ANDROID TABANLI MOBİL CİHAZLARDA GÖRME KUSURLARININ TESPİTİ

2020

BİLGİSAYAR MÜHENDİSLİĞİ

BİTİRME PROJESİ TEZİ

Nesim YILDIRIM

Sümeyra IŞIK

ANDROID TABANLI MOBİL CİHAZLARDA GÖRME KUSURLARININ TESPİTİ

Nesim YILDIRIM

Sümeyra IŞIK

Karabük Üniversitesi

Mühendislik Fakültesi

Bilgisayar Mühendisliği Bölümünde

Bitirme Projesi Tezi

Olarak Hazırlanmıştır.

KARABÜK

2020

Sümeyra IŞIK, Nesim YILDIRIM tarafından hazırlanan “ANDROID TABANLI MOBİL CİHAZLARDA GÖRME KUSURLARININ TESPİTİ“ başlıklı bu projenin Bitirme Projesi Tezi olarak uygun olduğunu onaylarım.

Dr.Öğr.Üyesi Yasin ORTAKCI .....

Bitime Projesi Danışmanı, Bilgisayar Mühendisliği Anabilim Dalı

...../…../2020

Bilgisayar Mühendisliği bölümü, bu tez ile Bitirme Projesi Tezini onamıştır

Dr.Öğr.Üyesi. Hakan KUTUCU .....

Bölüm Başkanı

"Bu projedeki tüm bilgilerin akademik kurallara ve etik ilkelere uygun olarak elde edildiğini ve sunulduğunu; ayrıca bu kuralların ve ilkelerin gerektirdiği şekilde, bu çalışmadan kaynaklanmayan bütün atıfları yaptığımı beyan ederim."

Sümeyra IŞIK

Nesim YILDIRIM

ÖZET

Bitime Projesi Tezi

ANDROID TABANLI MOBİL CİHAZLARDA GÖRME KUSURLARININ TESPİTİ

Sümeyra IŞIK

Nesim YILDIRIM

Karabük Üniversitesi

Bilgisayar Mühendisliği

Bilgisayar Mühendisliği Bölümü

Tez Danışmanı:

Dr.Öğr.Üyesi Yasin ORTAKCI

2020

Göz rahatsızlıklarının meydana gelmesi sonucunda baş ağrısı ve buna benzer problemler ortaya çıkmaktadır. Bu problemlerin sebebinin gözlerde meydana gelen rahatsızlıklardan dolayı oluşup oluşmadığını tespit etmek için göz testleri yapılmıştır. Uygulamada kullanılan göz testi Snellen Testi'dir. Bu test bir kişinin görme keskinliği değerinin, göz muayenesi esnasında "standart görme keskinliği" denen bir değerle kıyaslanarak ortaya çıkarılması sonucu hesaplanır. Bu problemin önüne geçmek adına Python dili, OpenCV kütüphanesi ve Dlib kütüphanesi kullanılarak uygulama geliştirildi. İstenen hedefe ulaşabilmek için Yönlü Gradyan Histogramı (HOG) algoritması içinde bulunan Face Landmarks algoritması uygulandı. Uygulamada bu testi gerçekleştirmek için 7 harf dizisi kullanıldı. Bu test 27 karakterden oluşmaktadır. Bu karakterlerin bulundukları satıra göre boyutları değişmektedir. Test sonucunda kullanıcın bildiği doğru harf oranına ve bulunduğu satıra göre sonuç elde edildi.

Anahtar Sözcükler : Snellen Göz Testi, Face Landmarks Algoritması, OpenCV, Dlib

TEŞEKKÜR

Bu tez çalışmasının planlanmasında, araştırılmasında, yürütülmesinde, oluşumunda ilgi ve desteğini esirgemeyen, engin bilgi ve tecrübelerinden yararlandığımız, yönlendirme ve bilgilendirmeleriyle çalışmamızı bilimsel temeller ışığında şekillendiren sayın hocamız Dr.Öğr.Üyesi Yasin ORTAKCI'ya sonsuz teşekkürlerimizi sunarız.

İÇİNDEKİLER

ŞEKİLLER DİZİNİ

TABLOLAR DİZİNİ

BÖLÜM 1

GİRİŞ

1.1. LİTERATÜR ÖZETİ

Tezin ilk aşaması olan yüzün algılanması ve yüzdeki verilerin elde edilmesi için Yönlü Gradyan Histogramı (HOG) algoritması içinde bulunan Face Landmarks algoritması kullanılmıştır. HOG algoritmasının en büyük avantajı hızlı olması ve hata oranının en az miktarda olmasıdır. Face Landmarks algoritmasında ise her yüzde özel olan 68 özel bölge, nokta ile işaretleniyor ve bu bölgelerle yüzde istenen veriler elde ediliyor.

Göz testlerini gerçekleştirmek için göz ve kamera arasındaki mesafe Face Landmarks algoritmasından elde ettiğimiz koordinat değerleri Python dili kullanılarak tespit edilmektedir. Genel olarak kullanıcı telefondan fotoğraf çekilecektir fotoğraf imageview içinde tutulacaktır. Bu fotoğraf PHP Server' a gönderilir ve Python kodları çalıştırılır. Python 'dan gelen değerler tekrar Android Studio' ya gönderilir. Bu değerler Snellen Testini uygulamak için kullanılır.

Uygulamada bu testi gerçekleştirmek için 7 harf dizisi kullanıldı. Bu test 27 tane 'E' harfinden oluşmaktadır. Bu karakterlerin bulundukları satıra göre boyutları değişmektedir. Test sonucunda kullanıcın bildiği doğru harf oranına ve bulunduğu satıra göre sonuç elde edildi.

1.2. PROJENİN AMACI

Snellen testi, görme keskinliğini ölçmek için kullanılan bir göz şemasına denir. Bu testin diğer adı, LogMAR çizelgesi olarak bilinmektedir. Snellen testi genellikle göz bozukluğu olan hastaların, yakın, uzak gibi ne kadar mesafeden net görebildiklerini anlayabilmek için teste tabi olurlar. Gözünde sorun var mı yok mu onu öğrenirler. Bu test ile bir kişinin görme keskinliği değeri, göz muayenesi esnasında standart görme keskinliğini denen o değerle kıyaslanır. Bu kişinin görüp görmediği anlaşılır.

Bu test genellikle 6 metre mesafeden uygulanmaktadır bizim bu uygulamadaki amacımız göz ile kamera arasındaki mesafeyi daha kısa tutarak, testi uygulamayı mümkün kılmaktadır.

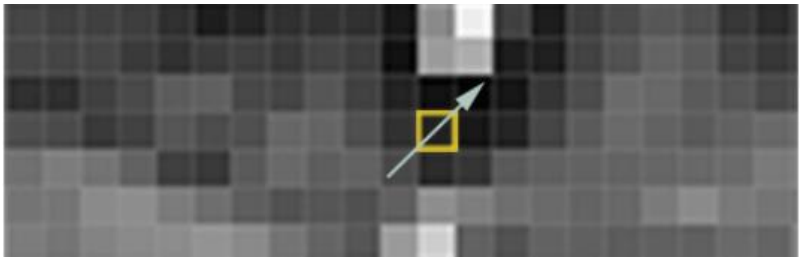
BÖLÜM 2

YÜZ ALGILANMASI

2.1. YÜZ ALGILAMA ALGORİTMASI:

Son yıllarda imgedeki piksellerin yönelim (θ) ve büyüklük değerlerinin karakteristiği olarak da adlandırılabilecek olan HOG algoritmasının kullanımı birçok alanda oldukça ilgi görmektedir. HOG kullanımı ilk defa Shashua ve Dalal tarafından önerilmiştir. Bir çok araştırmacı tarafından oldukça ilgi gören HOG yöntemindeki temel amaç imgeyi bir grup lokal histogramlar olarak tanımlamaktır. Bu gruplar, imgenin lokal bir bölgesindeki gradyanların yönelimlerinde, gradyanların büyüklüklerinin toplandığı histogramlardır. Bir imgenin HOG değerlerinin çıkarılması için gerekli olan aşamalar şu şekilde gerçeklenmektedir. İlk olarak imgenin, yatay ve dikey Sobel filtreleri uygulanarak, Ix ve Iy olmak üzere kenarları belirlenir. Sobel filtresi uygulanmış Ix ve Iy imgeleri kullanılarak, gradyan ve bu gradyanların yönelim açıları hesaplanır. Uyguladığımız HOG tanımlayıcısında lokal histogram bölgeleri tanımlanmamıştır. Bunun yerine bütün bir imge tek bir bölge olarak işleme alınmıştır[1].

Bizim uygulamamızda yüzleri bulmak için RGB (Kırmızı Yeşil Mavi-Red Green Blue) imge gri seviyeli imgeye dönüştürülür ve gri seviyeli imgenin yatay ve dikey gradyan değerleri elde edilir. Ardından, resimdeki her piksele birer birer bakılır. Her piksel için, onu doğrudan çevreleyen piksellere bakılır. Amacımız, mevcut pikselin onu çevreleyen piksellere oranla ne kadar karanlık olduğunu bulmaktır. Sonra şekil 2.1.'de gösterildiği gibi görüntünün hangi yöne koyulaştığını gösteren bir ok çizilir. Bu işlemi görüntüdeki her bir piksel için tekrarlanır, her pikselin yerini bir ok alınır. Bu oklara gradyanlar denir ve tüm görüntü boyunca ışıktan karanlığa akışı gösterir.

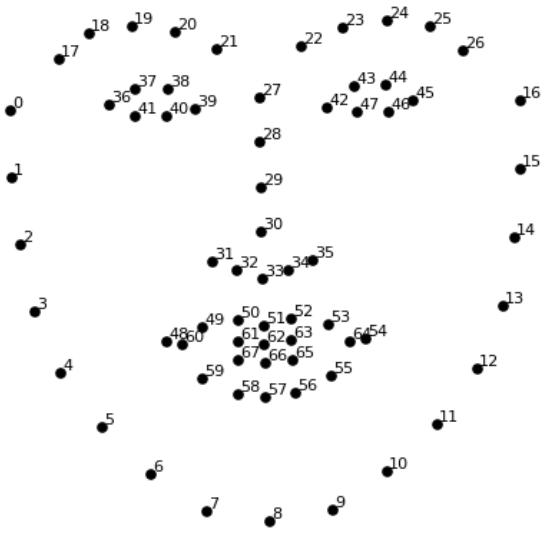


Şekil 2.1. Parlaklığın değiştiği yön.

Parlaklığın değiştiği yönü göz önünde bulundurarak, hem gerçekten karanlık görüntüler hem de gerçekten parlak görüntüler aynı şekilde temsil edilir. Görüntü her biri 16x16 piksel boyutunda küçük karelere bölünür. Her bir karede, her ana yönde kaç degradenin işaret ettiğini sayılır (kaç tane yukarı, yukarı, sağa, sağa vb.). Ardından görüntüdeki kareyi en güçlü ok yönleriyle değiştirilir. Sonuç olarak, orijinal görüntüyü bir yüzün temel yapısını basit bir şekilde yakalayan basit bir gösterime dönüştürülmüş olur. Bu şekilde yüzler bulunmuş olunur[2]. Yüz işaretlerini tahmin ederek yüz algılanmasını sağlayan Face Landmark Estimation algoritması kullanıldı.

2.1.1. FACE LANDMARKS ALGORİTMASI:

Yüz algılamanın birçok yolu bulunmaktadır, ancak tez projesinde Vahid Kazemi ve Josephine Sullivn tarafından icat edilen yaklaşım kullanıldı[3]. Temel fikir, her yüzde var olan 68 özel nokta; çenenin üstü, her gözün dış kenarı, her kaşın iç kenarı, vb. bölgeler belirlenir.



Şekil 2.2. Her yüzde bulunan 68 özel nokta

Uygulamamızda fotoğrafta ilk olarak Dlib kütüphanesi kullanılarak yüz algılanıyor ve Face Landmarks Algoritması kullanılarak algılanan, yüzün 68 özel noktası belirlenir. Bu 68 özel nokta iki göz arasındaki mesafe tespit edildi.

2.2. MESAFENİN BELİRLENMESİ:

Şekil 2.2.'de gösterilen noktalara göre iki göz arasındaki mesafe yani (39, 42) noktalar arasındaki noktaların her birinin konumu aşağıdaki formül ile hesaplandı.

$$A = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad (1)$$

İki göz arasındaki mesafeyi seçmemizin sebebi bu mesafe değerinin her yüz için sabit olmasıdır.

Genel olarak kamera kullanıcıya yaklaştırıldığında bu mesafe daha büyük çıkmaktadır. Uzaklaştırıldığında ise daha küçük çıkmaktadır.

2.aşamada A denkleminden elde edilen sonuç referans alınarak göz ile kamera arasındaki mesafe aşağıdaki adımlarla hesaplanır.

1. Kullanıcı ile kamera arasındaki mesafe 55 cm olacak şekilde çekilen fotoğraf ile kullanıcının iki gözü arasındaki mesafe(**Reference)** hesaplandı.
2. Kullanıcının test yapmak istediği mesafeye göre 2 göz arasındaki mesafe hesaplanır.
3. Elde edilen sonuçlar esas alınarak yüz ile kamera arasındaki mesafe aşağıdaki yöntem ile hesaplandı.

$$\mathbf{CameraToUserDistance} = \frac{(\text{distance\_between\_two\_eye} * \mathbf{Reference})}{\text{New\_ distance\_between\_two\_eye}} \quad (2)$$

4. Bulunan değere göre test gerçekleştirilecektir.

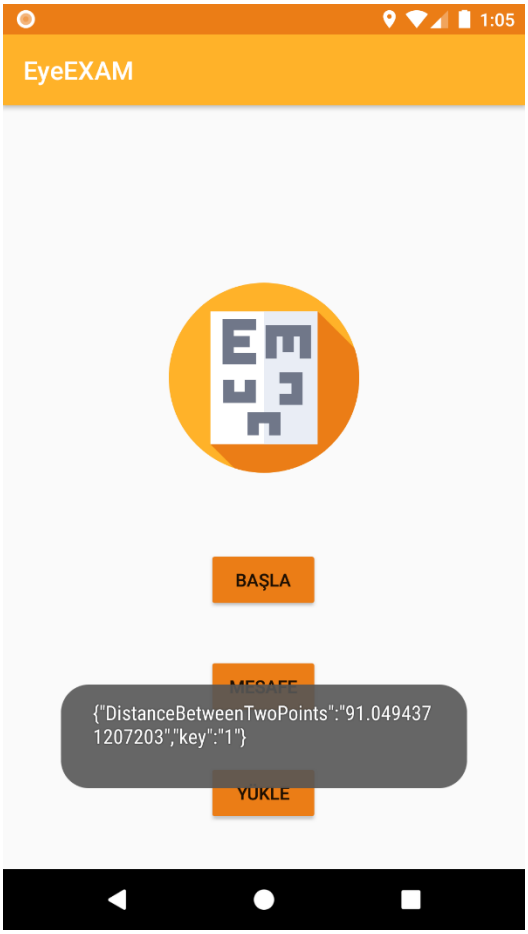
BÖLÜM 3

SERVER BAĞLANTISI

3.1. CLIENT OLARAK ANDROID KISMININ UYGULANMASI:

Yapılan uygulama mobil bir uygulama olduğu için Android Studio ile geliştirildi. Hesaplanacak mesafeler Android Studio ile hesaplanması yavaş olduğu için kodlar Python' da yazıldı. İşlemleri birbirinden ayırt edebilmek için anahtar değişkeni "key" belirledik. Android ' de kullanıcı ilk olarak başla butonuna tıkladığında galeriden 55 cm uzaklıktan çekilen fotoğraf seçilir ardından iki göz arasındaki mesafeyi hesaplamak için bu fotoğraf, key değeri 1 olarak Yükle butonuna tıklayarak Server'a Volley Kütüphanesi kullanılarak gönderilir. Server' dan gelen değer, telefon hafızasında SharedPreferences kullanılarak **"distance_between_two_points"** anahtarında tutulur.

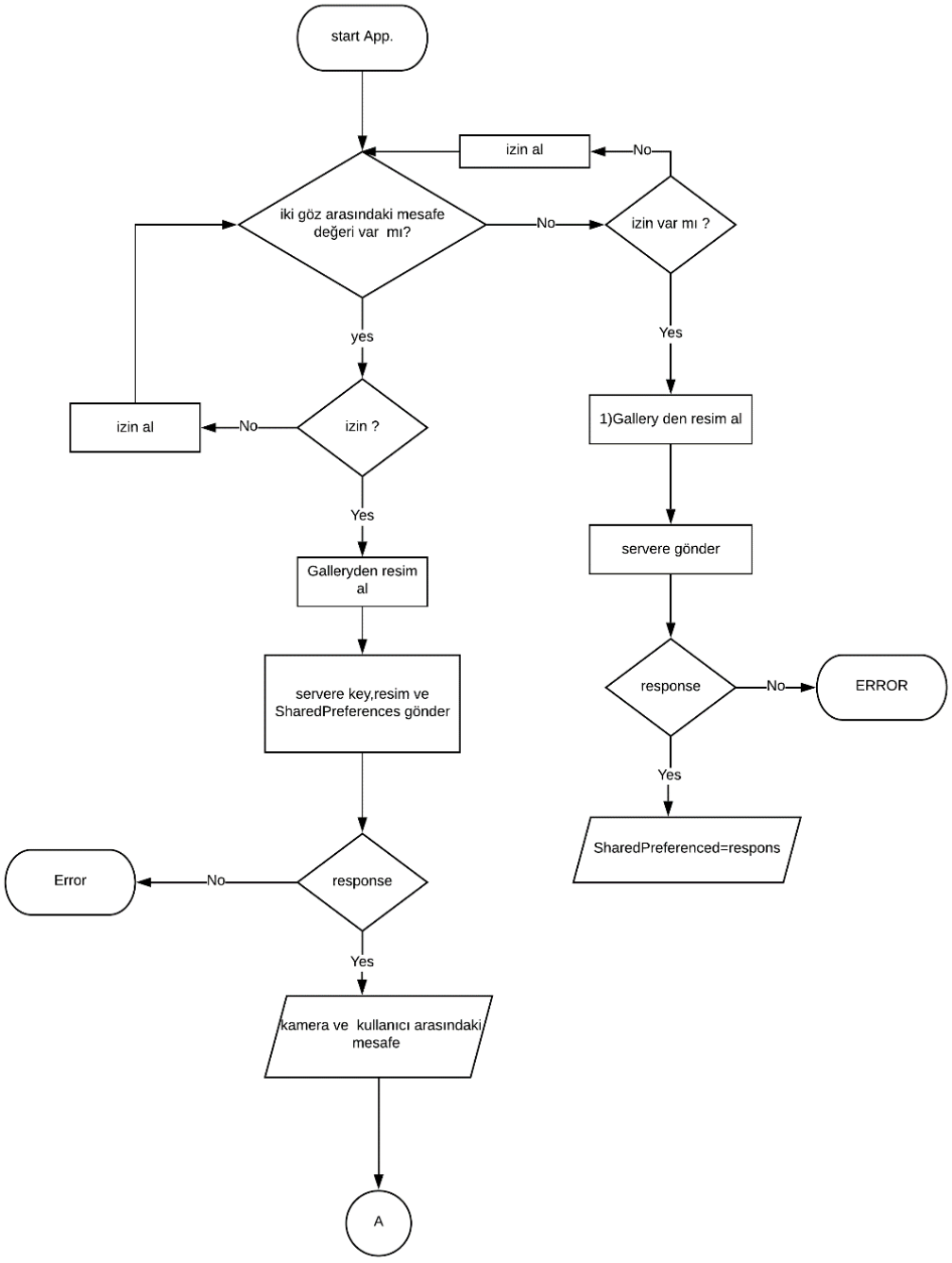
Server'dan gelen değer Şekil 3.1.'deki gibi test etmek amacıyla Toast mesajı kullanılarak iki göz arasındaki mesafe pixel cinsinden ekranda gösterilmektedir. Bu mesaj daha sonrasında ekrandan kaldırıldı. Mesafe butonuna tıklandığında ise galeriden test yapmak istenen uzaklıktan çekilen fotoğraf seçilir ayrıca kullanıcı bu süre zarfında aynı uzaklıkta sabit olarak kalmalıdır ardından key değeri 2 olarak **"distance_between_two_points" '**deki değerle resim server'a tekrardan gönderilir.



Şekil 3.1. Response değeri.

Server' dan gelen değerle kullanıcı ile kamera arasındaki mesafe belirlenir ve bu mesafeye göre yapılan testteki karakterlerin boyutları hesaplanır.

Android kısmındaki çalışma prensipleri aşağıdaki Şekil 3.2.’ te gösterildiği gibidir:

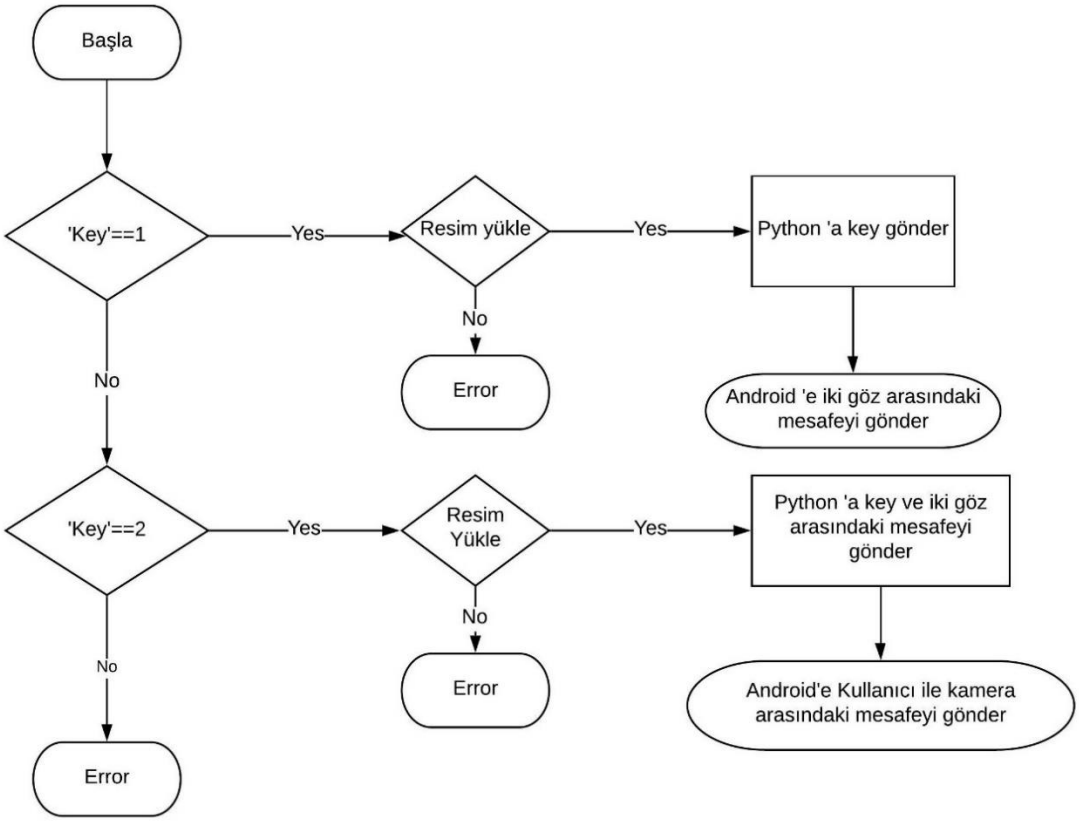


Şekil 3.2. Server’a fotoğraf gönderme

3.2. SERVER OLARAK PHP KISMININ UYGULANMASI:

Android’ den gelen key değeri 1 olduğunda resim kaydedilecektir. Daha sonra Python’ a key değeri gönderilecektir. Bölüm 2.2. ‘de anlatıldığı gibi (1) nolu formül kullanılarak iki göz arasındaki mesafe hesaplanır ve hesaplanan değer Android ‘ e gönderilir.

Android ‘ den gelen key değeri 2 ise resim kaydedilecektir. Daha sonra Python ‘a key ve iki göz arasındaki mesafe gönderilir. Python ‘dan gelen mesafe (2) nolu formül kullanılarak hesaplanıp tekrardan Android’ e gönderilir. Programın bu kısmının işleyişi Şekil 3.2.‘teki gibidir:

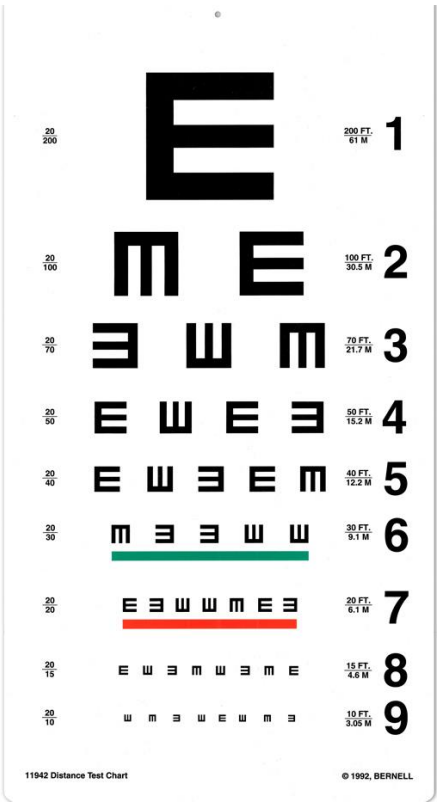


Şekil 3.3. Server’ın işleyişi

BÖLÜM 4

SNELLEN TESTİ

Görme keskinliği (GK) görsel olarak küçük farklılıkları ayırt edebilme yeteneğine denir. Oftalmolojik muayene esnasında eşik değerin belirlenmesi (ölçümü) göz hekimlerinin klinik kararlarını vermelerinde etkin bir rol oynar. Bir kişinin görme keskinliği değeri, göz muayenesi esnasında "standart görme keskinliği" denen bir değerle kıyaslanarak ortaya çıkarılır. Snellen testinde standart (normal) GK değeri, metre cinsinden 6/6 (ya da uzunluk ölçü birimi olarak fit kullanan ülkelerde 20/20) olarak kabul edilmiştir. Bunu biraz daha açıklayacak olursak, ayırma çizgisinin (/) sağ tarafında kalan (ikinci) rakam, standart olan görüş uzaklığını, yani neredeyse herkesin rahatlıkla okuyabildiği mesafeyi; sol tarafında kalan (ilk) rakam ise sizin o harfleri rahatça okuyabildiğiniz mesafeyi gösterir[4]. Uygulamada Şekil 4.1.'de gösterilen test uygulandı.



Şekil 4.1. Görme Keskinliği Ölçümü

4.1. GÖRME KESKİNLİĞİ TESTİ UYGULANMASI:

Snellen fraksiyonu metrik veya feet olarak belirtilebilir. Görme keskinliği testi neredeyse her zaman 6 metre (20 ft) mesafeden yapıldığından, Snellen fraksiyonunun payı, metrik veya feet değerinin kullanılmasına bağlı olarak neredeyse her zaman 6'dır (veya 20). Paydada ise bizim testi uygulayacağımız değer bulunur. Bu dönüşümü uygulamak için (3) nolu formül kullanıldı[5]. Bu formül ile görme keskinliği sonucu elde edilir. Örneğin Görme Keskinliği değeri 6/30 ise bu şu anlama gelmektedir; normal insanların 30 metre mesafeden gördüğü en küçük karakteri biz 6 metre mesafeden okuyabildik.

$$\text{Görme Keskinliği} = \frac{Test\ Yapılacak\ Mesafe}{Normal\ İnsanların\ Görebileceği\ En\ Küçük\ karakterin\ Mesafesi} \quad (3)$$

Şekil 4.1.'de gösterilen test daha küçük mesafeden yapılacağı için ilk 7 harf dizisi kullanıldı. Bu test 27 karakterden oluşmaktadır. Buna uygun farklı görme keskinliği notasyonları dönüşümü Tablo 4.1. 'deki gibidir, International Council of Ophthalmology tarafından yazılan Görme Keskinliği Ölçüm Standardı tablosundan yararlanıldı[6].

Feet	Metre	LogMAR
20/200	6/60	+1.0
20/100	6/30	+0.7
20/70	6/21	+0.544
20/50	6/15	+0.4
20/40	6/12	+0.3
20/30	6/9	+0.176
20/20	6/6	0.0

Tablo 4.1. Görme keskinliği notasyonları dönüşümü.

Tablo 4.2. ‘de herbir satır bir harf dizisini temsil etmektedir. Her dizide kaç tane harf bulunduğu Tablo 4.2. ‘de gösterilmiştir.

Feet	Harf Sayısı	Index(i)
20/200	1	0
20/100	2	1
20/70	3	2
20/50	4	3
20/40	5	4
20/30	5	5
20/20	7	6

Tablo 4.2. Görme keskinliği notasyonlarına denk gelen harf sayısı.

4.2. KARAKTER YÜKSEKLİĞİ HESAPLANMASI:

Snellen Görme Testi’nde herbir karakter 6 metre uzaklıktan uygulanmaktadır. Yani Tablo 4.1. ‘de gösterildiği gibi hazır olarak hesaplanmış değerler bulunmaktadır. Uygulamada kamera ile kullanıcı arasındaki mesafe yani test mesafesi değişken olduğu için her harfin boyutu tekrardan hesaplanmaktadır.

Kamera ile kullanıcı arasındaki mesafe artık Test Mesafesi olarak kabul edilecektir.

Server’dan gelen bu Test Mesafesi cm cinsinden gelmektedir. İlk olarak bu mesafe metreye dönüştürülür. Daha sonrasında bu metre değeri (4) nolu formül kullanılarak feet’e dönüştürülür.

$$Feet\_Değeri = \frac{Test\ Mesafesi\ (metre) * 20}{6} \qquad (4)$$

Tablo 4.2.'de gösterildiği gibi herbir satıra karşılık gelen, kontrol edilecek mesafenin değeri bulunur. Yani birinci satır 200 Feet (60 Metre) mesafeden kontrol ediliyor ama uygulamada test mesafesi farklı olduğu için bu mesafeye karşılık gelen değer, (5) nolu formül kullanılarak bulunur. Paydadaki değerleri bir dizide aşağıdaki gibi tanımlanır.

tablo_paydadaki_Degerleri = {200, 100, 70, 50, 40, 30, 20},

i = Tablo 4.2.'deki index değeridir.

$$Kontrol\ Edilecek\ Mesafe = \frac{FeetDeğeri\ *\ tablo\_paydadaki\_Degerleri[i]}{20} \quad (5)$$

Herbir harf 5x5 boyutlu matris üzerinde yapılmaktadır. Normal insanların görebileceği en küçük karakterin mesafesi, ''5 minute of arc'' tır. (1 arcminute = $1/60$) [5]. Türkiye de arc minute 'e arc dakika da denmektedir.

Kontrol edilecek mesafe değerine karakter boyutu ayarlanacaktır. Aşağıdaki (6) no'lu formülde mesafe değerini 0.65 metre olarak kabul ettik.

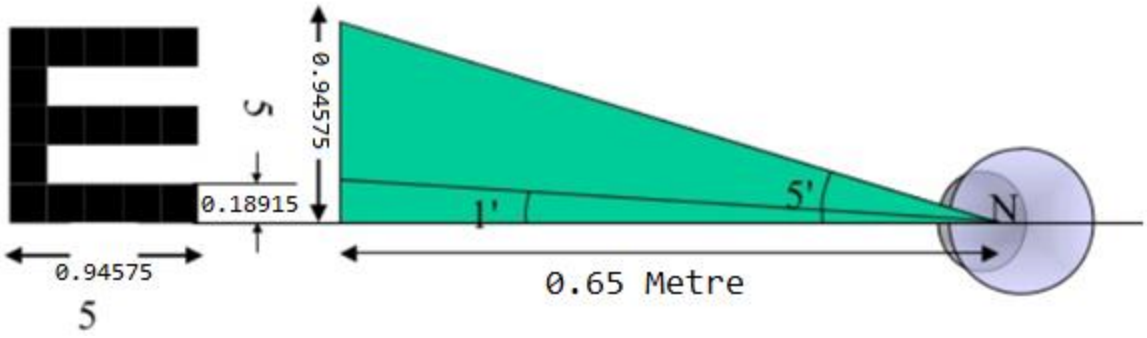
$$\tan\theta = \frac{karakter\ yüksekliği}{Kontrol\ Edilecek\ Mesafe} \quad (6)$$

En küçük karakterin yüksekliği(7) no'lu örnekteki gibi hesaplandı ve bulunan değerin harf üzerindeki etkisi Şekil 4.2. 'deki gibidir;

$$\tan\left(\frac{1}{60}\right) = \frac{1\ arcminute\ yüksekliği}{0.65\ Metre} \quad (7)$$

1 arcminute yüksekliği = 0.18915 mm

Karakter yüksekliği = 0.18915*5 = 0.94575 mm



Şekil 4.2. En Küçük Karakterin Yüksekliği(örnek).

BÖLÜM 5

GÖRME TESTİNİN SONUÇLARI VE DEĞERLENDİRMESİ

Her karakter için Görme Keskinliği ve Karakter Yüksekliği formülleri teker teker uygulanır.

Bizim uygulamamızda kullanıcının tam doğru okuyabildiği en son satırın görme keskinliği değeri sonuç olarak ekranda gösterilir. Herbir satırdaki harflerin ağırlık değeri hesaplandı. Kullanıcı her doğru yaptığı karakter test sonucuna eklenecektir ve görme keskinliği oranı ekranda gösterilir. Test sonucunda doğru bilme durumuna göre aşağıdaki tablodaki değerleriyle eşleşen durum ekranda görünecektir.

Sonuç değerlendirmesi Tablo 5.1. 'deki Görsel Okuma Becerisi ve Doktora gidilmeli mi? sütunlarında görüldüğü üzere A. Colenbrander kaynağından alınmıştır [7].

Görme Keskinliği	Feet	Sonuç	Görsel Okuma Becerisi	Doktora gidilmeli mi?
6/60	20/200	Büyük görme kaybı	10 cm'den küçük okumalar, İki gözle birlikte görme.	Evet
6/30 6/21	20/100 20/70	Orta görme kaybı	25... 12,5 cm'de okumak için güçlü okuma gözlükleri veya orta güçte büyüteçler gerekir.	Evet
6/15 6/12 6/9	20/50 20/40 20/30	Hafif görme kaybı	Kısa süreli normal okuma mesafesi.	Belirtiler varsa doktora gidebilir
6/6	20/20	İyi görüş	Normal okuma mesafesi.	Hayır

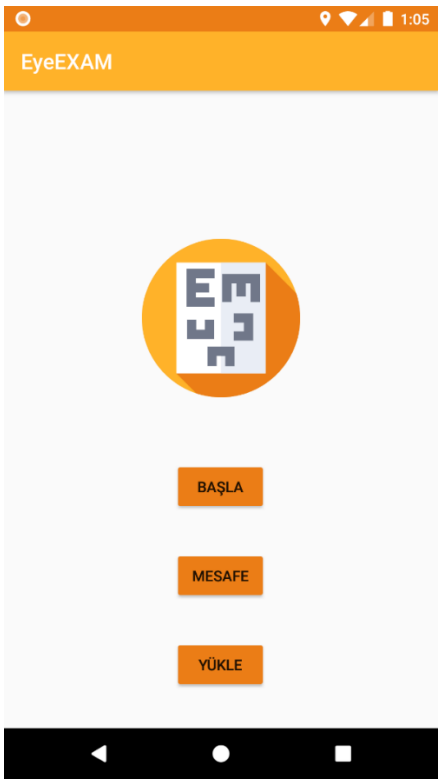
Tablo 5.1. Görme Keskinliği Aralıkları - Grafik Aralıkları - Okuma Yeteneği Aralıkları.

BÖLÜM 6

UYGULAMANIN ÇALIŞMA PRENSİBİ

6.1. UYGULAMANIN ANASAYFASI:

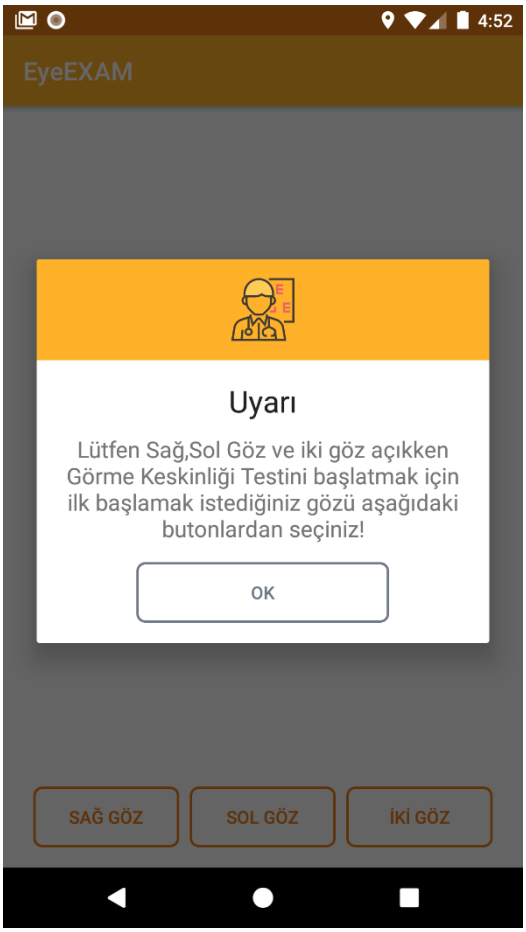
Uygulamanın anasayfasında 3 adet buton bulunmaktadır. Android ‘ de kullanıcı ilk olarak başla butonuna tıkladığında galeriden 55 cm uzaklıktan çekilen fotoğraf seçilir ardından iki göz arasındaki mesafeyi hesaplamak için bu fotoğraf, Yükle butonuna tıklayarak Server’a Volley Kütüphanesi kullanılarak gönderilir. Server’ dan gelen değer, telefon hafızasında SharedPreferences kullanılarak **"distance_between_two_points"** anahtarında tutulur. Mesafe butonuna tıklandığında ise galeriden test yapmak istenen uzaklıktan çekilen fotoğraf seçilir ayrıca kullanıcı bu süre zarfında aynı uzaklıkta sabit olarak kalmalıdır ardından **"distance_between_two_points"** ‘deki değerle resim server’a tekrardan gönderilir. Server’dan gelen değer test mesafesi olur. Uygulamanın bu aşamadaki arayüzü Şekil 6.1.’deki gibidir;



Şekil 6.1. Uygulamanın Anasayfası.

6.2. KULLANICIYA BİLGİLENDİRME MESAJI GÖSTERİLMESİ:

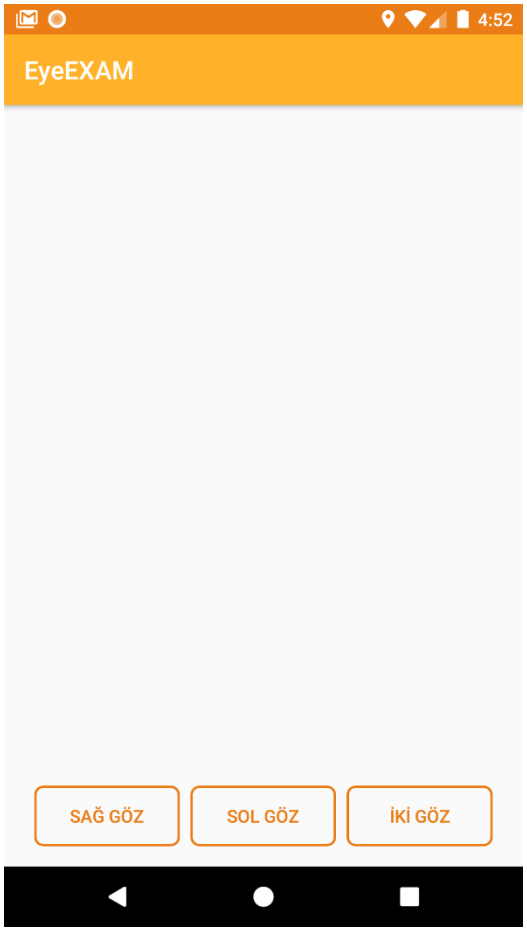
Server'dan test mesafesi gelince kullanıcıya testin uygulama aşamalarını anlatan, kullanıcıyı yönlendiren bir mesaj ekranda görünecektir. Dialog tasarımında OK butonu için arka planda ayrı bir drawable resource xml dosyası oluşturuldu. Layout klasöründe yeni bir my_dialog.xml oluşturuldu. Bu xml'de dialogun tasarımı yapıldı. Ana activity 'de bu dialog çağırıldı ve Şekil 6.2. 'deki gibi kullanıcıya gösterildi.



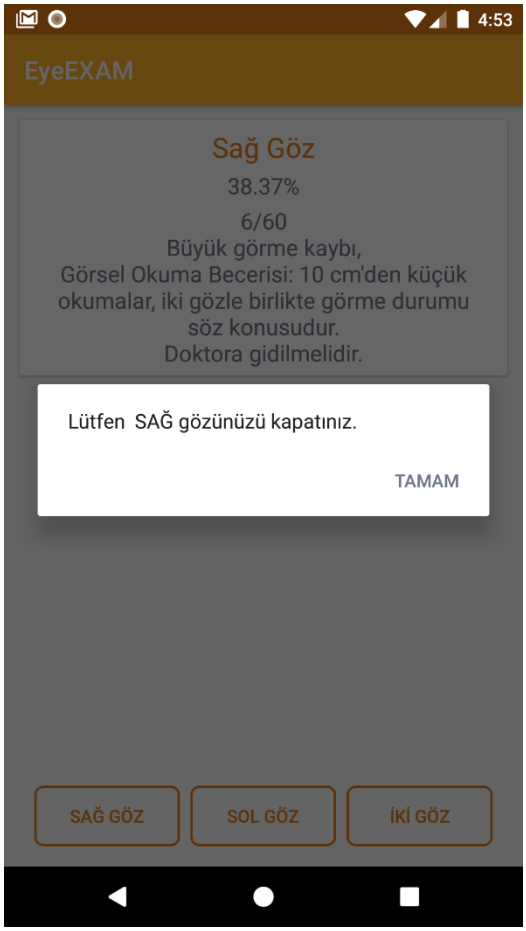
Şekil 6.2. Android Custom Dialog.

6.3. KULLANICININ TESTİ BAŞLATMASI VE SONUÇLARI GÖRMESİ:

Şekil 6.3.'de kullanıcıya sağ göz ,sol göz ve iki göz için ayrı ayrı testi uygulama seçenekleri sunulmaktadır. Şekil 6.4.'te kullanıcı testi sol gözü için uygulamak istediğinde ekrana 'Lütfen SAĞ gözünüzü kapatınız.' tarzında bir uyarı gelecektir.



Şekil 6.3. ControlActivity Sayfası.



Şekil 6.4. AlertDialog Mesajı.

6.3.1. KULLANICIYA TESTİN SONUCUNUN GÖSTERİLMESİ:

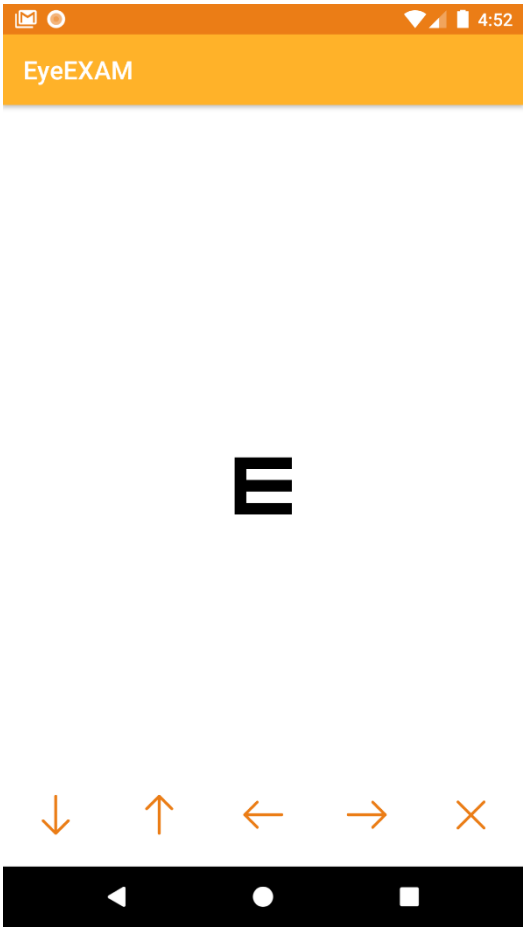
Kullanıcı testi her uyguladıktan sonra Şekil 6.5. ‘deki gibi CardView kullanarak testin değerlendirmesi kullanıcıya gösterilir. Şekil 6.5.’i elde edebilmek için aşağıdaki adımlar uygulandı:

1. İlk olarak gerekli olan kütüphaneler eklendi.
2. recycler_item.xml dosyası oluşturuldu. Bu dosyada CardView oluşturuldu bu CardView içinde title, result, comment TextView’ları oluşturuldu.
3. Java kısmında ise RecyclerItem.java ve MyAdapter.java class’ı ve bunun içinde ViewHolder alt class’ı oluşturuldu.
4. ControlActivity ‘ye ait xml dosyasına recyclerView eklendi. ControlActivity ‘de ise writeInList fonksiyonu oluşturuldu ve gelen test sonuçları bir listede tutuldu ve ekranda gösterildi.

Şekil 6.5. RecyclerView ile sonucun gösterilmesi.

6.4. KULLANICININ TESTİ UYGULAMASI:

BÖLÜM 4'te anlatılan harflerin boyutlarının adapte edilmesi Şekil 6.6.'daki gibidir. Kullanıcı yön butonlarına basınca bulunduğu harfin yönünün doğru bilinip bilinmediği test sonucuna yansımaktadır. Her testte bulunan 27 karakter sonlanınca test sonucu ControlActivity'ye result olarak gönderilmektedir.



Şekil 6.6. TestActivity sayfası.

BÖLÜM 7

SONUÇ

Birçok hastalığın belirtisi birbirine benzemektedir. Bu belirtilerin göz kusurlarından mı meydana gelip gelmediğini tespit edebilmek için bu uygulama geliştirilmiştir.

Projede göz testlerini mobil ortamda, insanlara yarar sağlayacak şekilde gerçekleştirmek için bu çalışmada bahsedildiği gibi yüz algılaması daha sonrasında kamera ve yüz arasındaki mesafe hesaplandı. Gerekli formüller kullanılarak Snellen Göz Testlerinin yazı boyutu, aradaki mesafeye göre uygulandı. Böylelikle her kişi bu testleri mobil uygulama aracılığı ile kolaylıkla istediği zaman uygulayabilir.

KAYNAKÇA:

[1] M. Peker, H. Altun, F. Karakaya, E. Elektronik Mühendisliği, and N. Üniversitesi, "HOG Temelli Bir Yöntem ile Ölçek ve Yönden Bağımsız Gerçek Zamanlı Nesne Tanıma."

[2] A. Geitgey, "Machine Learning is Fun! Part 4: Modern Face Recognition with Deep Learning." [Online]. Available: https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78#.3qm039gs2.

[3] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1867–1874.

[4] A. Şenyiğit, "Snellen Testi: Görme Keskinliği Ölçümü - Evrim Ağacı." [Online]. Available: https://evrimagaci.org/snellen-testi-gorme-keskinligi-olcumu-4901.

[5] Theodore P. Grosvenor, "Primary Care Optometry - Theodore P. Grosvenor - Google Kitaplar," *Elsevier Health Sciences*, 2007.

[6] G. B. Arden, "Visual acuity measurement standard," *J. Fr. Ophtalmol.*, vol. 11, no. 11, pp. 779–792, 1988.

[7] A. Colenbrander and M.-S. Francisco, "Colenbrander-Visual Acuity measurement tools. The COLENBRANDER Low Vision Measurement System."