; $(x^r, y^r, \theta^r)^T$, referans Minutiae özelliklerinin koordinatlarıdır; $(r_i, e_i, \theta_i)^T$, polar koordinat sistemindeki Minutiae özelliklerini temsil etmektedir; r_i, radyal mesafeyi; e_i, radyal açıyı; θ_i, referans Minutiae özelliğine göre Minutiae yönelimini temsil etmektedir.

2- Kutupsal koordinat sistemindeki taslak ve giriş Minutiae özellik noktaları, her Minutiae özellik noktasının Radyal açıları, artan sırayla birleştirilerek sembolik dizi olarak temsil edilmektedir (Jain ve diğ., 1997).

$$P_p = \left((r_1^P, e_1^P, \theta_1^P)^T, \dots, (r_M^P, e_M^P, \theta_M^P)^T\right) \quad \textbf{(4.14)}$$

$$Q_p = \left((r_1^Q, e_1^Q, \theta_1^Q)^T, \dots, (r_N^Q, e_N^Q, \theta_N^Q)^T\right) \quad \textbf{(4.15)}$$

$(r_*^P, e_*^P, \theta_*^P)$ ve $(r_*^Q, e_*^Q, \theta_*^Q)$: Sırasıyla yarıçapı, radyal açıyı ve Referans Minutiae özellik noktasına göre Minutiae özellik noktasının yön uyarlanmasını ifade etmektedir.

3- Bir dinamik programlama algoritmasıyla P_p ve Q_p arasında aşağıda açıklanan düzeltme uzaklığına erişebilmek için $P_p$ve Q_p dizeleri karşılaştırılmaktadır (Cormen ve diğ., 1990).

4- $P_p$ve Q_p arasında, Minutiae eşleşmelerini oluşturmak için mesafe düzenlemesi kullanılır. Karşılaştırma puanı M_{pq}, aşağıdaki formüle göre hesaplanmaktadır (Jain ve diğ., 1997).

$$M_{pq} = \frac{100N_{pair}}{\max\{M,N\}} \quad \textbf{(4.16)}$$

Bu denklemde, N_{pair}, taslak Minutiae özellik noktalarının sınırlayıcı kutulara düşen detay sayısıdır. Karşılaştırma puanı genelde minimum = 1, maksimum = 100 değerleri aralığında alınmaktadır. Eğer sonraki değerlerde hiçbir eşleşme oluşmazsa, eski değerler en iyi eşleştirmeleri göstermektedir (Jain ve diğ., 1997).

Kutupsal koordinat sisteminde Minutiae eşleştirmenin birçok avantajı vardır. Radyal özelliklere sahip parmak izlerindeki doğrusal olmayan bozulmalar gözlenebilmektedir. Parmak izi baskısındaki doğrusal olmayan bozulmalar, genellikle belirli bir bölgeden başlar ve dışa doğru doğrusal olmayan bir şekilde

yayılmaktadır. Bu nedenle kutupsal alan içinde modellenmektedir. Aynı zamanda Kartezyen uzay içindeki kutupsal alanda, giriş görüntüsü ile taslak arasındaki hizalama hatalarının ana bölümünü oluşturan rotasyonu formülüze etmek kolaylaşmaktadır. Nokta desenlerini temsil eden kutupsal koordinatlarda, noktaların radyal açılarının artan sırayla birleştirilmesiyle, sembolik bir dize oluşturulmaktadır. Bunun sonucunda, nokta desenlerinin eşleşmesi bir dize eşleme algoritması ile yapılmaktadır (Cormen ve diğ., 1990). Burada, bir dize eşleme algoritmasının içerdiği esnek bir ölçütle ilgilenilmektedir. Genel olarak bir dize eşleme, uzaklık düzenlemesine benzeyen belirli bir maliyet fonksiyonunun maksimizasyonu ya da minimizasyonu gibi düşünülmektedir. Dize eşleme algoritmasının maliyet fonksiyonunda esnek bir terimi ile bir miktar hata tolere edilebilmektedir. M ve N uzunluklarının verilen iki $P_p$ve Q_p dizeleri sırasıyla, düzenleme uzaklığı ve $C(M,N)$ değeridir. Kullanılan algoritma, aşağıdaki denklemler ile ardışık olarak tanımlanmaktadır (Jain ve diğ., 1997).

$$C(M,N)=\begin{cases}0 & \text{eğer } m=0, \text{veya } n=0\\ \min\begin{cases}C(m-1,n)+\Omega\\ C(m,n-1)+\Omega\\ C(m-1,n-1)+w(m,n)\end{cases} & 0<m\le M, \text{ve } 0<n\le N\end{cases} \quad \textbf{(4.17)}$$

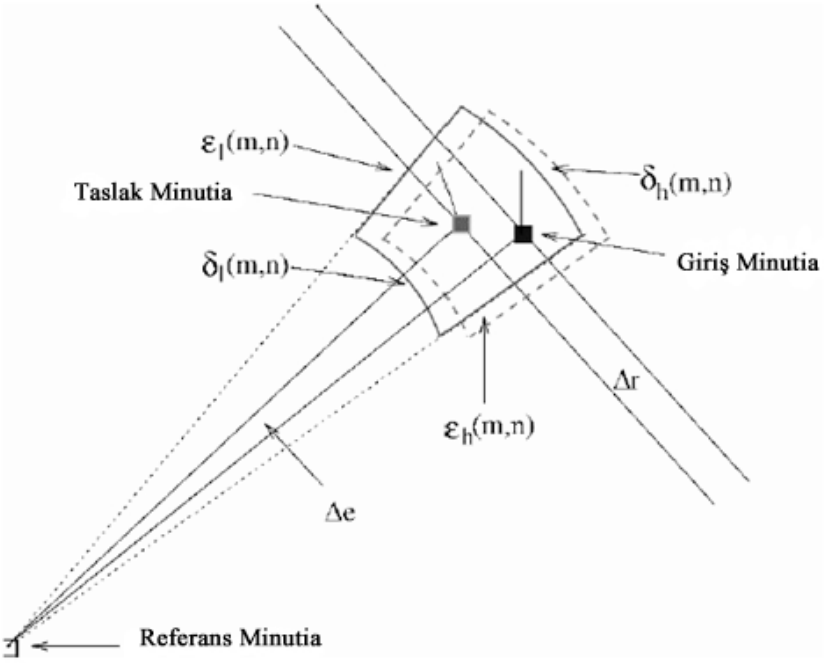
$$w(m,n)=\begin{cases}\alpha\left|r_m^P-r_n^Q\right|+\beta\Delta e+\gamma\Delta\theta & \text{eğer } \left|r_m^P-r_n^Q\right|<\delta, \Delta e<\epsilon \text{ ve } \Delta\theta<\varepsilon\\ \Omega & \text{diğer durumlarda}\end{cases} \quad \textbf{(4.18)}$$

$$\Delta e=\begin{cases}a & \text{eğer } \left(a=\left(e_m^P-e_n^Q+360\right)\bmod 360\right)<180\\ a-180 & \text{diğer durumlarda}\end{cases} \quad \textbf{(4.19)}$$

$$\Delta\theta=\begin{cases}a & \text{eğer } \left(a=\left(\theta_m^P-\theta_n^Q+360\right)\bmod 360\right)<180\\ a-180 & \text{diğer durumlarda}\end{cases} \quad \textbf{(4.20)}$$

Bu denklemlerde, α,β ve γ, sırasıyla δ,ϵ ve ε sınırlayıcı kutularını belirten her bir birleşen ile ilişkili bağıl değerlerdir; Ω ise eşleşmeler için önceden belirlenmiş bir sınır değeridir. Bunun gibi bir uzaklık düzenlemesi ile dize eşlemedeki esnek özellikler yakalanmaktadır. Bununla birlikte bu şema, Minutiae özellikleri ve doğrusal olmayan bozulmaların hatalı bölgelerinde oluşan eşleştirmelerin ters etkisini azaltabilmekte ancak dengeleyememektedir. Bu yüzden yerel doğrusal

olmayan bozulmaları, hatalı hizalama ve minimize işlemi esnasında, azaltma denemesi işlemi için uyarlanabilir bir mekanizma gerekmektedir. Bu uyarlamalı mekanizma ile Minutiae özellik noktalarının çevirim sırasının ayarlanması beklenmemektedir. Bu algoritma ile eşleşme işlemi esnasında hatalı bir eşleme bulunduğunda, sınırlayıcı kutu ayarlamasıyla bir uyum elde edilmektedir. Bu durum Şekil 4.13'de (Jain ve diğ., 1997) gösterilmektedir. Bu durum (4.21-4.26) denklemleriyle temsil edilmektedir (Jain ve diğ., 1997).



Şekil 4.13. Sınırlayıcı kutu

$$w'(m,n) = \begin{cases} \alpha\left|r_m^P - r_n^Q\right| + \beta\Delta e + \gamma\Delta\theta & \text{eğer} \begin{cases} \delta_1(m,n) \\ <(r_m^P - r_n^Q) \\ < \delta_h(m,n) \\ \\ \epsilon_1(m,n) \\ < \Delta e \\ < \epsilon_h(m,n) \\ \\ \Delta\theta < \varepsilon \end{cases} \\ \Omega & \text{diğer durumlarda} \end{cases} \quad \textbf{(4.21)}$$

$$\begin{pmatrix}\Delta r_a\\ \Delta e_a\end{pmatrix} = \begin{cases} \begin{pmatrix} r_m^P - r_n^Q \\ \Delta e \end{pmatrix} & \text{eğer} \begin{cases} \delta_1(m,n) \\ < (r_m^P - r_n^Q) \\ < \delta_h(m,n) \\ \\ \in_1 (m,n) \\ < \Delta e \\ < \in_h (m,n) \\ \\ \Delta\theta < \varepsilon \end{cases} \\ 0 & \text{diğer durumlarda} \end{cases} \quad \textbf{(4.22)}$$

$$\delta_1(m+1,n+1) = \delta_1(m,n) + \eta\Delta r_a \quad \textbf{(4.23)}$$

$$\delta_h(m+1,n+1) = \delta_h(m,n) + \eta\Delta r_a \quad \textbf{(4.24)}$$

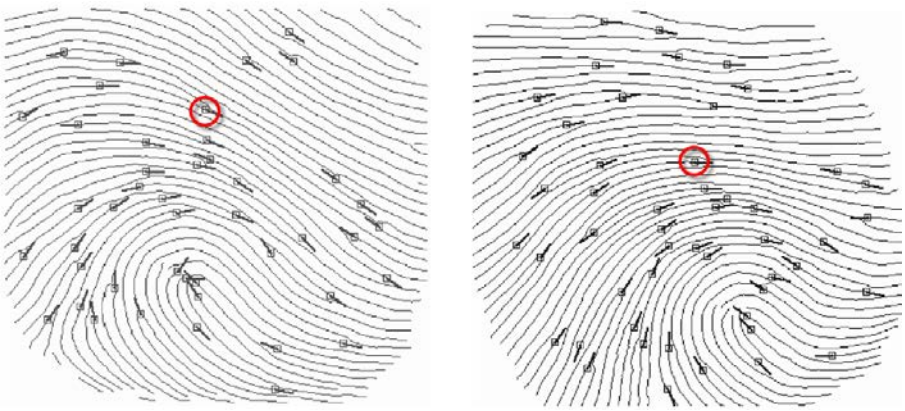
$$\in_1 (m+1,n+1) = \in_1 (m,n) + \eta\Delta e_a \quad \textbf{(4.25)}$$

$$\in_h (m+1,n+1) = \in_h (m,n) + \eta\Delta e_a \quad \textbf{(4.26)}$$

Burada, $(r_m^P, e_m^P, \theta_m^P)^T$ ve $(r_n^Q, e_n^Q, \theta_n^Q)^T$ Minutiae özellik çiftlerinin eşleşmesi için sınır değerleridir; $\delta_1(m,n)$, $\delta_h(m,n)$, kutupsal koordinat sistemindeki (yarıçap radyal açıları); $\in_1 (m,n)$ ve $= \in_h (m,n)$ uyarlamalı sınırlayıcı kutuları belirtmektedir; η, öğrenme oranını (hızını) belirtmektedir.

Bu esnek dize eşleme algoritması, performans için kritik olan birkaç parametreye sahiptir. Bu parametre değerleri parmak izi çözünürlüğüne göre değişmektedir (Jain ve diğ., 1997).

Taslak ve giriş Minutiae özellik kümeleri için eşleme algoritması uygulama sonuçları aşağıdaki şekillerde gösterilmektedir. Şekil 4.14'de solda giriş Minutiae ve sağda taslak Minutiae özellik noktaları gösterilmektedir (Jain ve diğ., 1997).



Şekil 4.14. Giriş ve taslak Minutiae özellik noktaları

Şekil 4.15'de çerçeveler ile işaretlenmiş Minutiae özelliklerine dayalı hizalama sonuçları solda gösterilmektedir. Çizgiler ile bağlanmış taslak Minutiae özellikleri ve onların ilişkileri arasındaki eşleştirme sonuçları ise sağda gösterilmektedir (Jain ve diğ., 1997).

Şekil 4.15. Minutiae özelliklerine dayalı hizalama ve eşleştirme sonuçları

4.8. Parmak İzi Okuma Teknolojileri

Bu çalışma kapsamında geliştirilen parmak izi tanıma sisteminde özellik noktalarının yer ve sırasının (Minutiae) elde edilmesi yöntemi kullanılmıştır. Minutiae yönteminin doğru sonuçlar verebilmesi, gerekli bilgilerin düzgün bir biçimde elde edilmesini gerektirmektedir. Parmak izinden gerekli özellikleri alabilmek için görüntü işleme teknikleri uygulanmaktadır. Günümüzde kullanılan ve geliştirilen teknikler şunlardır (Maltoni ve diğ., 2009).

- CCD veya CMOS kameralı optik sensörler
- Ultrasonik sensörler
- Katı hal elektrik alan sensörler
- Katı hal kapasitif sensörler
- Katı hal sıcaklık sensörler

4.9. Parmak İzi Okuma Yöntemi

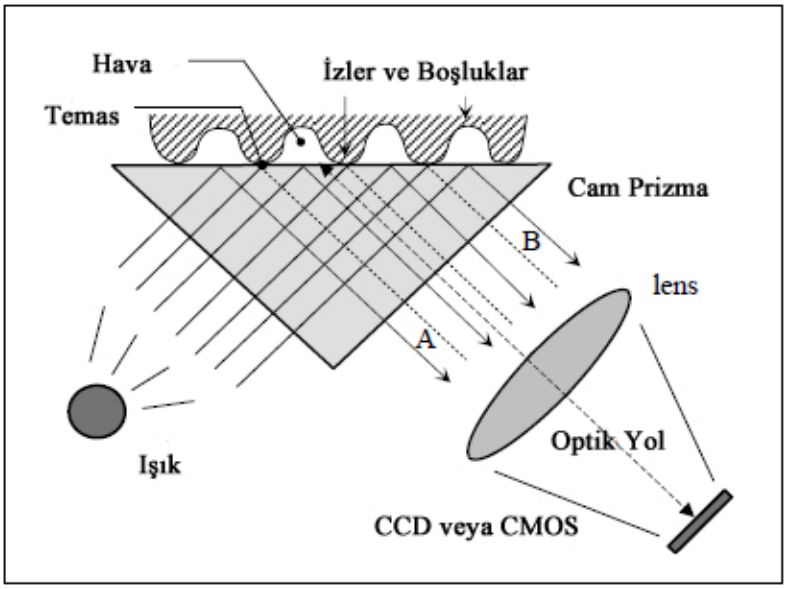
Geliştirilen parmak izi tanıma sistemi için tarayıcı alet (yakalama ve kayıt için), özellik çıkartım ve doğrulama için karşılaştırma ünitelerini içermektedir.

Görüntü yakalama için birçok teknik uygulanmakta olup bu tez çalışmasında ise düşük fiyatlı ve optik sensörler grubuna giren Secugen parmak izi okuyucu cihazı kullanılmıştır (Secugen, 2013). Secugen parmak izi okuyucu cihazı, Android işletim sistemini desteklemektedir. Bu biyometrik tarayıcı, yüksek çözünürlüklü, sorunsuz bir güvenlik çözümü sağlamak için optik algılayıcı kullanmaktadır. Şekil 4.16'da Secugen parmak izi okuyucu cihazı gösterilmiştir (Secugen, 2013).

Şekil 4.16. Secugen parmak izi okuyucu cihazı

4.10. Optik Sensörler

Parmak izinden gerekli özellikleri çıkarmak için optik sensörler kullanılmaktadır. Bu sensörlerin, sıcaklıktan etkilenmemeleri, maliyetlerinin az oluşları, 500 nokta çözünürlüğüyle daha kaliteli görüntü alınması gibi özellikleri bulunmaktadır. Parmaklarda oluşacak kesikler, nasırlar ve diğer nedenler optik tarayıcılar için sorun olmaktadır. Şekil 4.17'de bir toplam iç yansıma tabanlı optik sensör gösterilmiştir (Maltoni ve diğ., 2009). Parmak cihazın üzerine koyulduğunda parmak izini oluşturan çizgiler prizma yüzeyi ile temas etmekte ve boşluklar belli bir uzaklıkta kalmaktadır. Prizmanın bir kısmı ışık kaynağı LED ile aydınlatılmaktadır. Işık boşluk alanlarda yansıyarak parmak izlerini oluşmaktadır. Parmak izinin elde edilmesi toplam iç yansımalı sensörlerde, CCD kameralar yerine düşük maliyetli olan CMOS teknolojisi kullanılarak yapılmaktadır (Maltoni ve diğ., 2009). Toplam iç yansıma tabanlı cihazların üç boyutlu yüzey hassasiyeti bulunduğundan, bir resim veya parmak izi görüntüsü ile kandırılmamaktadır (Hıdımoğlu, 2010).



Şekil 4.17. Toplam iç yansımalı optik parmak izi sensörü

4.11. Java Nesne Serileştirme (Java Object Serialization)

Parmak izi tanıma sisteminde, veritabanı içerisinde herhangi bir kişiden alınan biyometrik parmak izi özelliklerine göre arama yapan kimlik tespiti sorgulama algoritması geliştirilmiştir.

Geliştirilen bu algoritma için, çıkarılan biyometrik parmak izi özelliklerinin Java nesne yönelimli programlama tekniğinde nesnelerin saklanması için kullanılan seri etme (serialization) yöntemi ile dosyaya saklanması ve arama yapılırken dosyadan seri etme işlemi ile geri getirilmesi sağlanmıştır. Böylelikle veritabanı üzerinde oluşan yük azaltılarak kimlik belirleme işleminin daha hızlı olması, bunun sonucunda ise geliştirilen sistemin daha performanslı çalışması sağlanmıştır.

Java nesne yönelimli bir programlama dili olduğu için, java ortamında uygulama geliştirilirken nesneler kullanılmaktadır. Java platformunda int, doublu, byte gibi primitive (basit) tipler dışında kalan her şey nesnelerden oluşmaktadır. Bununla birlikte Java'da kullanılan nesneler, sadece java sanal makinesi (JVM- Java Virtual Machine) ortamında çalışmaktadırlar. Platform dışında kalan nesnelerin bir anlamı bulunmamaktadır. Nesne yönelimli programlamayı destekleyen Java'da, tasarlanan nesnelerin tekrar kullanılabilmesi ve bu nesnelerin Java platformu dışında da hayata geçirilmesi, Java nesne serileştirme (Java object serialization) işlemi ile çok kolay bir şekilde yapılabilmektedir (javablog, 2014).

Bir nesne içinde bulunan alanlar (field) bir dosyaya yazdırıldığında, bu verilerin sadece değerleri (values) dosya içerisinde depolanmaktadır. Bu doğrultuda herhangi bir nesnenin alanlarındaki değerin string (karakter dizisi) veya int (tamsayı) tipinde olduğunun dosya üzerinde bir anlamı bulunmamaktadır. Java nesne serileştirme yöntemi ile bir nesnenin birebir kopyası, Java platformu dışında da depolanabilmektedir. Bu teknik ile daha sonra, serileştirilen nesne depolanan yerden çekilip, aynı durum ve özellikleri ile kullanılabilmektedir. Yapılan tüm bu işlemler, Java nesne serileştirme yöntemini oluşturmaktadır (javablog, 2014).

4.11.1. Nesne Serileştirmenin Temelleri

Java ortamında nesneleri serileştirmek için, serileştirilecek nesnelerin serileştirilebilir (serializable) olduğu sınıf bildirisinin başında belirtilmelidir. Nesne serileştirme işleminde, serileştirilecek olan nesnenin sınıf bildirisinde veya kalıtım ile türediği sınıfın bu yöntemi uygulaması (implement edilmesi) gerekmektedir. Nesneleri serileştirmek için Java platformu iki temel sınıf sunmaktadır. ObjectInputStream ve ObjectOutputStream sınıfları ile serileştirme işlemi yapılmaktadır. ObjectInputStream sınıfı, ObjectInput serileştirme işlemini uygular ve serileştirilen nesneyi tekrar akıştan okumak için kullanılır. ObjectOutputStream sınıfı ise ObjectOutput serileştirme işlemini uygular ve herhangi bir nesneyi akışa yazdırmak için kullanılmaktadır. ObjectInput serileştirme işlemi, readObject adında bir metot sunar ve serileştirilen nesneyi akıştan okumak için kullanılmaktadır. ObjectOutput serileştirme işlemi ise writeObject adında bir metot sunarak serileştirilen herhangi bir nesneyi bir akışa yazdırmak için kullanılmaktadır. Bu iki serileştirme işlemini sırasıyla uygulayan ObjectInputStream ve ObjectOutputStream sınıfları bu metotları kendi içinde tanımlayarak kullanıma hazırlamaktadırlar. Java ortamında nesneler bu iki sınıf kullanılarak serileştirilip dosyaya yazılırken ve dosyadan geri getirilirken, java.io paketinde sunulan FileInputStream veya FileOutputStream nesnelerinin eklenmesi gerekmektedir (javablog, 2014).

5. MOBİL ANDROID ORTAMINDA GELİŞTİRİLEN PARMAK İZİ TANIMA ve KİMLİK DOĞRULMA SİSTEMİ

Bu tez çalışması kapsamında biyometri ve mobil Android tabanlı çözümleri birleştiren kullanıcı arayüzü sistemi oluşturulmuş ve biyometri ID'lerini analiz etmek için güvenli bir parmak izi tanıma ve kimlik doğrulama sistemi geliştirilmiştir. Kullanıcı arayüzünün tasarım ve uygulanması Android platform temel alınarak yapılmıştır. Mobil ortamlar incelendiğinde Android işletim sisteminin en yaygın kullanılan bir işletim sistemi olması ve yazılım geliştiricileri için çözümler sunması gibi nedenlerden dolayı tercih edilmiştir. Parmak izi bilgileri için gerekli olan veriler biyometrik arayüz den alınmıştır. Bu çalışmada farklı biyometrik tanıma teknikleri incelenmiş, bu tekniklerden parmak izi tanıma yönteminin diğer yöntemlere göre başarı ve performansının daha yüksek ve çok daha güvenilir olması gibi nedenlerden dolayı tercih edilmesine karar verilmiştir. Parmak izi tanıma işleminde, Minutiae tabanlı parmak izi tanıma yöntemi kullanılmış ve parmak izi görüntüsü üzerinden noktasal özellikler elde edilerek eşleştirildiğinde, diğer yöntemlere göre daha başarılı ve performanslı olduğu için tercih edilerek kullanılmıştır. Bu nedenlerle, biyometrik sistemler incelenmiş ve kullanım alanları göz önüne alındığında düşük maliyetli ve taşınabilir bir biyometrik sistem tasarlanarak geliştirilmiştir.

5.1. Geliştirilen Kullanıcı Arayüzü Uygulaması

Çalışmanın bu kısmında önceki bölümlerde yapılan araştırmaların sonucu elde edilen bilgiler doğrultusunda geliştirilen mobil Android tabanlı parmak izi tanıma sisteminin tasarım ve geliştirme aşamaları anlatılmıştır. Geliştirilen sistemde kişilerin biyometrik parmak izi görüntülerinin elde edilmesi sağlanarak biyometrik özellikli bir veritabanı oluşturulmuştur. Alınan parmak izi görüntüsünden biyometrik parmak izi özelliklerinin elde edilebilmesi için noktasal ayrıntı tabanlı özellik çıkarma (Minutiae Extraction) yöntemi kullanılmıştır (Secugen, 2013). Daha sonra

bu veritabanı içerisinde kayıtlı bulunan herhangi bir kişiden alınan biyometrik parmak izi özelliklerine göre arama yapan kimlik tespiti sorgulama algoritması geliştirilmiştir.

Geliştirilen bu algoritma ile veritabanında arama yapılırken noktasal ayrıntı tabanlı eşleştirme (Minutiae Matching) yöntemi kullanılmıştır (Secugen, 2013). Geliştirilen bu algoritma için, çıkarılan biyometrik parmak izi özelliklerinin nesne yönelimli programlama tekniğinde nesnelerin saklanması için kullanılan seri etme (serialization) yöntemi ile dosyaya saklanması ve arama yapılırken dosyadan seri etme işlemi ile geri getirilmesi sağlanmıştır. Böylelikle veritabanı üzerinde oluşan yük azaltılarak kimlik belirleme işleminin daha hızlı olması, bunun sonucunda ise geliştirilen sistemin daha performanslı çalışması sağlanmıştır. Geliştirilen kimlik sorgulama algoritması ile sistemde kayıtlı olan kişiler arasında biyometrik özellikli bir arama yapılmış ve bu işlemin sonucuna göre kimlik belirleme işlemi başarıyla gerçekleştirilmiştir.

Geliştirilen Android tabanlı sistemin genel bileşenleri (USB parmak izi okuyucu, Android tabanlı cep telefonu veya tablet bilgisayar) Şekil 5.1'de gösterilmiştir.

Şekil 5.1. Sitemin bileşenlerinin genel bir görüntüsü

Çalışma kapsamında kişinin biyometrik parmak izi özelliklerine göre veritabanında arama yapılıp bulunabilmesi için iki farklı modül tasarlanmıştır. Bu modüllerden ilkinde kişinin biyometrik parmak izi özellikleri ve diğer kişisel bilgileri Android ortamdan alınmış ve daha sonra sistemin veritabanında saklanmıştır. Diğer modülde ise kimliği tespit edilecek kişinin Android ortamda biyometrik parmak izi özellikleri alındıktan sonra sistemin veritabanında sorgulama yapılarak aranması sağlanmıştır. Arama yapılırken, kimliği sorgulanan kişi ile veritabanında saklı olan kişiler arasında biyometrik parmak izi özellikleri ile sırasıyla karşılaştırmalar yapılmıştır.

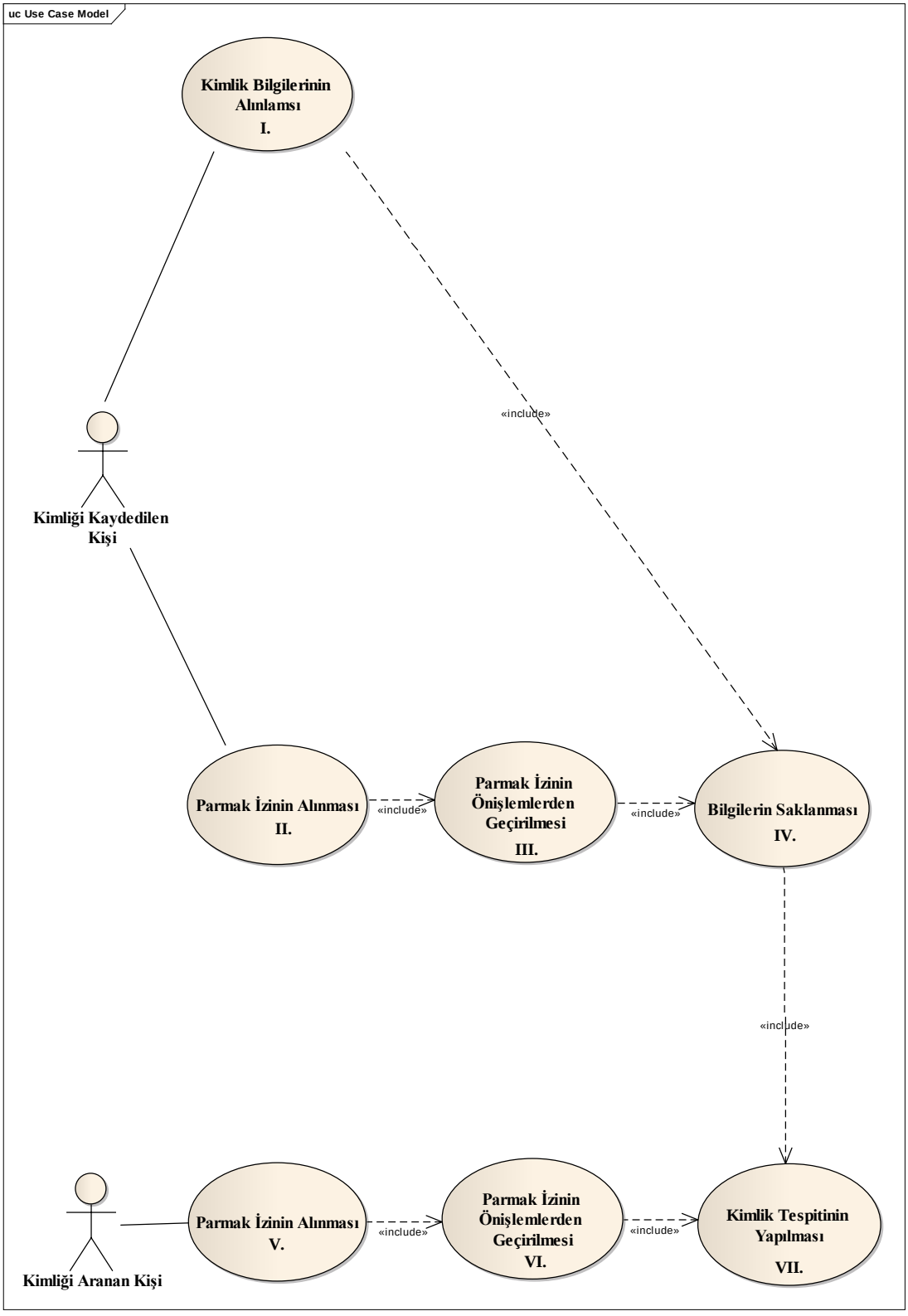
Karşılaştırma sırasında veritabanında saklı olan parmak izi özellikleriyle eşleşen kişilerin kimliğine ait bilgiler getirilerek kimlik tespiti gerçekleştirilmiştir. Parmak izi görüntüleri, Secugen parmak izi okuyucu cihazı kullanılarak, Android ortamdan alınmıştır (Secugen, 2013). Alınan parmak izi görüntüleri üzerinde önişlemlerin yapılarak görüntünün iyileştirilip temizlenmesi ve iyileştirilen bu görüntülerin üzerinden özellik noktalarını elde etmek için (Minutiae Extraction) algoritması ve Secugen Android SDK'sı (yazılım geliştirme kiti) kullanılmıştır (Secugen, 2013).

Kullanılan bütün modüller nesne yönelimli bir programlama dili olan java dili ile gerçekleştirilmiştir. Android sistemler üzerinde çalışacak uygulamalar Google tarafından sağlanan Android yazılım geliştirme araçları ve açık kaynaklı olarak oluşturulmuş olan kütüphaneleri kullanılarak geliştirilmiştir.

5.2. Sistemin Kullanım Durumu

Geliştirilen biyometrik kullanıcı arayüzü uygulamasında farklı durumlar düşünülerek genel amaçlı bir tasarım oluşturmak için diyagramlar hazırlanmıştır. Kullanım senaryosu diyagramı (use-case diagram) yazılım ve diğer sistemlerin modellenmesinde kullanılan Birleşik Modelleme Dili'nin (UML-Unified Modeling Language), on üç çeşit diyagramından biridir. Kullanım senaryosu diyagramı bir davranış diyagramıdır. Kullanım senaryosu, bir sistemden beklenen belirli bir davranışı gösterir ve sistemden beklenen gereksinimlerin tespit edilmesi amacıyla kullanılır. Kullanım senaryosu diyagramında genel olarak kullanım senaryoları ve bağımlılıkları ve ilişkileri ile aktörleri gösterilmektedir. Oluşturulan projeye ait

kullanım durum diyagramı Şekil 5.2'de gösterilmiştir. Tasarlanan sistemin kullanım durumları aşağıda detaylı olarak anlatılmıştır.



Şekil 5.2. Sistemin kullanım durumu diyagramı

I. *Kullanım Durumu:* Kimlik Bilgilerinin Alınması

Kısa Tanım: Kişinin kimlik bilgilerinin oluşturulduğu modüldür.

Ön Koşul: Kişi, parmak izi okuyucu ile parmağını düzgün bir şekilde koyup taratması gerekmektedir.

Son Durum: Kişinin kimlik bilgileri alınmıştır.

Ana Akış: T.C. Kimlik No, Adı, Soyadı, Telefonu ve E-mailinden oluşan kimlik bilgileri sisteme girilmektedir.

Hata Durumu: Hatalı kimlik bilgilerinin girişi sonucu, istenen kimlik tespit işlemleri gerçekleşememektedir.

II. *Kullanım Durumu:* Parmak İzinin Alınması.

Kısa Tanım: Sisteme kaydedilecek kişinin biyometrik parmak izi görüntüsünün oluşturulduğu modüldür.

Önkoşul: Parmak izi görüntüsünün hatasız bir biçimde alınması ve kimlik bilgilerinin eksiksiz olarak girilmesi gerekmektedir.

Son Durum: Biyometrik parmak izi görüntüsü Android ortamdan taranarak elde edilmiştir.

Ana Akış: Secugen parmak izi okuyucu cihazı kullanılarak, Android ortamda parmak izi görüntüsü hatasız bir şekilde taranarak oluşturulmuştur. Parmak izi görüntüsü üzerinde özelliklerin belirlenebilmesi için önişlemlerin yapılmasına olanak sağlanmıştır.

Hata Durumu: Parmak izi okuyucu cihazın sistem ile olan bağlantısının kopması veya parmaklar üzerinde bulunan çeşitli nedenlerden dolayı parmak izi görüntüsü düzgün bir biçimde alınamamaktadır.

III. *Kullanım Durumu:* Parmak İzinin Önişlemlerden Geçirilmesi

Kısa Tanım: Parmak izi görüntüsü üzerinde temizleme ve iz inceltme gibi önişlemlerin yapıldığı modüldür.

Ön Koşul: Parmak izi görüntüsünün başarılı bir biçimde alınmış olması gerekmektedir.

Son Durum: Oluşturulan parmak izi görüntüsü üzerinde iyileştirmeler yapılarak temizlenmesi sağlanmaktadır.

Ana Akış: Alınan parmak izi görüntüsü üzerinde özelliklerin belirlenmesi için çeşitli önişlemlerden geçirilmesi sağlanmıştır. Önişlemlerden geçirilerek özellikleri belirlenen parmak izi görüntüsü ve kişinin kimlik bilgileri ile birlikte veritabanında saklanması için uygun hale getirilmiştir.

Hata durumu: Parmak izi okuma cihazı ile görüntünün düzgün bir biçimde alınamaması sonucu iyileştirmelerde başarılı olunamamaktadır.

IV. *Kullanım Durumu:* Bilgilerin Saklanması

Kısa Tanım: Kişinin sisteme, alınan kimlik bilgileri ve çıkartılan biyometrik parmak izi özellikleri ile birlikte kaydedildiği modüldür.

Ön koşul: Kimliği tanıtılan kişinin parmak izi görüntüsü ve kimlik bilgilerinin hatasız bir biçimde alınması gerekmektedir.

Son Durum: Kişinin sisteme biyometrik parmak izi özellikleri ve kimlik bilgileri kaydedilerek tanıtılması yapılmıştır.

Ana Akış: Kişinin kimlik bilgileri ve parmak izi özellikleri, serialization işlemi ile nesne haline dönüştürülerek, Android SD karta dosya biçiminde kaydedilmektedir.

Hata Durumu: Parmak izi görüntüsünün ve kimlik bilgilerinin yanlış oluşturulmaları sonucu, hatalı kimlik kayıt işlemleri gerçekleşmektedir.

V. *Kullanım Durumu:* Parmak İzinin Alınması

Kısa Tanım: Sistemde aranacak kişinin, biyometrik parmak izi görüntüsünün oluşturulduğu modüldür.

Önkoşul: Parmak izi görüntüsünün hatasız bir biçimde alınması gerekmektedir.

Son Durum: Biyometrik parmak izi görüntüsü Android ortamdan taranarak elde edilmiştir.

Ana Akış: Secugen parmak izi okuyucu cihazı kullanılarak, Android ortamda parmak izi görüntüsü hatasız bir biçimde taranarak oluşturulmuştur. Parmak izi görüntüsü üzerinde özelliklerin belirlenebilmesi için önişlemlerin yapılmasına olanak sağlanmıştır.

Hata Durumu: Parmak izi okuyucu cihazın sistem ile olan bağlantısının kopması veya parmaklar üzerinde bulunan çeşitli nedenlerden dolayı parmak izi görüntüsü düzgün bir biçimde alınamamaktadır.

VI. *Kullanım Durumu:* Parmak İzinin Önişlemlerden Geçirilmesi

Kısa Tanım: Parmak izi görüntüsü üzerinde temizleme ve iz inceltme gibi önişlemlerin yapıldığı modüldür.

Ön Koşul: Parmak izi görüntüsünün başarılı bir biçimde alınmış olması gerekmektedir.

Son Durum: Oluşturulan parmak izi görüntüsü üzerinde iyileştirmeler yapılarak temizlenmesi sağlanmaktadır.

Ana Akış: Alınan parmak izi görüntüsü üzerinde özelliklerin çıkarılıp belirlenmesi için çeşitli önişlemlerden geçirilmesi sağlanmıştır. Önişlemlerden geçirilerek özellikleri çıkarılan parmak izi görüntüsü, veritabanında arama yapılabilmesi için uygun hale getirilmiştir.

Hata durumu: Parmak izi okuyucu cihazıyla görüntünün düzgün bir biçimde alınamaması sonucu iyileştirmelerde başarılı olunamamaktadır.

VII. *Kullanım Durumu:* Kimlik Tespitinin Yapılması

Kısa Tanım: Kimliği bulunacak kişinin parmak izinin sistemde aranarak kimlik tespitinin yapıldığı modüldür.

Ön Koşul: Kimliği bulunacak kişinin ve kimliği veritabanına tanıtılan kişilerin hatasız bir biçimde parmak izi görüntüleri alınmalı ve Minutiae noktasal özellikleri çıkartılmalıdır.

Son Durum: Kimliği bulunacak kişinin parmak izi görüntüsünün Minutiae özellikleri ile veritabanında bulunan parmak izlerinin Minutiae özellikleri, birebir karşılaştırılarak kimlik sorgulaması gerçekleştirilmiştir.

Ana Akış: Kişinin alınan parmak izi görüntüsünün Minutiae özellikleri ile veritabanında kayıtlı olan parmak izi görüntülerinin Minutiae özellikleri arasında sırasıyla karşılaştırmalar yapılmıştır. Bununla birlikte parmak izleri üzerinde noktasal özellik tabanlı eşleştirme işlemleri gerçekleştirilmektedir. Eşleştirme işlemleri sonucunda kişinin sisteme kayıtlı olup olmadığı hakkında karar verilmiştir. Eğer parmak izi görüntüleri bire bir eşleşiyorsa kişinin sisteme kayıtlı olduğu sonucuna varılmaktadır. Daha sonra eşleşen kişiye ait kimlik bilgileri veritabanından getirilerek kimlik tespiti gerçekleştirilmektedir. Fakat parmak izi görüntüsünün Minutiae özelikleri, veritabanında bulunan parmak izi görüntülerinin Minutiae özellikleri ile eşleşmiyorsa kişinin sisteme kayıtlı olmadığı sonucuna varılmaktadır.

Hata durumu: Siteme kayıtlı olan ve kimliği tespit edilecek kişilerin parmak izleri eksiksiz ve düzgün bir biçimde alınması gerekmektedir.

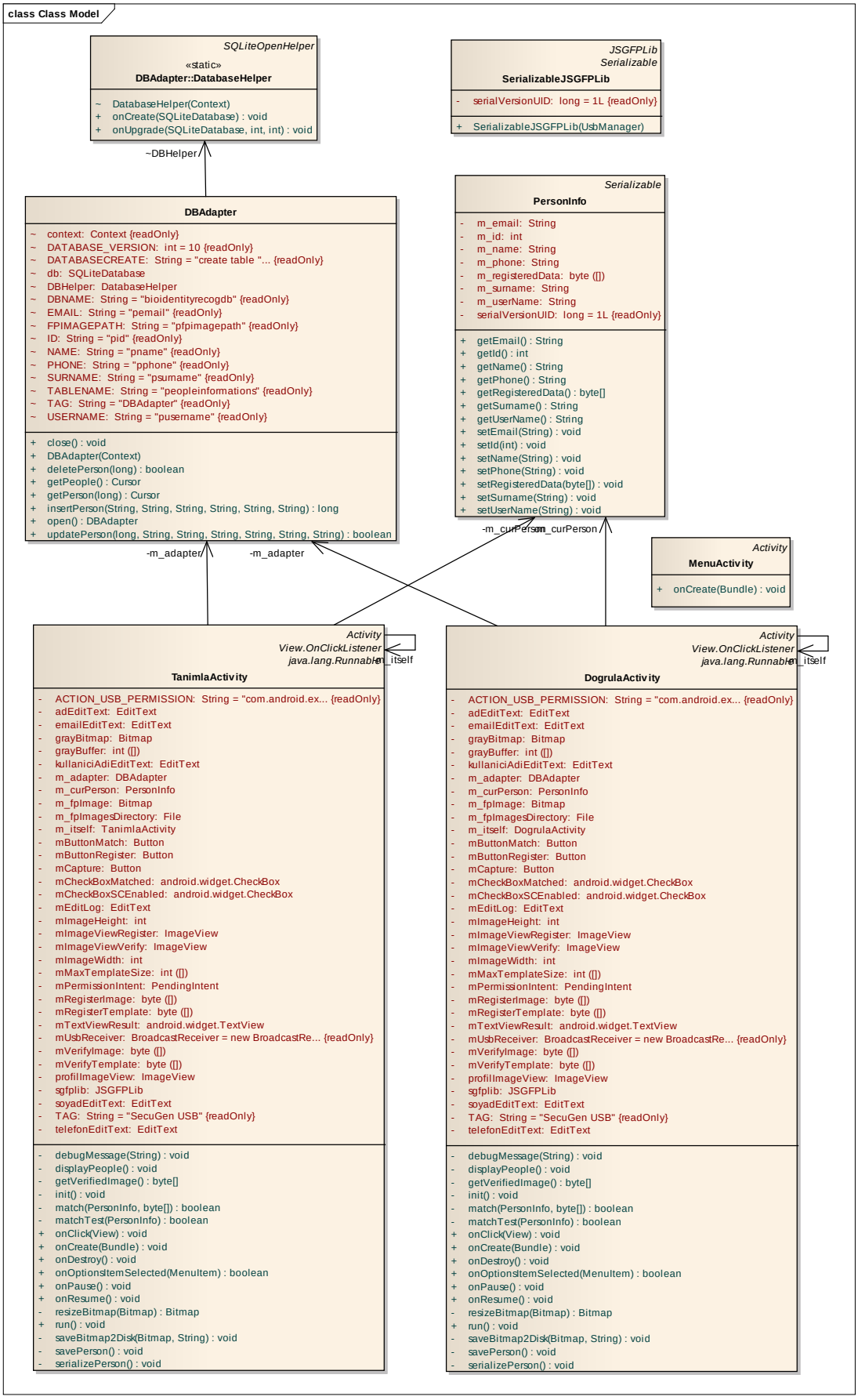
5.3. Sistemin Sınıf Yapısı

Gerçekleştirilen kullanıcı arayüzü uygulamasının sınıf diyagramı Şekil 5.3'de gösterilmiştir. Uygulamada kişilerin kimlik bilgilerinin saklanması için bir veritabanı yapısı oluşturulmuştur. Uygulamada DBAdapter sınıfı DatasaseHelper sınıfından türetilmiştir. DatabaseHelper sınıfında veritabanının oluşturulması sağlanmıştır. Kişilerin kimlik bilgilerinin tutulması için DBAdapter sınıfında bir veritabanı tablosu oluşturulmuştur. Bu tabloda kişilerin kimlik no, adı, soyadı, telefonu, e-mail bilgileri ve parmak izi görüntülerinin dosya adresleri saklanmıştır.

Parmak izi görüntüsünün Minutiae özellikleri çıkarıldıktan sonra elde edilen byte verileri Java Object Serialization (java nesne serileştirme) işlemi ile SRL uzantılı dosyaya kaydedilmiştir. Bu işlem için SerilizableJSGFPLib sınıfı oluşturulmuştur. Daha sonra kişilerin kimlik ve parmak izi özellik bilgileri için PersonInfo sınıfı oluşturularak nesnelerin serileştirilmesi sağlanmıştır.

Gerçekleştirilen Android parmak izi tanıma sistemine bir ana menüden giriş yapılması sağlanmıştır. Ana menüye giriş için MenuActivity sınıfı yazılmıştır. Ana menüden kimlik tanımlamak için TanımlaActivity sınıfı oluşturulmuştur. TanımlaActivity sınıfında kişinin parmak izi görüntüsü okunmuş ve Minutiae özellik noktaları çıkarılmıştır (Secugen, 2013). Kişinin kimliğini oluşturan diğer bilgilerin alınması sağlanmıştır. Alınan kimlik bilgileri ve parmak izi görüntüleri, serialization işlemi yapıldıktan sonra SD kartta oluşturulan FingerPrintImages klasörüne SRL uzantılı dosya biçiminde saklanmıştır. Ayrıca okunan parmak izi görüntüleri JPG uzantılı dosya biçiminde kaydedilerek diğer uygulamalar için parmak izi görüntüsünün alınabilmesi sağlanmıştır.

Ana menüden kimlik doğrulama işlemi için DogrulaActivity sınıfı oluşturulmuştur. DogrulaActivity sınıfında kimliği aranan kişinin parmak izi görüntüsü okunmuş ve Minutiae özellik noktaları çıkarılmıştır (Secugen, 2013). Daha sonra FingerPrintImages klasöründe saklanan parmak izi görüntülerinin Minutiae özellikleriyle sırasıyla karşılaştırılmıştır. Karşılaştırma sonucunda yeni okunarak elde edilen parmak izinin Minutiae özellik noktaları ile serileştirilen dosyalarda saklı olan parmak izinin Minutiae özellik noktaları eşleştiğinde kimlik tespiti gerçekleştirilmiştir. Eğer karşılaştırma işlemi sırasında eşleşme gerçekleşmemişse bir uyarı mesajı verilerek eşleşmediği bildirilmiştir.



Şekil 5.3. Sitemin sınıf diyagramı

5.4. Mobil Android Parmak İzi Tanıma Sisteminin Arayüz Mimarisi

Çalışma kapsamında planlanan mobil parmak izi tanıma ve kimlik doğrulama sisteminde, Android işletim sistemi kullanılarak USB parmak izi okuyucusu ile sisteme kimlik tanıtılmasının yapılması ve daha sonra güvenlik amacıyla yapılan kimlik sorgulamaları ile kişinin kimliğinin doğrulanarak tespit edilmesi için bir arayüz programı geliştirilmiştir.

Secugen parmak izi okuyucu cihazı, Android mobil platformlarda çalışabilmek üzere tasarlanmıştır (Secugen, 2013). Parmak izi tanımlama ve kimlik doğrulama işlemlerinde parmak izi görüntüsüne ait özellik noktalarının çıkarılması (Minutiae Extraction) ve özellik noktalarının eşleştirilmesi (Minutiae Matching) kısımlarında Secugen Android SDK'sı (yazılım geliştirme kiti) kullanılmıştır (Secugen, 2013). Elde edilen bu özellik noktaları kullanılarak Android ortamda parmak izi tanıma ve kimlik doğrulama uygulaması gerçekleştirilmiştir.

5.4.1. Sistemde Kullanılan Secugen Fonksiyonları

Kullanılan Secugen SDK fonksiyonları JSGFPLib sınıfına entegre edilmiştir. JSGFPLib sınıfı aygıt başlatma, parmak izi görüntüsünü alma, Minutiae özelliklerini çıkarma ve eşleştirme fonksiyonlarını içermektedir. JSGFPLib sınıfını kullanmak için, JSGFPLib sınıfını kullanan bir SerializableJSGFPLib sınıfı türetilmiştir. Oluşturulan SerializableJSGFPLib sınıfının yapısı aşağıdaki gibidir.

```
public class SerializableJSGFPLib extends JSGFPLib implements Serializable
{
	public SerializableJSGFPLib(UsbManager manager)
	{
		super(manager);
	}

	private static final long serialVersionUID = 1L;
}
```

Daha sonra SerializableJSGFPLib() fonksiyonu çağrılarak bir sgfplib nesnesi oluşturulmuştur. Bu nesne bir parametre olarak kurucusuna android.hardware.usb.UsbManager nesnesini geçirmektedir. Böylelikle sgfplib

nesnesinin USB parmak izi okuyucu cihazını kullanabilmesi sağlanmıştır. Oluşturulan sgfplib nesnesi aşağıdaki gibidir.

```
sgfplib = new
SerializableJSGFPLib((UsbManager)getSystemService(Context.USB_SERVICE));
```

SerializableJSGFPLib sınıfı ile USB cihazdan alınan parmak izi özellik bilgileri, nesne serileştirme işlemleri yapılarak dosya biçiminde SD karta kaydedilmiştir.

Parmak izi okuyucusu JSGFPLib nesnesi ile oluşturulduktan sonra, JSGFPLib nesnesi Init() fonksiyonu kullanılarak başlatılmıştır. JSGFPLib.Init() fonksiyonu kullanılarak aygıt adı alınmış ve aygıt adına karşılık gelen cihaz bilgilerine dayalı parmak izi algoritması modülü başlatılarak cihaz sürücüsü yüklenmiştir. Bu fonksiyonun kullanımı aşağıdaki gibidir (Secugen, 2013).

```
long error = sgfplib.Init(SGFDxDeviceName.SG_DEV_AUTO);
```

Uygulama sonlandırılmadan önce JSGFPLib.Close() fonksiyonu çağrılmıştır. Bu fonksiyon JSGFPLib nesnesi tarafından kullanılan belleği boşaltmaktadır. Kullanımı aşağıdaki gibidir (Secugen, 2013).

```
long error = JGSFPLib.Close();
```

Kullanılacak USB okuyucu sayısını belirlemek için OpenDevice() fonksiyonu kullanılmıştır (Secugen, 2013). Sadece bir adet USB okuyucu cihazının Android aygıt ile bağlantısını kurmak için, OpenDevice() fonksiyonunda sıfır değeri kullanılmıştır.

```
long error = sgfplib.OpenDevice(0);
```

Cihaz bilgileri, resmin yükseklik ve genişliği gibi gerekli bilgiler JSGFPLib.GetDeviceInfo() fonksiyonu çağırılarak alınmıştır. Cihaz, bilgileri SGDeviceInfoParam() fonksiyonunu kullanılarak elde edilmiştir. Okuyucu cihaz başlatıldıktan sonra, bir parmak izi görüntüsü alınmıştır. JSGFPLib nesnesinin parmak izi görüntüsünü yakalama fonksiyonları üç gruptan oluşmaktadır. Yakalanan

parmak izinin 256 gri-seviye görüntüleri ve görüntü genişliği ile görüntü yüksekliği JSGFPLib.GetDeviceInfo() fonksiyonu çağırılarak alınmıştır. Görüntü için tampon alan, çağıran uygulama tarafından sağlanmaktadır. JSGFPLib.GetImage() fonksiyonu ise bir parmağın var olup olmadığını kontrol etmeden bir görüntü yakalamak veya görüntü kalitesini kontrol etmek için kullanılmıştır (Secugen, 2013).

Parmak izi okuyucusu kullanılırken, çevresel faktörler ve ana sistemin özelliklerine bağlı olarak, bir parmak izi görüntüsünün parlaklığı değişebilmektedir. Secugen USB parmak izi okuyucu cihazı iyi bir görüntü kalitesi sağlamak ve parlaklığını ayarlamak için Akıllı Çekim (Smart Capture) olarak adlandırılan bir teknoloji kullanmaktadır (Secugen, 2013). Akıllı çekim varsayılan olarak etkin ayarlanmıştır. Akıllı çekimin JSGFPLib.WriteData() fonksiyonu kullanılarak aşağıdaki gibi devre dışı bırakılabilmesi sağlanmıştır. Burada, bir checkBox nesnesi seçili ise akıllı çekim aktif, aksi durumda devre dışı bırakılmıştır.

```
boolean smartCaptureEnabled = this.mCheckBoxSCEnabled.isChecked();
if (smartCaptureEnabled)
    sgfplib.writeData((byte)5, (byte)1);
else
    sgfplib.writeData((byte)5, (byte)0);
```

Parmak izini kaydetmek ve doğrulamak için, ilk önce parmak izi görüntüsü yakalanmıştır. Daha sonra görüntünün noktasal özellikleri çıkarılarak (Minutiae Extraction) bir taslak içinde veriler elde edilmiştir. Minutiae özelliklerinde izler, iz uçları, çatallanmalar, vadiler ve helezonlar gibi her bir parmak izine özgü temel noktasal özellikler bulunmaktadır.

Bir taslak oluşturmak ve bir parmak izi görüntüsünden Minutiae özelliklerini çıkarmak için JSGFPLib.CreateTemplate() fonksiyonu kullanılmıştır. Tampon alan uygulama programı tarafından sağlanmıştır. Minutiae özelliklerinin tampon boyutunu almak için JSGFPLib.GetMaxTemplateSize() fonksiyonu kullanılmıştır. Bu fonksiyon, bir taslağa veri için en fazla tampon boyutunu döndürmektedir. Gerçek taslak boyutu taslak oluşturulduktan sonra JSGFPLib.GetTemplateSize() fonksiyonu çağrılarak elde edilmiştir. JSGFPLib.CreateTemplate() fonksiyonu görüntünün sadece bir veri taslağını oluşturmaktadır (Secugen, 2013).

Oluşturulan taslaklar doğrulama işlemleri sırasında karşılaştırılmıştır. Her bir görüntü örneğinin Minutiae verileri çıkarılmakta ve daha sonra kayıtlı olan parmak izi görüntülerinin verileriyle eşleşmesi için birbirleriyle karşılaştırılmaktadır. Bu

karşılaştırma genellikle, yeni bir parola girmek için gerekli olan parola onay işlemine benzemektedir. Onaylama sırasında, yeni girilen Minutiae verisi sistemde kayıtlı olan Minutiae verileriyle karşılaştırılmıştır. Çalışma kapsamında geliştirilen sistemde taslak ve giriş Minutiae verilerini eşleştirmek için JSGFPLib.MatchTemplate() fonksiyonu kullanılmıştır (Secugen, 2013). Bu fonksiyon varsayılan biçim olarak aynı formata sahip iki taslak veri kümesini eşleştirmektedir.

Parmak izini kaydetmek için, ilk önce bir parmak izi görüntüsü yakalanmıştır. Daha sonra görüntünün Minutiae özellikleri çıkarılarak bir taslak içinde veriler elde edilmiştir. Her görüntüden Minutiae özellikleri çıkarılarak elde edilen veriler daha sonra kayıtlı parmak izi görüntüsünü eşleştirmek için birbirleriyle karşılaştırılmıştır.

Kayıt sürecinin genel akışı şu şekildedir:

- Yakalanan parmak izi görüntüleri JSGFPLib.GetImage() fonksiyonunda tutulması sağlanmıştır.
- Her yakalanan parmak izi görüntüsünün Minutiae özelliklerinin elde edilmesi, JSGFPLib.CreateTemplate() fonksiyonu kullanılarak yapılmıştır.
- Kayıt işlemini tamamlamak için bilgiler dosyalara ve veritabanına kaydedilmiştir.

Doğrulama işlemleri ise, sistemde kayıtlı bulunan Minutiae verilerine karşı yeni alınan Minutiae verilerinin eşleştirilmesiyle oluşturulmuştur. Yeni bir parmak izi görüntüsü taranarak alınmış ve bir taslak içerisinde görüntünün Minutiae verileri çıkarılmıştır.

Doğrulama sürecinin genel akışı şu şekildedir:

- Yakalanan parmak izi görüntüleri JSGFPLib.GetImage() fonksiyonunda tutulması sağlanmıştır.
- Her yakalanan parmak izi görüntüsünün Minutiae özelliklerinin çıkarılması JSGFPLib.CreateTemplate() fonksiyonu kullanılarak yapılmıştır.
- Yeni alınan taslakların, sistemde kayıtlı taslaklar ile karşılaştırılıp eşleştirilmesi JSGFPLib.MatchTemplate() fonksiyonu ile yapılmıştır.

5.4.1.1. Sistemin Eşleşme Başarı Puanı

Sistemde, kayıt veya doğrulama işlemleri sırasında gelişmiş bir kalite kontrolü için, eşleşme başarı puanının belirlenmesi gerekmektedir. Geliştirilen

sistemde eşleşme başarı puanının elde edilmesi güvenlik seviyesi ayarı kullanılarak yapılmıştır (Secugen, 2013). Bu puanı aşan Minutiae veri setleri kabul edilmekte, böylelikle eşleşme başarı puanı bulunmaktadır. Eşleşme başarı puanının altında kalan veriler ise reddedilmektedir. Eşleşme puanı 0 ile 199 arasında bir değere sahip olabilmektedir. Farklı güvenlik seviyeleri ayarlanması ile eşleşme başarı puanlarının değiştirilmesi sağlanmış ve böylelikle farklı sınır (threshold) değerleri oluşturulmuştur. Eşleşme başarı puanlarının oluşturduğu sınır değerleri ve güvenlik seviyeleri arasındaki ilişki Tablo 5.1'de gösterilmektedir (Secugen, 2013).

Tablo 5.1. Eşleşme puanı ile ilgili güvenlik seviyeleri

Güvenlik Seviye Sabiti	**Seviye Değeri**	**Karşılık gelen eşleşme puanı**	**Sınır Değeri (%)**
SL_NONE	0	0	% 0
SL_LOWEST	1	30	% 21
SL_LOWER	2	50	% 36
SL_LOW	3	60	% 43
SL_BELOW_NORMAL	4	70	% 50
SL_NORMAL	5	80	% 57
SL_ABOVE_NORMAL	6	90	% 64
SL_HIGH	7	100	% 70
SL_HIGHER	8	120	% 85
SL_HIGHEST	9	140	% 100

Geliştirilen sistemde uygulama türüne göre güvenlik seviyesinin ayarlanması SGFDxSecurityLevel metodunun kullanılmasıyla yapılmıştır (Secugen, 2013). Geliştirilen sistemde, parmak izi sadece kimlik doğrulama amacıyla kullanılmıştır. Bu nedenle, yanlış kabulleri (FAR) azaltmak için SL_NORMAL değerinden daha yüksek bir güvenlik seviyesine ayarlanmıştır. Eşleşme başarı puanı, güvenlik seviyesinin SL_HIGH değerine ayarlanarak % 70 sınır (threshold) değerinde olması şu şekilde sağlanmıştır.

```
sgfplib.MatchTemplate(pi.getRegisteredData(), verifiedTemplate,
   SGFDxSecurityLevel.SL_HIGH, matched);
```

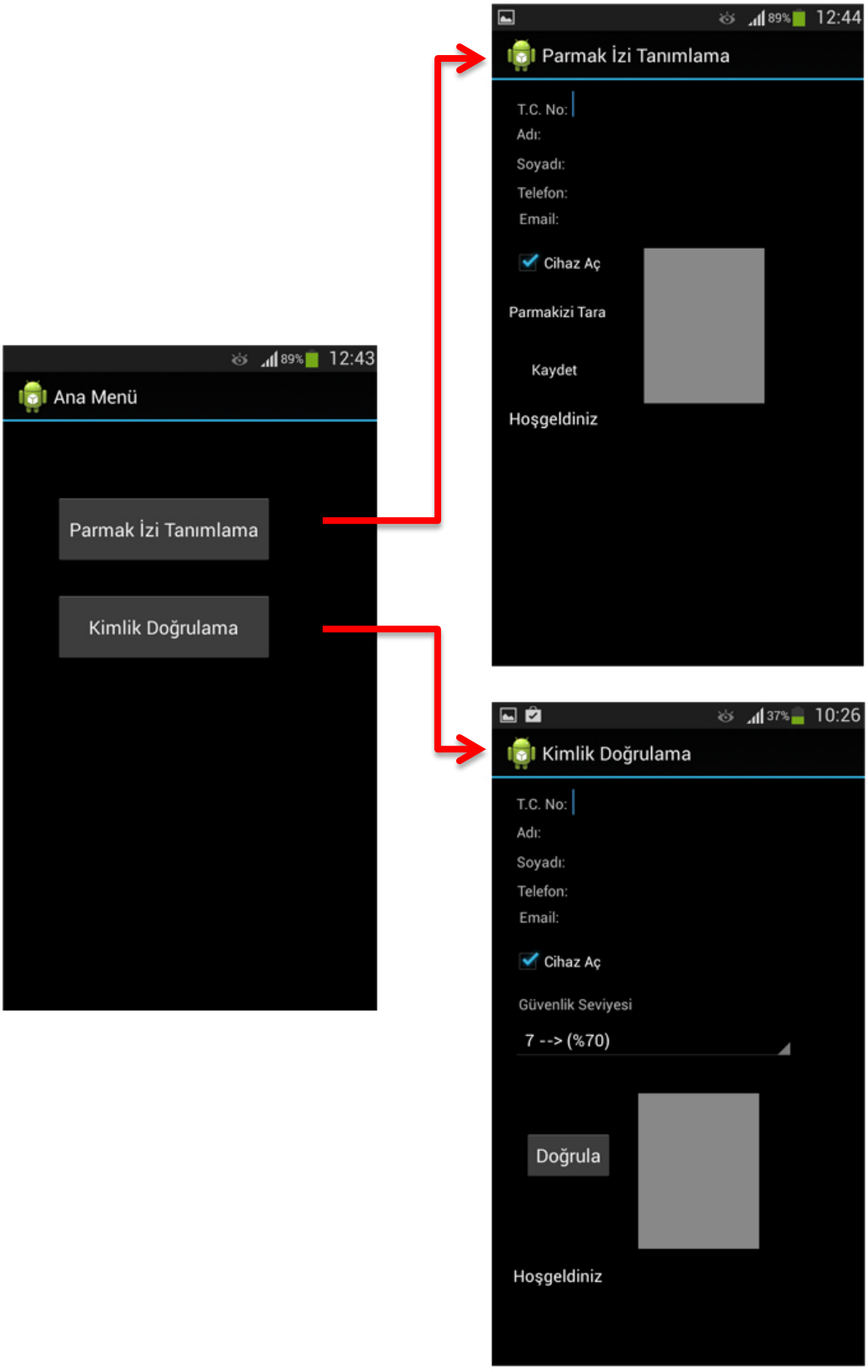
Burada MatchTemplate() fonksiyonu kullanılarak giriş parmak izi görüntüsünün Minutiae özellikleri ve veritabanında bulunan taslak parmak izi görüntüsünün Minutiae özellikleri % 70 sınır değerine göre karşılaştırılmıştır. Bu sınır değeri değiştirilerek sistemin hassasiyetinin ayarlanabilmesi sağlanmıştır.

5.5. Mobil Android Parmak İzi Tanıma Sisteminin Kullanıcı Arayüzü

Günümüzde tasarlanan programları kullanıcılar tercih ederlerken programın kullanılabilirliği ve görselliğine önem vermektedirler. Bir uygulama programının performanslı olmasının yanında kullanım kolaylığı ve tasarımının dikkat çekmesi gerekmektedir. Yazılım firmaları hazırlamış oldukları uygulama programlarını kullanıcılara hitap eden biçimde GUI (Graphical User Interfaces) tasarlamaktadırlar.

Biyometrik doğrulama günümüzde geleneksel şifreli doğrulama yöntemlerinin yerini almıştır. Doğrulama güvenlik seviyesini artırmış ve kullanımını basitleştirmiştir. Biyometrik örnek yakalandığında parmak izi doğrulaması yapılmak üzere parmak izinin özellikleri çıkarılarak kaydedilmektedir. Fakat bu kaydı biyometrik algoritmalar için en ideal şekilde hazırlamak üzere mobil cihazda bazı önlemlerin alınmış olması gerekmektedir. Örneğin, kullanıcılar gerçekten de mobil cihaza düzgün bir şekilde parmak izini okutuyor mu? Şeklinde kontrol edilmesi zorunludur. Biyometriler için doğrulama ve kimliklendirme olarak bilinen iki temel kullanım senaryosu vardır. Doğrulama yeni oluşturulan örneğin veritabanında bulunan sadece başka bir örnekle karşılaştırıldığı senaryodur ki bu birebir karşılaştırma olarak tanımlanır. Diğer biyometrik kullanım senaryosu ise kimliklendirme işlemidir ki bu da yeni oluşturulan bir örneğin veritabanında bulunan birçok kayıtlı örneklerle karşılaştırılması gerektiğinde kullanılmaktadır. Bu karşılaştırmaya da birin birçok ile karşılaştırılması denir. Her iki senaryonun da böyle bir kaydetme arayüzü ile desteklenmesi gerekmektedir.

Tasarlanan Android tabanlı sistemin iki önemli aşaması bulunmaktadır. Bunlardan ilki parmak izi tanımlama işlemlerinin yapılabilmesi için kişinin biyometrik parmak izi özelliklerinin ve kimlik bilgilerinin alınıp sisteme kaydedildiği kısımdır. Diğer önemli bölüm ise, kimlik tespitinin yapılabilmesi için tasarlanan kimlik doğrulama arayüz kısmından oluşmaktadır. Geliştirilen sitemde bulunan bu iki önemli kısım Parmak İzi Tanımlama ve Kimlik Doğrulama şeklinde tasarlanmıştır. Bu modüllere bir ana menü ile ulaşılması sağlanmıştır. Böylelikle istenildiği zaman sisteme parmak izi tanımlama işlemleri ya da sistemde bir arama yapılarak kimlik tespit işlemleri gerçekleştirilebilmektedir. Bu kullanıcı arayüz programı bir prototip olarak düşünülmüş ve Android ortamda uygulanarak gerçekleştirilmiştir. Şekil 5.4'de Programın ana menüsü, parmak izi tanımlama ve kimlik doğrulama arayüz tasarımları gösterilmiştir.



Şekil 5.4. Geliştirilen sistemin arayüz görüntüleri

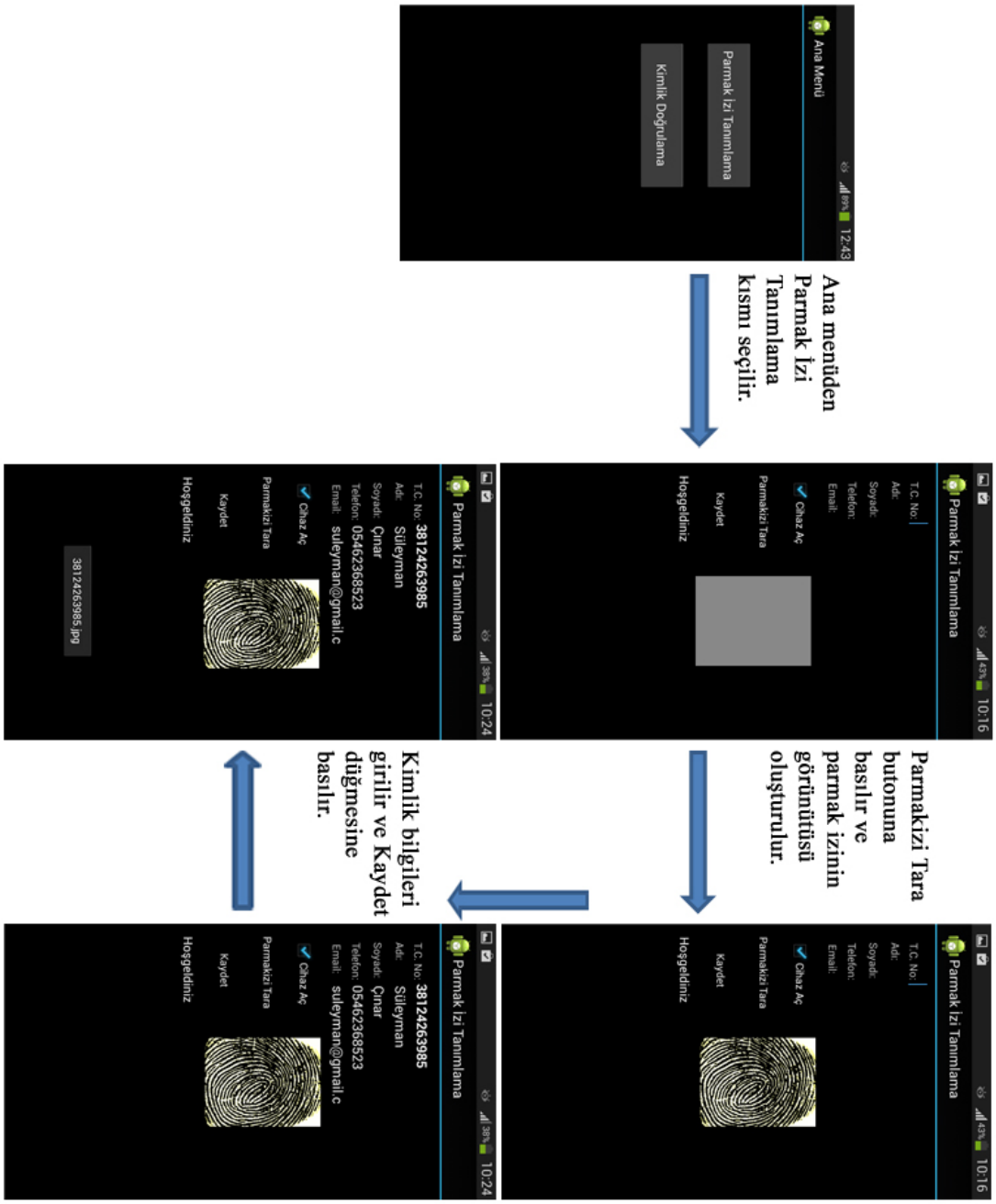
Kullanıcıya yukarıdaki şekilde gösterilen ana menü ekranından, sisteme kişinin parmak izinin tanımlanması veya sistemde kayıtlı olan kişiler arasında bir kimlik doğrulaması gerçekleştirebilmek için seçenekler sunulmuştur. Parmak izi tanımlama veya kimlik doğrulama kısımlarında Secugen parmak izi okuyucu cihazının mobil aygıt ile düzgün bir biçimde bağlantısı kurulmalıdır. Secugen parmak izi okuyucu cihazının USB arabirimi ile Android aygıta kolay bir şekilde bağlantısı yapılabilmektedir (Secugen, 2013). İşlemlerin hatasız ve düzgün bir biçimde gerçekleştirilmesi için parmak izi tanımlama ve kimlik doğrulama aşamalarında biyometrik parmak izi görüntülerinin parmak izi okuyucu cihazı ile hatasız bir şekilde alınması gerekmektedir.

5.5.1. Sisteme Parmak İzinin Tanımlanması

Sisteme kişinin biyometrik özelliklerinin kaydedildiği Activity aşamasıdır. Bu bölüm çalıştırılmadan önce mobil aygıta parmak izi okuyucu cihazının USB arabiriminden bağlantısı kurulmalıdır. Burada sisteme, kişinin parmak izinin tanıtılabilmesi için kişinin kimliğine ait bilgileri ve biyometrik parmak izi görüntüsünün noktasal özellikleri elde edilmiştir. Kimlik bilgileri arayüz programından girilmektedir. Bunun için T.C. No, Adı, Soyadı, Telefonu ve E-mail bilgileri doğru bir şekilde Android arayüzünün ilgili EditText alalarından girilmesi sağlanmıştır.

Başarılı bir parmak izi tanımlama işleminin gerçekleşebilmesi için parmak izi görüntüsünün düzgün bir biçimde alınması gerekmektedir. Bunun için Parmakizi Tara isminde bir düğme oluşturulmuştur. Parmak izi görüntüsü alındıktan sonra görüntü üzerinde önişlemler gerçekleştirilmiş ve daha sonra nesne serileştirme (object serialization) işlemi yapılarak parmak izi görüntüsü diğer alınan kimlik bilgileri ile birlikte SD karta dosya biçiminde kaydedilerek saklanmıştır. Burada nesne serileştirme işlemi yapılarak kimlik tespiti sorgulama algoritması geliştirilmiştir. Geliştirilen bu algoritma ile arama aşamasında parmak izi özelliklerine çok hızlı bir şekilde ulaşılması sağlanmıştır. Kimlik doğrulama işlemi yapılırken parmak izi verilerinin serileştirildiği dosyalardan sırasıyla alınarak karşılaştırma işleminin yapılması sağlanmıştır. Bununla birlikte SQLite veritabanı dosyasında kişinin kimliğine ait bilgilerin ve parmak izi görüntüsünün dosya adres bilgilerinin saklanması gerçekleştirilmiştir. Alınan bu bilgilerin, serileştirilerek SD

karta SRL dosyası biçiminde kaydedilmesi ayrıca SQLite veritabanı dosyasında bilgilerin saklanması, Kaydet düğmesi çalıştırıldığında yapılmıştır. Geliştirilen sistemin parmak izi tanımlama Activity kısmının aşamaları Şekil 5.5'de gösterilmiştir.



Şekil 5.5. Kimlik bilgilerinin tanımlanması

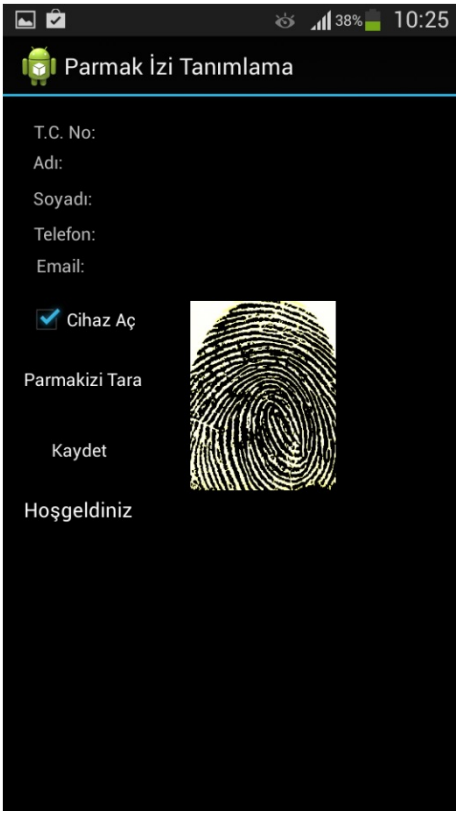
5.5.1.1. Parmak İzi Tanımlama Aşamasında Parmak İzinin Elde Edilmesi

Parmak izi görüntüsünün alınması ve önişlemlerden geçirilerek iyileştirilmesi işlemleri için Android arayüzünde Parmakizi Tara düğmesi oluşturulmuştur. Görüntünün alımı için GetImage() fonksiyonu kullanılmıştır (Secugen, 2013). Parmak izi görüntüsü alınmadan önce cihazın açık olup olmadığının kontrolleri yapılmıştır. Bu işlem, Cihaz Aç adında bir checkBox kontrolü ile gerçekleştirilmiştir. Eğer Cihaz Aç kontrolü seçili ise, parmak izi okuyucu cihazı görüntü alımı için aktif hale getirilmektedir. Parmak izi görüntüsü düzgün bir biçimde taranarak elde edildikten sonra gerekli önişlemlerden geçirilmiş ve parmak izinin bir görüntüsünün imageView nesnesinde gösterilmesi sağlanmıştır. Yapılan bu işlemler için geliştirilen yazılımın genel bir kısmı şu biçimdedir.

```
public void onClick(View v) {
	if (v== mCheckBoxSCEnabled)
	{
		boolean smartCaptureEnabled = this.mCheckBoxSCEnabled.isChecked();
		if (smartCaptureEnabled)
			sgfplib.writeData((byte)5, (byte)1);
		else
			sgfplib.writeData((byte)5, (byte)0);
	}
	if (v == mCapture) {
		//. . .
		long result = sgfplib.GetImage(buffer);

		profilImageView.setImageBitmap(b);
		//. . .

	}
}
```

Parmak izi okuyucusu kullanılırken, çevresel faktörler ve ana sistemin özelliklerine bağlı olarak, bir parmak izi görüntüsünün parlaklığı değişebilmektedir. Secugen USB parmak izi okuyucu cihazı iyi bir görüntü kalitesi sağlamak ve parlaklığını ayarlamak için Akıllı Çekim (Smart Capture) olarak adlandırılan bir teknoloji kullanmaktadır (Secugen, 2013). Akıllı çekim varsayılan olarak etkin ayarlanmıştır. Akıllı çekimin JSGFPLib.WriteData() fonksiyonu kullanılarak devre dışı bırakılabilmesi sağlanmıştır. Burada, bir checkBox nesnesi seçili ise akıllı çekim aktif, aksi durumda devre dışı bırakılmaktadır.

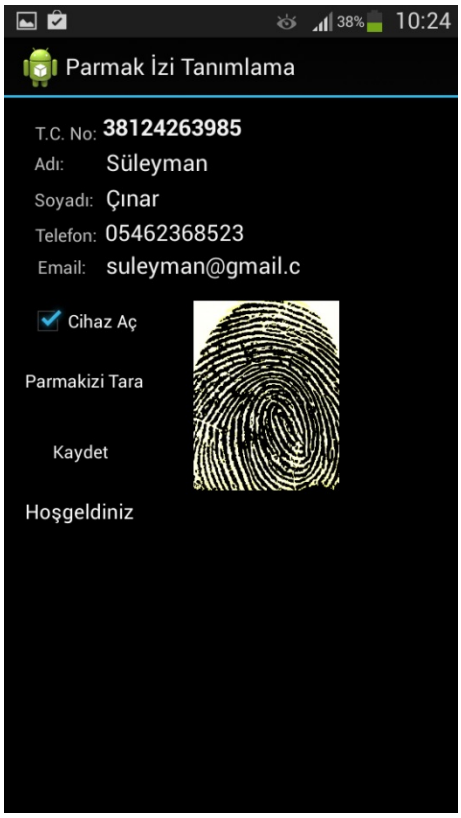
Parmak izi tanımlama kısmında parmak izi görüntüsü alındıktan sonra arayüz programının görüntüsü Şekil 5.6'da gösterilmiştir.



Şekil 5.6. Taranarak alınan parmak izi görüntüsü

5.5.1.2. Parmak İzi Tanımlama Aşamasında Kimlik Bilgilerinin Girilmesi

Parmak izi tanımlama işlemlerinin düzgün bir şekilde gerçekleşebilmesi için kişiye ait kimlik bilgilerinin eksiksiz bir biçimde girilmesi gerekmektedir. Kimlik bilgileri Android arayüzünden girilmektedir. Bu işlem için T.C. No, Adı, Soyadı, Telefonu ve E-mail alanlarının doğru bir şekilde Android arayüzünden ilgili editText alalarına girilmesi sağlanmıştır. Girilen kimlik bilgileri Android SQLite veritabanı dosyasına kaydedilerek saklanması gerçekleştirilmiştir. Bu aşamada T.C. No alanının boş geçilmemesi ve benzersiz bir numara verilmesi sağlanmıştır. T.C. No alanı veritabanı dosyasında birincil anahtar (primary key) olarak oluşturulduğundan veritabanı tablosunda bu alana ait bilgilerin tekrarlanmaması gerekmektedir. Kimlik doğrulama işlemlerinde, elde edilen parmak izi görüntüsü üzerinden kimlik tespiti yapıldığı için kişinin kimliğini oluşturan bu bilgilerin sistem kullanıcısı tarafından düzgün bir şekilde girilmesi gerekmektedir. Kimlik bilgileri ve parmak izi görüntüsünün eksiksiz bir şekilde alınması sonucu oluşan ekran görüntüsü Şekil 5.7'de gösterilmiştir.



Şekil 5.7. Alınan kimlik bilgileri ve parmak izi görüntüsü

5.5.1.3. Kimlik Bilgilerinin ve Parmak İzi Özelliklerinin Kaydedilmesi

Bu aşamada kişinin kimliğine ait girilen bilgilerin ve elde edilen parmak izi görüntüsüne ait Minutiae özelliklerinin kaydedilmesi gerçekleştirilmiştir. Kaydetme işlemleri, parmak izi tanımlama kısmında Kaydet düğmesi çalıştırıldığında yapılmıştır. Önceki aşamalarda düzgün bir şekilde alınan kişinin kimliğine ait bilgilerin, burada nesne serileştirme işlemleriyle ayrıca SRL dosyalarına kaydedilmesi sağlanmıştır. Bu işlem ile kişiye ait bilgiler, iki farklı biçimde kaydedilmiştir. Bunlardan ilkinde çıkarılan parmak izinin Minutiae özelliklerine nesne serileştirme işlemleri uygulanmış ve kişinin kimlik bilgilerini oluşturan editText alanlarıyla birlikte SRL uzantılı dosyalara kaydedilmesi sağlanmıştır. Oluşturulan bu SRL dosyaları, SD kartın içerisinde bir FingerPrintImages klasörüne kaydedilmiştir. Diğer kaydetme işleminde ise kişinin kimlik bilgileri ve parmak izi görüntüsünün adresi veritabanında saklanmıştır. Parmak izi görüntüleri ayrıca JPG formatına dönüştürülerek FingerPrintImages klasörüne kaydedilmiştir. Parmak izi görüntüleri JPG biçiminde saklanarak farklı uygulamalar tarafından bu görüntülere erişilebilmesine olanak sağlanmıştır. Kaydedilen parmak izi görüntüleri, JPG ve SRL

dosya biçimlerinde tutulmuştur. Nesnelerin serileştirilen verileri SRL dosya biçimlerinde saklanmıştır. Burada nesnelerin serileştirilerek kaydedilmesinin avantajı, parmak izi arama işlemleri gerçekleştirilirken veritabanı üzerinde oluşan yükün azaltılması ve sistemin çok hızlı bir şekilde verilere ulaşması ve böylece performansının arttırılması sağlanmıştır.

Oluşturulan SRL ve JPG dosyalarının, Android işletim sisteminde SD kartın içerisine kaydedilmesi için FingerPrintImages klasörünün oluşturulması şu biçimdedir.

```
m_fpImagesDirectory = new
File(Environment.getExternalStorageDirectory().getAbsolutePath(),
"FingerPrintImages");
m_fpImagesDirectory.mkdirs();
```

Kaydet düğmesinin çalıştırılması ile kişi bilgilerinin veritabanına kaydedilmesi savePerson() ve serializePerson() fonksiyonlarında gerçekleştirilmiştir.

```
kaydetButton.setOnClickListener(new View.OnClickListener() {
 public void onClick(View v) {
    try {
      //. .
      savePerson();

      serializePerson();
      //. . .
    }
    catch (Exception ex) {
       //. .  .
    }
  }
});
```

Elde edilen kimlik bilgileri savePerson() fonksiyonu içinde, PersonInfo sınıfı ile SQLite veritabanı dosyasına kaydedilmiştir. Arayüz kısmından alınan kimlik no, adı, soyadı, telefonu ve e-mail alanlarından oluşan kimlik bilgilerinin veritabanına kaydedilmesi sağlanmıştır. Parmak izi görüntülerinin Minutiae özellikleri CreateTemplate() fonksiyonu kullanılarak elde edilmiştir. Parmak izi görüntülerini diske kaydetme işlemleri saveBitmap2Disk() fonksiyonunda yapılmıştır. Elde edilen parmak izi görüntülerinin, dosya adlarını kimlik numarası ile dosya uzantılarını ise JPG olacak biçimde SD kartın içine kaydedilmiştir. Ayrıca veritabanı dosyasında, SD kartın içerisine kaydedilen JPG biçimindeki parmak izi resimlerinin dosya yolu adres bilgileri saklanmıştır. Böylelikle veritabanı dosyasından parmak izi

görüntülerine erişilebilme olanağı sağlanmıştır. Kimlik bilgilerinin kaydedildiği savePerson() fonksiyonu şu şekildedir.

```
private void savePerson() throws IOException, FileNotFoundException
{
  String userName = "", name = "", surname = "", phone = "", email = "",
  fpImagePath = "";

  fpImagePath = userName + ".jpg";
  //. .
  result = sgfplib.CreateTemplate(fpInfo, mRegisterImage,
  mRegisterTemplate);

  m_curPerson.setRegisteredData(mRegisterTemplate);

  m_adapter.open();

  m_adapter.insertPerson(userName, name, surname, phone, email,
  fpImagePath);

  m_adapter.close();

  saveBitmap2Disk(m_fpImage, fpImagePath);
}
```

Elde edilen parmak izi görüntüleri, saveBitmap2Disk() fonksiyonu kullanılarak FingerPrintImages klasörünün içerisine JPG dosya biçiminde kaydedilmiştir. Yapılan bu işlem ile farklı uygulamalar tarafından parmak izi görüntülerine kolay bir şekilde ulaşılmasına olanak sağlanmıştır.

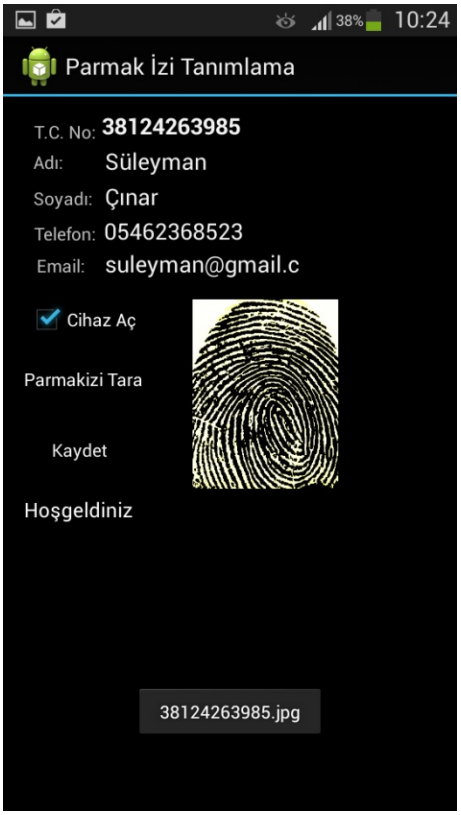
```
private void saveBitmap2Disk(Bitmap bitmap, String path) throws
IOException, FileNotFoundException
{
  FileOutputStream fos = new FileOutputStream(new
  File(m_fpImagesDirectory, path));

  bitmap.compress(CompressFormat.JPEG, 100, fos);
  fos.close();
}
```

Burada farklı bir yöntem daha geliştirilerek elde edilen parmak izi görüntülerinin Minutiae özellikleri ve diğer alınan kişisel kimlik bilgileri ile birlikte java nesne serileştirme işlemi uygulanmış ve SRL uzantılı dosyalara kaydedilmesi sağlanmıştır. Serileştirme işleminin gerçekleştirilebilmesi için, serializePerson() fonksiyonu içinde SerializableJSGFPLib sınıfı kullanılmıştır. Oluşturulan SRL dosyalarının isimleri kimlik numaraları, uzantıları SRL biçiminde ayarlanarak SD kartın içine kaydedilmesi işlemleri gerçekleştirilmiştir.

Burada nesne serileştirme için, ObjectOutputStream sınıfının writeObject metodu kullanılarak serleştirilen nesnelerin SRL dosyalarına kaydedilmesi aşağıdaki biçimde sağlanmıştır.

```
private void serializePerson() throws IOException
{
  ObjectOutputStream oos = new ObjectOutputStream(new
  FileOutputStream(new File(m_fpImagesDirectory,
  m_curPerson.getUserName() + ".srl")));

  oos.writeObject(m_curPerson);

  oos.close();
}
```

Android arayüzünde, parmak izi tanımlama ekranında kaydetme işleminden sonra oluşan ekran görüntüsü Şekil 5.8'de gösterilmiştir. Kaydetme işleminden sonra, kişinin sisteme başarıyla kaydedildiğini göstermek amacıyla dosya adı ve uzantısını gösteren bir uyarı mesajı verilmiştir.



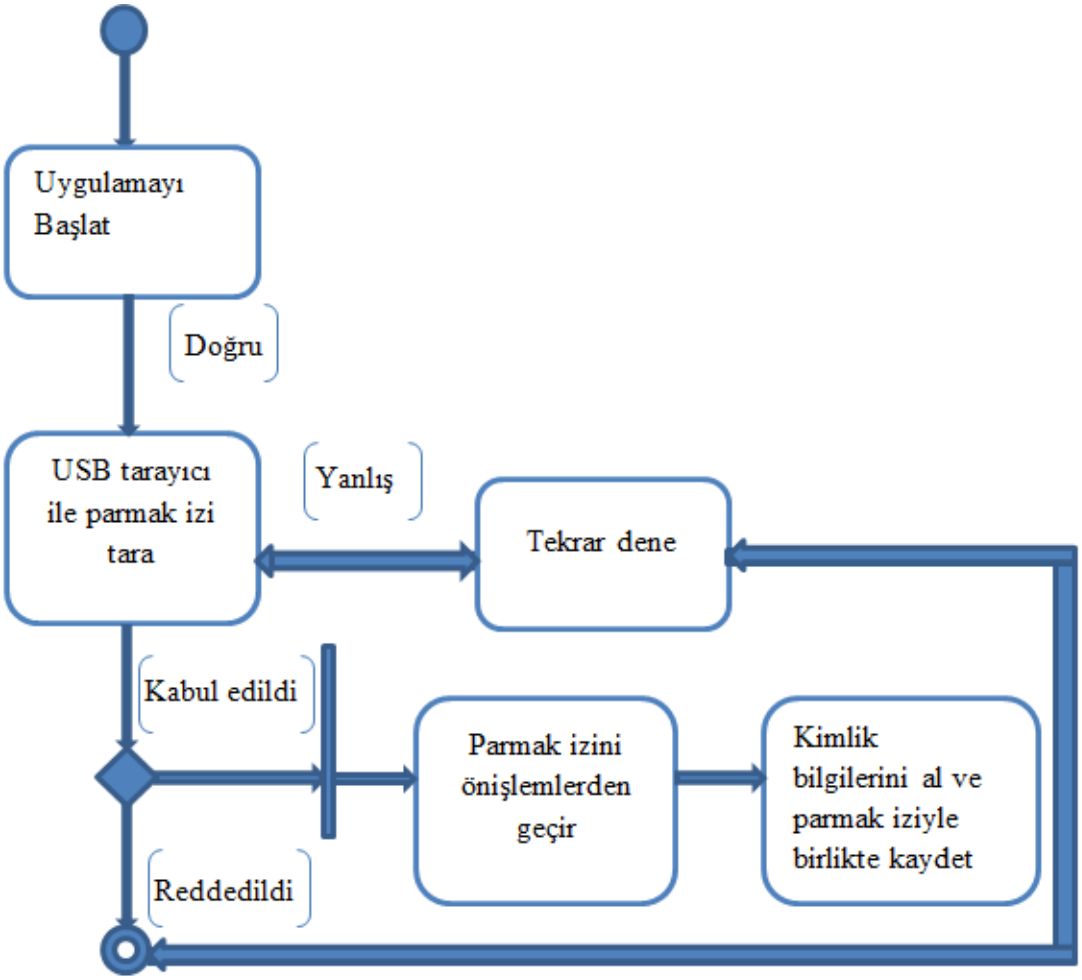
Şekil 5.8. Kimlik bilgilerinin kaydedilmesi

SD kartın içerisine kaydedilen JPG ve SRL dosyalarının bulunduğu FingerPrintImages klasörünün içeriği Şekil 5.9'da gösterilmiştir.

Şekil 5.9. SD karta kaydedilen JPG ve SRL dosyaları

5.5.1.4. Parmak İzi Tanımlama Aşamasının UML Aktivite Şeması

Parmak izi tanımlama kısmının dört fraklı yaşam döngüsü durumu vardır. Uygulama kullanıcı tarafından başlatılmaktadır. Bir sonraki aşamada kişinin USB parmak izi tarayıcısı ile parmak izi görüntüsü elde edilmektedir. Deneme başarılı olmazsa, kişinin yeni bir anlık tarama yapabilmesi sağlanmıştır. Düzgün bir şekilde taranarak alınan parmak izi görüntüsü üzerinde önişlemler yapılarak görüntü iyileştirilmesi yapılmıştır. Bu şekilde Minutiae özellik noktalarının tespit edilmesi sağlanmıştır. Daha sonra kişinin kimlik bilgileri de girildikten sonra bilgilerin kaydedilme işlemi gerçekleştirilmiştir. Şekil 5.10'da uygulamanın kullanıcı tarafından kaydedilmesi için UML aktivite şeması gösterilmiştir.



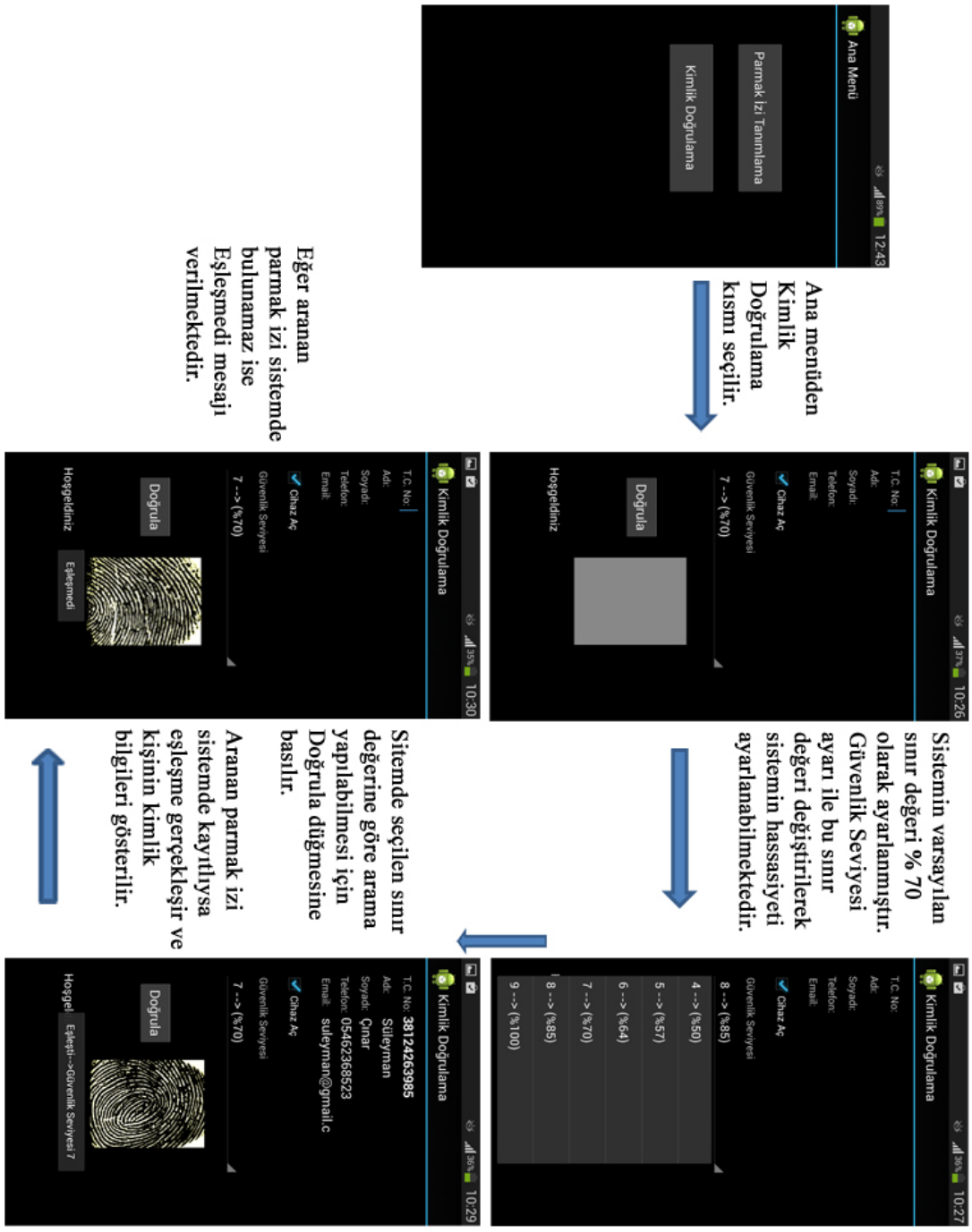
Şekil 5.10. Parmak izi tanımlama aşamasının UML aktivite şeması

5.5.2. Sistemde Kimlik Doğrulamanın Yapılması

Sitemde kişinin biyometrik parmak izi özelliklerine göre bir aramanın yapıldığı Activity aşamasıdır. Bu bölüm çalıştırılmadan önce mobil aygıta parmak izi okuyucu cihazının USB arabiriminden bağlantısı kurulmalıdır. Burada sistemde kimlik araması yapılabilmesi için kimliği aranan kişinin biyometrik parmak izi görüntüsünün elde edilmesi sağlanmıştır. Parmak izi tanımlama aşamasında olduğu gibi parmak izi görüntüsünün düzgün bir biçimde elde edilmesi gerekmektedir.

Sistemin veritabanında aranan kişinin kimliğini sorgulamak için, Doğrula düğmesi oluşturulmuştur. Doğrula düğmesi çalıştırıldığında, önce aranan kişinin parmak izi görüntüsü elde edilmektedir. Daha sonra önişlemlerden geçirilerek Minutiae özellik noktaları çıkarılan parmak izi görüntüsü ile sistemde saklı olan Minutiae özellik noktaları arasında sırasıyla karşılaştırılmalar yapılmıştır. Eğer giriş

Minutiae özellik noktaları ile taslak Minutiae özelik noktaları eşleşmişse kişinin kimlik tespiti gerçekleşmekte ve kişiye ait sistemde saklanan diğer kimlik bilgileri getirilmektedir. Fakat Minutiae özellik noktaları eşleşmiyorsa sistem tarafından eşleşmedi şeklinde uyarı mesajı verilmiştir. Gerçekleştirilen kimlik doğrulama Activity kısmının aşamaları Şekil 5.11'de gösterilmiştir.



Şekil 5.11. Kimlik bilgilerinin doğrulanması

5.5.2.1. Kimlik Doğrulama Aşamasında Parmak İzinin Elde Edilmesi

Geliştirilen sistemde parmak izi özelliklerine göre arama yapılarak kimlik tespitinin yapılabilmesi için Doğrula düğmesi oluşturulmuştur. Doğrula düğmesi çalıştığında ilk önce kimliği aranan kişinin parmak izi görüntüsü elde edilmekte ve gerekli önişlemlerden geçirilen görüntünün iyileştirilip Minutiae özellikleri çıkarılmaktadır. Daha sonra elde edilen Minutiae verisi ile serileştirilerek saklanan parmak izi görüntülerinin Minutiae verileri arasında sırayla karşılaştırmalar yapılır. Minutiae verileri serileştirilerek saklanan SRL dosyalarından alınıp karşılaştırıldığından arama çık hızlı gerçekleştirilmektedir. Bu şekilde veritabanı üzerinde oluşan yük azaltılmış ve sistemin performansı da artırılmış olmaktadır. Sistemde arama yapılarak karşılaştırılan Minutiae özellik noktaları eşleştiğinde, kişi sistemde bulunarak kimlik doğrulaması gerçekleştirilir. Eşleşen kişinin parmak izi görüntüsü ve diğer kimlik bilgileri, serileştirilerek saklandığı dosyadan getirilerek arayüz programında gösterilmesi sağlanır.

Aşağıdaki algoritma ile serileştirilip SD kartın içinde saklanan SRL uzantılı dosyalara sırasıyla erişilir. SD kartın içerindeki FingerPrintImages klasöründe seri edilerek saklanan dosyaların adedi öğrenilir. Daha sonra getVerifiedImage() fonksiyonu içerisinde giriş parmak izi görüntüsünün Minutiae özellik noktaları verifiedTemplate (doğrulanacak taslak) dizisi içinde elde edilir. Burada parmak izi görüntüsünü elde etmek için GetImage() fonksiyonu kullanılmış ve bu görüntü üzerinden Minutiae özellik noktalarının çıkarılması için CreateTemplate() fonksiyonu kullanılmıştır. Elde edilen giriş parmak izinin Minutiae özellikleri verifiedTemplate dizisinde tutulur. Daha sonra FingerPrintImages klasöründeki dosya adedi kadar bir döngü kurularak dosya uzantılarına sırasıyla bakılır. Burada, sadece SRL uzantılı dosyalar sıra ile okunur ve ObjectInputStream sınıfının readObject metodu ile serileştirilen nesneler dosyadan geri getirilir. Böylelikle veritabanında saklanan kişilere ait parmak izi görüntülerinin Minutiae özellikleri ve kimlik bilgileri elde edilir. Bir sonraki adımda serileştirilen dosyadan elde edilen Minutiae özelliklerinin sorgulanması için, giriş Minutiae özellikleri ile birlikte match() fonksiyonuna gönderilir. Arayüz kısmından Doğrula düğmesi çalıştırıldığında, sistemde serileştirilerek depolanan bilgileri aramak için geliştirilen algoritma şu şekildedir.

```
public void onClick(View v) {
  //. . .

  File [] files = m_fpImagesDirectory.listFiles();

  int len = files.length;

  byte [] verifiedTemplate = getVerifiedImage();

  for (int i = 0; i < len; i++) {
     String fileName = files[i].getName();
     int index = 0;

     if ((index = fileName.lastIndexOf('.')) == -1 ||
     fileName.substring(index).compareToIgnoreCase(".srl") != 0)
       continue;

     fis = new FileInputStream(files[i]);
     ois = new ObjectInputStream(fis);

     pi = (PersonInfo)ois.readObject();

     if (match(pi, verifiedTemplate)) {
       bMatchFlag = true;
       break;
     }
}
```

Giriş ve taslak Minutiae özellikleri arasında bir eşleşme oluşursa aranan kişi sistemde kayıtlı olan kişi olduğu için, bu kişiye ait diğer kimlik bilgileri de arayüz programına getirilir. Eğer sistemin veritabanında yapılan sorgulamalar sonucu kişi sistemde bulunamıyorsa "Eşleşmedi" şeklinde bir mesaj verilerek sistemde kayıtlı olmadığı hakkında bir bilgi verilmesi sağlanmıştır.

```
if (bMatchFlag) {
      String msg = "";

      kullaniciAdiEditText.setText(pi.getUserName());
      adEditText.setText(pi.getName());
      soyadEditText.setText(pi.getSurname());
      telefonEditText.setText(pi.getPhone());
      emailEditText.setText(pi.getEmail());
}
else
     Toast.makeText(m_itself, "Eşleşmedi", Toast.LENGTH_LONG).show();

}
```

Sistemde sorgulanarak arama yapılan kişinin, parmak izi görüntüsünün alınması ve bu görüntü üzerinde ön işlemlerin yapılarak Minutiae özellik

noktalarının tespit edilmesi getVerifiedImage() fonksiyonu içerinde şu şekilde gerçekleştirilmiştir.

```
private byte [] getVerifiedImage()
{
  //. . .
  long result = sgfplib.GetImage(mVerifyImage);
  //. . .
  profilImageView.setImageBitmap(b);
  this.mImageViewVerify.setImageBitmap(b);

  result = sgfplib.CreateTemplate(fpInfo, mVerifyImage, mVerifyTemplate);
  return mVerifyTemplate;
}
```

Sitemin sınır değerinin spinner (açılır liste) elemanı kullanılarak değiştirilebilmesi için global (genel) olarak securityLevel (güvenlik seviyesi) adında bir değişken tanımlanmıştır. Sistemin ilk çalıştığı anda varsayılan sınır değerinin % 70 olarak ayarlanması için spinner (açılır liste) kontrol elemanının setSelection() metoduna 7 tamsayı değeri gönderilmiştir. Böylelikle sistemin güvenlik seviyesinin, varsayılan olarak yedinci elemanı seçili hale getirilmiş ve % 70 sınır değerine göre karşılaştırma başarısının olması sağlanmıştır. Spinner elemanı her değiştiğinde onItemSelected() metodu içine girilerek parent.getItemIdAtPosition(pos) işlemi ile securityLevel değişkenine bir tamsayı değer döndürülmüştür. Böylelikle spinner kontrol elemanında seçili olan listedeki seçenek belirlenmiştir. Spinner açılır liste elemanı ile seçilen sınır değerine göre karşılaştırma işlemleri, Doğrula düğmesine basıldığında match() fonksiyonu içerisinde securityLevel değişkenin değerine göre gerçekleştirilmiştir. Spinner elemanı ile yapılan bu işlemler şu şekilde ifade edilmiştir.

```
spinner.setSelection(7);
spinner.setOnItemSelectedListener(new OnItemSelectedListener(){
      public void onItemSelected(AdapterView<?> parent, View view,
      int pos, long id) {
            if(parent.getItemIdAtPosition(pos)==0){
                  securityLevel = 0;
            }else if(parent.getItemIdAtPosition(pos)==1){
                  securityLevel = 1;
            }else if(parent.getItemIdAtPosition(pos)==2){
                  securityLevel = 2;
            }else if(parent.getItemIdAtPosition(pos)==3){
                  securityLevel = 3;
            }else if(parent.getItemIdAtPosition(pos)==4){
                  securityLevel = 4;
```

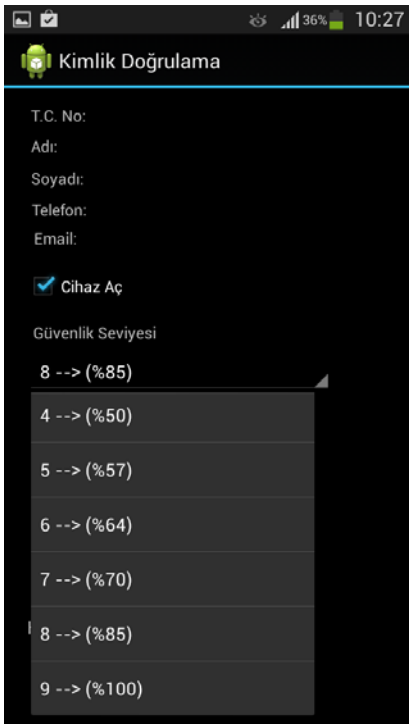
```
            }else if(parent.getItemIdAtPosition(pos)==5){
                  securityLevel = 5;
            }else if(parent.getItemIdAtPosition(pos)==6){
                  securityLevel = 6;
            }else if(parent.getItemIdAtPosition(pos)==7){
                  securityLevel = 7;
            }else if(parent.getItemIdAtPosition(pos)==8){
                  securityLevel = 8;
            }else if(parent.getItemIdAtPosition(pos)==9){
                  securityLevel = 9;
            }
      }
      //. . .
});
```

Giriş Minutiae özellikleri ile sistemde depolandığı dosyasından getirilerek oluşturulan taslak Minutiae özellikleri match() fonksiyonu içerisinde karşılaştırılmıştır. Burada giriş ve taslak Minutiae özellikleri MatchTemplate() fonksiyonu kullanılarak karşılaştırılmıştır. Karşılaştırma işlemi yapılırken SGFDxSecurityLevel güvenlik seviyesinin SL_HIGH ayarı kullanılarak eşleşme başarı puanına göre varsayılan sınır değerinin % 70 olması sağlanmıştır. Böylelikle eşleşme esnasında, sınır değerin altında kalan Minutiae noktaları kabul edilmemiş fakat sınır değerin üzerinde olan Minutiae noktaları kabul edilmiştir. Bu sınır değerinin, SGFDxSecurityLevel güvenlik ayarı değiştirilerek securityLevel değişkeninin spinner açılır listesindeki seçili olan durumuna göre değiştirilmesi gerçekleştirilmiştir. Sistem güvenliğinin artırılması gereken uygulamalarda daha yüksek bir güvenlik seviyesine ayarlanabilmesi için spinner açılır liste elemanındaki seçili olan eleman securityLevel değişkeni ile kontrol edilmiştir. Bu değişkenin durumuna göre sistemin hassasiyeti değiştirilerek daha farklı yüzdeliklerdeki sınır değerine göre karşılaştırma başarısının elde edilmesi şu ifadelerin geliştirilmesi ile yapılmıştır.

```
private boolean match(PersonInfo pi, byte [] verifiedTemplate)
{
      boolean[] matched = new boolean[1];

      if(securityLevel==0){
            sgfplib.MatchTemplate(pi.getRegisteredData(),
            verifiedTemplate, SGFDxSecurityLevel.SL_NONE, matched);
            //. . .
      }
      else if(securityLevel==1){
            sgfplib.MatchTemplate(pi.getRegisteredData(),
            verifiedTemplate, SGFDxSecurityLevel.SL_LOWEST, matched);
            //. . .
      }
```

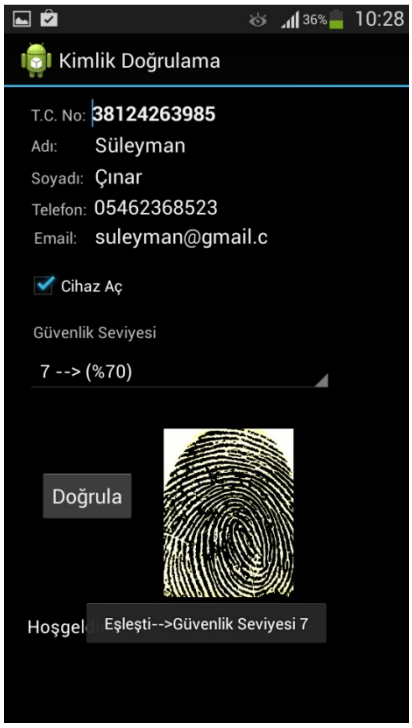
```
        else if(securityLevel==2){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_LOWER, matched);
              //. . .
        }

        else if(securityLevel==3){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_LOW, matched);
              //. . .
        }
        else if(securityLevel==4){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_BELOW_NORMAL,
              matched);
              //. . .
        }
        else if(securityLevel==5){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_NORMAL, matched);
              //. . .
        }
        else if(securityLevel==6){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_ABOVE_NORMAL,
              matched);
              //. . .
        }
        else if(securityLevel==7){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_HIGH, matched);
              //. . .
        }
        else if(securityLevel==8){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_HIGHER, matched);
              //. . .
        }
        else if(securityLevel==9){
              sgfplib.MatchTemplate(pi.getRegisteredData(),
              verifiedTemplate, SGFDxSecurityLevel.SL_HIGHEST, matched);
              //. . .
        }
       return matched[0];
}
```

Programın arayüz kısmından güvenlik seviyesi farklı değerlerde seçilerek, securityLevel değişkeninin spinner açılır listesindeki seçili olan durumuna göre SGFDxSecurityLevel güvenlik ayarının değiştirilmesi Şekil 5.12'de gösterilmiştir.



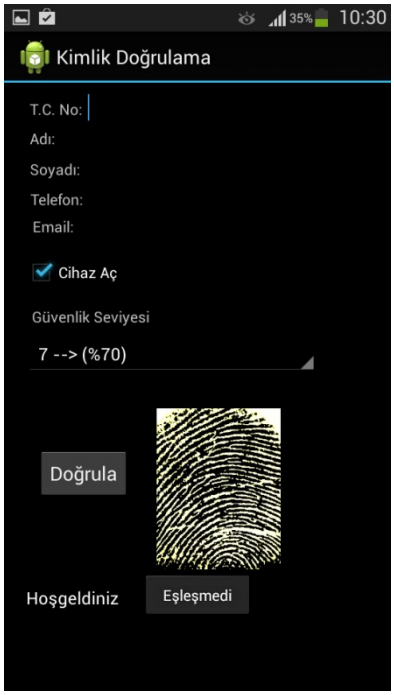
Şekil 5.12. Güvenlik seviyesi ayarı ile sınır değerinin değiştirilmesi

Şekil 5.13'de Sistemde arama yapılarak karşılaştırılan Minutiae özellik noktaları eşleştiğinde, kişi sistemde bulunarak kimlik doğrulamasının gerçekleştirildiği gösterilmiştir.



Şekil 5.13. Kişinin sistemde bulunması ve kimlik doğrulamanın gerçekleşmesi

Eğer kişinin elde edilen Minutiae verileri sistemde yapılan aramalar sonucunda bulunamazsa Şekil 5.14'de olduğu gibi bir uyarı mesajı verilerek eşleşmenin gerçekleşmediği sistem tarafından bildirilmiştir.



Şekil 5.14. Eşleşmenin gerçekleşmemesi sonucu verilen bilgi

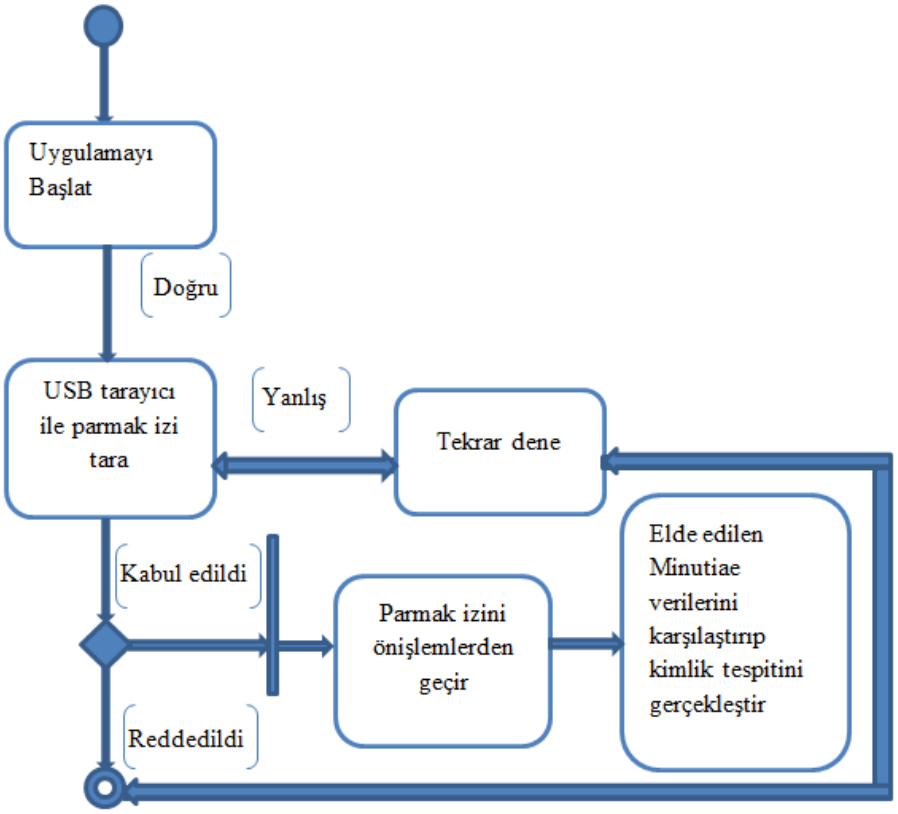
5.5.2.2. Kimlik Doğrulama Aşamasında Minutiae Algoritmasının Kullanılması

Parmak izi görüntüsünün noktasal özelliklerini çıkarma (Minutiae Extraction) ve parmak izi görüntüsünün noktasal özelliklerinin eşleştirilmesi (Minutiae Matching) işlemlerinde Minutiae algoritması kullanılmıştır (Secugen, 2013). Minutiae algoritmasının kullanılmasının nedeni ise parmak izinin farklı konumlarda taransa bile algoritmanın sağladığı üstünlükler sonucu eşleşmenin gerçekleşmesidir. Bunun sebebi, Minutiae algoritmasında çıkarılan parmak izi özellik noktalarının x değeri, y değeri, lokal izin yönü, iz noktasının özelliği (hat sonu, çatal) elde edilerek kaydedilmesidir. Giriş özellikli nokta ile taslak özellikli nokta arasındaki fark belirlenen piksel sayısından büyük değilse ve özellikleri de aynı ise iki noktanın aynı olduğuna karar verilmektedir. Eşleşen noktalar sayılır, eşleşen noktaların sayısı belirlenen sayıdan büyük ise iki parmak izi görüntüsünün aynı kişiye ait olduğu sonucuna varılmaktadır.

Gerçekleştirilen Mobil Android parmak izi tanıma ve kimlik doğrulama sisteminde, bu dört bilgi tanıma ve doğrulama işlemlerinde kullanılmıştır (Secugen,2013).

5.5.2.3. Kimlik Doğrulama Aşamasının UML Aktivite Şeması

Kimlik doğrulama sisteminin dört fraklı yaşam döngüsü durumu vardır. Uygulama kullanıcı tarafından başlatılmaktadır. Bir sonraki aşamada kişinin USB parmak izi tarayıcısı ile parmak izi görüntüsü alınmaktadır. Deneme başarılı olmazsa, kişinin yeni bir anlık tarama yapabilmesi sağlanmıştır. Düzgün bir şekilde taranarak alınan parmak izi görüntüsü üzerinde önişlemler yapılarak görüntü iyileştirilmesi yapılmış ve Minutiae verileri elde edilmiştir. Daha sonra yeni elde edilen Minutiae verisiyle sistemde kayıtlı olan parmak izi görüntülerinin Minutiae verileri arasında sırasıyla karşılaştırmalar yapılmıştır. Yapılan karşılaştırmalar sonucu kişi sisteme kayıtlı veya değil şeklinde kimlik tespitinin kararı verilmektedir. Şekil 5.15'de uygulamanın kullanıcı tarafından kimlik doğrulaması yapılabilmesi için UML aktivite şeması gösterilmiştir.



Şekil 5.15. Kimlik Doğrulama Aşamasının UML aktivite şeması

5.6. Sistemin Performans Ölçütlerinin Değerlendirilmesi

Önceki bölümlerde anlatıldığı üzere parmak izi tanıma sisteminin başarısını test etmek için performans ölçütleri kullanılmaktadır. Bu ölçütler, yanlış kabul oranı (FAR-False Acceptance Rate), yanlış red oranı (FRR-False Reject Rate), başarı oranı ve eşit hata oranı (EER- Equal Error Rate) değerlerinden oluşmaktadır. Yanlış kabul oranı, bir karşılaştırma sonucundaki yanlışlıkla kabul edilen farklı parmak izlerinin sayısının, toplam parmak izlerinin sayısına oranıdır. Yanlış red oranı ise, aynı parmak izinin farklı karşılaştırmalardaki reddedilme sayısının, parmak izlerinin toplam sayısına oranıdır. Karşılaştırma başarı oranı, karşılaştırılan iki parmak izindeki ortak özellik noktalarının benzerlik oranını vermektedir. Eşit hata oranı EER, aynı kesişme noktasında oluşan hata oranıdır. Yüzde olarak ifade edilen EER, yanlış red oranının yanlış kabul oranına eşit olduğu noktayı temsil etmektedir (Maltoni ve diğ., 2009). Sistemin güvenirliliğini artırmak için EER değerinin düşük bir seviyede olması gerekir. Bu oran, bir biyometrik sistemin doğruluğunu belirlerken kullanılan en önemli kriterdir. Sistemin performans değerlendirmesi için FAR ve FRR oranları kullanılmıştır. FAR ve FRR oranlarını hesaplamak için (5.1) ve (5.2) denklemleri kullanılmıştır (Maltoni ve diğ., 2009).

Biyometrik sistemlerde, karşılaştırılan benzer özelliklerin sayısı önceden belirlenen bir sınır (threshold) değeriyle kıyaslanmaktadır. Bu değerden fazla benzer özellik varsa geçiş izni verilmekte, yoksa reddedilmektedir. Bu sınır değeri değiştirilerek sistemin güvenlik hassasiyeti kolay bir şekilde değiştirilebilmektedir. Geliştirilen sistemde bu sınır değerinin değiştirilebilmesi için, SGFDxSecurityLevel güvenlik seviyesi ayarları kullanılmıştır (Secugen, 2013). Bir biyometrik sistem, yetkili bir kişiyi reddettiğinde FRR, reddetmesi gereken birini onayladığında ise buna FAR denmektedir. Burada sistemin güvenirliliği açısından FAR hatalarından kaçınılması gerekmektedir.

Sistemin performans ölçütlerine göre test edilerek değerlendirilebilmesi için, yirmi kişiye ait aynı parmak izinin çeşitli örneklerinden bir veritabanı oluşturulmuştur. Yanlış kabul ve red oranlarını hesaplamak için, veritabanında ki bir kişiye ait parmak izi, kendisinin diğer izleriyle beraber veritabanında bulunan diğer kişilere ait parmak izleriyle karşılaştırılmış ve bunun sonucuna bakılmıştır. Bunun sonucunda, başka bir kişiye ait parmak izinin bu kişiye ait bir parmak izi olma

sonucu yanlış kabul oranı olarak hesaplanmıştır. Kişinin kendisine ait parmak izini tanımaması ise yanlış reddedilme oranı olarak hesaplanmıştır.

$$(\%)\text{FAR} = \frac{\text{Yanlış Kabul Edilen Parmak izi Sayısı}}{\text{Toplam Karşılaştırma Sayısı}} \text{x}100 \quad \textbf{(5.1)}$$

$$(\%)\text{FRR} = \frac{\text{Yanlış Reddedilen Parmak izi Sayısı}}{\text{Toplam Karşılaştırma Sayısı}} \text{x}100 \quad \textbf{(5.2)}$$

5.6.1. Sonuçların Analizi

Sistemin performans analizinin test edilebilmesi için yirmi kişiden alınan, aynı parmağa ait dört farklı parmak izi örneği ile toplam seksen parmak izi görüntüsünden oluşan bir veritabanı kullanılmıştır. Elde edilen karşılaştırma sonuçlarına göre bu sistemin sahte parmak izlerinden gerçeklerini güvenli bir şekilde ayıt edebildiği görülmüştür. Kullanılan Minutiae algoritması ile farklı parmak izlerinin uygun bir eşik değeri seçildiğinde iyi bir doğruluk oranı elde edildiği görülmüştür. Karşılaştırma işleminde SGFDxSecurityLevel güvenlik seviyesinin SL_HIGH değerine ayarlanarak % 70 sınır (threshold) değerinde olması sağlanmıştır (Secugen, 2013). Buna göre karşılaştırma işlemi esnasında kimlik tespiti için sınır değeri % 70 seçilmiştir. Bu sınır değerinin altındaki oranlar başka bir ize ait kabul edilmiş, üzerindeki değerler ise aynı iz olarak kabul edilmiştir. Yapılan eşleştirmelerin sonucunda % 70 sınır değeri seçildiğinde bir kişinin başka bir parmak izini yanlışlıkla kabul etmesi iki kez, kendi parmak izini yanlışlıkla reddetmesi sekiz kez gerçekleşmiştir. Bu veriler doğrultusunda % 70 sınır değerine bağlı olarak FAR ve FRR değerleri şu şekilde hesaplanmıştır.

$$\text{FAR} = \% 2{,}5$$
$$\text{FRR} = \% 10$$

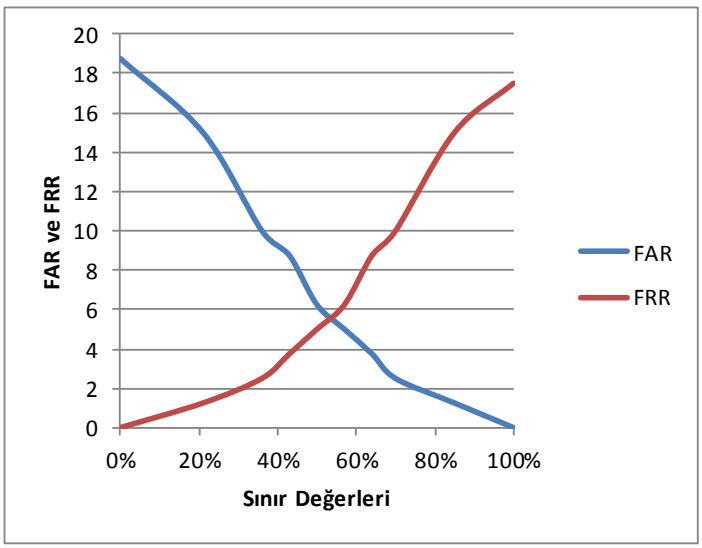
Geliştirilen sistemde parmak izi eşleşmesi için kullanılan sınır değer, eşleşme için belirlenen sayıda noktanın eşleşmesi ve eşleşme oranının % 70'den büyük olması yani giriş Minutiae özellik noktalarıyla taslak Minutiae özellik noktalarının arasındaki eşleşen noktaların sayısının, karşılaştırılan tüm Minutiae özellik noktalarının sayısına oranı 0.70 den büyük olması sağlanmıştır. Güvenlik seviyesi

ayarı kullanılarak farklı sınır değerlerinde yapılan uygulama sonuçları Tablo 5.2'de gösterilmiştir.

Tablo 5.2. Güvenlik seviyesi ayarı kullanılarak yapılan uygulama sonuçları

Sınır Değeri(%)	Yanlış Kabul Edilen Parmak İzi Sayısı	Yanlış Reddedilen Parmak İzi Sayısı	FAR	FRR
% 0	15	0	18,75	0
% 21	12	1	15	1,25
% 36	8	2	10	2,5
% 43	7	3	8,75	3,75
% 50	5	4	6,25	5
% 57	4	5	5	6,25
% 64	3	7	3,75	8,75
% 70	2	8	2,5	10
% 85	1	12	1,25	15
% 100	0	14	0	17,5

Sistemde kullanılan sınır değerin yüksek seçilmesi, yanlış kabul oranını düşürürken, yanlış red oranını artırmıştır. Yanlışlıkla yapılan kabullerin oranının düşük olması, sisteme yetkisiz kişilerin giriş yapabilme olasılığının düşük olduğunu göstermiştir. Sistemin eşik değerinin SGFDxSecurityLevel güvenlik seviyesi ayarı kullanılarak FAR ve FRR oranlarının değiştirilmesiyle sistemin hassasiyetinin de istenildiği gibi ayarlanabildiği görülmüştür. Siteminin hassasiyetinin değiştirilerek elde edilen FAR ve FRR hata oranlarının bu değişimi Şekil 5.16'deki grafikte gösterilmiştir.



Şekil 5.16. Sistemin FAR ve FRR hata oranlarının değişimi

6. SONUÇLAR VE ÖNERİLER

Çalışmada farklı biyometrik doğrulama teknikleri incelenerek mobil arayüz için gereklilikler tespit edilmiş ve Android ortamda taşınabilir bir parmak izi tanıma ve kimlik doğrulama arayüz uygulaması geliştirilmiştir. Uygulamanın ana ekranında iki farklı modül tasarlanarak, sisteme kişinin parmak izinin tanıtılması veya sistemde bir arama yapılarak kimlik sorgulaması gerçekleştirilmiştir. Sonuçlar incelendiğinde, çalışma kapsamında geliştirilen Android tabanlı parmak izi tanıma ve kimlik doğrulama arayüz uygulamasının düzgün çalıştığı, istenilen başarıyı yakaladığı görülmüştür.

Parmak izi tanımlama ve doğrulama aşamalarında, parmak izi görüntüsünün noktasal özelliklerinin elde edilmesi (Minutiae Extraction) ve elde edilen bu özellik noktalarına göre karşılaştırılması sırasında (Minutiae Matching) algoritmaları kullanılarak (Secugen, 2013) Android platformda bir arayüz prototipinin geliştirilerek uygulanabildiği görülmüştür.

Elde edilen karşılaştırma sonuçlarına göre bu sistemin sahte parmak izlerinden gerçeklerini güvenli bir şekilde ayıt edebildiği görülmüştür. Kullanılan Minutiae algoritması ile farklı parmak izlerinin % 70 sınır (threshold) değerinde olması sağlandığında iyi bir doğruluk oranı elde edilebilmiştir.

Geliştirilen sistemde kişi doğrulama ve kimliklendirme amacıyla Android ortamda java nesne serileştirme işlemi uygulanmış ve elde edilen bilgiler SD kartın içerisine dosya biçiminde kaydedilmiştir. Daha sonra bu veritabanı içerisinde herhangi bir kişiden alınan biyometrik parmak izi özelliklerine göre arama yapan kimlik tespiti sorgulama algoritması geliştirilmiştir.

Geliştirilen bu algoritma ile veritabanında arama yapılırken noktasal ayrıntı tabanlı eşleştirme (Minutiae Matching) yönteminin uygulanabilmesi sağlanmıştır. Geliştirilen bu algoritma için, çıkarılan biyometrik parmak izi özelliklerinin nesne yönelimli programlama tekniğinde nesnelerin saklanması için kullanılan seri etme (serialization) yöntemi ile dosyaya saklanması ve arama yapılırken dosyadan seri

etme işlemi ile geri getirilmesi sağlanmıştır. Böylelikle veritabanı üzerinde oluşan yük azaltılarak kimlik belirleme işleminin daha hızlı olması, bunun sonucunda ise geliştirilen sistemin daha performanslı çalışması sağlanmıştır. Geliştirilen kimlik sorgulama algoritması ile sistemde kayıtlı olan kişiler arasında biyometrik özellikli bir arama yapılmış ve bu işlemin sonucuna göre kimlik belirleme işleminin başıyla gerçekleştirilebildiği görülmüştür.

Bu tez çalışması; Android platformda parmak izi görüntüsü ile birlikte farklı biyometrik tanıma sistemlerinin aynı anda kullanımı sağlanarak bir arayüz programı ile birleştirebilir. Android parmak izi tanımanın yanında Android yüz tanıma ve Android ses tanıma algoritmaları kullanılarak daha güçlü bir biyometrik doğrulama ve tanıma arayüz programı geliştirilebilir. Opencv teknolojisi de sisteme entegre edilerek daha gerçek zamanlı bir Android yüz tanıma sistemi gerçekleştirilebilir.

Gerçekleştirilen Mobil Android tabanlı parmak izi tanıma ve kimlik doğrulama sisteminin bir sunucu taraflı çalışması için geliştirilebileceği, aynı anda farklı yerlerden sisteme hem parmak izi tanımlama hem de sistemde bir kimlik doğrulamanın yapılabilmesi sağlanabilir.

7. KAYNAKLAR

Akın H., Karaçam B., Gürpınar K., (2002), *Kimliklendirmede Biyometrik Yöntemlerin Kullanım Alanları*, Yıllık Adli Tıp Toplantıları, Antalya, 48-51

android-apps, (2013). Erişim Tarihi: 10 Eylül 2013,
http://www.android-app-market.com/android-activity-lifecycle.html

android-app-market, (2013), Android Architecture-The Key Concepts of Android OS Erişim Tarihi: 15 Eylül 2013,
http://www.android-app-market.com/android-architecture.html

Bilgigunlugum.net, (2014), Android Aktiviteler, Erişim Tarihi: 01 Şubat 2014,
http://www.bilgigunlugum.net/android/2android_aktivite.html

Chaudhary S., Nath R., (2009). *A Multimodal Biometric Recognition System Based on Fusion of Palmprint, Fingerprint and Face.* Advances in Recent Technologies in Communication and Computing, 2009. ARTCom '09. International Conference on, IEEE Veritabanı.

Cormen T.H., Leiserson C.E., Rivest R.L., (1990). *Introduction to Algorithms*. New York: McGraw-Hill, Erişim Tarihi: 13 Haziran 2013,
http://net.pku.edu.cn/~course/cs101/2007/resource/Intro2Algorithm/IntroductionToAlgorithm_v2_en.pdf

Dadlani C., Passi A. K., Sahota H., Kumar M. K., (2006). *Fingerprint Recognition Using Minutiae-Based Features*, Indian Institue of Technology Delhi, Erişim Tarihi: 15 Mart 2013,
http://www4.comp.polyu.edu.hk/~csajaykr/myhome/teaching/biometrics/final_report.pdf

Developer, A., (2013). Erişim Tarihi: 23 Eylül 2013,
Available at: http://developer.android.com/guide/basics/what-is-android.html

developer, a., (2013). Erişim Tarihi: 23 Eylül 2013,
http://developer.android.com/reference/android/app/Activity.html

Emeraldinsight, (2013). Erişim Tarihi: 27 Eylül 2013,
http://www.emeraldinsight.com/content_images/fig/1610220405001.png

Femila M.D., Irudhayaraj A.A., (2011). *Biometric system*, Electronics Computer Technology (ICECT), 2011 3rd International Conference, IEEE Veritabanı.

Fergytech, (2013). Erişim Tarihi: 1 Ekim 2013, http://www.fergytech.com/android/

Grother P., (2006). *Performance and Interoperability of the INCITS 378 Fingerprint Template*, National Institute of Standards and Technology.
Erişim Tarihi: 01 Temmuz 2013,
http://biometrics.nist.gov/cs_links/minex/minex04/minex_scenario2_benefit.pdf

Henry E., (1900), *Classification and Uses of Finger Prints, Routledge*.

Hıdımoğlu K., (2010). *Web Kamera Kullanımı ile Parmak İzi Tanıma ve Kimlik Tespiti Doğrulama*, Yüksek Lisans Tezi, YTÜ, Fen Bilimleri Enstitüsü, İstanbul.

Hibben M.W., (2010). Erişim Tarihi: 12 Kasım 2013,
http://www.technomicon.com/ElectroPolitics/ElectroPolitics-11-15-10.html

Jain A.K., Hong L., Pankanti S., Bolle R., (1997). *An Identity Authentication System Using Fingerprints*, Proc of the IEEE, vol, 85, no.9,1365-1388, IEEE Veritabanı.

Jain A., Hong L., Bolle R., (1997*). On-Line Fingerprint Verification*,
IEEE Transactions On Pattern Analysis And Machine Intelligence, vol. 19, no. 4, pp.302-314, IEEE Veritabanı.

Jain A.K., Flynn P., Ross A. A., (2007), *Handbook of Biometrics*. Springer Science Business Media, LLC. ISBN-13: 978-0-387-71040-2

Javablog, (2014). Java Object Serialization, Erişim Tarihi: 01 Şubat 2014
Kaynak: http://www.javablog.org/55-java-object-serialization

Kakıcı A. (2013). *Biyometrik Tanıma Sistemleri*. Erişim Tarihi: 01 Mayıs 2013,
http://www.ahmetkakici.com/genel/biyometrik-tanima-sistemleri/

Karu K., Jain A.K., (1996) *Fingerprint Classification, Pattern Recognition*,
vol. 29, no. 3, pp. 389-404.

Lee H.C., Gaensslen R.E., (1991). *Advances in Fingerprint Technology*,
New York: Elsevier.

Maltoni D., Maio D., Jain Anil K., Prabhakar S., (2009). *Handbook of Fingerprint Recognition*, Springer-Verlag London Limited

Mobi Thinking, (2013). Erişim Tarihi: 17 Eylül 2013,
http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats#subscribers

Mudholkar S.S., Shende P.M., Sarode M.V., (2012), *Biometrics Authentication Technique For Intrusion Detection Systems Using Fingerprint Recognition*, International Journal of Computer Science, Engineering and Information Technology, (IJCSEIT), Vol.2, No.1

Ranade A., Rosenfeld A., (1993). *Point Pattern Matching by Relaxation*, Pattern Recognition, vol. 12, no. 2, pp. 269-275.

Rao A.R., (1990). *A Taxonomy for Texture Description and Identification*. New York: Springer-Verlag.

Ratha N., Chen S., Jain A.K., (1995). *Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images*, Pattern Recognition, vol. 28, no. 11, pp. 1,657-1,672.

Secugen, (2013). Programming Manual for FDx SDK Pro for Android Erişim Tarihi: 10 Nisan 2013, http://www.secugen.com/products/sdk_pro.htm#android

Sönmez E.B., Özbek N.Ö., Özbek Ö. (2007). *Avuç İzi ve Parmak İzine Dayalı Bir Biyometrik Tanıma Sistemi,* Akademik Bilişim'07 - IX. Akademik Bilişim Konferansı Bildirileri, Dumlupınar Üniversitesi, Kütahya

Şamlı R., Yüksel E., (2009). *Biyometrik Güvenlik Sistemleri*, Akademik Bilişim'09 - XI. Akademik Bilişim Konferansı Bildirileri, Harran Üniversitesi, Şanlıurfa

Technomicon, (2013). Erişim Tarihi: 20 Eylül 2013, http://www.technomicon.com/TechnomiconImages/ElectroPoliticsImages/EP-11-15-10Images/JavaAndroidFlowchart.jpg

Tilton C.J., (2009). *Biometric Standards - Overview*, Daon, Erişim Tarihi: 01 Eylül 2013, http://www.nws-sa.com/biometrics/Biometric_Standards_White_Paper_March2009.pdf

Ton J., Jain A.K., (1989). *Registering Landsat Images by Point Matching*, IEEE Transactions Geoscience and Remote Sensing, vol. 27, no. 5, pp. 642-651, IEEE Veritabanı.

Varlık A., (2008). *Digital Fotogrametri Teknikleri ile Kişi Tanıma*, Doktora Tezi, S.Ü. Fen Bilimleri Enstitüsü, Jeodezi ve Fotogrametri Anabilim Dalı.

8. ÖZGEÇMİŞ

1984 yılında İstanbul'da doğdu. İlköğrenimini ikamet ettiği Sultanbeyli'de tamamladı. Daha sonra, Ümraniye Teknik Lisesi Bilgisayar Yazılım bölümünü 2003 yılında bitirdi. 2005 yılında Marmara Üniversitesi, Teknik Eğitim Fakültesi, Bilgisayar ve Kontrol Teknolojisi Öğretmenliğinde başlamış olduğu lisans eğitimini 2009 yılında tamamladı. 2009 yılında, İstanbul Arel Üniversitesi Meslek Yüksek Okulu Elektronik Bölümü kadrosunda Öğretim Görevlisi olarak çalışmaya başladı. 2013 yılından itibaren İstanbul Arel Üniversitesinde Bilgisayar Mühendisliği Bölümünde Rektörlük görevlendirmesi ile çalışmaktadır. 2011-2014 yılları arasında Haliç Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği programında Yüksek Lisans eğitimini aldı.