TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

TAŞ KAĞIT MAKAS OYUNU

13011704 — Abdullah KAYAHAN
13011705 — Uhud POYRAZ

BİLGİSAYAR PROJESİ

Danışman
Doç. Dr. Songül ALBAYRAK

Ocak, 2017

TEŞEKKÜR

Bizim için mantığı basit ancak içeriği oldukça ağır olan bu proje bize birçok açıdan katkı sağladı. Araştırma, kriz yönetimi, planlama, iş bölümlesi ve raporlama gibi birçok yetimizi bu proje sayesinde geliştirdik.

Proje yapımı sırasında bizden desteğini esirgemeyen proje danışmanımız Doç. Dr. Songül ALBAYRAK'a, görüntü işleme konusundaki bilgilerini bizimle paylaşmaktan çekinmeyen hocamız Arş. Grv. Abdülkadir ALBAYRAK'a, raporlama konusunda bizden yardımını esirgemeyen arkadaşımız Osman AYDIN ve Muhammet Ali MİNTAŞ'a ve projemize destek veren bütün arkadaşlarımıza teşekkür ederiz.

Ortaya çıkan bu proje umarız bizden sonra birçok insana ilham verir.

Abdullah KAYAHAN
Uhud POYRAZ

İÇİNDEKİLER

KISALTMA LİSTESİ

FPS	Frame Per Second
TL	Türk Lirası
HOG	Histogram Of Oriented Gradients
K-NN	K-Nearest Neighbors
JVM	Java Virtual Machine

ŞEKİL LİSTESİ

TABLO LİSTESİ

ÖZET

TAŞ KAĞIT MAKAS OYUNU

Abdullah KAYAHAN
Uhud POYRAZ

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Doç. Dr. Songül ALBAYRAK

Video çerçevesi içerisinden belirli bir nesnenin görüntüsünün alınması ve sınıflandırılması işlemi bir çok alanda kullanılmaktadır. Görüntü işleme yöntemleri ile arkaplanın temizlenmesi ve nesnenin algılanması oldukça kolaylaştırılmış ve bir çok sisteme entegre edilmiştir. Özellikle endüstriyel ve askeri alanlarda kullanılan görüntü işleme yöntemleri, bu projede bir oyun geliştirmek için kullanılmaktadır. Bir nesnenin arkaplandan ayrılması, HOG değerlerinin çıkartılması ve sınıflandırılması işlemleri proje kapsamında kullanılan işlemler arasında yer almaktadır.

Anahtar Kelimeler: ten tanıma, el tanıma, openCv, yönlü histogram, makine öğrenmesi, eğitimli öğrenme, görüntü işleme,video işleme.

YILDIZ TEKNİK ÜNİVERSİTESİ
ELEKTRİK - ELEKTRONİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

ABSTRACT

ROCK PAPER SCISSORS GAME

Abdullah KAYAHAN
Uhud POYRAZ

Department of Computer Engineering
Computer Project

Advisor: Assoc. Prof. Dr. Songül ALBAYRAK

Receiving a specific object image in the video frame and classification process is used in many fields. With image processing methods, background cleaning and object detection are simplified and integrated into many systems. Image processing methods, especially used in industrial and military fields, are used to develop a game in this project. The background separation of an object, the extraction of HOG values, and the classification operations are among the processes used in the project.

Keywords: Skin detection, hand detection, openCv, histogram of gradient, machine learning, supervised learning, image processing, video processing.

YILDIZ TECHNICAL UNIVERSITY
FACULTY OF ELECTRICAL AND ELECTRONICS
DEPARTMENT OF COMPUTER ENGINEERING

1
Giriş

Günümüzde gelişen bilgisayar teknolojilerinde görüntü işleme önemli bir yer tutmaktadır. Yüz ve vücut tanımlama ve tanıma, hareket algılama, seri üretimlerde üretimin doğruluk kontrolü bunlardan sadece birkaç tanesini oluşturmaktadır. Oyun piyasası da bu gelişimden etkilenerek özel kamera sistemleri ve hareket sensörleriyle sanal bir gerçeklik yakalama çabası içindedir.

Bu projede amaç herkesin bildiği bir oyun olan taş-kağıt-makas oyununu, standart bilgisayar kamerası ile görüntü işleme algoritmaları kullanarak bilgisayara karşı oynamayı sağlayacak bir sistem geliştirmektir. Projenin temel amacı bir video çerçevesi içerisinde eli algılamak, yapılan üç temel hareketi tanımlamak ve oluşturulan veritabanından bu hareketleri eşleştirmektir.

Bu proje, bir oyun olarak bakıldığında çok fazla bir anlam ifade etmese de, birçok farklı projeden ilham alacak ve birçok farklı projeye ilham verecek yapıdadır.

2
Ön İnceleme

OpenCv, 1999 yılında İntel tarafından geliştirilmeye başlanmış açık kaynak kodlu görüntü işleme kütüphanesidir.[1] Açık kaynak olması bu kütüphanenin zamanla gelişmesini sağlamıştır. Başlangıçta sadece C dili için olan kütüphane, günümüzde birçok programlama diline hitap etmektedir. Bunun yanı sıra en çok kullanılan görüntü işleme kütüphanesi olması sebebiyle proje kapsamında yapılan araştırmalarda çeşitli kaynaklar ve örnek projeler bulma imkanını sağlamıştır. Ancak kullanılan programlama dilinde (Java) örnek sayısının oldukça az olduğu gözlemlenmiştir.

Proje kapsamında yapılan araştırmalar sonucunda, projenin yapımı sırasında en dikkat edilecek hususun, elin video çerçevesi içerinden çok temiz bir şekilde algılanması olduğu gözlemlenmiştir. Proje kapsamında incelenen benzer projelerde eli arka plandan çok rahat ayırdıkları görülmektedir.[2] Fakat bu projelerde genellikle arka plan tek bir renk olarak tasarlanmış. Ayrıca farklı ortam koşullarında test edilip edilmediği net olarak bilinmemektedir. Bu nedenle projenin değişen ortam koşullarına uygun olarak tasarlanması kararlaştırılmıştır.

Özetle, her türlü ortamda, elin görüntü işleme yöntemleriyle ayırt edilmesinde yüksek performans sağlayan örnek projeler bulunamamıştır. Ancak bulunan örnekler projeye oldukça katkı sağlamış faklı bakış açıları kazandırmıştır.

3
Fizibilite

Projenin fizibilitesi; teknik fizibilite, zaman fizibilitesi, ekonomik fizibilite ve yasal fizibilite başlıkları altında çıkarılmıştır.

3.1 Teknik Fizibilite

Bu proje, OpenCv kütüphanesi kullanarak, Java programlama diliyle Eclipse üzerinde kodlanacaktır. Veritabanı olarak PostgreSQL kullanılacaktır.

Geliştirmede Eclipse kullanılmasının sebebi, yaygın kullanımı ve bizlerin bu geliştirme ortamına daha hakim olmamızdır.

Geliştirmede kullanılan Java programlama dili, projenin gelişimi aşamasında mobil platforma (Android) kolayca entegre edilebilir şekilde olması amacıyla seçilmiştir. Ayrıca, proje kapsamında yapılan ön incelemelerde Java dilinde OpenCv örnekleri oldukça az olduğu gözlemlendiğinden, bu dil ile yapılan ve OpenCv kütüphanesini kullanan uygun bir örnek proje olması amacıyla Java dili seçilmiştir.

Veritabanı olarak PostgreSQL seçilmesinin sebebi ise bu veritabanı ortamına daha hakim olmamızdır.

3.2 Zaman Fizibilitesi

Projenin tamamlanma süresi, projenin onaylandığı tarihi takip eden haftadan itibaren 65 iş günü olarak belirlenmiştir. İş bölümleri ve süreleri Tablo 3.1 ve Şekil 3.1'den görülebilir.

Tablo 3.1 İş Zaman Çizelgesi

	Task Name	**Duration**	**Start**	**Finish**
1	Proje Gerçekleştirme Aşamaları	65 Days	03.10.2016	30.12.2016
2	Proje Gereksinimlerinin Belirlenmesi	2 Days	03.10.2016	04.10.2016
3	Örnek Projelerin İncelenmesi	5 Days	05.10.2016	11.10.2016
4	Sistem Analizi Ve Fizibilite	5 Days	12.10.2016	18.10.2016
5	Proje Modüllerinin Belirlenmesi	1 Day	19.10.2016	19.10.2016
6	Modüllerin Planlanması	2 Days	20.10.2016	21.10.2016
7	Milestone	0 Day	21.10.2016	21.10.2016
8	El Algılama Modülünün Tasarımı	17 Days	24.10.2016	15.11.2016
9	Video Görüntüsünün Alınması	1 Day	24.10.2016	24.10.2016
10	El Nesnesinin Algılanması	7 Day	25.10.2016	02.11.2016
11	Elin Çevresinin Çizdirilmesi	3 Days	03.11.2016	07.11.2016
12	Planlanan Modlülün Gerçeklenmesi	5 Days	8.11.2016	14.11.2016
13	Modülün Testi	1 Day	15.11.2016	15.11.2016
14	Milestone	0 Day	15.11.2016	15.11.2016
15	Eğitim Ve Test Modülü	17 Days	16.11.2016	08.12.2016
16	Eğitim Ve Test Verisi Toplanması	5 Days	16.11.2016	22.11.2016
17	Veri Analizi	2 Days	23.11.2016	24.11.2016
18	HOG Derğerlinin Çıkartılması	2 Day	25.11.2016	28.11.2016
19	Sınıflandırma	2 Day	29.11.2016	30.11.2016
20	Planlanan Modülün Kodlanması	5 Days	01.12.2016	07.12.2016
21	Modülün Testi	1 Day	08.12.2016	08.12.2016
22	Milestone	0 Days	08.12.2016	08.12.2016
23	Oyunun Gerçeklenmesi	15 Days	09.12.2016	29.12.2016
24	Tasarımın Kodlanması	5 Days	09.12.2016	15.12.2016
25	Oyunun Kodlanması	10 Days	16.12.2016	29.12.2016
26	Milestone	0 Day	29.12.2016	29.12.2016
27	Test	1 Days	30.12.2016	30.12.2016

Şekil 3.1 Gantt Diyagramı

3.3 Ekonomik Fizibilite

Proje kapsamında kullanılan kütüphane ve platformlar ücretsiz olarak kullanıcılara sunulmaktadır. Bunlara ek olarak Tablo 3.2 de diğer maliyetler hesaplanmıştır.

Tablo 3.2 Maliyet Tablosu

Web Kamerası	50 TL
Geliştirme Bilgisayarı (2 Adet)	2000 TL
Programın Çalışacağı Bilgisayar	1000 TL
Geliştirici Elemanlar (2 Kisi)	433 x 2 x 13 = 11258* TL
Toplam	14308 TL

*((Asgari ücert)/3) x Kisi Sayısı) x Projenin Hafta Sayısı

3.4 Yasal Fizibilite

Projenin yapılabilmesi için bölüm başkanlığından gerekli izinler alındı. Projenin yapılabilmesi için gerekli olan geliştirme ortamları açık kaynak kodlu olup ücretsiz olarak dağıtılmaktadır. Herhangi bir lisans sınırlaması olmayan bu ortamlar projede kullanılmıştır. Projenin yasalara uygunluğu kontrol edilmiş ve hiçbir engelin olamadığına karar verilmiştir.

4
Sistem Analizi

Bu aşamada ihtiyaçların belirlenmesi ve proje gereksinimlerinin saptanması için danışmanımız Doç. Dr. Songül ALBAYRAK ile toplantılar yapılmış, bu toplantılar neticesinde kullanılacak yöntemler ve izlenilecek yollar belirlenmiştir. Kullanılacak kütüphane olan OpenCv incelenmiş ve gerekli fonksiyonların tespiti yapılmıştır.Son olarak da veri akış diyagramları çizilerek projenin sistem analizi tamamlanmıştır.

4.1 OpenCV

OpenCV yaygın olarak kullanılan açık kaynak bir görüntü işleme kütüphanesidir. OpenCv beş temel bileşenden oluşur. Bunlar aşağıda sıralanmıştır.[3]

- **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.

Proje kapsamında bu beş temel bileşenden en uygun olan bileşenler seçilip kullanılacaktır.

4.2 HOG

Açılımı "Histogram of Oriented Gradients" olan HOG kullanımı ilk defa Shashua[4] ve Dalal[5] tarafından önerilmiştir. HOG yönteminde amaç görüntüyü bir grup lokal histogram olarak tanımlamaktır. Bu gruplar, görüntünün lokal bir bölgesindeki gradyanların yönelimlerinde, gradyanların büyüklüklerinin toplandığı histogramlardır. Bir görüntünün HOG değerlerinin çırakılması için gerekli olan hesaplamalar şu şekilde gerçeklenmektedir. İlk olarak görüntünün, yatay ve dikey Sobel filtreleri uygulanarak, I_x ve I_y olmak üzere kenarları belirlenir (Denklem 4.1 ve 4.2). Sobel filtresi uygulanmış I_x ve I_y görüntüleri kullanılarak, gradyan ve bu gradyanların yönelim açılarının hesaplanması için (G ve θ) aşağıdaki formüller kullanılır.[6]

4.2.1 HOG Description

Bir HOG Description oluştururken aşağıdaki bilgiler kullanılmaktadır.

1. Block Size: (8x8) Görüntüyü tarayacağı blokların boyutunu ifade eder. Bu bloklar görüntü üzerinde birer kare atlayarak soldan sağa ve yukarıdan aşağıya doğru gezerek görüntünün histogramını çıkarmak için kullanılmaktadır.
2. Cell Size: (4x4) Her bir block içerisindeki kareleri verilen değer kadar yeni karelere ayırmaktadır. Bu sayı ne kadar fazla olursa histogram değeri o kadar hassas ve desciptionun boyutu o kadar uzun olmaktadır.
3. Win Size: (64x48) Sisteme giren görüntünün boyutunu ifade eder.
4. Bin: (9) Histogram oluşturmak için aralıkların belirlenmesi gerekir. Bin değeri histogramın kaç aralıktan oluşacağını belirtmektedir.

Bu girdiler sonucunda, OpenCV kütüphanesinin "hogCompute" methodu tarafından 5940 uzunluğunda bir HOG Description vektörü hesaplanmaktadır. Bu vektörün her bir değeri sınıflandırma için kullanılacak olan öznitelikleri oluşturmaktadır.

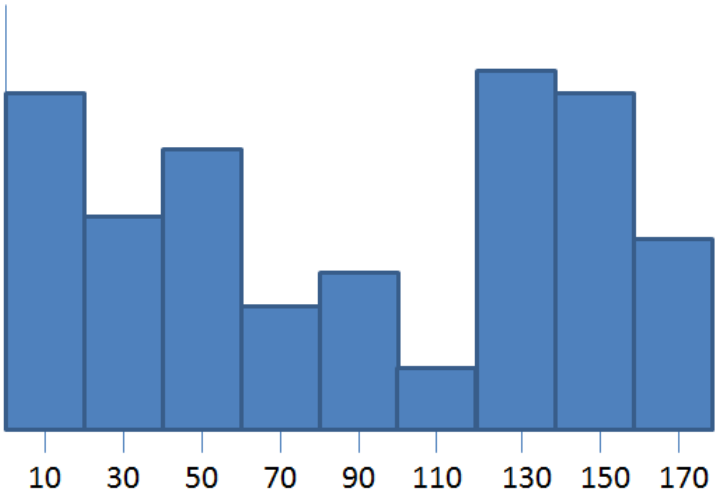
$$I_x = I * S_y \tag{4.1}$$

$$I_y = I * S_d \tag{4.2}$$

$$|G| = \sqrt{I_x^2 + I_y^2} \tag{4.3}$$

$$\theta = \arctan \frac{I_x}{I_y} \tag{4.4}$$

Formüller sonucu hesaplanan her değer 0-180 derecelik belirli aralıkla bölünmüş bir vektör üzerine yerleştirilir. Bu yerleştirme sonucunda ayrılmış her bin kendine ait bir değer alır. Binlerin HOG işlemi sonucu temsili görüntüsü Şekil 4.1'deki gibidir.



Şekil 4.1 Bin Grafiği

4.3 K-Nearest Neighbors

Kısaca K-NN olarak adlandırılan bu algoritma, oyun esnasında yapılan el şeklini veritabanında bulunan el şekilleri ile karşılaştırarak en uygun sonucu bulmamızı sağlamaktadır.

Bu algoritma beş adımdan oluşur.[7]

1. Öncelikle K değeri belirlenir.Bu değer yeni gelen verinin oluşan kümeler içinde en yakın kaç komşusuna bakacağını belirtir.
2. Diğer nesnelerden hedef nesneye olan öklit uzaklıkları hesaplanır.
3. Uzaklıklar sıralanır ve en minimum uzaklığa bağlı olarak en yakın komşular bulunur.
4. En yakın komşu kategorileri toplanır.
5. En uygun komşu kategorisi seçilir.

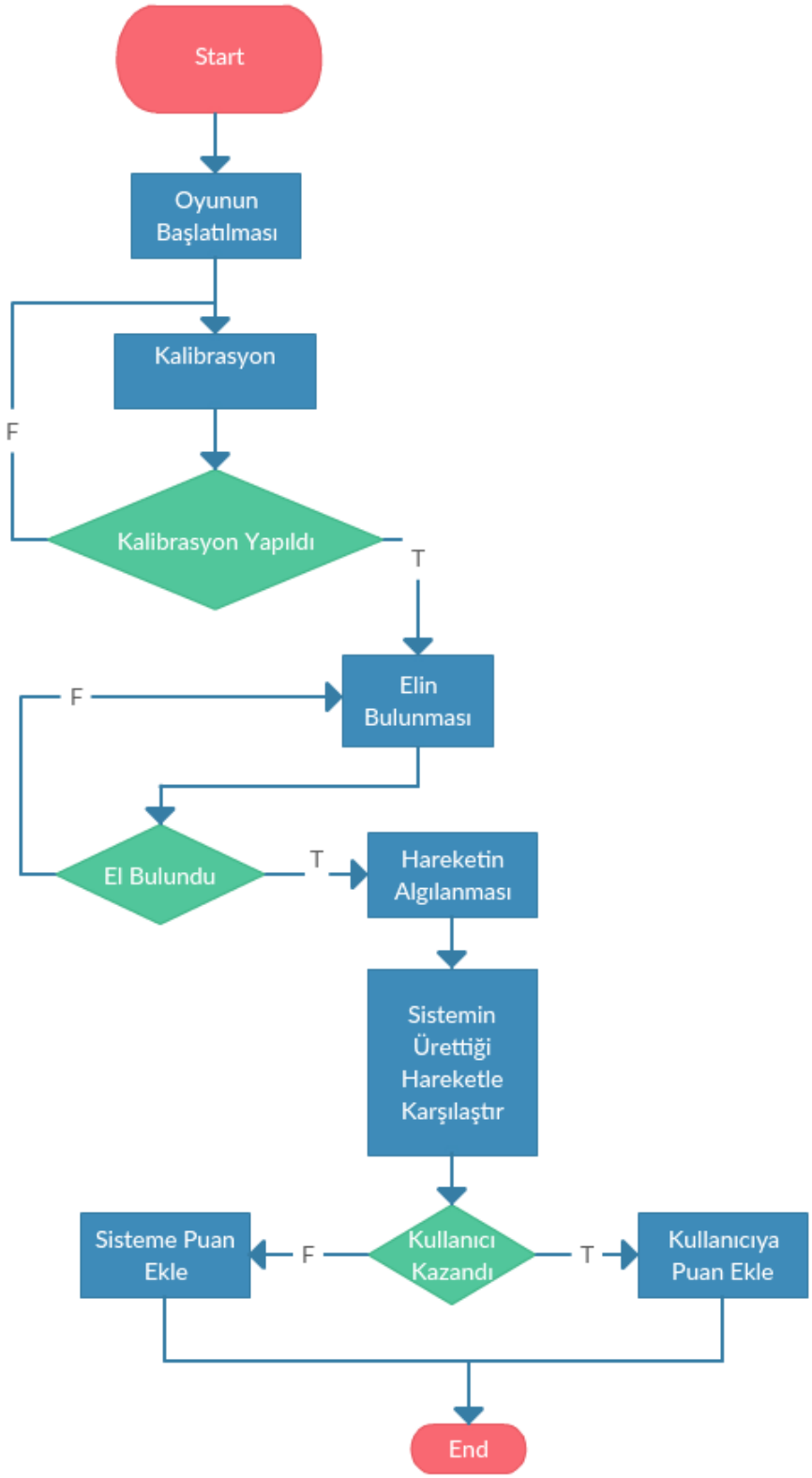
4.4 Standart Web Kamerası

Proje kapsamında yapılan araştırmalar sonucu, günümüzde hemen hemen her dizüstü bilgisayarda bulunan ve haricen kullanılan web kameraları ortalama olarak benzer özelliklere sahiptir. Bu özellikler;

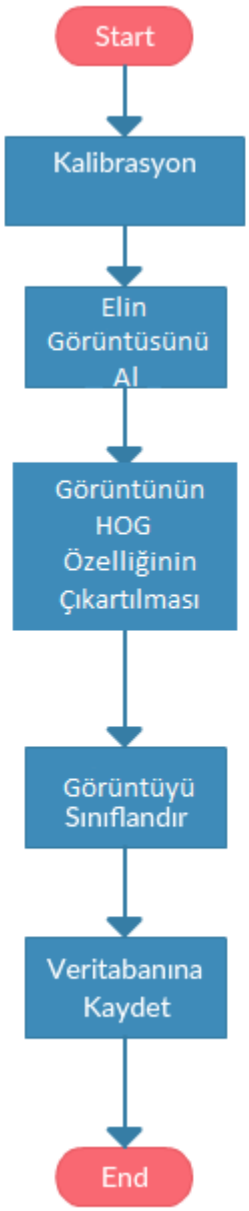
- **Çözünürlük:** 640 x 480,
- **Pixel:** 1.3,
- **Lens:** Standart Lens,
- **Video Çekimi:** 30 FPS.

Şeklinde sıralanabilir.

4.5 Program Akış Diyagramları



Şekil 4.2 Oyun Akış Diyagramı



Şekil 4.3 Eğitim Akış Diyagramı

4.6 Kullanım Senaryoları

4.6.1 Kalibrasyon

Aktörler: Kullanıcı

4.6.1.1 Amaç

Bu senaryo sistemin ortam koşullarını ve buna bağlı olarak kullanıcının el renk skalasını anlamasını amaçlamaktadır.

4.6.1.2 Olay Akışı

1. Temel Akış
    (a) Aktör kalibrasyon bölümüne girer.
    (b) Elini, ekranda bulunan işareti avucunun içine gelecek şekilde ayarlar.
    (c) Aktör kalibre et butonuna tıklar.
    (d) Sistem ortam koşullarını ve kullanıcının el renginin skalasını anlamıştır.

4.6.2 Sistemin Eğitimi

Aktörler: Kullanıcı

4.6.2.1 Amaç

Bu senaryo sistemin kullanıcının el hareketini tanıması için gerekli olan veritabanını oluşturmayı amaçlar.

4.6.2.2 Olay Akışı

1. Temel Akış
    (a) Kullanıcı "Sistemi Eğit" bölümüne girer.
    (b) Elinin resmini çeker.
    (c) Elinin şekline göre “Taş", "Kağıt" yada "Makas” resmi olarak işaretler.
    (d) Bu bilgiler sisteme kayıt edilir.
    (e) Her eğitim için bu akış tekrarlanır.

4.6.2.3 Ön-Koşul

Kullanıcının kalibrasyon işlemini yapmış olması beklenir.

4.6.2.4 Son-Koşul

Eğitim verisi sisteme eklenir.

4.6.3 Oyun

Aktörler: Kullanıcı

4.6.3.1 Amaç

Bu senaryoda kullanıcının oyunu nasıl oynayacağı amaçlanmaktadır.

4.6.3.2 Olay Akışı

1. Temel Akış
    (a) Kullanıcı oyunu başlatır.
    (b) “Taş", "Kağıt" veya "Makas” ‘dan herhangi birini seçerek 3 saniye boyunca eliyle seçilen hareketin şeklini yapar.
    (c) Sistem sayacı sıfırladığında elinin görüntüsü çekilir.
    (d) Sistem bu görüntüyü veritabanında ki eğitim seti ile karşılaştırarak elin şekline karar verir.
    (e) Sistemde rastgele olarak “Taş", Kağıt" ve "Makas”dan birini seçer.
    (f) Üstünlük kimde ise o kazanır ve hanesine puan eklenir.
    (g) Oyun 5 tur boyunca b adımından itibaren tekrarlanır.
2. Alternatif Akış
    (a) Sistem el şeklini tanımazsa, Temel Akış bölümü madde b’ye geri döner.

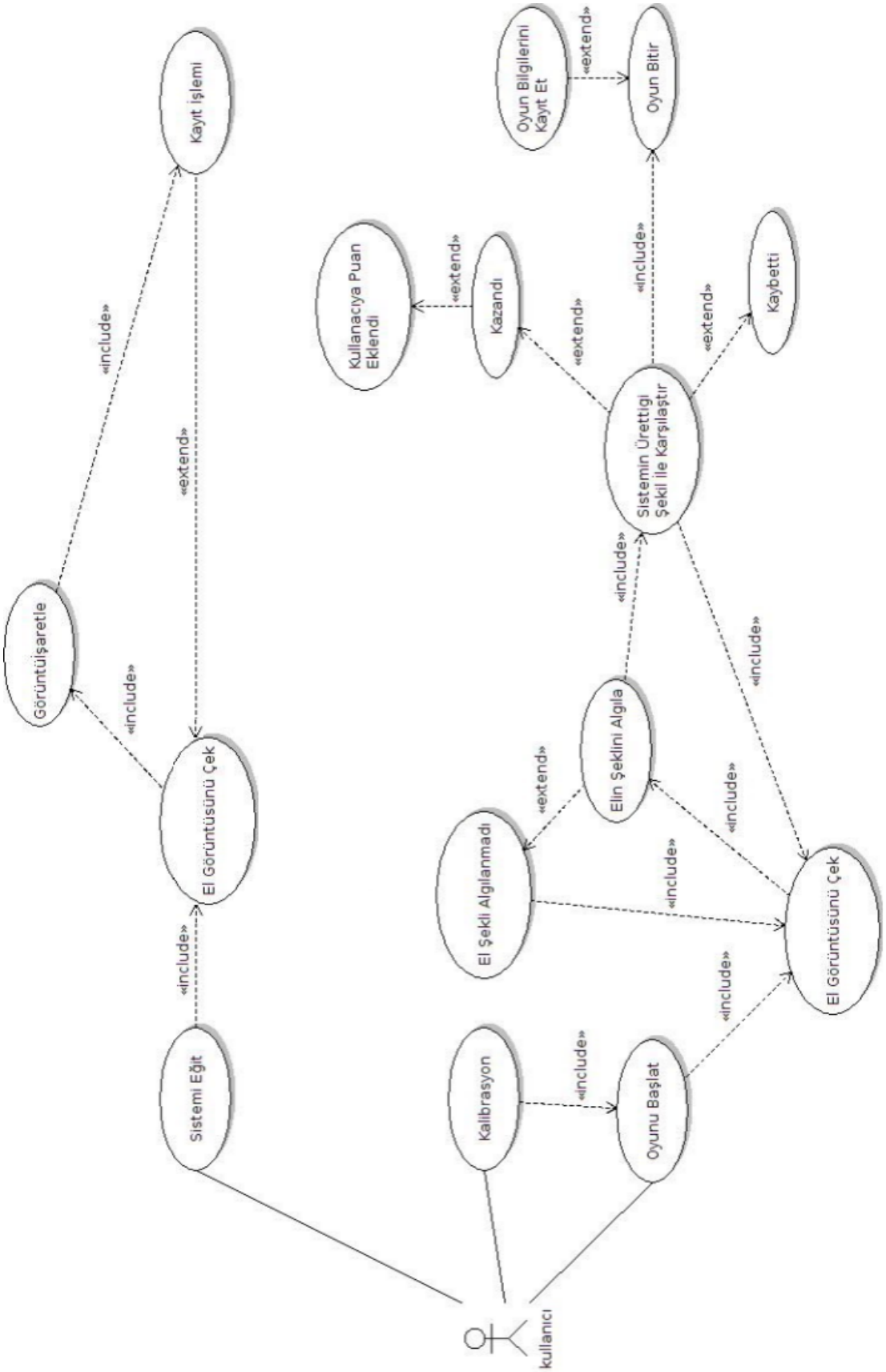
4.6.3.3 Ön-Koşul

Kullanıcının kalibrasyon işlemini yapmış olması beklenir.

4.6.3.4 Son-Koşul

Oyun bilgileri kayıt edilir.

4.7 Use Case Diyagramı



Şekil 4.4 Use Case Diyagramı

5
Sistem Tasarımı

5.1 Yazılım Tasarımı

Projenin iki temel modül üzerinde çalışması planlanmıştır. Bunlar, "El Algılama Modülü" ve "Eğitim Ve Test Modülü" olarak belirlenmiştir. Bu modüller projenin iki temel amacını yansıtmaktadır.

5.1.1 El Algılama Modülü

Bu modül, kullanıcının oyunu oynamak ve sistemi eğitmek için ortak olarak kullanacağı bir modül olarak tasarlanmıştır.

Modülün temel görevi, kullanıcının el rengi üzerinden renk kalibrasyonunu ve elin arka plandan ayrılmasını sağlamak olarak tanımlanmıştır.

El nesnesi arkaplandan ayrıldıktan sonra, hem oyun için hemde eğitim için kullanıma uygun hale gelmektedir.

5.1.2 Eğitim ve Test Modülü

Bu modül, kullanıcının sistemi eğitmesi ve oyun içinde kullanıcıdan gelen el şekillerini veritabanındaki mevcut el şekilleriyle karşılaştıracak bir modül olarak tasarlanmıştır.

Modülün temel görevi, oyun için kullanılacak "Taş-Kağıt-Makas" şekillerinin sisteme tanıtılması ve bir veritabanına etiketlenerek kayıt edilmesi olarak tanımlanmıştır. Modülün diğer bir görevi ise, oyun esnasında kullanıcıdan gelen el şekillerini algılayıp veritabanındaki diğer şekillerle karşılaştırarak kullanıcının yaptığı şekli tanımak olarak tanımlanmıştır.

5.2 Veritabanı Tasarımı

Projede, el şekillerinin sınıflandırılması ve bu şekillerin HOG değeri ve sınıf bilgisinin kayıt edilmesi gerekmektedir. Bu amaçla bir veritabanı kullanılması gerekmektedir.

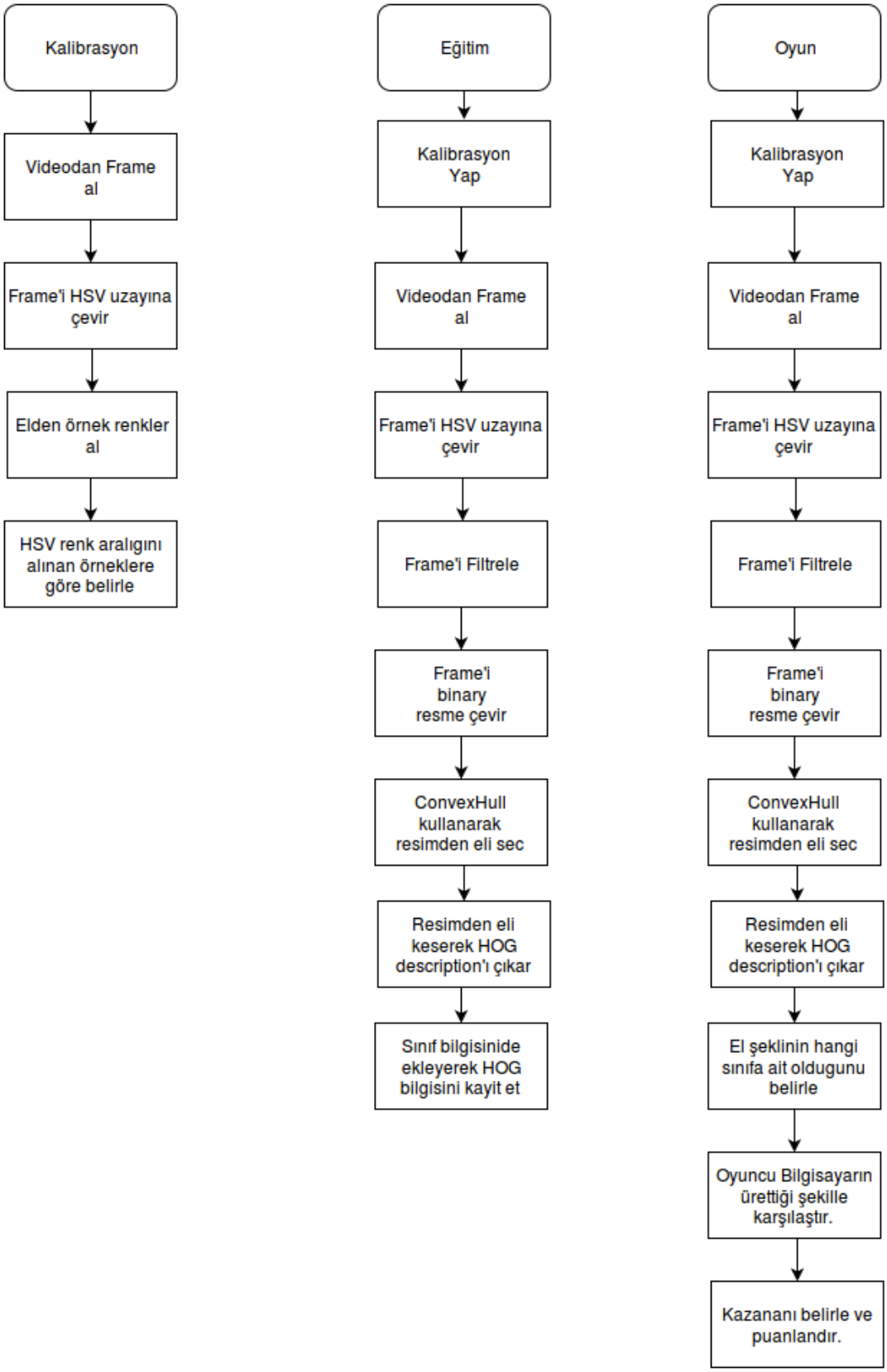
5.3 Girdi-Çıktı Tasarımı

Programın girdileri, kullanıcının kamerasından alınan video görüntüsündeki el nesnesi olarak belirlenmiştir. Bunun yanı sıra eğitim için kullanılan el şekilleri de programın girdileri arasında yer almaktadır.

Programın çıktıları, kullanıcının yaptığı el şekli ile veritabanındaki el şekillerinin eşleşmesi sonucu döndürülecek olan şeklin sınıf etiketi olarak belirlenmiştir. Oyun esnasında bu geri dönüş puanlama işlemi için kullanılacaktır.

5.4 Akış Diyagramı

Projeye ait modüllerin temel akış diyagramları Şekil 5.1'de görülebilir.



Şekil 5.1 Akış Diyagramı

6
Uygulama

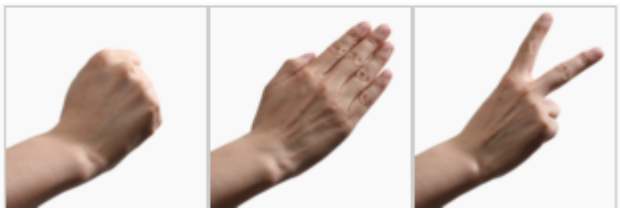
6.1 Oyun Hakkında Genel Bilgi

Taş-Kağıt-Makas oyunu iki kişi arasında oynanan bir oyundur ve kuralları oldukça basittir. Kurallar,

1. Taş, Makası kırarak yener,
2. Kağıt, Taşı sararak yener,
3. Makas, Kağıdı keserek yener,
4. Beraberlik durumunda oyun baştan başlatılır.

Şeklinde sıralabilir.

Şekil 6.1'de oyunda kullanılacak şekiller soldan sağa Taş-Kağıt-Makas olarak gösterilmektedir.

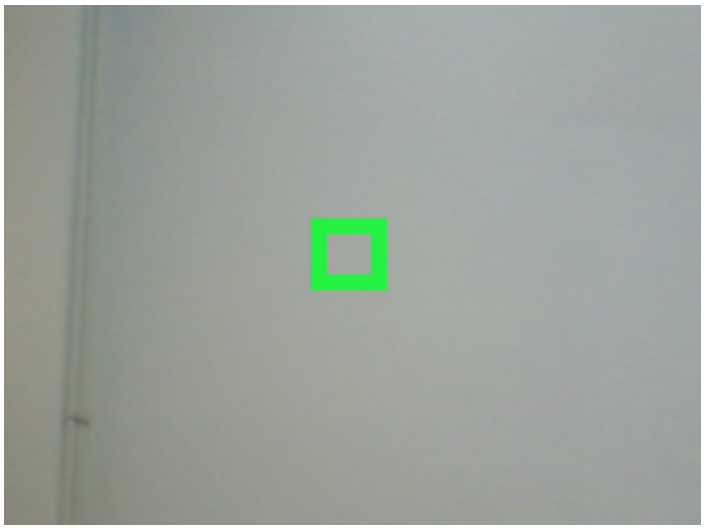


Şekil 6.1 Taş-Kağıt-Makas Şekilleri

6.2 Oyun Kurallarının Uygulanması

Proje kapsamında oyunun temel kuralı görüntü işleme yöntemleri kullanılarak gerçeklenmektedir.

Kullanıcı öncelikle sistemin kalibrasyonunu yapmalıdır. Kalibrasyon işlemi, elin rengini almak için gerekli olan bir işlemdir. Bu işlemi yapabilmek için video çerçevesi içerisinde bulunan kare içerisine el getirilerek "Kalibre Et" butonuna basılır. Kalibre ekranı Şekil 6.2'de görülmektedir. Bu işlem sonucunda o kare içerisinde kalan renk aralıkları alınarak arkaplandan eli ayırmak için kullanılır. Bu aşamadan sonra kullanıcı oyuna başlayabilir veya sistemi eğitebilir.



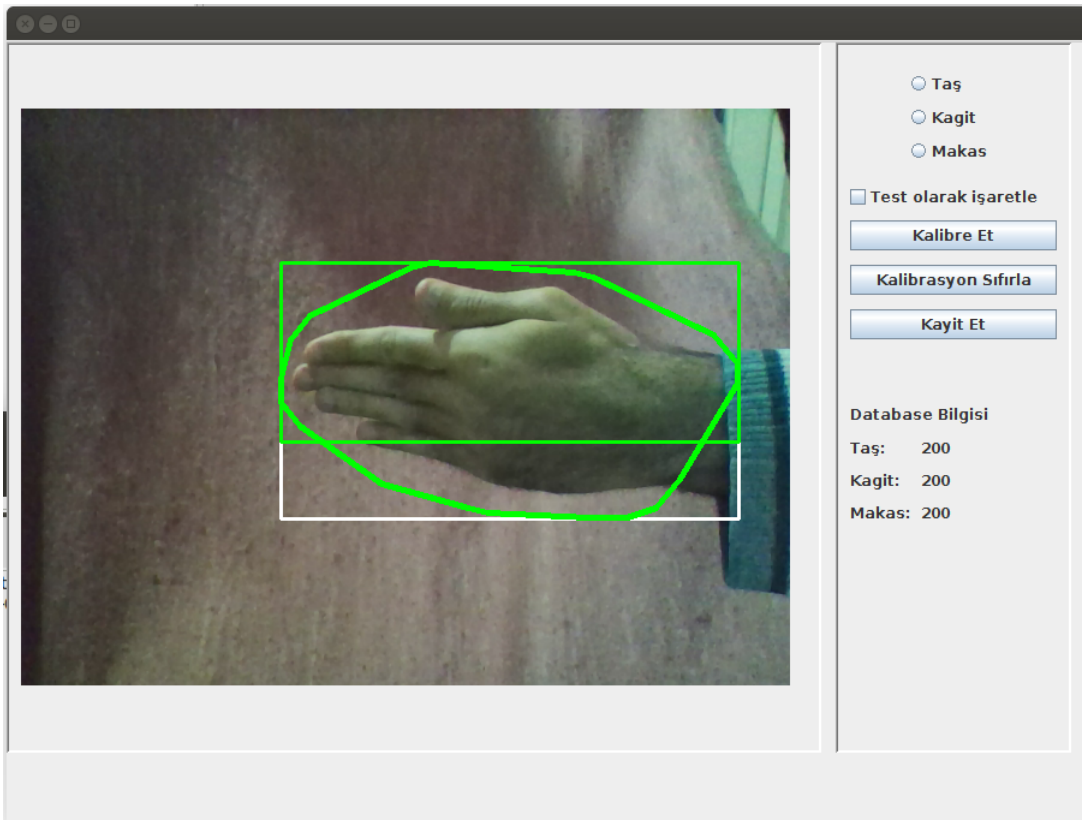
Şekil 6.2 Kalibrasyon Ekranı

6.2.1 Sistemin Eğitimi

Kullanıcı kalbirasyon işlemi ardından kendi el şekillerini sistem veritabanına ekleyerek sistemin eğitimine katkı sağlayabilir. Sistemin eğitimi, sistemin başarısını en yükseğe çıkartmak için tasarlanan bir yapıdır.

Modüllerin gerçeklenmesi aşamasında "Eğitim ve Test Modülü" gerçeklenirken "Cambridge Hand Gesture Data Set" [8] veri setinden yararlanılmıştır. Projenin ilerleyen dönemlerinde bu veri seti veritabanından kaldırılıp yerine gerçek kullanıcılardan toplananan veriler kaydedilmiştir. Bu veriler her kullanıcıdan her bir şekil için 20 örnek alınarak oluşturulmuş.

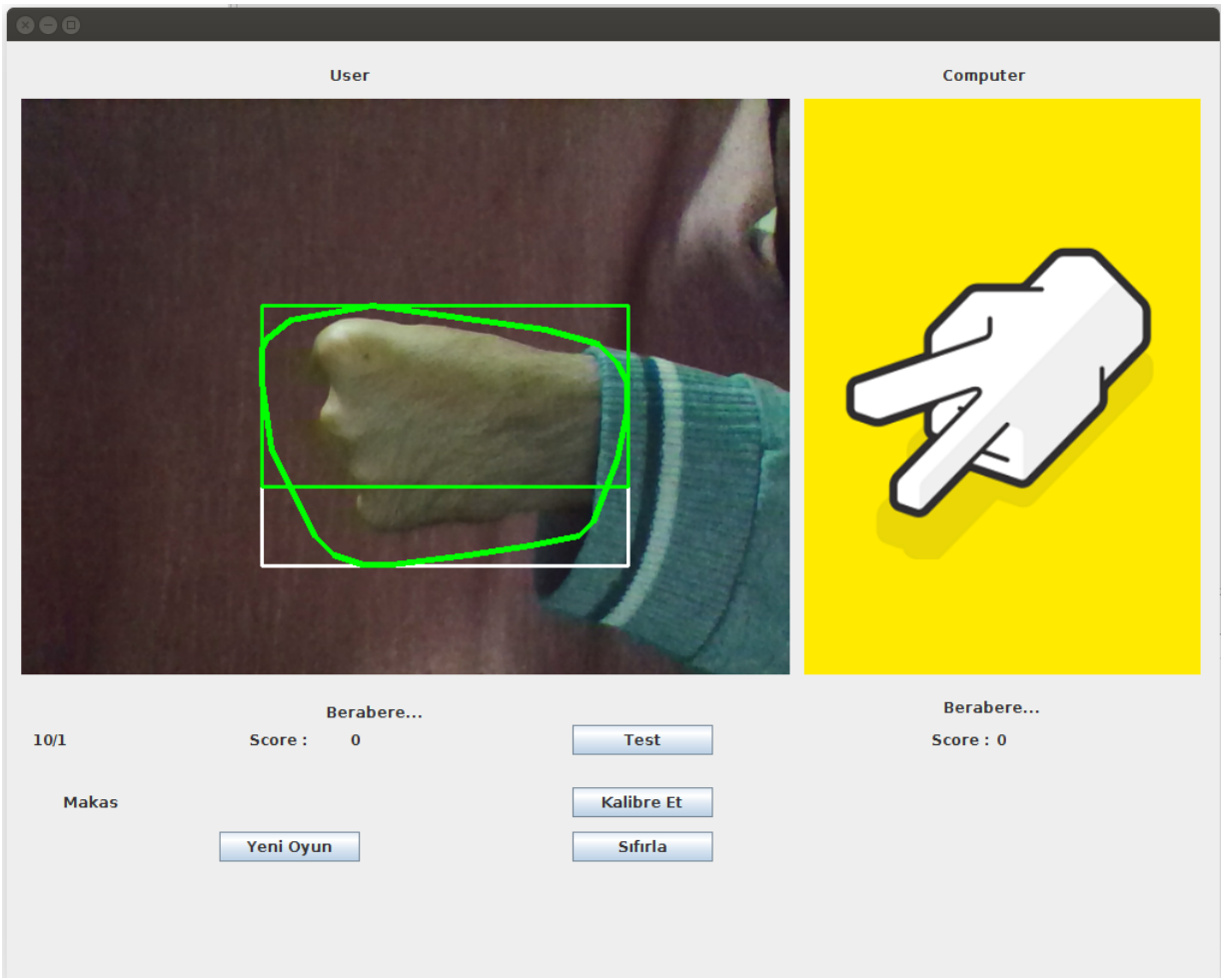
Bu işlemler HOG Description değerlerinin veritabanına yazılması sonucunda oluşturulur. Bu değerler için kullanılan parametreler HOG Description başlığı altında incelenmiştir.



Şekil 6.3 Eğitim Ekranı

6.2.2 Oyun

Kullanıcı kalibrasyon işlemini eğitim ve oyun kısmında ayrı ayrı yapmalıdır. Oyun kısmında kullanıcının kalibre ettiği el görüntüsü anlık olarak ekrandadır. "Oyuna Başla" butonuna bastığı anda sistem üçten geriye doğru sayarak kullanıcının bu geri sayım sonunda elin şeklini yapmış olmasını bekler. Yapılan şeklin HOG değerleri çıkartılır ve veritabanında daha önceden var olan şekiller ile karşılaştırma yapılır. Yapılan şekle göre sistem, veritabanından uygun olan sınıf etiketini bulur ve sistem bu sırada rastgele olarak bir şekil seçer. Kullanıcının yaptığı şekil ile sistemin yaptığı şekil karşılaştırılarak, oyun kurallarına göre kim üstünlük sağladıysa ona puan ekler. Beraberlik durumunda ise puanlamada bir değişim olmaz oyunun tekrar başlatılması beklenir.



Şekil 6.4 Oyun Ekranı

7
Deneysel Sonuçlar

Proje belirli aralıklarda test işleminden geçmiştir. Bu test işlemleri ardından elde sonuçlar aşagıda incelenmiştir.

HOG Description sonucunda oluşturulan vektörler eğitim kısmında veritabanına şeklin etiketi (sınıf bilgisi) ile birlikte kaydedilmektedir. Bu aşamadan sonra oyun kısmında kullanıcının yaptığı el şeklini, oluşturulan bu veritabanından K-NN ile en benzer 1 komşuluğa bakılarak tahmin etmeye çalışılmaktadır.

Proje kapsamında, bu veriler proje dışında bir programda test edilmiştir. Weka isimli bu progam ile yapılan testler sonucunda başarının ortalama %80 olduğu gözlemlenmiştir. Proje içerisinde ise yeni gelen şeklin sınıflandırılması işlemi iki şekilde karşılaştırılmıştır. Bunlar;

1. Kullnıcı eğitim yapmadan önce: Bu kısımda veritabanında kullanıcıya ait el şekilleri bigisi bulunmamaktadır. Başarı eski kullanıcıların verileri üzerinden test edilmiştir ve başarı %63-86 arasında değişim gösterdiği belirlenmiştir. Confusion Matrix Tablo 7.1 ve Tablo 7.2'de gösterilmektedir.

   Weka çıktı özetleri:

   (a) Correctly Classified Instances 19 %63.3333

   (b) Kappa statistic: 0.45

   (c) Mean absolute error 0.2459

   (d) Root mean squared error 0.4927

   (e) Relative absolute error %55.3191

   (f) Root relative squared error %104.5107

   (g) Total Number of Instances 30

Tablo 7.1 Confusion Matrix 1

a	b	c	<– classified as
3	7	0	a = 1
0	10	0	b = 2
0	4	6	c = 3

(a) Correctly Classified Instances 26 %86.6667

(b) Kappa statistic 0.8

(c) Mean absolute error 0.0914

(d) Root mean squared error 0.2971

(e) Relative absolute error %20.5674

(f) Root relative squared error %63.0249

(g) Total Number of Instances 30

Tablo 7.2 Confusion Matrix 2

a	b	c	<– classified as
9	1	0	a = 1
0	10	0	b = 2
0	3	7	c = 3

2. Kullanıcı eğitim yaptıktan sonra: Bu kısımda veritabanında kullanıcıya ait el şekilleri bigisi halihazırda bulunmaktadır. Başarı eski kullanıcıların verileri ve kendi verileri üzerinden test edilmiştir ve başarı %83-96 arasında değişim gösterdiği belirlenmiştir. Confusion Matrix Tablo 7.3 ve Tablo 7.4'de gösterilmektedir.

   Weka çıktı özetleri:

   (a) Correctly Classified Instances 25 %83.3333

   (b) Kappa statistic 0.75

   (c) Mean absolute error 0.1133

   (d) Root mean squared error 0.3322

   (e) Relative absolute error %25.4967

   (f) Root relative squared error %70.4793

   (g) Total Number of Instances 30

Tablo 7.3 Confusion Matrix 3

a	b	c	<– classified as
8	2	0	a = 1
1	9	0	b = 2
1	1	8	c = 3

(a) Correctly Classified Instances 29 %96.6667

(b) Kappa statistic 0.95

(c) Mean absolute error 0.025

(d) Root mean squared error 0.1486

(e) Relative absolute error %5.6291

(f) Root relative squared error %31.5248

(g) Total Number of Instances 30

Tablo 7.4 Confusion Matrix 4

a	b	c	<– classified as
9	1	0	a = 1
0	10	0	b = 2
0	0	10	c = 3

8
Performans Analizi

Projenin performansını, proje kapsamında kullanılan platformalar oldukça etkilemektedir. Platformaların biribirleri ile uyumları ve ayrı ayrı kullanıldıklarında verilen performanslar değişim göstermektedir.

Proje gerçeklenirken, geliştirme dili olarak Java, görüntü işleme kütüphanesi olarak OpenCV ve veritabanı olarak PostgreSQL kullanılmıştır.

Her birinin ayrı ayrı perfonması birbirinden bağımsız olarak oldukça başarılı sonuçlar getirmektedir. Ancak birlikte kullanıldıkları zaman performanslarında düşüş gözlenmektedir. Bunun sebebi bu platformaların uyumluluğudur.Örneğin OpenCV kütüphanesi C++ ile kodlanmış bir kütüphanedir. Java ile bu kütüphanenin kullanılması performans düşüşüne sebebiyet vermektedir. Bunun sebebi Java'da işlerin bir sanal makine üzerinden yürütülüyor olmasıdır. Java Virtual Machine (JVM) tüm Java işlemelerinin üzerinden geçtiği bir köprü görevi gördüğünden bu işleri biraz yavaşlatmaktadır. Eğer bu proje Java yerine C++ ile derlense daha iyi performans elde edilebilir.

Veritabanına eklenen HOG Decription değerlerinin uzun olması da performansı oldukça etkilemektedir. Bunun sebebi 1500 kayıt için yaklaşık 200 milyon karakter taraması gerekmektedir. Bu da oyun ekranının açılış hızını düşürmektedir. Kayıt sayısı arttıkça bu değerlerde artacak buna bağlı olarak performansta düşüş görülecektir.

9
Sonuç

Projede, genel el şekli tanıma başarısı yaklaşık %80 olduğu gözlemlenmiştir. Yöntem olarak OpenCV kütüphanesi içerisindeki HOG yöntemleri kullanılarak tanımlama ve eğitim kısmı yapılmıştır. K-NN algoritması ile de HOG değerlerinin yeni gelen şekiller için sınıflandırılması sağlanmıştır. K-NN algoritmasında 1 komşuluk ile sınıf belirleme işlemi yapılmıştır. K-NN'de komşuluk sayısı arttıkça başarının düştüğü gözlemlenmiştir.

Uygulama başarısı belirli kısıtlamalar kabul edilerek test edilmiştir. Bunlar;

1. Ekranda sadece el görüntüsü olmalı, yüz, kol gibi üzerinde kıyafet olmayan vücut parçalarının frame içerisinde kalmaması gerekmektedir.
2. Kullanıcı kalibrasyon işlemini her seferinde eksiksiz olarak yapmalıdır.

Bunların dışında, arkaplan ne kadar sade (tek renk) olursa başarı oranı o kadar yükselmektedir.

Gelecek çalışmalarda bu sistem daha yüksek çözünürlüklü kameralar ile veya Kinect kullanılarak geliştirilebilir. Bu sayede renk kalitesi artan örneklerin ayrımı daha belirgin olabilir. Kinect kullanırak yapılan sistem ise mesafe bilgisinden ötürü oldukça yüksek başarılar sağlayabilir.

Referanslar

[1] OpenCV, *Opencv tutorials*, 2016. [Online]. Available: http://docs.opencv.org/2.4/doc/tutorials/tutorials.html.

[2] S. Andresen, *Hand detection cv*, 2015. [Online]. Available: https://github.com/simena86/handDetectionCV.

[3] M. Pişkin, *Opencv nedir?* 2016. [Online]. Available: http://mesutpiskin.com/blog/opencv-nedir.html.

[4] A. Shashua, Y. Gdalyahu, and G. Hayon, "Pedestrian detection for driving assistance systems: Single-frame classification and system level performance," 2004.

[5] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," 2005.

[6] M. Peker, H. Altun, and F. Karakaya, "Hog temelli bir yöntem ile ölçek ve yönden bağımsız gerçek zamanlı nesne tanıma,"

[7] E. UZUN, *K-nn algoritması*, 2016. [Online]. Available: http://bilgmuh.nku.edu.tr/erdincuzun/post/k-nn-algoritmasi.

[8] T.-K. Kim and R. Cipolla, *Cambridge hand gesture data set*, 2009. [Online]. Available: http://www.iis.ee.ic.ac.uk/icvl/ges_db.htm.

Özgeçmiş

1. ÜYENİN KİŞİSEL BİLGİLERİ

İsim-Soyisim: Abdullah KAYAHAN
Doğum Tarihi ve Yeri: 20.01.1992, Yozgat
E-mail: abdullah.kayahan1@gmail.com
Telefon: 0538 783 03 97
Staj Tecrübeleri: TRT Bilgi İşlem Dairesi Başlanlığı Yazılım Departmanı ANKARA

2. ÜYENİN KİŞİSEL BİLGİLERİ

İsim-Soyisim: Uhud POYRAZ
Doğum Tarihi ve Yeri: 24.02.1991, İstanbul
E-mail: uhud.poyraz@gmail.com
Telefon: 0538 350 04 96
Staj Tecrübeleri: Plus Proje Yazılım Departmanı İSTANBUL

Proje Sistem Bilgileri

Sistem ve Yazılım: Linux İşletim Sistemi, Java
Gerekli RAM: 2GB
Gerekli Disk: 512MB