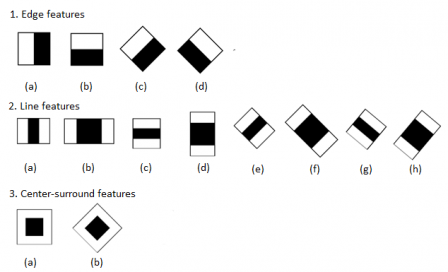
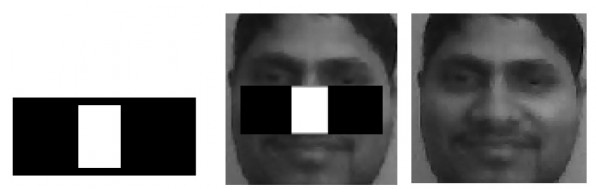
# Haar Cascade (Özel Benzer Özellikler) sınıflandırıcısı

En temel manada belirli bir algoritmaya göre bulunması istenen nesneler önce bilgisayara tanıtılır ve daha sonra ona benzer şekillerin bulunduğu resimler veya video frameleri taranarak o nesne bulunmaya çalışılır.

Eğitim için içerisinde aranılan nesnenin bulunduğu pozitif resimlere ve içerisinde o nesnenin bulunmadığı negatif resimlere ihtiyaç vardır. Sınıflandırıcı eğitimde, pozitif resimlerdeki nesneleri aşağıdaki gibi belirli büyüklerde ayarlanmış çerçevelerle tarayarak çerçeve içerisinde bulunan siyah bölgedeki piksel değerleri toplamı ile beyaz bölgedeki piksel değerleri toplamlarından karanlık aydınlık değerler kontrol edilerek belirli hedef değerler oluşturulur.

Feature denilen bu çerçevelere zayıf sınıflandırıcılar denilmektedir. Çünkü tek başına doğru bir sınıflandırıcı olamazlar. Bir nesnede bu zayıf sınıflandırıcılardan birçoğu olacaktır ve bu zayıf sınıflandırıcıların toplandığı noktada büyük doğruluk oranıyla aranılan nesne var demektir. Sınıflandırıcı, en temel mantığıyla bu şekilde çalışmaktadır.

Çerçeveler aşağıdaki gibi örnek pozitif resimler üzerinde taranır.



Yukarıdaki çerçeve için yanakların parlaklık oranının burun bölgesindeki parlaklık oranından daha düşük olması ile burun kısmı seçilebilir.



Aynı zamanda göz bölgesinin beyaz ile gösterilen alt bölgeden daha karanlık olması da bu özelliklerden bir tanesidir. Haarcascade sınıflandırıcısında, buna benzer birçok özellikler içinde nesnenin bulunduğu resimler üzerinden geçirilerek değerler oluşturulur. Örneğin yüz taramasında ağız, burun, alın, saç gibi bölgelerde birçok karanlık aydınlık özellikleri oluşturulacaktır. Bunların her birinden hedef değerler oluşturulmaktadır. Ve bu işlem çerçeve büyüklükleri değiştirilerek diğer aşamalarda tekrar edilmektedir.

Bu çerçeveler (zayıf sınıflandırıcılar) her resim boyutu için düşünüldüğünde yüz binlerce çekirdek oluşacaktır. Negatif resimler üzerinde tarama yapılarak içinde nesne bulunmadığı için kullanılmayacak olan çerçevelerin büyük çoğunluğu elenecektir. Pozitif resimlerde nesne seçilerek nesnedeki kullanılacak çerçeveler belirlenecektir. Bunun için eğitim sırasında pozitif resimlerde nesnenin milimetrik seçimine dikkat edilmelidir. Pozitif ve negatif resim örneklerinin çok olması istenilen nesnede daha iyi sonuçlar almak için önemlidir.

Bu işlemlerin hem eğitimde hem de nesnelerin bulunmasında bilgisayarı çok yoracağı ve işlemlerin uzun süreceği düşünülebilir. Reel time görüntü işlemede hız çok önemlidir. Haarcascade sınıflandırıcısında öncelikle resimlerin integralleri alınır. Böylelikle piksel değerlerinin tek tek toplamları hesaplanmak yerine integralle hesaplanmış olmaktadır. Böylelikle bilgisayardan büyük bir işlem gücü kaldırılacaktır.

Ayrıca nesnelerin bulunması aşamasında da her bir büyüklükteki çerçeve tarafından tekrar tekrar taranmak yerine sadece önceki aşamalarla eşleşme olan kısımlar taranarak bir işlem yükü de oradan azaltılmaktadır. Bu açılardan ve yaptığı iş açısından oldukça hızlı oldukları söylense de uygulama aşamasında reel time çalışıldığında alınan görüntüyü belirli oranda yavaşlattığı görülmektedir. Bu hız, sınıflandırıcının eğitilme şekli, örnek sayısı gibi durumlara göre değişmektedir.

İstenilen nesnelerin bulunabilmesi için sınıflandırıcının referans alacağı min hit rate ve max false alarm rate değerleri vardır. sınıflandırıcı eğitimde her seferinde belirli eğitim algoritması ile bu değerlere ulaşmaya çalışır. minhitrate minimum isabet oranını ve max false alarm rate ise hangi hata oranındaki nesnelerin gösterileceğini ifade eder.

Eğitim yapıldıktan sonra .xml uzantılı bir dosya oluşturulacak ve bu dosya ile opencv kütüphanesi kullanılarak istenilen nesne bulunabilecektir.

## Haar Cascade Kullanımı

Haar Cascade eğittikten sonra çıkan .xml uzantılı dosyayı veya internetten indirdiğiniz hazır cascade opencv yüklüyoruz

detector = cv2.CascadeClassifier(xml\_path)

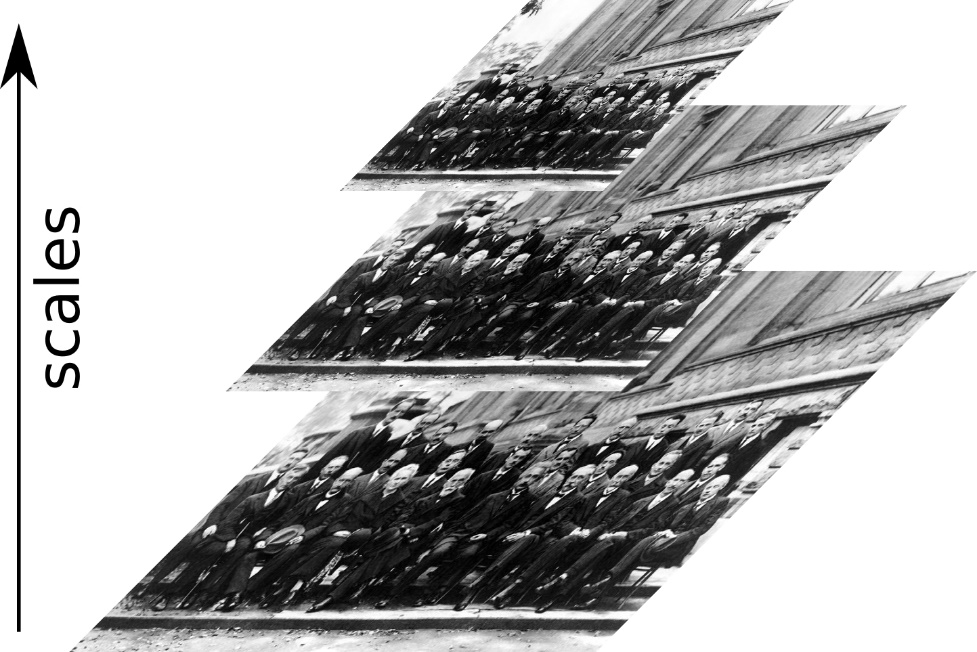
Haar cascade yüklendikten sonra detectMultiScale fonksiyonu sayesinde nesne tespiti yapılabilinir.

results = detector.detectMultiScale(image, scaleFactor=1.05, minNeighbors=5, minSize=(30, 30))

detectMultiScale fonksiyonu değer olarak x,y koordinatları ile dikdörtgenin w,h boyutlarını döndürür

Haar cascadeler false positive değerler vermeye çok eğilimlilerdir bu yüzden iyi bir şekilde parametre tuning ile bu riski azaltmayı deneyebilirsiniz.

ScaleFactor: Her görüntü ölçeğinde görüntü boyutunun ne kadar küçültüleceğini belirten parametre.

Scale Factor, Scale Piramiti oluşturmak için kullanılır Kısacası. Modelinizin eğitim sırasında tanımlanmış sabit bir boyutu vardır. Bu, bu boyutta bir yüz varsa görüntüde algılandığı anlamına gelir. Bununla birlikte, giriş görüntüsünü yeniden ölçeklendirerek, daha büyük bir yüzü daha küçük bir yüze doğru yeniden boyutlandırabilir ve algoritma için algılan abilir hale getirebilirsiniz. 

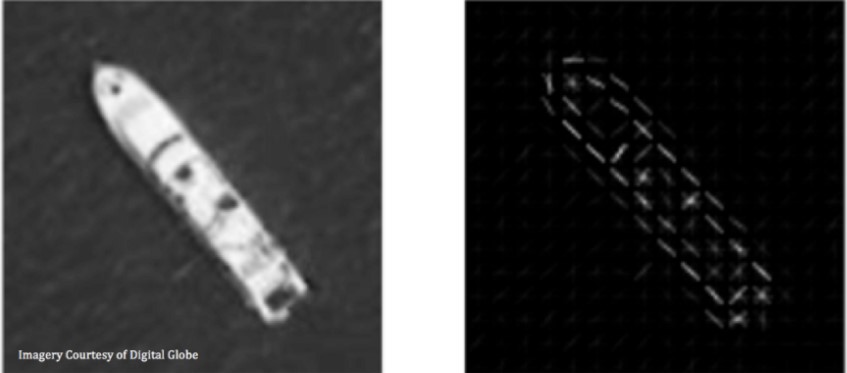
minNeighbors: Her aday dikdörtgenin kaç komşusunun onu tutması gerektiğini belirten parametre.

minSize: Tespit edilecek nesnenin olabileceği en düşük boyutu. Bu değerden küçük nesneler tespit edilmez

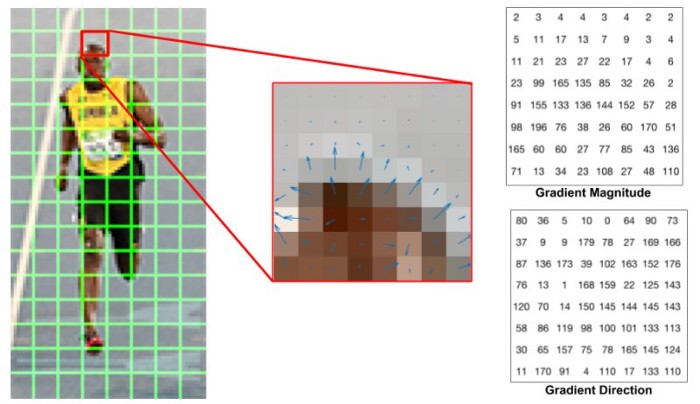
maxSize: : Tespit edilecek nesnenin olabileceği en büyük boyut. Genelde bu kullanılmaz

# HOG (Histogram of Oriented Gradients) - Yönlü Gradyanlar Histogramı

Nesne tanımada sıklıkla kullanılan ve bir nesneyi içerdiği açılar ile betimleyen bir tanımlayıcı olarak ifade edebiliriz

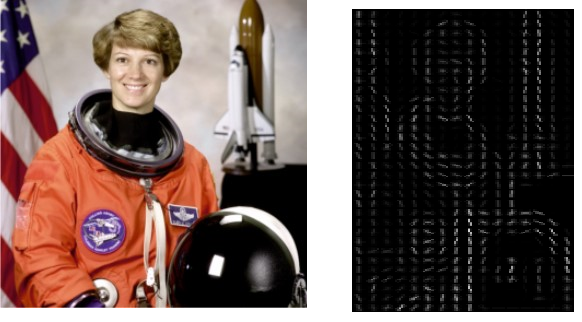


Görüntü hücrelere bölünür ve her hücre için gradyan histogramı hesaplanır. 8x8 hücreler, 8x8x3 = 192 piksel değerleri içerir. Bu yamanın gradyanı, piksel başına 8x8x2 = 128 sayıya kadar ekleyen 2 değer (büyüklük ve yön) içerir. 8x8 hücre kullanılmasının nedeni özellikleri yakalamak için yeterince büyüktür.



Normalizasyon tercihe bağlı olarak yapılabilir. Parlaklık, kontrast ve diğer aydınlatma etkilerinden kaçınmak için kullanılır.

Son olarak bu işlemle bir görüntü elde edilir ve algoritma burada son bulur.



HOG ile bulduğumuz şekilleri SVM ile eğiterek Nesne tespitinde Kullanılabilinir.

SVM ise kısaca, Destek Vektör Makineleri (Support Vector Machine) genellikle sınıflandırma problemlerinde kullanılan gözetimli öğrenme yöntemlerinden biridir. Bir düzlem üzerine yerleştirilmiş noktaları ayırmak için bir doğru çizer. Bu doğrunun, iki sınıfının noktaları için de maksimum uzaklıkta olmasını amaçlar. Karmaşık ama küçük ve orta ölçekteki veri setleri için uygundur.