



# ROBOTİK ELE SAYILARI ÖĞRETME

Cevat AKTAŞ  
Salih TOSUN

DANIŞMAN : PROF. DR. VASİF V. NABİEV



## ÖZET

Projede yapay zeka ve görüntü işleme birlikte kullanılmıştır. Şekil 7 'de görüntü işleme ve yapay zeka arasındaki ilişki gösterilmiştir. Projede herhangi bir hazır kütüphane kullanılmadan , bilgisayardan alınan görüntü ile parmakların değişimi algılanarak robotik ele sayılar öğretilir. Parmak sayısının oluşturabileceği çeşitlikler göz önünde bulundurularak hangi parmak kapalı veya açıksa robotik elin aynı parmağın kapatılması veya açılması sağlandı. Özellik vektörü olarak el yapısının ağırlık merkezine göre parmak uçlarına olan uzaklıkların değerlendirilmesi gerçekleştirilmiştir. Böylelikle parmak hareketleri yorumlanarak bilgisayara rakamlar öğretilmiş olunur ve robotik el ile rakamlar gösterilir.

El görüntüsünün alınıp tanımlanması kısmı C# programlama dilinde yazılıp, robotik elin hareketi ise servo motorlar kullanılarak Arduino Nano ile gerçekleştirilmiştir. Şekil 9 'da robotik el gösterilmiştir.

## YAPILAN ÇALIŞMALAR

İnsanın antropometrik yapılarından olan elin incelenmesi çeşitli araştırmalarda kullanılmaktadır. Bu çalışmalara örnek olarak biyometri, kriminolojik tespit-tanıma sistemleri, işaret dilinin yorumlanması, dokunmalı ekranların tasarımı, animasyonlar gösterilebilir. Kameradan alınan el görüntüsünün hareketleri, yorumlanarak robotik elin hareket ettirilmesi sağlanır. Şekil 8 'de proje adımları gösterilmiştir.

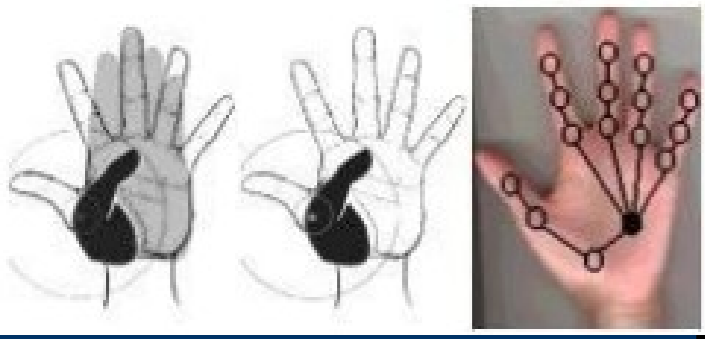
İlk önce el görüntüsünün alınması için siyah renkte bir arka plan gereklidir. Arka planın siyah olması önemlidir. Çünkü doğru sonuçlar elde edebilmek için parlaklığın ve ortamdaki gürültünün en aza indirgenmesi gereklidir. İkinci işlem olarak, elde edilen görüntünün Otsu ile Binary yapılması işlemidir. 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. Binary görüntünün elde edilmesi Şekil 1'de gösterilmiştir.

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 görülür. Baş parmak dışında geri kalan parmakların boylarının büküm kısımlarına oranı yaklaşık altın orana, yani 1.62 değerine eşit olmaktadır. El geometrisi Şekil 2 'de gösterilmektedir. Sıradaki işlem, el görüntüsünün sınırlarının belirlenmesi için **Contour Tracing Algoritması** kullanılmaktadır. Bu algoritma ile görüntü üzerinden başlangıç noktasından sınır izlenerek tekrar aynı noktaya geri dönülmektedir. Nesnenin çevresinin belirlenmesi için sekiz yönlü arama yapılmaktadır. Algoritma aşağıda 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.



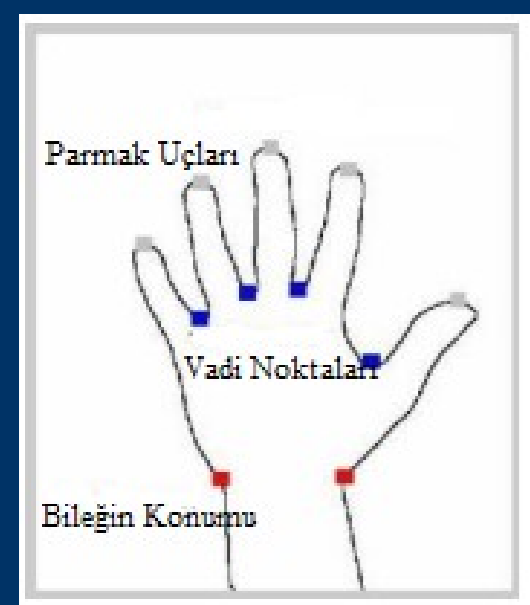
Şekil 1. Ön İşlemler



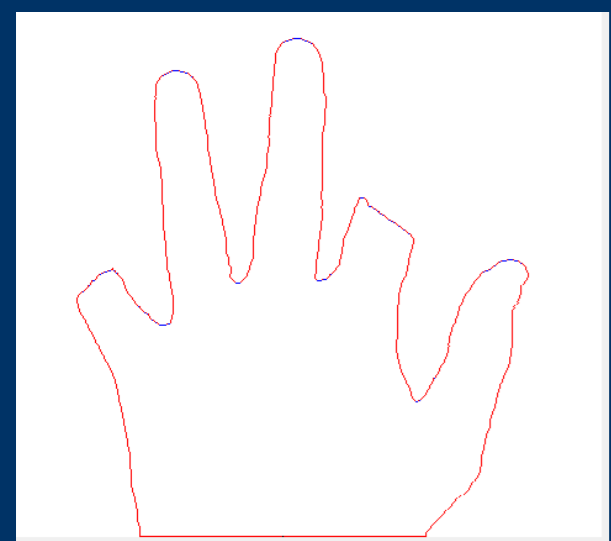
Şekil 2. El Geometrisi



Şekil 3. Sekiz Yönlü Arama



Şekil 4. Tepe Tırmanma Algoritması



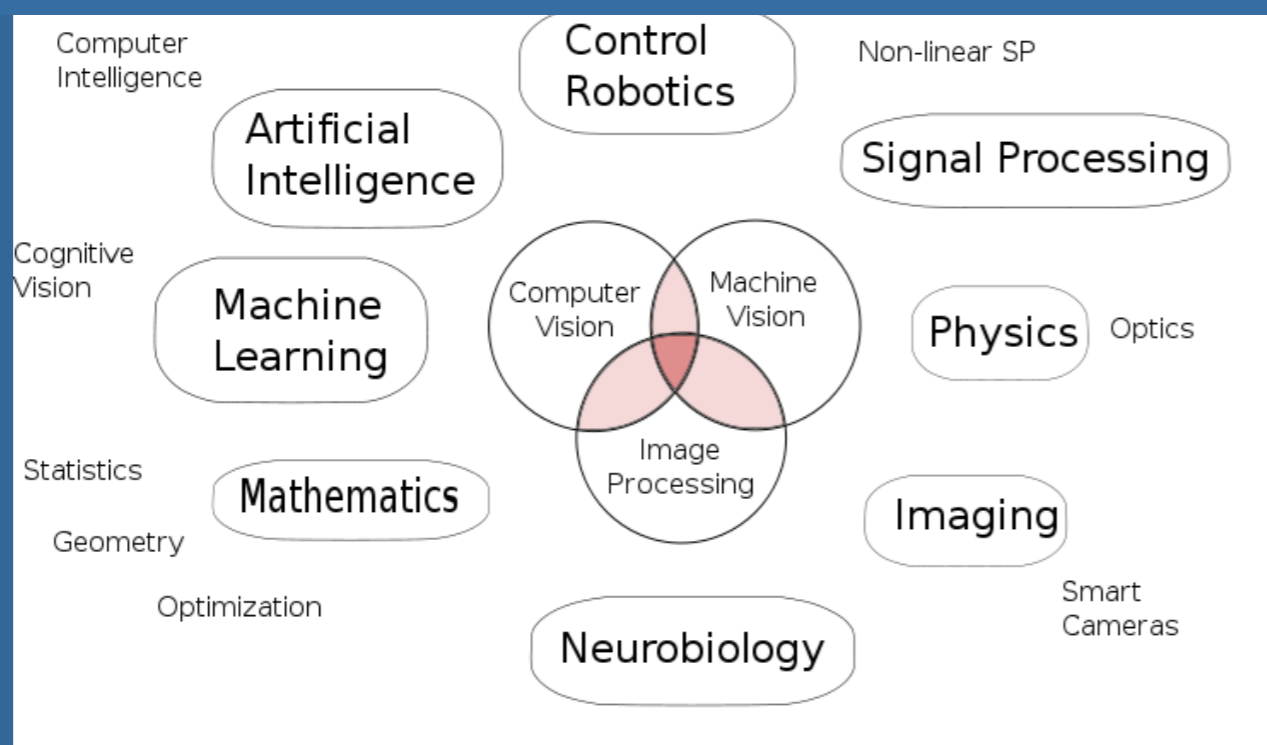
Şekil 5. Uç Noktaların Algılanması



Şekil 6. Sonuç



Şekil 9. Robotik El



Şekil 7. Görüntü İşleme ve Yapay Zeka İlişkisi

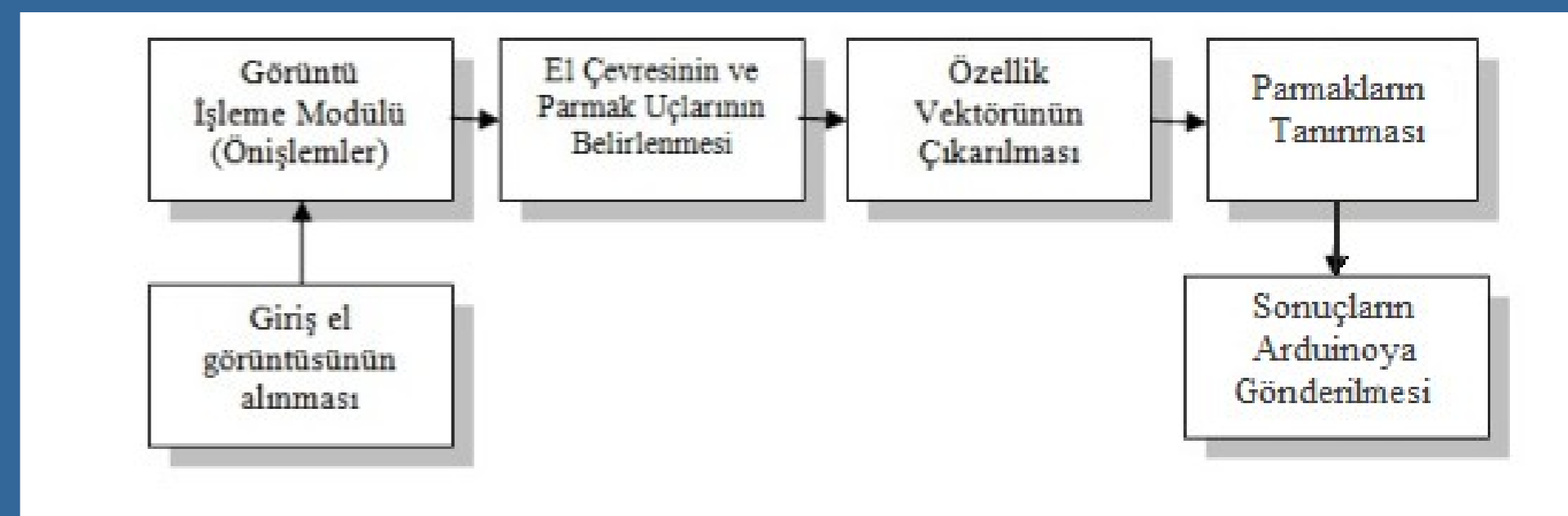
Sırada parmak uçlarının algılanması işlemi var. Parmak araları belirlenirken iki parmak arasında kalan siyah bölgede en alttaki noktalar seçilmektedir. Dolayısıyla 4 ara noktanın belirlenmesi gerekmektedir. Resim, üstten başlanarak satırlar boyunca taranır ve siyah aralıklar belirlendikten sonra aralıkların orta noktalarına göre parmakların avuç ile 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 **Contour Tracing** algoritması ile elde edilen eli çevreleyen pikseller dizisinden faydalanılabilir. Bu diziye **Tepe Tırmanma (Hill Climbing) arama algoritması** uygulanarak parmakların 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.

Tepe tırmanma algoritması Şekil 4 'de gösterilmiştir. 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 5 'de gösterilmiştir.

Bir sonraki adım özellik vektörünün belirlenmesi ve parmakların tanınması işlemidir. Parmak uçları her bir parmak için ağırlık merkezinden en uzak noktanın özelliğini verir. Yani maksimum mesafe değeri ağırlık mekezi ve her bir parmağın komşu parmağa olan mesafesinin arasındadır. Mesafe ölçümü Öklid mesafe formülü ile yapıldı. Tanıma birimi, veritabanıyla özellik çıkarma vektöründen elde edilen giriş imgesini karşılaştırır. Böylece tanıma ve doğrulama için sonuçlar verilir. Sistem çıkarılan özellikleri kaydeder, tanıma işlemi için veritabanı ile özellik vektörü eşleştirilir. Doğrulama işlemi basit olup dosyaya kayıtlı parmak uzunlukları ile o anda kameradan alınıp işlenen görüntü karşılaştırılır. Eğer karşılaştırılan parmaklar arasında belli bir mesafe farkı varsa o parmak kapatılmıştır denilir.

En son adımda, açılan(0) veya kapatılan(1) parmaklar büyük ve küçük karakterler olarak Arduino IDE 'de okunur ve fonksiyonlarla parmakların açılıp kapanması yapılır.



Şekil 8. Proje Adımları

## SONUÇ ve ÖNERİLER

Projenin uygulanması ve kolay programlanabilir olması açısından Arduino kullanıldı. Bir nesnenin özellik vektörünün nasıl çıkarıldığı ve veritabanı ile karşılaştırma yapılarak nesnenin belirlenmesi öğrenildi. Gelecekte yapay zeka ve görüntü işleme geliştirilip ve bu alanlarda yapılan çalışmalar birçok sektörde kullanılacaktır.