

TEKNOFEST

HAVACILIK, UZAY VE TEKNOLOJİ FESTİVALİ

İNSANLIK YARARINA TEKNOLOJİ YARIŞMASI

PROJE DETAY RAPORU

PROJE KATEGORİSİ: Engelli Dostu

PROJE ADI: Dijital Gözüm ile Gerçek Zamanlı Göz ve Baş Hareketleri Kullanılarak Bilgisayar Kontrol Yazılımı Geliştirilmesi

TAKIM ADI: Cybereyes

Başvuru ID: 50968

TAKIM SEVİYESİ: Lise

İçindekiler

1. Proje Özeti (Proje Tanımı)	2
2. Problem/Sorun:.....	3
3. Çözüm	3
4. Yöntem	4
4.1 Kullanılan Yazılım ve Kütüphaneler.....	4
4.2 Program Açılışı.....	4
4.3 Göz Kırpma ve Algılama.....	5
4.4 Göz merkezinin hesaplanması	5
4.5 Göz bebeğinin tespit edilmesi	5
4.5 Kaş hareketi algılama	6
4.6 Kısayol İşlemleri	6
4.7 Programı uyku moduna alma Kısayolu	7
4.8 Mod (Seçenek) 1.....	7
4.9 Mod (Seçenek) 2.....	8
4.10 Arayüz	8
5. Yenilikçi (İnovatif) Yönü	8
6. Uygulanabilirlik.....	9
7. Tahmini Maliyet ve Proje Zaman Planlaması	9
8. Proje Fikrinin Hedef Kitle (Kullanıcılar):.....	9
9. Riskler	10
10. Kaynaklar	10

1. Proje Özeti (Proje Tanımı)

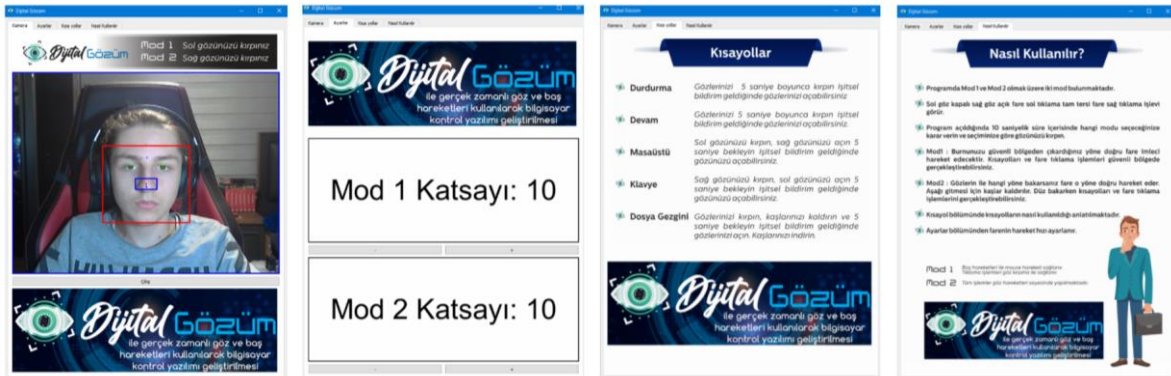
Bilgisayarların ve cihazların göz ve baş hareketleri ile kontrol edilebilmesi engelli bireylerin bilgisayar ile iletişime geçmesi ve bilgisayar ile aramızdaki iletişimi sırasında komut gönderme imkânı güzel bir destekleyici araç olacaktır. Bilgisayar ile dar bir bant genişliği ile iletişim kurmaktayız. Bilgisayar ile iletişim organlarımızı kullanarak komut gönderebilmemiz ve iletişim araçlarını arttırabilmemiz bu cihazları kullanmamız esnasında performansımızı geliştireceği gibi zamandan da tasarruf sağlayacaktır. Bu çalışma ile bilgisayarlardaki webcamler kullanılarak, baş hareketleri, göz hareketleri, jest ve mimikler tespit edilerek takip edilmektedir. Göz ve baş hareketlerinin belirlenen güvenli alanlar ile kıyaslaması sağlanmakta ve görüntü işleme teknikleri kullanılarak gerçek zamanlı ölçüm yapılmaktadır. Çalışmamız, bilgisayar kullanımında baş ve göz hareketlerini kullanabilmemiz için ek giriş kaynağı olacak ve bant genişliğini arttıracaktır. Göz hareketleri bilgisayar girdilerinden çok farklı olduğundan mevcut görüntü işleme kütüphanelerine eklemeler yapılarak, iki gözün farklı kombinasyonları, baş hareketleri, jest ve mimikler kullanılarak çok sık kullandığımız fare ve klavye kısayollarını da içeren komutlar tanımlanmıştır. Ayrıca kullanıcıların tercih edebileceği farklı modlar bir arayüz ile kullanıcıya sunulmuştur.

2. Problem/Sorun:

Gelişen teknoloji, sürekli artan veri ve kaynaklar ile bilgisayar kullanımı her geçen gün daha da artmakta birlikte gündelik hayatımızın vazgeçilmez bir parçası olmuştur. Fakat fiziksel engelli bireyler fare ve klavye kullanamadıkları için bilgisayarlar ile iletişime geçememektedirler. Bilgisayar kullanmak için aracı kişilere veya piyasadaki açık kaynak kodlu olmayan pahalı aracı cihazları edinmek zorundadırlar. Bu nedenlerden dolayı Dünya Sağlık Örgütü'nün verilerine göre dünya nüfusunun %15 gibi büyük bir bölümü çağımızın teknolojisi olan bilgisayarlar ve getirdikleri İnternet gibi teknolojilerden mahrum kalmaktadırlar. Bunun yanında engelsiz kullanıcılar ise daha etkin bilgisayar kullanımı için farklı programlar ararken bir yanda da bilgisayar ile iletişimi farkında olmadan daha etkin hale getirme çabasındadırlar. Araç kullanırken, ellerimizi kullanarak yaptığımız eylemler sırasında bilgisayar ve akıllı cihazlarımıza komut gönderememekteyiz çünkü bu cihazları kullanırken parmaklarımıza, ellerimize ve dolayısıyla kollarımıza ihtiyaç duyarız. Bireysel engellilerin teknolojiye ayak uydurmaları ve bilgisayar kullanımını kolaylaştırmak için geliştirdiğimiz bu proje aynı zamanda bilgisayar kullanımını daha etkin ve hızlı hale getirebilmek için de tasarlanmıştır.

3. Çözüm

Bilgisayarı sadece klavye ve Mouse ile değil, diğer duyu organlarımızı da kullanabilmemiz bilgisayarı çok daha etkin kullanmamızı sağlayacaktır. Biz de hazırlamış olduğumuz bu proje ile göz ve baş hareketlerini, jest ve mimikleri de kullanarak bilgisayara gönderdiğimiz komutları daha etkin ve zengin olmasını sağlamaktayız. Göz ve baş hareketleri ile Bilgisayarları yönetmek, daha hızlı, daha kullanışlı ve daha doğal kullanabilmeleri için güzel bir araçtır. İcat edilen cihazların etkileşim teknikleri sınırlıdır, o cihazı kullanabilmek için ellerimizi, parmaklarımızı ve kollarımız kullanmamız gerekir. Mevcut teknolojilerde, bilgisayardan kullanıcıya yönelim, kullanıcıdan bilgisayara göre daha zordur. Biz projemiz ile tek taraflı olan bu diyalogları, çift taraflı olmasına imkân sağlıyoruz. Bilgisayarlar, kamera aracılığıyla bizden gelen hareketleri girdi medyası olarak algılayacak gelen bu verileri hızlı bir şekilde işleyerek kullanıcı ile arasındaki bant genişliğini genişletmiş olacaktır. Göz hareketleri geleneksel bilgisayar girdilerinden çok farklı olduğundan iki gözün yapabileceği hareketler farklı kombinasyonlarla işlenerek komutlar geliştirilmiştir.



Şekil 1. Ekran Arayüzleri

İnsanoğlunun göz hareketleriyle yaptığı jest ve mimikler incelenerek küresel olarak benzer komutlar belirlenmiştir. Araştırmalarımız sonucunda bu desenler incelenerek yüksek düzeyde anlama sahip birçok göz hareketi tespit edildi. Bu duruma bir örnek verecek olursak bir duruma onay vermek için yaklaşık 1 saniye iki gözümüzü aynı anda kapatıp sonra tekrar açarız. Bilgisayarda da "Enter" tuşunun görevini göreceğiz komut olarak bu hareket belirlenmiştir. Bu tarz kombinasyonlar göz ile bilgisayar kullanımını daha doğal ve kullanışlı hale getirilebilmesi için farklı diyaloglar oluşturulmuştur.

4. Yöntem

4.1 Kullanılan Yazılım ve Kütüphaneler

Görüntü işleme metotları Python programlama dili ve OpenCV Kütüphanesi kullanılarak geliştirilmiştir. OpenCV, intel firması tarafından geliştirilmiş ve BSD lisansı ile ücretsiz kullanılabilen açık kaynak kodlu bir “bilgisayarla görme” kütüphanesidir. OpenCV, resim ve videoları işleyerek anlamlı veriler çıkarmak için kullanılmaktadır. NumPy (Numerical Python) kütüphanesi hesaplamaları çok hızlı bir şekilde sonuçlandırabilen gelişmiş bir matematik kütüphanesidir. Görüntü işleme sırasında birden çok nesneyi ve noktayı izlemek ve işleyebilmek için dlib kütüphanesi geliştirilmiştir. Dlib kütüphanesi sayesinde yüz üzerinde 68 nokta belirlenmiştir. Bu noktalar yüz üzerinde tasarlanan sanal bir koordinat sisteminden alınan noktalardır. PyAutoGUI kütüphanesi ile fare ve klavye kontrolü sağlanabilmektedir. Fare ve klavye kısayollarını kullanmak için PyAutoGUI kütüphanesinden yararlanılmıştır. Kullanıcılara işitsel dönütler verebilmek amacıyla python da en sık kullanılan kütüphanelerden biri olan playsound kütüphanesi projemizde kullandık. Süreli işlemler yapılması gerektiği durumlar için datetime kütüphanesi, kullanıcıların programımızı rahatlıkla kullanabilmeleri için grafiksel arayüzler tasarlanabilen PyQt5 ve yardımcı kütüphaneler kullanılmıştır. Programımızı executable (çalıştırılabilir) hale getirebilmek ve tüm bilgisayarlarda rahatlıkla kullanılabilmesi için pyinstaller kütüphanesinden yararlanılmıştır.

4.2 Program Açılışı

While döngüsü içerisinde cv2.VideoCapture() fonksiyonu ile kamera seçimini webcam olacak şekilde ayarlanması sağlandı. Read() fonksiyonu ile görüntü almaya başlanmaktadır. Read() fonksiyonu ile alınan görüntüler ters alınmaktadır. Cv2.flip() fonksiyonu ile bu görüntü çevrildikten sonra dlib.get_frontal_face_detector() fonksiyonu ile yüz bulunur. Programın başında bir bekleme süresi bulunmaktadır ve kalan süre programın arayüzünde görünür. Kullanıcı seçimini yaparak kimseye ihtiyaç duymadan bilgisayarını rahatlıkla kullanabilmektedir. Set dizisi oluşturularak datetime ile saniye set dizisinin içerisine tanımlandı. Belirtilen süreden bu set uzunluğu çıkarılarak kalan süre hesaplanarak programın arayüzünde yazdırılabilmektedir. Set uzunluğu bizim seçtiğimiz sayı ile eşleştiği takdirde bir değişken belirlendi. Bu bir arttırılarak süre bittikten sonra diğer aşamaya geçilmesi sağlandı. Süre bitiminde kırılan göze göre mod seçilir. Program açılışının çalışmasını sağlayan kodlar şekil 7’de gösterilmektedir.

```
while True:
    _, frame = webcam.read()
    frame = cv2.flip(frame, 1)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = detector(gray)

    if len(secondSet) < second:
        now = datetime.datetime.now()
        nowSecond = now.strftime("%S")
        secondSet.add(nowSecond)
        secondFrame = second - len(secondSet)
        cv2.putText(frame, f"{secondFrame}", (550, 75), font, 2, blue)

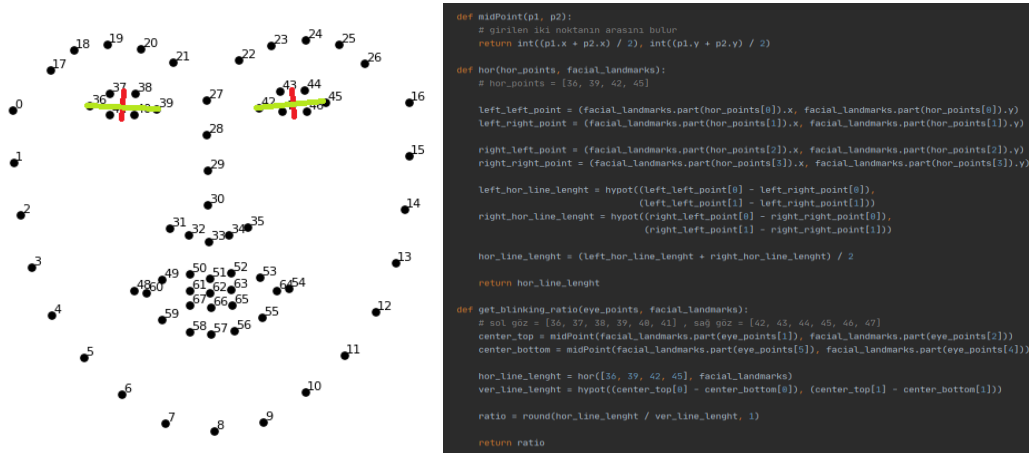
    if len(secondSet) == 11:
        secondPass += 1

    try:
        # ekrana anlık olarak bir yüz daha girmesi gibi durumlarda
        # hata verip programın çökmesini önlemek için try except içine alıyoruz
        for face in faces:
```

Şekil 7. Programın açılması ve seçim işleminin yapılması

4.3 Göz Kırpma ve Algılama

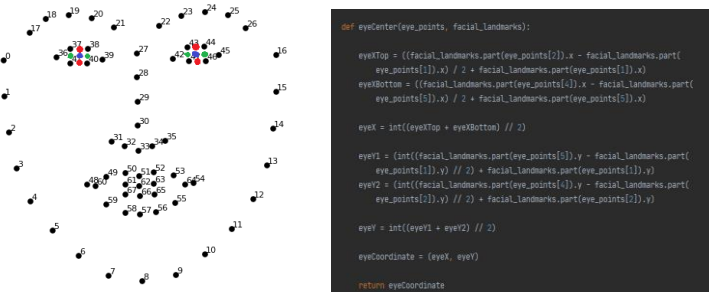
Dlib kütüphanesi sayesinde göz üzerinde belirlediğimiz noktaları kullanarak farklı yüz tipleri yüzünden oluşabilecek hata payı oranlama yapılarak ortadan kaldırılmıştır. Bu noktalar sol göz için 36,37,38,39,40,41 sağ göz için ise 42,43,44,45,46,47 olmak üzere 6 şar noktalardır. Bu hata oranı iki farklı fonksiyon sayesinde giderilmektedir. Fonksiyonlardan birincisi olan Hor gözün sol ve sağ noktaları arasındaki uzaklığı Pisagor yöntemi ile bulup ortalamasını alır böylece gözlerin yatay uzunluğu bulunmuştur. İkinci fonksiyon olan Get_blink_ratio sayesinde ise bir gözde diğer noktalardan yararlanarak gözün üstünde ve altında yeni noktalar oluşturulmuş pisagor yöntemi ile dikey uzunluk bulunup hor fonksiyonundan gelen yatay uzunluk ile oranlama yapılmıştır. Bu oran göz kırpma algılama işlemlerinde kullanılmaktadır. İkinci fonksiyon her iki göz için de ayrı ayrı kullanılmaktadır. Çıkan sonucun ortalaması ise kırpma algılamanın ayrı ayrı yapılmasının tespitinde ve ortalama doğruluk payını arttırmak için gerekmektedir.



Şekil 2. Hor() ve get_blinking_ratio() fonksiyonlarının hesapladığı noktalar ve gösterim

4.4 Göz merkezinin hesaplanması

Göz üzerinde bulunan altı noktanın dördü ortada kare (sol 37,38,40,41 sağ 43,44,46,47) oluşturacak şekildedir. Her bir kenarı oluşturan köşelerin orta noktasını midPoint fonksiyonu ile bulup x ve y değerlerinin ortalamasını alıp bir değişkene atıyoruz. Böylece gözün ortasının koordinat değerlerini buluyoruz.

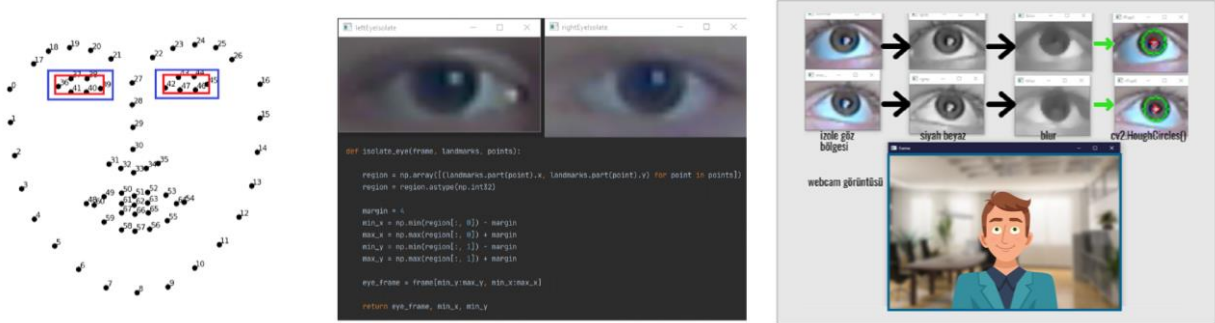


Şekil 3. Göz merkezinin hesaplanması fonksiyon ve gösterim

4.5 Göz bebeğinin tespit edilmesi

Göz bebeğini bulmak ve hatalardan kaçmak için göz bölgesini izole edip izole bölgede göz bebeği aramalıyız. Göz noktalarının bulunduğu koordinatları sol göz [36,37,38,39,40,41], sağ göz [42,43,44,45,46,47] sırası ile Numpy dizisine dönüştürdükten sonra çıkan elemanın sol sütunu noktaların x koordinatındaki yerini, sağ sütunu ise y koordinatındaki yerini vermektedir. Sol sütunun

ve sağ sütunun en küçük değerleri izole görüntünün sol üst koordinatıdır. Sol ve sağ sütunların en büyük değerleri ise izole görüntünün sağ alt koordinatını vermektedir. Oluşan görüntünün hata payını önlemek için alan büyütülür. Webcam görüntüsünün belirttiğimiz koordinatlar arasındaki bölümünü değişkene atayarak izole etme işlemi yapıyoruz.

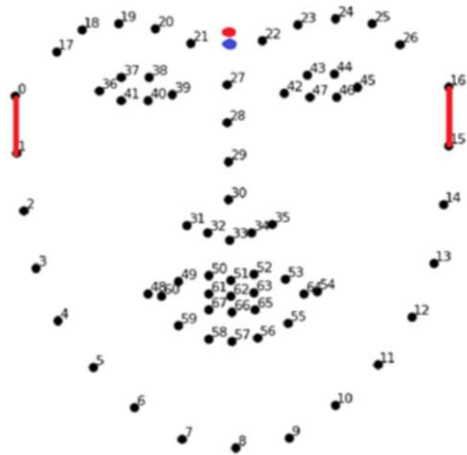


Şekil 4. İzole göz bebeğinin tespit edilmesi fonksiyon ve gösterim

İzole edilen sol ve sağ göz bölgeleri içerisinde OpenCV kütüphanesinde bulunan “Hough Circles Transform” yöntemi uygulanmaktadır. Cv2.circles metodu kullanılmak suretiyle iç ve dış dairelerin görüntüleri çizdirilmektedir. Daire bulunmadığı zaman ise none çıktısı verilmektedir. Göz kırpmada doğruluk oranı yükseltmek için none çıktısı kullanılır. Daire merkezi göz bebeğinin koordinatlarını vermektedir. Bu değerler bir değişkene atanıp Mod 2 de fare kontrolünde kullanılmaktadır. Kapalı gözde ve aydınlık olmayan bir ortamda göz bebeği bulunamayacağı için hata vermektedir. Bunun önüne geçmek için ise try except içerisine alınmaktadır ve iki göz için de tüm işlemler yapılmaktadır.

4.5 Kaş hareketi algılama

Bilgisayar kullanımında göz aşağıya baktığı esnada webcamin konumundan dolayı algılayacağı açıdan uzaklaşmakta ve göz bebeğinin yönü tespit edilememektedir. Göz aşağıya bakıldığı anlarda gerçekleşmesini istediğimiz işlemler için kaşların hareketlerinden yararlanılması uygun görüldü. Ayrıca kaşların hareketleri sayesinde kısayol işlemleri tanımlandı. Dlib kütüphanesinde kaşlar üzerindeki noktalar bulunmaktadır (17,18,19,20,21,22,23,24,25,26). En sol ve sağdaki (17,2) noktaların ortası midPoint fonksiyonu ile bulunur (kırmızı). Orta noktanın hareket ettiğini anlamak için sabit bir nokta gerekir. Yüzün belli bir bölümünün uzunluğu (0,1 16,15 arası uzunluk) burnun kaşa en yakın noktasına eklenerek sabit nokta (mavi) oluşturulur. Orta noktanın sabit noktanın altında veya üstünde olmasına göre işlemler yapılır



Şekil 5– Kaş hareketlerini algılamak için belirlenen noktalar

4.6 Kısayol İşlemleri

Yaptığımız testler sonucunda göz kırıldığında çıkan oran sol göz, sağ göz veya ikisinin de

ortalaması alındığında bir sayıyı geçmektedir. Eğer bir göz üzerinde göz bebeği bulunamıyor fakat diğer gözde bulunabiliyor ve diğer gözün kırpma oranı daha büyük ve ortalama kırpma oranını geçiyor ise o göz kapalı diğer göz açık anlamına gelmektedir. İki göz de bulunamıyor ve ortalama kırpma oranını geçiyor ise iki gözün de kapalı olduğu anlaşılmaktadır. Bu durumlardan yararlanarak aşağıdaki kısayollar programlanmıştır. Bu kısayollar her iki modda da çalışmaktadır.

4.7 Programı uyku moduna alma Kısayolu

İki göz kapalı olduğu durumda bir değişken artırılır geçen saniyeyi anlamak için anlık tarih datetime ile alınıp time ile saniyeye çevirilip listeye atılır ilk eleman ilk göz kırpma zamanına denk gelirken son eleman en son göz kırpma anına denk gelir eğer aralarındaki fark beş saniyeden küçük veya eşit ise ve göz kırpma sayısı ortalama beş saniye göz kırpma değerinden büyük veya eşit ise diğer çalışan if else bloklarını durdurmak için set dizisine eleman eklenir diğer if else bloklarına set uzunluğu sıfır olduğu zaman çalışır şeklinde yazacağız ki eleman eklendiği zamanki set uzunluğu bir olduğu zaman çalışmasınlar. Tekrardan kısa yolun çalışabilmesi için değişken ve liste sıfırlanır, playsound ile işitsel bildirimde “durduruldu” bulunulur. Eğer aralarındaki fark beş saniyeden büyük ise ve göz kırpma sayısı ortalama beş saniye göz kırpma değerinden küçük ise tekrardan çalışabilmesi için değişken ve liste sıfırlanır. Diğer kısayolların çalışma mantığı da benzer şekildedir. Programın devam etmesi için gözler kapalı beş saniye beklenir. Sol göz beş saniye kapalı olduğunda masaüstüne dönülür. Sağ göz beş saniye kapalı olduğunda ekran klavyesi açılır bu sayede kullanıcı bilgisayar ile etkileşime geçer. Gözler kapalı kaşlar beş saniye kalkık dosya gezini açılır. Sol göz kapalı sağ göz açık ise sol tıklama tam tersi durumda sağ tıklama işlemi gerçekleştirilir.

4.8 Mod (Seçenek) 1

Bu seçenekte baş hareketleri ile fare kontrolü sağlanır. Baş hareketi tespiti için burun izlenmektedir [30]. Fare hareketi için burun ucu [30] ile yüzün ortasının koordinatlarının farkına göre işlem yapılmalıdır. Yüzün köşeleri olarak [0,16,12] noktaları seçildi. Yeni noktalar kullanılarak şekil 8’de gösterilen kırmızı dikdörtgen çizdirilmektedir. Bu dikdörtgen oranlama ile yüzümüzün tamamını alacak şekilde çizdirilmekte ve ortalanmaktadır. Kullanıcının başını sabit tutmasına gerek kalmadan yanlışlıkla baş hareketleri sonucu istenmeyen fare hareketlerinin önüne geçilebilmesi için yüz alanın büyüklüğüne göre güvenli alan oluşturuldu. Bu güvenli alan yüzün ortasında küçük bir alandır. Burun noktası bu alandan dışarı çıktığı zaman fare hareket etmektedir. Oluşturulan güvenli bölge şekil 8 üzerinde mavi dikdörtgen ile gösterilmiştir.



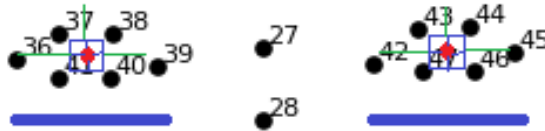
Şekil 8. Mod 1 yüz tespiti ve güvenli alan oluşturulması

Burunun x değeri mavi dikdörtgenin sol üst noktasının x değerinden küçük ise kullanıcı kafasını sola çevirip güvenli bölgeden çıkarmış ve fareyi sola çevirmek istiyor demektir. X değerleri arasındaki fark uygun görülen bir kat sayı ile çarpılmaktadır ve farenin anlık konumunun x değerinden çıkarılmakta bunun sonucunda fare sola doğru gitmeye başlamaktadır. Eğer burnun x değeri mavi dikdörtgenin sağ alt noktasının x değerinden büyük ise kullanıcı kafasını sağa çevirip

güvenli bölgeden çıkarmış olacaktır. Bunun sonucunda fare sağa hareket edecektir. X değerleri arasındaki fark kat sayı ile çarpılmakta ve farenin anlık konumuna göre x değerine eklenir bunun sonucunda fare sağa doğru gitmektedir. Burnun y değeri mavi dikdörtgenin sol üst noktasının y değerinden küçük ise kullanıcı kafasını yukarıya çevirmek suretiyle güvenli bölgeden dışarıya çıkarmıştır. Bunun sonucunda y'ler arasındaki fark bizim belirlediğimiz bir kat sayı ile çarpılmaktadır. Farenin anlık konumundan y değeri çıkarılarak farenin yukarıya doğru gitmesi sağlanır. Eğer burnun y değeri mavi dikdörtgenin sağ alt noktasının y değerinden büyük ise kullanıcı kafasını aşağıya eğmiş ve güvenli bölgeden çıkarmıştır. Bu durumda fare aşağıya doğru hareket etmelidir. Y değerleri arasındaki fark belirlenen katsayı ile çarpılır ve farenin anlık konumunun y değerine eklenir bunun sonucunda fare yukarıya doğru hareket eder. Kullanıcı katsayıyı arayüz üzerinde "Ayarlar" bölümünden artırıp azaltarak fare hareket hızını değiştirebilir. Tüm kısayollar ve tıklama işlemleri kafa hareket ettirirken yanlış tıklamaları önlemek amacıyla güvenli alanda çalıştırılabilir.

4.9 Mod (Seçenek) 2

Bu seçenekte gözün hareket ettiği yöne göre fare kontrolü sağlanır. Mod1 çalışma prensibinin göze uygulanmış halidir. Her göz hareketinde farenin hareket etmemesi için gözün büyüklüğüne göre güvenli bölge oluşturulur bu gözün ortasında küçük bir alandır buradaki göz hareketleri fare hareketine yansımaz. Güvenli bölge Şekil 10 üzerinde mavi dikdörtgen ile gösterilmektedir. Göz bebeklerinin x ve y değerleri ile göz ortasının x,y değerleri arasındaki farkın ortalaması alınır. Göz bebeklerinin güvenli alandan çıktığı yöne doğru kat sayı ile çarpılıp farenin konumunda işlemler yapılarak kullanıcın baktığı yöne doğru hareket ettirilir. Tüm kısayollar ve tıklama işlemleri göz hareket ettirirken yanlış tıklamaları önlemek amacıyla güvenli alanda çalıştırılabilir.



Şekil 10. Mod 2 için oluşturulan güvenli alan ve göz merkezi tespiti

4.10 Arayüz

Arayüz dört bölümden oluşmaktadır: kamera, ayarlar, kısayollar, nasıl kullanılır. Kamera bölümünde kullanıcıya webcamden işlenen görüntü gösterilir. Ayarlar bölümünde farenin hareket hızını değiştirebilir ve kısayolları kendine göre özelleştirebilir. Kısayollar ve nasıl kullanılır bölümünde unutulması durumu için programın çalışması hakkında bilgiler verilmektedir.

5. Yenilikçi (İnovatif) Yönü

Engelli bireylerin bilgisayar kullanması için günümüzde bulunan çözümler kullanıcıya kendine göre özelleştirme imkânı sunmayan bilgisayara ve bireyin üzerine takılan ek pahalı cihazlardır. Bireyin üstüne takılan cihazlar onu rahatsız edebileceği gibi stok sorunları ve fiyatları olsun ulaşılması çok zor ürünlerdir. Bizim projemizde ise sadece bilgisayar ve webcam gerekli olduğundan bu tür sorunlar teşkil etmemektedir.

Program orta kalitede bir webcam ile doğru ışık kullanıldığında yaklaşık %95 seviyelerinde doğruluk oranıyla çalışmaktadır. Programımız Python ile programlanarak Windows işletim sisteminde çalışabilecek şekilde tasarlanmıştır.

Projemizi denemelerimizde ilk kullanımlarda en çok karşılaştığımız problem kısayol kombinasyonlarını hatırlayamamamız oldu. Bu nedenle programımızın arayüzüne bir yardım düğmesi ekledik. Kullanıcılar bu yardımı kullanarak kısayolları kolaylıkla hatırlayabilmektedir.

Projemizde geliştirdiğimiz yazılım sayesinde program bilgisayar başlatıldığında otomatik açılmakta ve webcam görüntülerini işleyebilmektedir. Programımızdaki arayüz sayesinde kullanıcı

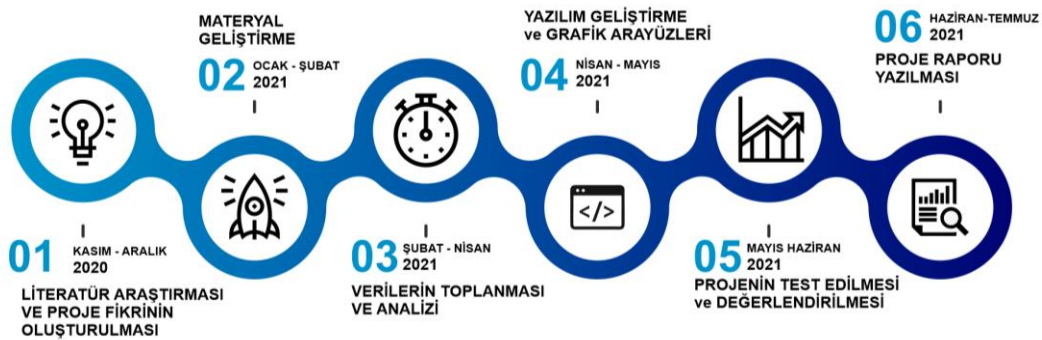
programı açarken belirli yönergeleri görüp takip ederek istediği modda açabilmektedir. İster baş hareketleri ve göz hareketleri isterse sadece göz hareketleri ile bilgisayarı kullanabilmektedir. Program birçok kişi üzerinde test edildi ve sorunsuz çalışmaktadır.

Teknoloji hayatımızın her alanında etkin olarak kullanılmaktadır. Her yaşta bireyler teknolojiyi kullanmaktadır. Özellikle internet üzerindeki içeriklerin artışı, birçok şeyin dijital hale gelmesi ile bilgisayar, cep telefonu ve tablet kullanımını her geçen gün arttırmaktadır. Bu cihazları kullanırken kollarımıza, ellerimize ve el parmaklarımıza ihtiyaç duyarız. Hayatımızın önemli bir süresini ayırdığımız bu faaliyet sırasında bu uzuvlarımızı kullanamamaktayız. Projemiz geliştirildiği ve başka platformlara da aktarıldığı düşünülürse bu uzuvlarımıza ihtiyaç duymadan çok daha kolay işlerimizi yerine getirebiliriz. Ellerimizi kullanarak birçok psikomotor becerileri eyleme dönüştürürken aynı zamanda bilgisayar ve diğer cihazları kullanabiliriz. Bu proje hayatımızdaki verimi büyük oranda attırırken hayatımızı çok daha eğlenceli hale getirecektir.

6. Uygulanabilirlik

Bu projeye bilgisayar ile olan iletişimimizi daha etkin hale getirdik. Özellikle engelli bireylerin başkalarına ihtiyaç duymadan bilgisayarları etkin bir şekilde kullanımına olanak sağlamaktayız. Denemelerimiz ve pratik çalışmalarımız sonucunda bu programı bir oyuna alışmak gibi zamanla daha da hızlı ve aktif bir şekilde kullanabileceğimizi gözlemledik. Kısıyollarda kullanılan algoritmalar zaman içinde bütünleşerek çok rahat kullanılabilen. Program orta kalitede bir webcam ile doğru ışık kullanıldığında yaklaşık %95 seviyelerinde doğruluk oranıyla çalışmaktadır. Programımız Python ile programlanarak Windows işletim sisteminde çalışabilecek şekilde tasarlandı. Çünkü Python programlama dili sürekli geliştirilen, açık kaynak kodlu, esnek ve işlevsel bir yazılımdır. Program basit işlemler sonucunda Linux, Unix, Mac işletim sistemleri için de derlenebilmektedir. Kısacası Windows işletim sistemli, webcam bulunduran bilgisayar ve programa alışmaya hevesi olan her birey programı kullanabilir.

7. Tahmini Maliyet ve Proje Zaman Planlaması



Projemizin uygulanabilmesi için bilgisayar ve bilgisayara bağlı bir webcam yeterli olup maliyeti oldukça düşüktür.

8. Proje Fikrinin Hedef Kitle (Kullanıcılar):

Yazılımımız engelli bireyler için tasarlanmıştır. . Kas ve iskelet sisteminde yetersizlik, eksiklik ve fonksiyon kaybı olanlar, kol, parmak ve omurgalarında kısıklık, eksiklik, fazlalık, hareket kısıtlılığı, şekil bozukluğu, kas güçsüzlüğü, kemik hastalığı olanlar, felçliler, serebral palsi, spastikler ve spina bifida olanlar yüz kaslarını kullanabilen bireyler çağımızın en büyük teknolojik gelişmelerinden olan bilgisayarlar kullanamamaktadır. Geliştirmiş olduğumuz bu proje ile bu bireyler de rahatlıkla bilgisayara komutlar gönderebilmesi sağlanacaktır.

9. Riskler

Gelişen teknolojiler ile görüntüler artık daha kolay işlenebilmektedir. En basit bilgisayar bile artık işlemleri çok hızlı bir şekilde yapabilmektedir. Fakat programımız kaliteli bir kamera ve doğru ışık (Yüzün aydınlık ve karartmaya sebep olacak baş arkasında güçlü bir ışık kaynağının olmaması gerekmektedir. Aksi durumda yüz karanlıkta kalacak programımız çok istikrarlı çalışmayacaktır veya çalışmayacaktır.) ile kullanıldığında doğruluk oranı %95 seviyelerine çekilebilir. Projemizin en büyük risklerinden birisi ışığın yeterli olmamasıdır. Ortamdaki ışık yeterli seviyede değilse kameradan gelen blurlu görüntü tam olarak işlenememektedir. İkinci risk ise programa alışma süresidir. Programa alışma zamanı geciktiği takdirde kullanım isteği azalabilir.

10. Kaynaklar

Arı, F. ve Durna, Y.(2015a). Labview ile Gerçek Zamanlı Göz bebeği Kornea Yansıması Takibi. 23.Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 16-19 Mayıs, Malatya, 2182-2185. Ayvaz, U., Çoban, A., Gürüler, H., Peker, M. (2016). Python Dilinin Özellikleri, Programlama Eğitiminde ve Yazılım Dünyasındaki Yeri Bilişim Sistemleri Mühendisliği Bölümü, Muğla Drewes, H. (2010). Eye Gaze Tracking for Human Computer Interaction. Dissertation, Munich: Ludwig-Maximilians.

Hough, P. (1962). Method and means for recognizing complex patterns. US Patent, 3069654. Ghani Usman, M., Chaudry, S., Sohail, M., Geelani, M.N. (2013). GazePointer: A Real Time Mouse Pointer Control Implementation Based On Eye Gaze Tracking

Sreedevi, J., Reddy, M.S., Satyanarayana, B. (2020) Eyeball Movement based Cursor Control using Raspberry Pi and OpenCV

Sohail, M., Geelani, M.N., Ghani, M.U., Chaudhry, S., GazePointer: A Real-Time Mouse Pointer Control Implementation Based on Eye Gaze Tracking

Eldem, A., Eldem, H. , Palalı A. BEÜ Fen Bilimleri Dergisi 6(2), 44-48, 2017

OpenCV, Hough Circle Transform Tutorial, Erişim Tarihi : 10.07.2020,

https://docs.opencv.org/3.4/d4/d70/tutorial_hough_circle.html

OpenCV, JavaScript Hough Circle Transform Tutorial, Erişim Tarihi : 10.07.2020,

https://docs.opencv.org/master/d3/de5/tutorial_js_houghcircles.html

OpenCV, Python Hough Circle Transform Tutorial, Erişim Tarihi : 10.07.2020,

https://docs.opencv.org/master/da/d53/tutorial_py_houghcircles.html Erişim Tarihi : 10.07.2020,

<https://opencv.org/> Erişim Tarihi : 10.07.2020,

<http://dlib.net/> PyAutoGUI's documentation, Erişim Tarihi : 10.07.2020,

<https://pyautogui.readthedocs.io/en/latest/> Erişim Tarihi : 10.07.2020,

<https://numpy.org/> PyQt5 Reference Guide, Erişim Tarihi : 01.07.2020,

<https://www.riverbankcomputing.com/static/Docs/PyQt5/>

Python datetime Library Documentation, Erişim Tarihi : 12.07.2020,

<https://docs.python.org/3/library/datetime.html>

Python time Library Documentation, Erişim Tarihi : 15.07.2020, <https://docs.python.org/3/library/time.html>

Gaze controlled keyboard OpenCv, Erişim Tarihi : 05.08.2020,

<https://pysource.com/gaze-controlled-keyboard-opencv/> International Journal for Research in Applied Science & Engineering Technology (IJRASET), Erişim Tarihi : 05.09.2020,

<https://www.ijraset.com/files/serve.php?FID=15922> OpenCV, Python Image Thresholding Tutorial, Erişim Tarihi : 18.07.2020,

https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

Saleem, S. M. T. , Fareed, S. ,Bibi, F. , Khan, A. ,Gohar, S. , Ashraf, H. H. , IMouse: Eyes Gesture Control System, Erişim Tarihi:21.10.2020, [https://thesai.org/Downloads/Volume9No9/Paper_68-](https://thesai.org/Downloads/Volume9No9/Paper_68-IMouseEyes_Gesture_Control_System.pdf)

[IMouseEyes_Gesture_Control_System.pdf](https://thesai.org/Downloads/Volume9No9/Paper_68-IMouseEyes_Gesture_Control_System.pdf)