

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**



**PROJENİN ADI**

**ROBOTİK ELE SAYILARI ÖĞRETME**

**TASARIM PROJESİ**

**259181 CEVAT AKTAŞ**

**259173 SALİH TOSUN**

**2015-2016 GÜZ DÖNEMİ**

**KARADENİZ TEKNİK ÜNİVERSİTESİ  
MÜHENDİSLİK FAKÜLTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

**PROJENİN ADI**

**ROBOTİK ELE SAYILARI ÖĞRETME**

**TASARIM PROJESİ**

**259181 CEVAT AKTAŞ**

**259173 SALİH TOSUN**

**Bu projenin teslim edilmesi ve sunulması tarafımda uygundur.**

**Danışman : PROF. DR. VASİF V. NABİYEV**

.....

**2015-2016 GÜZ DÖNEMİ**



## IEEE Etik Kuralları IEEE Code of Ethics



Mesleğime karşı şahsi sorumluluğumu kabul ederek, hizmet ettiğim toplumlara ve üyelerine en yüksek etik ve mesleki davranışta bulunmaya söz verdiğimi ve aşağıdaki etik kurallarını kabul ettiğimi ifade ederim:

1. Kamu güvenliği, sağlığı ve refahı ile uyumlu kararlar vermenin sorumluluğunu kabul etmek ve kamu veya çevreyi tehdit edebilecek faktörleri derhal açıklamak;
2. Mümkün olabilecek çıkar çatışması, ister gerçekten var olması isterse sadece algı olması, durumlarından kaçınmak. Çıkar çatışması olması durumunda, etkilenen taraflara durumu bildirmek;
3. Mevcut verilere dayalı tahminlerde ve fikir beyan etmelerde gerçekçi ve dürüst olmak;
4. Her türlü rüşveti reddetmek;
5. Mütenasip uygulamalarını ve muhtemel sonuçlarını gözeterek teknoloji anlayışını geliştirmek;
6. Teknik yeterliliklerimizi sürdürmek ve geliştirmek, yeterli eğitim veya tecrübe olması veya işin zorluk sınırları ifade edilmesi durumunda ancak başkaları için teknolojik sorumlulukları üstlenmek;
7. Teknik bir çalışma hakkında yansız bir eleştiri için uğraşmak, eleştiriye kabul etmek ve eleştiriye yapmak; hatları kabul etmek ve düzeltmek; diğer katkı sunanların emeklerini ifade etmek;
8. Bütün kişilere adilane davranmak; ırk, din, cinsiyet, yaş, milliyet, cinsi tercih, cinsiyet kimliği, veya cinsiyet ifadesi üzerinden ayrımcılık yapma durumuna girişmemek;
9. Yanlış veya kötü amaçlı eylemler sonucu kimsenin yaralanması, mülklerinin zarar görmesi, itibarlarının veya istihdamlarının zedelenmesi durumlarının oluşmasından kaçınmak;
10. Meslektaşlara ve yardımcı personele mesleki gelişimlerinde yardımcı olmak ve onları desteklemek.

IEEE Yönetim Kurulu tarafından Ağustos 1990'da onaylanmıştır.

## ÖNSÖZ

Üniversite döneminde edindiğimiz teknik ve yazılımsal deneyimleri pratik de uygulayabilmek mühendisliğin önemli bir gayesidir. Bu amaçla üniversite hayatı boyunca yapmış olunan projeler gelecek dönemdeki iş hayatı için bir ön hazırlık olarak katkı sağlar. Tasarım Projesi sayesinde Bilgisayar Mühendisliği alanında önemli iki dal olan Bilgisayar Yazılımı ve Donanımı dallarını bir arada bulundurduğundan önemli bir tecrübe edinilmesini sağlar. “*ARDUINO İLE PARMAK HAREKETLERİNİN KONTROLÜ*” projesi ile webcamden alınan el görüntülerinin basit hareketlerini robotik ele yaptırmaya çalışan bir proje tasarımı gerçekleştirildi. Gerçekleştirilen projenin başlangıcından itibaren sonuna kadar her safhasında bize yol gösteren , projenin yapılmasında fikirleriyle bizi teşvik eden çok değerli hocamız **PROF. DR. VASİF V. NABIYEV** ‘e teşekkür ederiz. Proje’nin gerçekleştirilmesin de karşılaşılan sıkıntılarda yardımını hiç esirgemeyen arkadaşlarımıza çok teşekkür ederiz. Ayrıca üniversite hayatımız boyunca bizlerden desteklerini hiçbir zaman ayırmayan ailelerimize gönülden saygı ve sevgilerimizi sunarız.

Adı SOYADI

Cevat AKTAŞ

Salih TOSUN

Trabzon 2015

## İÇİNDEKİLER

<b>IEEE ETİK KURALLARI</b>	<b>II</b>
<b>ÖNSÖZ</b>	<b>III</b>
<b>İÇİNDEKİLER</b>	<b>IV</b>
<b>ÖZET</b>	<b>V</b>
<b>1- GİRİŞ ve GENEL BİLGİLER</b>	<b>1</b>
1.1 Giriş.....	1
<b>2- EL GÖRÜNTÜSÜNÜN ALINMASI</b>	<b>2</b>
<b>3- GÖRÜNTÜNÜN ÖNİŞLEMLERİ</b>	<b>3</b>
3.1 Opening İşlemi.....	3
3.2 Otsu ile Görüntünün Binary Yapılması.....	4
<b>4- EL GEOMETRİSİ TANIMA</b>	<b>6</b>
4.1 El Geometrisi.....	6
4.2 El Görüntüsünün İşlenmesi.....	7
4.2.1 Elin Sınır Değerlerinin Bulunması .....	7
4.2.2 Parmak Uçlarının Algılanması.....	8
<b>5- ÖZELLİK VEKTÖRÜNÜN BELİRLENMESİ</b>	<b>10</b>
<b>6- PARMAKLARIN TANINMASI</b>	<b>11</b>
<b>7- ARDUINO DONANIMI ve DİĞER DONANIM ELEMANLARI</b>	<b>12</b>
7.1 Arduino Nedir? .....	12
7.2 Arduino Nano Özellikleri .....	13
7.3 Servo Motor Kullanımı.....	14
7.4 Arduino IDE ' de Kodların Yazılması.....	15

<b>8- SONUÇLAR</b>	<b>19</b>
<b>9- ÖNERİLER</b>	<b>19</b>
<b>10- KAYNAKLAR</b>	<b>20</b>
<b>STANDARTLAR VE KISITLAR</b>	<b>21</b>

## ÖZET

Çalışmada herhangi bir hazır kütüphane kullanılmadan, bilgisayardan alınan görüntü sayesinde el durumunun(parmakların) değişimi algılanarak robotik ele sayılar öğretilir. Parmak sayısının oluşturabileceği çeşitlilikler göz önünde bulundurularak hangi parmak kapalı veya açıksa robotik elin aynı parmağı kapalı veya açıklığı sağlandı. Özellik vektörü olarak el yapısının ağırlık merkezine göre parmak uçlarına olan uzaklıkların değerlendirilmesi gerçekleştirilmiştir. Böylelikle parmak hareketleri yorumlanarak bilgisayara rakamlar öğretilmiş olur ve robot el sayesinde dinamik olarak rakamlar gösterilir.

Projenin el görüntüsünü alıp tanımlama kısmı C# programlama dilinde yazılıp, robot elin hareketleri ise Arduino Nano ile gerçekleştirilmiştir.

## 1-GİRİŞ ve GENEL BİLGİLER

### 1.1 Giriş

İnsanın antropometrik yapılarından olan elin incelenmesi çeşitli araştırmalarda kullanılmaktadır. Bu araştırmalara örnek olarak biyometri, kriminolojik tespit-tanıma sistemleri, işaret dilinin yorumlanması, dokunmalı ekranların tasarımı, animasyonlar gösterilebilir. Çalışmada kamera yardımıyla alınan el parmağı hareketleri yorumlanarak robotik elin hareket ettirilmesi sağlanır. Çalışmanın genel yapısı şekil 1'de gösterilmiştir.



Şekil 1. Projenin Adımları

İlk önce kameradan alınan el görüntüsü Görüntü İşleme Modülüne gönderilerek ön işlemler uygulanmaktadır. Sonra sınırları belirlenen gri seviye ayrık el görüntüsünden sağ ve sol el konumu, parmak uçları belirlenmektedir. Belirlenen elin ağırlık merkezinden parmak uçlarına olan uzaklıklar özellik vektörünü oluşturmaktadır. Bu özellik vektörüne göre parmakların bükülüp bükülmediği anlaşılır. Böylece hangi parmağın kapalı olduğu Arduinoya gönderilip ilgili servo motorlar setlenerek parmaklar kapatılır.



## 2-EL GÖRÜNTÜSÜNÜN ALINMASI

El görüntüsünün alınması işlemi için siyah renkte bir arka plan gereklidir. Arka planın siyah olması önemlidir. Çünkü doğru sonuçlar elde edebilmek için parlaklığın ve ortamdaki gürültünün en aza indirgenmesi gereklidir. Ayrıca elin kameraya gösterileceği yer olarak ışıktan korunaklı etrafı kapalı bir kutu kullanılmalıdır(Şekil 2).



Şekil 2. Kutu

El görüntüsünün kameradan alınması için C#'da basit bir arayüz oluşturularak pictureboxlar kullanılmıştır. Ancak doğrudan video özelliğini desteklemediği için programa AForge.Video.dll ve AForge.Video.DirectShow.dll kütüphanelerini ekleyip videoyu başlatırız. Videodan resimleri almak için buton, timer gibi çeşitli toolar kullanarak videodan dinamik capture yapabiliriz.

```
cam = new VideoCaptureDevice(webcam[comboBox1.SelectedIndex].MonikerString);  
cam.NewFrame += new NewFrameEventHandler(cam_NewFrame);  
cam.Start();
```

Yukarıdaki kod bloğu ile webcam başlatılarak başlatılarak frame'ler yakalanarak video oluşturulur ve daha sonra aşağıdaki kod bloğu ile pictureboxa resim atanır .

```
Bitmap bit = (Bitmap)eventArgs.Frame.Clone();  
pictureBox2.Image = bit;
```

### 3-GÖRÜNTÜNÜN ÖNİŞLEMLERİ

#### 3.1 Opening İşlemi

Alınan görüntünün doğru bir şekilde işlenebilmesi için gürültü minimize edilmelidir. Bunun için çeşitli yöntemler mevcut olup biz projemizde Opening(Açılma) işlemi kullanarak gürültüyü azaltabiliriz. Opening işlemi iki alt işlemden oluşur. İlk olarak Aşındırma(Erosion) kullanılarak görüntüdeki tuz,biber,işık gibi gürültüler aşındırılarak kaybedilmeye çalışılır. İkinci aşamada ise Genişleme(Dilation) işlemi kullanılarak aşındırılan görüntüdeki güçlü kalan alanların genişletilmesi sağlanır.Böylece resme bu iki filtre uygulanarak gürültünün azaltılması ve görüntünün netleştirilmesi sağlanmış olur.



Şekil 3. Önce



Şekil 4. Sonra

Yukarıdaki şekillerde görüldüğü gibi ilk alınan görüntü şekil 3'teki gürültü gibidir. Bu görüntüye erosion ve dilation işlemleri peşi sıra uygulandığında şekil 4'teki gürültüsüz görüntü elde edilir.

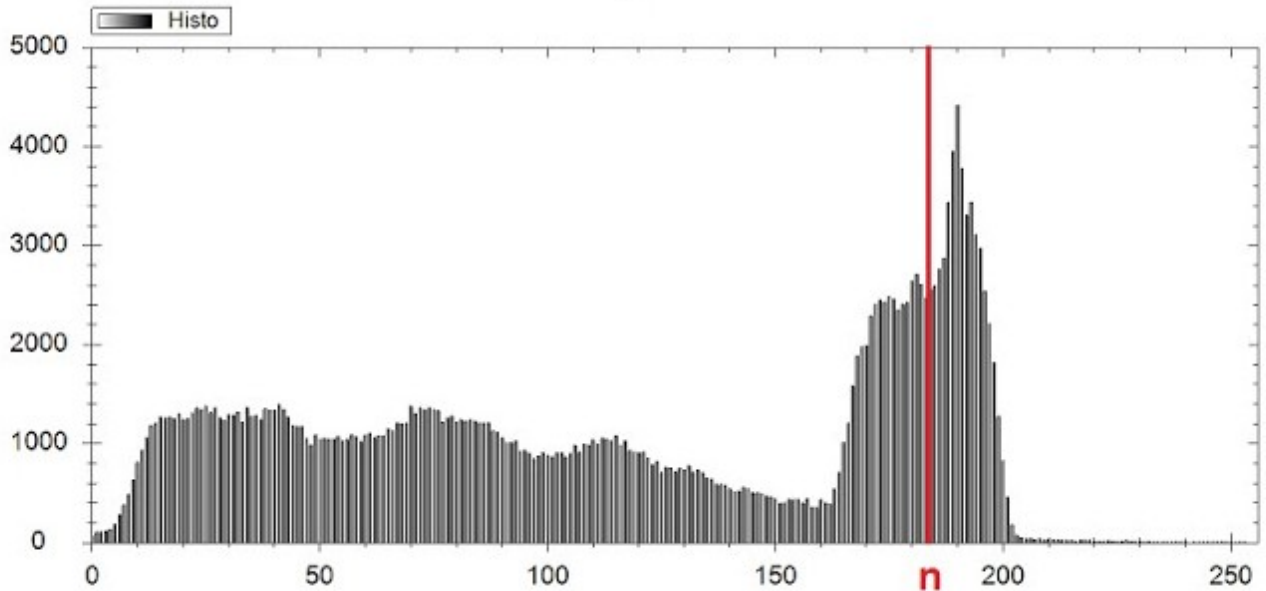
### 3.2 Otsu ile Görüntünün Binary Yapılması

Bölütleme(segmentation) aşamasında kullanılan bu işlem ile gri düzeydeki resim 0-1 lerden (0-255) oluşan bir görüntüye dönüştürülür. Eşikleme (thresholding) görüntüyü işlemek üzere binary hale getirme işlemidir. Algoritma gri seviyeli görüntüler üzerinde çalıştığından önce görüntü  $gri=0.3*kırmızı+0.59*yeşil+0.11*mavi$  formülü ile gri seviye tonlanmış ardından hesaplamalar yapılmıştır.



Şekil 5. Ön İşlemler

*Histogram* tabanlı bir eşikleme yöntemi olan *Otsu Algoritması*, histogramda eşikleme yapılabilecek en uygun konumun bulunması için kullanılır.



Şekil 6. Gri Seviyeye Dönüştürülmüş Resme Ait Histogram

Otsu Algoritmasında histogramın her bir elemanını eşik gibi düşünerek her biri için "*Weight*", "*Mean*" ve "*Variance*" diye tabir edilen veriler hesaplanır.

Şekil 6'da  $n$ 'ye kadar olan değerler background kabul edilir. Sonraki değerler ise foreground olarak kabul edilir. Burada her ikisi için  $\text{mean}(m1, m2)$ ,  $\text{weight}(w1, w2)$ , variance değerleri hesaplanır. Varyans değerine bakılarak renk değerlerinin ortalamaya ne kadar yakın olduğu görülebilir.  $\text{Pr}(x)$  ifadesi  $x(i)$ 'nin gelme olasılığıdır. Düzgün dağılımlı  $N$  uzunluklu bir dizi için bu değer  $1/N$  dir.  $M \times N$  bir görüntü için konuşacak olursak histogram dizisi hesaplandıktan sonra  $i$  tonunun gelme olasılığı  $\text{histogram}[i]/(M \times N)$  dir.

$$\omega_1(t) = \sum_0^t \text{Pr}(i) \quad \mu_1(t) = \sum_0^t \text{Pr}(i)X(i)$$

$$\omega_2(t) = \sum_{t+1}^{255} \text{Pr}(i) \quad \mu_2(t) = \sum_{t+1}^{255} \text{Pr}(i)X(i)$$

$$\sigma^2 = \sum_{i=0}^{N-1} (x_i - \bar{x}) \text{Pr}(x_i) \quad (\text{Variance})$$

Sınıflar arası varyans ise aşağıdaki formül ile hesaplanır.

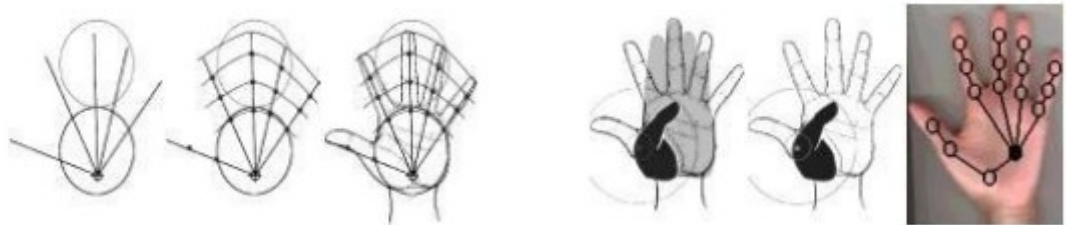
$$\sigma_s(t) = \sigma - \sigma_b(t) = \omega_1(t) * \omega_2(t) (\mu_1(t) - \mu_2(t))$$

Yazılan program,  $t=0$  ilk değer ile başlayarak 255'e kadar her değer için sınıflar arası varyansı hesaplamakta ve en son maksimum varyans değerini veren  $t$  değerini şekil 6'daki  $n$  eşik değeri olarak döndürmektedir. Bu eşik öncesinde kalan yani background demiş olduğumuz kısımlardaki değerler 0 diğerleri 1 (yani 255) olarak görüntü düzenlenir. Böylece binary görüntü elde edilmiş olur. (Şekil 5)

## 4-EL GEOMETRİSİ TANIMA

### 4.1 El Geometrisi

El yapısının anlaşılması için parmakların, bileğin yerinin, elin genişliğinin, parmağın konumunun belirlenmesi önem taşır. Elin yapısı incelendiğinde, elin yaklaşık iki eşit dikdörtgen (veya kare) alanı kapsadığı görülmektedir. Bu alanlara ovaler çizilerek, alt kısımda seçilen noktadan (ilk üçü  $15^\circ$  olmak koşulu ile) 4 doğru geçilir. 4 parmağı (işaret, orta, yüzük ve serçe) ifade eden doğrular tepesi orta parmakta olan 3 kavisle kesilir. Parmak ucuna yakın olan eklem, parmağın  $1/3$ 'nin üstünde yerleşir. Geriye kalan kısım ise yarıya bölünür ve sonuçta parmağın 2 eklemi bulunur. 3. eklem ise parmakların avuçla kesiştiği kısımlarda saklıdır. Başparmak diğer parmalardan daha farklıdır ve kendi oku üzerinde hareket etmektedir. Baş parmak avuç içerisine taban noktasına eğildiğinde serçe parmağın başlangıç noktasına dokunmakta ve işaret parmağının 2. eklemine kadar dönebilmektedir. İşaret ve yüzük parmakları yaklaşık aynı uzunlukta olmakta, yüzük parmağı yaklaşık  $1/2$  parmak yastığı kadar daha uzun olmaktadır. Serçe parmağı yüzük parmağının üst eklemine kadar uzanmaktadır. Serçe parmağı yaklaşık bilek eni uzunluğundadır. Baş parmak dışında geriye kalan parmakların boylarının büküm kısımlarına olan oranı yaklaşık altın orana, yani 1.62 değerine eşit olmaktadır. Ayrıca elin yapısının kare değil, daha çok kama biçiminde olduğunun bilinmesi önemlidir.(şekil 7)



Şekil 7. El Geometrisi

## 4.2 El Görüntüsünün İşlenmesi

### 4.2.1 Elin Sınır Değerlerinin Bulunması

Önişlemler, alınan el görüntüsündeki gürültülerin bastırılması, kenarların algılanması, resmin ikili hale dönüştürülmesi işlemleri uygulandı. Arka plandan arındırılmış el görüntülerinin sınırlarının belirlenmesi için **kontur izleme algoritması** kullanılmaktadır.

Bu algoritma ile ikili görüntü üzerinde başlangıç noktasından sınır izlenerek tekrar aynı noktaya geri dönülmektedir. Nesnenin çevresinin belirlenmesi için sekiz yönlü arama yapılmaktadır. Bu algoritmada izlenen adımlar aşağıdaki verilmiştir:

1. Yeni piksel değeri başlangıç pikseli değilken arama yapılır.
2. X pikseli için 1 yönünden aramaya başlanır. 1 yönünün seçilebilmesi için; 1'in siyah piksel, 1 için bir önceki yön sayılan 8'in beyaz piksel olması ve bir adım önce ilerleme yaptığımız yön 1 için zıt yön sayılan 5 yönü değilse, yeni konum 1 nolu piksel olur ve işlemlere bu pikselden devam edilir.
3. Eğer 2. adım sağlanmazsa bir sonraki değerli piksele geçilir.

Bu algoritma geliştirilirse; yeni piksel değeri bulunabilmesi için artan sıradaki yönlerde arama yapılır. Yön değerinin yeni değer olarak atanması için; kendi değerinin siyah, ondan bir önceki arama yönündeki pikselin beyaz olması ve bir önceki arama yönünün ona zıt olan yönlerden biri olmaması gerekir. Zıt yön değerine bakılmasının nedeni; tekrar eski değere dönmesini önlemek içindir. Şekil 8'de el çevresinin işaretlenmesinin sonuçları gösterilmiştir.

8	1	2
7	X	3
6	5	4



Şekil 8. Sekiz yönlü arama ve sonucu

#### 4.2.2 Parmak Uçlarının Algılanması

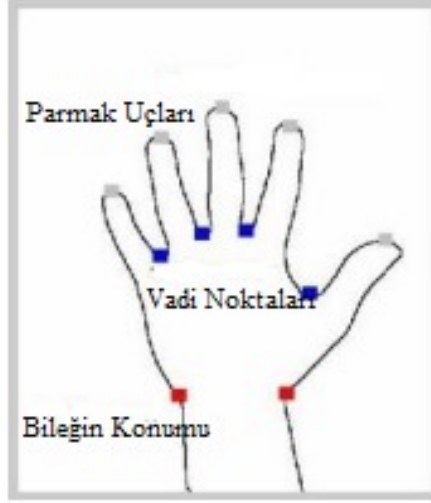
Parmak araları belirlenirken iki parmak arasında kalan siyah bölgede en alttaki noktalar seçilmektedir. Dolayısıyla 4 ara noktanın belirlenmesi gerekmektedir. Resim, üstten başlayarak satırlar boyunca taranır ve siyah aralıklar belirlendikten sonra aralıkların orta noktalarına göre parmakların avuçla kesişme noktaları seçilir. İlk ve son nokta için bölgedeki iç kısımlardan aşağıya doğru ilerlenir. Bileğin bulunmasında satırlar boyunca aşağıdan yukarıya doğru bileğin bulunduğu muhtemel beyaz aralık belirlenir. Soldan ve sağdan ikili geçişlere göre ilk referans noktaları seçilir. Sabit adımlarla bir üst seviyedeki bilgilerle var olan bilgiler karşılaştırılır ve tekrarlı biçimde bir artışa rastlanırsa işlemler durdurularak bilek yeri tespit edilir.

Parmakların uç noktalarını bulmak için kontur izleme algoritması ile elde edilen eli çevreleyen pikseller dizisinden faydalanılabilir. Bu diziye **tepe tırmanma (hill climbing) arama algoritması** uygulanarak parmakların yeri ve kapalı veya açık olması yorumlanabilir. Bu algoritma aşağıdaki şekilde özetlenebilir:

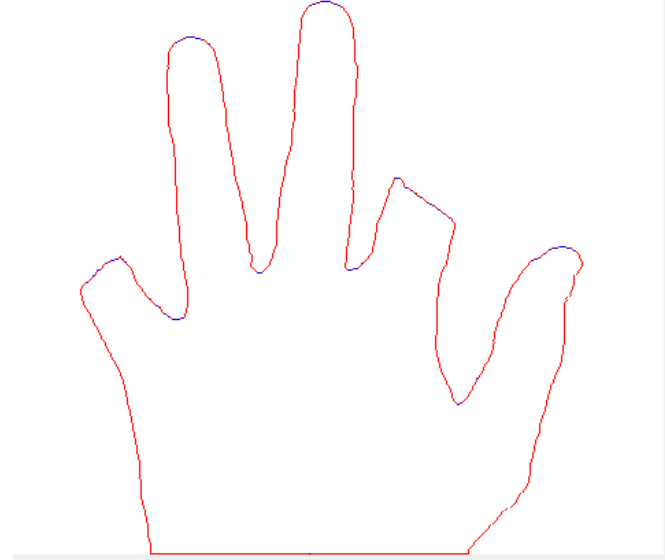
1. Başlangıç konum seçilir ve geçerli konum olarak atanır.
2. Bir sonraki konum komşu konum olarak alınır.
3. Komşu konumun değeri geçerli konumun değerinden büyükse, komşu konum geçerli konum olarak atanır.
4. Eğer geçerli konum komşu konumdan eşik değeri kadar büyükse geçerli konum yerel maksimum olarak belirlenir.

Söz konusu algoritma ele uygulandığında parmak uçlarının bulunmasında kolaylıklar sağlamaktadır. Çünkü parmak sınırlarının ağırlık merkezine göre uzaklıkları doğrusal olarak artıp azalmaktadır. Bu sınırları içeren diziler grafiksel olarak gösterilirse elde edilen pikler parmak uçlarını gösterecektir. Pikler arasındaki minimum noktalar ise parmak aralarına(vadiye) karşılık gelecektir .(şekil 9)

Parmak uçları maksimum, parmak araları ise ardışık şekilde minimumlara uygun gelmektedir. Bir pikselin parmak ucu olup olmadığına bakmak için pik noktalarının 15 piksel önceki ve 15 piksel sonraki değeri arasındaki uzaklık hesaplanır. Uzaklık değeri bir parmak kalınlığı şeklinde yorumlanabiliyorsa parmak ucu olarak atanır.



**Şekil 9. Tepe Tırmanma Algoritmasının Uygulanması**



**Şekil 10. Uç Noktaların Algılanması**

Yukarıdaki şekillerde görüldüğü gibi parmak uçları algılanıp bu pikseller mavi renge boyanmıştır.



## 5- ÖZELLİK VEKTÖRÜNÜN BELİRLENMESİ

El sınırı belirlendikten sonra bulunan bu kapalı yapının ağırlık merkezi kolaylıkla bulunabilir( $G_x, G_y$ ).

$$G_x = \frac{\sum_{j=1}^{\text{en}} \sum_{i=1}^{\text{boy}} F_x(i, j)}{\text{siyah piksel sayısı}}, \quad G_y = \frac{\sum_{i=1}^{\text{boy}} \sum_{j=1}^{\text{en}} F_y(i, j)}{\text{siyah piksel sayısı}}.$$

Özellik vektörünün oluşturulmasında kontur izleme algoritması ile elde edilen dizinin elemanlarının ağırlık merkezine olan uzaklıkları hesaplanır. Bu diziye tepe tırmanma (hill climbing) arama algoritması uygulanarak parmakların yeri ve kapalı veya açık olması yorumlanabilir. Parmak uçları her bir parmak için ağırlık merkezinden en uzak noktanın özelliğini verir. Diğer bir deyişle, maksimum mesafe değeri ağırlık merkezi ve her bir parmağın komşu parmağa olan mesafesinin arasındadır. Mesafe ölçümü, Öklid mesafe formülü ile aşağıdaki gibi yapılır:

$(X, Y)$  elin dış hatlarında olan noktalardır ve ağırlık merkezi elin  $(X_{ref}, Y_{ref})$  konumundadır. Öklid mesafesi  $D_E$  dış hat ve referans noktası arasındadır.

$$D_E = \sqrt{(X - X_{ref})^2 + (Y - Y_{ref})^2}$$

## 6- PARMAKLARIN TANINMASI

Tanım birimi, veri tabanıyla özellik çıkarma vektöründen elde edilen giriş imgesini karşılaştırır. Böylece tanıma veya doğrulama için sonuçlar verilir. Sistem, çıkarılan özellikleri kaydeder, tanıma işlemi için veri tabanıla özellik vektörü eşleştirilir. Bir kişinin kimliğinin tanınması iki temel tipte sınıflandırılabilir: doğrulama ve teşhis.

Doğrulama, kimlik iddia eden kişiyi doğrular ya da reddeder. Doğrulama sisteminde, bireysel talepler, önceden sistemde kayıtlı olan kullanıcılar içindir. Sistem, veritabanında kayıtlı olan kişi ile o kişi olduğunu iddia eden kişiyi özellik vektörüyle(elde edilen veri) eşleştirerek onaylar veya reddeder. Böylece, doğrulama sistemi bire bir eşleştirme işlemi yapar. Doğrulama ve tanıma amacıyla eşleştirme yapmak için, bazı mesafe fonksiyonları, minimum mesafe sınıflandırıcısı ve bağıntı fonksiyonuna dayanır.

Bu projemizde doğrulama işlemi basit olup dosyaya kayıtlı parmak uzunluklarıyla o anda kameradan alınıp işlenilen görüntü karşılaştırılır. Eğer karşılaştırılan parmaklar arasında belli bir mesafe farkı varsa o parmak kapatılmıştır diyoruz..

Kamera yardımıyla alınan el görüntüsünden parmaklar ve bu parmakların konumu dikkate alınarak uygun nota seslendirilmesi gerçekleştirilmiştir. Uygulamalarda elin konumu ve yönü sabitlendirilmediğinde, ağırlık merkezinden yola çıkıldığından bazı hataların olduğu görülmüştür. Ayrıca bitişik parmaklar olan durumlarda parmakların konumlarının belirlenmesinde zorluklar oluşmakta ve bundan dolayı doğru notanın seslendirilmesinde hatalar ortaya çıkmaktadır. Bu hatalar elin yapısı dikkate alınarak giderilebilir.

## 7- ARDUINO DONANIMI ve DİĞER DONANIM ELEMANLARI

### 7.1 Arduino Nedir?

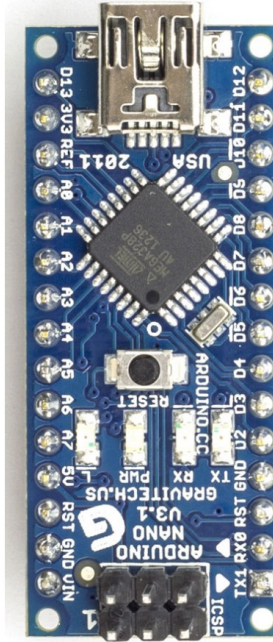
Arduino, açık kaynak kodlu geliştirilmiş olup yazılım ve donanım tabanlı bir elektronik prototipidir. Arduino mikrodenetleyici programlamak ve bu mikrodenetleyiciyi giriş-çıkış pinleri sayesinde başka fiziksel cihazlarla haberleşmesini sağlamak için üretilmiş platformdur. En önemli özelliği, ayrıntılardan kurtarıp doğrudan mikrodenetleyiciyi programlamaya imkan sağlar. Arduino'nun temel bileşenleri;

1. Arduino IDE
2. Arduino Bootloader
3. Arduino Kütüphaneleri (servo kütüphanesi kullanıldı.)
4. AVR Dude
5. Derleyici (AVR-GCC)

Arduino IDE geliştirme ortamı Processing dilinin özelliklerini taşır. Arduino bootloader mikrodenetleyiciyi programlayabilmek için kullanılan programın adıdır. Programlayıcılar kullanarak mikrodenetleyicileri kullanmak gerekir. Seri haberleşme üzerinden de bu programlamayı yapabiliriz. Bunun için mikrodenetleyicinin kendi program belleğini programlaması gerekmektedir. Bootloader bu işi yapan programdır. Kodlar derlendikten sonra programlamak için bu program kullanılır. Proje , Arduino çeşitlerinden nano ile gerçekleştirildi.

## 7.2 Arduino Nano Özellikleri

Microcontroller : ATmega168/ ATmega328  
Operating Voltage : 5 V  
Input Voltage : 7-12 V  
Digital IO/PWM : 14 /6  
Analog In/Out : 8  
DC Current per I/O Pin : 40 mA  
Flash Memory : 16 KB/32 KB  
SRAM : 1 KB/2 KB  
EEPROM : 512 bytes/1 KB  
Clock Speed : 16 MHz



Arduino	Microcontroller
1(TX)	- PD1(TXD)
0(RX)	- PD0(RXD)
D2	- PD2(INT0)
D3	- PD3(INT1)
D4	- PD4
D5	- PD5
D6	- PD6
D7	- PD7
D8	- PB0
D9	- PB1
D10	- PB2(SS')
D11	- PB3(MOSI)
D12	- PB4(MISO)
D13	- PB5(SCK)
A0	- PC0
A1	- PC1
A2	- PC2
A3	- PC3
A4	- PC4(SDA)
A5	- PC5(SCL)
A6	- ADC6
A7	- ADC7

Şekil 11. Arduino Nano

Şekil 11. de Arduino Nano ve özellikleri gösterilmiştir. Arduino Nano , Atmega328 temelli bir mikrodenetleyici kartıdır. Üzerinde 14 adet dijital giriş/çıkış pini (6 tanesi PWM çıkışı olarak kullanılabilir) , 8 analog giriş , 16Mhz kristal , usb soketi , ICSP konektörü ve reset tuşu bulundurmaktadır. Kart üzerinde mikrodenetleyicinin çalışması için gerekli olan her şey bulunmaktadır. Kolayca usb kablosu üzerinden bilgisayara bağlanabilir , adaptör veya pil ile çalıştırılabilir. Program yüklemek ve bilgisayar haberleşmesi yapmak için üzerinde FTDI FT232 usb-seri dönüştürücü bulundurmaktadır. 5 servo motor için dijital I/O pinlerinden 5 tanesi projede kullanıldı. Bu servo motorlardan en fazla iki tanesini ile Arduino Nano platformunda 5V gerilim ile beslenmelidir. Geri kalan motorların beslenmeleri için PC Power güç kaynağından yararlanıldı.

### 7.3 Servo Motor Kullanımı



Şekil 12. Servo Motor



Şekil 13. Servo Motorların Yerleştirilmesi

Şekil 12 de Servo motor gösterilmiştir. Servo, içinde bulunan geri besleme devresi sayesinde sistemde oluşacak hatayı bulan, bu hataları denetleyen ve gideren elemanlardır. Servo motorlar kullanıcıdan aldığı komutları yerine getirirler. Bu komutlar hız, ivme ve mekaniksel konum olabilir. Bu komutları aynı zamanda geri besleme olarak çıkış olarak da verirler. Servo motor içinde bir çok ara elemanı bulundurur. Elektrik motorları, sürücü ve kontrol devreleri vb. bunlara örnektir. Bu özelliklerinden dolayı sistemlerde çıkış hareketlerini kontrol edici amaçlı kullanılırlar. Servo motorlar birçok alanda kullanılmaktadır. Robotlar, radarlar, sanayi makinelerinde, sanayi robot kollarında, tıbbi cihazlarda ... gibi alanlarda kullanılırlar. Değişken yük ve hız değişikliği yapılabilmeleri, kararlılıklarının yüksek olması, pozisyon değeri verilebilmeleri ve belli konumsal periyotla çalışmalarından dolayı tercih edilmektedirler. Servo motoru alıcılara bağlarken kahverengi giriş (GND), kırmızı giriş (5V), sarı giriş (signal) olarak kullanılır. Şekil 13'de 5 tane servo motorun robotik elin parmakları ile bağlantısı gösterilmiştir.

## 7.4 Arduino IDE ' de Kodların Yazılması

```
#include <Servo.h>
Servo myservo1;
Servo myservo2;
Servo myservo3;
Servo myservo4;
Servo myservo5;

int pos = 0;

void setup() {
  Serial.begin(9600);
  myservo1.attach(2);
  myservo2.attach(3);
  myservo3.attach(4);
  myservo4.attach(5);
  myservo5.attach(6);
}
void loop() {

  if(Serial.available()>0)
  {
    char parmak = Serial.read();
    switch(parmak)
    {
      case 'a':
        Serial.println("a");
        parmakBirKapat();
        break;

      case 'b':
        Serial.println("b");
        parmakIkiKapat();
        break;

      case 'c':
        Serial.println("c");
        parmakUcKapat();
        break;

      case 'd':
        Serial.println("d");
        parmakDortKapat();
        break;

      case 'e':
        Serial.println("e");
        parmakBesKapat();
        break;

      case 'A':
        Serial.println("A");
        myservo1.write(180);
        break;

      case 'B':
        Serial.println("B");
        myservo2.write(180);
        break;
    }
  }
}
```

```

        case 'C':
        Serial.println("C");
        myservo3.write(180);
        break;

        case 'D':
        Serial.println("D");
        myservo4.write(180);
        break;

        case 'E':
        Serial.println("E");
        myservo5.write(180);
        break;
    }
}

void parmakBirKapat(){
    myservo1.write(180);
    myservo1.write(0);
}
void parmakIkiKapat(){
    myservo2.write(180);
    myservo2.write(0);
}
void parmakUcKapat(){
    myservo3.write(0);
    delay(3000);
    myservo3.write(180);
}
void parmakDortKapat(){
    myservo4.write(180);
    myservo4.write(0);
}

void parmakBesKapat(){
    myservo5.write(0);
    delay(3000);
    myservo5.write(180);
}

```

Yukarıdaki tabloda Arduino IDE ' de yazılan kodlar gösterilmiştir. 5 tane servo motor nesnesi tanımlandı. *loop* fonksiyonunda Visual Studio'dan gönderilen verilerin kontrolü yapıldı. Aşağıdaki tabloda bu kontrolün neden bu şekilde yapıldığı anlatılacak. Parmakların bükülmesi veya açılması için fonksiyonlar tanımlanmıştır.

```

SerialPort sp = new SerialPort();
try
{
    sp.BaudRate = 9600;
    sp.PortName = "COM7";
    sp.Open();
}
catch
{
    MessageBox.Show("bağlantı hatası");
}
if (sp.IsOpen == true) {
    if (parmakSonuc[0] == 1)
        sp.Write("a");
    else
        sp.Write("A");

    string data = sp.ReadLine();
    while (data != "a" || data!="A")
        goto etiket;

etiket :    if (parmakSonuc[1] == 1)
            sp.Write("b");
            else
                sp.Write("B");

    string data1 = sp.ReadLine();
    while (data1 != "b" || data1 != "B")
        goto etiket1;

etiket1:    if (parmakSonuc[2] == 1)
            sp.Write("c");
            else
                sp.Write("C");

    string data2 = sp.ReadLine();
    while (data2 != "c" || data2 != "C")
        goto etiket2;

etiket2:    if (parmakSonuc[3] == 1)
            sp.Write("d");
            else
                sp.Write("D");

    string data3 = sp.ReadLine();
    while (data3 != "d" || data3 != "D")
        goto etiket3;

etiket3:    if (parmakSonuc[4] == 1)
            sp.Write("e");
            else
                sp.Write("E");
}

```



Yukarıdaki tabloda Arduino Nano haberleşmesini sağlamak için *SerialPort* nesnesi tanımlandı. BaudRate (saniyede transfer edilen bit sayısı) tanımlanmıştır. Kullanılacak port ismi tanımlandı ve bağlantı açıldı. *parmakSonuc* dizisinde her bir parmağa ait değer sırayla serçe parmaktan baş parmağa kadar 1 (bükülmüş) veya 0 (bükülmemiş) değeri tutulur. Seri haberleşmede *String* bir veri gönderilemediğinden karakter olarak gönderildi. Arduino IDE de *Serial.print()* ile karakterler yazılıp , bu karakterleri Visual Studio'da okuyup *while* döngüleri içinde kontrolleri yapıldı.

## 8- SONUÇLAR

- Görüntü alınırken ortamadaki gürültünün görüntüyü bozduğu görülmüştür ve özel bir düzenek kurulmuştur.
- Görüntünün gürültüsü giderilerek binary forma dönüştürülmüştür.
- Elin çevresi ve yerel ekstremum noktaları hesaplanarak bilgisayara elin geometrik yapısı öğretilmiştir.
- Uygulamalarda elin konumu ve yönü sabitlendirilmediğinde, ağırlık merkezinden yola çıkıldığından bazı hataların olduğu görülmüştür.
- Bitişik parmaklar olan durumlarda parmakların konumlarının belirlenmesinde zorluklar oluşmaktadır.
- Arduino Nano üzerinden servo motor bağlantıları yapıldı.
- Kamera yardımıyla alınan el görüntüsünden parmaklar ve bu parmakların konumu dikkate alınarak Arduinoya gönderilmiştir.
- Parmakların bükülüp açılması gerçekleştirildi.

## 9- ÖNERİLER

Projemizde uygulanması ve kolay programlanabilir olması açısından Arduino Nano kullandık. Bu modulün boyutunun küçük olması ve dijital I/O giriş çıkış sayısı fazla olmasından dolayı uygulama geliştirme açısından tercih edilme sebeplerinden sadece birkaçıdır. Ayrıca machine learning ile ilgili çalışmalarda , yapay zekadan faydalanılarak gerçekleştirdiğimiz proje birçok sektörde kullanılması mümkündür.

## 10- KAYNAKLAR

- [1] <http://www.hindawi.com/journals/tswj/2014/267872>
- [2] **Ravikiran J, Kavi Mahesh, Suhas Mahishi, Dheeraj R, Sudheender S, Nitin V Pujari** , “ Finger Detection for Sign Language Recognition ” , **Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I** IMECS 2009, March 18 - 20, 2009, Hong Kong
- [3] **Mokhtar M. Hasan** , Pramod K. Mishra , “ **Real Time Fingers and Palm Locating using Dynamic Circle Templates** ” , *International Journal of Computer Applications (0975 – 8887) Volume 41– No.6, March 2012*
- [4] **Van Bang Le, Anh Tu Nguyen, and Yu Zhu** , “ Hand Detecting and Positioning Based on Depth Image of Kinect Sensor” , *International Journal of Information and Electronics Engineering, Vol. 4, No. 3, May 2014*
- [5] **Jagdish Lal Raheja** , Karen Das , Ankit Chaudhary , “ Fingertip Detection: A Fast Method with Natural Hand ”
- [6] <https://www.arduino.cc/>
- [7] <http://arduino.stackexchange.com/questions/5274/c-and-arduino-problem-with-serial-port>
- [8] **Vasif Vagifoğlu NABIYEV** “ Yapay Zeka:İnsan-Bilgisayar Etkileşimi ” , Seçkin Yayıncılık 2012
- [9] <https://cevataktas.wordpress.com/category/goruntu-isleme/>

## STANDARTLAR ve KISITLAR FORMU

Projenin hazırlanmasında uyulan standart ve kısıtlarla ilgili olarak, aşağıdaki soruları cevaplayınız.

1. Projenizin tasarım boyutu nedir? (Yeni bir proje midir? Var olan bir projenin tekrarı mıdır? Bir projenin parçası mıdır? Sizin tasarımınız proje toplamının yüzde olarak ne kadarını oluşturmaktadır?)

Projenin gerçekleşmesi için Arduino Nano modülünden yararlanılmıştır. Böylelikle parmakların amaca uygun servo motorların kontrol edilmesiyle bükülüp açılması sağlanmıştır. Bu proje yeni bir proje olmasına rağmen bu alanda teknoloji hızla geliştiğinden günümüzde geniş bir alanda yayılım gösterdiğinden gelecek için kesinlikle bir yönelim alanıdır.

2. Projenizde bir mühendislik problemini kendiniz formüle edip, çözdünüz mü? Açıklayınız.

Projede parmağın fiziksel özelliklerini bilgisayara öğretmek için hesaplanması gereken yerel ekstremum noktaları için recursive bir algoritma yazıldı.

3. Önceki derslerde edindiğiniz hangi bilgi ve becerileri kullandınız?

Hazırladığımız projede yazılım kısmında Java, Yapay Zeka ve Görüntü İşleme derslerinden öğrendiğimiz bilgiler ile, donanım kısmı ise Elektrik ve Elektronik Devreler dersleri yardımıyla gerçekleştirilmiştir.

4. Kullandığınız veya dikkate aldığınız mühendislik standartları nelerdir? (Proje konunuzla ilgili olarak kullandığınız ve kullanılması gereken standartları burada kod ve isimleri ile sıralayınız).

Mühendislik standartları dikkate alındığında tasarlanan projenin masaüstü uygulaması olarak kullanılması sağlanmıştır. Bu kapsamda Arduino ve C# programlama dili kullanılarak geliştirilmiştir.

5. Kullandığınız veya dikkate aldığınız gerçekçi kısıtlar nelerdir? Lütfen boşlukları uygun yanıtlarla doldurunuz.

a) Ekonomi

Tasarımın en önemli ekonomik yükü kullanılan Arduino Nano ve servo motorların maliyetidir.

b) Çevre sorunları:

Projenin çevreye olumsuz bir etkisi bulunmamaktadır.

c) Sürdürülebilirlik:

Günümüzde popüler bir konu olan machine learning alanında etkin bir şekilde kullanılmaktadır. Bu proje ile birçok alanda insan yaşamını kolaylaştıracak ve teknolojik gelişmeler ile bu alandaki çalışmalar daha da ileri götürülecektir.

d) Üretilebilirlik:

Üretilebilirlik açısından bir çok alanda kullanılabilir.

e) Etik:

Tasarlanan projenin etik açısından bir sakıncası yoktur.

f) Sağlık:

Sağlık açısından bir sakıncası yoktur.

g) Güvenlik:

Güvenlik ile ilgili herhangi bir problem oluşturmamaktadır.

h) Sosyal ve politik sorunlar:

Tasarımın herhangi bir sosyal ve politik sorunu bulunmamaktadır.