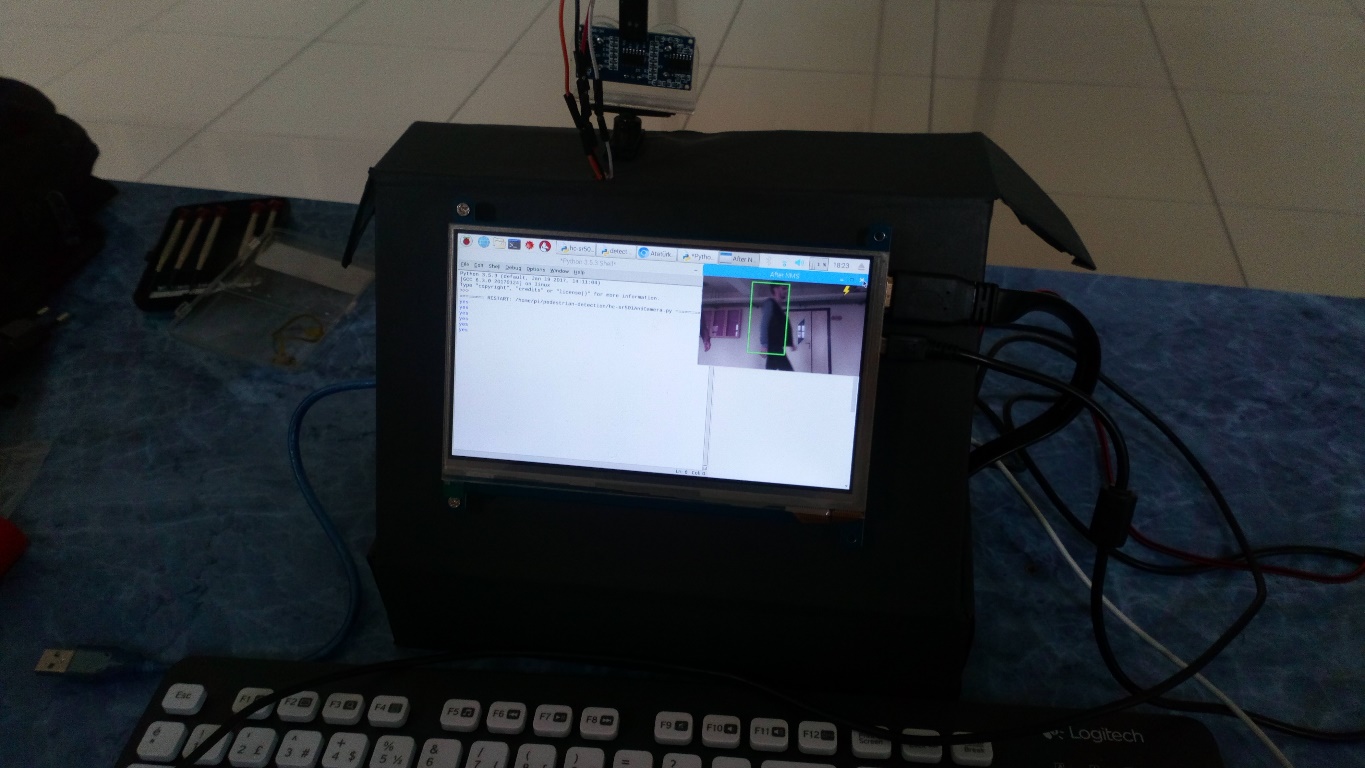
1. **Proje Adı:** ATİT(Alan Tarama ve İnsan Tespiti)
2. **Proje Amacı:** Belirli bir alan içerisinde güvenlik veya benzeri amaçlar için sensorlar ve görüntü işleme metotları kullanılarak insan tespit edilmesi. Sabit bir sistem üzerinde sensor bağlantıları yapılarak sınırlanan alan içerisinde herhangi bir canlının varlığı belirlenecektir. Canlı tespiti yapıldığında ortam üzerinden alınan resimde görüntü işleme metotları kullanılarak sensorlar tarafından tespit edilen varlığın insan olup olmadığı belirlenecektir.
3. Projede Kullanılacak Metod **:** Projede sensör taramaları için raspberry pi 3 üzerinden PIR ve HC-SR04 sensörlerin kullanılması ve görüntü işleme için raspberry pi 3 kamera mödülü kullanılacaktır.Görüntü işleme de hem MobilNet mimarisi hem de Tek Çekim dedektörü (SSD) çerçevesinde birleştirilerek nesne algılama için derin öğrenme yapılacaktır.
4. Projenin Tanımlanması**:** Projede sabit bir sistem üzerinde sensör bağlantıları yapılarak sınırlı alan içerisinde herhangi bir canlı olup olmadığını belirlemeye çalışılacaktır. Canlı bir varlığın olduğu belirlendiğinde kayıtta olan kamera üzerinden anlık fotoğraf çekilerek fotoğraf üzerinde görüntü işleme ile bazı kısıtlar dahilinde sensörler tarafından tespit edilen varlığın insan olup olmadığını belirlenecektir.
5. Proje Maliyeti**:**

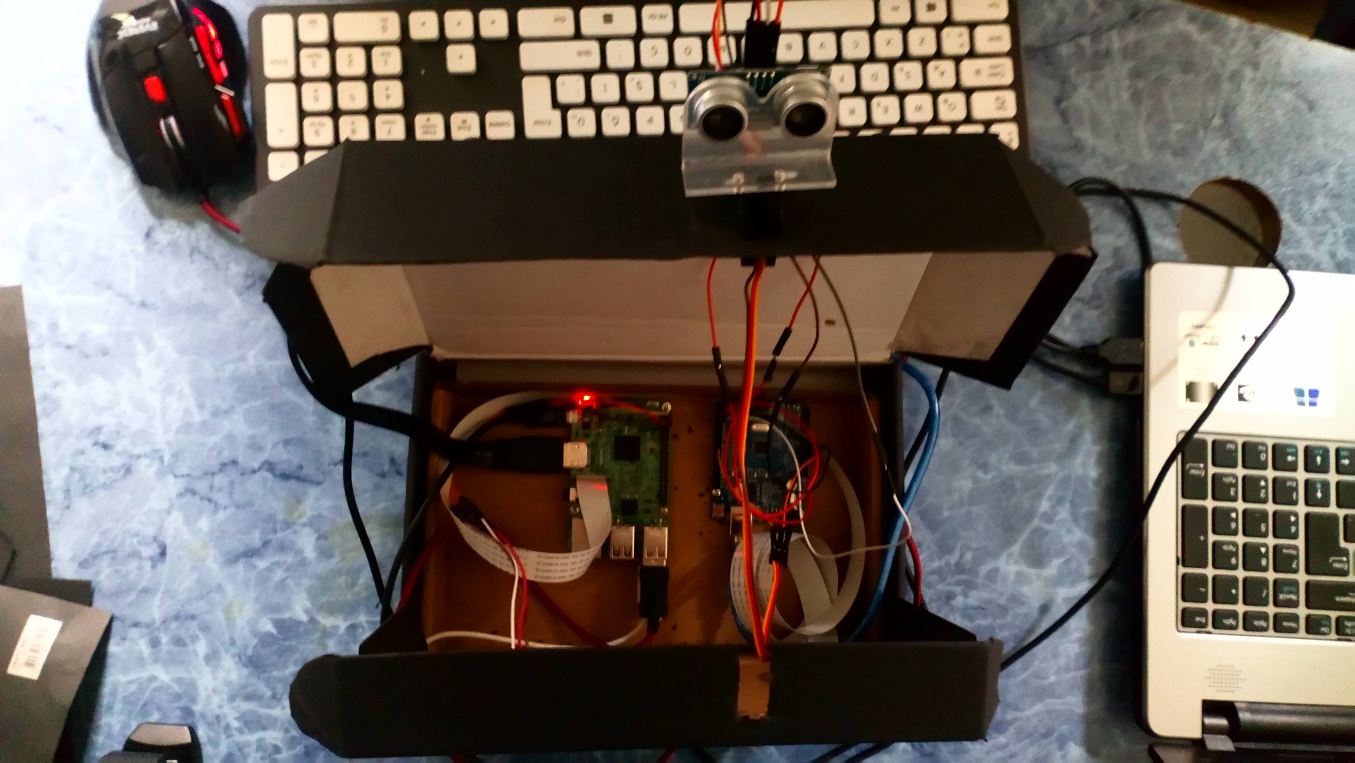
|  |  |
| --- | --- |
| Ürün | Birim Fiyat |
| Hareket Sensörü \*2 | 3,45 TL |
| HC-SR04 | 7,45 TL |
| Raspberry Pi 3 | 144,50 TL |
| Kızılötesi Kamera Modülü | 160 TL |
| Servo Motor | 26,50 TL |
| Bread Board | 14,90 TL |
| Jumper Kablo | 10 TL |
| Hafıza Kartı | 39,50 TL |
| 7 inc Dokunmatik Ekran | 222 TL |
| **TOPLAM MALİYET** | **632,80 TL** |

# Proje Planı

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | Bölümün Adı | **Süresi (Hafta)** | **Başlangıç**  **Zamanı** | **Bitiş Zamanı** |
| 1 | Sensör tespiti ve sensörlerin kullanım alan ve koşullarının incelenmesi | 3 | 1. Hafta Başı | 3. Hafta Sonu |
| 2 | Görüntü işleme teknik bilgi ve kod öğrenme | 7 | 1. Hafta Başı | 7. Hafta Sonu |
| 3 | Sensör temini ve bağlantı şemalarının çizilmesi | 5 | 4. Hafta Başı | 8. Hafta Sonu |
| 4 | Bir video üzerinde resim alınması ve bir obje tespiti yapılması | 6 | 8. Hafta Başı | 13. Hafta Sonu |
| 5 | Yapılan sistem üzerinden görüntü olarak insan tespitinin yapılması | 7 | 14. Hafta Başı | 19. Hafta Sonu |
| 6 | Arduino üzerinde sensör bağlantılarının yapılması ve gerekli yazılımın tamamlanması | 7 | 9. Hafta Başı | 15. Hafta Sonu |
| 7 | Ortam taraması üzerinde deneme yapılması | 3 | 16.Hafta Başı | 18. Hafta Sonu |

# Projenin Genel Görüntüsü









**PROJE AFİŞİ**



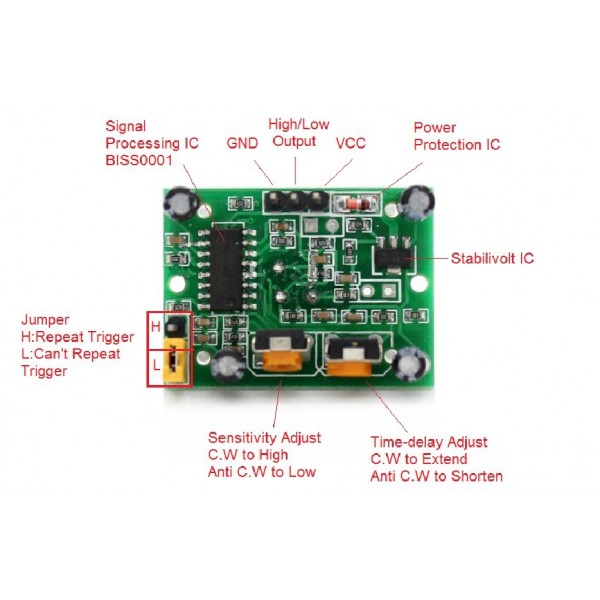
# PROJEDE KULLANILAN SENSÖRLER

Projede belirlenen ortamın taranması hedeflenmektedir. Belirlenen alan içerisine bir varlığın veya nesnenin giriş yaptığını algılaya bilmek için belirlenen alan HC-SR04 sensör ile tarama yapılarak giriş yaptığı tespit edilecektir. PIR sensör ile belirlenen alan içerisine sonradan giriş yapan varlığın canlı olup olmadığı tespit edilecektir.

HAREKET SENSÖRÜ**:**

Bu sensor PIR sistemi temeliyle işleyen bir yapıdır. PIR sensörler görüş alanlarındaki insanların ve sıcak kanlı canlıların yaydıkları IR ışıkları algılayabilen sensörlerdir. Genellikle hareket sensörü olarak kullanılırlar. Canlı hareket ettiğinde sensörün çokgen yapısından dolayı algıladığı ışığın anlık olarak kesilip yeniden sensör üzerine düşmesi ile PIR sensör hareketi algılar. Sensorün algıladığı sinyallerin hareket halindeki bir cisim, canlı ve nesneyle karşılaşması durumunda bizlere o alanda hareket olduğu bilgisini iletir. Bizlerde bu hareketin bilgisiyle görüntü işlemeyle yapının insan olup olmadığını algılama temeline dayanan bir sensör kullanımı gerçekleştitecegiz.

Teknik Özellikleri**:**

****

* Voltaj Aralığı: DC 4.5-20V
* Akım: <50uA
* Output: High 3.3 V / Low 0V
* Tetiklenme: L tekrarlanamaz / H tekrarlanabilir
* Gecikme Zamanı: 5-200s Ayarlanabilir
* Block zamanı: 2.5s
* Boyutlar: 32mm\*24mm
* Voltaj Aralığı: DC 4.5-20V
* Akım: 50ua
* Ölçüm Menzili: Max 7 m

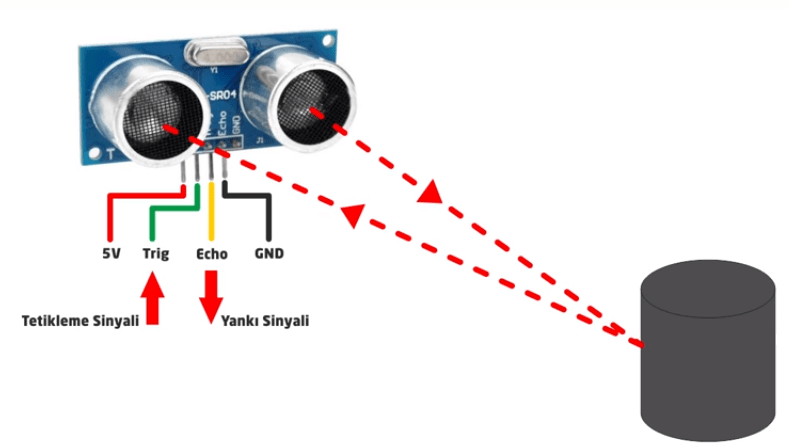
**Şekil 1:**PIR sensörü görüntüsü

MESAFE SENSÖRÜ**:**

Bizim kullanacağımız sensor ultrasonik tarama temeline dayanan yapıya sahiptir. Ultrasonik ses dalgaları 20 kHz ile 500 kHz arasında frekanslara sahip ses dalgalarıdır. Bizim duyabildiğimiz 300 Hz-14000 Hz bandının üzerindedirler. Ultrasonik sensörler ultrasonik ses dalgaları yayan ve bunların engellere çarpıp geri dönmesine kadar geçen süreyi hesaplayarak aradaki uzaklığı belirleyebilen sensörlerdir. Bu sensör sisteme servo motor ile yerleştirilerek istenilen alanı geniş şekilde tarayacağız. Sensorün yolladığı sinyallerin bir cisim, canlı ve nesneyle karşılaşması durumunda bizlere o alanda tehtid olduğu bilgisini iletir.

Ultrasonik Teknoloji**:**

Bu hareket sensörü çeşidi, hareketi algılamak için Doppler sinyalizasyonlarını kullanır. Sensörler, alanda bulunan nesnelerden yansıyacak ultrasonik ses dalgaları gönderir ve sonra ses dalgalarının geri dönme süresini hesaplar. Alanda bir hareket olduğunda, bu ses dalgaları sensör alıcısına farklı frekanslarda dönecektir, bu da sensörün hareketi algılamasını sağlar. Bu teknoloji, sensörün direkt görüş alanında olmadığı ya da aktivite seviyesinin düşük olduğu zamanlarda kullanım için idealdir. Algılama mesafesi PIR dedektörlere göre daha geniştir. Algılama mesafeleri sensörün bulunduğu ortamdaki fiziki şartlar (Isı, nem, toz ), kullanıldığı yükseklik ve çevredeki fiziki engellere bağlı olarak değişir.



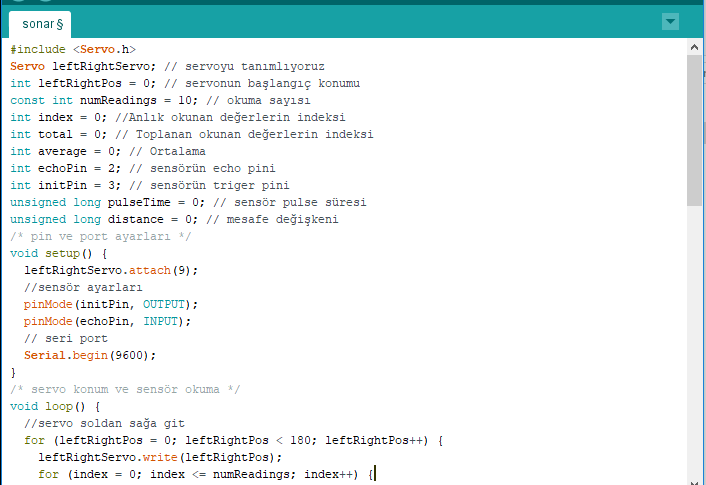
**Şekil 2:** HC-SR04 çalışma mantığı

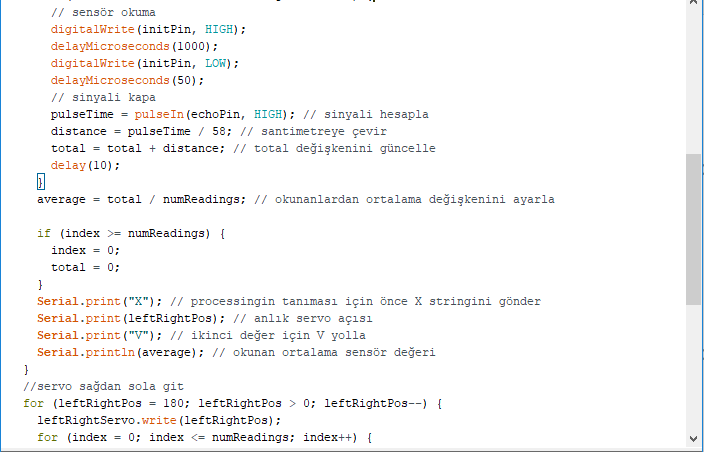
### HC-SR04 Ultrasonik Sensör Özellikleri

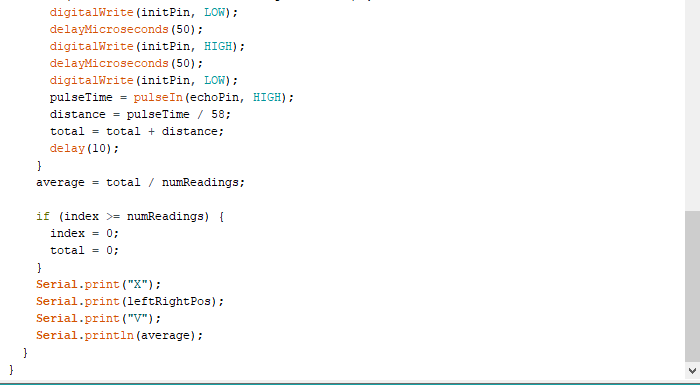
* Çalışma Voltajı:5 V DC
* Çalışma Frekansı:40Hz
* Çektiği Akım:15 Ma
* Görme Açısı:15 derece
* Tetik Bacağa Giriş Sinyali:10 us TTL Darbesi
* Echo Çıkış Sinyali:Giriş TTL Sinyali ve Mesafe Oranı
* Ölçüm Hassasiyeti:3 mm
* Ölçülen Mesafe:2cm/400

SONAR EKRAN

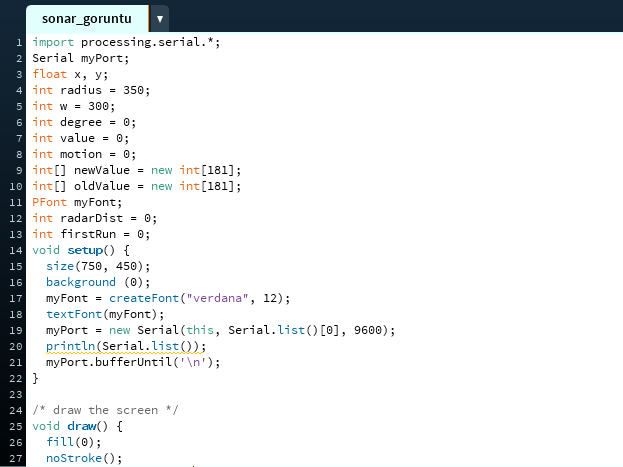
ARDUİNO KODU :

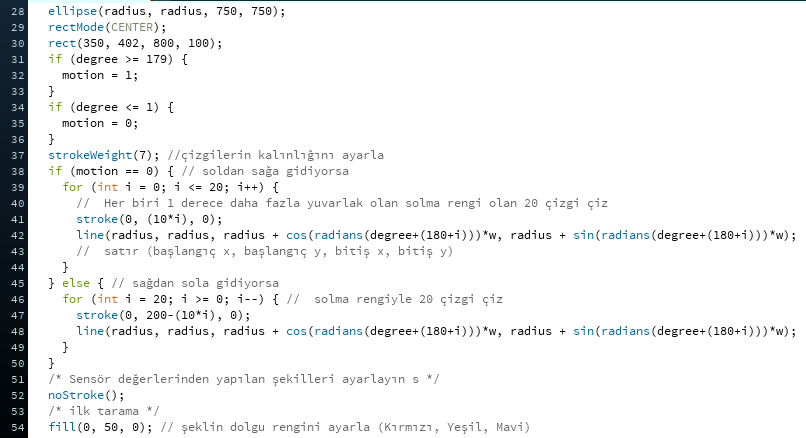


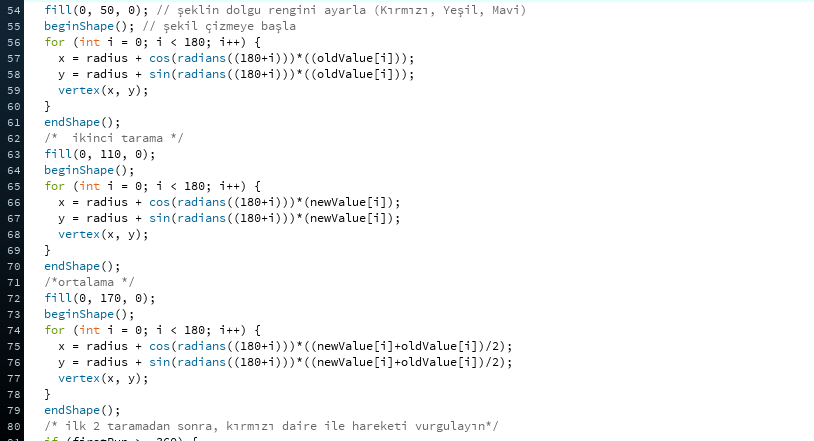


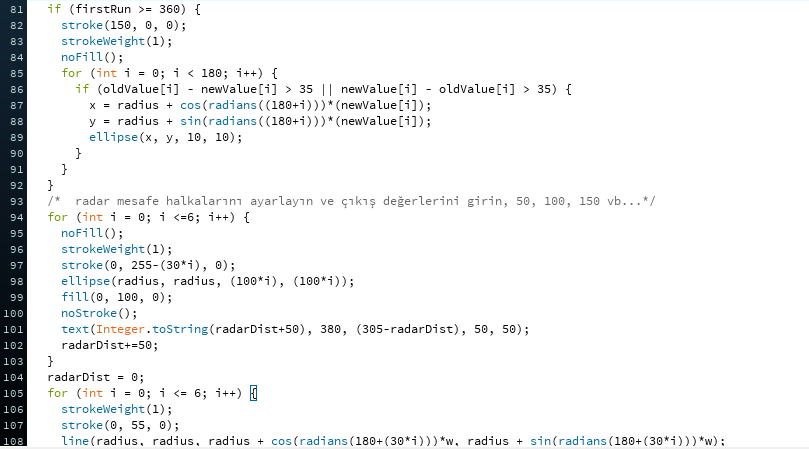


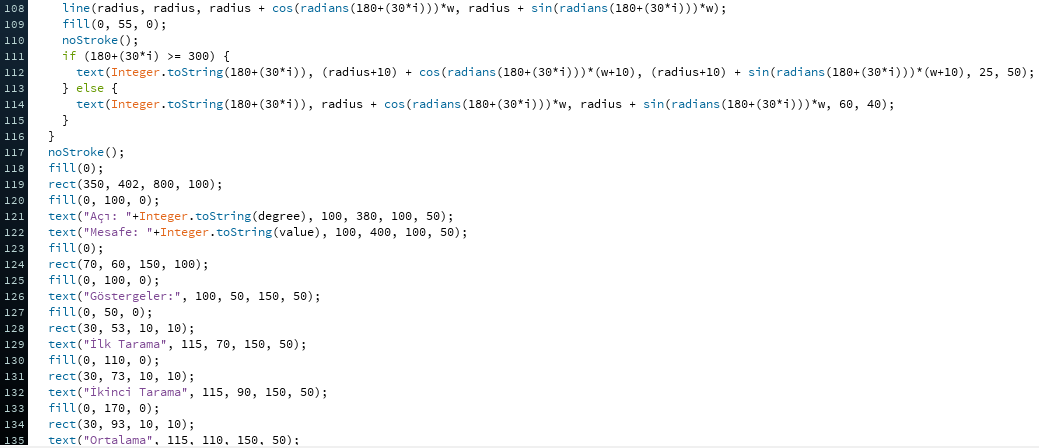
**PROCESSİNG KODU :**

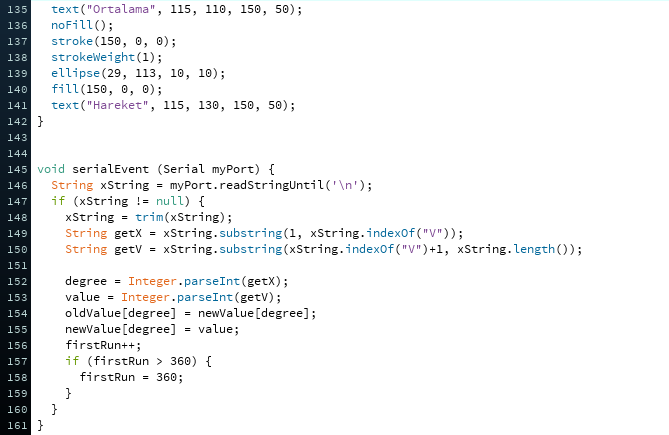




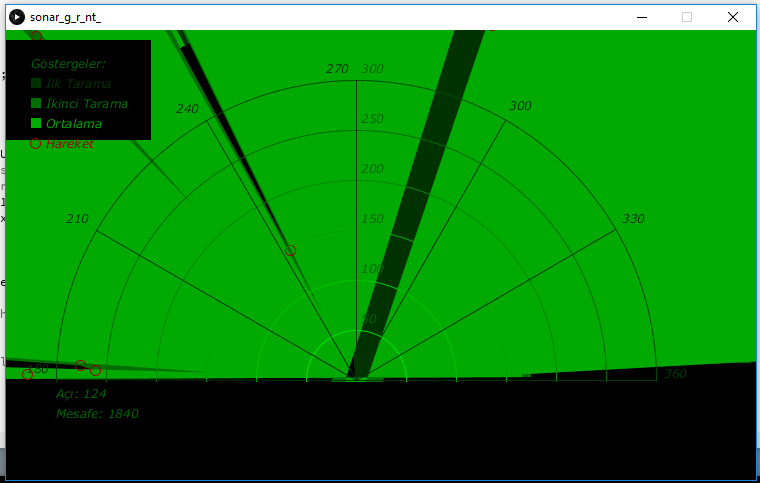








**SONAR EKRAN GÖRÜNTÜSÜ**

****

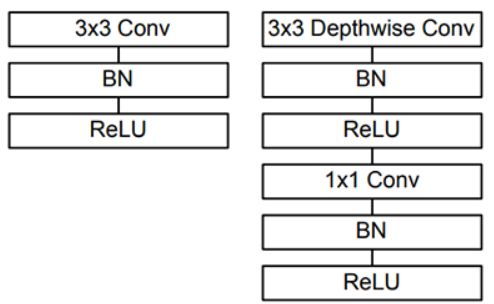
**Şekil 3:** Sonar Ekranı

**Çalışma Mantığı :** Ultrasonic sensor , Servo Motor , Arduino Uno ile kurduğumuz sistemde alan iki kez taranarak ortamın cisim haritası çıkartıldı. Bu sayede ortamdaki nesnelerin konumu ve mesafesi ölçülmüş oldu. Bu sabit nesneler yukardaki örnek ekran görüntüsünde yeşil alanlar olarak tespit edildi. Sistemin hareket algılanması durumunda ise hareket konıumu sistem kullanıcısı tarafından algılaması için kırmızı daireler ile belirtildi. Kullanıcı ekrandan kırmızı daireler sayesinde haraketli cismin açısal konumunu ve mesafesini belirlecektir.

**PROJEDE KULLANILAN GÖRÜNTÜ İŞLEME METODLARI**

Projede OpenCV içerisindeki MobileNET, SSD (Tek Çekim Dedektörü) ve DNN (Derin Sinir Ağı) kullanılmıştır.

**MobileNET :** Mobil Vizyon Uygulamaları için Verimli Konvolüsyon Sinir Ağları (Google):



**Şekil 4 :** MobileNet Mimarisi

MobileNets, hafif derin sinir ağlarını oluşturmak için derinlikli ayrılabilir kıvrımlar kullanan aerodinamik bir mimariye dayanmaktadır. Gecikme ve doğruluk arasında verimli bir şekilde ticaret yapan iki basit küresel hiper parametreyi sunuyoruz.

Bu hiper parametreler, model oluşturucunun, problemin kısıtlamalarına dayanarak, uygulama için doğru boyutlu modeli seçmesine izin verir. Kaynak ve doğruluk tradeoffları üzerinde kapsamlı deneyler sunmaktayız ve ImageNet sınıflandırmasında diğer popüler modellere kıyasla güçlü performans göstermektedir. Daha sonra çok çeşitli uygulamalarda MobileNets'in etkinliğini ve nesne tespiti, finegrain sınıflandırması, yüz özellikleri ve büyük ölçekli coğrafi konumlandırma gibi kullanım durumlarını gösteririz.(11)

**SSD (Tek Çekim Dedektörü) :**

SSD olarak adlandırdığımız yaklaşımımız, sınırlayıcı kutuların çıkış alanını, özellik harita konumu başına farklı en boy oranlarına ve ölçeklere göre bir dizi varsayılan kutuya ayırır.Tahmin zamanında, ağ her bir varsayılan kutuda her nesne kategorisinin varlığı için puanlar üretir ve nesne şeklini daha iyi eşleştirmek için kutuya ayarlamalar üretir. Ek olarak, ağ, farklı boyutlardaki nesnelerin doğal olarak ele alınabilmesi için, farklı çözünürlüklerdeki çoklu özellikli haritalardan gelen tahminleri birleştirir.(12)

SSD, nesnel teklifler gerektiren yöntemlere göre basittir, çünkü teklif üretme ve sonraki piksel veya özellik yeniden örnekleme aşamalarını tamamen ortadan kaldırır ve tek bir ağda tüm hesaplamaları kapsüller. Bu, SSD'yi bir algılama bileşeni gerektiren sistemlere entegre etmek için eğitmeyi ve basitleştirmeyi kolaylaştırır. PASCAL VOC, COCO ve ILSVRC veri kümeleri üzerindeki deneysel sonuçlar, SSD'nin ek bir nesne önerme adımını kullanan ve çok daha hızlı olan ve hem eğitim hem de çıkarım için birleşik bir çerçeve sunarken, yöntemlere rekabetçi bir doğruluğa sahip olduğunu doğrulamaktadır.

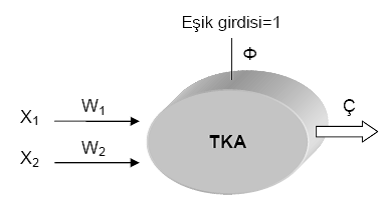
İçin 300 × 300 giriş, SSD bir Nvidia Titan X ve 512 × 512 için 59 FPS'de VOC2007 testinde % 74.3 mAP elde etti.(13,14)

SSD, daha hızlı R-CNN modeli ile kıyaslanabilir bir performans sergileyen SSD'nin% 76.9'unu yakaladı. Diğer tek aşamalı yöntemlerle karşılaştırıldığında, SSD daha küçük bir giriş görüntüsü boyutuyla bile çok daha iyi doğruluğa sahiptir.(15)

**DNN (Derin Sinir Ağı**) **:**

Canlıların davranışlarını inceleyip, matematiksel olarak modelleyip, benzer yapay modellerin üretilmesine *sibernetik* denir. Eğitilebilir, adaptif ve kendi kendine organize olup öğrenebilen ve değerlendirme yapabilen yapay sinir ağları ile insan beyninin öğrenme yapısı modellenmeye çalışılmaktadır. Aynı insanda olduğu gibi yapay sinir ağları vasıtasıyla makinelerin eğitilmesi, öğrenmesi ve karar vermesi amaçlanmaktadır.

**Tek Katmanlı Algılayıcılar**  
Tek katmanlı yapay sinir ağları sadece girdi ve çıktı (Ç) katmanlarından oluşur. Çıktı üniteleri bütün girdi ünitelerine (X) bağlanmaktadır ve her bağlantının bir ağırlığı (W) vardır. İki girdi ve bir çıktıdan oluşan tek katmanlı bir yapay sinir ağı Şekil 5 te verilmiştir.

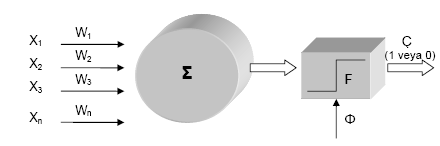


**Şekil 5:** Tek Katmanlı Algılayıcı Modeli (Öztemel,2006)

Bu ağlarda süreç elemanlarının değerlerinin ve dolayısıyla ağın çıktısının sıfır olmasını önleyen bir de eşik değeri (Φ) vardır ve değeri daima 1’dir. Ağın çıktısı ağırlıklandırılmış girdi değerlerinin eşik değeri ile toplanması sonucu bulunur. Bu girdi ile bir aktivasyon fonksiyonundan geçirilerek ağın çıktısı hesaplanır. Tek katmanlı algılayıcılarda çıktı fonksiyonu doğrusal bir fonksiyondur ve 1 veya -1 değerlerini almaktadır. Eğer çıktı 1 ise birinci sınıfa -1 ise ikinci sınıfa kabul edilmektedir.

**Basit Algılayıcılar**  
Basit algılayıcılar modeli 1950’li yılların sonlarında Rosenblatt tarafından ortaya atılmıştır. Esas olarak, basit algılayıcılar sınıflandırma amacıyla kullanılan, McCulloch ve Pitts tarafından geliştirilmiş daha karmaşık bir modeldir. Bir araya gelmiş sinir hücrelerinin miktarına bağlı olarak, basit algılayıcılar ile değişik sayıda sınıflandırma problemi çözebilir. Doğru bir sınıflandırma için sınıfların düzlemsel olarak ayrılması gerektiği gösterilmiştir (Nygren, 2004, Minsky ve Papert,1969).

Basit algılayıcılar bir sinir hücresinin birden fazla girdiyi alarak bir çıktı üretmesi prensibine dayanmaktadır. Ağın çıktısı bir veya sıfırdan oluşan mantıksal değerdir. Çıktının hesaplanmasında eşik değer fonksiyonu kullanılır. Ağın yapısı Şekil 6 da verilmektedir.

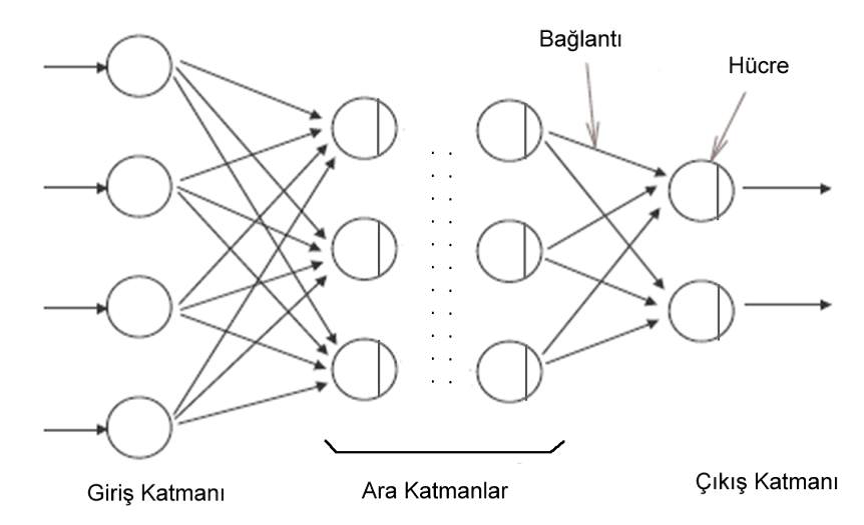


**Şekil 6**: Basit Algılayıcı Model (Elmas,2003)

**Adaline**  
1959’da, Stanford üniversitesinden Bernard Widrow, basit nöron benzeri elemanlara dayanan ve “Adaline” (Adaptive Linear Nöron) olarak adlandırılan bir adaptif lineer elemanı geliştirmiştir. Adaline yapısı tüm sinir ağlarının en basitidir ve öğrenme için danışmanlı öğrenmeyi kullanır. Adaline ve iki tabakalı biçimi olan “madaline” (Multiple Adaline); ses tanıma, karakter tanıma, hava tahmini ve adaptif kontrol gibi çok çeşitli uygulamalar için kullanılmıştır. Daha sonraları adaline, ayrık bir çıkı\_ yerine sürekli bir çıkış üretmek için geliştirilmiştir. Widrow, telefon hatları üzerindeki ekoları elimine etmeye yarayan adaptif filtreleri geliştirmede, adaptif lineer eleman algoritmasını kullanmıştır. Bununla ilk defa YSA’lar gerçek bir probleme uygulanmıştır. Adaline birçok uygulama için oldukça iyi çalışmasına rağmen lineer problem uzayıyla sınırlıdır. Lineer transfer fonksiyonu kullanırlar. Giriş ve istenilen çıkış desenlerinin tekrar tekrar ağa uygulanmasıyla eğitim gerçekleştirilir. Desenlerin doğru sınıflara ayrılmasıyla, hatalar minimize edilerek öğrenme gerçekleştirilir. Eğitimden sonra adaline, yeni girişleri kazandığı deneyime göre sınıflandırabilir.

**Çok Katlı Algılayıcılar (ÇKA)**  
Algılayıcı ve Adaline yöntemleri doğrusal olmayan çözümler üretemediği için hem mimari hem de eğitim algoritması açısından iyileştirilmiş Çok Katmanlı Algılayıcı (MLP) ağı önerilmiştir. Mimari açıdan doğrusal olmayan aktivasyon fonksiyonuna sahip birçok nöronun birbirine hiyerarşik olarak bağlandığı bir yapıya sahip olan MLP, Algılayıcı ve Adaline yöntemlerinin avantajları yanı sıra geri-yayılım adındaki öğrenme sistemini kullanmaktadır ve genel olarak yapay sinir ağları ileri beslemeli ve geri beslemeli ağlar olarak ikiye ayrılmaktadır.

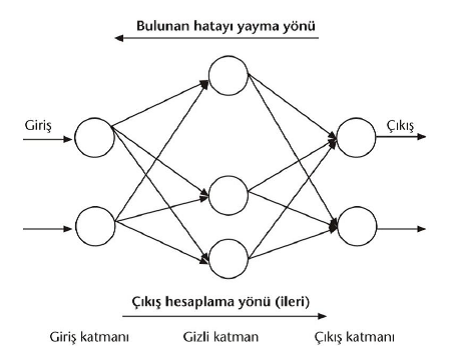
**İleri Beslemeli Ağlar**  
İleri beslemeli sinir ağları tek yönlü sinyal akışı için izin verir. Ayrıca, ileri beslemeli sinir ağları çoğu katmanlar halinde organize edilmektedir. Üç katmanlı ileri beslemeli sinir ağı bir örneği Şekil 7’de gösterilmiştir. Bu şebeke giriş düğümleri, iki gizli katman ve bir çıkış katmandan oluşur. Tipik aktivasyon fonksiyonları Şekil 4’te gösterilmiştir. Bu sürekli aktivasyon fonksiyonları katmanlı ağlar gradyan tabanlı eğitimi için izin verir.



**Şekil 7 :** İleri Beslemeli Sinir Ağlara Örnek(Öztemel, 2006)

İleri beslemeli yapay sinir ağında, hücreler katmanlar şeklinde düzenlenir ve bir katmandaki hücrelerin çıkışları bir sonraki katmana ağırlıklar üzerinden giriş olarak verilir. Giriş katmanı, dış ortamlardan aldığı bilgileri hiçbir değişikliğe uğratmadan ara (gizli) katmandaki hücrelere iletir. Bilgi, ara ve çıkış katmanında işlenerek ağ çıkışı belirlenir. Bu yapısı ile ileri beslemeli ağlar, doğrusal olmayan statik bir işlevi gerçekleştirir. İleri beslemeli 3 katmanlı yapay sinir ağının, orta katmanında yeterli sayıda hücre olmak kaydıyla, herhangi bir sürekli fonksiyonu istenilen doğrulukta yaklaştırabileceği gösterilmiştir. En çok bilinen algoritma olan geriye yayılım öğrenme algoritması, bu tip yapay sinir ağların eğitiminde etkin olarak kullanılmaktadır. Ağa, hem örnekler hem de örneklerden elde edilmesi gereken çıktılar (beklenen çıktılar) verilmektedir. Ağ kendisine gösterilen örneklerden genellemeler yaparak problem uzayını temsil eden bir çözüm uzayı üretmektedir. Daha sonra gösterilen benzer örnekler için bu çözüm uzayı sonuçlar ve çözümler üretebilmektedir.

**Geri Beslemeli Yapay Sinir Ağları**  
Geri beslemeli Yapay Sinir Ağları (YSA)’ da, en az bir hücrenin çıkışı kendisine ya da diğer hücrelere giriş olarak verilir ve genellikle geri besleme bir geciktirme elemanı üzerinden yapılır. Geri besleme, bir katmandaki hücreler arasında olduğu gibi katmanlar arasındaki hücreler arasında da olabilir. Bu yapısı ile geri beslemeli YSA, doğrusal olmayan dinamik bir davranış gösterir. Dolayısıyla, geri beslemenin yapılış şekline göre farklı yapıda ve davranışta geri beslemeli YSA yapıları elde edilebilir. Aşağıda bulunan Şekil 8’de iki katmanlı ve çıkışlarından giriş katmanına geri beslemeli bir YSA yapısı görülmektedir .



**Şekil 8**: Geri Beslemeli YSA (Kabalcı , 2015)

**Geri Yayılım Algoritması (BackPropagation)**  
Geri yayılım algoritması, sinir ağının denetimli sınıfına giren genel bir algoritmadır. Daha öncede belirtildiği gibi girişlerle çıkışlar arasındaki hata sinyali bulunarak, ağırlıklar bu hata sinyaliyle güncellenmektedir. Hata yani e(t), arzu edilen çıkış t(t) ile sinir ağının çıkışı y(t) arasındaki farktır.

e(t) = t(t) – y(k) ; t=1,..,m                                            (2)

Geri yayılım öğrenme algoritmasının temel yapısı, zincir kuralı kullanılarak, ağ üzerindeki tüm ağırlıklara E hata fonksiyonunun etkilerini yaymaktır. Böylelikle toplam hata değerini minimize etmektedir.

E\_Top= lim┬(t→∞)⁡((∑\_(t=1)^tE^t )/t)                   (3)

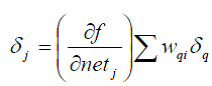
Herhangi bir ‘t’ denemesinde Et değeri küçültülebilirse sistemin hatasının azalacağı Eşitlik 3’te kolaylıkla gözlenmektedir. Sistem hatasındaki azalmayı temel olarak alarak ve bu azalmayı destekleyecek şekilde ağ üzerindeki ağırlık değerleri yeniden belirlenecek yöntemler kullanılmaktadır. Eğitme işlemi ve eğitimden sonraki test işlemi bu akışa göre yapılır. Bu algoritma ile xi. giriş için, i ve j kat işlem elemanları arasındaki ağırlıklardaki wji(t) değişikliği hesaplanır. Bu ifade,

[http://www.derinogrenme.com/wp-content/uploads/2017/02/4.png](http://www.derinogrenme.com/wp-content/uploads/2017/02/4.png)                                      (4)

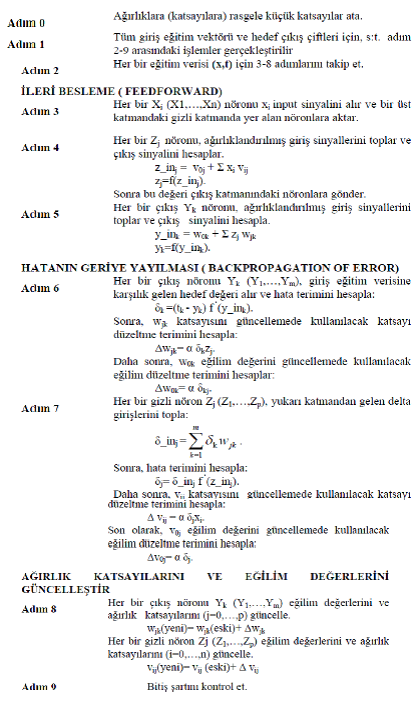
olarak verilir. Eşitlik 4’de η öğrenme katsayısı, α momentum katsayısı ve δj ara veya çıkış katındaki herhangi bir j nöronuna ait bir faktördür. Çıkış katı için bu faktör aşağıdaki şekilde verilir.

[http://www.derinogrenme.com/wp-content/uploads/2017/02/5.png](http://www.derinogrenme.com/wp-content/uploads/2017/02/5.png)                                   (5)

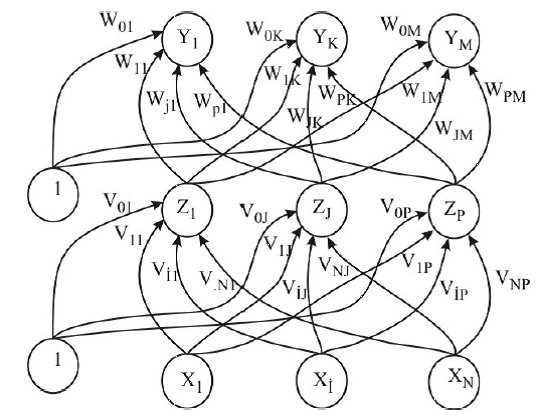
Burada yj(t) ise j işlemci elemanının hedef çıkışıdır. Ara katlardaki (İşlem Elemanları – Nöronlar ) İşlem Elemanları (İE) için ise bu faktör,

[](http://www.derinogrenme.com/wp-content/uploads/2017/02/6.png)                                             (6)

olarak verilir. Ara katlardaki İE’ler için herhangi bir hedef çıkış olmadığından, Eşitlik 5 yerine Eşitlik 6 kullanılır. Bu duruma bağlı olarak çıkış katından başlayarak δj faktörü, bütün katlardaki İE’ler için hesaplanır. Daha sonra Eşitlik 4’deki formüle bağlı olarak, bütün bağlantılar için ağırlıkların güncelleştirilmesi gerçekleştirilir. Geri yayılım algoritmasında kullanılacak aktivasyon fonksiyonu birkaç önemli karakteristiğe sahip olmalıdır. Aktivasyon fonksiyonu, sürekli, türevi alınabilir ve tekdüze bir şekilde azalmayan bir fonksiyon olmalıdır. Bu fonksiyonun türevinin kolay alınıyor olması tercih sebebidir. Genellikle, fonksiyonun minumum ve maksimum asimtotlar arasında uzanması beklenir.



**Şekil 9** : Geri Yayılma Algoritması(Yüksek, 2007)



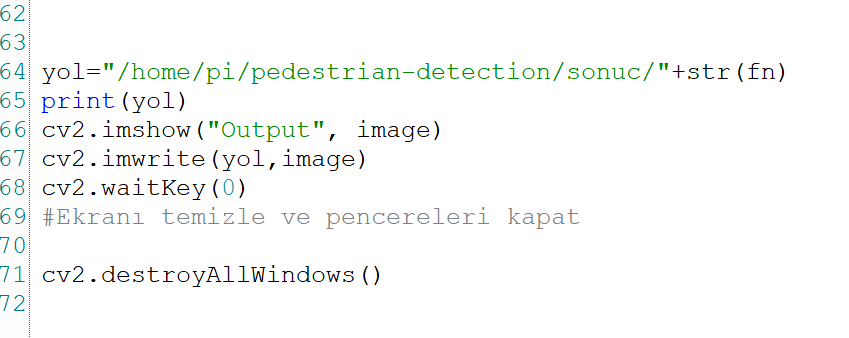
**Şekil 10** : İleri beslemeli çok katmanlı sinir ağı: x, z, y vektörleri ile V, W matrisleri (Yüksek vd, 2007)

**PIR KODU**



**GÖRÜNTÜ İŞLEME KODU**

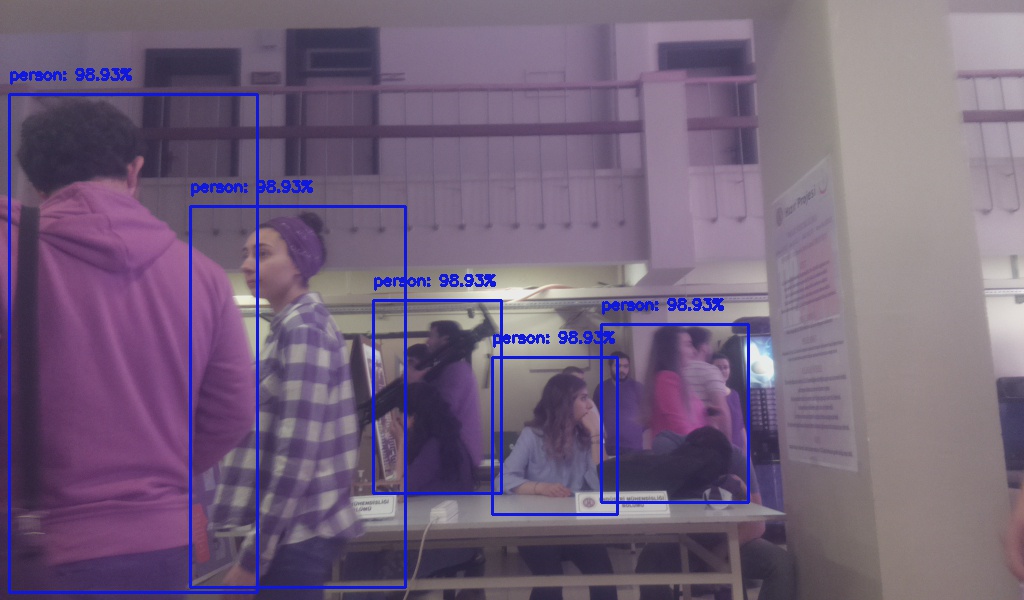
****

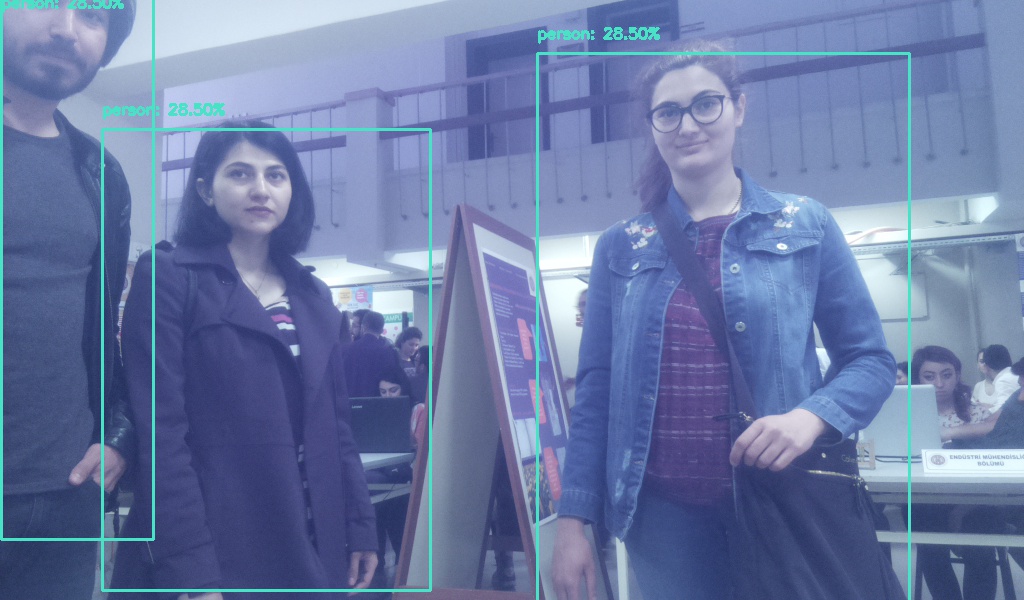


**DEMO ÇALIŞMALARI**

****

****

****

****

# Kaynaklar

1-)Projenin maliyet hesaplarının yapıldığı ve ürünlerin satın alındığı mağaza:

<<https://www.robotistan.com/sensor>>

2-) Bu araştırma ufak çaplı olup PIR sensorunun arduino üzerine entegre edilerek çalışma mantığının kavranması açısından faydalı oldu.

<<http://www.projehocam.com/pir-sensoru-ile-hareket-dedektoru-arduino/>>

3-) Raspery pi üzerine PIR sensorunun entegrasyonunun yapılması amaçlı GPIO(General Purpose Input/Output) datasheet incelenerek entegre edilmesini sağladık.

<<https://diyhacking.com/raspberry-pi-gpio-control/>>

4-) PIR sensoru kullanılarak yüz tanıma

National Conference on Product Design (NCPD 2016), July 2016

Mr. Ashwin K Kashyap, Mr. Abhishek S, Mr. Abhishek S A, Ms. Ratnakumari. C UG Students, Dept. of CSE, PESITM, Shivamogga

<<http://mobilityresearchforum.com/ncpd2016/wp-content/uploads/2016/08/NCPD2016_paper_37.pdf>>

Raspberry Pi details:

<[www.raspberrypi.org/](http://www.raspberrypi.org/)>

Passive infrared sensor kullanımı:

<[https://en.wikipedia.org/wiki/Passive\_infrared\_ sensor/](https://en.wikipedia.org/wiki/Passive_infrared_%20sensor/)>

Raspberry Pi Camera Module:

< [www.raspberrypi.org/products/camera-module/](http://www.raspberrypi.org/products/camera-module/)>

<<https://www.python.org/>>

<<https://www.reddit.com/>>

Facial Recognition: OpenCV On The Camera Board:

<[www.raspberrypi.org/blog/facial-recognitionopencv-on-the-camera-board/](http://www.raspberrypi.org/blog/facial-recognitionopencv-on-the-camera-board/)>

Haar Cascades:

<<http://alereimondo.no-ip.org/OpenCV/34>>

5-)Ultrasonic mesafe sensoruyle alakalı ufak çaplı bir proje yaparak çalışma mantığının ve arduino üzerinde entegresinin kontrolü sağlandı.

<<http://www.toptechboy.com/arduino/lesson-20-arduino-lcd-project-for-measuring-distance-with-ultrasonic-sensor/>>

6-)Ultrasonic mesafe sensorunu Raspery pi üzerinde entegrasyonu için datasheet araştırması:

<<https://www.maxbotix.com/Raspberry-Pi-With-Ultrasonic-Sensors074/>>

7-)Canlı yayın yapan bir kameradaki görüntüden hareketi algılamak için yaptığımız araştırma makalesi:

Proceedings of the World Congress on Engineering and Computer Science 2012 Vol I WCECS 2012, October 24-26, 2012, San Francisco, USA

8-)Deep learning makalesinden insan yüzü ayırt etme üzerinde araştırdığımız makaleler:

Bastien F., Lamblin P., Pascanu R., Bergstra J., Goodfel-low I. J., Bergeron A., Bouchard N., Bengio Y.: Theano:new features and speed improvements. Deep Learning and Unsu-pervised Feature Learning NIPS 2012 Workshop, 2012.

<<https://arxiv.org/pdf/1211.5590.pdf>>

Ojala T., Pietikainen M., Harwood D. ̈ : A comparative study of texture measures with classification based on featured distribu-tions.

< <http://www.umiacs.umd.edu/~raghuram/ENEE731/Texture/lpbslides.pdf>>

<<https://www.cs.toronto.edu/~ranzato/publications/taigman_cvpr14.pdf>>

9-)Yüz tanıma ve nesne ayırt etme için algoritma araştırmalarının yapılması:

<<https://www.researchgate.net/profile/Musa_Peker/publication/272747896_A_Real-Time_and_Motion-Sensitive_Security_Application_with_Face_Recognition/links/54ed68440cf27fbfd7724d72/A-Real-Time-and-Motion-Sensitive-Security-Application-with-Face-Recognition.pdf>>

<<http://acikerisim.deu.edu.tr/xmlui/bitstream/handle/12345/7922/305570.pdf?sequence=1&isAllowed=y>>

<<http://www.iaeng.org/publication/WCECS2012/>>

<<http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp634-639.pdf/>>

<<http://www.iaeng.org/publication/WCECS2012/WCECS2012_pp1110-1115.pdf/>>

10-)Yüz tanıma algoritmalarını araştırırken kullandığımız makaleler ve kaynaklar:

Viola-Jones face recognition algorithm:

<<http://www.ipol.im/pub/art/2014/104/article.pdf>>

EigenFaces and FisherFaces face recognition algorithm:

<<http://www.cs.haifa.ac.il/~rita/visual_recog_course/talks/Eigenfaces%20vs%20Fisherfaces.pdf>>

Opencv de açık kaynak kod araştırması:

<<https://docs.opencv.org/3.1.0/index.html>>

11-)Google : Andrew G. Howard , Menglong Zhu , Bo Chen , Dmitry Kalenichenko , Weijun Wang , Tobias Weyand , Marco Andreetto , Hartwig Adam

12-).Uijlings, JR, Van de Sande, KE, Gevers, T., Smeulders, AW: Nesne tanıma için seçici arama. IJCV 104 , 154 (2013)

CrossRef Google Akademik

13-).Ren, S., He, K., Girshick, R., Sun, J .: Daha hızlı R-CNN: bölge teklif ağları ile gerçek zamanlı nesne tespitine doğru. İçinde: NIPS (2015)

Google Akademik

14-).O, K., Zhang, X., Ren, S., Sun, J .: Görüntü tanıma için derin artık öğrenme. İçinde: CVPR (2016)

Google Akademik

15-).Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y .: Aşırı ateş: Bütünleştirme, yerelleştirme ve konvolüsyon ağlarını kullanarak algılama. İçinde: ICLR (2014)

Google Akademik