

Kathı AdaBoost Karar Ağaçlarıyla Yüz Sezimi

Face Detection Using Boosted Tree Classifier Stages

Cem DEMİRKİR

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 öznetelikleri kullanan zayıf sınıflandırıcıların ardışık katlarından oluşan üstün nitelikli ve hızlı bir sınıflandırıcı ile yüz sezimi anlatılmaktadır. Her bir sınıflandırıcı katı basit öznetelikleri kullanan zayıf sınıflandırıcı ağaçlarından oluşmaktadır. Her bir ağaç kök düğüme karşılık gelen basit sınıflandırıcı, hatasının belli bir eşik değerini aşması halinde ikiye bölünmekte ve oluşan ağaç eğitim kümesi üzerindeki sınıflandırma hatası eşik değerin altına çekilene kadar bölünmektedir. Bu yolla katlar çoğaldıkça daha zorlu örneklerle eğitilecek olan basit sınıflandırıcıların gerektiğinde 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 iletişimi 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 ayırtılma dayanan yöntemlerden, göz, çene, ağz gibi öğelerin uzamsal kompozisyonunu kullananlara, bütünsel görünüşe dayalı istatistiksel yöntemlere kadar pek çok farklı teknik sunulmuştur. Temel olarak bu tür algoritmalar öznetelik-tabanlı veya imge-tabanlı (görünüşe dayalı) olmak üzere iki ana grupta toplanabilir. İmge-tabanlı yöntemlerin yüz imgelerindeki yüksek seviyedeki değişkenliklerle başa çıkmada daha etkin olduğu gözlenmektedir [1], [4].

Sung [5] yüz ve yüz olmayan eğitim kümesini modifiye edilmiş K-ortalama yöntemine 12 topağa ayırmıştır. Her bir topağa Temel Bileşen Analizi (TBA) uygulanarak herhangi bir imge vektöründen TBA uzayında yüz ve yüz olmayan uzaylara uzaklıkları içeren 24 boyutlu öznetelik 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 şekildeki üç farklı yapay sinir ağının farklı ölçeklerdeki ve pozisyonlardaki sonuçlarını VE/VEYA kuralları zincirleriyle birleştiren bir karar ağı kullanmıştır.

Yang [7] EM algoritmasını kullanarak eğitim kümesini hem toplayan, 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 topağı, sınıflar arası saçılımı maksimum kılan, sınıf içi saçılımı ise en küçültlen dönüşüm matrisini Fisher Doğrusal Ayrıştırma Analizi (LDA) elde etmiştir. Böylece boyut indirgenmiş vektörlere ait her bir topağı bir Gauss ile modelleyerek, bir test imgesinin yüzsellikine maksimum olasılık değeri bakarak karar vermiştir.

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

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

Bahsedilen yaklaşımlar, yüzsellik kararı verilecek imge parçacıklarını aydınlatma ve farklı kamera karakteristiklerinden kaynaklanan etkileri hafifletmek için belli normalizasyon işlemlerinden geçirmektedir. Genelde bunlar histogram denkleştirme, lineer anlamda en iyi düzlem çıkarma, kenar bölgeleri maskeleymedir.

Tüm bu yöntemlerde karar verebilmek için karmaşık ve işlem yükü yüksek sınıflandırıcıları imgenin her bir ölçeğinin her bir noktasında dolaştırmak gerekmektedir. Dolayısıyla gözlem penceresinin yüz binler mertebesinde noktada konuşlanıp kararların verilmesi normal bir kişisel bilgisayarda dakikalar almaktadır. Viola [1] Haar taban fonksiyonu benzeri öznetelikleri her ölçek ve konumda çok hızlı hesaplayan "toplam imge gösterimi" adlı bir yöntem geliştirmiştir. Ayrıca, AdaBoost [11] tekniği kullanılarak az sayıda Haar türü özneteliklerle zayıf sınıflandırıcılardan güçlü birer sınıflandırıcı meydana getirilmektedir. Yöntemin üçüncü bir özelliği ise, her birinde yüz olmayan örneklerin belli bir kısmını reddeden, yüz olanları ise büyük bir oranda kabul eden ardışık sınıflandırıcılar şeklinde örgütlenmesidir. Böylece basit sınıflandırıcılar kararların çoğunu vermekte, karmaşık sınıflandırıcılar ise çok daha az sayıda aday bölge ile çalışmaktadır. Nihayet yüz üzerinde aydınlatmadan kaynaklanan farklılıkları gidermek amacıyla uygulanan normalizasyon işlemi, gene bir toplam imge kavramıyla hızlı bir biçimde yapılabilir.

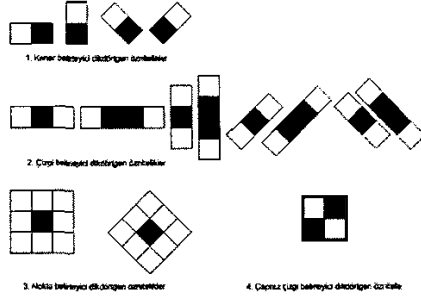
Biz bu çalışmada, daha karmaşık sınıflandırıcılar Haar benzeri öznetelikleri kullanan ağaç mimarisinde yer alan sınıflandırıcılar ile

oluşturduk. Ağaç üzerindeki zayıf bir sınıflandırıcının eğitim kümesi üzerindeki hatası belli bir eşik değerini aştığında, bölünerek düğüm sayısı çoğaltılmakta, aksi takdirde bölmeye gerek kalmadan sadece kök düğümünden oluşan basit bir sınıflandırıcı olarak kalmaktadır. Böylece her bir sınıflandırıcı katında hesaplanacak basit öznetelik sayısı azaltılıp eğitim sürecinde istenilen performans değerine daha kolay yakınsamak amaçlanmaktadır. Kullanılan eğitim yordamı ile ardışık olarak birbirinden daha fazla sayıda sınıflandırıcı içeren, kuvvetli sınıflandırıcı katları ardarda çıkarılmaktadır.

2. Öznetelikler

2.1. Kullanılan Öznetelikler

Bu çalışmada basit sınıflandırıcıları oluşturulmak için kullanılan öznetelikler Papageorgiou [12] tarafından sunulan ve [2]'de tanımlanan geliştirilmiş Haar öznetelikleridir. Şekil 1'de gösterilen farklı tipteki özneteliklerin (imge özneteliklerini elde etmek için kullanılan pencereler) her biri mümkün olan her konum ve ölçek için belli bir alanı üstünde olacak şekilde üretilmektedir. Buna göre kullanmış olduğumuz 20x20'lik tarama penceresinde bu yolla oluşturulmuş Şekil 1'de gösterilen tiplere ait yaklaşık 60000 adet öznetelik oluşturulmaktadır. Bir öznetelik 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 öznetelikleri görülmektedir.

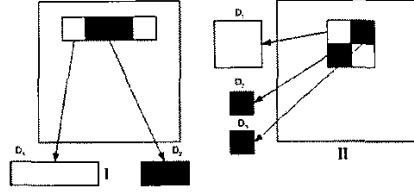


Şekil 1 Genelleştirilmiş Haar öznetelik tipleri

Örneğin Şekil 2'deki I ve II numaralı öznetelik değerleri f_i ve f_{II} , 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 özneteliğe ait beyaz dikdörtgenin ağırlığı -1, siyah olanın ise 2 olmaktadır.

2.2. Toplam İmge

Bir imgeye ait genelleştirilmiş Haar özneteliklerinin hızlı bir biçimde hesaplanması için iki tür toplam imge tanımlanmaktadır; toplam imge, imge üzerinden piksellerin kısmi toplamına karşılık gelmektedir. Birincisi dik görünüme sahip öznetelikleri hesaplamak için kullanılan Toplam İmge (TI), diğeri ise 45 derece eğimli öznetelikleri hesaplamak için kullanılan Eğimli Toplam İmge (ETI)'dir. Şekil 3 (a) ve (b)'de sırasıyla dik görünüme sahip öznetelik hesabı için piksel değerleri toplanması gereken imge alanı ile eğimli görünüme sahip öznetelik hesabı için piksel değerleri toplanması gereken imge alanı koyu olarak gösterilmektedir. Buna göre imge üzerindeki herhangi bir (x,y) noktasına karşılık gelen TI(x,y) ve ETI(x,y) değerleri, W genişliğinde, H yüksekliğindeki bir I imgesi için



Şekil 2 Farklı tipteki özneteliklerin bir test imgesi içinde gösterimi ve tanımlayıcı dikdörtgenlere ayrıştırılması. Şekilde f_i ve f_{II} öznetelik 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.

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

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

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

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

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

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

$$TI(x, y) = TI(x, y - 1) + TI(x - 1, y) + I(x, y) - TI(x - 1, y - 1)$$

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

$$ETI(x, y) = ETI(x - 1, y - 1) + ETI(x + 1, y - 1)$$

$$-ETI(x, y - 2) + I(x, y) + I(x, y - 1)$$

$$ETI(-1, y) = ETI(x, -1) = ETI(x, -2) = 0 \quad (4)$$

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

formülleriyle özyineli olarak hesaplanabilir.

2.3. Özneteliklerin hızlı olarak hesaplanması

Toplam imge değerleri elimizde bulunduğu anda, herhangi bir nokta için öznetelik hesaplaması pencere içindeki piksel toplamını yapacağımıza toplam imgeden birkaç değer okuyarak kolayca hesaplanabilir. Şekil 3(c)'deki dikey görünümlü sol üst köşesi (x,y) koordinatlarındaki w genişliğinde ve h yüksekliğindeki D_1 dikdörtgenine ait piksel değerlerinin toplamı, imgeye TI'ye ait dört değerle şöyle hesaplanabilir:

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

Benzer durum Şekil 3 (c)'deki eğimli D_2 dikdörtgeni için de geçerlidir ve dikdörtgendeki piksel toplamı, imgenin ETI değerleri cinsinden kolayca hesaplanabilir:

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

2.4. Öznitelik değerlerinin normalizasyonu

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

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

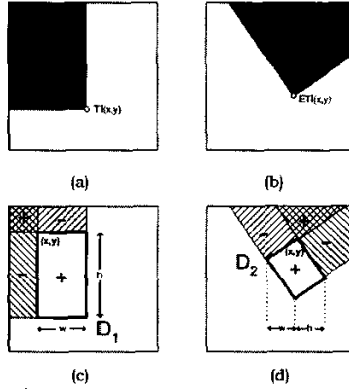
işlemi uygulanmaktadır. Burada μ tarama penceresi içerisindeki piksellerin toplamı, σ ise bu piksellerin standart sapmasıdır. Tarama penceresine düşen piksel değerlerinin standart sapması T_I ve Karesel T_I (KTİ) değerleri kullanılarak kolayca hesaplanabilir. Bu nedenle düzgeleme katsayısının hızlı bir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

$$KT_I(x,y) = \sum_{(i,j) \in S} I'(i,j) \quad (8)$$

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

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

$S(x,y)$ tarama penceresinin (x,y) konumundayken kapsadığı piksel kümesidir.



Sekil 3 (a) T_I 'nin (x,y) noktası için değerleri toplanan imge bölgesi, (b) ET_I 'nin (x,y) noktası için değerleri toplanan imge bölgesi, (c) Dikey görünümü öznitelik hesabı için gereken bir dikdörtgenin T_I değeri cinsinden ifadesi (d) Eğimli görünümü öznitelik hesabı için gereken bir dikdörtgenin ET_I değeri cinsinden ifadesi.

3. Sınıflandırıcılar

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

3.1. Basit Sınıflandırıcı Ağacı

Bir giriş imgesi x değeri ağaç düğümlerinde bulunan

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

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

yönünü göstermektedir. Düğümlerdeki öznitelik kıyaslayıcı basit sınıflandırıcı $h_j \cdot x$ 'ler tarafından verilen 1 kararları ağacın sol düğümlüne, 0 kararları ise ağacın sağ düğümlüne atlanmasını sağlayarak, yaprak düğümlere değin devam eder. Soldaki yaprak düğüme düşen x değerleri için pozitif sınıf kararı, sağdaki yaprak düğüme düşenler için ise negatif sınıf kararı verilmektedir.

Tablo 1 AdaBoost tekniği kullanarak basit sınıflandırıcı ağaçlarından oluşan pekiştirilmiş sınıflandırıcının eğitim yordamı.

1) Pozitif ve negatif sınıfa ait eğitim kümesi örnekleri $(x_1, y_1), \dots, (x_n, y_n)$ topla. $y_i = \{0, 1\}$ pozitif ve negatif örneklere ait sınıf etiketlerini göstermektedir.
2) Eğitim kümesi örneklerine ait ağırlık katsayılarını $w_i = \begin{cases} \mu_{+} & \text{eğer } y_i = 1 \\ \mu_{-} & \text{eğer } y_i = 0 \end{cases}$ olacak şekilde ata.
3) Tüm eğitim kümesini kullanarak tüm öznitelik kümesi için eşik değerlerini belirleyerek basit sınıflandırıcı kümesini $\{h_j\}$ 'yi oluştur ve sakla.
4) $k=1, \dots, N$ için <ul style="list-style-type: none"> a) Eğitim kümesi ağırlıklarını toplamdan 1 olacak şekilde düzgele. b) Bir ağaç sınıflandırıcı $t_k(x)$'e ait kök düğümü oluştur ve ağaç seviyesi göstergesini sıfırla. <ul style="list-style-type: none"> i) O anki ağaç seviyesi için düğüm listesi çıkar. Listedeki tüm düğümler için: <ul style="list-style-type: none"> 1) Düğüme ulaşan eğitim kümesi elemanları ile basit sınıflandırıcılar için $\epsilon_j = \sum_i w_i h_j(x_i) - y_i$ hatalarını belirle. 2) Minimum hata ϵ_{\min} veren öznitelik sınıflandırıcısını ilgili düğüme ata. ii) Mevcut ağacın eğitim kümesi üzerindeki hatası 0.4'den büyükse o anki düğüm listesinde bulunan düğümler için çocuk düğümleri oluştur, ağaç seviyesi göstergesini bir artır, i) nci adıma geri dön, hata 0.4'den küçükse c) adımına geç. c) Bulunan ağaç sınıflandırıcı $t_k(x)$'i kullanarak eğitim kümesi ağırlıklarını $w_{i+1} = \begin{cases} w_i \cdot \beta & x \text{ doğru sınıflandırılmışsa} \\ w_i & x \text{ yanlış sınıflandırılmışsa} \end{cases} \quad \beta = \frac{\epsilon_{\min}}{1 - \epsilon_{\min}}$ olacak şekilde güncelle.
5) Oluşturulan kuvvetli sınıflandırıcı $H(x) = \begin{cases} 1 & \sum_{k=1}^N \alpha_k t_k(x) \geq \frac{1}{2} \sum_{k=1}^N \alpha_k \\ 0 & \text{aksi takdirde} \end{cases}$

3.2. Kuvvetli sınıflandırıcı katları

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

Tablo 2 Çok katlı sınıflandırıcıyı oluşturmak için kullanılan eğitim yordamı.

1. Her sınıflandırıcı katının gerçekleştirilmesi gereken <ul style="list-style-type: none"> a) maksimum yanlış kabul oranı f b) minimum sezim oranı d parametrelerini belirle.
2. Tüm çok katlı sınıflandırıcının gerçekleştirilmesi gereken hedef yanlış kabul F_{hedef} parametresini belirle.

3. Oluşturulan tüm sınıflandırıcılara 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_e adet negatif, P_e adet pozitif sınıfa ait test kümesi örneği topla.

5. Kat sayacı s 'yi sıfırda

6. Kuvvetli sınıflandırıcı katına ait başarımlı ölçüt değişkenlerini 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. $F > F_{\text{hedef}}$ olduğu sürece

a) Kuvvetli sınıflandırıcı katı ve ilk yeni sınıflandırıcısını oluştur.

b) Eğitilecek yeni sınıflandırıcı olduğu sürece

i) Ağaç sınıflandırıcıyı eğitim kümesini kullanarak oluştur.

ii) Eğer ($d_e = 1$ ve $f_e < f$) ise

1) $d_t \geq d$ olacak şekilde sınıflandırıcı katının eşik değerini ayarla.

2) Eğer $f_t > f$ ise kata yeni bir ağaç sınıflandırıcı ekle.

iii) Eğer ($d_e = 1$ ve $f_e < f$) değil ise

1) ise kata yeni bir ağaç sınıflandırıcı ekle.

iv) b) adımına geri dön.

c) F 'yi $F * f_e$ olacak şekilde güncelle.

d) Eğer $F > F_{\text{hedef}}$ ise oluşan çok katlı sınıflandırıcıyı kullanarak eğitim kümesi için ayrılan yüz olmayan resimlerden N_e adet yanlış alarm üreten 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 üreten negatif örnek topla. Ve 6ncı adıma dön.

3.3. Ardışık Çok Katlı Sınıflandırıcı

Pekiştirilmiş sınıflandırıcı katlarını art arda ekleyerek küçük yanlış kabul olasılığı ve yüksek doğru karar oranlı 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 az, gerçekleştirilebilir. Bu amaçla oluşturulacak her kat, kendisinden daha önceki katlardan oluşan tümeşik 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 başarımlı ölçütleri ile öğrenmektedir.

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

4. Deneyler ve Sonuçlar

Eğitim kümesini oluşturmak amacıyla FERET [13] ve ORL veritabanında bulunan ön görünümlü yüzler içeren resimlerden 20x20 boyutlarında hizalanmış ve ölçeklenmiş şekilde yüz resimleri çıkartılmıştır. 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 için sağlanması gereken performans kriterleri $f=0.5$, $d=0.995$, $F_{\text{hedef}} \approx 10^{-4}$ olarak alınmıştır.

Eğitim sonucunda ortaya çıkan katlı sınıflandırıcının ilk sekiz katına ilişkin, sınıflandırıcı katlarındaki ağaç sayısı ve içerdikleri öznelilik 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	Ağaç sayısı	Öznelilik 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'inden çıkarılan önden görünümlü 230 yüz içeren 80 resim üzerinde denenmiştir. Elde edilen performans değerleri Tablo 4'de verilmektedir. Performans değerinin diğer yöntemlere kıyasla aynı sezim oranı için daha fazla yanlış kabul sayısı ürettiği gözlemlenmiştir. Sezim oranının iyileştirilmesi ve yöntemin farklı yüz pozisyonları üzerinde çalışabilmesi için çalışılması düşünülmektedir.

Tablo 4 Sınıflandırıcının ve diğer [6], [1] başarımlı 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. Kuranov, V. Pisarevsky, *Empirical Analysis of Detection of Boosted Classifiers for Rapid Object Detection*, MRL Technical Report, December 2002.
- [3] E. Hjeltnas, B. Kee 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.