****

**BM495 BİTİRME PROJESİ**

**KULAK İZİ TANIMA SİSTEMİ**

*Mertcan Durmuş*

141180024

**ANKARA**

İçindekiler

[Özetçe 3](#_Toc42764656)

[EAR PRINT RECOGNITION SYSTEM 3](#_Toc42764657)

[1. Giriş 3](#_Toc42764658)

[1.1. Literatür Taraması 3](#_Toc42764659)

[2. Metodoloji ve Metotlar 5](#_Toc42764660)

[2.1. Python 5](#_Toc42764661)

[2.2. Java 5](#_Toc42764662)

[2.3. Opencv 5](#_Toc42764663)

[2.4. Anaconda Yazılım Geliştirme Ortamı 6](#_Toc42764664)

[2.5.AngularYazılım Geliştirme Çatısı 6](#_Toc42764665)

[2.6. Keras Kütüphanesi 7](#_Toc42764666)

[2.7. Scikit-Learn Kütüphanesi 7](#_Toc42764667)

[5. GERÇEKLEŞTİRİLEN ÇALIŞMALAR 7](#_Toc42764668)

[5.1. Projede kullanılacak veritabanları 7](#_Toc42764669)

[5.1. Kulak izi tanımlamak için kullanılacak metrikler 7](#_Toc42764670)

[5.2. Metriklerin Opencv ile Gerçekleştirimi 9](#_Toc42764671)

[5.3. Derin öğrenme metotları ile kulak izi kullanılarak cinsiyet belirleme 12](#_Toc42764672)

[5.4. Kulak izi kullanarak yeniden kullanıma yönelik kimlik doğrulama sistemi 14](#_Toc42764673)

[Sonuç 17](#_Toc42764674)

[Kaynaklar 18](#_Toc42764675)

# Özetçe

Bu projede kulak izinden kimlik tanımlama işleminin yazılımsal olarak gerçekleştirimi yapılmıştır. Öncelikle görüntü işleme teknikleri ile 2 boyutlu fotoğraf veritabanları üzerinden görüntüler üzerinde işlem yapılabilir hale getirildi. Ardından belirlenen özellik çıkarım teknikleri ile belirlenmiş benzersiz kabul edilebilir özellik vektörleri çıkarıldı. Bu metrikler makine öğrenmesi modelleri üzerinden eğitilecek modeller ile tahmin edilebilir bir noktaya taşınmaya çalışıldı. Son olarak belirlenmiş metrikler kullanılarak bir doğrulama sistemi web uygulaması olarak ortaya kondu.

*Anahtar kelimeler: Kulak izi, web, doğrulama, güvenlik, görüntü işleme*

# EAR PRINT RECOGNITION SYSTEM

In this project, software identification of the earprint was carried out software. First of all, with image processing techniques, images can be processed on 2D photo databases. Subsequently, unacceptable feature vectors determined by the feature extraction techniques were removed. These metrics were tried to move to a predictable point with the models to be trained through machine learning models. A validation system has been introduced as a web application, using the recently determined metrics.

*Keywords: Earprint, web, verification, security, image processing*

# 1. Giriş

Bu çalışmada kulak izinden kimlik tanıma işleminin yazılımsal olarak gerçekleştirimi için yapılmış çalışmalar raporlanacaktır. Bu çalışmalar veri toplama, veri ön işleme, kimlik tanımlama için kullanılabilecek benzersiz metrikleri belirleme, bu metrikleri görüntü üzerinden çıkarma, çıkarılmış özellik vektörlerini makine öğrenmesi algoritmaları ile tanınacak bir model oluşturma sonrasında da bu modeli ve metrikleri web platformunda gerçek zamanlı olarak ortaya koyma işlemi gerçekleştirilecektir.

## Literatür Taraması

Pflug ve diğerleri [1] IET BIOMETRICS adlı kitap aracılığıyla kulak izi için belirli metrikler ortaya çıkarma ve bunları 2 boyutlu,3 boyutlu görseller üzerinde benzersiz kulak izi tanıma işleminin nasıl gerçekleştirileceği aşama aşama anlatmışlardır.. Bu çalışma temel olarak kulak izini benzersiz olarak elde edeceğimiz metriklerin ortaya çıkarılmasını amaçlamıştır ve bu alanda bir yayındır.

Wintershein ve diğerleri [2] “International Conference on Intelligent Information Hiding and Multimedia Signal Processing” konferansı üzerinden bizimle buluşan bir çalışma yapmışlardır. Bu çalışma da kulak tanıma metriklerinin çoğaltılmasına karşılık kulak yakalama alanında çalışmaların az olmasıdır. Bu çalışmada görüntü üzerinde ve gerçek zamanlı olarak kulak image’inin nasıl yakalanacağı ve işleneceği hale nasıl getirileceği temel alınmıştır.

Busch ve diğerleri [3] ISB (International Conference on Biometrics) 2013 konferansında yayınlanan bir çalışma yapmışlarıdır. Bu çalışma da görüntü üzerinde kulak kısmını yakalamak ve işlemeye hazır hale getirmek aşamalarını otomatize etmek üzerine kuruludur. Bu işlemin nasıl yapılacağı bu çalışmada anlatılmıştır.

Ross ve diğerleri [4] IJCB (International Joint Conference on Biometrics) konferansında yayınlanmış bir çalışma yapmışlardır. Bu çalışma da denetimsiz (unsupervised) makine öğrenmesi teknikleri ile kulak veritabanları üzerinde belirli kümeleme ve metrik çıkarma (feature extraction) işlemleri gerçekleştirmişlerdir.

Yuan ve diğerleri[5] “international wokshop on biometric person authantication” konferansında kulak biyometrisi ile ilgili çalışmalarını yayınlamışlar ve bu çalışmada kulak veritabanı oluşturma, kulak algılama ve kulak tıkanıklığı(veri toplama konusunda) problemleri için çözüm önerileri getirmişlerdir. Bunun yanında 2 boyutlu ve 3 boyutlu kulak izi tanıma ile ilgili çalışmalarını sunmuşlardır.

Chen ve Diğerleri [6] [IEEE Transactions on Pattern Analysis and Machine Intelligence](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34) aracığıyla 3 boyutlu kulak görselleri için kulak algılama, kulak tanımlama ve doğrulama çalışmasını gerçekleştirmişlerdir. Kulak tespiti için tek bir referans model kullanmışlardır. Kulak sarmalını ve antihelix parçalarını bulan yeni bir yaklaşım önermişlerdir.

Hurley ve diğerleri [7] Computer vision and image understanding aracılığı ile çalışmalarını yayınlamışlardır. Çalışmalarında kulak çevresine gauss kuvvet alanı uygulamışlar ve kulak biyometrisini daha aşikar ortaya çıkarmaya çalışmışlardır.

# 2. Metodoloji ve Metotlar

Kulak izinden kimlik tanıma işlemi parmak izi veya yüz izi ile karşılaştırıldığında çok daha etkin bir yöntem olduğu görülmektedir. Kulak izi “stable biometric” olarak yani zamanla eskimeyen, değişmeyen ve renk bakımından da oldukça sabit olması son olarak çevresel faktörlerden yaşlanma, yıpranma gibi çok az zarar görmesi kulak izi tanımayı parmak izi tanıma ya da yüz izi tanımaya karşı öne çıkarmıştır. Bu çalışma ile kulak izi tanıma yazılım ortamında gerçekleştirilmesi planlanmaktadır.

## 2.1. Python

Python, nesne yönelimli, yorumlamalı, etkileşimli bir programlama dilidir. Girintilere dayalı basit sözdizimi, dilin öğrenilmesini ve akılda kalmasını kolaylaştırır. Modüler yapıdadır. Windows, unix, Linux ve dağıtımları, mac, amiga gibi hemen hemen her platformada ve işletim sisteminde çalışabilir. Python ile sistem programlama, kullanıcı ara yüzü programlama, web programlama, uygulama yazılımı geliştirme gibi birçok alanda yazılım geliştirebilirsiniz. Büyük yazılımların hızlı bir şekilde prototiplerinin üretilmesi ve denenmesi gerektiği durumlarda tercih edilir. Çünkü diğer c ve c# gibi dillere kıyasla çok daha az bir kod satırı ile çok fazla bir iş yükünü yerine getirebilir. Geliştirilmesi 1990 yılında guido van rossum tarafndan amsterdam’da başlamış ve günümüzde halen devam etmektedir.

Çalışmamızda da python dilini tercih etmemiz bize makine öğrenmesi algoritmalarını hazır kütüphaneler ile sunması ve oldukça hızlı olmasıdır.

Python kurulumu için python.org sitesi üzerinde işletim sistemi seçimi yapılarak indirme işlemi ve aynı adreste bulunan kurulum rehberi ile kurulum işlemi gerçekleştirilebilir.

## 2.2. Java

Java, sınıf tabanlı, nesne yönelimli ve mümkün olduğunca az uygulama bağımlılığına sahip olacak şekilde tasarlanmış genel amaçlı bir programlama dilidir. Uygulama geliştiricilerin bir kez yazmasına, her yerde çalışmasına izin vermesi amaçlanmıştır. Derlenmiş Java kodu Java'yı yeniden derlemeye gerek kalmadan tüm platformlarda çalışabilir anlamındadır. Java uygulamaları genellikle, temel bilgisayar mimarisinden bağımsız olarak herhangi bir Java sanal makinesinde çalışabilen bayt koduna derlenir. Java sözdizimi C ve C ++ 'ya benzer, ancak her ikisinden de daha az düşük seviyeli olanaklara sahiptir [16].

## 2.3. Opencv

Opencv (Open Source Computer Vision) açık kaynak kodlu görüntü işleme kütüphanesidir. C ile programlanmaya başlanmış sonrasında ise birçok algoritması c++ ile kodlanmış ve c++ il kodlanmaya devam etmektedir. Açık kaynak kodlu bir kütüphanedir ayrıca BSD lisansı altında geliştirilmektedir. BSD lisansına sahip olması bu kütüphaneyi istediğiniz projede ücretsiz olarak kullanabileceğiniz anlamına gelir.  OpenCV platform bağımsız bir kütüphanedir. C++, C, Python, Java, Matlab, kütüphanesi aracılığıyla da programlama dilleri ile kolaylıkla OpenCV uygulamaları geliştirilebilir [8].

Bileşenleri;

Core: OpenCV’nin temel fonksiyonları ve matris, point, size gibi veri yapılarını bulundurur. Ayrıca görüntü üzerine çizim yapabilmek için kullanılabilecek metotları ve XML işlemleri için gerekli bileşenleri barındırır.

HighGui: Resim görüntüleme, pencereleri yönetme ve grafiksel kullanıcı arabirimleri için gerekli olabilecek metotları barındırır.

Imgproc: Filtreleme operatörleri, kenar bulma, nesne belirleme, renk uzayı yönetimi, renk yönetimi ve eşikleme gibi neredeyse tüm fonksiyonları içine alan bir pakettir.

Imgcodecs: Dosya sistemi üzerinden resim ve video okuma/yazma işlemlerini yerine getiren metotları barındırmaktadır.

Videoio: Kameralara ve video cihazlarına erişmek ve görüntü almak ve görüntü yazmak için gerekli metotları barındırır.

## 2.4. Anaconda Yazılım Geliştirme Ortamı

Makine öğrenmesi ve veri bilimi için gerekli araçların tamamını bünyesinde barındıran ve bize sunana yazılım geliştirme platformlarının bütünüdür. Bu noktada ihtiyacımız olan kütüphaneleri gerekli dosya ve dizin işlemleri ile birlikle bize sunması bu platformu tercih etmemize neden olmuştur. <https://www.anaconda.com/> adresine giderek işlem yapılacak olan işletim sistemi seçimiyle birlikte indirilebilir. Sistem gereksinimlerini özetlemek gerekirse en az 4 gb ram bellek ve 2.8 ghz hızında 4 çekirdek ihtiyacı duyar. Detaylı sistem gereksinimlerine <https://docs.anaconda.com/anaconda-enterprise/system-requirements/> adresinden erişilebilir [11].

## 2.5.AngularYazılım Geliştirme Çatısı

Angular, Google tarafından geliştirilen ve desteklenen SPA(Single Page Application) uygulama yapılmasına olanak sağlayan javascript kütüphanesidir. Single Page Applicationlarda yönlendirme işlemi backend tarafından değil client tarafında yapılır. Bir tane index sayfamız oluyor ve client tarafında sayfalar yönlendirme ile değiştiriliyor ve render ediliyor. Bu sayede yeni bir sayfaya geçmiş etkisi veriyor. Angular modüler bir yapıya sahiptir. Uygulumanızı parçalara bölebilir ve tekrar kullanabilirsiniz. Diğer frameworklere göre daha esnek bir yapıya sahiptir. Sizlere daha fazla kontrol edebileceğiniz alanlar sunar. Diğer SPA uygulamalarından ayıran en büyük farkı ise paketler ile birlikte gelmesidir.

## 2.6. Keras Kütüphanesi

Keras; tenserflow ve theano derin ögrenme kütüphaneleri backend olarak kullanan bu kütüphaneler üzerinde geliştirilmiş model oluşturmayı ve geliştirmeyi oldukça kolaylaştıran pyhon modülüdür. Config dosyasında değişiklil ile tenserflow ile theano arasında geçiş yapılabilmektedir. Temelde sinir ağı modelleri oluşturmak için kullanılmaktadır. Açık kaynak kodludur [10].

## 2.7. Scikit-Learn Kütüphanesi

Scikit-learn, denetimli ve denetimsiz öğrenmeyi destekleyen açık kaynaklı bir makine öğrenme kütüphanesidir. Ayrıca model kurma, veri ön işleme, model seçimi, değerlendirme ve diğer birçok yardımcı program için çeşitli araçlar sunar [9].

# 5. GERÇEKLEŞTİRİLEN ÇALIŞMALAR

## 5.1. Projede kullanılacak veritabanları

Bu çalışmada kullanılacak olan veritabanları;

AMI EAR DATABASE [12]

IIT DELHI EAR DATABASE [13,14]

EarVN1.0 DATASET [15]

Toplamda 30.000 üzeri kulak görüntü verisi bulunmuştur.

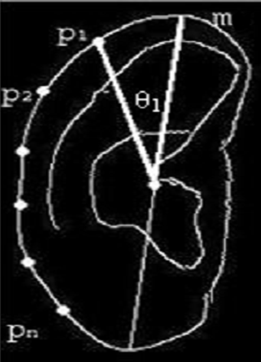
## 5.1. Kulak izi tanımlamak için kullanılacak metrikler

İki adet feature vektörü ile kulak izi tanıma işlemi gerçeklenmeye çalışılacaktır. Bu feature vektörleri grayscale image üzerinden elde edilmiştir. Öncelikle eldeki görüntü grayscale, gri ve çizgisel bir hal aldırılacaktır. Bunu yapmak için opencv kütüphanesinden yararlanılacaktır. Sonrasında bu gri yapıda kulağn dışından geçen en uzun çevre Öklid mesafesi kullanılarak hesaplanacaktır. Bu çevreyi n eşit parçaya bölen n+1 çizgi merkezden başlanarak sağa ve sola çizilecektir. Bu çizgiler ile en uzun çevrenin kesişim noktalarıyla merkez arasındaki açı bizi ilk özellik vektörümüz olacaktır.

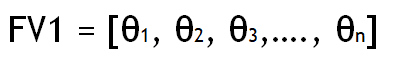


Şekil 5.1.1 çevre ile paralellerin kesişim noktaları

Şekil 5.1.1 de görüldüğü gibi en uzun kenar ile paraleller kesiştirilmiştir.



Şekil 5.1.2 çevre ile paralellerin kesişim noktaları ile merkez arasındaki açılar



Bu şekilde açılar birleştirilerek ilk özellik vektörümüz oluşturulmuştur.

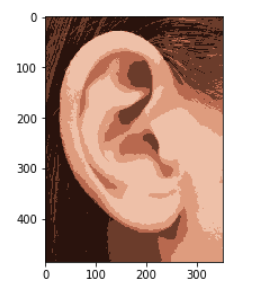
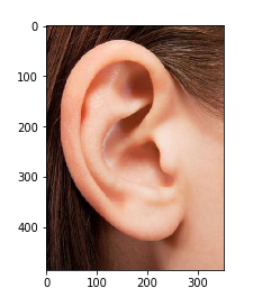
İkinci özellik vektörümüz ise bu paraleller ile en uzun çevre hariç tüm kesişim noktalarımızdır. Bu vektörün yapısına Şekil 5.1.3 aracılığı ile görebilirsiniz.



Şekil 5.1.3 ikinci özellik vektörü

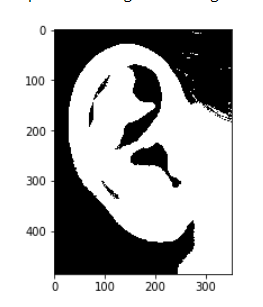
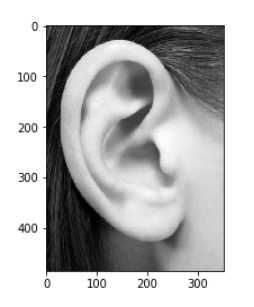
## 5.2. Metriklerin Opencv ile Gerçekleştirimi

Burada çalışma da kullanmak üzere belirlediğimiz metriklerin python dili ve opencv kütüphanesi ile gerçekleştirimi gösterilecektir.ilk olarak görüntü üzerinde segmentasyon işlemi gerçekleştirilmiştir. Böylece gürntü üzerinde kenar değerlerini daha kolay ve belirgin bulmak amaçlanmıştır.



Şekil 5.2.1 knn segmentasyon

Burada resim üzerinde knn makine öğrenmesi algoritması kullanılarak segmentasyon işlemi gerçekleştirilmiş ve böylece eşik değerlerini daha belirgin olarak elde etmek amaçlanmıştır. Bu işlem görüntüdeki pikselleri değerlerine göre knn algoritmasına sokarak bu algoritmadan birbirine daha yakın olan piksel değerlerinin kendisi dönmüştür.



Şekil 5.2.2 gri-hal segmentasyonu

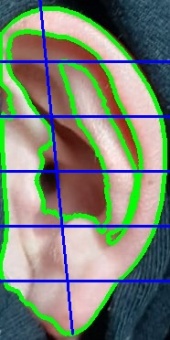
Burada görsel gri renklere yani her piksel 0-1 arasında bir değere eşitlenmiştir. Burada bir gray-scale adı verilen segmentasyon yani bir çeşit kümeleme işlemi uygulanmıştır. Fakat knn algoritmasındaki kadar başarılı bir çıktı elde edilememiştir. Bunun sebebi elde etiğimiz görüntü kulak biyometrisini ortaya çıkarmış olsa da görüntünün sağ alt kısmında da görüldüğü üzere yüzü kulak kısmı ile birlikte gruplamıştır. Yani yüz kısmı da beyaz kabul edilmiş öyle bir çıktı gösterilmiştir.

Segmentasyon işleminde sonra eşik değerleri kendi yazdığımız bir python script’i üzerinden elde edilmiş ve bu eşik değerleri kullanılarak contourlar yani çerçeveler bulunmuştur.



Şekil 5.2.3. bulunan çerçeveler

Şekil 3.6’da resim üzerinde segmentasyon işlemi yapıldıktan sonra bulunan çerçeveler görülmektedir.sonrasında bu çerçeveler iç ve dış olarak gruplandırılmıştır ve dış çerçevenin en uzun köşegeni Öklid mesafesi kullanılarak tespit edilmiştir.



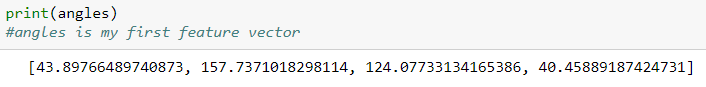
Şekil 5.2.4 bulunan çerçeveler ve paraleller

Şekil 5.2.4’da buluna dış çerçeveye Öklid mesafesi kullanılarak bulunan en uzun köşegen ve bu köşegenin merkezinden de geçecek n adet paraleller çizdirilmiştir. Bu aşamada sonra bu paraleller ile dış çerçevenin kesişin noktaları piksel olarak hesaplanacak ve merkezden geçek doğrular ile birleştirilecektir.



Şekil 5.2.5. merkez ve kesişim noktalarının birleşmesi

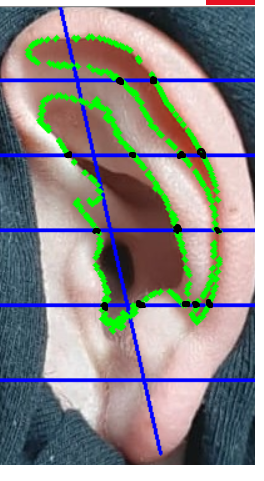
Şekil 5.2.5’de merkez ile kesişim noktaları birleştirilmiştir. Bu noktadan sonra bu doğrular ile en uzun köşegen arasındaki açılar hesaplanacak ve bu bizim ilk özellik vektörümüz olacaktır.



Şekil 5.2.6. birinci özellik vektörü

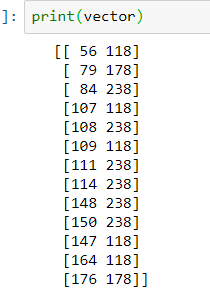
Son olarak iki doğru yani kesişim noktalarına çizilen doğru ve en uzun köşegen doğrusu arasındaki açı hesaplanmış ve bizim ilk özellik vektörümüz oluşmuştur.

Daha sonrasında ise ikinci özellik vektörümüzü ortaya çıkardık. İkinci özellik vektöründe açılar yerine kesişim noktalarının konumlarının kullanılmasının asıl amacı açılara nazaran konumların kişiden kişiye daha fazla oranda farklılık göstermesidir.



Şekil 5.2.7. iç çevre ve paralel doğrular

Şekil 5.2.7’de iç çevremiz ve paralel doğrular görülmektedir. Burada paraleller ile iç çevredeki çemberlerin kesişim noktaları bizim ikinci özellik vektörümüzdür. Yani buradaki siyah noktaların piksel olarak konumları bizim ikinci özellik vektörümüzdür.



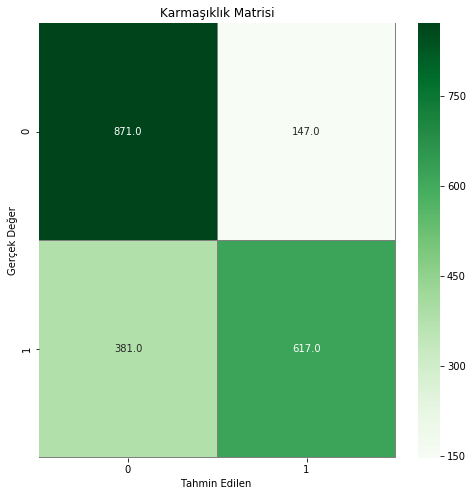
Şekil 5.2.8. ikinci özellik vektörümüz

Şekil 5.2.9.’da ikinci özellik vektörümüzü oluşturan noktaların konumları gösterilmiştir.

Böylece kulak izi tanımlama için kullanılabilecek benzersiz metrikleri ortaya çıkarmış oluyoruz.

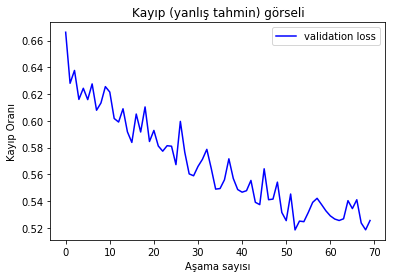
## 5.3. Derin öğrenme metotları ile kulak izi kullanılarak cinsiyet belirleme

Burada elimizdeki veri setlerini kullanarak kulak görüntüleri üzerinden cinsiyet belirleme işlemini gerçekleştirmeyi amaçladık. Knn ve cnn ve lineer regresyon algoritmaları ile üç ayrı model oluşturduk. Verileri kadım ve erkek olarak etiketledikten sonra algoritmalarımızı bu veriler üzerinde egitip test ettik. Burada knn algoritması 0.758 doğruluk katsayısı(accuracy) değeri döndürdü. Lineer regresyon algoritması 0.80 doğruluk katsayısı değeri döndürdü. Son olarak da cnn algoritması 0.95 doğruluk katsayısı değeri döndürdü.



Şekil 5.3.1 karmaşıklık matrisi

Şekil 5.3.1’de oluşturulan cnn algoritması için karmaşıklık matrisi gözükmektedir. Burada çok başarılı çıktılar elde edemediğimiz için model üzerinde belli başlı parametre iyileştirmeleri yaparak gözlemlerimizi tekrarladık.



Şekil 5.3.2 kayıp oranı

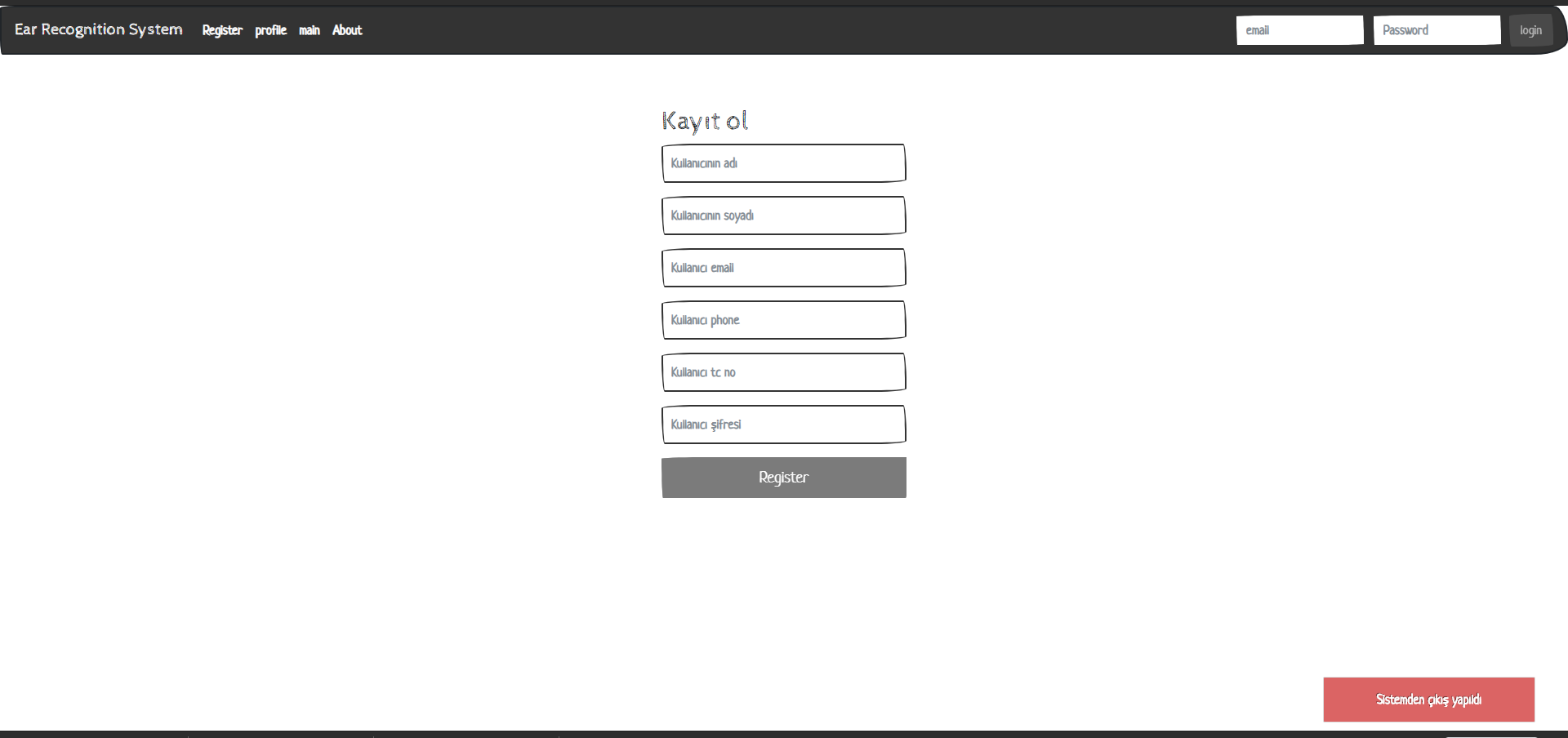
Şekil 5.3.2’de oluşturulan cnn algoritması için farklı aşama değerleri ile cinsiyeti doğru belirleme arasında ki ilişki gösterilmiştir. Buradan hareketle algoritmanın eğitim aşama sayısı arttıkça kadın ve erkek kulak görüntüleri daha iyi ayrılmıştır.

Son olarak kendi belirlediğimiz metrikler ile cinsiyet belirleme işlemini gerçeklemek istedik. Burada dış kulak çerçeve uzunluğunu tek metrik kabul ederek işlem gerçekleştirdiğimiz de erkek kulak görsellerinin kadın kulak görsellerinden açık ara farklı olduğunu tespit ettik .bir erkek için deger 2000’li rakamlar etrafında hareket ederken kadın kulak görselleri için bu rakam 800-1200 arasında değişmektedir. Haliyle tek bir metrik kullanarak bu işlemi gerçekleştirebileceğimize karar verdik.

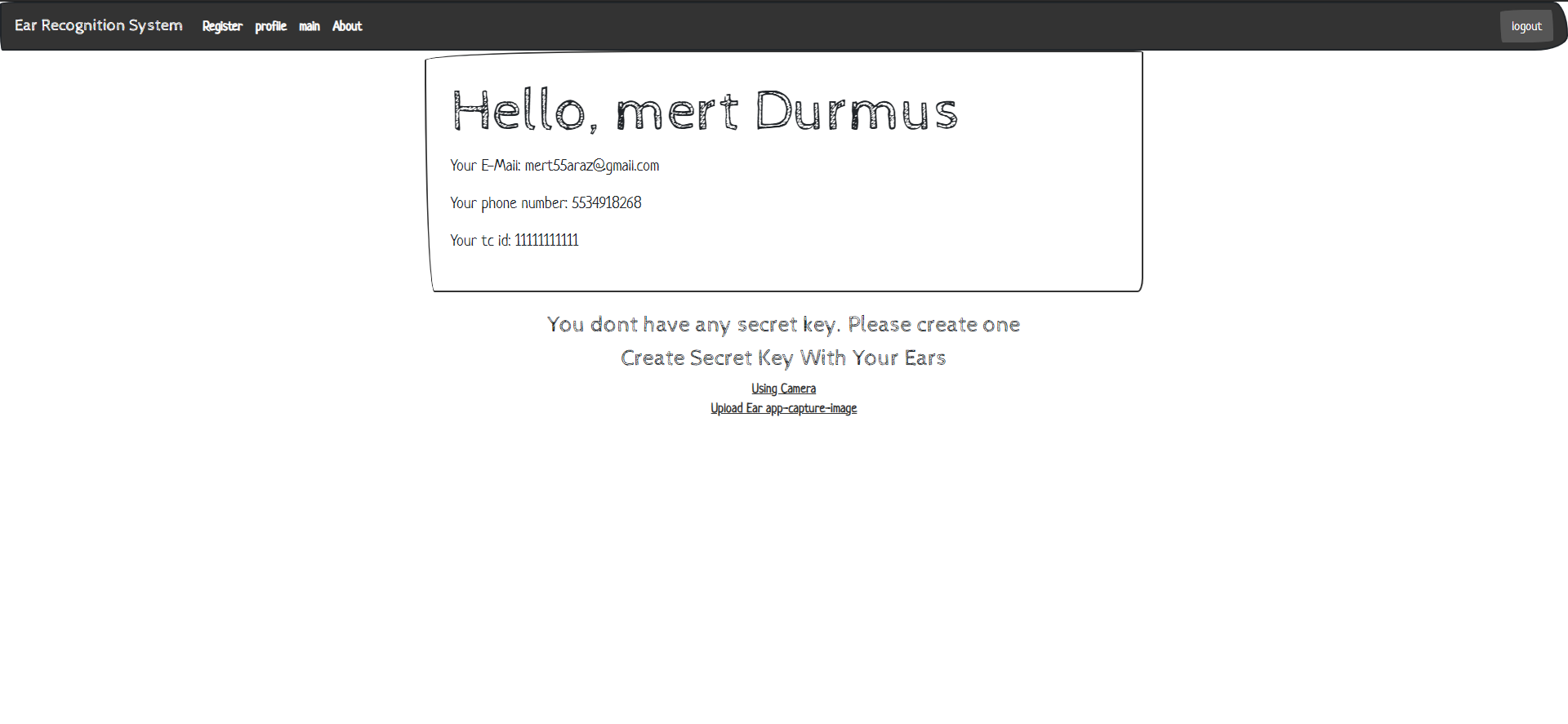
Bütün bu çalışmaları bir mobil uygulama bünyesinde gerçek zamanlı çalışır hale getirmek bir sonraki temel amacımız olacaktır.

## 5.4. Kulak izi kullanarak yeniden kullanıma yönelik kimlik doğrulama sistemi

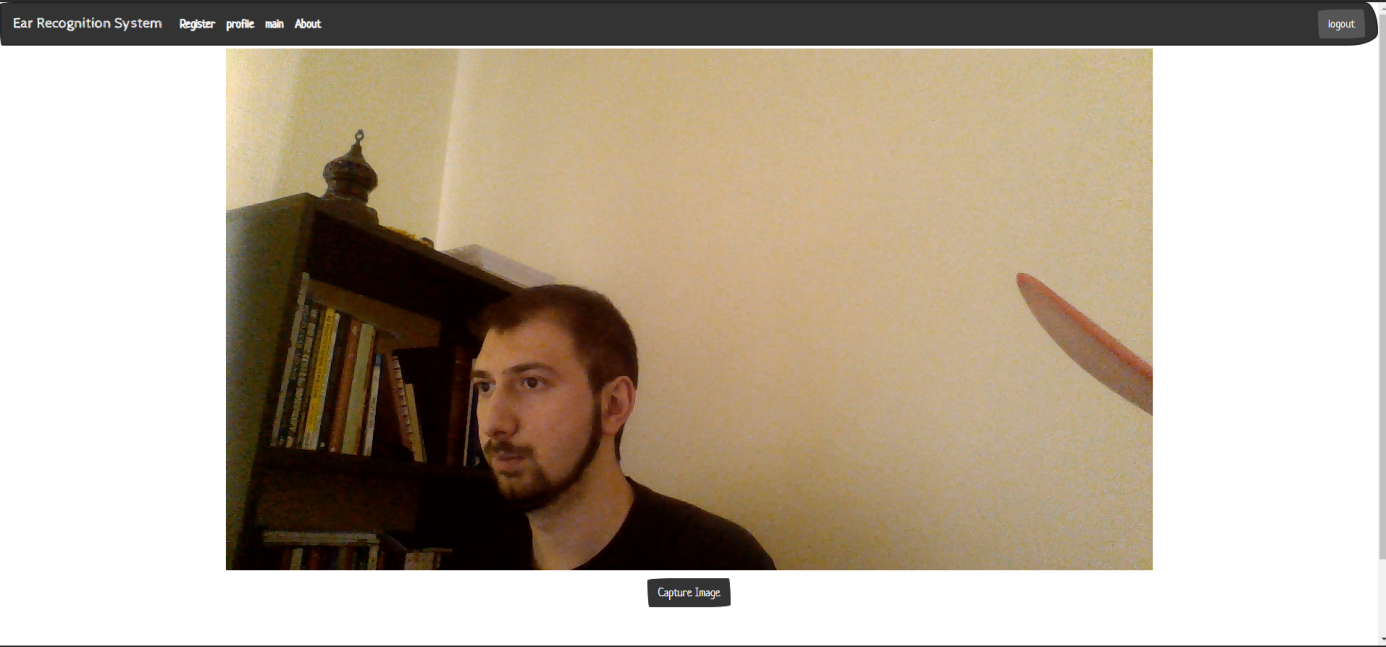
Bu kısımda elimizde var olan metrikler ile belirlediğimiz kulak tanımaya yönelik özellik vektörleri ile kullanıcının istediği platforma ekleyebileceği ikincil bir doğrulama katmanı gerçekleştirilmiştir. Bu katman google doğrulama sistemi gibi çalışmaktadır. Tek farklı google bu işlemi karekod ve mail hesabı ile yaparken biz bu işlemi kulak izi ile yapmış olduk. Yani biyometrik veriyi kullanarak dada güvenli bir doğrulama katmanı gerçekleştirimi oluşturduk.



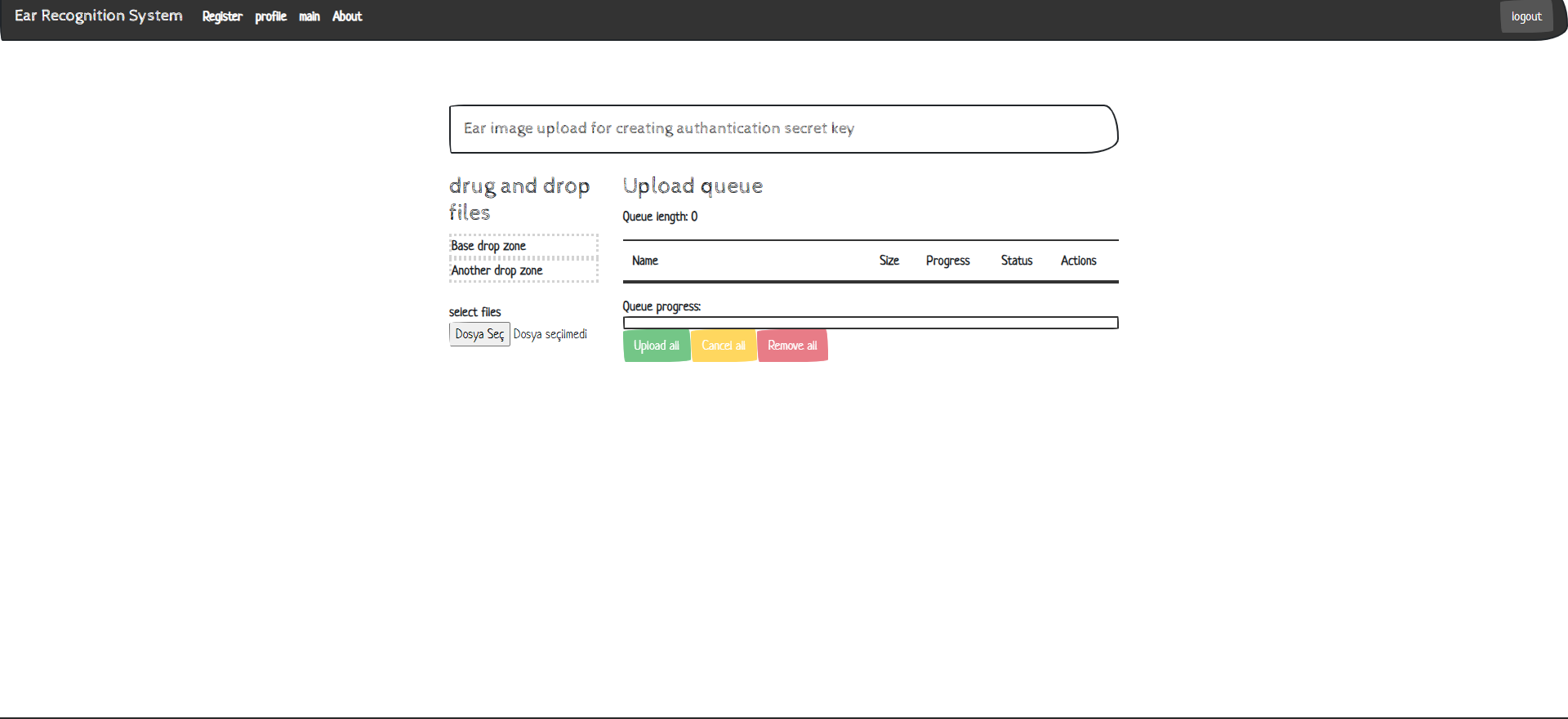
Şekil 5.4.1 Sistem ekran görüntüsü



Şekil 5.4.2 Sistem ekran görüntüsü

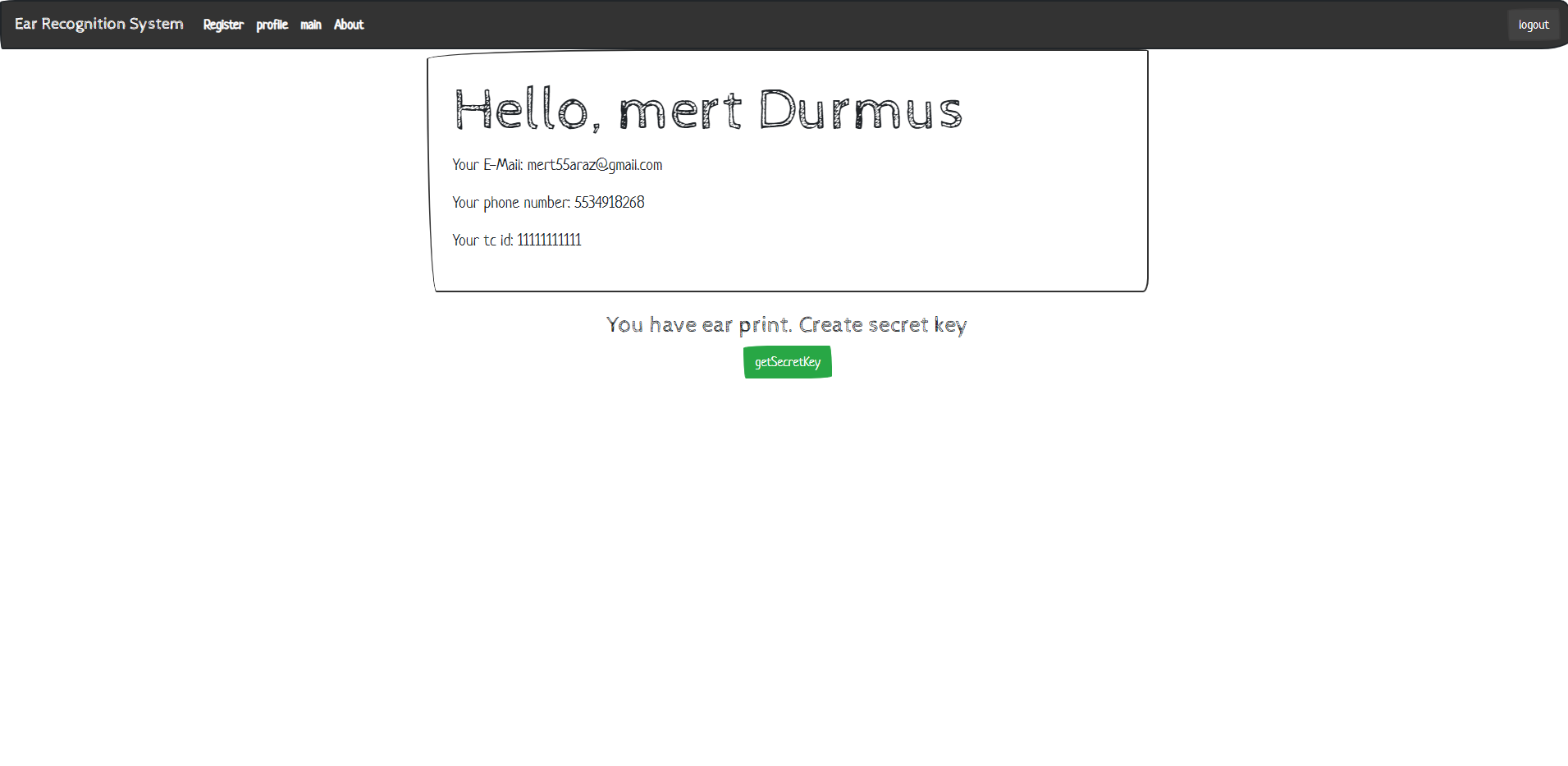


Şekil 5.4.3 Sistem ekran görüntüsü

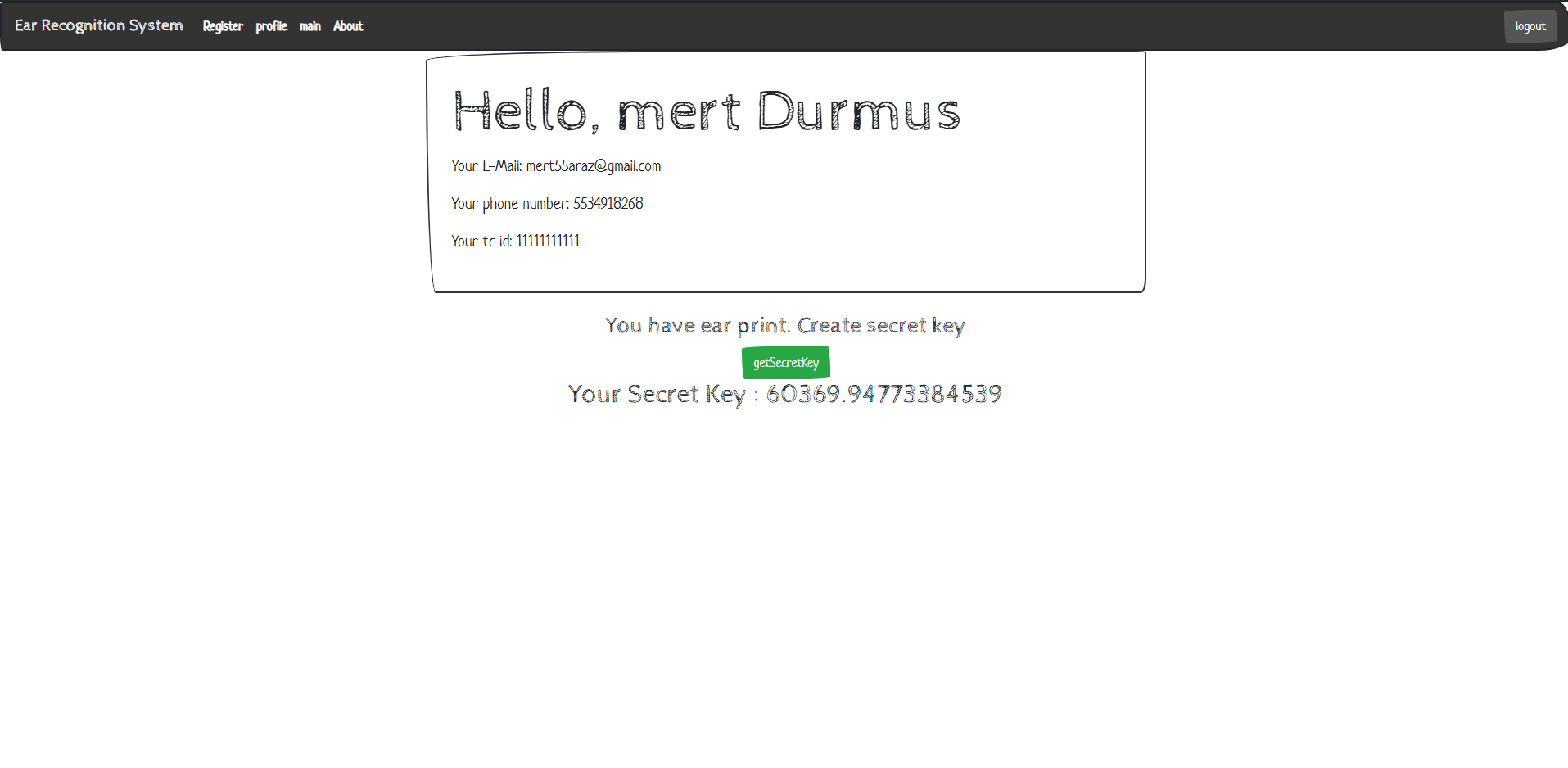


Şekil 5.4.4 Sistem ekran görüntüsü

Ekran görüntülerinde görüldüğü üzere kullanıcı sisteme kayıt olur. Sonrasında ister kameradan isterse fotoğraf yükleyerek sisteme kulak görüntüsü verir. Sistem bu kulak görüntüsünden kulak izini çıkarır ve bu kulak izi vektörlerini kullanarak kullanıcının istediği online platformlarda kullanabileceği ikincil bir doğrulama metriği olacak gizli anahtarı üretir.



Şekil 5.4.5 Sistem ekran görüntüsü



Şekil 5.4.6 Sistem ekran görüntüsü

Şekil 5.4.6 da görüldüğü gibi eğer kullanıcı kulak izine sahipse kullanıcıya istediği platformlarda kullanması için ikincil anahtarı üretilir. Aynı anahtar kullanıcının kullanmak istediği platforma da gönderilir. Böylece kullanıcı kulak izi biyometrik verisi ile ikincil bir doğrulama katmanına sahip olmuş olur.

# Sonuç

Bu çalışma ile bitirme projesi raporlanmıştır. Kulak izi tanımak ve bu izi kullanmak için yapılmış işlemleri özetlemek gerekirse; kulak tanıma işlemleri için veri toplama, veri seti oluşturma ilk adım olarak gösterilebilir. Sonrasında projede önüme aydınlatacak literatür taraması ve yapılmış çalışmaların incelenmesi gelir. Bu aşamadan sonra kulak izi tanımak için kullanacağımız metriklerin bulunması ve içlerinden seçim yapılması da bir sonraki aşamayı tanımlar. Sonrasında metrikleri ve veri setlerini kullanarak kulak izi görüntülerinden cinsiyet belirleme işlemi yapılmıştır. Son olarak yazılımı ayaklandıracak yazılım araçlarının belirlenmesi ile projeye devam edilmiştir. Kulak izi kullanarak ikincil bir doğrulama sistemi web platformları için implemente edilmiştir.

Sonuç olarak bu çalışmada biyometrik veri amaçlı kulak izi tanıma sistemi geliştirilmiştir. Bu kulak izi verisi ile farklı platformlarda kullanıcıların kullanabileceği bir doğrulama katmanı oluşturulmuştur. Bu katman web uygulaması olarak hayata geçmiştir.

# Kaynaklar

**[1]** A. Pflug ve C. Busch. Ear biometrics: a survey of detection, feature extraction and recognition methods. Biometrics, IET, 1(2):114–129, 2012.

**[2]** A. Pflug, A. Winterstein, ve C. Busch. Ear detection in 3D profile images based on surface curvature. In Proceedings of IEEE International Conference on Intelligent Information Hiding and Multimedia Signal Processing, 2012.

**[3]** A. Pflug, A. Winterstein, and C. Busch. Robust localization of ears by feature level fusion and context information. In Proceedings of the International Conference on Biometrics (ICB), 2013

**[4]** A. Ross, and C. Busch. 2d ear classification based on unsupervised clustering. In In Proceedings of International Joint Conference on Biometrics (IJCB), 2014.

**[5]** Li Y. , Zihichun M. ,Zhengguang X. , international wokshop on biometric person authantication, “Using Ear Biometrics for Personal Recognition” (2020)

**[6]** Chen H. , Bir B. , [IEEE Transactions on Pattern Analysis and Machine Intelligence](https://ieeexplore.ieee.org/xpl/RecentIssue.jsp?punumber=34), “Human Ear Rcognition in 3d”(2007/2020)

**[7]** Hurley D. , Nixon M. Carter J. , Computer vision and image understanding “Force Filed Feature Extraction for Ear Biometrics”(2005/2020)

**[8]** Internet, opencv api reference: [https://docs.opencv.org/3.0-beta/modules/refman.html (2019](https://docs.opencv.org/3.0-beta/modules/refman.html%20(2019) tarihnde erişildi.)

**[9]** İnternet: <https://scikit-learn.org/stable/getting_started.html> , (2020 tarihinde erişildi.)

**[10]** İnternet: <https://keras.io/>, (2020 tarihinde erişildi.)

**[11]** İnternet: <https://docs.anaconda.com/>, (2019 tarihinde erişildi.)

**[12]** Ami Ear Database, Esther Gonzalez,Luis Alvarez and Luis Mazorra internet: <http://ctim.ulpgc.es/research_works/ami_ear_database/> (2019)

**[13]** Ajay Kumar and Chenye Wu, “Automated human identification using ear imaging,” *Pattern Recognition*, vol. 41, no. 5, March  2012.

**[14]** IIT DELHI EAR DATABASE, internet: <https://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Ear.htm> (2019)

**[15]** Ho Chi Minh City Open University, EarVN1.0: A new large-scale ear images dataset in the wild. [vinh.th@ou.edu.vn](mailto:vinh.th@ou.edu.vn) (2019)

**[16]** İnternet: <https://en.wikipedia.org/wiki/Java_(programming_language)>, (2020 tarihinde erişildi.)