



# İnsan ve Araç Algılamama

HAZIRLAYAN: EMRULLAH HOCAOĞLU

2016 BAHAR – SAYISAL İŞARET İŞLEME

# İçerik

- ▶ Computer Vision hakkında
- ▶ Matlab CV kütüphanesi
- ▶ Matlab vision.PeopleDetector ile insan algılama
- ▶ Matlab ile yüz algılama
- ▶ Matlab ile video'da hareket eden araçları algılama

# Computer Vision (Bilgisayar Görüşü)

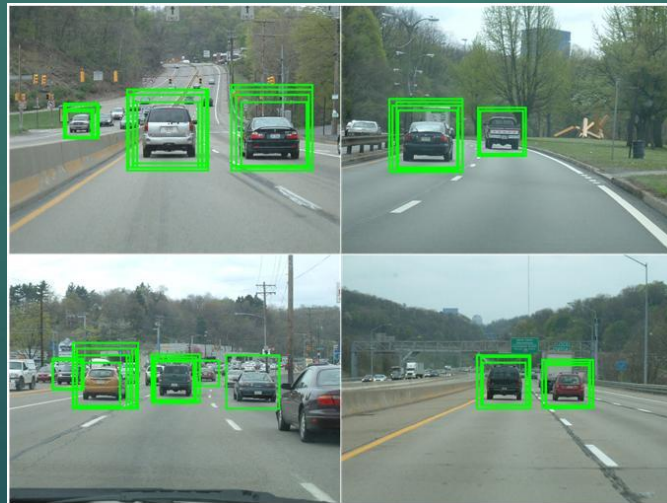
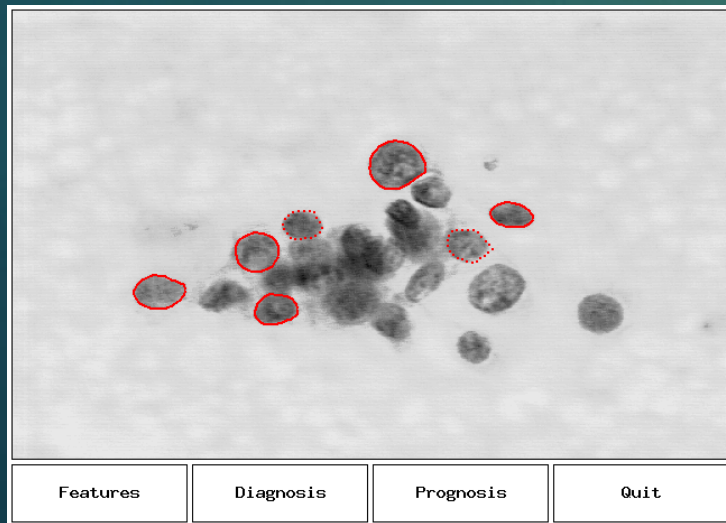
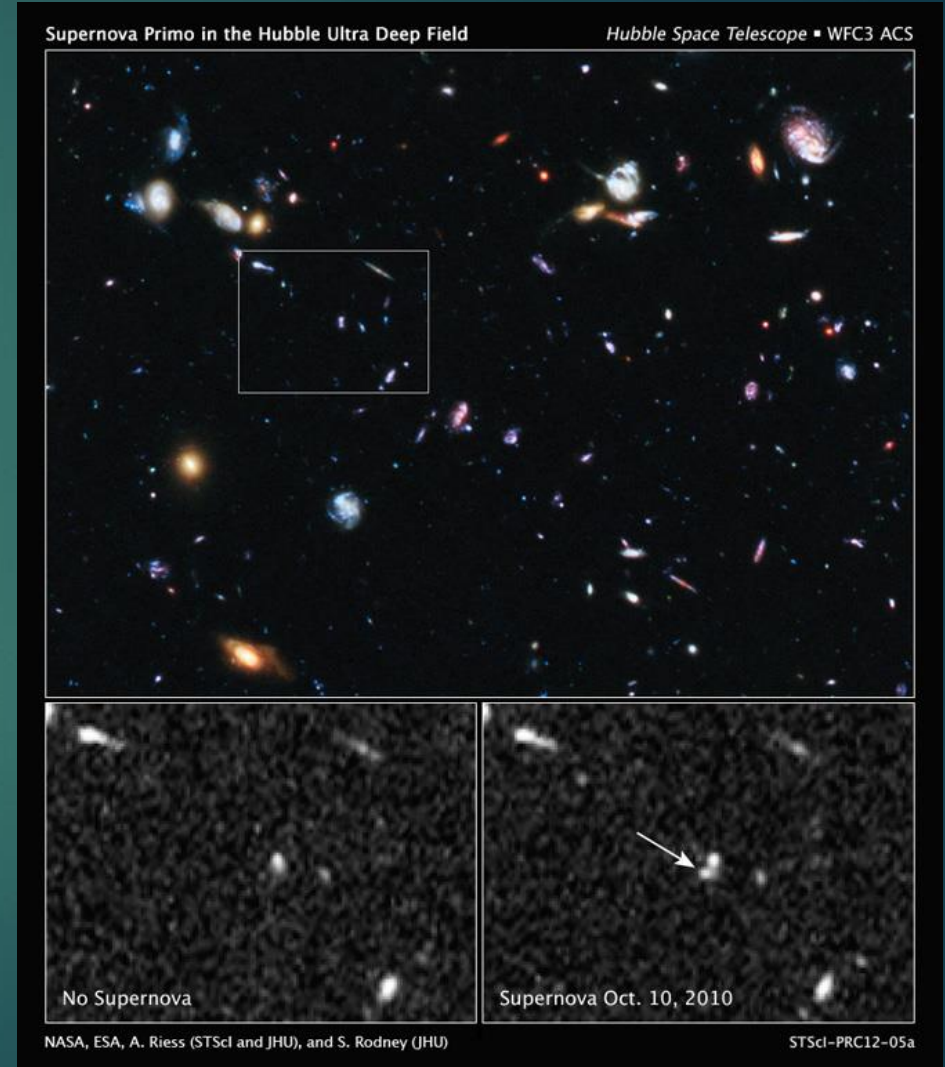
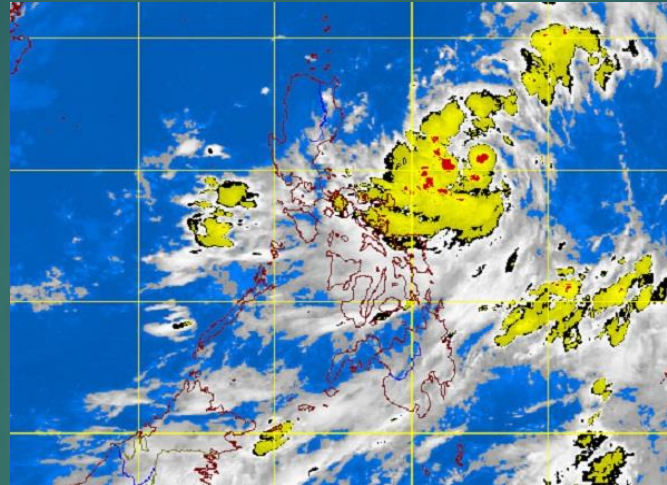
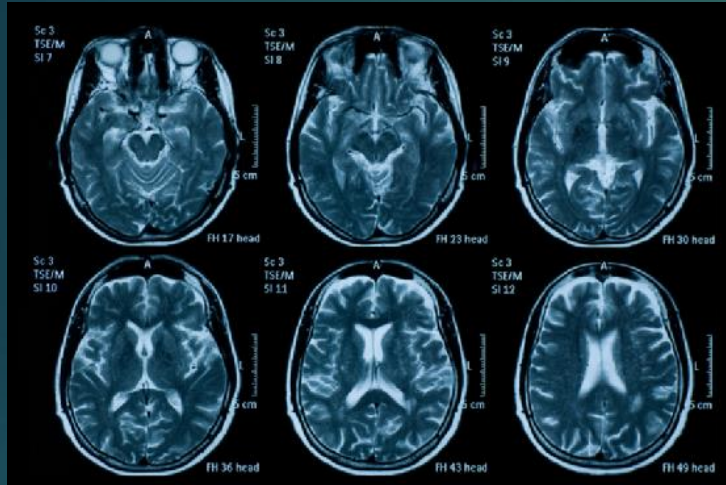
Wikipedia'ya göre computer vision:

Resimleri edinme, işleme, analiz etme, anlama –genel olarak çok boyutlu veriler - metodlarını numerik ve sayısal veriler elde etmek için kullanan bir alandır.

Neden „Computer Vision?“

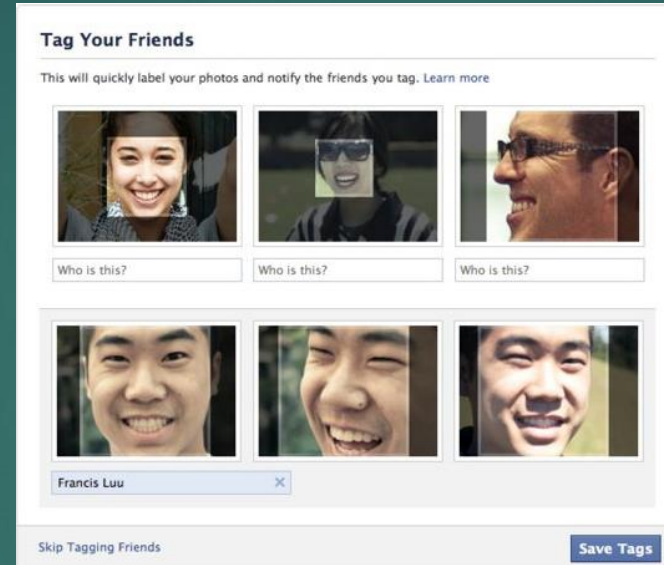
- ▶ Resimler ve videolar artık her tarafta –sosyal ağlar-,
- ▶ Artan iş yoğunluğunu otomasyona çevirebilmek,
- ▶ Güvenlik mekanizmalarını,
- ▶ Bilimsel verilerin hızlı analizi vb.

# Computer Vision kullanım örnekleri





# Computer Vision kullanım örnekleri



# Matlab ile İnsan Algılama

```
resim_algilayici = vision.PeopleDetector;  
resim = imread('resim/yolda_insan_01_kirpilmis.jpg');  
  
[bboxes,scores] = step(resim_algilayici, resim);  
  
res = insertObjectAnnotation(resim, 'rectangle',bboxes,scores);  
figure, imshow(res)  
title('İnsan Algılama ve Cerceve Cizimi');
```

# İnsan Algılama - Sonuç





# Adım 1:

`resim_algilayici = vision.PeopleDetector`

- ▶ Bu fonksiyon bir resimden „Odaklı Geçişlerin Histogramı“ (**histogram of oriented gradients**) metodunu ve eğitilmiş bir „destek vektör makinesi“ (**Support vector machine**) sınıflandırıcısı kullanarak „insanları“ algılar.
- ▶ Bu obje, dik konumda önü kapatılmamış insanları algılar.
- Support Vektor Machine: Makine öğrenmesinde, veriyi sınıflandırma ve regresyon analizi için kullanmak üzere analiz eden öğrenme metodu ile ilişkili bir denetimli öğrenmedir.

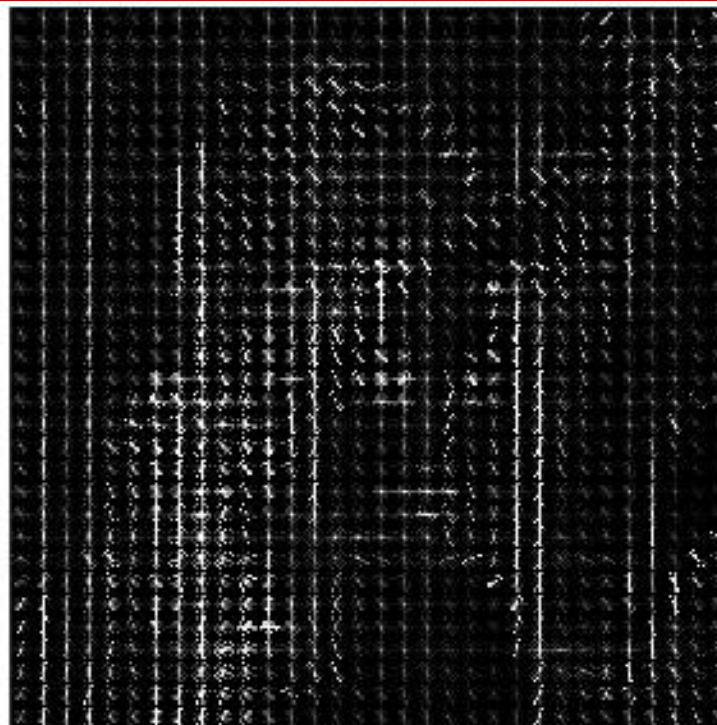


# Odaklı Geişlerin Histogramı HOG

Girdi Resmi



Odaklı Geişlerin Histogramı



# Adım 2:

```
[bboxes,scores] = step(resim_algilayici, resim);
```

- ▶ **bboxes** : Algılanmış nesnelerin resimdeki korninat, genişlik ve yüksekliklerini içeren matris.
- ▶ **scores** : Nesne algılamanın doğruluk bilgisini gösteren sayı vektörü. Algılamadaki doğruluk ile sayısal ifade arasında doğrusal bir orantı vardır. (özgüven)
- ▶ **step() fonksiyonu** : HOG özelliklerini kullanarak ayakta duran insanları algılayan fonksiyon.
- Resimde belli bir bölgede insan algılaması yapmak istiyorsak, step fonksinuna 4 elemanlı bir vektörü 3. bir girdi olarak vermeliyiz. [ x y genislik yükseklik] şeklinde.

# Adım 3: Görseli Çizdirme

```
res = insertObjectAnnotation(resim, 'rectangle', bboxes, scores);  
imshow(res);  
title('İnsan Algılama ve Cerceve Çizimi');
```

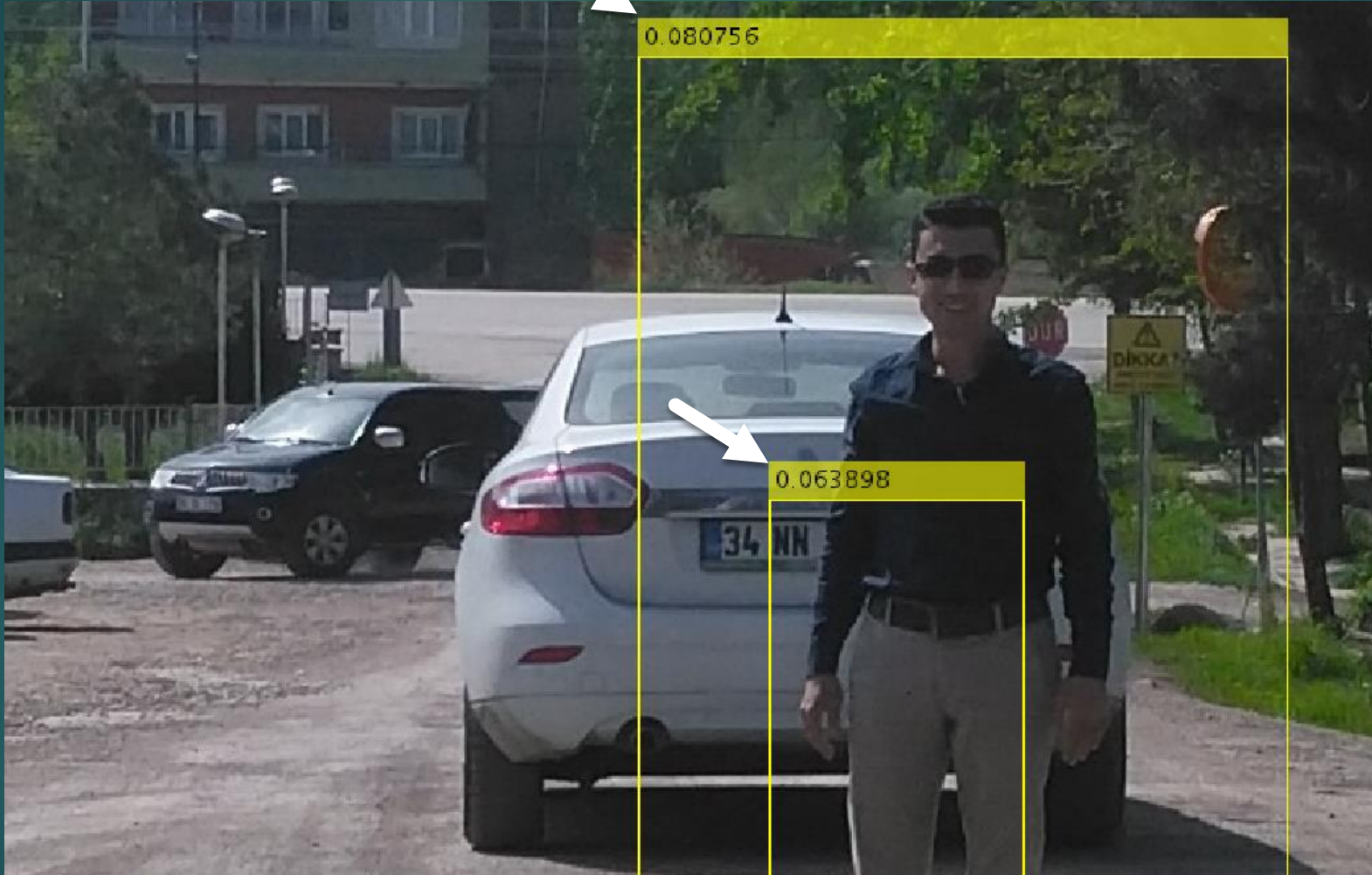
Nesne Pozisyon  
Matrisi

Etiket Vektörü

- ▶ `insertObjectAnnotation()`: istenilen şekil ve etiketli resmi çizdirir.
- ▶ Ek parametreler ad, değer şeklinde ikili olarak verilir. Örneğin ....  
,scores, 'TextColor', 'black');
- ▶ Döndürdüğü değer RGB truecolor bir resimdir.
  - ❑ Annotation: not, dipnot
  - ❑ Truecolor: 24 bit renk derinliği

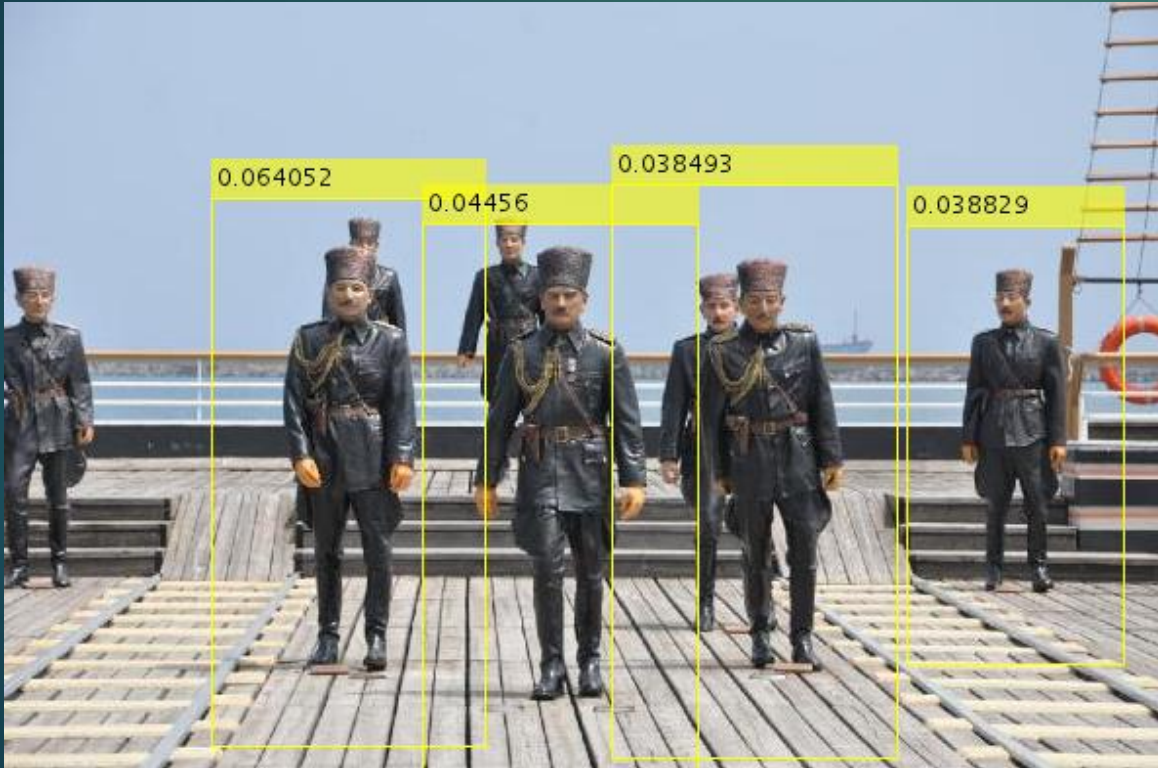


# Algılamada Hata



Hatalı algılanan karenin skoru daha düşük gösteriliyor.

# Dikkat: önü kapanmamış & „dik“ duran insan



Önü kapanmış figürler tespit edilemiyor.



Dik durmayan insanlar algılanamıyor.

# Yüz Algılama İle İnsan Algılama

```
resim = imread('resim/yolda_insan_01_kirpilmis.jpg');

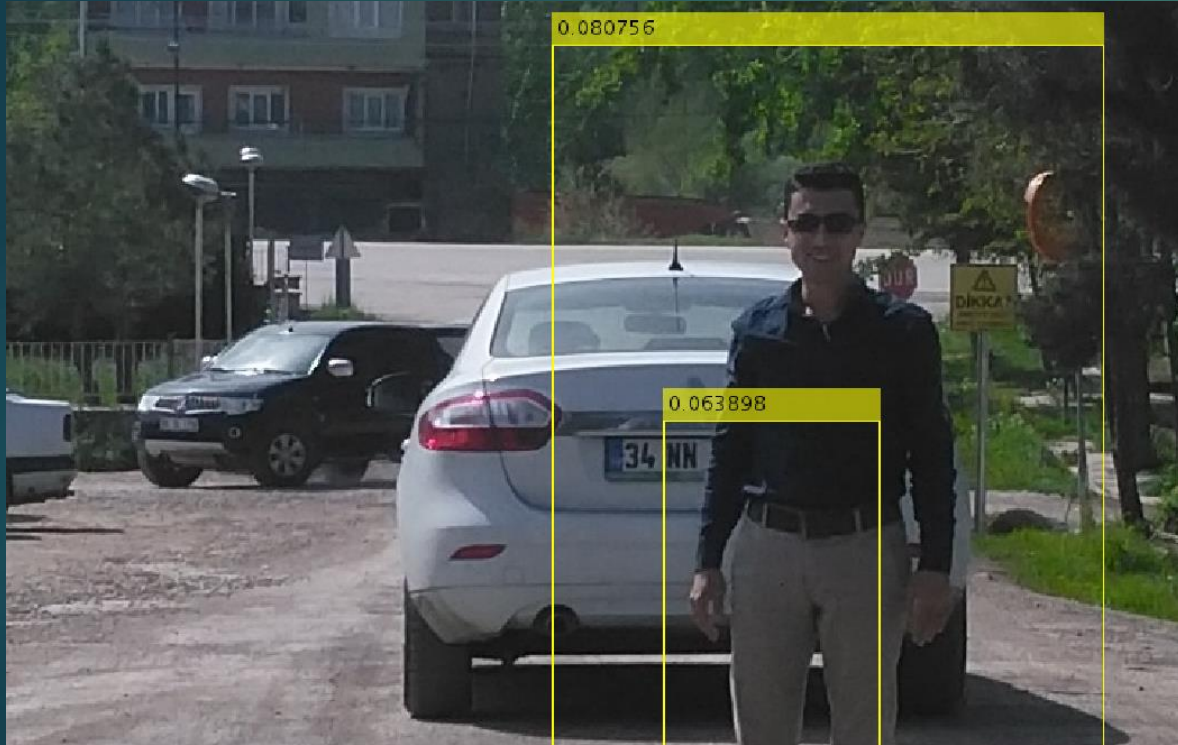
yuzAlgilayici = vision.CascadeObjectDetector;
kutucuklar = step(yuzAlgilayici, resim);

res = insertObjectAnnotation(resim, 'rectangle', kutucuklar, 'yuz');
imshow(res);
```

- ▶ vision.CascadeObjectDetector():  
Parametre almadığında öntanımlı olarak yüz algılıyor. (FrontalFaceCART)
- ▶ Alabildiği diğer parametreler:  
Gövde Üstü (Upper Body), Göz Çifti (Eye Pair), Tek Göz(Single Eye), Profil Yüz(Profile Face), Dudak (Mouth), Burun (Nose)



# Yüz Algılama – İnsan Algılama



# Yüz Algılama – İnsan Algılama



Tüm yüzlere ek olarak fazladan bir yüz algılanmış.



# Araç Algılama: Görüntü Parçalama ve Damla Analizi

Ne yapacağız?

1. Sabit bir arka plan bulmaya çalışacağız.
2. Arka planımızı gürültüden arındıracağız.
3. Her bir resim çerçevesini arka plan ile karşılaştıracacağız.
4. Değişikliğin boyutu bizim arzu ettiğimiz değerde ise „arabayı bulmuş“ olduğumuzu kabul edeceğiz.



# Adım 1:

## Arka Plan Tespiti

```
arkaPlanAlgilayicisi = vision.ForegroundDetector('NumGaussians', 3, 'NumTrainingFrames', 50);  
  
videoOkuyucusu = vision.VideoFileReader('resim/arac_yolda_2.avi');  
  
for i = 1:50  
    cerceve = step(videoOkuyucusu); % sonraki frame  
    arkaPlan = step(arkaPlanAlgilayicisi, cerceve);  
end  
  
imshow(arkaPlan); title('Arka Plan');
```

Katıştırmada ki Gaussian mod sayısı

Öğrenmede kullanılacak Çerçeve sayısı

- ❑ NumGaussians: Tipik olarak 3, 4, 5 gibi sayılar alır. Çoklu arka plan modları için bu değer 3 ve yukarısı olmalıdır. Ön tanımlı değer: 5
- ❑ NumTrainingFrames: Başlangıçta arka planı algılayabilmek için kullanacağımız video çerçeve sayısıdır. Ön tanımlı değer: 150.

# Adım 1:

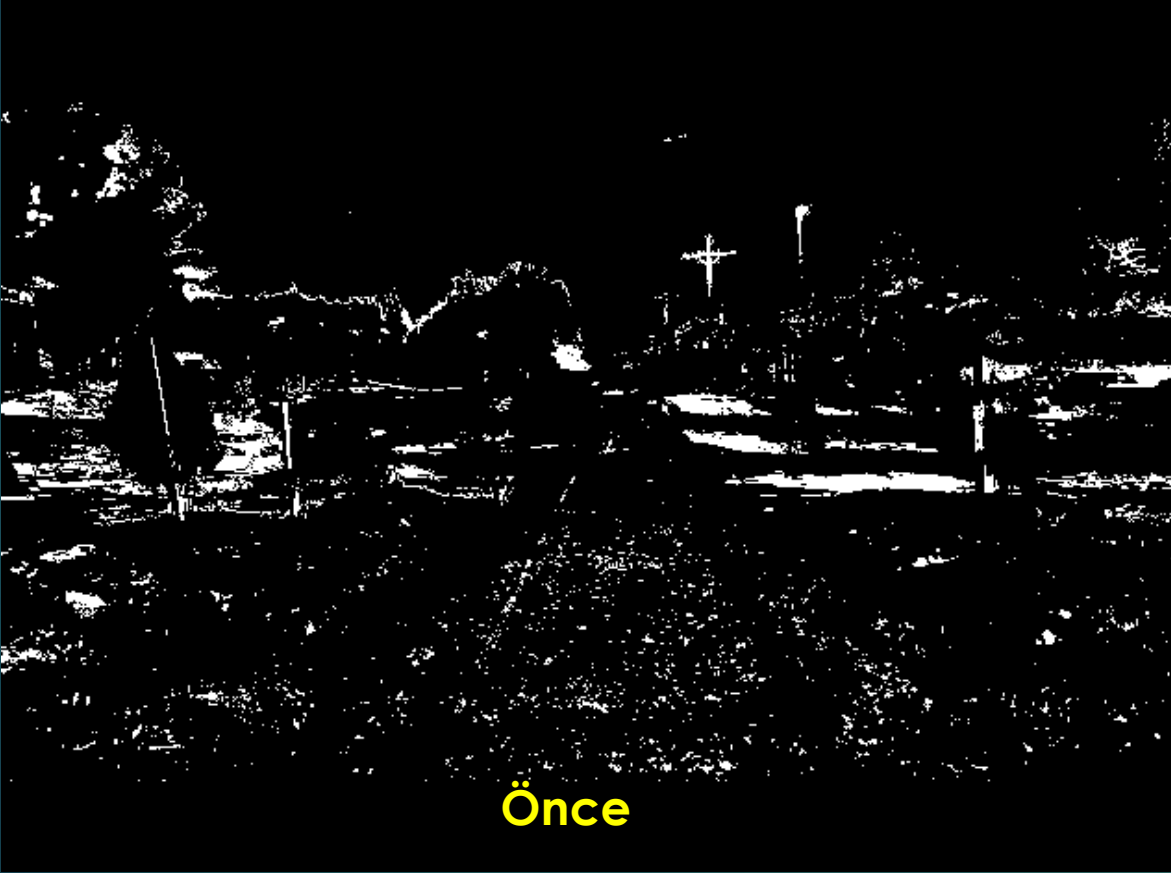
## Arka Plan Tespiti - Temizlik

```
se = strel('square', 3);  
temizArkaPlan = imopen(arkaPlan, se);  
figure; imshow(temizArkaPlan); title('Temiz Arka Plan');
```

- ❑ **strel('square', 3)**: Kenarı 3 pixel olan kare bir yapı elementi oluşturur. Böylece 3x3 pixelden daha küçük görüntülerden kurtulacağız.  
Square, circle, rectangle vb. Parametreler alabilir.
- ❑ **imopen(arkaPlan, se)**: arkaPlan ögesini, 3x3'den daha küçük öğelerden arındırarak dönderir.

# Adım 1:

## Arka Plan Tespiti – Gürültü Temizleme





# Adım 2:

## Hareketli Nesneleri Algılama

```
damlaAnalizi = vision.BlobAnalysis('BoundingBoxOutputPort', true, ...  
    'AreaOutputPort', false, 'CentroidOutputPort', false, ...  
    'MinimumBlobArea', 500);  
kutucuklar = step(damlaAnalizi, temizArkaPlan);  
  
sonuc = insertShape(cerceve, 'Rectangle', kutucuklar, 'Color', 'green');  
figure; imshow(sonuc); title('Algılanmış Arac');
```

► vision.BlobAnalysis();

Ad, değer sıralaması ile ilgili parametre alır.

MinimumBlobArea ile algılamak istediğimiz nesnenin büyüklüğünü seçebiliriz.

# Matlab Çalışması



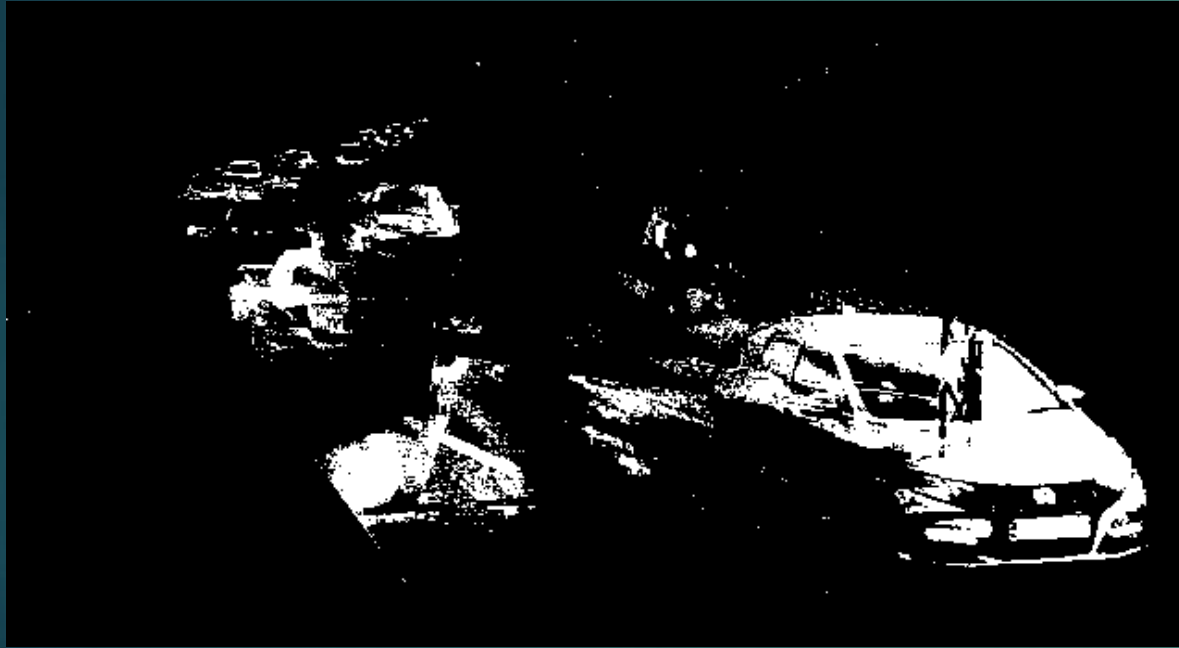
Kamerada titreme olmasından dolayı sabit nesnelerde yer değiştireyormuş gibi algılandı.

# Örnek Video 1

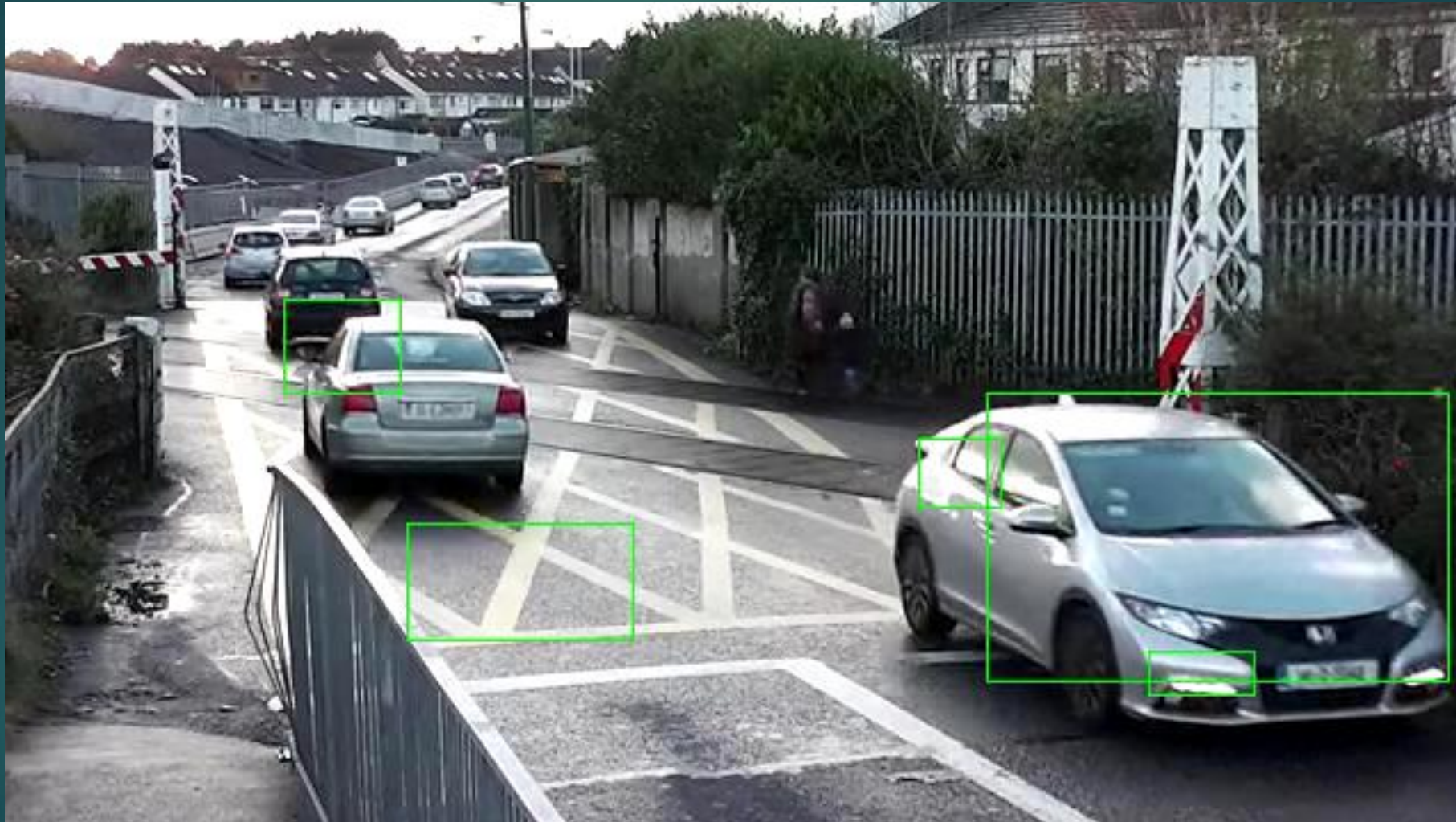




# Örnek Video 2 – Arka Plan



# Örnek Video 2



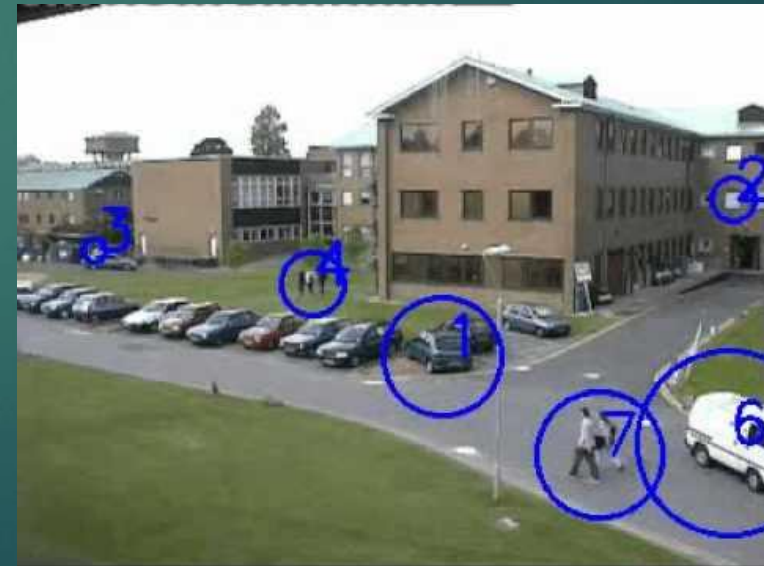
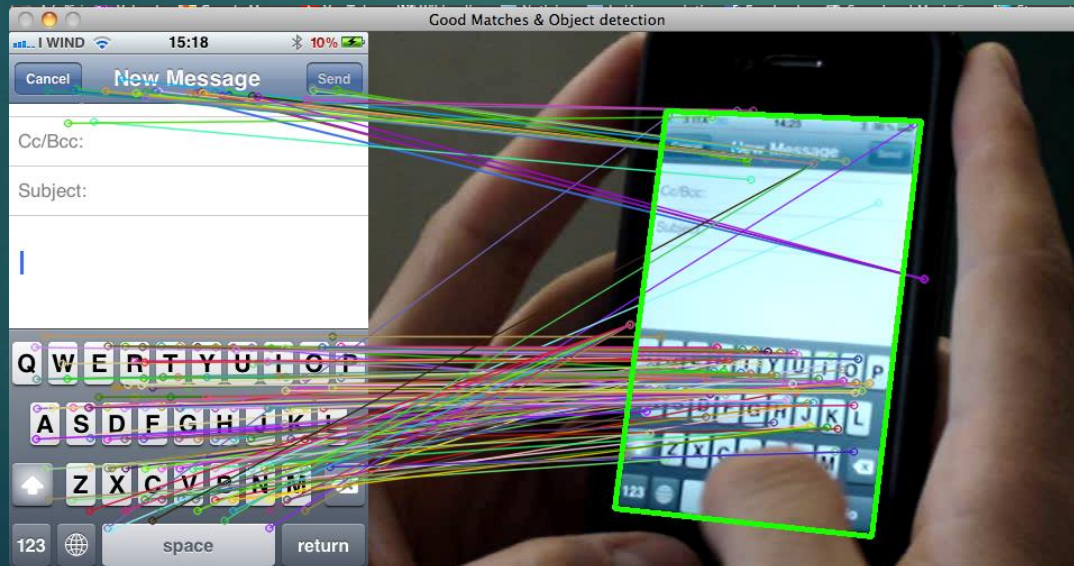
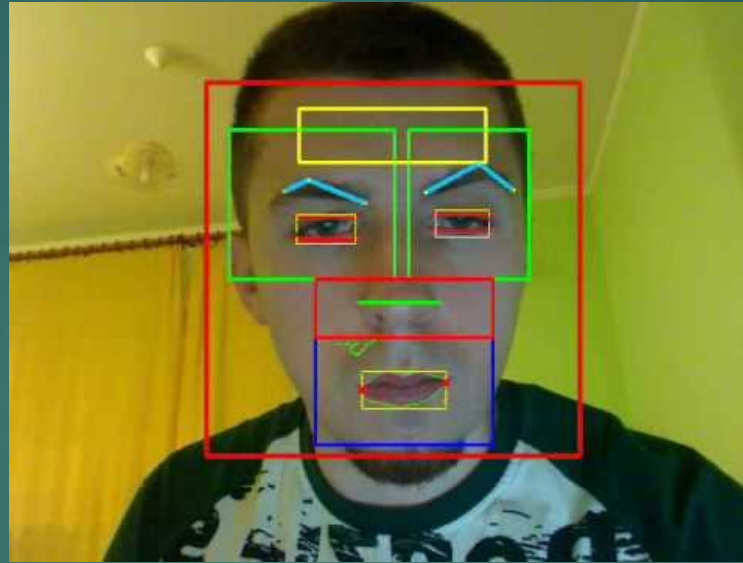
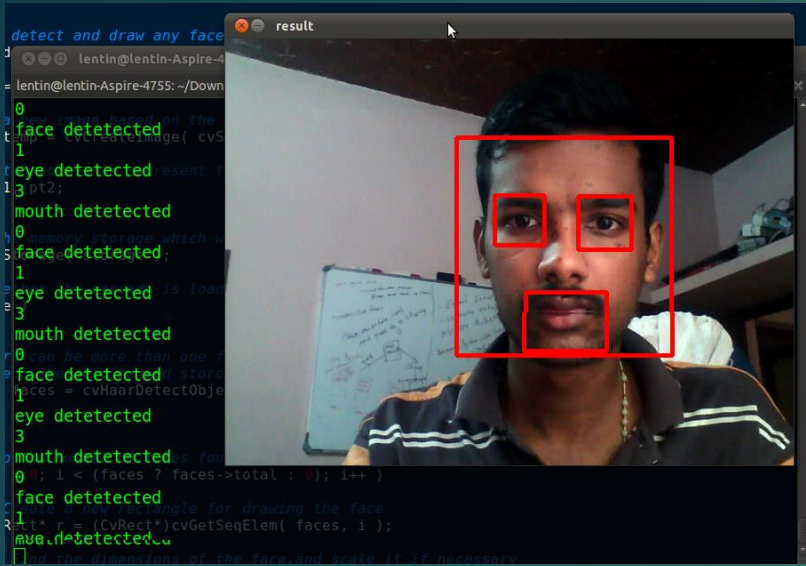
# OpenCV Kütüphanesi



- ▶ Açık kaynaklıdır.
- ▶ C++, Java, Python gibi dillere arayüzü vardır.
- ▶ Desteklediği diller sebebiyle IOT cihazlarda tercih edilebilir.
- ▶ Neleri destekliyo?
  - ▶ Resim ve video işleme,
  - ▶ Makine öğrenmesi,
  - ▶ Nesne algılama,
  - ▶ Derin Nöron Ağları,
  - ▶ Bulanık matematik ile resim işleme vb.



# OpenCV





# Kaynaklar

- ▶ [https://en.wikipedia.org/wiki/Computer\\_vision](https://en.wikipedia.org/wiki/Computer_vision)
- ▶ <https://cs.brown.edu/courses/cs143/lectures/01.pdf>
- ▶ [http://cn.mathworks.com/help/pdf\\_doc/vision/vision\\_ug.pdf](http://cn.mathworks.com/help/pdf_doc/vision/vision_ug.pdf)
- ▶ <http://www.mathworks.com/help/vision/ref/vision.peopledetector-class.html>
- ▶ <https://suryatejacheedella.wordpress.com/2014/07/28/car-detection-in-matlab/>