

KARABÜK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



SENIOR PROJECT DERSİ

PROJE RAPORU

**PYTHON PROGRAMLAMA DİLİNDE GÖRÜNTÜ
İŞLEME YÖNTEMİYLE TEMASSIZ BİLGİSAYAR
KONTROLÜ**

Proje Öğrencisi:

1810205092 – Abdullah Furkan YEĞİN

Öğretim Üyesi:

Dr.Öğr.Üyesi Burhan SELÇUK

2020

İÇİNDEKİLER

BÖLÜM 1: PROJENİN AMACI

BÖLÜM 2: GÖRÜNTÜ İŞLEME NEDİR?

BÖLÜM 3: GÖRÜNTÜ İŞLEMENİN KULLANILDIĞI ALANLAR

BÖLÜM 4: GÖRÜNTÜ İŞLEME ADIMLARI

**BÖLÜM 5: GÖRÜNTÜ İŞLEME GERÇEKLEŞTİRİRKEN KULLANILACAK
UYGULAMALAR VE KÜTÜPHANELER**

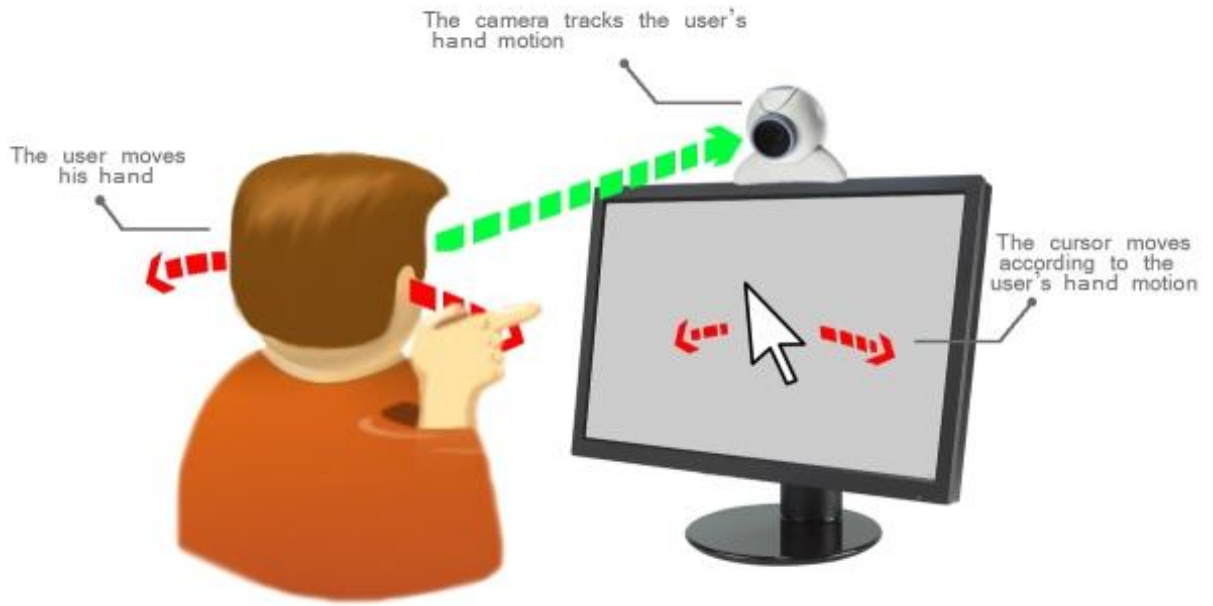
BÖLÜM 6: GÖRÜNTÜ İŞLEMENİN GERÇEKLEŞTİRİLMESİ

**BÖLÜM 7: OLUŞTURULAN PROJE İLE KULLANILMAK ÜZERE UNITY
OYUN MOTORU İLE ARAYÜZ TASARIMI**

BÖLÜM 8: SONUÇ

BÖLÜM 1: PROJENİN AMACI

Bu projede amaç klavye, fare veya başka bir giriş birimi kullanılmadan yalnızca kamera kullanılarak bilgisayar üzerinde istenilen işlemleri gerçekleştirmektir. Bu amaçla çeşitli görüntü işleme teknikleri kullanılarak kamera üzerinden alınan veriler işlenmiş ve istenilen komutların bilgisayar tarafından anlaşılıp uygulanması sağlanmıştır.

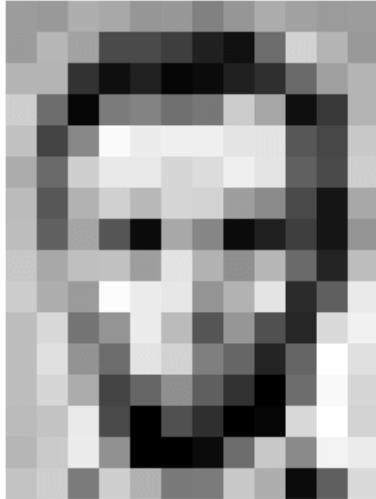


Araştırmalarım sonucu karşıma çıkan en yaygın 3 seçenekten birini tercih etmem gerekti. Bunlar Matlab, C++ ile OpenCV ve Python ile OpenCV. Bu üç seçenekten kaynak fazlalığı, kullanım kolaylığı ve görüntüleme teknikleri ve makine öğrenmesi üzerine özel frameworklerin bulunması ve Python'ın programlama dilleri arasındaki popülerliğinden dolayı Python ile OpenCV seçeneğini tercih ettim.

BÖLÜM 2: GÖRÜNTÜ İŞLEME NEDİR?

Günümüzde teknoloji, hayatımızın her alanında karşımıza çıkan ve bizi her defasında şaşırtmayı başaran yenilikler ile olmazsa olmazımız haline gelmektedir. Geçmişte ancak odalar dolusu makinenin yapabildiği işleri şimdi cebimize sığan cihazlarla gerçekleştirebiliyoruz. Bizlerden kilometrelerce uzakta bulunan insanlarla yüz yüzeymiş gibi konuşabiliyoruz. Teknolojinin gelişmesi bu ve bunun gibi birçok alanda bizlere faydası olmaktadır. Bu konudaki bir diğer gelişme görüntü işleme tekniğidir. Dünya çapında meşhur birçok şirket bu teknolojiyi kullanır ve akademik olarak bilgisayar bilimlerde araştırma konusudur. Ancak en önemlisi ülkemizin çok başarılı olduğu savunma sanayisinde hayati öneme sahiptir ve her geçen gün gelişen bir teknoloji alanıdır.[1]

Görüntü işleme görüntü elde etme veya mevcut görüntü verisini çeşitli algoritmalar aracılığıyla analiz edip bir sonuç çıkarma yöntemidir. Bu yöntemde görüntü verileri bir bütün olarak değil parça parça analiz edilir. Görüntü üzerindeki en küçük noktaya piksel adı verilir. Pikseller içerisinde sahip oldukları renklerin matrisler halinde sayısal değerlerini taşırlar. Görüntü işleme de bu pikseller üzerindeki sayısal değerler ile işlem yapılmasıdır.(Şekil2.1)[2]



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

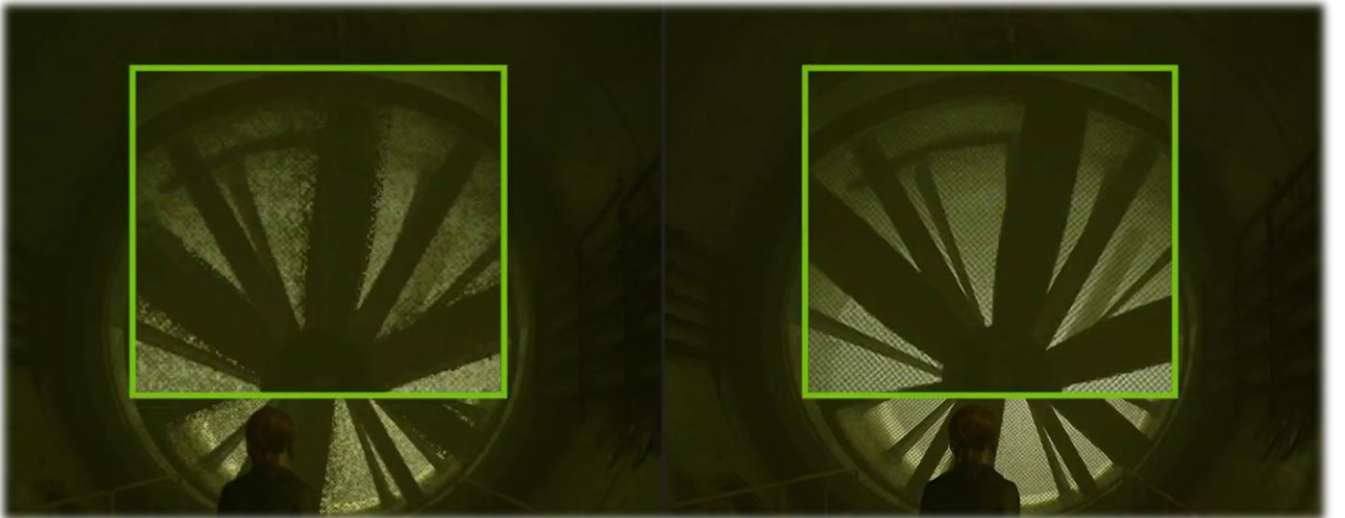
157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	105	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	85	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Şekil 2.1.

BÖLÜM 3: GÖRÜNTÜ İŞLEME HANGİ ALANLARDA KULLANILIR?

Görüntü işleme tekniği günümüzde farklı farklı alanlarda farklı farklı şekillerde karşımıza çıkmaktadır. Bunlardan bazıları plaka tanıma, yüz tanıma, videoların işlenmesi, eski, fotoğrafların ve videoların upscale edilerek netleştirilmesi, yani sahip olduğu gürültülerden kurtulması, tıbbi olarak hastanelerde kullanılan görüntüleme sistemlerinde, güvenlik alanında binaların girişlerinde bulunan XRAY cihazlarında ve araştırma yapılırken gözle görülmeyen ışınların tespitinde, otonom cihazların yoldaki çizgileri takip edebilmesi, etrafındaki araçları ve nesneleri tanıyabilmesi, trafik ışıklarını algılayıp hız sınırını gösteren tabelaları görüp algılayabilmesi için görüntü işleme yönteminin kullanılması sadece kullanım alanlarının birkaçıdır.[3]

Upscaling görüntü işleme metodu sayesinde düşük çözünürlüklü fotoğraf ve videoların kalitesi arttırılabilmektedir. Üstelik geçmişte görüntü işleme sadece basılı fotoğraflar üzerinde gerçekleştiriliyorken, günümüzde gelişen teknoloji sayesinde gerçek zamanlı olarak da çalışabilmektedir. Bu konudaki en güncel teknolojilerden birisi NVIDIA şirketinin geliştirmiş olduğu DLSS teknolojisidir. Bu teknoloji sayesinde uygulamalar, özellikle oyunlar, düşük çözünürlüğe sahip olsalar dahi görüntü işleme ve yapay zeka kullanılarak anlık olarak görüntü iyileştirilmesi yapılabilmektedir. Bu sayede donanım üzerine düşen işlem yükü azaltılarak daha akıcı ve daha kaliteli görüntüler almak mümkün olmaktadır. (Şekil 3.1.)

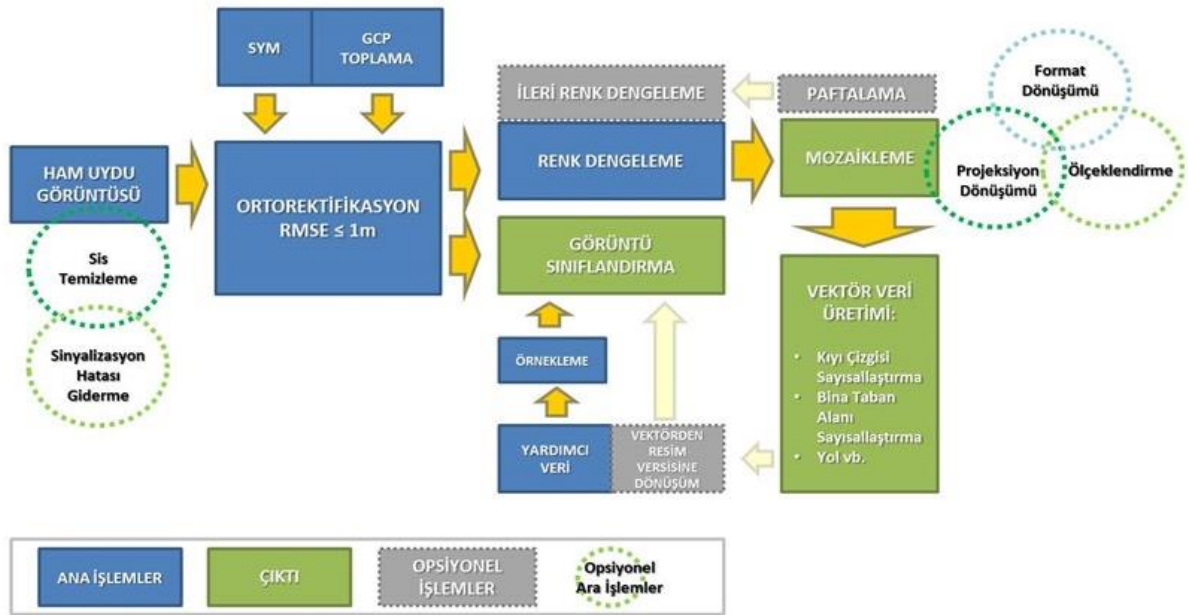


Şekil 3.1.NVIDIA DLSS Görüntü İşleme ve Yapay Zeka Uygulama Örneği [4]

BÖLÜM 4: GÖRÜNTÜ İŞLEME ADIMLARI NELERDİR?

Görüntü işleme temel olarak üç adımda incelenir:

- 1.Çeşitli araçlar ile görüntünün aktarımı
- 2.Görüntü analizi
- 3.Analiz edilip işlenen verinin sonucunun alınması



Şekil 4.1. Görüntü İşleme İş Akış Şeması [5]

Görüntü işleme analog ve dijital olarak ikiye ayrılmaktadır. Analog yöntemde sinyal dalgalarıyla veri alımı gerçekleştirilir. Ancak bu yöntem eski olduğundan uygulanması pahalı ve gerçekleştirilmesi yavaştır. Verilerin dijital olarak elde edilip yorumlanması daha tutarlı ve daha masrafsız olduğundan genel olarak dijital yöntem tercih edilmektedir.

BÖLÜM 5.1: GÖRÜNTÜ İŞLEME GERÇEKLEŞTİRİRKEN KULLANILACAK UYGULAMALAR

1.PYTHON

Görüntü işleme için birçok farklı programlama dili mevcuttur. Bunların başında C++, C#, Java, Python ve MATLAB gelmektedir. Bu diller arasından tercih etmiş olduğum programlama dili kolay kullanımı, kaynak çeşitliliği ve popülerliği nedeniyle Python olmuştur.



Oldukça geniş bir kullanım alanı olan Python en çok kolay kullanım özelliğiyle dikkat çekmektedir. Birçok programlama dilinde bulunan ayrıntılı syntax kurallarının bu dilde oldukça fazla hafifletilmesi kullanıcı sayısının artmasındaki en önemli etkenlerden biridir. Yapısındaki bu kolaylık sayesinde program yazmanın yanında yazılan programın kodlarını okumakta oldukça kolay hale gelmiştir. Bir diğer özelliği de desteklediği kütüphanelerin sayısının çok olmasıdır. Modüler yani üzerine kurulum ve kaldırma işlemlerinin kolaylığı sayesinde güncel tutulup daha performanslı çalışılması sağlanmaktadır. Dinamik bir altyapıya sahip olan bu dil C, C++ ve Java ile bağlantı kurmayı destekler.[6]

PYPL (Popularity of Programming Language Index) sitesine göre Python 2019 Ekim itibariyle bir önceki yıla göre en çok büyüyen ve en yaygın kullanılan programlama dilidir. (Şekil 5.1)[7]

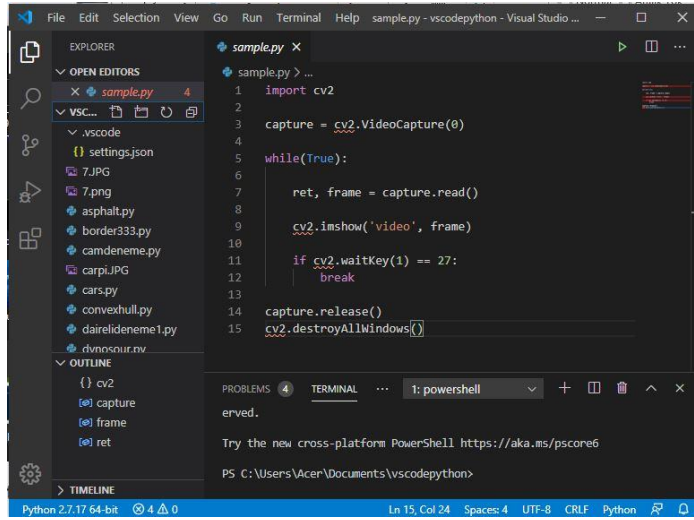
Worldwide, Oct 2019 compared to a year ago:				
Rank	Change	Language	Share	Trend
1		Python	29.49 %	+4.5 %
2		Java	19.57 %	-2.4 %
3		Javascript	8.4 %	+0.1 %
4		C#	7.35 %	-0.4 %
5		PHP	6.34 %	-1.2 %
6		C/C++	5.87 %	-0.4 %
7		R	3.82 %	-0.2 %

Şekil 5.1



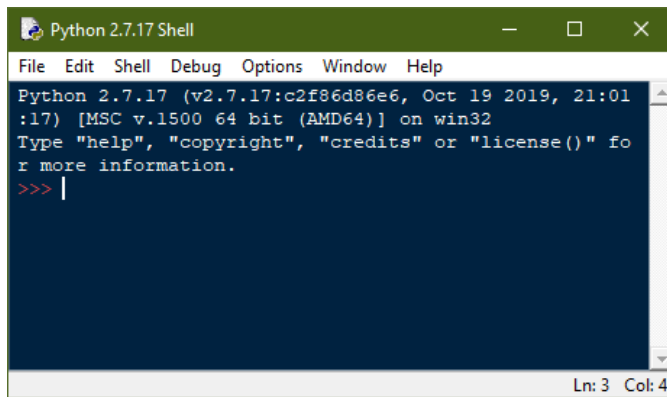
2.VISUAL STUDIO CODE

Microsoft'un geliřtirdiđi bir geliřtirme ortamı (IDE) olan Visual Studio Code yine Microsoft'un geliřtirdiđi bir geliřtirme programı olan Visual Studio'nun basitleřtirilmiř halidir denilebilir. Bu IDE; Python, C++, C#, JavaScript, Java, PHP gibi programlama dillerini desteklemektedir. Kullanımı pratik olan bu IDE'de harici paket kurulumu da oldukça kolaydır. Windows, Linux ve macOS iřletim sistemlerinde çalıřabilmektedir. [8]



3.PYTHON IDE

Sahip olduđu arayüzü Visual Studio Code kadar gösteriřli olmasa da sade yapısı sayesinde küçük iřleri hızlı bir řekilde yapmamı sađladı. Projenin büyük bir kısmında Visual Studio Code'u kullanmıř olsam da Python kurulduđuında birlikte gelen ve projenin bir kısmında kullandıđım Python IDE'yi de kullandıđımı belirtmek istedim.



BÖLÜM 5.2: GÖRÜNTÜ İŞLEME GERÇEKLEŞTİRİRKEN KULLANILACAK KÜTÜPHANELER

1.OPENCV

OpenCV (Open Source Computer Vision Library) gerçek zamanlı görüntüler üzerinden işlemler yapmaya odaklanan, kullanımı kolay, farklı programlama dillerini destekleyen, açık kaynak kodlu bir kütüphanedir. Bu kütüphanenin içinde sadece görüntü işlemeye yönelik birçok algoritma bulunmaktadır. Microsoft, IBM, Google gibi firmalar tarafından da kullanılan bir kütüphanedir. Bu kütüphane birçok kurum ve kuruluş tarafından çeşitli amaçlarla kullanılmaktadır. Örneğin insan tespiti ile bina güvenliği, plaka tespiti ile trafik kontrolü, yüz tanıma ile kişi tespiti gibi olaylarda bu kütüphane yaygın olarak tercih edilmektedir. [8]



2.NumPy

Bu kütüphane kamera üzerinden elde edilecek görüntülerin piksellerinin matematiksel tabanda kaydedilerek üzerinde işlem yapılmasını sağlayacaktır.

NumPy (Numerical Python) genel olarak bilimsel hesaplamalarda kullanılan bir matematik kütüphanesidir. İçerisinde bulunan diziler sayesinde Python'un orijinal haline göre daha hızlı ve kullanımı daha kolay olmaktadır. NumPy'nin en önemli kuralı oluşturulan dizinler içerisinde aynı tür elemanlar bulundurulmalıdır yani veri tiplerinin aynı olması gerekmektedir.[10]



3-4.PYNPUT ve WXPYTHON

Diğer iki kütüphane pynput ve wxpython kütüphaneleridir. Pynput kütüphanesi arabirim aygıtlarını simüle etmeyi, wxpython kütüphanesi ise ekran boyutunun hesaplamalarda kullanılması için programa aktarımını sağladı.

BÖLÜM 6.1: PYTHON ÜZERİNE GEREKLİ KÜTÜPHANELERİNİN KURULMASI VE ÇAĞIRILMASI

NumPy ve OpenCV kütüphaneleri Python üzerine komut istemi üzerinden yüklendi. Bunun için kurulum paketleri indirip kurmaya yarayan bir paket yönetim sistemi olan 'pip' (Package Installer for Python) kullanıldı.

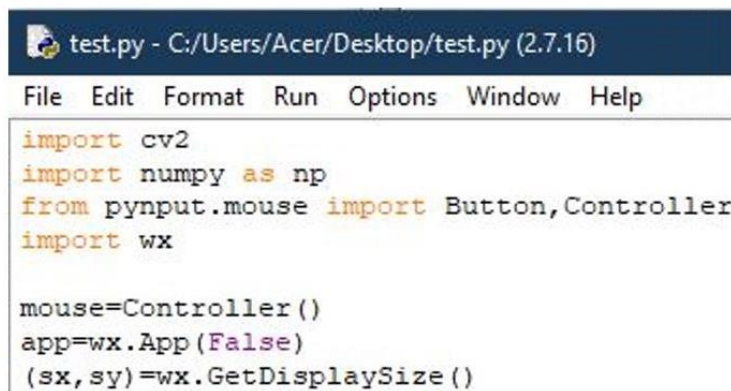
Öncelikle Python'ın disk üzerinde kurulu olduğu klasöre komut istemi üzerinden ulaşarak 'Scripts' klasörüne giriş yapıldı. Daha sonra kurulacak kütüphaneler pip vasıtasıyla yüklendi.

```
C:\Python27\Scripts>pip install numpy  
C:\Python27\Scripts>pip install opencv-python
```

Daha sonra Python IDE ile giriş aygıtları arasında bağlantıyı sağlayabilmek için Python'a ait 'pynput' kütüphanesi kullanıldı. Bu kütüphane sayesinde klavye ve fare girişleri yapılabilmekte veya yapılan girişlerin görüntülenebilmesi mümkün olmaktadır. Bu kütüphaneyi kullanabilmek için konsol üzerinden pip komutu ile yükleme işleminin gerçekleştirildi.

```
C:\Python27\Scripts>pip install wxpython  
C:\Python27\Scripts>pip install pynput
```

Kurulan kütüphaneler 'import' komutuyla IDE içerisinde çağırılarak üzerinde program yazmaya hazır hale getirildi.



```
test.py - C:/Users/Acer/Desktop/test.py (2.7.16)  
File Edit Format Run Options Window Help  
import cv2  
import numpy as np  
from pynput.mouse import Button, Controller  
import wx  
  
mouse=Controller()  
app=wx.App(False)  
(sx,sy)=wx.GetDisplaySize()
```

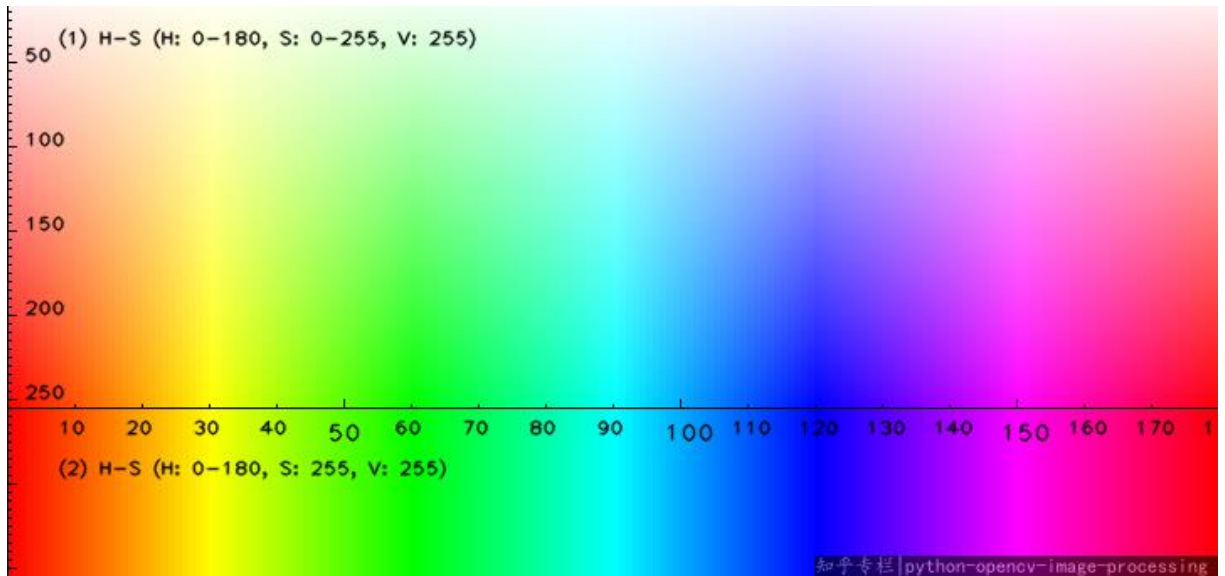
BÖLÜM 6.2: HSV RENK MODELİ KULLANILARAK İSTENİLEN RENK ARALIĞINA SAHİP NESNELERİN TESPİTİ

HSV renk modeli, renkleri RGB renk modeli gibi kırmızı, yeşil ve mavi renklerin birleşimi ile oluşturmak yerine her renge ait bir tanım bulundurmaktadır. Bu tanımlar üçe ayrılır:

1.kısım olan ‘Hue’ kısmı rengin tonunu belirlemek için kullanılır. OpenCV içerisinde kullanılan HSV renk modelinde Hue değeri 0 ile 180 arasında değer alır. Diğer uygulamalarda maksimum renk değeri 360’a kadar çıkabilmektedir.

2.kısım olan ‘Saturation’ bölümünde rengin yoğunluğu belirlenir. 0 ile 255 arasında değer alır. 255 değeri rengin çok yoğun olduğunu ifade eder.

3.kısım olan ‘Value’ değeri parlaklığı belirtir. Yine 0 ile 255 arasında değer alır. 0 tamamen siyah, 255 tamamen beyaz demektir.



HSV renk modeline göre düzenlenmiş renk tablosu

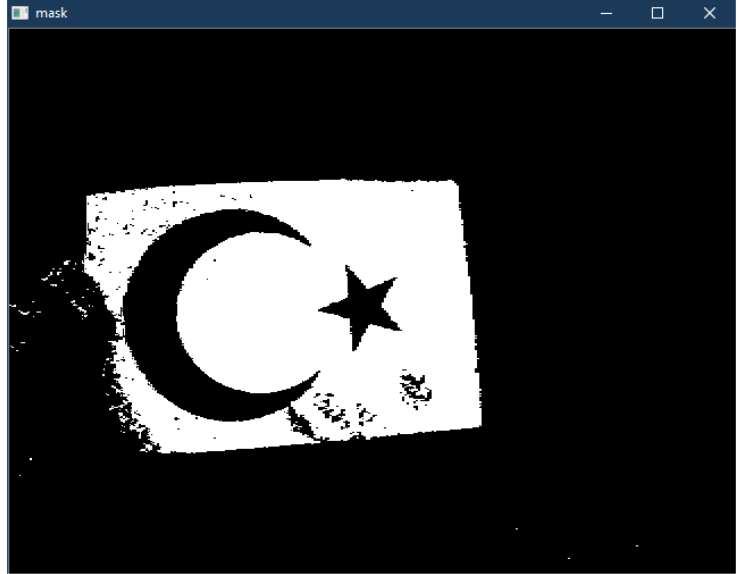
Python üzerinden istenilen renklere sahip cisimler belirlenirken direk renk değeri girilmek yerine belirli bir renk aralığı girilmesi tercih edilmektedir. Bu şekilde başarılı olma olasılığı artmaktadır. Bunun için program içerisinde istenilen renk aralığına dair alt ve üst limitler girilir.

Örnekte kırmızı renk için 'Hue' değeri 0 ile 10 arasında, 'Saturation' değeri ise 100 ile 255 arasında, 'Value' değeri ise 100 ile 255 arasında tutulmuştur.

```
lowerLimit=np.array([0,100,100])  
upperLimit=np.array([10,255,255])
```

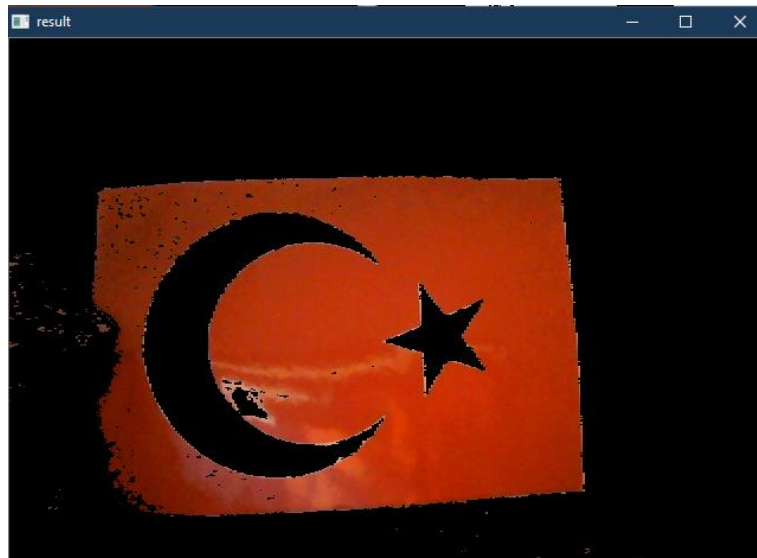
OpenCV fonksiyonlarından 'inRange' ile bu aralıktaki renk değerleri tespit edilmiştir.

```
mask=cv2.inRange(hsv,lowerLimit,upperLimit)
```



Daha sonra iki görüntünün dizi bitlerini 'bitwiseand' işlemine tabi tuttuğumuzda sadece istediğimiz renk aralığındaki cisimlerin orijinal hallerini görebilmekteyiz.

```
result=cv2.bitwise_and(frame,frame,mask=mask)
```



Ardından istenilirse sonuç üzerinde de tekrar deęişiklik yapılabilir. Örnek olarak median, bilateral ve gaussian efektleri çıktı üzerine uygulanmıştır. Bu algoritmalar OpenCV'ye ait olan ve kendi kütüphanesi üzerinden çağırılan algoritmalarlardır. Bu algoritmalar sayesinde görüntü üzerinde oluşan gürültüler azaltılabilmektedir. Hangi algoritmanın tercih edileceęi gerçekleştirilen uygulamaya göre deęişiklik gösterebilmektedir.

```
blur=cv2.GaussianBlur(result,(15,15),0)  
median=cv2.medianBlur(result,15)  
bilateral=cv2.bilateralFilter(result,15,75,75)
```

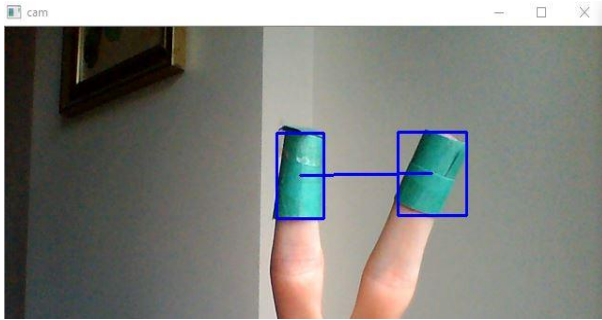
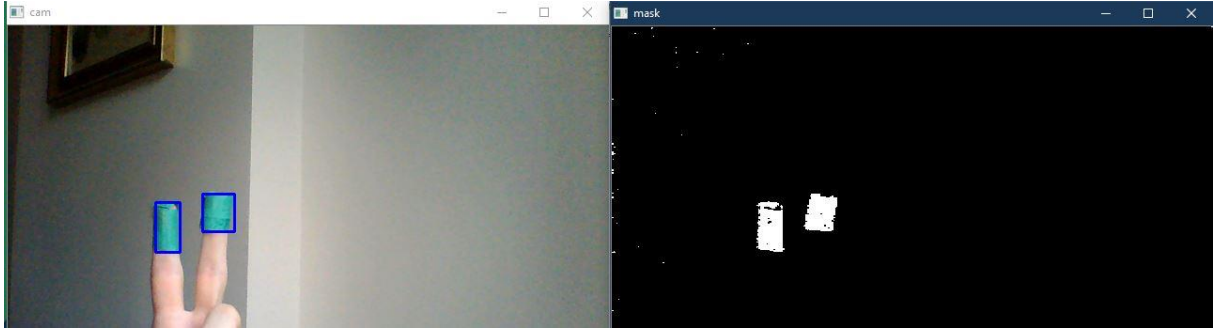


BÖLÜM 6.3: İSTENİLEN RENK ARALIĞINA SAHİP İKİ CİSİM İLE FARE KONTROLÜ

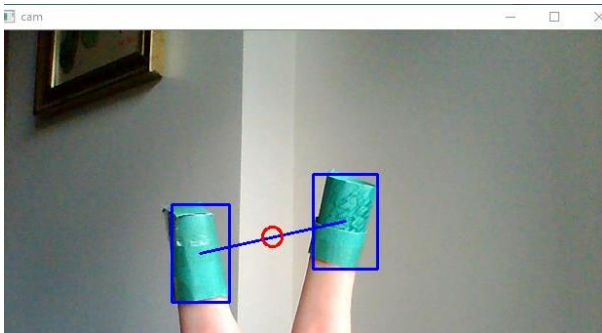
Ekranda istenen renk aralığına sahip iki nesne belirdiğinde bu nesnenin sınırlarının tespit edilebilmesi için findContours fonksiyonu kullanıldı. Ardından bu sınırların belirginleştirilmesi için rectangle fonksiyonuyla dikdörtgen içine alındı.

```
_,conts,h = cv2.findContours(maskFinal.copy(),cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
if(len(conts)==2):

    x1,y1,w1,h1=cv2.boundingRect(conts[0])
    x2,y2,w2,h2=cv2.boundingRect(conts[1])
    cv2.rectangle(img,(x1,y1),(x1+w1,y1+h1),(255,0,0),2)
    cv2.rectangle(img,(x2,y2),(x2+w2,y2+h2),(255,0,0),2)
```



Uygulamada amaç iki nesnenin birbirine yakınlığı kullanılarak tıklama yapılması olduğundan iki nesnenin arasında bir çizgi oluşturularak bağlantı kuruldu.



Daha sonra imleç hareketlerinde merkez olarak kullanılmak üzere aralarındaki mesafe hesaplanarak orta noktasına daire eklendi.

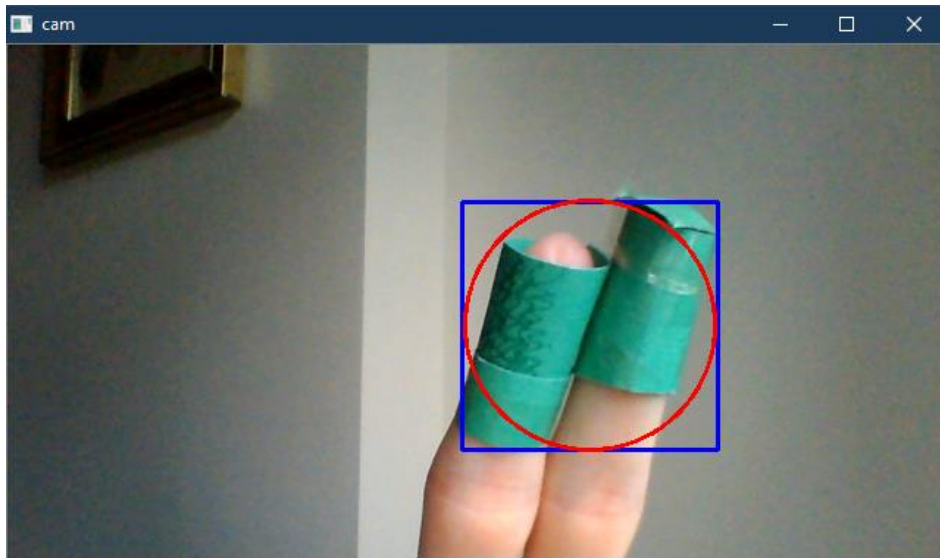
Fare imlecini hareket ettirmek için daha önce 'GetDisplaySize' ile elde edilen boyutu kamera çözünürlüğüne bölüp bu değeri pynput özelliği olan mouse.position'a atayarak farenin konumunun değiştirilmesi sağlandı.

Fare tıklamasını simule edebilmek için bir koşul atanması gerekmekte. Bu koşul iki nesnenin arasındaki uzaklığın ölçülerek bir alt sınır belirlenmesi ve aralarındaki mesafe bu limitin altına düştüğünde fare tıklaması simule edilebilmelidir. Gerekli mesafenin ne kadar olduğu hesaplanarak if koşulu eklendi. Bu koşulların while döngüsü içine alınarak sürekli sorgulanması sağlandı.

```
while True:

    if (abs((w*h-openw*openh)*100/(w*h))<22):
        pinchFlag=1
        mouse.press(Button.left)
        openx,openy,openw,openh=(0,0,0,0)
```

Koşullar sağlanırsa fare tıklaması gerçekleşir. Bu durumu belli etmek için tıklama gerçekleştiğinde büyük bir daire ile tıklama yapıldığı gösterildi.

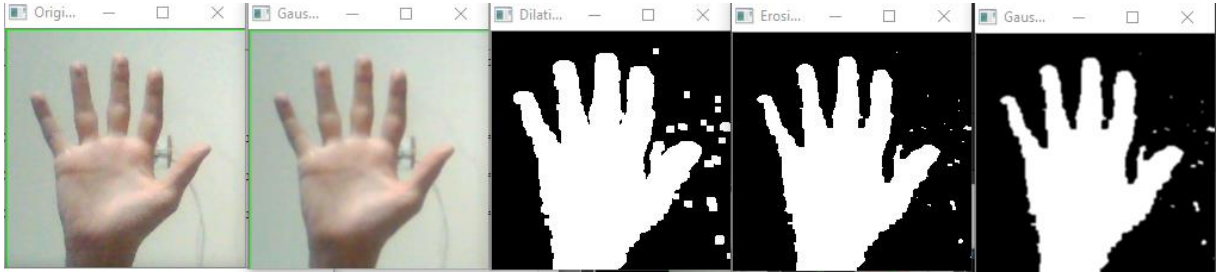


İki nesne birbirinden uzaklaştığında ise if koşulu false değerini göndererek fare release komutuyla tıklama sona erdirilip serbest bırakılır.

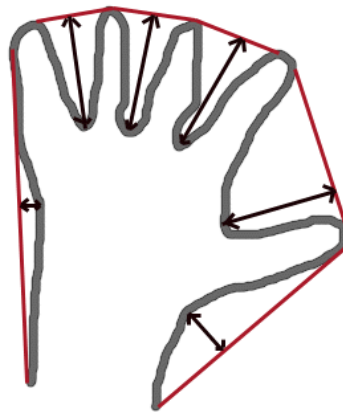
```
mouse.release(Button.left)
```

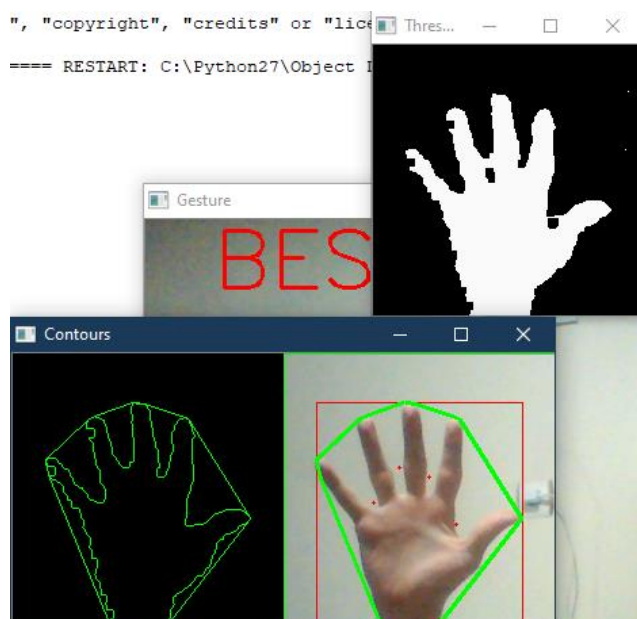
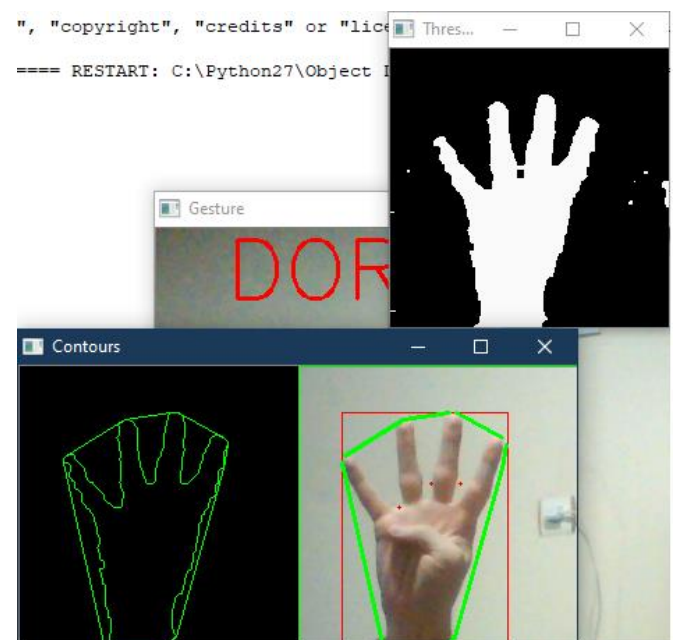
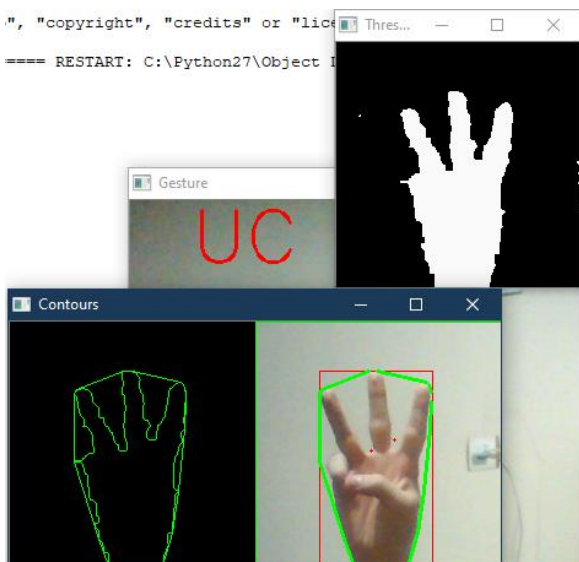
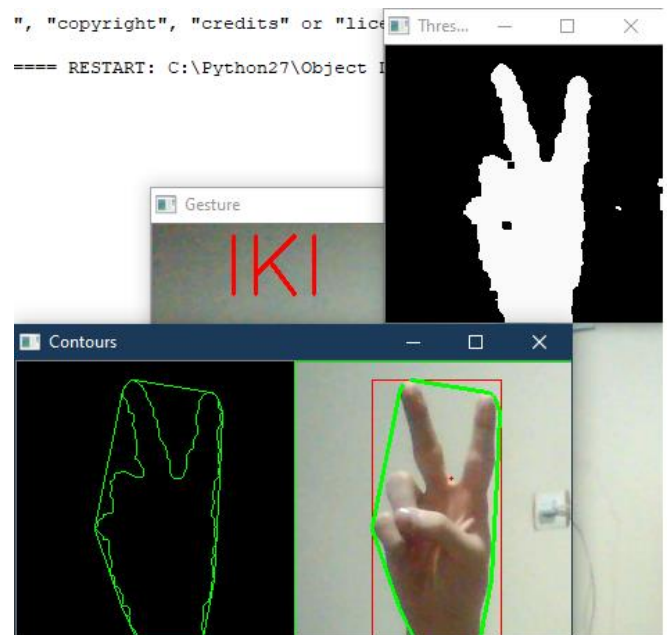
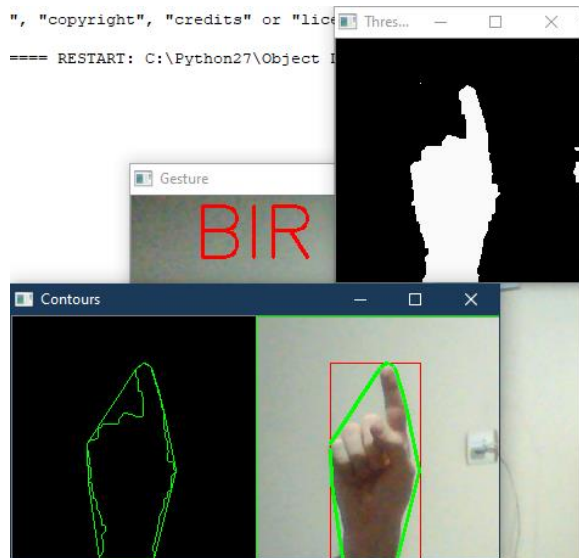
BÖLÜM 6.4 : UYGUN IŞIK KOŞULLARI ALTINDA ELİN TESPİT EDİLMESİ

Daha önce kullanılan iki noktanın birbirine uzaklığına göre oluşturulan konsept yerine elin tamamını algılayan bir konseptte geçiş yapıldı. Kameranin eli tanınması için ten rengine ait HSV renk değerleri programa eklendi. Ancak gündüz saatlerinde ortamdaki ışıklandırmanın değişiklik göstermesinden dolayı istenilen sonuç elde edilemedi. Bundan dolayı akşam sabit ışıklandırma ile el tespiti yapıldı. El tespiti yapıldıktan sonra oluşan gürültüyü azaltmak için çeşitli filtreler kullanıldı.



Daha sonra ele fonksiyon atayabilmek için elin mevcut şeklini tanıyıp ona göre çıktı verebilen kod yazıldı. Böylece parmaklar arasındaki mesafe kullanılarak bilgisayarın el hareketlerine tepki verebilmesi sağlanmış oldu.[12]





Bir sonraki aşamada her bir durum için farklı eylemler atandı. Örneğin 5 durumunda ekranın ve elin ortasında birer nokta oluşturuldu. Bu iki nokta arasındaki konum farkı ile (sol-üstünde, sol-altında, sağ-üstünde, sağ-altında bulunması) fare imlecinin hareketi kesintili de olsa sağlandı. Ardından 2 durumuna fare sol tıklama komutu atandı ve yaklaşık olarak istenilen konuma getirilen fare imleci ile tıklama gerçekleştirildi.

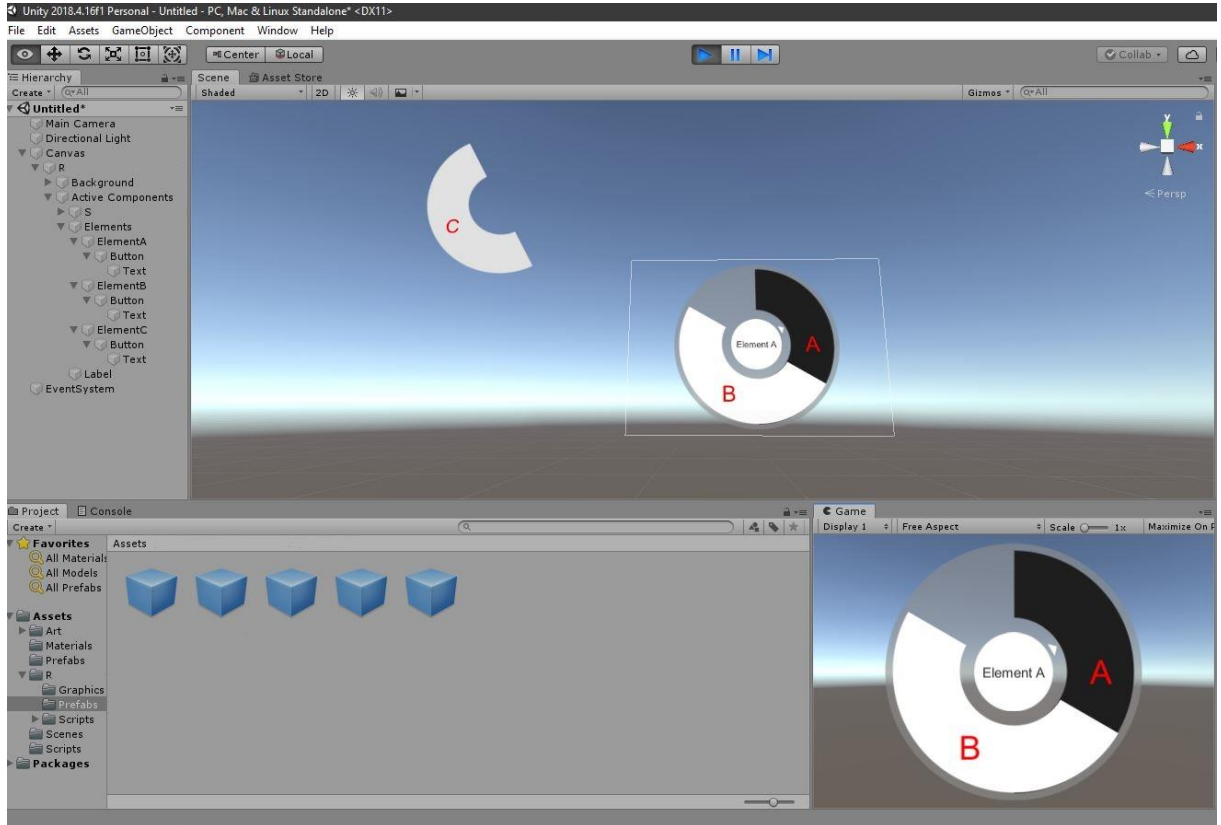
Farklı bir uygulamada ise 1-2-3-4 durumlarına yön tuşları atanarak sanal ortamda temassız olarak araba sürüldü. Ancak ani refleks gerektiren durumlarda bu atamanın pek de başarılı olmadığı gözlemlendi. Ayrıca zaman zaman eli tanıyamama gibi sorunlar ortaya çıktı.

Sonuç olarak farklı ışık koşullarında farklı sonuçlar elde edildiğinden ten renginin bilgisayarın rahatça algılayabileceği bir renk olmadığı ve kameranin farklı ışık koşullarında mevcut renkleri, olduğundan farklı renkte veya tonda algılayıp istenmeyen komutları istemsiz bir şekilde çalıştırabildiğinden kameralar tarafından en net algılanabilen renk olan yeşil rengin bir önceki yöntemle kullanılmaya devam edilmesine karar verildi.

BÖLÜM 7: OLUŞTURULAN PROJE İLE KULLANILMAK ÜZERE UNITY OYUN MOTORU İLE ARAYÜZ TASARIMI

Fare hareketlerinin başarılı bir şekilde gerçekleştirilmesine rağmen çözünürlüğü yüksek olan bilgisayarlarda kullanımın zor olacağı ve bilgisayar kullanım bilgisi yetersiz olan kişilerin de bilgisayarı kullanabilmeleri için bir arayüz tasarlanması gerektiği düşünüldü. Bu arayüz tasarımı yapabilmek için birçok seçenek değerlendirilmiş olup en sonunda Unity oyun motoru kullanılmasına karar verildi.

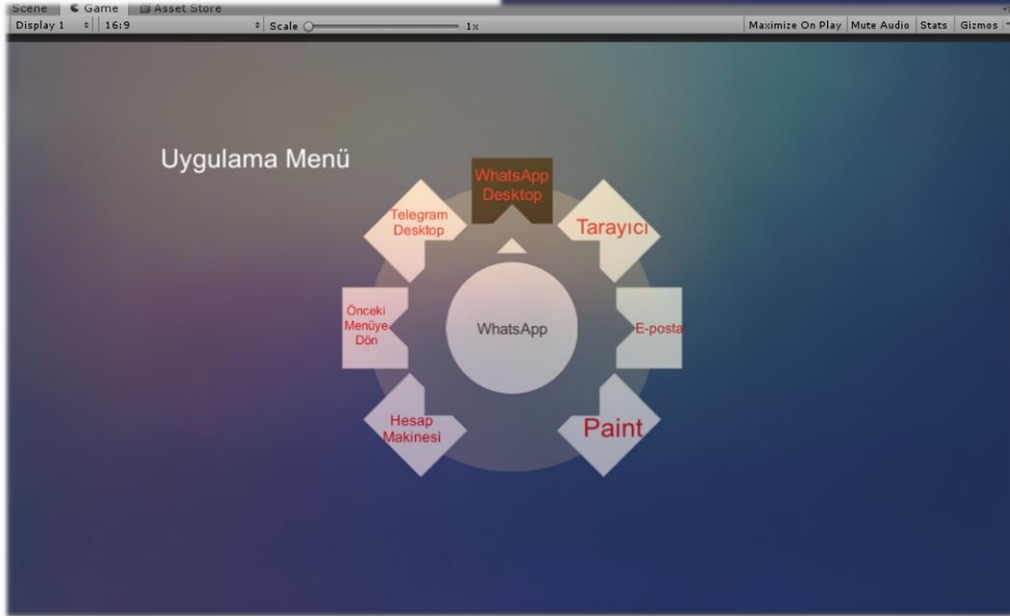
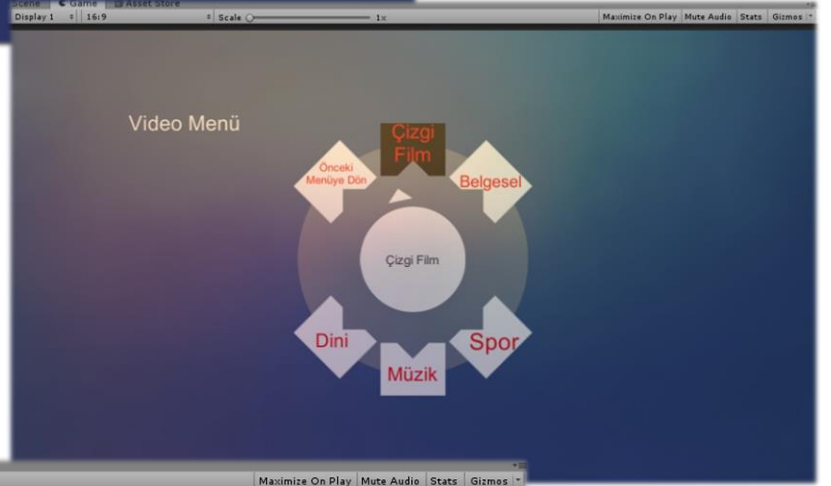
Bu oyun motoru genelde kompleks oyunlar oluşturulmak için kullanılmakta olup tarafımda sadece menü tasarımlarında kullanılan kısmı kullanılacaktır. Bu tasarımda radial menu framework kullanılmıştır. [11]





Unity C# uygulama dilini kullanmaktadır. Bu dil kullanılarak yönlendirmeler ve sahneler arası geçiş gerçekleştirildi. Programda menülerdeki butonlar kullanılarak istenilen uygulamaya, web sayfasına veya ayarlara erişilmesi sağlandı.





BÖLÜM 8: SONUÇ

Oluşturulan arayüzde kullanım kolaylığını sağlamak için tüm seçenekler en kolay şekilde sunulmaya çalışıldı. Her bir butona en sık kullanılan uygulama ve web sayfaları eklenmeye çalışıldı. Oluşturulan bu menülerde imleci butonların üzerine getirmeye gerek kalmadan butonların sahip olduğu açılı içerisinde herhangi bir tıklanma gerçekleşirse o butona basılmış olması sağlandı. Ayrıca istenilen durumda ekran klavyesi kullanılarak tarayıcı veya video sayfası üzerinden farklı içeriklere de gidilmesi mümkün hale getirildi. Bu şekilde kısmi fiziksel engeli bulunan kişiler başta olmak üzere, bilgisayara aşina olmayan çocuk veya yaşlı herkesin kolaylıkla bilgisayar kullanabilmesi sağlanmıştır.

KAYNAKÇA

- [1], [5]: Yeni Başlayanlar İçin Görüntü İşlemeye Giriş, <https://peakup.org/blog/yeni-baslayanlar-icin-goruntu-islemeye-giris/>
- [2]: Python ile Görüntü İşlemeye Giriş, <https://www.mobilhanem.com/python-ile-goruntu-isleme-giris/>
- [3]: Görüntü İşleme Nedir? Hangi Alanlarda Kullanılır?, <https://www.biliminsesi.com/goruntu-isleme-nedir-hangi-alanlarda-kullanilir/>
- [4]: NVIDIA DLSS 2.0 , <https://www.nvidia.com/en-us/geforce/news/nvidia-dlss-2-0-a-big-leap-in-ai-rendering/>
- [6]: Python Nedir? Ve Nereelerde Kullanılır?, <https://alimutlu.com.tr/python-nedir-nerelerde-kullanilir>
- [7]: PYPL Popularity of Programming Language, <https://pypl.github.io/PYPL.html>
- [8],[9]: Görüntü İşleme İçin Öğrenilmesi Gereken Yapılar, <https://peakup.org/blog/goruntu-isleme-icin-ogrenilmesi-gereken-yapilar/>
- [10]: Veri Bilimi İçin Temel Python Kütüphaneleri-1 : Numpy, <https://medium.com/bili%C5%9Fim-hareketi/veri-bilimi-i%C3%A7in-temel-python-k%C3%BCt%C3%BCphaneleri-1-numpy-750429a0d8e5>
- [11] Radial Menu Framework, <https://assetstore.unity.com/packages/tools/gui/radial-menu-framework-50601>
- [12] Convex Hull using OpenCV in Python and C++, <https://www.learnopencv.com/convex-hull-using-opencv-in-python-and-c/>