

Katlı AdaBoost Karar Ağaçlarıyla Yüz Sezimi

Face Detection Using Boosted Tree Classifier Stages

Cem DEMİRKIR

Elektronik Mühendisliği Bölümü
Hava Harp Okulu, Yeşilyurt/İSTANBUL
cemd@boun.edu.tr

Bülent SANKUR

Elektronik Mühendisliği Bölümü
Boğaziçi Üniversitesi, Bebek/İSTANBUL
sankur@boun.edu.tr

Özetçe

Bu çalışmada, her biri bir takım basit öznitelikleri kullanan yüz sınıflandırıcıları arasındaki ilişkilerden oluşan üstün nitelikli ve hızlı bir sınıflandırıcı ile yüz sezimi anlatılmaktadır. Her bir sınıflandırıcı katı basit öznitelikleri kullanan yüz sınıflandırıcı ağaçlarından oluşmaktadır. Her bir ağaç kök düğümü karşılık gelen basit sınıflandırıcı, hatasının belli bir eşik değerini aşması halinde ikiye bölmekte ve oluşan ağaç eğitim kümesi üzerindeki sınıflandırma hatası eşik değerinin altına çekilené kadar bölünmektedir. Bu yolla katlar çoğaldıkça daha zorlu örneklerde eğitilecek olan basit sınıflandırıcıların gerektirdiğinde kuvvetlendirilerek sınıflandırıcı katlarının eğitim hatalarının belli bir performans değerine daha hızlı olarak yakınsaması sağlanmaktadır.

Abstract

In this work a classifier structure, which is based on using simple and weak classifiers based on Haar like features which are easily computed by integral image representation values, [1], and putting these weak classifiers into the nodes of a binary tree structure by performing AdaBoost algorithm, is described to construct boosted tree classifier stages detecting dynamic objects such as human faces. Each classifier stage consists of decision trees containing simple classifiers at its nodes. Each node classifier is constructed using only one simple feature and is divided into sub nodes if the decision tree error on the training set is greater than a predetermined threshold. This division process continues till the error is lower than the threshold and a maximum tree depth is not exceeded. By this approach a faster convergence to a certain classifier performance value is tried to obtain.

1. Giriş

Bir imgede insan yüz ya da yüzlerinin varlığının saptanması ve yerlerinin bulunması, kısaca yüz sezimi gerek güvenlik, gerekse insan-bilgisayar iletişimini uygulamalarında kullanılmaktadır. İnsan yüzünün dinamik bir yapıya sahip olması, aydınlatma, poz, ifade, aksesuarlar ve yaş etmenleriyle, tek bir birey için bile çok fazla değişkenlik göstermesi, yüz sezimini zor bir problem haline getirmektedir. Literatürde yüz aynılarını dayanın yöntemlerden, göz, cene, ağız gibi özelliklerin uzamsal kompozisyonunu kullanınlara, bütünsel görünüşü dayalı istatistiksel yöntemlere kadar pek çok farklı teknik sunulmuştur. Temel olarak bu tür algoritmalar öznitelik-tabanlı veya imgé-tabanlı (görünüşe dayalı) olmak üzere iki ana grupta toplanabilir. Imgé-tabanlı yöntemlerin yüz imgelerindeki yüksek seviyedeki değişkenliklerle başa çıkmada daha etkin olduğu gözlemlenmektedir [1], [4].

Sung [5] yüz ve yüz olmayan eğitim kümesini modifiye edilmiş K-ortalamaları yöntemiyle 12 toplağı ayırmıştır. Her bir toplağı Temel Bileşen Analizi (TBA) uygulayarak herhangi bir imgé vektöründen TBA uzayında yüz ve yüz olmayan uzaylara uzaklıklar içeren 24 boyutlu öznitelik vektörlerini oluşturmuş ve sınıflandırmayı da bir yapay sinir ağıyla gerçekleştirmiştir.

Rowley [6], her biri yüz imgesinin belli bir kısmına bakan saklı katmanında 26 birimli bir yapay sinir ağını eğitmiştir. Bu şekilde üç farklı yapay sinir ağıının farklı ölçeklerdeki ve pozisyonlarındaki sonuçlarını VE/VEYA kuralları zincirleriyle birleştiren bir karar ağı kullanmıştır.

Yang [7] EM algoritmasını kullanarak eğitimi kümesini hem topaklayan, hem de yerel boyut indirmesini sağlayan Faktör Analizini kullanmıştır. Böylece elde ettiği karışım modeli ile yüzsellik haritasını elde etmiştir. Ayrıca Yang [7] SOM algoritması ile 25 yüz ve 25 yüz olmayan toplağı, sınıflar arası saatlımı maksimum kılan, sınıf içi saçılımı ise en küçükten dönüsüm matrisini Fisher Doğrusal Ayırıcı Analizi (LDA) elde etmiştir. Böylece boyut indirgenmiş vektörlere ait bir toplağı bir Gauss ile modelleyerek, bir test imgesinin yüzsellüğine maksimum olabilirlik değerine bakarak karar vermiştir.

Osuna [8] Support Vector Machine (SVM) [9] yöntemi yüz sezimine uygulamıştır. İkinci dereceden polinom çekirdeklü fonksiyon kullanarak yüz ve yüz olmayan örnekleri birbirinden en büyük uzaklıkla ayırmamasını sağlayan destek vektörleri, yani eğitim imgelerini, bulmuştur.

Schneiderman [10] dalgacık katsayılarını kullanarak oluşturduğu öznitelik vektörlerini kullanarak eğitim kümesinden olabilirlik değerleri $P(\text{imgé} | \text{yüz})$ ve $P(\text{imgé} | \text{yüz olmayan})$ fonksiyonlarını çıkarmıştır. Yeni bir imgenin yüzsellik kararını olabilirlik oran testi uygulayarak karar vermiştir.

Bahsedilen yaklaşımlar, yüzsellik kararını verecek imgé parçacıklarını ayırdıkları ve farklı kameraların karakteristiklerinden kaynaklanan etkileri hafifletmek için belli normalizasyon işlemlerinden geçirmektedir. Genelde bunlar histogram denkleştirme, linear anlaşıma en iyi düzlemlerle çıkarma, kenar bölgeleri maskelerini.

Tüm bu yöntemlerde karar verebilmek için karmaşık ve işlem yükü yüksek sınıflandırıcılar imgenin her bir öleşinin her bir noktasında dolaştırmak gerekmektedir. Dolayısıyla gözlemeşinin yüz biner mertebede noktada konuşlampa kararların verilmesi normal bir kişisel bilgisayarda dakikalar almaktadır. Viola [11] Haar tabanlı fonksiyonları benzeri öznitelikleri her ölçük ve konumda çok hızlı hesaplayan "toplamlı imgé gösterimi" adlı bir yöntem geliştirmiştir. Ayrıca, AdaBoost [11] teknigi kullanılarak az sayıda Haar türü özniteliklerle yüz sınıflandırıcılarından güçlü birer sınıflandırıcı meydana getirilmektedir. Yöntemin üçüncü bir özelliği ise, her birinden yüz olmayan örneklerin belli bir kısmını reddeden, yüz olanları ise büyük bir oranda kabul eden ardışık sınıflandırıcılar şeklinde örgütlenmesidir. Böylece basit sınıflandırıcılar kararlarını coğunu vermekle, karmaşık sınıflandırıcılar ise çok daha az sayıda aday bölge ile çalışmaktadır. Nihayet yüz üzerinde aydınlatmadan kaynaklanan farklılıklarını gidermek amacıyla uygulanan normalizasyon işlemi, gene bir toplam imgé kavramıyla hızlı bir biçimde yapılmaktadır.

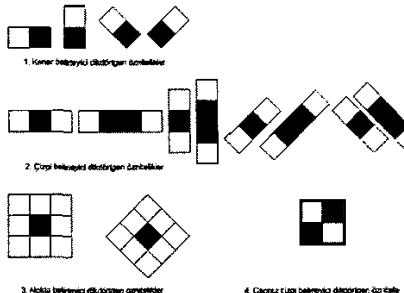
Biz bu çalışmada, daha karmaşık sınıflandırıcılar Haar benzeri öznitelikleri kullanan ağaç mimarisinde yer alan sınıflandırıcılar ile

olusturdu. Ağac üzerindeki zayıf bir sınıflandırıcının eğitim kümesi üzerindeki hatası belli bir eşik değerini aşındırı, bölünerek düğüm sayısını coğaltılmaktır, aksi takdirde bölmeye gerek kalmadan sadece kök düğümden oluşan basit bir sınıflandırıcı olarak kalmaktadır. Böylece her bir sınıflandırıcı katında hesaplanacak basit öznitelik sayısı azaltılıp eğitim sürecinde istenilen performans değerine daha kolay yaklaşmak amaçlanmaktadır. Kullanılan eğitim yordamı ile ardışık olarak birbirinden daha fazla sayıda sınıflandırıcı içeren, kuvvetli sınıflandırıcı katları ardarda çarpmaktadır.

2. Öznitelikler

2.1. Kullanılan Öznitelikler

Bu çalışmada basit sınıflandırıcıları oluşturmak için kullanılan öznitelikler Papagergiou [12] tarafından sunulan ve [2]'de tanımlanan genelleştirilmiş Haar öznitelikleridir. Şekil 1'de gösterilen farklı tipteki özniteliklerin (imge özniteliklerini elde etmek için kullanılan pencereler) her bir mümkün olan her konum ve ölçek için belli bir alan üzerinde olacak şekilde üretilmektedir. Buna göre kullanılmış olduğumuz 20x20'lük tarama penceresi içerisindeki her yolla oluşturulmuş Şekil 1'de gösterilen tiplere ait yaklaşık 60000 adet öznitelik oluşturulmaktadır. Bir öznitelik iki veya üç dikdörtgen ile tanımlanmakta ve değeri ise bu dikdörtgenler içerisindeki piksel değerlerinin ağırlıklı toplamı olmaktadır. Şekil 2'de bir test penceresi içerisinde yerleştirilmiş farklı tipteki basit Haar öznitelikleri gözükmemektedir.

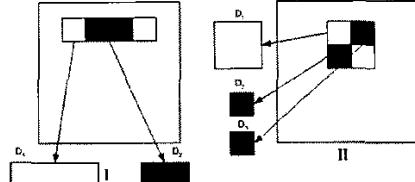


Şekil 1 Genelleştirilmiş Haar öznitelik tipleri

Örneğin Şekil 2'deki I ve II numaralı öznitelik değerleri f_i ve f_n , tanımlayıcı $D_i, i = 1, 2, 3$ dikdörtgenlerine ait bölgelerdeki piksel değerleri toplamı ve ağırlıkları cinsinden eşitlik (1)'de tanımlanmaktadır. Buna göre Şekil 2'deki I numaralı öznitelikte ait beş dikdörtgenin ağırlığı -1, siyah olanın ise 2 olmaktadır.

2.2. Toplam İmge

Bir imgeye ait genelleştirilmiş Haar özniteliklerinin hızlı bir biçimde hesaplanması için iki tür toplam imge tanımlanmaktadır; toplam imge, imge üzerinden piksellerin kısmı toplamlarına karşılık gelmektedir. Birincisi dik görünümüne sahip öznitelikleri hesaplamak için kullanılan Toplam İmge (T_I), değeri ise 45 derece eğimli öznitelikleri hesaplamak için kullanılan Eğimli Toplam İmge (ET_I)dir. Şekil 3 (a) ve (b)'de sırasıyla dik görünümüne sahip öznitelik hesabı için piksel değerleri toplanması gereken imge alanı ile eğimli görünümüne sahip öznitelik hesabı için piksel değerleri toplanması gereken imge alanı koyu olarak gösterilmektedir. Buna göre imge üzerindeki herhangi bir (x,y) noktasına karşılık gelen T_I(x,y) ve ET_I(x,y) değerleri, W genişliğinde, H yüksekliğindeki bir I imgesi için



Şekil 2 Farklı tipteki özniteliklerin bir test imgesi içinde gösterimi ve tanımlayıcı dikdörtgenlerin ayrıntıları. Şekilde f_i ve f_n öznitelik değerlerini, $D_i, w_i, i = 1, 2, 3$ sırasıyla tanımlayıcı dikdörtgenler ve ağırlıklarını göstermektedir. PikselToplam ilgili dikdörtgenin kapsadığı bölgedeki imge piksellerinin toplamını, Alan ise ilgili dikdörtgenin alanını göstermektedir.

$$f_i = \sum_{i=1}^3 w_i \text{PikselToplam}(D_i), \sum_{i=1}^3 w_i \text{Alan}(D_i) = 0 \quad (1)$$

$$f_n = \sum_{i=1}^3 w_i \text{PikselToplam}(D_i), \sum_{i=1}^3 w_i \text{Alan}(D_i) = 0 \quad (2)$$

$$T_i(x, y) = \sum_{i \leq y, j \leq x} I(i, j),$$

$$ET_i(x, y) = \sum_{i \leq y, i \leq y - |x - j|} I(i, j) \quad (2)$$

$$0 \leq i \leq H - 1, \quad 0 \leq j \leq W - 1$$

T_i(x,y) ve ET_i(x,y) değerleri, bir önceki noktada hesaplanan değerlerinden yararlanarak hızlı bir biçimde bulunabilir:

$$T_i(x, y) = T_i(x, y - 1) + T_i(x - 1, y) + I(x, y) - T_i(x - 1, y - 1) \quad (3)$$

$$T_i(-1, y) = T_i(x, -1) = T_i(-1, -1) = 0$$

$$ET_i(x, y) = ET_i(x - 1, y - 1) + ET_i(x + 1, y - 1) \quad (4)$$

$$-ET_i(x, y - 2) + I(x, y) + I(x, y - 1) \quad (4)$$

$$ET_i(-1, y) = ET_i(x, -1) = ET_i(x, -2) = 0$$

$$ET_i(-1, -1) = ET_i(-1, -2) = 0$$

formüllerile özyineli olarak hesaplanabilir.

2.3. Özniteliklerin hızlı olarak hesaplanması

Toplam imge değerleri elimizde bulunduğuunda, herhangi bir nokta için öznitelik hesaplaması pencere içindeki piksel toplamını yapacağımıza toplam imgeden birkaç değer okuyarak kolayca hesaplanabilir. Şekil 3(c)'deki dikey görünümü sol üst köşesi (x,y) koordinatlarındaki w genişliğinde ve h yüksekliğindeki D₁ dikdörtgenine ait piksel değerlerinin toplamı, imgeye T_I'ye ait dört değerle söyle hesaplanabilir:

$$\text{PikselToplam}(D_1) = T_i(x - 1, y - 1) + T_i(x + w - 1, y + h - 1) \quad (5)$$

$$-T_i(x - 1, y + h - 1) - T_i(x + w - 1, y - 1)$$

Benzer durum Şekil 3 (c)'deki eğimli D₂ dikdörtgeni için de geçerlidir ve dikdörtgendeki piksel toplamı, imgenin ET_I değerleri cinsinden kolayca hesaplanabilir:

$$\text{PikselToplam}(D_2) = ET_i(x - h + w, y + w + h - 1) + ET_i(x, y - 1) \quad (6)$$

$$-ET_i(x - h, y + h - 1) - ET_i(x + w, y + w - 1)$$

2.4. Öznitelik değerlerinin normalizasyonu

İmge piksel değerlerindeki aydınlatmadan kaynaklanan etkiyi azaltmak amacıyla (x,y) noktasındaki $I(x,y)$ piksel değeri

$$I_{\text{norm}}(x,y) = \frac{I(x,y) - \mu}{\sigma} \quad (7)$$

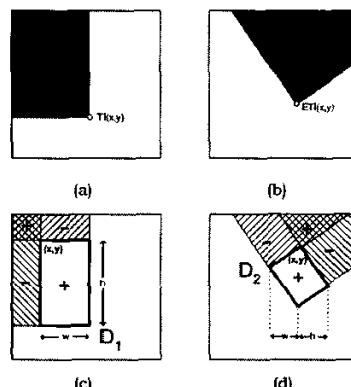
işlemi uygulanmaktadır. Burada μ tarama penceresi içerisindeki piksellerin toplamı, σ ise bu piksellerin standart sapmasıdır. Tarama penceresine düşen piksel değerlerinin standart sapması T_1 ve Karesel T_1 (KTI) değerleri kullanılarak kolayca hesaplanabilir. Bu nedenle düzgeleme katsayısının hızlı bir biçimde hesaplanabilmesi için üçüncü bir toplam imge türü olan

$$KTI(x,y) = \sum_{(i,j) \in S_1(x,y)} I^2(i,j) \quad (8)$$

KTI(x,y) değerlerinin hesaplanması gerekmektedir. Düzgeleme katsayısı için gerekli olan standart sapma değerinin (x,y) konumundaki değeri, alanı A olan bir tarama penceresi için

$$\sigma(x,y) = \sqrt{A \sum_{(i,j) \in S_1(x,y)} I^2(i,j) - \left(\sum_{(i,j) \in S_1(x,y)} I(i,j) \right)^2} / A \quad (9)$$

$S(x,y)$ tarama penceresinin (x,y) konumdayken kapsadığı piksel kümeleridir.



Şekil 3 (a) T_1 'nin (x,y) noktası için değerleri toplanan imge bölgesi, (b) ET_1 'nin (x,y) noktası için değerleri toplanan imge bölgesi, (c) Dikey görünümü öznitelik hesabi için gerekli bir dikdörtgenin T_1 değerleri cinsinden ifadesi (d) Eğimli görünümü öznitelik hesabi için gerekli bir dikdörtgenin ET_1 değerleri cinsinden ifadesi.

3. Sınıflandırıcılar

Sınıflandırıcı olarak Haar özniteliklerini kullanan basit ve zayıf sınıflandırıcılar birleştirerek kuvvetli bir sınıflandırıcı oluşturan AdaBoost teknigi kullanılmaktadır. Basit sınıflandırıcı olarak düğümlerde basit Haar öznitelik değerlerini eşik değerleri ile kıyaslayarak pozitif sınıf (yüz) ve negatif sınıf (yüz olmama) karar veren zayıf sınıflandırıcılar içeren sınıflandırıcı kullanılmaktadır.

3.1. Basit Sınıflandırıcı Ağacı

Bir giriş imgesi x değeri ağaç düğümlerinde bulunan

$$h_j \cdot x = \begin{cases} 1 & \text{if } f_j \cdot x > s_j T_j \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

ve j numaralı öznitelik değerini kullanan basit sınıflandırıcılar ile test edilmektedir. Eşitlik (10)'da $f_j(x)$, x imgesi için j numaralı öznitelik tarafından üretilen değeri, $s_j = (+1, -1)$ değeri ise eşitsizliğin

yönüne göstermektedir. Düğümlerdeki öznitelik kıyaslayıcı basit sınıflandırıcı $h_j \cdot x$ 'ler tarafından verilen 1 kararları ağacın sol düğümüne, 0 kararları ise ağacın sağ düğümüne atlamasını sağlayarak, yaprak düğümleri deigin devam eder. Soldaki yaprak düzgume düşen x değerleri için pozitif sınıf karar, sağdaki yaprak düzgume düşenler için ise negatif sınıf karar verilmektedir.

Tablo 1 AdaBoost teknigi kullanarak basit sınıflandırıcı ağaçlarından oluşan pekiştirilmiş sınıflandırıcının eğitimi yordamı.

1)	Pozitif ve negatif sınıf ait eğitim kümesi örnekleri $(x_1, y_1), \dots, (x_n, y_n)$ topla. $y_i = \{0, 1\}$ pozitif ve negatif örneklerde sınıf etiketlerini göstermektedir.
2)	Eğitim kümesi örnekleri ait ağırlık katsayılarını $w_{i,1} = \begin{cases} \frac{1}{2} & \text{eğer } y_i = 1 \\ \frac{1}{2} & \text{eğer } y_i = 0 \end{cases}$ olacak şekilde ata.
3)	Tüm eğitim kümesini kullanarak tüm öznitelik kümesi için eşik değerlerini belirleyerek basit sınıflandırıcı kümelerini $\{h_j\}$ 'yi oluştur ve sakla.
4)	$k=1, \dots, N$ için <ol style="list-style-type: none"> Eğitim kümesi ağırlıklarını toplamla 1 olacak şekilde düzle. Bir ağaç sınıflandırıcı $t_k(x)$ 'e ait kök düğümü oluştur ve ağaç seviyesi göstergesini sıfır. i) Anki ağaç seviyesi için düğüm listesi çıkar. Listedeki tüm düğümler için: <ol style="list-style-type: none"> Düğüme ulaşan eğitim kümesi elemanları ile basit sınıflandırıcılar için $\epsilon_j = \sum_i w_i h_j(x_i) - y_i$ hatalarını belirle. Minimum hata ϵ_{\min} veren öznitelik sınıflandırıcısını ilgili düzgume ata. Mevcut ağaçın eğitim kümesi üzerindeki hatası 0.4'den büyüğe o anki düzgüm listesinde bulunan düğümler için çocuk düğümleri olusur, ağaç seviyesi göstergesini bir artır, i) nci adıma geri dön, hata 0.4'den küçükse c) adımlına geç. Bulunan ağaç sınıflandırıcı $t_k(x)$ 'i kullanarak eğitim kümesi ağırlıklarını $w_{i,1,1} = \begin{cases} \frac{w_{i,1}}{\beta} & \text{eğer } h_j(x_i) > s_j T_j \\ \frac{w_{i,1}}{\beta} & \text{eğer } h_j(x_i) \leq s_j T_j \end{cases} \quad \beta = \sum_{i=1}^n w_{i,1}$
5)	Oluşan kuvvetli sınıflandırıcı $h(x) = \begin{cases} 1 & \text{if } \sum_{k=1}^K \alpha_k t_k(x) \geq \frac{1}{2} \sum_{i=1}^n \alpha_i \\ 0 & \text{otherwise} \end{cases}$

3.2. Kuvvetli sınıflandırıcı katları

AdaBoost teknigi kullanılarak pekiştirilmiş sınıflandırıcı, ağaç sınıflandırıcılarının birer ağırlıklı toplamından oluşmaktadır. Kuvvetlendirilmiş sınıflandırıcının oluşturulmasında kullanılan eğitim yordamı Tablo 1'de açıklanmaktadır.

Tablo 2 Çok katlı sınıflandırıcı oluşturmak için kullanılan eğitim yordamı.

1. Her sınıflandırıcı katının gerçekleşmesi gereken
a) maksimum yanlış kabul oranı F
b) minimum sezik oranı d
parametrelerini belirle.
2. Tüm çok katlı sınıflandırıcının gerçekleşmesi gereken hedef yanlış kabul F_{hedef} parametresini belirle.

3. Oluşturulan tüm sınıflandırıcılar ait yanlış kabul F^* ve D değerlerinin 1 olarak başlat.
3. N_e adet negatif, P_e adet pozitif sınıfa ait eğitim kümesi örneği topla.
4. N_t adet negatif, P_t adet pozitif sınıfa ait test kümesi örneği topla.
5. Kat sayısı S 'yi sıfırda
6. Kuvvetli sınıflandırıcı katma ait başlangıç ölçüt deşifrenlerini oluştur.
 - a) Sınıflandırıcı katın eğitim kümesi üzerindeki yanlış kabul oranı f_e , sezim oranı d_e
 - b) Sınıflandırıcı katın test kümesi üzerindeki yanlış kabul oranı f_t , sezim oranı d_t
6. $F > F_{\text{bedef}}$ olduğu sürece
 - a) Kuvvetli sınıflandırıcı katı ve ilk yeni sınıflandırıcısını oluştur.
 - b) Eğitilecek yeni sınıflandırıcı olduğu sürece
 - i) Ağac sınıflandırıcı eğitimi kümesini kullanarak oluştur.
 - ii) Eğer ($d_e = 1$ ve $f_e < f_t$) ise
 - 1) $d_e \geq d$ olacak şekilde sınıflandırıcı katının çıkış değerini ayarla.
 - 2) Eğer $f_t > f$ ise kata yeni bir ağac sınıflandırıcı ekle.
 - iii) Eğer ($d_e = 1$ ve $f_e < f_t$) ise
 - 1) kata yeni bir ağac sınıflandırıcı ekle.
 - iv) b) admına geri dön.
 - c) F 'yi F^*f_e olacak şekilde güncelle.
 - d) Eğer $F > F_{\text{bedef}}$ ise oluşan çok katlı sınıflandırıcıyı kullanarak eğitim kümesi için ayrılan yüz olmayan resimlerden N_e adet yanlış alarm uren negatif örnek, test kümesi için ayrılan yüz resimlerden P_t adet sezimlenen pozitif örnek, test için ayrılan yüz olmayan resimlerden N_t adet yanlış alarm uren negatif örnek topla. Ve 6nci adıma dön.

3.3. Ardişik Çok Katlı Sınıflandırıcı

Pekştirilmiş sınıflandırıcı katları art arda ekleyerek küçük yanlış kabul olasılığı ve yüksek doğru karar oranları bir nesne sezici gerçekleştirmek mümkündür [1]. Katlardaki kuvvetli sınıflandırıcılar ile yüksek bir sezim oranı, örneğin % 99.5, ve bir nesne sezici için yüksek olabilecek yanlış kabul oranı, örneğin % 50 den daha gerçeklenebilir. Bu amaçla oluşturulacak her kat, kendisinden daha önceki katlardan oluşan tümleşik sınıflandırıcı tarafından verilen yanlış yüz kabullerini negatif eğitim kümesi kabul ederek eğitime devam etmektedir. Böylece oluşan her yeni kat, bir önceki katın üstesinden gelemediği zor örnekleri daha karmaşık ve çok sayıda basit sınıflandırıcı kullanarak, örneğin % 99.5 doğru sezim ve % 50'den az yanlış kabul olasılığı gibi başarım ölçütleri ile öğrenmektedir.

Ardışık olarak daha fazla sayıda pekştirilmiş sınıflandırıcı içeren katları otomatik olarak çıkarmak amacıyla Tablo 2'de verilen eğitim yordamı uygulanmıştır.

4. Deneyler ve Sonuçlar

Eğitim kümesini oluşturmam amacıyla FERET [13] ve ORL veritabanında bulunan ön görünümlü yüzler içeren resimlerden 20x20 boyutlarında hizalanmış ve ölçeklenmiş şekilde yüz resimleri çkartılmıştır. Elde edilen yüzlerden 5000 tanesi eğitim kümesi ve 1000 tanesi eğitim sırasında kullanılan test kümesi olarak ayrılmıştır. Eğitim kümesi ve eğitim sırasında kullanılan test kümesinin negatif örnekleri ise elimizde bulunan yaklaşık 400 adet 1536x1024 boyutlarında yüz içermeyen manzara resimlerinden çıkarılmıştır.

Eğitim içi sağlanmasa gerekten performans kriterleri $f=0.5$, $d=0.995$, $F_{\text{bedef}} = 10^4$ olarak alınmıştır.

Eğitim sonucunda ortaya çıkan katlı sınıflandırıcının ilk sekiz katına ilişkin sınıflandırıcı katlarındaki ağac sayısı ve içerdikleri öznitelik sayısı miktarlarına ilişkin bilgiler ve her katta test kümesi

üzerinde elde edilen yanlış kabul oranına ilişkin veriler Tablo 3'de sunulmuştur.

Tablo 3 Çok katlı sınıflandırıcıya ilişkin yapısal bilgiler

Kat numarası	Sınıflandırıcı sayısı	Ağac sayısı	Öznitelik sayısı	Yanlış kabul %
1	12	12	12	34
2	20	20	20	38
3	22	22	22	45
4	25	25	25	46
5	39	39	39	48
6	95	95	111	34
7	110	110	156	38
8	189	189	399	29

Sezim hızı 384x286 boyutlarındaki bir imgede Pentium 2 GHz makine kullanarak ortalama 0.4 saniye olmaktadır. Elde edilen sınıflandırıcı CMU test set' den çkarları önden görünümlü 230 yüz içeren 80 resim üzerinde denenmiştir. Elde edilen performans değerleri Tablo 4'de verilmektedir. Performans değerinin diğer yönlerine kıyasla aynı sezim oranı için daha fazla yanlış kabul sayısı ürettiği gözlenmiştir. Sezim oranının iyileştirilmesi ve yöntemin farklı yüz pozlarını sezerek şekilde kabiliyetinin artırılması için çalışılması düşünülmektedir.

Tablo 4 Sınıflandırıcının ve diğer [6], [1] başarım değerleri

Toplam Yüz Sayısı	Yanlış Kabul Sayısı	Sezim Sayısı
230	50	203 (% 88)
507	31	(%86)[6]
507	50	(%88)[1]

5. Kaynakça

- [1] P. Viola, M. J. Jones, *Rapid Object Detection Using A Boosted Cascade of Simple Features*, IEEE CVPR, 2001.
- [2] R. Lienhart, A. Kurian, V. Pisarevsky, *Empirical Analysis of Detection of Boosted Classifiers for Rapid Object Detection*, MRL Technical Report, December 2002.
- [3] E. Hjelmas, B. Kei Low, *Face Detection: A Survey*, Computer Vision and Image Understanding 83, 236-274 (2001).
- [4] M.-H. Yang, D. Kriegman, N. Ahuja, *Detecting Faces in Images : A Survey*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 1, pp. 34-58, 2002.
- [5] K. Sung, T. Poggio, *Example-Based Learning for View Based Face Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, 39-51, 1998.
- [6] H. A. Rowley, S. Baluja, T. Kanade, *Neural Network-Based Face Detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 20, No. 1, 23-38, January 1998.
- [7] M.-H. Yang, D. Kriegman, N. Ahuja, *Face Detection Using Multimodal Density Models*, Computer Vision and Image Understanding, 84, 264-284, 2001.
- [8] E. Osuna, R. Freund, F. Girosi, *Training Support Vector Machines: An Application to Face Detection*, ICVPR, San Juan, June 1997.
- [9] V. Vapnik, *Nature of Statistical Learning Theory*, Springer Verlag, 1995.
- [10] H. Schneiderman, T. Kanade, *A Statistical Method for 3D Object Detection Applied to Faces and Cars*, IEEE Conference on Computer Vision and Pattern Recognition, (2000).
- [11] Y. Freund, R. E. Schapire, *A Decision-theoretic Generalization of On-line learning and An Application to Boosting*, Computational Learning Theory, 23-37, 1995.
- [12] C. Papageorgiou, M. Oren, T. Poggio, *A General Framework for Object Detection*, ICCV, 1998.
- [13] P.J. Phillips, H. Moon, P. Rauss, and S. Rizvi, *The FERET Evaluation Methodology for Face-Recognition Algorithms*, IEEE Proc. Computer Vision and Pattern Recognition, pp. 137-143, June 1997.