



RAPOR BAŞLIŞI

HAZIRLAYAN

ADI SOYADI : RABA ŞENLİK

ÖĞRENCİ NUMARASI : 190303027

TESLİM TARİHİ : 20/11/2022

DERS ADI : MAKİNE ÖĞRENMESİ

DERS YÜRÜTÜCÜSÜ : SAEİD AGAHIAN

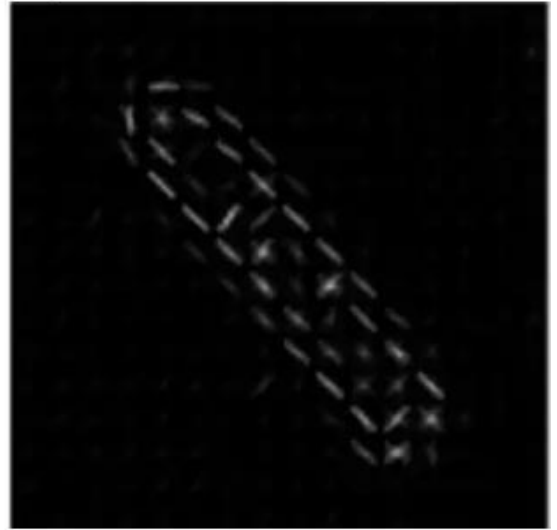
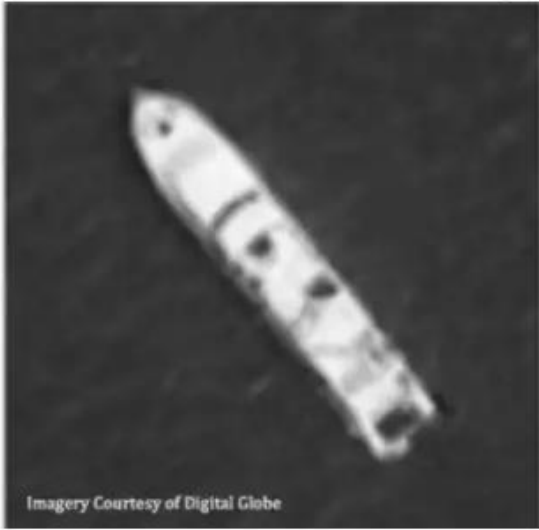
RESİM VERİ SETİ SINIFLANDIRMA

Makine öğrenmesi dersi projesi kapsamında 1000 k     ve iki sınıftan olu  an resim veri setini sınıflandırmaya   alı  tım. Sınıflandırma i  lemini yaparken HOG ile   zniteliklerini   ıkardım ve sınıflandırma algoritması olarak da KNN sınıflandırma algoritmasını kullandım. Projemi anlatmaya ge  meden   nce HOG ve KNN den bahsetmek istiyorum.

HOG (Histogram of Oriented Gradients):

HOG, g  r  nt  lerdeki b  lgelerin hareketlerini tespit etmek i  in kullanılan bir algoritmadır. Bu algoritma, bir g  r  nt  n  n   zelliklerini belirlemek i  in g  r  nt  n  n b  l  nmesine dayanır ve her b  lge i  in bir histogram   retir. Bu histogramlar, b  lgelerdeki hareketleri ve y  nlerini belirtir ve g  r  nt  n  n neredeyse herhangi bir y  nde kaydırılması durumunda bile benzer   zellikleri belirtir.

Kısacası, nesne tanımada kullanılan ve bir nesneyi i  erdi  i a  ılar ile nitelendiren bir tanımlayıcı olarak ifade edebiliriz.



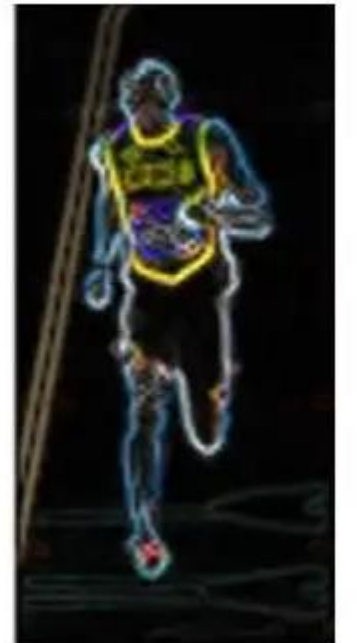
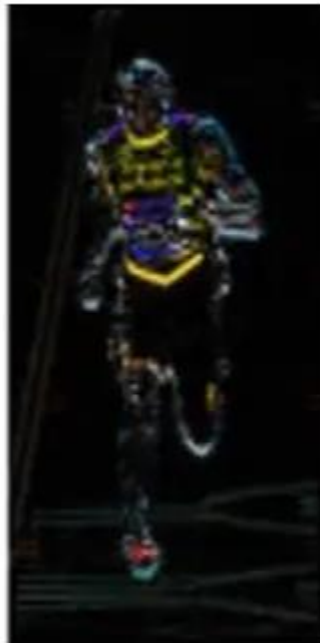
Eski bir algoritma olmasına ra  men hala pop  lerli  ini korumaktadır. Bu algoritma; y  z ve g  r  nt   algılamada, yaya algılamada, otonom ara  larda g  zetim tekniklerinde ve akıllı reklamcılıkta sıklıkla kullanılabilir.  tedir.

Şimdi algoritmamızın nasıl çalıştığını ve bu çalışma adımlarını inceleyelim:

1. Özellik Tanımlama
2. Ön İşleme
3. Gradyan Hesaplama
4. Gradyanların Histogramını Çıkarma
5. Normalizasyon
6. Görüntüyü Görselleştirme

Şimdi tek tek bu adımları inceleyelim.

1. **Özellik Tanımlama:** Görüntüdeki gereksiz bilgileri atarak görüntüyü basitleştirir ve yararlı bilgileri ön plana çıkarır.
2. **Ön işleme:** Görüntünün ileride oluşabilecek sorunlarını önlemek için yeniden boyutlandırmaya ihtiyacı var. Görüntü herhangi bir boyutta olabilir. Onun için biz bu adımda tüm görüntülerin boyutunu sabit bir en boy oranına getirmemiz gerekiyor. Yaygın olan en boy oranı 1:2'dir, yani görüntüler 100:200, 500:1000... olabilir.
3. **Gradyan Hesaplama:** HOG yapmadan önce görüntünün yatay ve dikey gradyanlarının hesaplanması gerekir. Bu durum da görüntünün çekirdek yardımıyla filtrelenmesi ile sağlanır. Kenarlar ve önemli noktalar bulunur. Gauss filtreleme, pencerenin boyutuna bağlı olarak merkezi pikselden uzaklığına göre merkezi piksele en çok ve komşu piksele azalan sırada ağırlık verir.



Gradyan hesabı şu şekilde yapılır:

$h = [-1, 0, 1]$ olarak verilen bir I imgesinin x doğrultusundaki gradyanı:

$$G_x = I * h, y$$

x doğrultusundaki gradyan

y doğrultusundaki gradyanı ise:

$$G_y = I * (-h)^T$$

y doğrultusundaki gradyan

Bu denklemler sonucu aşağıdaki eşitlik ortaya çıkar. Bir pikselin gradyan şiddeti aşağıda verilmiştir.

$$\rho(x, y) = \sqrt{G_x(x, y)^2 + G_y(x, y)^2}$$

Bir Pikselin Gradyan Şiddeti

Bulunan gradyanlar ile x, y noktasındaki piksele ait açı değeri de aşağıdaki denklem ile elde edilir.

$$\theta(x, y) = \tan^{-1} \left(\frac{G_y(x, y)}{G_x(x, y)} \right)$$

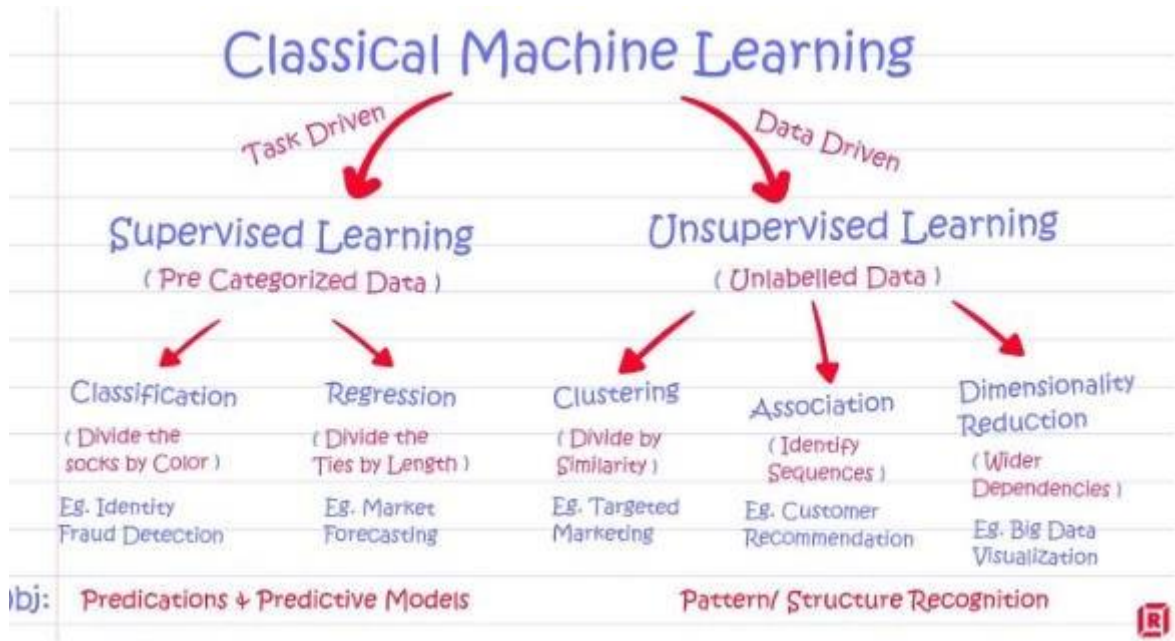
Bir Piksele Ait Açı Değeri

4. **Gradyanların Histogramını Çıkarma:** Görüntü hücrelere bölünür ve her hücre için gradyan histogramı hesaplanır.
5. **Normalizasyon:** Tercihe bağlı olarak yapılabilir. Parlaklık, kontrast ve diğer etkilerinden kaçınmak için kullanılır.
6. **Görüntüyü Görselleştirme:** Son olarak bu işlemle bir görüntü elde edilir ve algoritma burada son bulur.

HOG 'dan kısa ve net bir şekilde bahsettikten sonra şimdi de KNN sınıflandırıcısından bahsetmek istiyorum. Ama KNN 'e değinmeden önce sınıflandırma nedir bunu biraz anlatalım:

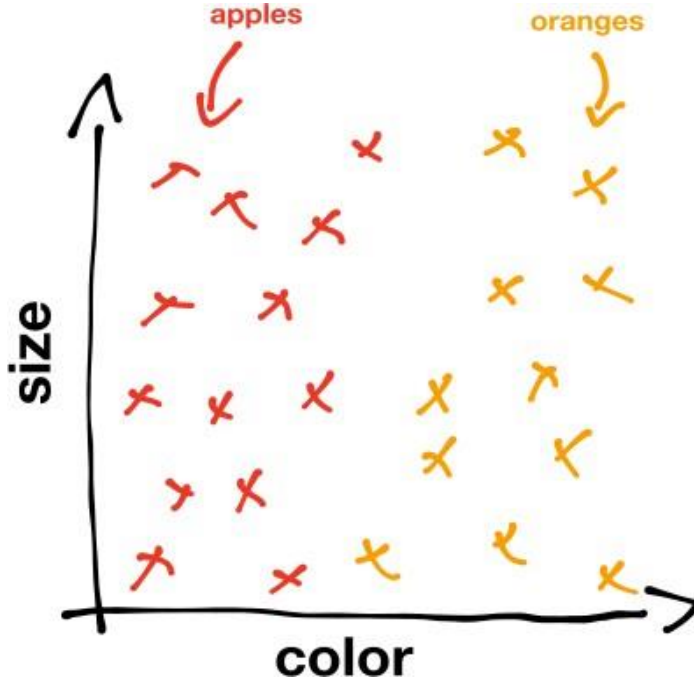
KNN (K-Nearest Neighbors) SINIFLANDIRMA:

Denetimli makine öğrenmesi algoritmalarından olan sınıflandırma; Bir eğitim verisi üzerinden bir model elde edip ve bu modeli tahminde kullanmasıdır. Bir nevi, elindeki veriden elde ettiği tecrübe ile yeni gelen verileri tahmin etmesidir. Eğitim verisinde özellikler ve bu özelliklere ait sonuç bilgisi tutulur. Sonuç bilgisi ve yapılan tahmin sonucu ile kategorik bir bilgi elde edilir.

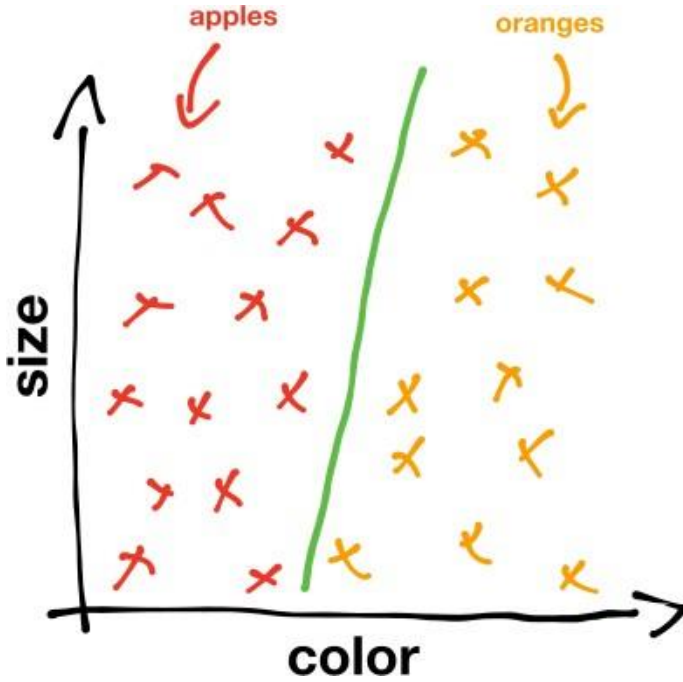


Örneğin;

Elimizde birçok elma ve portakal resmi bulunsun. Her resimden meyvelerin rengini ve boyutunu belirleyebiliriz. Gözetimli öğrenme için ilk adım etiketli eğitim verilerini edinmek gerekir. Bizim örneğimizde, her biri elma veya portakal olarak etiketlenen çok sayıda meyve resmi elde etmemiz anlamına geliyor. Bu resimlerden, renk ve boy bilgilerini çıkarabilir ve bunları elma ya da portakal olmakla ilişkilendirebiliriz. Örneğin, etiketli eğitim verimizi aşağıdaki gibi grafiğe dökebiliriz:



Kırmızı x'ler elma ve turuncu x'ler portakal olarak etiketlenmiş. Görebileceğiniz gibi veri de bir “pattern” var gibi gözüküyor. Kırmızı x'ler solda ve turuncu x'ler sağda kümelenmiş. Algoritmamızında bu tür “pattern” leri öğrenmesini istiyoruz. Bu problem için, amacımız, iki etiketli grubu bir karar sınırı (decision boundary) ile ayıracak algoritmayı yazmak. Basit bir karar sınırı aşağıdaki gibi olabilir:



Şimdi sınıflandırmayı anladığımıza göre KNN den bahsedelim:

KNN genellikle sınıflandırma problemleri için kullanılan bir denetimli öğrenme algoritmasıdır. Bu algoritmada benzerlik fonksiyonlarını kullanır.

Her ne kadar KNN algoritması k-means algoritmasına benzese de büyük farklılıklar da içermektedir. KNN algoritması bir eğitim verisi içerirken k-means algoritması bir eğitim verisi içermez. Yeni bir değer geldiğinde K değerine mesafeler hesaplanır ve yeni değer bir kümeye ilave edilir. Mesafe hesaplama işleminde ise k-means ve hiyerarşik kümeleme de kullanılan öklid uzaklığı, manhattan uzaklığı gibi uzaklık hesaplama yöntemleri kullanılabilir. Bu örneğimizde yine öklid uzaklığı kullanılacaktır.

Bu algoritma beş adımdan oluşur.

1. Öncelikle K değeri belirlenir.
2. Diğer nesnelerden hedef nesneye olan öklit uzaklıkları hesaplanır.
3. Uzaklıklar sıralanır ve en küçük uzaklığa bağlı olarak en yakın komşular bulunur.
4. En yakın komşu kategorileri toplanır.
5. En uygun komşu kategorisi seçilir.

Adımları örnek bir veri üzerinde inceleyelim. 4 adet verimiz ve verilerimizin sınıfları olsun. Hangi sınıfta olduğunu bilmediğimiz 5. verimiz gelsin. Bu veri hangi sınıftan olabilir?



Adım 1:

K değerini 3 olsun.

Adım 2: Yeni eklenen nesne ile diğer nesneler arasındaki uzaklıkları bul.

$$Fark_e = \sqrt{(X_i - X_{yeni})^2 + (Y_i - Y_{yeni})^2}$$

	Yeni Nesne ile Uzaklık
Nesne 1	4
Nesne 2	5
Nesne 3	3
Nesne 4	3.606

Adım 3: Uzaklıkları Sırala, k adet nesneyi seç.

Sıralama	Sınıfları
Nesne 4	Sınıf_2
Nesne 3	Sınıf_2
Nesne 1	Sınıf_1
Nesne 2	Sınıf_1

Nesne 4, Nesne 3 ve Nesne 1 en yakın 3 uzaklıktır.

Adım 4: Sınıf kategori sayılarını belirleme: 2 adet Sınıf_2 ve 1 adet Sınıf_1 sınıfından vardır.

Adım 5: Seçim: Bu durumda kazanan Sınıf_2 kazanandır.



Eğitim kümesinin büyük olması ve k değerini uygun seçilmesi KNN açısından çok önemlidir. KNN algoritmasına karar verdiğinizde eğitim verinizi ve k değerini artırarak başarıya bakın. Başarı sabitlenmeye başladığında iyi bir KNN tahmin sistemi yapmışsınızdır. Bu algoritmanın kötü tarafı her ekleme işleminde uzaklıkların hesaplanmasıdır.

KAYNAKÇA

1. <https://www.kaggle.com/code/hasansezertaan/machine-learning-dersleri-s-n-flan-d-rma>
2. https://erdincuzun.com/makine_ogrenmesi/k-nn-algoritmasi/
3. <https://makineogrenimi.wordpress.com/2017/05/26/siniflandirma-classification/>
4. <https://bilgisayarkavramlari.com/2013/03/31/siniflandirma-classification/>
5. https://erdincuzun.com/makine_ogrenmesi/04-makine-ogrenmesi-siniflandirma/
6. <https://www.kaggle.com/code/hasansezertaan/machine-learning-dersleri-s-n-flan-d-rma>
7. <https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f>
8. https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients
9. <https://debuggercafe.com/image-recognition-using-histogram-of-oriented-gradients-hog-descriptor/>
10. <https://www.kaggle.com/code/manikg/training-svm-classifier-with-hog-features>