

Ardışıl Sınıflandırıcılar Kullanılarak Profil Yüz Sezimi

Sercan Baykara

YÜKSEK LİSANS TEZİ

Elektrik-Elektronik Mühendisliği Anabilim Dalı

Mayıs 2018

Profile Face Detection Using Cascade Classifiers

Sercan Baykara

MASTER OF SCIENCE THESIS

Department of Electrical Electronics Engineering

May 2018

Ardışıl Sınıflandırıcılar Kullanılarak Profil Yüz Sezimi

Sercan Baykara

Eskişehir Osmangazi Üniversitesi
Fen Bilimleri Enstitüsü
Lisansüstü Yönetmeliği Uyarınca
Elektrik-Elektronik Mühendisliği Anabilim Dalı
Telekomünikasyon ve Sinyal İşleme Bilim Dalında
YÜKSEK LİSANS TEZİ
Olarak Hazırlanmıştır

Danışman: Doç. Dr. Hakan Çevikalp

Mayıs 2018

ONAY

Elektrik ve Elektronik Mühendisliđi Anabilim Dalı Yüksek Lisans öđrencisi Sercan Baykara'nın YÜKSEK LİSANS tezi olarak hazırladıđı "Ardışıl Sınıflandırıcılar Kullanılarak Profil Yüz Sezimi" başlıklı bu çalıřma, jürimizce lisansüstü yönetmeliđin ilgili maddeleri uyarınca deđerlendirilerek oybirliđi ile kabul edilmiřtir.

Danıřman : Doç. Dr. Hakan Çevikalp

İkinci Danıřman : -

Yüksek Lisans Tez Savunma Jürisi:

Üye : Doç. Dr. Hakan Çevikalp

Üye : Dr. Öğr. Üyesi Hasan Serhan Yavuz

Üye : Dr. Öğr. Üyesi Cihan Topal

Fen Bilimleri Enstitüsü Yönetim Kurulu'nun tarih ve
..... sayılı kararıyla onaylanmıřtır.

Prof. Dr. Hürriyet ERŐAHAN
Enstitü Müdürü

ETİK BEYAN

Eskişehir Osmangazi Üniversitesi Fen Bilimleri Enstitüsü tez yazım kılavuzuna göre, Doç. Dr. Hakan Çevikalp danışmanlığında hazırlamış olduğum “Ardışıl Sınıflandırıcılar Kullanılarak Profil Yüz Sezimi” başlıklı Yüksek Lisans tezimin özgün bir çalışma olduğunu; tez çalışmamın tüm aşamalarında bilimsel etik ilke ve kurallara uygun davrandığımı; tezimde verdiğim bilgileri, verileri akademik ve bilimsel etik ilke ve kurallara uygun olarak elde ettiğimi; tez çalışmamda yararlandığım eserlerin tümüne atıf yaptığımı ve kaynak gösterdiğimi ve bilgi, belge ve sonuçları bilimsel etik ilke ve kurallara göre sunduğumu beyan ederim. 17.05.2018

Sercan Baykara

ÖZET

Yüz sezimi, verilen herhangi bir imge içerisinde yüz bulunup bulunmadığına karar vermek, eğer varsa yüzün pozisyonunu ve kapladığı bölgeyi belirlemektir. Bu tezde iki farklı öznelik çıkarma yöntemi birleştirilmiş ve üç farklı ardışıl sınıflandırıcılar kullanılarak sağ ve sol profil yüz sezimi algoritması tasarlanmıştır. Tez kapsamında profil yüz sezme sistemini eğitmek için internet üzerinden toplanan günlük hayattan çekilmiş imgeler kullanılmış, bu imgelerde bulunan sağ ve sol profil yüzlerine ait öznelikler Yerel İkili Örüntüler (YİÖ) ve Yönlü Gradyan Histogramları (YGH) betimleme teknikleri yardımıyla ayrı ayrı çıkarılarak öznelik vektörleri elde edilmiş, elde edilen bu öznelikler vektörleri birleştirilerek tek bir öznelik vektörü (YİÖ+YGH) oluşturulmuştur. Çalışmada imgelere ait özneliklerin çıkarılmasından sonra imgelerdeki yüzlerin konumlarını belirlemede kayan penceler yönteminden yararlanılmış, ardışıl sınıflandırıcı olarak ise sırasıyla Doğrusal Destek Vektör Makineleri (DDVM), Doğrusal Hiper Küre ve Doğrusal Olmayan Hiper Küre kullanılmıştır. Tasarlanan yüz sezimi algoritmasının başarımını ölçmek için PASCAL VOC metriği ve Precision-Recall eğrilerinden elde edilen ortalama kesinlik skorlarından yararlanılmıştır. Yapılan deneysel çalışmalarda internet üzerinden toplanan eğitim setinde kullanılmayan günlük hayattan çekilmiş rastgele imgeler üzerinde algoritma test edilmiş OPENCV ile birlikte gelen Viola&Jones tarafından önerilen ardışıl sınıflandırıcıdan çok daha iyi sonuçlar elde edilmiştir.

Anahtar Kelimeler: Doğrusal Destek Vektör Makineleri, Doğrusal Hiper-Küre, Doğrusal Olmayan Hiper-Küre, Ardışıl Sınıflandırıcılar, Profil Yüz sezimi

SUMMARY

Face detection is to decide if there is a face in any given image and to determine the position of the face and the area it covers. In this thesis, profile face detection is designed by combining two different feature extracting methods and using a cascade of three different classifiers. Within the scope of thesis, images collected over the internet are used to train the face detection system. The features of right and left profile faces in these images are extracted by using Local Binary Pattern (LBP) and Histogram of Oriented (HOG) methods. Then, these feature vectors are combined to create a single feature vector (LBP+HOG). After creating combined feature vector, the sliding window method is used to determine the positions of the profile faces in the images. To classify the images Linear Support Machine (LSVM), Linear Hypersphere (LHS) and Nonlinear Hypersphere (NHS) methods are used respectively in a cascade structure. To measure the performance of the designed face detection algorithm, the PASCAL VOC metric and Precision-Recall curves are used. The experimental studies have shown that the proposed algorithm gives much better results than Viola & Jones profile face detector, which comes with OPENCV, on test images that are not used in our training set collected over the internet.

Keywords: Linear Support Vector Machine (LSVM), Linear Hypersphere (LHS), Nonlinear Hypersphere (NHS), Cascade classifiers, Profile face detection

TEŞEKKÜR

Çalışmalarım boyunca farklı bakış açıları ve bilimsel katkılarıyla beni aydınlatan, yakın ilgi ve yardımlarını esirgemeyen ve bu günlere gelmemde en büyük katkı sahibi sayın hocam Doç. Dr. Hakan ÇEVİKALP'e teşekkürü bir borç bilirim.

Tez çalışmama ait tüm süreçte bir an bile destek ve yardımlarını esirgemeyen ve her zaman beni motive ederek giriştiğim tüm çalışmalarımın başarıyla üstesinden gelmemi sağlayan sevgili eşim Canset KOÇER BAYKARA'ya en derin teşekkürlerimi sunarım.

Ayrıca; çalışmalarım süresince sabrını, sevgisini ve desteğini esirgemeyerek beni daima destekleyen annem ve babama içtenlikle teşekkür ederim.

İÇİNDEKİLER

| | <u>Sayfa</u> |
|--|--------------|
| ÖZET | vi |
| SUMMARY | vii |
| TEŞEKKÜR | viii |
| İÇİNDEKİLER | ix |
| ŞEKİLLER DİZİNİ | xi |
| ÇİZELGELER DİZİNİ | xiii |
| SİMGELER VE KISALTMALAR DİZİNİ | xv |
| 1.GİRİŞ VE AMAÇ | 1 |
| 2.LİTERATÜR ARAŞTIRMASI | 3 |
| 2.1 Genel Yüz Sezme Sistemleri | 3 |
| 2.1.1 Genel yüz sezme sisteminin temel birimleri | 3 |
| 2.1.2 Yüz sezme sistemlerini zorlaştıran faktörler | 6 |
| 2.2. Literatürdeki Yüz Sezme Yöntemleri | 7 |
| 3. MATERYAL VE YÖNTEM | 14 |
| 3.1 Veri Tabanı Oluşturulması | 16 |
| 3.1.1 Eğitim veri tabanı | 16 |
| 3.1.2 Test veri tabanı | 20 |
| 3.2 İmge Betimleme (Öznitelik Çıkarımı) | 21 |
| 3.2.1 Yerel İkili Örüntüler (YİÖ) | 21 |
| 3.2.2 Yönlü Gradyan Histogramları (YGH) | 24 |
| 3.3 Sınıflandırma | 25 |
| 3.3.1 Destek Vektör Makineleri (DVM) | 25 |
| 3.3.2 Doğrusal Hiper-Küre sınıflandırıcısı | 35 |
| 3.3.3 Doğrusal Olmayan Hiper-Küre sınıflandırıcısı | 38 |
| 4.BULGULAR VE TARTIŞMA | 40 |
| 4.1 Sağ Profil Yüz Sezme Test Sonuçları | 44 |
| 4.2 Sol Profil Yüz Sezme Test Sonuçları | 50 |
| 4.3 Viola&Jones Profil Yüz Sezme Test Sonuçları | 55 |
| 4.4 Profil Yüz Sezme Yöntemlerinin Karşılaştırılması | 58 |
| 5. SONUÇ VE ÖNERİLER | 59 |

İÇİNDEKİLER (devam)

| | <u>Sayfa</u> |
|---|---------------------|
| KAYNAKLAR DİZİNİ | 60 |
| EK AÇIKLAMALAR | 64 |
| Ek Açıklama-A: Yerel İkili Örüntüler (YİÖ) Matlab Kodları..... | 65 |
| Ek Açıklama-B: Yönlü Gradyan Histogramları (YGH) Matlab Kodları | 68 |
| Ek Açıklama-C: (YİÖ+YGH) Matlab Kodları | 71 |
| Ek Açıklama-D: Sınıflandırma Eğitimi İçin Kullanılan Matlab Kodları | 72 |



ŞEKİLLER DİZİNİ

| <u>Sekil</u> | <u>Sayfa</u> |
|---|--------------|
| 2.1. Günümüz yüz tanıma sistemleri uygulamaları basamakları..... | 3 |
| 2.2. Genel yüz sezme sisteminin temel birimleri..... | 3 |
| 2.3. Sınır tespiti örneği | 4 |
| 2.4. Aydınlatma normalleştirme örneği | 4 |
| 2.5. Histogram eşitleme örneği..... | 5 |
| 2.6. Medyan filtreleme örneği | 5 |
| 2.7. Yüz sezme yöntemleri | 7 |
| 3.1. Çalışmada kullanılan yöntemle ait şematik gösterim | 14 |
| 3.2. Çalışma kapsamında kullanılan ardışıl sınıflandırıcı..... | 15 |
| 3.3. Annotation tool programı arayüzü..... | 16 |
| 3.4. Veri tabanında kullanılan imgelerin oluşturulması | 17 |
| 3.5. Eğitim veri tabanındaki pozitif imgelerden örnekler..... | 19 |
| 3.6. Test veri tabanı için web ortamından toplanan resimlerden örnekler | 20 |
| 3.7. Test veri tabanındaki pozitif imgelerden örnekler ve saklanan koordinatları | 21 |
| 3.8. Çeşitli dairesel YİÖ P,R operatörleri..... | 22 |
| 3.9. YİÖ operatörünün elde edilmesi ve bitlerin ağırlık katsayıları | 23 |
| 3.10. YİÖ histogramının elde edilmesi..... | 23 |
| 3.11. YGH algoritma uygulama şekli..... | 24 |
| 3.12. İki sınıflı bir veri setini ikiye ayıran örnek doğrular (Öğüdücü, 2007) | 27 |
| 3.13. Farklı sınıflar arasındaki uzaklığı en iyi ayıran düzlem: B1 (Öğüdücü, 2007)..... | 28 |
| 3.14. Doğrusal ayrılabilen iki sınıflı veri setlerinin genel gösterimi (Öğüdücü, 2007)..... | 29 |
| 3.15. Verilerin doğrusal ayrılabilen durumu genel gösterimi (Öğüdücü, 2007)..... | 31 |
| 3.16. Doğrusal Hiper-Küre sınıflandırıcısı anlatımı (Chen vd.,2011)..... | 35 |

ŞEKİLLER DİZİNİ (devam)

| <u>Şekil</u> | <u>Sayfa</u> |
|---|---------------------|
| 4. 1. Maksimum olmayanı bastırma uygulanmamış yüz sezme algoritması çıktıları..... | 41 |
| 4. 2. Sağ ve sol profil yüz sezme sistemi tarafından başarı ile bulunan örnekler..... | 42 |
| 4. 3. Sağ ve sol profil yüz sezme sistemi tarafından hatalı bulunan örnekler | 43 |
| 4. 4. Sağ profil test 1-7 Precision-Recall Eğrileri grafiği | 48 |
| 4. 5. Aynı örnek üzerindeki sağ profil test 1-7 çıktıları..... | 49 |
| 4. 6. Sol profil test 2-4-7 Precision-Recall Eğrileri grafiği | 54 |
| 4. 7. Aynı örnek üzerindeki sol profil test 1-7 çıktıları | 55 |
| 4. 8. Viola&Jones Precision-Recall Eğrileri grafiği | 56 |
| 4. 9. Viola&Jones profil yüz sezme test çıktıları..... | 57 |

ÇİZELGELER DİZİNİ

| <u>Cizelge</u> | <u>Sayfa</u> |
|---|--------------|
| 4.1. Karışıklık -hata matrisi (confusion-error matrix) (Öğüdücü, 2007)..... | 40 |
| 4.2. Test 1 Sağ profil yüz sezme kullanılan değerler..... | 45 |
| 4.3. Test 1 sağ profil yüz sezme test sonuçları | 45 |
| 4.4. Test 2 sağ profil yüz sezme kullanılan değerler | 45 |
| 4.5. Test 2 sağ profil yüz sezme test sonuçları | 45 |
| 4.6. Test 3 sağ profil yüz sezme kullanılan değerler | 46 |
| 4.7. Test 3 sağ profil yüz sezme test sonuçları | 46 |
| 4.8. Test 4 sağ profil yüz sezme kullanılan değerler | 46 |
| 4.9. Test 4 sağ profil yüz sezme test sonuçları | 46 |
| 4.10. Test 5 sağ profil yüz sezme kullanılan değerler | 46 |
| 4.11. Test 5 sağ profil yüz sezme test sonuçları | 47 |
| 4.12. Test 6 sağ profil yüz sezme kullanılan değerler | 47 |
| 4.13. Test 6 sağ profil yüz sezme test sonuçları | 47 |
| 4.14. Test 7 sağ profil yüz sezme kullanılan değerler | 47 |
| 4.15. Test 7 sağ profil yüz sezme test sonuçları | 47 |
| 4.16. Sağ profil yüz sezme test sonuçları | 48 |
| 4.17. Test 1 sol profil yüz sezme kullanılan değerler | 50 |
| 4.18. Test 1 sol profil yüz sezme test sonuçları..... | 50 |
| 4.19. Test 2 sol profil yüz sezme kullanılan değerler | 51 |
| 4.20. Test 2 sol profil yüz sezme test sonuçları..... | 51 |
| 4.21. Test 3 sol profil yüz sezme kullanılan değerler | 51 |
| 4.22. Test 3 Sol profil yüz sezme test sonuçları | 51 |
| 4.23. Test 4 sol profil yüz sezme kullanılan değerler | 52 |
| 4.24. Test 4 sol profil yüz sezme test sonuçları..... | 52 |
| 4.25. Test 5 sol profil yüz sezme kullanılan değerler | 52 |
| 4.26. Test 5 sol profil yüz sezme test sonuçları..... | 52 |
| 4.27. Test 6 sol profil yüz sezme kullanılan değerler | 53 |
| 4.28. Test 6 sol profil yüz sezme test sonuçları..... | 53 |
| 4.29. Test 7 sol profil yüz sezme kullanılan değerler | 53 |

ÇİZELGELER DİZİNİ (devam)

| <u>Cizelge</u> | <u>Sayfa</u> |
|--|---------------------|
| 4.30. Test 7 sol profil yüz sezme test sonuçları..... | 53 |
| 4.31. Sol profil yüz sezme test sonuçları | 54 |
| 4.32. Viola&Jones profil yüz sezme test sonuçları | 56 |
| 4.33. Profil yüz sezme yöntemlerinin karşılaştırılması | 58 |



SİMGELER VE KISALTMALAR DİZİNİ

| <u>Kısaltmalar</u> | <u>Açıklama</u> |
|--------------------|---|
| BPNN | Back Propagation Neural Network |
| DVM | Destek Vektör Makineleri |
| DDVM | Doğrusal Destek Vektör Makineleri |
| FFNN | Feed Forward Neural Network |
| GIF | Graphic Interchange Format |
| GPU | Graphic Processing Unit |
| HOG | Histogram of Oriented |
| HSV | Hue Saturation Value |
| JPEG | Joint Photographic Expert Group |
| LBP | Local Binary Pattern |
| LHS | Linear Hypersphere-Doğrusal Hiper Küre |
| LSVM | Linear Support Vector Machines |
| NHS | Nonlinear Hypersphere-Doğrusal Olmayan Hiper Küre |
| NN | Neural Network |
| PNG | Portable Network Graphics |
| RGB | Red Green Blue |
| SOM | Self Organizing Map |
| SVM | Support Vector Machines |
| TBA | Temel Bileşenler Analizi |
| TIFF | Tagged Image File Format |
| VOC | Visual Object Classes |
| YCbCr | Y: Luminance, Cb: Chroma Blue, Cr: Chroma Red |
| YGH | Yönlü Gradyan Histogramları |
| YİÖ | Yerel İkili Örüntüler |
| YUV | Y: Luminance, U: Chrominance1, V: Chrominance2 |

1.GİRİŞ VE AMAÇ

Yüz sezimi sayısal belirli bir imgede insan yüzünün varlığını, konumunu ve ölçeğini tespit etmek için kullanılan bir bilgisayarlı görü uygulamasıdır. Gelişen teknoloji ile birlikte, askeri, ticari, adli, güvenlik ve robotik uygulama alanlarında yüz tanıma teknolojilerinin artmasıyla birlikte yüz tanıma sistemlerinin ilk aşaması olan yüz sezimi popüler bir konu haline gelmiştir. Giriş kontrol sistemleri, kalabalık gözetim otomasyonları, multimedya haberleşmesi (yapay yüzlerin oluşturulması) ve içerik tabanlı yüz veri tabanlarının yönetimi gibi birçok uygulamada gelişmiş yüz tanıma teknolojilerinin kullanılmasına ihtiyaç duyulmuş ve bu alanda geçtiğimiz son 35 yıl boyunca oldukça fazla sayıda çalışma ve araştırma yapılmıştır. Son yıllarda yapılan birçok önemli gelişme ve çalışmaya rağmen yüz tanıma için bilgisayar tabanlı bir modelin oluşturulması yüz bilgilerinin karmaşık ve değişken olmalarından ötürü oldukça zor bir süreçtir. Çoğu yüz tanıma uygulamaları ellerinde yeterince kaliteli ve normalize edilmiş yüz resimleri oldukları varsayımı altında çalışmaktadırlar. Bu durum yüz tanıma uygulamalarının başarımlarının yüz sezimi uygulamalarının başarımına fazlasıyla bağımlı olduğu gerçeğini ortaya çıkarmaktadır.

Son yıllarda özellikle iki yöntem yüz sezimi uygulamalarında yüksek başarımları ve verimliliği ile ön plana çıkmıştır: Destek Vektör Makineleri (DVM) ve Boosting yaklaşımını kullanan ardışıl sınıflandırıcılar Viola&Jones (Viola ve Jones, 2004). Viola ve Jones, yüz sezme için dalgacık dönüşümüne benzer öznitelikler ve AdaBoost tekniğini kullanarak ardışıl bir sınıflandırıcı dizayn etmiştir. Bu sınıflandırıcının ön katmanlarındaki basit sınıflandırıcılar, son katmanlarda ise daha güçlü sınıflandırıcılar kullanılmıştır. Destek Vektör Makineleri de yüz sezimi uygulamalarında oldukça sık kullanılan iki sınıfla çalışan başarılı bir sınıflandırıcı yöntemidir.

Bu tez çalışmasındaki amaç, iki farklı öznitelik çıkarma yönteminin birleştirilmesi (Yerel İkili Örüntüler + Yönlü Gradyan Histogramları) ve üç farklı ardışıl ikili ve tek sınıflı sınıflandırıcı kullanılmasıyla (Doğrusal Destek Vektör Makineleri + Doğrusal Hiper Küre + Doğrusal Olmayan Hiper Küre) özgün bir profil yüz sezimi algoritması tasarlayarak, söz konusu bu algoritmanın literatürde en çok bilinen yüz sezme

algoritmalarından biri olan Viola&Jones tarafından önerilen AdaBoost tabanlı ardışıl sınıflandırıcılı profil yüz sezme algoritması ile aynı şartlar altında Matlab ortamında karşılaştırılması sonrası elde edilen sonuçlara göre daha iyi bir alternatif olduğunu göstermektedir.

Bu tez çalışması 4 ayrı bölümden oluşmaktadır. Bunlar sırayla; **Literatür Araştırması, Materyal ve Yöntem, Bulgular ve Tartışma** ve en son ise **Sonuçlar ve Öneriler** bölümleridir. Her bölümün genel içeriği aşağıda belirtilmiştir.

İlk olarak, **Literatür Araştırma Bölümü**'nde yüz sezimi sistemi genel tanımları, yüz sezme sistemini etkileyen faktörler ve literatürde yaygın olarak kullanılan yüz sezimi algoritmaları ele alınmıştır.

Daha sonra, bu tez kapsamında tasarlanan profil yüz sezimi algoritmasını oluşturan kısımların detaylı bilimsel açıklamalarına ve algoritmanın nasıl oluşturulduğuna dair detaylı anlatıma **Materyal ve Yöntem Bölümü**'nde yer verilmiştir.

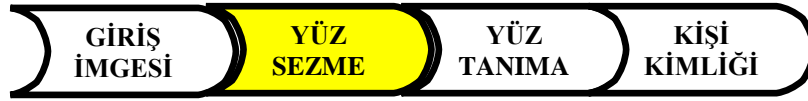
Bulgular ve Tartışma Bölümü'nde ise, tasarlanan yüz sezimi algoritmasıyla Matlab ortamında elde edilen test sonuçları ve Viola&Jones tarafından önerilen AdaBoost tabanlı ardışıl sınıflandırıcılı profil yüz sezme algoritması ile bu tez kapsamında tasarlanan yüz sezimi algoritmasının Matlab ortamında aynı şartlar altında karşılaştırılması sonrası elde edilen sonuçlar yer almaktadır.

Son olarak, tezin son bölümünü oluşturan **Sonuçlar ve Öneriler Bölümü**'nde bu tez çalışmasından elde edilen sonuçlardan bahsedilmiştir. Önerilen yöntemin daha sonraki çalışmalara yararlı bilgiler sağlaması ve geliştirilmesi için önerilerde bulunularak tez tamamlanmıştır.

2.LİTERATÜR ARAŞTIRMASI

2.1 Genel Yüz Sezme Sistemleri

Günümüzde gelişen teknolojiyle birlikte geliştirilen yüz tanıma sistemlerinin önemli bir parçasını Şekil 2.1’de görüldüğü üzere yüz sezme sistemleri oluşturmaktadır.

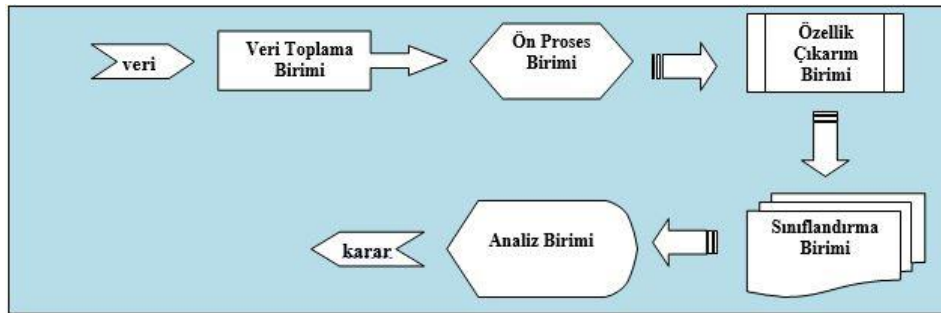


Şekil 2. 1. Günümüz yüz tanıma sistemleri uygulamaları basamakları

Yüz sezme sistemlerinin amacı, sisteme rastgele bir imge verildiğinde, verilen imgede herhangi bir yüz olup olmadığını belirlemek ve mevcutsa, imgede belirlenen yüzün konumunu bir çerçeve ile döndürmektir.

2.1.1 Genel yüz sezme sisteminin temel birimleri

Bir yüz sezme sistemi, imge bilgisinden belirli özellik dizisini süzen, bu özellik dizisini veri tabanındaki diğer özellik dizileriyle karşılaştıran ve bu karşılaştırma sonucuna dayanarak karar verme kuralına göre yürüten bir yapıya sahiptir. Genel bir yüz sezme sistemi Şekil 2.2’de görülmektedir. Beş birimden oluşan sistem içeriğinde: Veri Toplama Birimi, Ön Proses Birimi, Özellik Çıkarma Birimi, Sınıflandırma Birimi ve Analiz Birimi yer almaktadır. Bu birimlerin her biri aşağıda açıklanmıştır:



Şekil 2. 2. Genel yüz sezme sisteminin temel birimleri

Veri Toplama Birimi: Veri toplama birimi insan makine ara yüzünü tanımlar, bundan dolayı, yüz sezme sisteminin performansı için çok önemlidir. Uygun bir kamera veya tarayıcı kullanılarak imgenin ham bilgisi elde edilir. Kameranın teknik özellikleri de edinilen veri kalitesi açısından çok önemlidir.

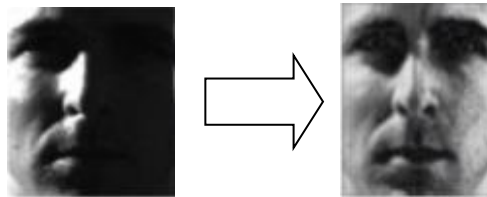
Ön Proses (İşlem) Birimi: Veri toplama birimi tarafından elde edilen bilginin kalitesi istenilen düzeyde olmayabilir. Bu verinin, ileriki işlemler için uygun olup olmadığının tanımlanması amacıyla öncelikle sisteme özgü hesaplama yapılır. Buna ek olarak, elde edilen bilgi kalitesini yükseltmek için bir sinyal iyileştiriciye tabi tutulur. Aşağıda bazı ön işleme yöntemleri anlatılmaktadır.

- **İmge boyutu normalizasyonu:** Veri toplama birimi yardımıyla alınan imgenin yüz sezme sisteminin üzerinde işlem yapabileceği boyutlara getirilmesi işlemidir.
- **Sınır Tespiti (Edge Detection):** İmge işlemenin en önemli ön işleme yöntemlerinden biri olup imge üzerindeki sınırları ayırt etmek için kullanılır. Şekil 2.3'te sınır tespiti örneği gösterilmektedir.



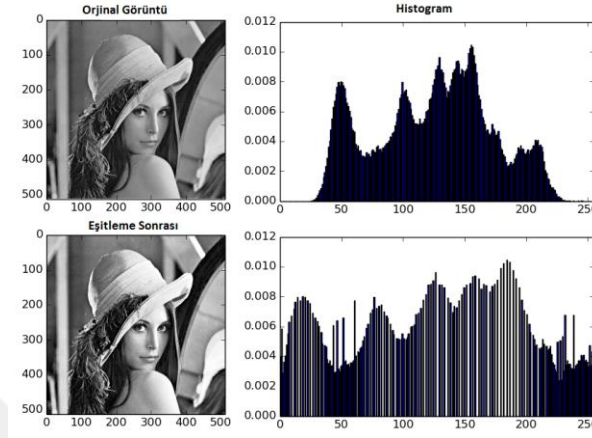
Şekil 2. 3. Sınır tespiti örneği

- **Aydınlatma Normalleştirme (Illumination Normalization):** Bu ön işleme, imge tabanlı sistemler için özel bir işlemdir. Aydınlatma normalleştirme sisteminin genel amacı, gözlemlenen imgenin farklı çevrelerde yakalandığında aydınlatma etkilerini azaltmaktır. Şekil 2.4'te aydınlatma normalleştirme örneği gösterilmektedir.



Şekil 2. 4. Aydınlatma normalleştirme örneği

- Histogram Eşitleme (Histogram Equalization) : İmgenin her yoğunluk düzeyinin eşit sayıda piksel içermesi için yapılan işleme histogram eşitleme adı verilir. Şekil 2.5'te histogram eşitleme örneği gösterilmektedir.



Şekil 2. 5. Histogram eşitleme örneği

- Medyan Filtreleme (Median Filtering): Ortanca filtreleme basit ve çok etkili bir şekilde gürültü gidererek filtreleme işlemidir. Normal olarak imgedeki gürültüyü azaltmak için kullanılır. Dürtü ve tuz-biber gürültülerinin giderilmesinde etkin başarımlar sağlamaktadır. Şekil 2.6'da medyan filtreleme örneği gösterilmektedir.



Şekil 2. 6. Medyan filtreleme örneği

Öznitelik Çıkarım Birimi: Ön işleme basamağından sonra yüze ait özellikleri ifade etmek için bir dizi belirgin ayırıcı öznitelik çıkarılır. Örneğin, bazı yüz sezme sistemlerinde bir yüz imgesindeki gözler, burun ve ağız arasındaki mesafe ve konum öznitelik çıkarım birimi ile belirlenir. Eğitim sırasında, bu öznitelik dizileri veri tabanında saklanır ve çoğunlukla şablon olarak anılır.

Sınıflandırma Birimi: Karşılaştırma sonuçları oluşturmak için depolanmış şablonlar ile çıkarım yapılmış öznitelikler karşılaştırılır. Yüz tabanlı biyometrik bir sistemde, şablon dizileri ve giriş imgesi arasındaki en küçük mesafe tespit edilir ve benzer sonuçlar kaydedilir.

Analiz Birimi: Bu birimde farklı ölçeklerden gelen kararlar birleştirilir ve üst üste gelen ve yanlış olan algılamalar silinir. Yüz sezme sistemi tarafından son değerlendirmeler yapılarak sistemin ilk veriye göre kararı verilir.

2.1.2 Yüz sezme sistemlerini zorlaştıran faktörler

İnsan beyni için yüz sezimi konusunda sorun teşkil etmeyen ancak yüz sezme sistemlerini zorlaştıran birçok faktör bulunmaktadır:

- **Yüzün başka bir cisim tarafından kapanması (occlusion):** Özellikle toplu fotoğraflarda bazı yüzler diğer yüzleri kısmen ya da tamamen kapatarak görünmesine engel olmaktadır. Ayrıca imge içinde yüzün bir kısmı güneş gözlüğü, nesne vb. tarafından kasıtlı ya da kasıtsız kapanabilmektedir.

- **Aydınlatma ve görüntüleme koşulları:** İmgedeki aydınlatma miktarı, ışığın geliş açısı ve şiddeti ile değişkenlik gösterebilmektedir. Yüzün yapısal özelliklerine ve aydınlanma kaynağına bağlı olarak yüzde oluşacak gölgeler, yüzün bazı özelliklerinin sistem tarafından yanlış değerlendirilmesine neden olabilecek problemlere yol açabilir.

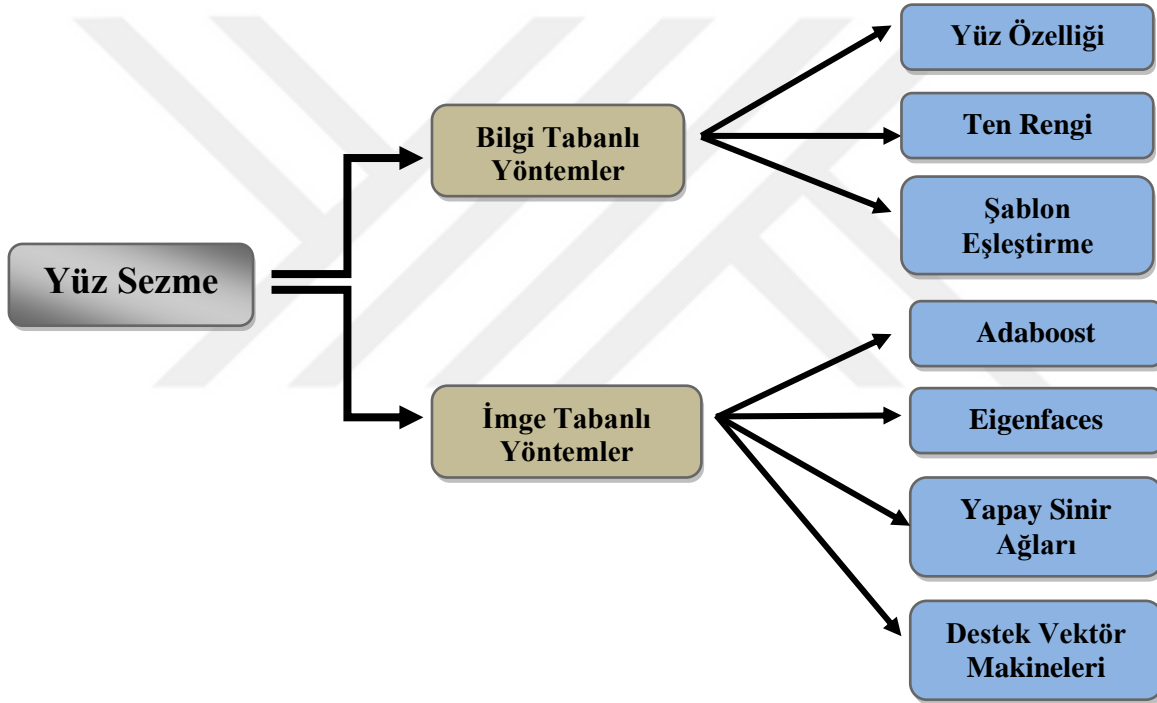
- **Yüz ifadesi:** Yüz ifadesi jest ve mimiklere göre yüz nesnesinin görünümü değişkenlik gösterebilmektedir.

- **Yapısal bileşenler:** İmgelerde bulunabilecek sakal, bıyık veya gözlükler çok çeşitli şekillerde, boyutlarda ve renklerde olabilmekle birlikte yüz sezimi performansını etkilemektedir.

- **Kafa yönelimi:** İmgelerdeki yüzler ön cepheden, yandan ya da çeşitli ara açılardan görünebilmektedir. Kameranın yüzü gördüğü açıya göre burun ya da gözler gibi bazı yapısal özelliklerin görünümü değişmektedir. Bu özellikler bazı açılar için kısmen görünebilmekte bazıları için ise hiç görünmemektedir.

2.2. Literatürdeki Yüz Sezme Yöntemleri

Yüz sezme yüz tanıma sistemlerinin ilk basamağıdır. Yüz sezme sisteminin çıktısı yüz bölgesinin tamamının ve yüz özelliklerinin (gözler, ağız, kaş, burun vb.) bulunduğu yüz bölgesinin konumu olabilir. Literatürdeki yüz sezme yöntemlerinin kesin bir şekilde sınıflandırılması oldukça zordur. Çünkü çoğu algoritma yüz seziminin doğruluğunu artırmak için birkaç farklı yöntemin birleşimi şeklindedir. Yüz sezme sistemleri, genel olarak Bilgi Tabanlı (Knowledge-Based) ve İmge Tabanlı (Image-Based) şeklinde iki grup altında sınıflandırılabilir. Şekil 2.7’de yüz sezme yöntemleri gösterilmektedir.



Şekil 2. 7. Yüz sezme yöntemleri

Bilgi Tabanlı yöntemler Yüz Özellikleri, Ten Rengi ya da Şablon Eşleştirmeye ait bilgileri kullanır. Yüz Özellikleri, insan yüzlerini sezme için gözleri, ağız, burnu ve diğer yüz hatlarını kullanır. Ten rengi diğer renklerden farklı ve eşsiz olup özelliği poz ve kapanmaya bağlı olarak değişmez. Ten rengi RGB, YCbCr, HSV, YUV gibi her bir renk uzayında ve istatistiksel modellerde modellenir. Yüz diğer nesnelere ayrılan eşsiz modeli sayesinde tarama ve yüzleri sezme için şablonu üretilebilir.

Yüz özellikleri insan yüzleri için önemli bilgilerdir ve standart imgeler bu özellikler kullanılarak oluşturulabilir. Literatürde, yüz özellik tabanlı birçok yüz sezimi algoritması bulunmaktadır. Zhi-fang vd. (2003) yüz ve yüz özelliklerinin sezimini tene benzeyen alanları YCbCr renk uzayı ile çıkartarak yapmış ve tene benzeyen alanlardaki kenarları tespit etmiştir. Daha sonra kenarları çıkartılmış alanlardaki gözler Temel Bileşenler Analizi (TBA) yöntemiyle bulunmuştur. Son olarak, ağız geometrik bilgilere bağlı olarak bulunmuştur. Diğer bir yaklaşımda, tene benzeyen alanlar Normalize RGB renk uzayı ile çıkartılmış ve yüz şablon eşleme ile doğrulanmıştır. Gözleri, kaşları ve ağız bulmak için doğrulanmış yüz imgesine renk yılanları uygulanmıştır (Seo vd., 2002). Ruan ve Yin (2009) ten alanlarını YCbCr renk uzayında parçalara ayırmış ve yüzleri Doğrusal Destek Vektör Makineleri (DDVM) ile doğrulamıştır. Yüzün son olarak doğrulanması için, gözler ve ağız Cb ve Cr fark bilgileri ile bulunmuştur. Göz alanları için Cb değeri Cr değerinden daha büyük ve ağız alanı için Cr değeri Cb değerinden daha büyüktür. Başka bir uygulama tene benzeyen alanları istatistiksel model ile parçalara ayırmıştır. İstatistiksel model Cb ve Cr kanalının YCbCr renk uzayındaki ten rengi değerlerinden yapılmıştır. Daha sonra yüz adayları parçalara ayrılmış alanın dikdörtgensel oranına göre seçilmiştir. Son olarak, adaylar göz ve ağız haritaları ile doğrulanmıştır (Zhao vd., 2006). Ayrıca, RGB renk uzayı tene benzeyen alanları ayırmak için kullanılabilir ve ten rengine benzeyen alanlar yüz adayı olabilmesi için çıkarılır. Aday yüz özellikleri bulunarak doğrulanır. Gözler ve ağız ikizkenar üçgen özelliğine bağlı olarak bulunur. Bu özelliğe göre, iki göz ve bir ağız bir ikizkenar üçgen oluşturur ve ayrıca iki göz arasındaki mesafe ile gözlerin orta noktasının ağza olan mesafesi eşittir. Gözlerin ve ağzın bulunmasından sonra, İleri Beslemeli Sinir Ağları (FFNN) kullanılarak yüz adayının son doğrulaması yapılır (Lin, 2005). Berbar vd. (2006) YCbCr renk uzayıyla imgeyi parçalara ayırmış, gözler ve ağız parçalara ayrılmış imge ve kenarlı imgenin birleşimi üzerinden bulunmuştur. Son doğrulama için, gözlerin ve ağzın pozisyonunu doğrulamak amacıyla imgenin yatay ve dikey profilleri kullanılmıştır. Tüm yöntemler işlem zamanından kazanmak adına imge içerisindeki yüz olmayan nesnelere elemek için ilk olarak ten segmentasyonunu kullanmışlardır.

Ten rengi insan yüzünün en önemli özelliklerinden birisidir. Ten rengi parametrelili ve parametresiz yöntemlerle birlikte modellenebilir. Ten rengi alanları eşik değer alanı, eliptik modelleme, istatistiksel modelleme (örn. Gauss modellemesi) ya da Sinir Ağları

cinsinden tespit edilebilir. Ten rengi, RGB, YCbCr ve HSV gibi renk uzayları içerisinde tanımlanabilir. RGB ışık değişimlerine karşı hassas olmasına rağmen YCbCr ve HSV ışık değişimlerine karşı hassas değildir. Buna sebep olarak iki renk uzayının ayrı yoğunluklara ve renk kanalına sahip olması gösterilebilir. Literatürde birçok ten rengi tabanlı algoritma mevcuttur. Kherchaoui ve Houacine (2010) ten rengini YCbCr renk uzayındaki Cb ve Cr kanalları ile birlikte Gauss Dağılım Modelini kullanarak modellemiştir. Daha sonra ten benzeri alanlar alanın sınırlayıcı kutu oranına göre yüz adayları olarak seçilmiş ve adaylar şablon eşlemeyle doğrulanmıştır. Başka bir yöntem verilen imgeyi ön işlemlerden geçirerek ilk basamakta arka plan kısmını çıkarır. Bu işlem YCbCr renk uzayındaki Y bileşeni üzerinde kenar tespiti uygulanarak yapılır. Sonra, kapalı alanlar ön plan kısmı olarak alınmak için doldurulur. Bundan sonra YCbCr şartlı renk uzayı üzerinde ten segmentasyonu yapılır. Segmente edilen parçalar aday olarak alınır, aday imgenin entropisi hesaplanarak doğrulama yapılır ve eşik değeri kullanılarak yüz adayları doğrulanır (Huang vd., 2010). Qiang-rong ve Hua-lan (2010) yüz seziminden önce beyaz denge düzeltmesi (white balance correction) uygulamışlardır. Segmentasyon için renk değeri önemlidir ve böylece imgeyi elde ederken renkler hatalı rengi yansıtabilir. Bunun üstesinden gelmek için, beyaz denge düzeltmesi ilk adım olarak yapılmalıdır. Sonra, ten rengi benzeri alanlar YCbCr içindeki eliptik model kullanılarak segmente edilir. Ten alanlarının bulunmasından sonra, bu alanlar gri tonlamalı imgeler için kenarlandırılmış imgelerle birleştirilir. Son olarak, birleştirilmiş alanlar sınırlayıcı kutu oranının ve sınırlayıcı kutu içindeki alanın kontrol edilmesiyle yüz olarak doğrulanır. Diğer bir uygulama ten benzeri alanları Cb, Cr, Normalize r ve Normalize g içerisindeki eşik değeri ile parçalara ayırmıştır. Daha sonra yüz olabilecek adaylar, sınırlandırıcı kutu oranına, sınırlandırıcı kutu alanı ve içindeki alan oranına ve bölgenin minimum alanına göre seçilmiştir. Son olarak, adayların seçiminden sonra yüz adaylarını bulmak için AdaBoosting yöntemi uygulanmıştır (Li vd., 2010). Ayrıca, ten rengi YCbCr renk uzayı içindeki Cb ve Cr kanalları içinde eliptik bölge içerisinde modellenilebilir. Ten benzeri alan eğer renk değeri eliptik bölgenin içerisindeyse parçalara ayrılır ve aday alanlar şablon eşleme kullanılarak doğrulanır (Aiping vd., 2010). Peer vd. (2003) yüz sezimini YCbCr renk uzayı içinde sadece ten segmentasyonu kullanarak yapmış ve araştırmacılar ten rengi şartlarını RGB renk uzayı içinde de oluşturmayı başarmışlardır. Ten rengi modellemesi için diğer bir yaklaşım Öz Organizasyon Haritası (SOM) Sinir Ağları (NN) ile yapılmıştır. Ten segmentasyonunun

uygulanmasından sonra, her bir segment aday olarak alınmış ve bu segmentlerin eliptik bölge içine uygun olup olamayacağı doğrulanmıştır (Kun vd., 2006).

Yüz sezimi hakkında diğer bir önemli bilgi insan yüzünün şablonudur. Şablon eşleme pencere tarama tekniği ya da parçalara ayrılmış alanlar üzerinden uygulanabilir. Tarama tekniği 20x20 ya da 30x30 piksel gibi küçük boyutlardaki pencerelerle birlikte uygulanır. Bu yaklaşım orijinal imgenin tümünü tarar ve tekrar taramaya uygun hale getirecek bazı iterasyonlarla imgenin boyutunu düşürür. İmge boyutunu düşürmek büyük ve orta boyutlu yüzlerin yerini belirlemede önemlidir. Ancak, bu yüzlerin yerini belirlemede aşırı işlem zamanı gerektirir. Parçalara ayrılmış alanlardaki şablon eşleştirme taramaya göre daha az işlem zamanı gerektirir, çünkü bu yöntem sadece parçalara ayrılmış kısımların eşlemesini dikkate alır. Literatürde şablon eşleştirme kullanılarak yapılmış birçok uygulama vardır (Chen vd., 2009; Guo vd., 2010). Chen vd. (2009), tam-yüz şablonu yerine yarım-yüz şablonu kullanmışlardır. Bu yöntem işlem zamanını düşürmektedir ve bu yarım-yüz yüz yönelimlerine uyarlanabilmektedir. Diğer bir yaklaşım, imge benzeri olmayan fakat imgenin bazı parametrelerden (örneğin boyut, şekil, renk ve pozisyon) oluşan soyut şablonlar kullanmıştır. Ten benzeri alanlar YCbCr renk uzayına bağlı olarak parçalara ayrılmıştır. Sonra, göz ve göz kısımları soyut şablonları parçalara ayrılmış kısımlara uygulanmıştır. Birinci şablonlar gözlerin bölgesini, ikinci şablonlar her bir gözün yerini belirlemektedir. İkinci şablon ayrıca gözlerin yönelimini belirlemektedir. Daha sonra Doku şablonu yüz aday bölgesini doğrulamak için uygulanır (Guo vd., 2010).

İmge Tabanlı yöntemler yüz ve yüz olmayan imgeler arasında karşılaştırma yapmak için eğitim/öğrenme yöntemlerini kullanmaktadırlar. Bu yöntemler için, geniş sayıda yüz ve yüz olmayan imgeler sistemin doğruluğunu arttırmak amacıyla eğitilmelidir. Adaboost, EigenFace, Yapay Sinir Ağları ve Destek Vektör Makineleri yüz sezme algoritmalarında çoğunlukla kullanılan sınıflandırıcı türlerindedir. Yüz ve yüz olmayan imgeler AdaBoost yönteminde dalgacık özelliği cinsinden tanımlanır. EigenFace yönteminde yüz ve yüz olmayan imgelerin öznitelik vektörünü üretmek için TBA kullanılır. Ayrıca, TBA verilen bilgi vektörünü sıkıştırmak için kullanılır. (DVM)'de yüz ve yüz olmayan imgeleri tanımlamak için çekirdek fonksiyonu oluşturulur. Yüz ve yüz olmayan imgeler ayrıca Yapay Sinir Ağlarındaki yapay sinir yapıları tarafından sınıflandırılır.

AdaBoost zayıf sınıflandırıcılardan güçlü sınıflandırıcılar oluşturan bir algoritmadır. Yüz adayları AdaBoost algoritması uygulanarak bulunur. Daha sonra, doğrulama Kademeli Sınıflandırıcılar kullanılarak yapılır. Bu algoritma yüzün, sol, sol+45, ön, sağ+45 ve sağ pozlarını kullanabilir (Bayhan ve Gökmen, 2008).

TBA yukarıda da bahsedildiği gibi EigenFace yöntemindeki yüz ve yüz olmayan imgelerin öznitelik vektörünü üretmek için kullanılır. Pencere tarama tekniği ve segment bazlı analiz bu amaçla kullanılabilir. Eigenface uygulamaları literatürde mevcuttur (Wang ve Yang, 2008; Tsai vd., 2006; Mohan ve Sudha, 2009). Wang ve Yang (2008), yüz adaylarını bulmak için şablon eşleştirmesi uygulamaktadır. Daha sonra, 2D TBA öznitelik vektörünü çıkarmak için kullanılır. İmge matrisi vektör yerine doğrudan 2D TBA'ya verilir. Bu durum kovaryans matrisi hesaplamak için gereken hesaplama süresini azaltır. TBA uygulandıktan sonra Minimum Uzaklık Sınıflandırıcı (Minimal Distance Classifier), yüz ve yüz olmayan durumlar için TBA verilerinin sınıflandırılmasında kullanılır. Bir diğer algoritma, TBA ve Yapay Sinir Ağlarını kullanır. İlk olarak verilen imgeye yüz adaylarını çıkarmak için TBA uygulanır. Sonra aday imgeler, yüz olmayan imgeleri elemek için Yapay Sinir Ağları ile sınıflandırılır. Son olarak, yüz adayları yüz bölgelerindeki kenarların geometrik dağılımı ile doğrulanır (Tsai vd., 2006). Bununla beraber, TBA ve AdaBoost uygulamaları pencere tarama tekniği ile birlikte uygulanır. İlk olarak TBA uygulanır, elde edilen bu öznitelik vektörü AdaBoost yönteminin girdisi olarak kullanılır. AdaBoost yöntemi için güçlü sınıflandırıcılar TBA öznitelik vektörlerinden üretilir (Mohan ve Sudha, 2009).

Yüz ve yüz olmayan imgeler yüz sezmede iki sınıf olmasına rağmen Sinir Ağları örüntü tanıma ve sınıflandırma probleminde iyidir. Yine pencere tarama tekniği veya segment-bazlı uygulama imgeleri sınıflandırmada kullanılabilir. Sinir Ağlarının sezme probleminde kullanılan birçok türü vardır; Algılayıcı (Perceptron), Geri Yayılım Sinir Ağları (Backpropagation NN), Olasılıksal Sinir Ağları (Probabilistic NN), Radyal Temelli Sinir Ağları (Radial Basis NN), Kendi Kendini Organize Eden Rekabetçi Sinir Ağları (Self Organizing Competitive NN), örnek olarak verilebilir (Anijantis vd., 1999; Anagnostopoulos vd., 2002; Liu vd., 2010; Ye-Zhengchun ve Lin-Hongji, 2007; Tayyab ve Zafar, 2009; Aziz ve Ramlee, 2009). Anijantis vd. (1999) Sinir Ağlarını sınıflandırmak

için imge ön işlemlerini uygulamışlar ve daha sonra iki tabanlı algılayıcı kullanarak bu girdinin yüz veya yüz olmayan girdi olup olmadığına karar vermişlerdir. Pencere tarama tekniği yüzleri sezme için kullanılır. Anagnostopoulos vd. (2002) ten benzeri bölgeleri RGB renk uzayında Bulanık Mantık (Fuzzy Logic) ile parçalara ayırarak ve parçalara ayrılan bölgelerin yüz bölgesini doğrulamak için Olasılıksal Sinir Ağlarını (Probabilistic NN) kullanmışlardır. Diğer bir yaklaşım, hesapsal zamanı azaltmak için ten benzeri bölgeleri YUV renk uzayında pencere tarama tekniğine dayalı olarak çıkartmaktadır. Daha sonra, çıkartılan bölgeler yüz adaylarının oluşturulması için Geri Yayılım Sinir Ağlarına (BPNN) gönderilir. Son olarak yüz adayları Bayes Kararı ile doğrulanır (Liu vd., 2010). Ten benzeri bölgelerin parçalara ayrılması YCbCR ile uygulanır. Daha sonra öznelik vektörü parçalara ayrılmış bölgede değişmez dalgacık momenti (wavelet invariant moment) tarafından oluşturulur. Bu aşamadan sonra öznelik vektörü Kendi Kendini Organize Eden Rekabetçi Sinir Ağları (Self Organizing Competitive NN) ile sınıflandırılır. Bu algoritma ön ve profil yüzleri 20 fps video imgesine kadar algılayabilir (Ye-Zhengchun ve Lin-Hongji, 2007). Bunun yanında HSV renk uzayı ten benzeri bölgeleri parçalara ayırmak için kullanılabilir ve parçalara ayrılmış bölgenin öznelik vektörü iki boyutlu ayırık kosinüs dönüşümü ile hesaplanır. Son aşamada ise, ten benzeri bölgelere ait öznelik vektörlerini sınıflandırmak için BPNN kullanılır (Tayyab ve Zafar, 2009). Bir diğer yandan sadece Radyal Tabanlı Sinir Ağları, tarama teknikleri ile birlikte yüzü sezme için kullanılabilir (Aziz ve Ramlee, 2009).

İmge tabanlı yöntemlerde sık kullanılan sınıflandırıcılardan biri de Destek Vektör Makineleridir. DVM yüz ve yüz olmayan imgeleri sınıflandıran çekirdek fonksiyonunu yapılandırmak için yüz ve yüz olmayan imgelerle eğitilir. Bazı DVM uygulamaları (Wang vd., 2002; Shavers vd., 2006; Jee vd., 2004; Jin vd., 2004) yayımlanmıştır. Farklı bir yaklaşım yüz adaylarını bulmak için uygulanmıştır (Wang vd., 2002). Yüz adayı gözlerin konumuna bağlı olan genelleştirilmiş simetri ile bulunur. Son olarak, yüz adayı DVM ile doğrulanır/sınıflandırılır. Bunun yanında DVM tarama tekniği ile birlikte yüzleri sezme için kullanılabilir (Shavers vd., 2006). Jee vd. (2004) YCbCr tabanlı olarak ten benzeri bölge segmentasyonu uygulamışlar ve göz adayı bölge içindeki kenar bilgisi ile bulunmuştur. Sonra gözler DVM kullanılarak doğrulanmıştır. Doğrulama sonrasında, yüz adayı gözün pozisyonuna göre çıkartılmıştır. Son bir doğrulama olarak, yüz adayı DVM'ye gönderilmiştir. DVM için bir diğer yaklaşım tek sınıfa dayalı DVM kullanılmasıdır. Yüz

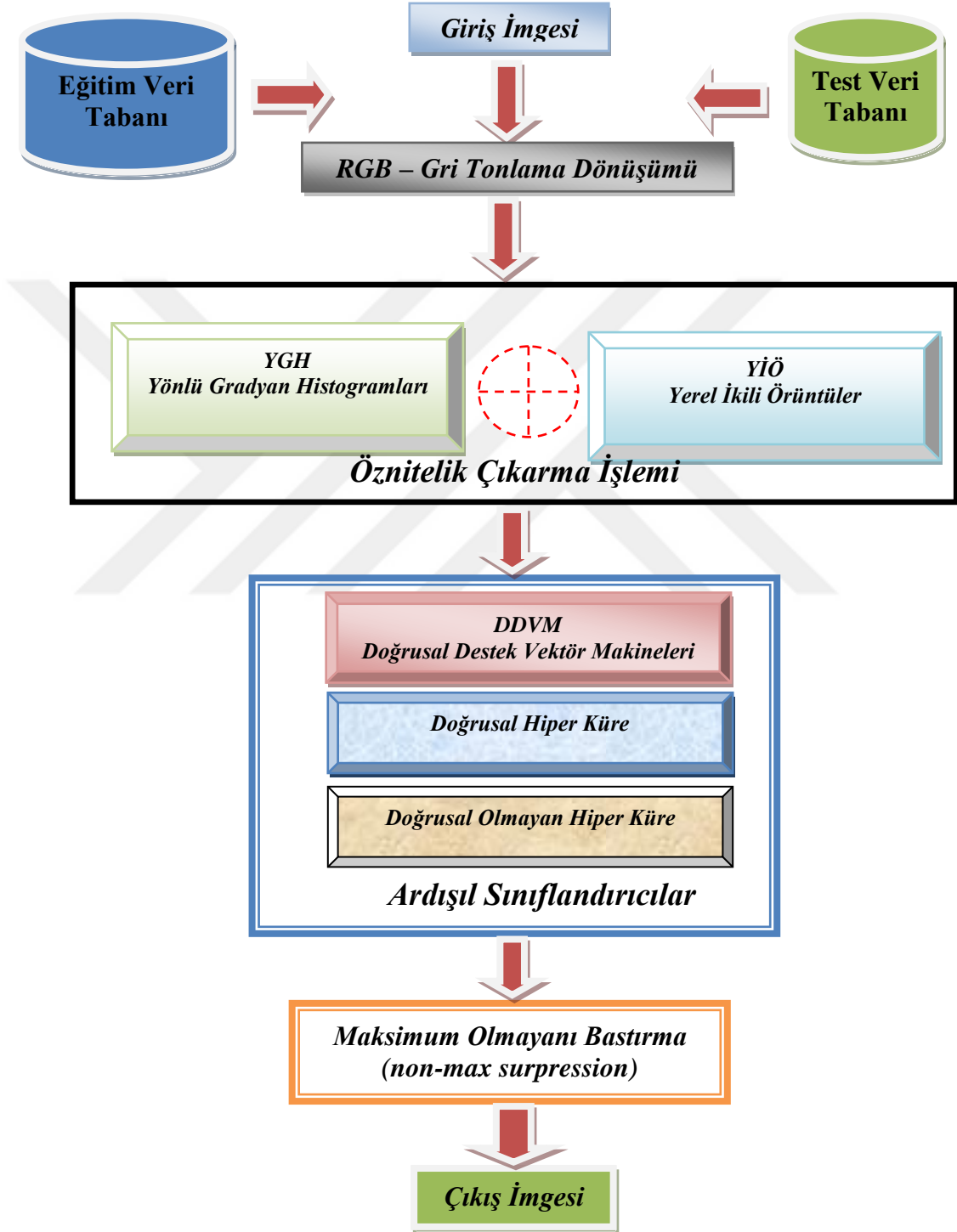
ve yüz olmayan olarak iki sınıf oluşturmak yerine yüz olmayan imgelerin modellenmesinin zor olması sebebiyle sadece tek bir yüz sınıfı oluşturulur (Jin vd., 2004).

Bu yöntemlerin yanında yüz sezimi için derin öğrenme tabanlı yöntemlerde kullanılmaya başlanmıştır (Redmond vd., 2016), (Liu vd., 2016), (Ren vd., 2015). Bu yöntemlerde genel olarak pozisyonları bulunacak tüm nesnelerin konumları (yüzler, arabalar, yayalar, hayvanlar vb.) işaretlenerek sistemler eğitilir ve test esnasında da birden çok nesnenin konumu eş zamanlı olarak hızlı bir şekilde bulunabilmektedirler. Bu sistemlerin tek dezavantajı yüksek hesap karmaşası içermeleri sebebiyle GPU kullanılması zorunluluğudur.



3. MATERYAL VE YÖNTEM

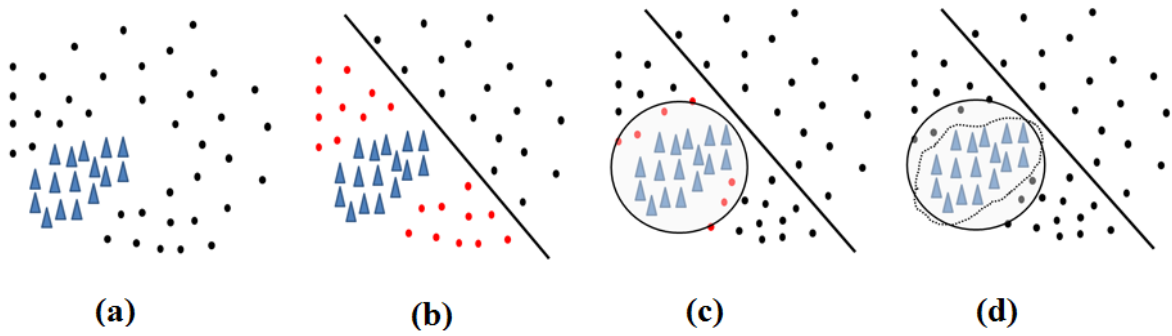
Çalışmada kullanılan yöntemle ait şematik gösterim Şekil 3.1’de gösterilmektedir.



Şekil 3. 1. Çalışmada kullanılan yöntemle ait şematik gösterim

Günlük hayattan kareleri yansıtan dijital imgelerde arka plan oldukça karmaşıktır ve büyük farklılıklar gösterebilir. Sınıflandırıcıların eğitilmesi sırasında arka planı oluşturan örnek sayısı genellikle yüze ait örnek sayısından çok daha fazladır ve bu sebeple sınıflandırma problemi sınıf örnek sayıları bakımından oldukça dengesizdir.

Biz bu çalışmada yüz sezimi için 3 katmandan oluşan ardışıl bir sınıflandırıcı sistemi geliştirdik. Ardışıl sınıflandırıcısının ilk katmanında hızlı bir sınıflandırıcı olan doğrusal destek vektör makineleri kullanıldı. Bu katmanın amacı yüze ait hemen hemen tüm örnekleri geçirerek, arka plana ait örneklerin birçoğunu elemektir. İkinci aşamasında ise yüz sınıfını doğrusal bir hiper küre ile modelleyen sınıflandırıcı kullanılmıştır. Bu sınıflandırıcı da oldukça hızlı olup destek vektör makineleri ile uyumlu bir şekilde çalışarak ilk katmanda geçen birçok arka plan görüntüsünü Şekil 3.2’de görüldüğü gibi elemektedir. Son katmanda ise doğrusal olmayan hiper küre kullanılmıştır. Bu sınıflandırıcı hız bakımından diğer iki katmandaki sınıflandırıcılara oranla çok daha yavaştır. Fakat her iki katmandaki sınıflandırıcıyı da geçen az sayıda örneğe uygulanmaktadır.



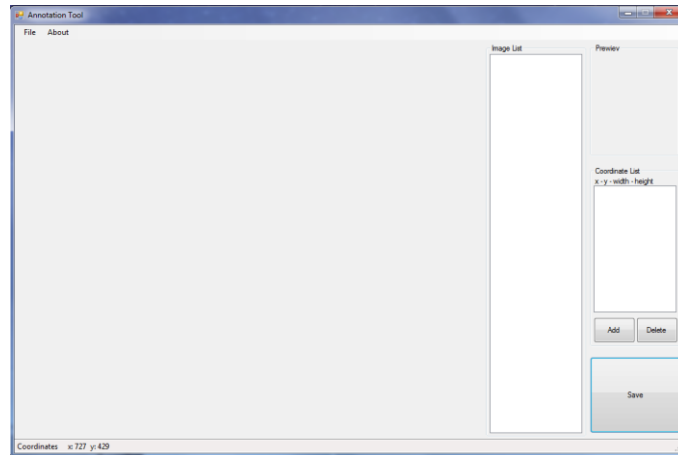
Şekil 3. 2. Çalışma kapsamında kullanılan ardışıl sınıflandırıcı: **(a)** Yüz sınıfına ve arka plana ait temsili örnekler. Yüz sınıfına ait örnekler mavi renkte üçgenlerle arka plan ise siyah renkli yuvarlaklarla gösterilmiştir. **(b)** Birinci katmanda uygulanan Doğrusal Destek Vektör Makinelerinin çıktısı. Yanlış olarak yüz sınıfına atanan arka plan görüntüleri (false positives) kırmızı renk ile gösterilmiştir. **(c)** İkinci katmandaki Doğrusal Hiper-Küre sınıflandırıcısı birinci katmanı geçen birçok arka plan örneğini elemiştir. **(d)** Son katmanda kullanılan Doğrusal Olmayan Hiper-Küre sınıflandırıcısı yüz sınıfını çok doğru bir şekilde modeller ve her iki katmandan geçen yanlış örnekleri eleyerek en son kararları verir.

Tez çalışması kapsamında yüz sezme sistemini eğitmek için internet üzerinden toplanan günlük hayattan çekilmiş imgeler kullanılmış, bu imgelerde bulunan sağ ve sol profil yüzlerine ait öznitelikler YİÖ ve YGH betimleme teknikleri yardımıyla ayrı ayrı çıkarılarak öznitelik vektörleri elde edilmiş, elde edilen bu öznitelikler vektörleri birleştirilerek tek bir öznitelik vektörü (YİÖ+YGH) oluşturulmuştur.

3.1 Veri Tabanı Oluşturulması

3.1.1 Eğitim veri tabanı

Tez çalışması kapsamında sistemin eğitilmesi amacıyla veri tabanının oluşturulması için web ortamından profil resmi içeren günlük hayattan çekilmiş resimler toplanmıştır. Toplanan bu resimlerden yüz kısımlarının kesilmesi için “Annotation Tool” programı kullanılmıştır. Programa ait arayüz Şekil 3.3’te gösterilmektedir. Bu programda profil yüz resmi içeren resimler programa yüklenebilmekte ve yüz kısımları kolay bir şekilde seçilip kesilerek “Bitmap, GIF, JPEG, PNG, TIFF” dosya formatında saklanabilmektedir. Tez çalışması kapsamında veri tabanındaki resimlerin saklanmasında “.PNG” formatı kullanılmıştır. Program aynı zamanda resim üzerinde yüz olarak seçilen kısmın koordinatlarını da kaydedebilmektedir. Şekil 3.4’te veri tabanında kullanılan imgelerin Annotation Tool programı yardımıyla nasıl oluşturulduğu gösterilmektedir.



Şekil 3. 3. Annotation tool programı arayüzü



Web ortamından toplanan profil yüz imgesi içeren resim



Annotation Tool Programına kesme işlemi yapılacak resmin yüklenmiş hali



Annotation Tool Programı yardımıyla yüz imgesinin seçilmesi

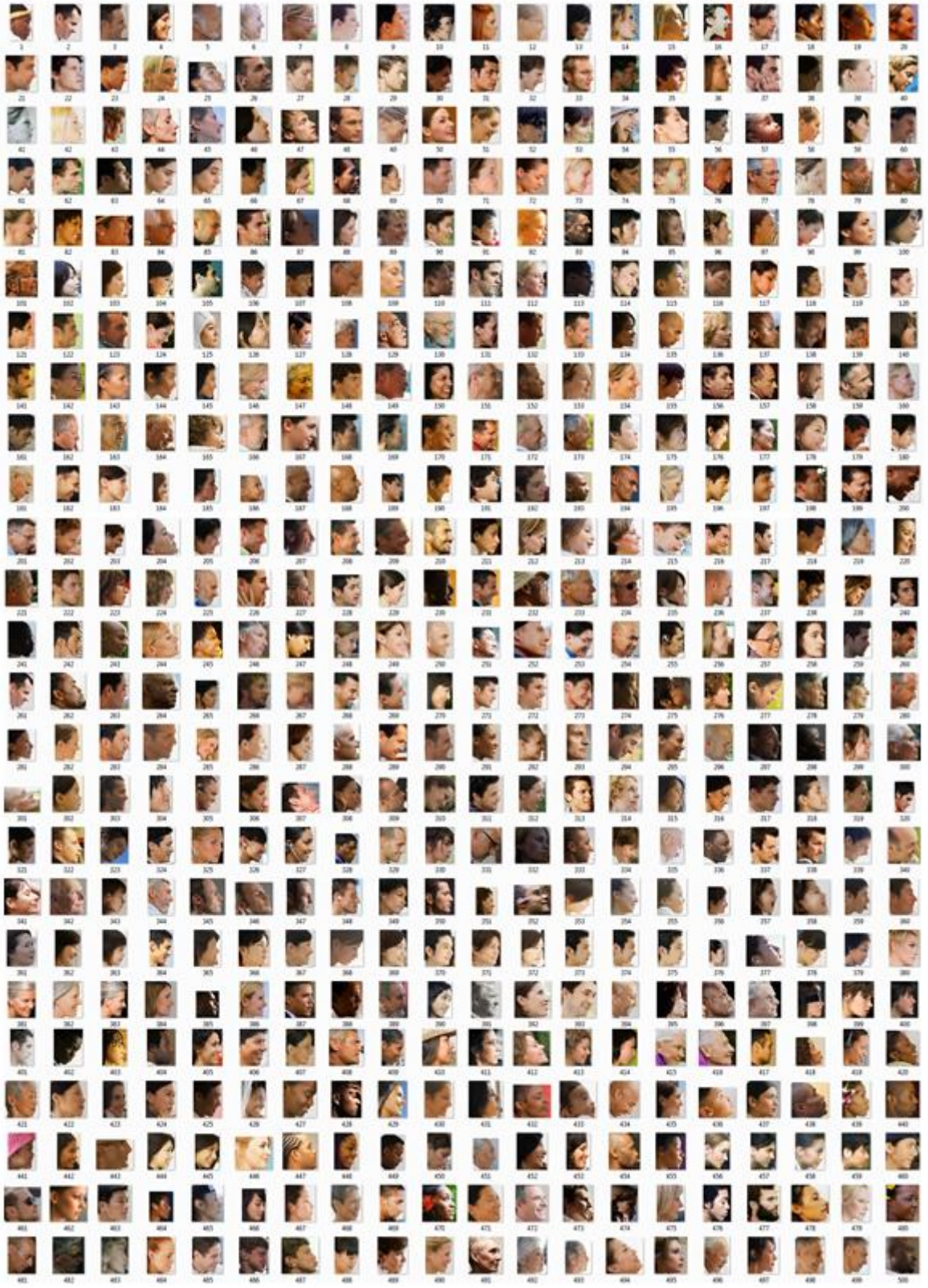


Çalışma kapsamında kullanılan veri tabanında kullanılan imgenin son hali

Şekil 3. 4. Veri tabanında kullanılan imgelerin oluşturulması

Çalışma kapsamında web ortamında toplanan resimler Annotation Tool programı yardımıyla Şekil 3.4'te gösterildiği şekilde kesilip saklanarak profil yüz sınıfına ait eğitim veri tabanı oluşturulmuştur. Veri tabanındaki profil yüz imgesi içeren son örnek sayısı 21180 adet olup bu resimler eğitim kısmında “Pozitif İmgeler” olarak adlandırılmıştır. Şekil 3.5'te sisteme ait veri tabanındaki pozitif imgelerden örnekler yer almaktadır. Sağ ve sol profil için veri tabanları aynı örneklerden oluşmaktadır. Web ortamında toplanan ve program yardımıyla kesilip saklanan imgelerin ters simetrisi alınarak (mirror) veri tabanının aynı örneklerden oluşması sağlanmıştır. Çalışma kapsamında eğitim veri tabanında kullanılan arka-plan örnek sayısı 170.000 adet olup bu resimler eğitim kısmında “Negatif İmgeler” olarak adlandırılmıştır.

Eğitim sırasında veri tabanındaki ilk örneklerin tamamı 35x28 boyutlarına sığacak şekilde örneklenmişler ve bu örnekler kullanılarak Yerel İkili Örüntü (YİÖ) ve Yönlü Gradyan Histogramları (YGH) öznitelikleri elde edilerek imge betimleme tamamlanmıştır. İmge betimlemenin ardından profil yüz sezme sistemindeki ardışıl sınıflandırıcılar eğitilmiş ve sistem binlerce yeni imge üzerinde test edilerek sistemin konumlarını doğru olarak döndüremediği yüz resimleri (hard positives) ve sistemin yanlışlıkla yüz olarak döndürdüğü arka-plan örnekleri (hard negatives-false positives) toplanmıştır. Sistem daha sonra bu yeni zor örnekler de eklenerek tekrar eğitilmiş ve yeni imgeler üzerinde test edilmiştir. Bu işlem birkaç kez tekrarlanarak profil yüz sezme sistemine son şekli verilmiş olup yukarıda eğitim veri tabanı için verilen 21180 pozitif ve 170.000 negatif örnek sayısına ulaşılmıştır.

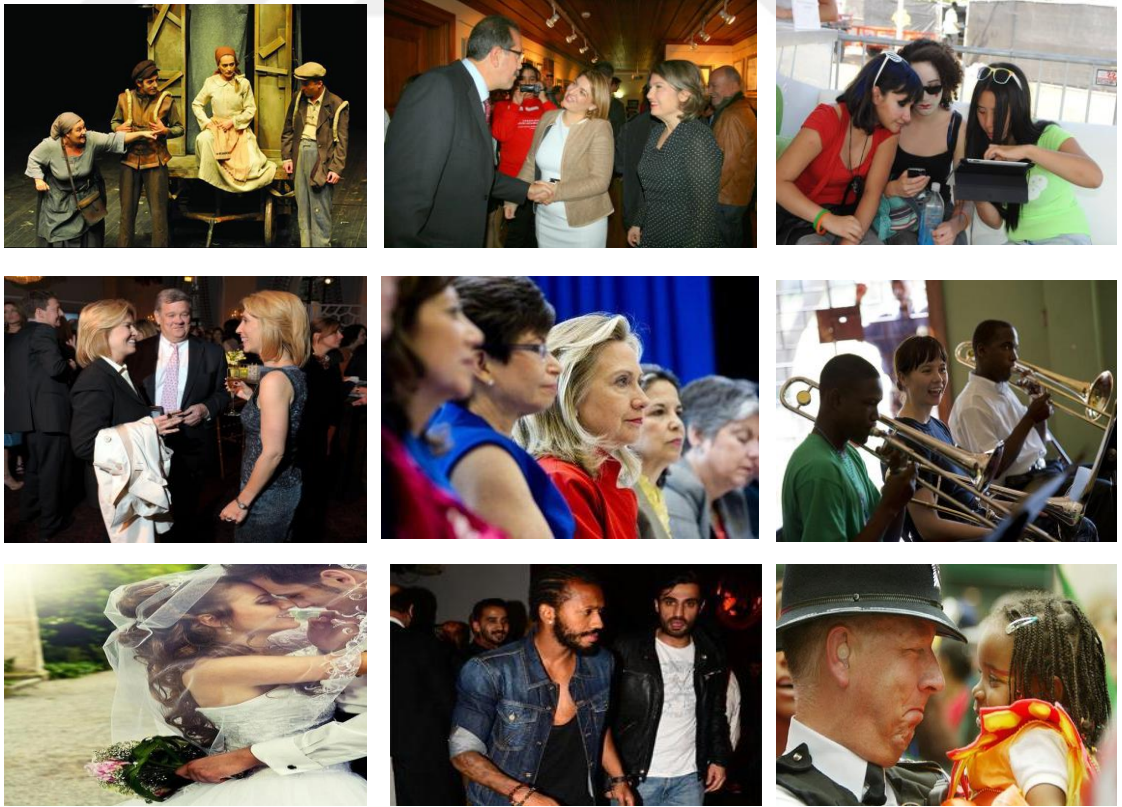


Şekil 3. 5. Eğitim veri tabanındaki pozitif imgelerden örnekler

3.1.2 Test veri tabanı

Test veri tabanı için eğitim veri tabanında olmayan yine web ortamından profil resmi içeren günlük hayattan çekilmiş zor resimler toplanmıştır. Toplanan bu resimlerin profil yüz kısımları “Annotation Tool” programı yardımıyla kesilerek saklanmıştır. Ancak bu defa eğitim veri tabanından farklı olarak program yardımıyla kesilen resimlerin koordinatları da saklanmıştır. Kesilen imgelerin bu koordinatları “GroundTruth” doğrulama test kısmında sistemin başarı oranını test etmek amacıyla kullanılmıştır.

Test veri tabanının oluşturulması için web ortamından toplanan 239 adet profil resmi içeren resim kullanılmıştır. Bu resimlerin bazılarının birden fazla profil yüz resmi içermesi nedeniyle profil yüz imgelerinin tamamının koordinatlarının saklanması sonucu 527 adet farklı profil yüz resmi elde edilmiştir. Şekil 3.6’da test veri tabanı için web ortamından toplanan resimlerden örnekler ve Şekil 3.7’de test veri tabanındaki pozitif imgelerden örnekler ile saklanan koordinatları yer almaktadır.



Şekil 3. 6. Test veri tabanı için web ortamından toplanan resimlerden örnekler

| | | | | | |
|----------|---|-----|-----|-----|-----|
| 001. jpg | 2 | 78 | 149 | 33 | 37 |
| | | 176 | 89 | 26 | 31 |
| 009. jpg | 2 | 228 | 55 | 131 | 182 |
| | | 586 | 207 | 24 | 31 |
| 042. jpg | 1 | 269 | 165 | 127 | 184 |
| 119. jpg | 5 | 18 | 10 | 86 | 116 |
| | | 95 | 21 | 85 | 118 |
| | | 194 | 60 | 76 | 111 |
| | | 254 | 102 | 50 | 87 |
| | | 313 | 115 | 63 | 98 |
| 142. jpg | 2 | 120 | 16 | 53 | 67 |
| | | 66 | 38 | 27 | 34 |
| 145. jpg | 4 | 256 | 109 | 125 | 147 |
| | | 77 | 57 | 94 | 106 |
| | | 874 | 136 | 85 | 98 |
| | | 975 | 156 | 47 | 64 |
| 199. jpg | 1 | 75 | 58 | 155 | 155 |
| 206. jpg | 1 | 247 | 59 | 160 | 170 |
| 225. jpg | 2 | 158 | 224 | 151 | 236 |
| | | 603 | 120 | 120 | 159 |



Şekil 3. 7. Test veri tabanındaki pozitif imgelerden örnekler ve saklanan koordinatları

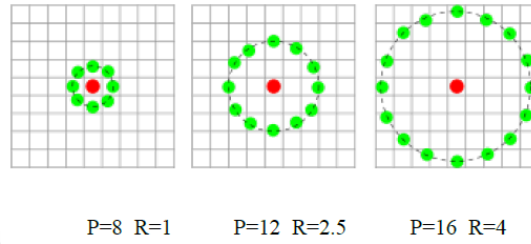
3.2 İmge Betimleme (Öznitelik Çıkarımı)

İmge betimleme amacıyla Yönlü Gradyan Histogramları ile beraber Yerel İkili Örüntü öznitelikleri kullanılmıştır. YGH öznitelikleri nesnelere genel görünüşlerini betimlerken, YİÖ ise yüzlerin doku (texture) bilgilerini betimlediğinden bu iki öznitelik birbirlerini tamamlar niteliktedir. YGH ve YİÖ öznitelikleri birlikte kullanılırken, bu öznitelikler normalizasyon uygulanarak arka-arkaya bağlanmıştır.

3.2.1 Yerel İkili Örüntüler (YİÖ)

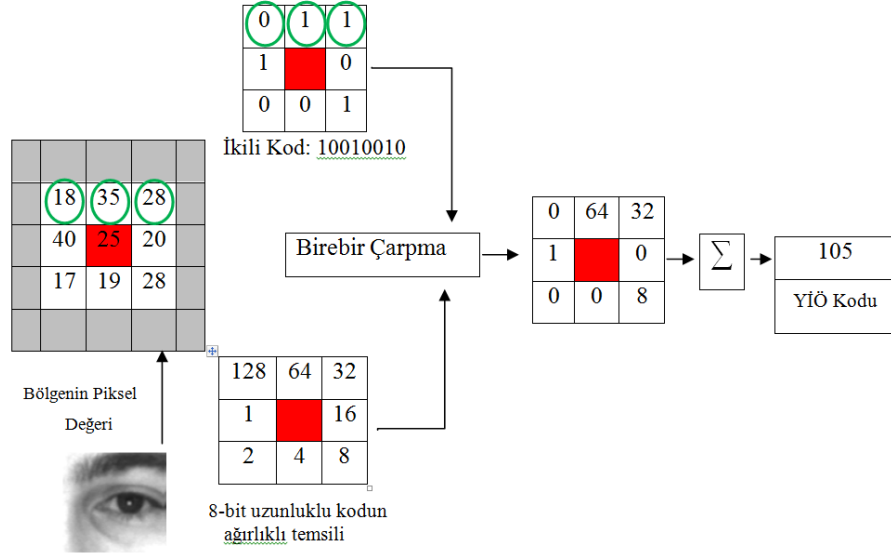
Yerel İkili Örüntüler ilk olarak 1990 yılında doku spektrumu modeli olarak tanıtılmıştır (He, D.C. ve Wang, L., 1990). Ojala vd. (1996) ile öznitelik çıkarımında güçlü bir yaklaşım haline gelmiş Ahonen vd., (2006) yerel komşuluk değerlerine dayalı bir özellik çıkarma yöntemi olarak yüz tanıma uygulamalarında kullanmışlardır. Wang vd. (2009) ise bu yöntemi Yönlü Gradyan Histogramları (YGH) ile birleştirerek farklı bir yaklaşım kullanarak yüksek performans elde etmişlerdir.

YİÖ operatörü ile özellik vektörünün hesaplanması hızlı ve kolaydır. Gri seviye üzerinde uygulanan bu yöntemin ışık değişimlerine karşı duyarlılığı az olduğu için tanıma yöntemlerinde sıkça tercih edilir. YİÖ operatörü endüstriyel kontrol, 3 boyutlu doku yüzeylerinin sınıflandırılması, yüz tanıma, yüz bulma, yaşın belirlenmesi, yüz ifadesi tanıma, arka plan modelleme, hareketli nesnelerin tespiti gibi çeşitli bilgisayar uygulamalarında kullanılmış ve iyi performans göstermiştir (Sarode ve Patil, 2013).

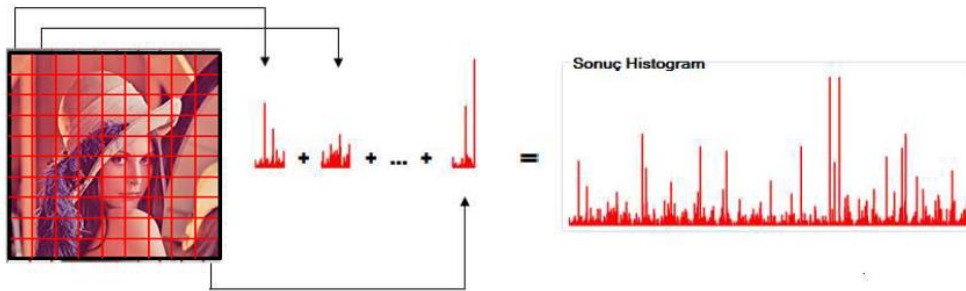


Şekil 3. 8: Çeşitli dairesel YİÖ P,R operatörleri

Uygulamalarda genelde komşuluk değerlerine göre farklı operatörler kullanılır. Bu operatörler P merkez piksel etrafında alınacak komşu sayısını, R örnekleme yarıçapını temsil edecek şekilde YİÖP, R gösterimiyle ifade edilir. Şekil 3.8'de çeşitli YİÖ operatörleri gösterilmektedir. Çalışmada yarıçapı 1 olan 8 nokta komşuluklu YİÖ8,1 operatörü kullanılmıştır. YİÖ operatörü 3x3 boyutlu bir çekirdektir ve görüntüdeki her pikseli, merkez piksel kabul ederek etiketleme işlemi gerçekleştirir. Merkez değer eşik değer olarak kabul edilir ve etrafındaki sekiz adet komşu parlaklık değeri, bu eşik değeri ile kıyaslanır. Orta noktadaki referans pikselinin çevresindeki gri-seviye değeri bu noktanın gri-seviye değerinden büyük veya bu noktanın değerine eşitse 1, bu noktanın değerinden küçükse 0 değeri verilir. Böylelikle piksel gri-seviyeleri farkından oluşan ikili bir kod elde edilir. Daha sonra bu kod onluk sayı sistemine çevrilir. YİÖ operatörünün elde edilmesi ve bu bitlerin ağırlık katsayıları Şekil 3.9'da gösterilmiştir.



YİÖ ile etiketlenen görüntü üzerindeki verilerin dağılımlarını belirlemek için histogram kullanılır. Görüntü üzerinde genel bir histogram oluşturmaktansa yerel dağılımlara göre yerel histogram oluşturulması için imge belirli boyutlarda (örneğin 10x10) bloklara ayrılır ve daha sonra her bloğun histogramı çıkarılır. Bu amaçla bu çalışmada imge pencereleri birbirleriyle örtüşmeyen $m \times n$ alt bölgeye bölünmüş ve her bir bölge için 59 boyutlu YİÖ histogramları elde edilmiştir (59 binden oluşan histogram kodlamasında 58 bin birörnek (uniform) değerler için geriye kalan tek bin ise bir örnek olmayan (non-uniform) değerler için kullanılmıştır). Daha sonra bu histogramlar L1 normu kullanılarak normalize edilip arka-arkaya bağlanmıştır. Bu sayede her bir imge penceresi $d = m \times n \times 59$ boyutlu öznelik vektörü ile betimlenmiştir. Şekil 3.10'da YİÖ Histogramının elde edilmesi gösterilmiştir.

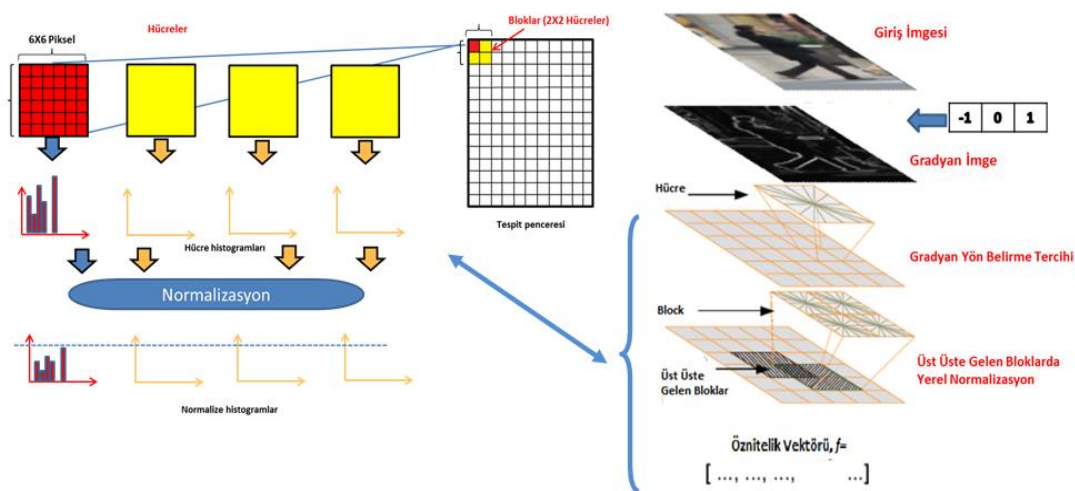


3.2.2 Yönlü Gradyan Histogramları (YGH)

Yönlü Gradyan Histogramları (YGH) imgedeki tüm lokasyonlardan yoğun bir özellik çıkarmak için kullanılan popüler bir tekniktir. Dalal ve Triggs, (2005) geliştirdikleri nesne konum algoritmasında, nesnelerin özniteliklerini çıkartılmasında bu tekniği kullanarak algoritmalarında yüksek başarımları elde etmiştir.

YGH yerel imge görüntülerinin, bu bölgelere ait kenar yönlerinin ya da gri-seviye gradyanlarının yerel dağılımları kullanılarak etkili bir şekilde kodlanabildiği varsayımı üzerine kuruludur. Çalışmalarımızda (Dalal ve Triggs, 2005) verilen temel yöntemin üzerine (Felzenszwalb vd., 2010) yapılan değişiklikler kullanılarak elde edilen YGH öznitelikleri kullanılmıştır.

YGH öznitelikleri bulunurken izlenen temel yöntem Şekil 3.11'de gösterilmiştir. İlk olarak imgeye ait gradyanlar hesaplanmaktadır. Daha sonra imge hücre (cell) adı verilen ve birbiriyle örtüşmeyen dikdörtgenel bölgelere bölünmüş ve her bir hücreye ait gradyan yönlerinin histogramları hesaplanmıştır. Son olarak belli sayıda hücreler bir araya getirilerek blok adı verilen yapılar oluşturulmuş ve her bir bloktaki hücreye ait histogramlar yerel olarak normalize edilmiştir. Daha sonra bu histogramlar arka-arkaya bağlanmak suretiyle bir imge penceresine ait YGH öznitelikleri elde edilmiştir.



Şekil 3. 11. YGH algoritma uygulama şekli

İmge gradyanlarını bulmak için her iki yönde $[-1,0,1]$ filtresi kullanılmıştır. Hücreler deneylerde 6×6 piksel olarak seçilmişlerdir. Bloklar Şekil 3.11’de görüldüğü gibi 2×2 ’lik hücreler kullanılarak oluşturulmuştur. Yön histogramlarının binlerini oluşturmak için $[0^\circ-180^\circ]$ dereceleri arası 9 eşit parçaya bölünmüştür. Bu yol izlenerek her bir hücreden elde edilecek öznitelik sayısı 36 olmaktadır. Bu çalışmada Felzenszwalb tarafından paylaşımına açılan YGH öznitelik çıkarma kodları kullanılmıştır.

3.3 Sınıflandırma

Tez çalışmasında sınıflandırma amacıyla üç farklı ardışıl ikili ve tek sınıflı sınıflandırıcı (Doğrusal Destek Vektör Makineleri + Doğrusal Hiper Küre + Doğrusal Olmayan Hiper Küre) kullanılmıştır. Ardışıl sınıflandırıcısının ilk iki katmanında sırasıyla hızlı sınıflandırıcılar olan Doğrusal Destek Vektör Makineleri ve Doğrusal Hiper Küre, üçüncü ve son katmanda ise her iki katmandaki sınıflandırıcıyı da geçen az sayıda örneğe uygulanan ve ilk iki katmandaki sınıflandırıcılara oranla çok daha yavaş olan Doğrusal Olmayan Hiper Küre kullanılmıştır.

3.3.1 Destek Vektör Makineleri (DVM)

Destek Vektör Makineleri, ilk olarak Cortes ve Vapnik (1995) tarafından istatistiksel öğrenme teorisine bağlı bir eğitilmiş sınıflandırma yöntemi olarak tanıtılmıştır. Destek Vektör Makineleri ses tanıma, görsel nesne tanıma, teks sınıflandırma gibi zorlu örüntü tanıma problemlerinde oldukça sık kullanılan başarılı bir yöntemdir (Burges, 1996).

DVM, dağılımı hakkında herhangi bir bilgi tahmini içermeyen eğitim veri setlerine bağlı bir şekilde öğrenme yaparak ve yeni veriler üzerinde tahminlerde bulunmaya çalışarak genelleştirme yapan bir makine öğrenme algoritmasıdır.

DVM’nin temel çalışma prensibi, iki sınıfa ait veri kümelerini birbirinden en iyi şekilde ayıran karar sınırlarının bir başka ifadeyle “hiper düzlemlerin” belirlenmesidir (Vapnik, 2000). DVM’nin en genel sınıflandırma problemi; “doğrusal” olarak ayrılabilen “iki sınıflı” bir veri setinin sınıflandırılmasıdır. DVM bu problemi çözmek için, sınıflar arasındaki sınırın maksimum olduğu yeri belirleyerek iki sınıf arasındaki en iyi ayırıcı düzlemi tespit etmeye çalışır. Böylelikle en iyi ayırıcı düzlem yardımıyla, her bir sınıfa ait

verilerin, ayırıcı düzleme olan uzaklıklarını maksimum seviyeye getirir. Bu sayede, hiper düzlemlere en yakın öğrenme verileri iki farklı sınıf arasındaki sınırı temsil eden destek vektörlerini oluşturur (Öğüdücü, 2007)

Destek Vektör Makineleri, Doğrusal Destek Vektör Makineleri ve Doğrusal Olmayan Destek Vektör Makineleri şeklinde veri setinin doğrusal ya da doğrusal olmama durumuna göre ikiye ayrılmaktadır. Geliştirdiğimiz ardışıl sınıflandırıcıların ilk basamağında Doğrusal Destek Vektör Makineleri kullanıldığından sadece bu kısım ile ilgili açıklamalara tez çalışması kapsamında yer verilmiştir.

Doğrusal Destek Vektör Makineleri kullanılan verilerin doğrusal ayrılabilme ya da ayrılamama durumuna göre açıklanmıştır.

a) Veri Setinin Doğrusal Ayrılabilme Durumu:

İki sınıflı bir sınıflandırma tekniği olan Destek Vektör Makinelerinde, $(g(x) = \text{sign}(f(x)))$ şeklindeki karar fonksiyonu kullanılarak sınıflandırma yapılması amaçlanır. Bu iki sınıf çoğunlukla “pozitif” (+1) ve “negatif” (-1) sınıf şeklinde gösterilmektedir. Giriş verileri, doğrusal olarak ayrılabilirdiği takdirde; sayısı sonsuza ulaşan ayırıcı düzleme bakarak karar sınırını en büyük yapacak olan ayırıcı düzlemi tespit etmeyi hedefler. Bu sayede, test verilerini sınıflandıracak karar fonksiyonu eğitim sonucu oluşturulan en iyi ayırıcı düzlem yardımıyla belirlenir.

DVM, kendisine en yakın noktalar arasındaki uzaklığı maksimum yapan ayırıcı düzlemin tespit edilmesi için $(x \in \mathcal{R}^N$: temsili öznitelik vektörü, $y_i \in \{-1, +1\}$: temsili sınıf etiketleri olmak üzere); (+1) ve (-1) etiketli verileri birbirinden ayırabilme özelliğine sahip çok sayıda ayırıcı düzlem çizilmesine imkan tanır. Tespit edilen bu ayırıcı düzlem en iyi ayırıcı düzlem olarak adlandırılır ve bu düzleme komşuluğu bulunan, sınır genişliğini sınırlandıran noktalar ise destek vektörleri olarak adlandırılır. Denklem (3.1), bu ayırıcı düzlem üzerindeki herhangi bir x_i noktası için sağlanmaktadır. Burada w ayırıcı düzlemin normal vektörü, b bias parametresini göstermektedir.

$$(w \cdot x_i) + b = 0 \quad (3.1)$$

Elimizde DVM eğitimi için kullanılacak N elemanlı bir veri olduğunu varsayalım;

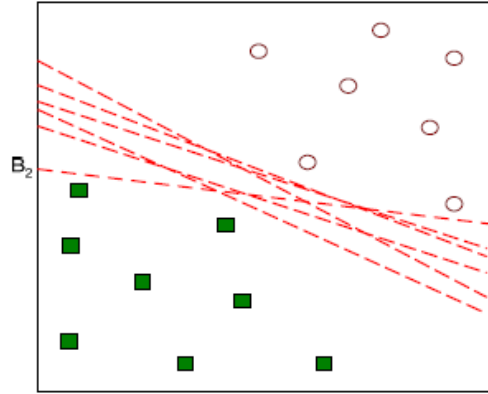
$$E = \{x_i, y_i\}, i = 1, 2, \dots, N$$

Böylece, en iyi karar sınırı eşitsizlikler (3.2) ve (3.3) yardımıyla gösterilebilir.

$$(w \cdot x_i) + b \geq +1, y_i = +1 \text{ için} \quad (3.2)$$

$$(w \cdot x_i) + b \leq -1, y_i = -1 \text{ için} \quad (3.3)$$

Bu sayede, iki sınıflı bir veri setini ikiye ayırabilecek sonsuz sayıda doğru çizilebilir. Şekil 3.12'de bu durum gösterilmektedir.



Şekil 3. 12. İki sınıflı bir veri setini ikiye ayıran örnek doğrular (Öğüdücü, 2007)

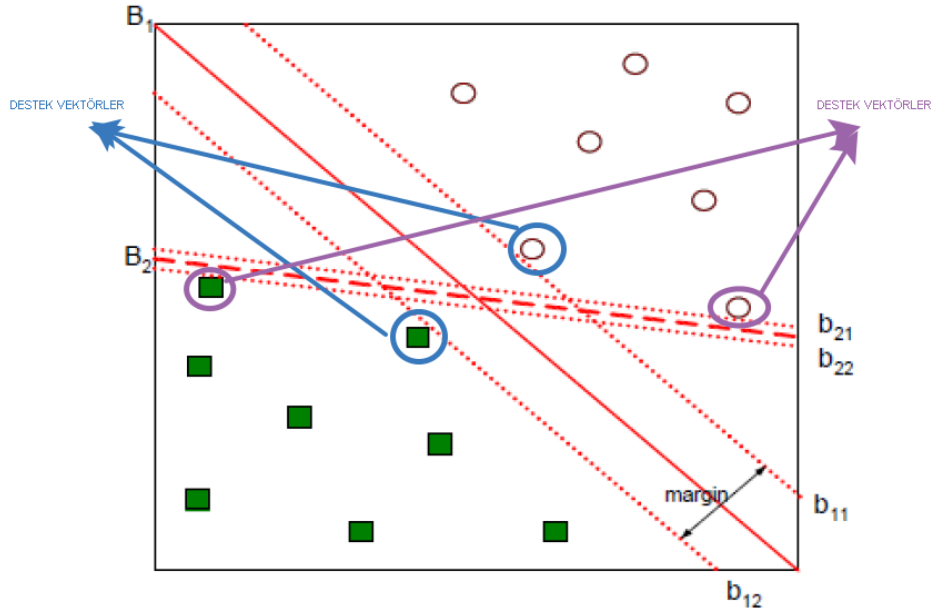
Buradaki amaç, bilinmeyen bir veri kümesi ile karşılaşıldığında sınıflama hatasını en aza indirecek, bir başka deyişle; farklı sınıftan örnekler içerisinde uzaklığı en büyük yapan doğruyu seçmektir. Bunun nedeni, her iki sınıfa ait verilerden de mümkün olduğunca uzakta olan karar sınırının, en iyi ayırıcı olmasıdır. Sınır ne kadar büyük olursa, tahminin eğitim setinde güvenilirliğini artırarak yeni örnekler üzerinde tahmin başarı oranının daha iyi olmasını sağlar.

Şekil 3.13'te hiper düzlemler kesikli çizgilerle gösterilmektedir. Bu hiper düzlemlere en yakın eğitim verileri ise destek vektörleri olarak adlandırılmaktadır. Denklem (3.4) ve (3.5) yardımıyla hiper düzlemler, Denklem (3.6) ile ise sınıflandırıcı karar fonksiyonu ifade edilmektedir. Hiper düzlemler arasındaki uzaklık ise marjini olarak tanımlanmaktadır. Şekil 3.13'te ayrıca marjini en büyük seviyeye getiren düzlem olan en iyi ayırıcı düzlem "B₁" gösterilmektedir.

$$(w \cdot x_i) + b \geq \pm 1 \quad (3.4)$$

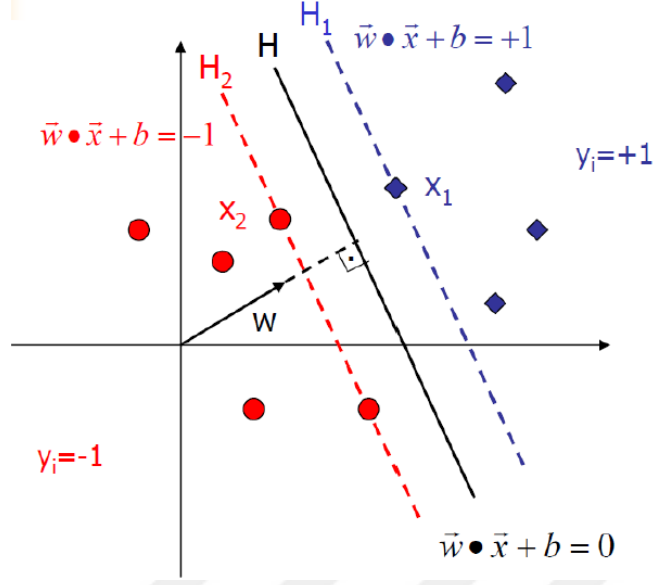
$$y_i((w \cdot x_i) + b) - 1 \geq 0, y_i \in \{-1, +1\}, \quad (3.5)$$

$$f(x) = \text{sign}((w \cdot x_i) + b) \quad (3.6)$$



Şekil 3. 13. Farklı sınıflar arasındaki uzaklığı en iyi ayıran düzlem: B1 (Öğüdücü, 2007)

Şekil 3.14'te ise doğrusal olarak ayrılabilen iki sınıflı veri setlerinin genel gösterimi yer almaktadır. Bu genel gösterimde marjini maksimize eden en iyi ayırıcı hiper düzlem ve diğer ayırıcı hiper düzlemler gösterilmektedir.



Şekil 3. 14. Doğrusal ayrılabilen iki sınıflı veri setlerinin genel gösterimi (Öğüdücü, 2007)

w ve b değerlerinin hesaplanmasıyla en iyi ayırıcı düzlem bulunur. Denklem (3.4) yardımıyla $\frac{-1-b}{\|w\|}$ ve $\frac{1-b}{\|w\|}$ değerleri kullanılarak sınır hiper düzlemlerinin orijine olan uzaklıkları sırasıyla hesaplanır. Hiper düzlemler arası uzaklık ise $\frac{2}{\|w\|}$ olarak gösterilir. En iyi ayırıcı düzlemin sınıra olan uzaklığı, $\|w\|$ ifadesinin mümkün olduğunca minimum hale getirilmesiyle, maksimuma çıkarılmaya çalışılır ve bu sayede en iyi ayırıcı düzlem bulunur. Bu durumda maksimum sınırın bulunması için $\min \left[\frac{1}{2} \|w\|^2 \right]$ ifadesi Denklem (3.5)'te belirtilen koşula uygun olarak hesaplanmalıdır (Vapnik 2000). Sonuç olarak, en iyileme problemi denklem (3.7)'deki gibidir.

$$\min \left[\frac{1}{2} \|w\|^2 \right] \text{ koşulu ile, } y_i((w \cdot x_i) + b) - 1 \geq 0, y_i \in \{-1, +1\}, \quad (3.7)$$

Bu problem, koşul ve eşitlik yardımıyla ifade edilebilen doğrusal olmayan bir kısıtlı en iyileme (constrained optimization) problemidir. Bu problemin çözülebilmesi için Lagrange fonksiyonunun ve Lagrange çarpanlarının ($\alpha_i, i = 1, \dots, N$) kullanılması gerekmektedir. Burada, α_i değerleri Lagrange çarpanlarıdır (Fletcher, 2008).

$$L_p = \frac{1}{2} \|w\|^2 - \sum_{i=1}^N \alpha_i [y_i (w \cdot x_i + b - 1)] \quad (3.8)$$

Denklem (3.8)'de verilen Lagrange denkleminde, w ve b parametreleri kullanılarak denklem en küçüklenir, α_i çarpanlarına göre ise en büyüklenir. Böylece problemin, Denklem (3.9)'daki gibi ifade edilmesi yoluyla en iyileme problemleri farklı biçimlere dönüştürülebilirler.

$$\arg \max_{\alpha} \min_{w,b} \{L_p\} \quad (3.9)$$

Bu işlem ise, Lagrange denkleminin doğru değişkenlerine bakılarak kısmi türevlerinin alınıp çözülmesi ve sonuçların denklemde doğru bir biçimde yerlerine konmasıyla ve elenmesiyle yapılır. Sonuç olarak, en büyülenecek bağıntı sadece Lagrange çarpanlarında mümkün (Vapnik 2000) olmakla birlikte w ve b değişkenleri kullanılarak Birincil Lagrange denkleminin kısmi türevlerinin alınmasıyla, denklem (3.10), (3.11) ve (3.12) elde edilir (Fletcher, 2008).

$$\frac{dL_p}{dw} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i \quad (3.10)$$

$$\frac{dL_p}{db} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad (3.11)$$

$$L_d = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j x_i^T x_j, \alpha_i \geq 0, \forall i \quad (3.12)$$

Denklem (3.12)'de belirtilen karmaşık problem bir ‘‘QP’’ problemidir. Bu problemin kolay bir şekilde çözülebilmesi için Karush-Kuhn-Tucker (KKT) tamamlayıcı koşulu kullanılarak denklem (3.13) elde edilir (Fletcher, 2008).

$$\alpha_i (y_i (w \cdot x_i) + b) - 1 = 0, i = 1, \dots, N \quad (3.13)$$

(3.13) eşitliğinden b yalnız bırakılır. b , eşitlik (3.14)'te verildiği gibidir.

$$b = y_i - w \cdot x_i \quad (3.14)$$

Eğitim setinde yer alan her bir örnek için bir adet Lagrange çarpanı bulunmaktadır. Bu sayede, denklemin çözümü esnasında bulunan pozitif değerli Lagrange çarpanlarının α_i değerli vektörleri x_i destek vektörlerini oluşturur. Oluşturulan bu destek vektörleri Denklem (3.13)'ü sağlayan hiper düzlemler üzerinde yer alır.

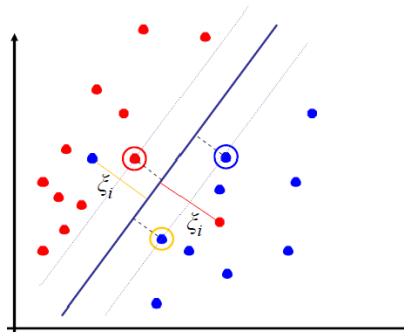
Denklem (3.6)'da verilen karar fonksiyonunun hesaplanan x_i destek vektörleri ve α_i ağırlıklarıyla Denklem (3.10) kullanılarak tekrar ifade edilmesiyle;

$$f(x) = \text{sign}((w \cdot x_i) + b) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i (x \cdot x_i) + b \right) \quad (3.15)$$

karar fonksiyonu Denklem (3.15) olarak yeni şekliyle elde edilir. Bu noktadan sonra, yeni bir x verisinin test edilmesi için Denklem (3.15) uygulanır. Bu denklemin sonucu, eğer pozitif ise x verisinin birinci sınıfa ait olduğu, aksi durumlarda ise ikinci sınıfa ait olduğu kabul edilir.

b) Veri Setinin Doğrusal Ayrılama Durumu:

Her ne kadar veri setlerinin doğrusal olarak ayrılabilmesi mümkün olmakla birlikte, birçok sınıflandırma uygulamasında veri setlerinin doğrusal olarak ayrılması mümkün değildir. Bu durumda ise sınıflandırma problemi alternatif bir çözüm olarak, bir hata değişkeninin tanımlanması ile çözülmektedir. Bu durum Şekil 3.15'te gösterilmektedir.



Şekil 3. 15. Verilerin doğrusal ayrılama durumu (Öğüdücü, 2007)

Veri setlerinin doğrusal olarak ayrılabilmesi durumunda, en iyi ayırıcı düzlemin belirlenmesi için (3.2) ve (3.3) eşitsizlikleri ξ_i değişkeni yardımıyla tekrar yazılarak (3.16), (3.17) ve (3.18) eşitsizlikleri elde edilir. Burada, ξ_i değişkeni eğitim hatası sapması değeri olarak adlandırılır.

$$(w \cdot x_i) + b \geq +1 - \xi_i, y_i = +1 \text{ için} \quad (3.16)$$

$$(w \cdot x_i) + b \leq -1 + \xi_i, y_i = -1 \text{ için} \quad (3.17)$$

$$\xi_i \geq 0, \forall i \quad (3.18)$$

Buna göre yukarıdaki eşitsizlikler yardımıyla kısıtlar Denklem (3.19) gibi yazılabilir.

$$y_i ((w \cdot x_i) + b) - 1 + \xi_i \geq 0, y_i \in \{-1, +1\} \text{ ve } \xi_i \geq 0 \quad (3.19)$$

$\xi_i < 0$ olması durumu, bir x_i örneğinin yanlış sınıflandırılmış olduğu, $0 < \xi_i < 1$ durumunda ise bu örneğin aslında iki sınır hiper düzlemi arasında yer alarak doğru sınıflandırılmış olduğu anlamına gelmektedir.

$\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i$ fonksiyonunu en küçükleyen w vektörü yardımıyla genelleştirilmiş en iyi ayırma düzlemi belirlenir. Bu durumda maksimum sınırın bulunması için $\min \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \right]$ ifadesi denklem (3.16) koşuluna bağlı olarak hesaplanmalıdır (Vapnik, 2000). Sonuç olarak, en iyileme problemi denklem (3.20)'deki gibi ifade edilebilir. Burada sınır ile hata arasındaki üst sınır parametresi C ile gösterilmektedir.

$$\min \left[\frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i \right] \text{ koşulu ile, } y_i ((w \cdot x_i) + b) - 1 + \xi_i \geq 0, \quad (3.20)$$

$$y_i \in \{-1, +1\}$$

Lagrange çarpanlarının $0 < \xi_i < C$ arasında kalması C üst sınırına bağlıdır (Vapnik, 2000). Doğrusal olmayan bir kısıtlı en iyileme problemi, denklem (3.20)'de belirtilen koşul ve eşitliğe göre Lagrange çarpanları $(\alpha_i, i = 1, \dots, N)$ ve fonksiyonu kullanılarak çözülebilmektedir. Burada, α_i değerleri Lagrange çarpanlarıdır (Fletcher 2008).

$$L_p = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \xi_i - \sum_{i=1}^N \alpha_i [y_i (w \cdot x_i + b) - 1 + \xi_i] - \sum_{i=1}^N \mu_i \xi_i \quad (3.21)$$

μ_i değeri, ξ_i değerinin pozitif olmasını sağlayan Lagrange parametresi olmak üzere Denklem (3.21), w , b ve ξ_i değişkenlerine göre en küçüklenir, α_i çarpanlarına göre en büyüklenir. Bu şekilde problemin, Denklem (3.22)'de ki gibi ifade edilmesi yoluyla en iyileme problemleri farklı biçimlere dönüştürülebilirler.

$$\arg \max_{\alpha} \min_{w, b} \{L_p\} \quad (3.22)$$

Bu işlem ise, Lagrange denkleminin doğru değişkenlerine bakılarak kısmi türevlerinin alınıp çözülmesi ve sonuçların denklemde doğru bir biçimde yerlerine konmasıyla ve elenmesiyle yapılır. Sonuç olarak, en büyüklenecek bağıntı sadece Lagrange çarpanlarında mümkün (Vapnik 2000) olmakla birlikte w , b ve ξ_i değişkenleri kullanılarak Birincil Lagrange denkleminin kısmi türevlerinin alınmasıyla, Denklem (3.23), (3.24), (3.25) ve (3.26) elde edilir (Fletcher, 2008).

$$\frac{dL_p}{dw} = 0 \Rightarrow w = \sum_{i=1}^N \alpha_i y_i x_i \quad (3.23)$$

$$\frac{dL_p}{db} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad (3.24)$$

$$\frac{dL_p}{d\xi_i} = 0 \Rightarrow C - \alpha_i - \mu_i = 0 \quad (3.25)$$

$$L_d = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j, C > \alpha_i \geq 0, \forall i \quad (3.26)$$

Problemin çözülebilmesi için (KKT) tamamlayıcı koşulu yardımıyla Denklem (3.27) elde edilir (Fletcher, 2008).

$$\alpha_i (y_i (w \cdot x_i + b) - 1 + \xi_i) = 0, i = 1, \dots, N \quad (3.27)$$

Denklem (3.27)'den b çekilir. b , denklem (3.28)'de verildiği gibidir.

$$b = y_i (1 - \xi_i) - w \cdot x_i \quad (3.28)$$

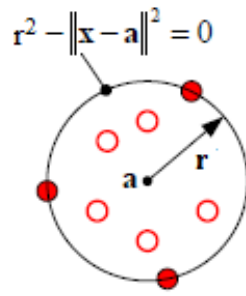
Eğitim setinde yer alan her bir örnek için bir adet Lagrange çarpanı bulunmaktadır. Böylelikle, denklemin çözümü esnasında bulunan pozitif değerli Lagrange çarpanlarının $C > \alpha_i \geq 0, \forall i$ aralığında yer alan α_i değerli vektörleri x_i destek vektörlerini oluşturur. Oluşturulan destek vektörleri Denklem (3.27)'yi sağlayan hiper düzlemler üzerinde yer alır. Denklem (3.6)'da verilen karar fonksiyonun hesaplanan x_i destek vektörleri ve α_i ağırlıklarıyla Denklem (3.23) kullanılarak tekrar ifade edilmesiyle;

$$f(x) = \text{sign}((w \cdot x_i) + b) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i (x \cdot x_i) + b \right) \quad (3.29)$$

karar fonksiyonu Denklem (3.29) olarak yeni şekliyle elde edilir. Bu noktadan sonra, yeni bir x verisinin test edilmesi için Denklem (3.29) uygulanır. Bu denklemin sonucu, eğer pozitif ise x verisinin birinci sınıfa ait olduğu, aksi durumlarda ise ikinci sınıfa ait olduğu kabul edilir.

3.3.2 Doğrusal Hiper-Küre sınıflandırıcısı

Bu yöntem ilk olarak Tax ve Duin (2004) tarafından önerilmiş olup amaç bir sınıfa ait tüm örnekleri içeren en küçük ve kompakt hiper-küreyi bulmaktır. Hiper-küre sınıflandırıcısı kendinden önce gelen Doğrusal Destek Vektör Makineleri (DDVM) sınıflandırıcısı ile birbirlerini tamamlayacak şekilde çalışarak, DDVM'nin yanlışlıkla yüz sınıfına atadığı arka-plana ait örnekleri başarılı bir şekilde elemektedir. Şekil 3.16'da Doğrusal Hiper-Küre sınıflandırıcısı anlatımı gösterilmektedir (Chen vd., 2011).



Şekil 3. 16. Doğrusal Hiper-Küre sınıflandırıcısı anlatımı (Chen vd.,2011)

Şekil 3.16'da da görüldüğü üzere hiper küreyi temsil eden iki parametre vardır: kürenin merkezi a ve kürenin yarıçapı r . Hiper-kürenin merkezi ve yarıçapı veri örneklerinin giriş uzayındaki konumlarına göre değişir ve Denklem (3.30)'daki ikilenik programlama (quadratic programming) probleminin çözülmesiyle bulunurlar. Bir sınıfa ait d -boyutlu uzaydaki veri örnekleri $\{x_i \in R^d\}_{i=1,\dots,n}$ şeklinde olsun. Bu durumda hiper küre bulma problemi aşağıdaki gibi ifade edilebilir:

$$\begin{aligned} \min: & r^2 + C \sum_{i=1}^N \xi_i, \\ \text{s.t:} & \begin{cases} \|x_i - a\|^2 \leq r^2 + \xi_i, \\ \xi_i \geq 0, \end{cases} \end{aligned} \quad (3.30)$$

Genellikle bu problemi çözmek yerine bu problemin Denklem (3.31)'deki verilen çift eşleniği (dual) çözülür.

$$\begin{aligned} \min : & \sum_i \sum_j \alpha_i \alpha_j x_i \cdot x_j - \sum_i \alpha_i x_i \cdot x_i, \\ \text{s.t.} & \begin{cases} \sum_i \alpha_i = 1, \\ 0 \leq \alpha_i \leq C. \end{cases} \end{aligned} \quad (3.31)$$

Yukarıdaki formülde α_i 'ler Lagrange katsayıları olup $C \in [0,1]$ ise kullanıcının belirlediği ve kompakt modele çok uzak olan aykırı değerleri belirlemede kullanılan bir katsayıdır. Bu ikilenik eniyileme problemi de konveks bir problem olup, global bir en iyi noktası vardır. Problemin çözümünü veren optimal α_i katsayıları belirlendikten sonra hiper-kürenin merkezi $a = \sum_{i=1}^n \alpha_i x_i$ formülü ile yarıçapı ise $0 < \alpha_i < C$ şartını sağlayan Lagrange katsayısına karşılık gelen herhangi bir x_i örneğinin hiper-kürenin merkezine olan uzaklığına bakılarak bulunur,

$$r = \|x_i - a\|. \quad (3.32)$$

Eğer elimizde yüz sınıfına ait olmayan örnekler (bizim problemimizde arka-plana ait örnekler) varsa bu örneklerden yararlanılarak daha iyi hiper-küre modelleri tanımlanabilir. Bu durumda amaç yüz sınıfına ait örnekleri içeren ve yüz sınıfına ait olmayan örnekleri dışarıda bırakan bir hiper-küre bulmaktır. Elimizde yüz sınıfına ait n_1 adet ve i ve j indisleriyle gösterilen örneklerle birlikte arka-plana ait n_2 adet l ve m indisleriyle belirtilen örnekler olduğunu varsayalım. Yüze ait örnekleri içeren ve arka-plana ait örnekleri dışarıda bırakan en kompakt hiper-küre bulma problemi denklem (3.33)'teki ikilenik en iyileme problemi şeklinde ifade edilebilir.

$$\begin{aligned} \min_{a, r \geq 0, \xi \geq 0} & \quad r^2 + C_1 \sum_i \xi_i + C_2 \sum_l \xi_l \\ \text{s.t.} & \quad \|x_i - a\|^2 \leq r^2 + \xi_i, \quad i=1, \dots, n_1 \\ & \quad \|x_l - a\|^2 \leq r^2 + \xi_l, \quad l=1, \dots, n_1 \end{aligned} \quad (3.33)$$

Daha önceki durumda olduğu gibi bu problemi çözmek yerine genellikle problemin Denklem (3.34)'te verilen çift eşleniği çözülür.

$$\begin{aligned} \min_a \quad & \sum_{i,j} \alpha_i \alpha_j \langle x_i, x_j \rangle + \sum_{l,m} \alpha_l \alpha_m \langle x_l, x_m \rangle - 2 \sum_{l,j} \alpha_l \alpha_j \langle x_l, x_j \rangle + \left(\sum_l \alpha_l \|x_l\|^2 - \sum_i \alpha_i \|x_i\|^2 \right) \\ \text{s.t.} \quad & \sum_i \alpha_i - \sum_l \alpha_l = 1, \quad , \quad \forall i, j \quad 0 \leq \alpha_i \leq C_1, \quad 0 \leq \alpha_l \leq C_2 \end{aligned} \quad (3.34)$$

Bu sınıflandırıcı iki sınıfa ait örnekleri de kullanmasına rağmen DVM'lerinden oldukça farklı prensipte çalışır. DVM iki sınıfı birbirinden ayıracak hiper-düzlem döndürürken, hiper-küre sınıflandırıcısı ise yüze ait sınıfın örneklerini içeren ve arka-plan sınıfına ait örnekleri içermeyen bir küre döndürür. Bu durumda iki sınıfı ayıran karar sınırları DVM'lerinden farklı olarak doğrusal değildir. En iyi Lagrange katsayıları bulunduğundan sonra hiper-kürenin merkezi Denklem (3.35) ile bulunur (Çevikalp ve Triggs, 2012).

$$a = \sum_{i=1}^{n_1} \alpha_i x_i - \sum_{l=1}^{n_2} \alpha_l x_l \quad (3.35)$$

Yaptığımız deneysel çalışmalarda arka-plana ait örneklerin dahil edilmesi sınıflandırıcının performansını arttırdığından hiper-küre sınıflandırıcısı için Denklem (3.34)'te verilen formülasyon kullanılmıştır. Bu ikilenik en iyileme problemi de konvektir ve global en iyi noktası vardır. Eğitim setindeki örneklerin sayısının fazla olduğu durumlarda bu ikilenik problemde DVM'nde olduğu gibi "Sequential Minimal Optimization" tekniği kullanılarak çözülebilir (Çevikalp ve Triggs, 2012). Bu durumda çok boyutlu Hessian matrisini oluşturmaya gerek kalmadan, iteratif olarak sadece kısıtları bozan iki örneğin Hessian matrisleri kullanılır. Biz bu amaçla <http://cmp.felk.cvut.cz/> sayfasında paylaşımına açılmış ikilenik program çözücüyü geliştirerek çok fazla sayıda veriyle çalışacak hale getirdik ve hiper-küreleri bulmak için geliştirdiğimiz bu algoritmayı kullandık.

Hiper-küreyi karakterize eden parametreler (kürenin merkezi a ve kürenin yarıçapı r) bulunduktan sonra bir test örneğinin, x_{test} , kürenin merkezine olan uzaklığı bulunur ve bu uzaklık yarıçapla karşılaştırılarak o örneğin yüz sınıfına ait olup olmadığı belirlenir.

Matematiksel olarak ifade edilirse:

$$\begin{aligned} |(\|x_{test} - a\|) - r| \leq \theta &\Rightarrow x_{test} \text{ yüz sınıfına ait;} \\ |(\|x_{test} - a\|) - r| > \theta &\Rightarrow x_{test} \text{ arka-plana ait.} \end{aligned}$$

3.3.3 Doğrusal Olmayan Hiper-Küre sınıflandırıcısı

Ardışıl sınıflandırıcının son katmanında Doğrusal Olmayan Hiper-Küre sınıflandırıcısı kullanılmıştır. Hiç şüphesiz sınıfların topolojisi çok daha karmaşık ve konveks olmayan yapılar olabilir. Bu durumda sınıfları doğrusal hiper-kürelerle modellemek uygun değildir. Bu problemin çözümü için izlenen yol, örnekleri kernel hilesini kullanarak bu modelin uygun olduğu çok daha büyük boyutlu bir uzaya çıkarmak ve bu yeni uzayda hiper kürenin parametrelerini bulmaktır. Bu işlem Destek Vektör Makinelerinde olduğu gibi Denklem (3.34)'deki iç çarpımların kernel fonksiyonlarla değiştirilmesi ile kolayca gerçekleştirilir (Çevikalp vd., 2012). Daha açık bir şekilde ifade edilirse her bir iç çarpım $\langle x_i, x_j \rangle$, kernel fonksiyonu $k \langle x_i, x_j \rangle = \langle \phi(x_i), \phi(x_j) \rangle$ ile değiştirilmelidir. Burada $\phi: R^d \rightarrow \mathfrak{F}$ giriş uzayındaki örnekleri büyük boyutlu yeni uzaya aktaran fonksiyonu ifade etmektedir. Kullanılan kernel fonksiyonlar arasında m . dereceden polinom fonksiyonu $k(x_i, x_j) = (\langle x_i, x_j \rangle)^m$, Gaussian kerneli $k(x_i, x_j) = \exp(-\gamma \|x_i, x_j\|^2)$ ve sigmoid kerneli $k(x_i, x_j) = \tanh(\kappa \langle x_i, x_j \rangle) + \Theta$ sayılabilir. Bu sayede çok daha karmaşık sınıfların topolojileri daha doğru bir şekilde modellenebilir (Çevikalp vd., 2012).

Doğrusal olmayan durumda bir örneğin hiper-küreye olan uzaklığının hesaplanabilmesi için sıfırdan farklı Lagrange katsayılarına karşılık gelen destek vektörleri kullanılarak kernel fonksiyonların hesaplanması gerekir. Bu sebeple Doğrusal Olmayan Hiper-Küre sınıflandırıcısı, Doğrusal Hiper-Küre sınıflandırıcısına oranla oldukça yavaştır.

Fakat bu sınıflandırıcı son katmana gelen az sayıda örneğe uygulandığı için konum bulma sisteminin genel hızını çok kötü etkilememektedir.

Doğrusal Olmayan Hiper-Küre sınıflandırıcısı sınıflandırma başarımı yönünden DVM sınıflandırıcıları ile karşılaştırıldığında iki durum ortaya çıkmaktadır: Yüz grubuna ait örnek sayısı çok fazla ise başarımları yönünden bu iki sınıflandırıcısı arasında pek bir fark gözlenmemiştir. Fakat örnek sayısı az ise Doğrusal Olmayan DVM sınıflandırıcısı çok daha başarılı sonuçlar vermiştir.



4.BULGULAR VE TARTIŞMA

Bu tez çalışması kapsamında geliştirilen yüz sezme yöntemi bölüm 3.1.2’de anlatılan test veri tabanı üzerinde test edilmiştir. Geliştirdiğimiz yüz sezme yönteminin başarımını ölçmek için PASCAL VOC ölçütü kullanılmıştır. Bu ölçüte göre yüz sezme yönteminin döndürdüğü pencere R ile elle işaretlenen yüz konumlarını içeren pencereler Q

arasındaki örtüşmenin tüm alana olan oranı, $\frac{alan|Q \cap R|}{alan|Q \cup R|}$, %45’ten büyükse bu doğru bir

yüz olarak sayılmıştır. Aksi durumda bulunan pencere yanlış (false positive – FP) olarak kabul edilmiştir. Bunun bir sebebi farklı kişiler tarafından işaretlenen yüz bölgesinin kişilere bağlı olarak farklı olarak yorumlanmasıdır. Test çalışmasında sınıflandırıcının performansını gösteren ve TP, FP, FN ve TN gibi terimleri açıklayan karışıklık matrisi (confusion matrix) Çizelge 4.1’de gösterilmektedir (Öğüdücü, 2007). Daha sonra PASCAL VOC ölçütü kullanılarak doğru ve yanlış konumlar olarak belirlenen pencerelerin skorları kullanılarak Precision-Recall eğrileri elde edilmiş ve başarımlarını performansları olarak bu eğrilerden elde edilen ortalama kesinlik skorları (average precision – mAP) kullanılmıştır.

Çizelge 4. 1. Karışıklık-hata matrisi (confusion - error matrix) (Öğüdücü, 2007)

| | | GERÇEK DEĞER (Deneylerle Onaylanan) | |
|---|-------------------|--|---|
| | | <i>Pozitifler</i> | <i>Negatifler</i> |
| TAHMİNİ DEĞER (Test sonucu ortaya çıkan) | <i>Pozitifler</i> | *TP True Positive (Doğru Pozitifler) | *FP (Hard Negatives) False Positive (Yanlış Pozitifler) |
| | <i>Negatifler</i> | *FN(Hard Positives) False Negative (Yanlış Negatifler) | *TN True Negative (Doğru Negatifler) |

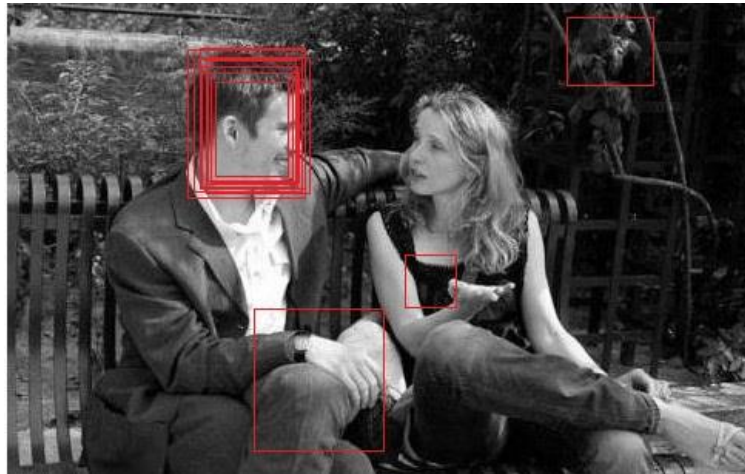
***True Positive** : Pozitif bir örneğin doğru veya uygun tahminlerinin sayısıdır.

***False Negative** : Negatif bir örneğin yanlış veya uygun olmayan tahminlerinin sayısıdır.

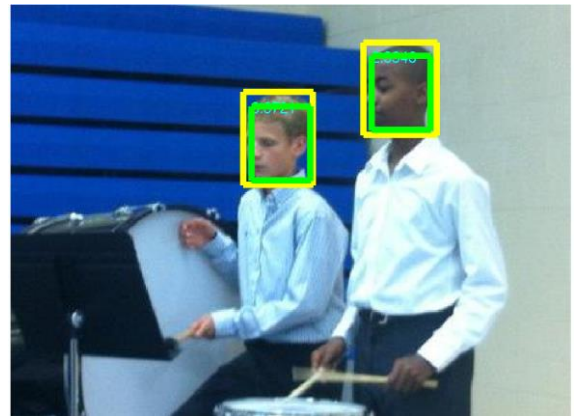
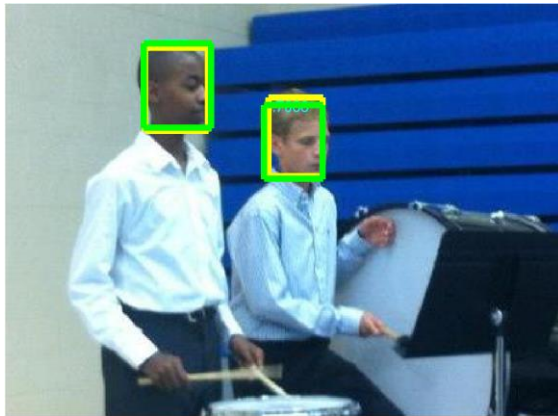
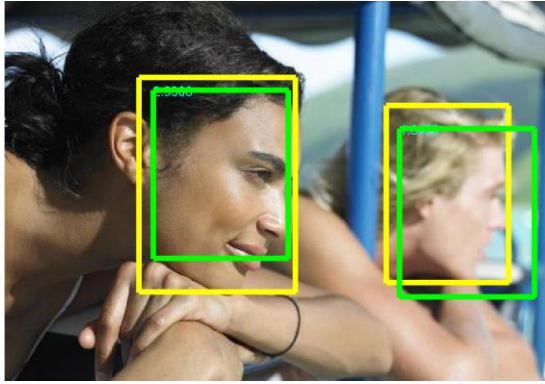
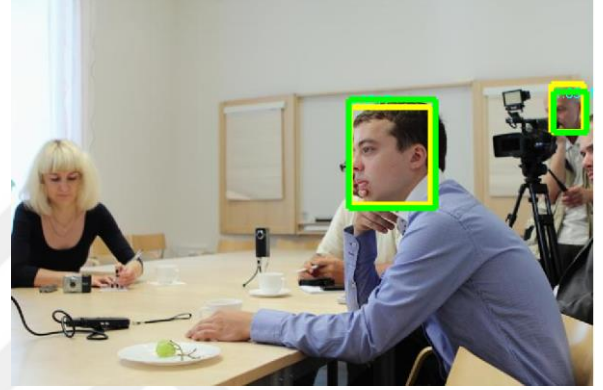
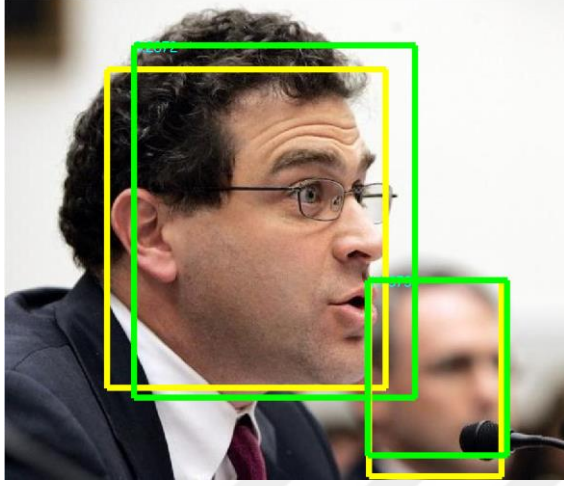
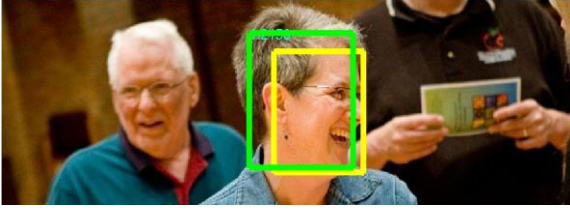
***False Positive** : Pozitif bir örneğin yanlış ya da uygun olmayan tahminlerinin sayısıdır.

***True Negative** : Negatif bir örneğin doğru veya uygun tahminlerinin sayısıdır.

Test aşamasında kayan pencereler yöntemi kullanılmıştır. Bu yöntemde göre ilk olarak 1.15 ölçek skalası kullanılarak imge alt örnekleme işlemine tabi tutulmuş ve farklı ölçeklerden oluşan bir imge piramidi oluşturulmuştur. Uygulama sırasında kullanılan pencere yatay düzlemde 3 piksel dikey düzlemde 4 piksel kaydırılmıştır. Bu şekilde imgede yüzün bulunduğu bölgelerin etrafında birçok pencere Şekil 4.1’de görüldüğü gibi yüz olarak sınıflandırılmaktadır. Bu pencereler kullanılarak yüzün bulunduğu en iyi (en yüksek skora sahip) tek bir konum bulunmalıdır. Bu amaçla çalışma sırasında maksimum olmayanı bastırma (non-max suppression) tekniği kullanılmıştır. Bu tekniğe göre ardışıl sınıflandırıcıların yüz olarak sınıflandırdığı tüm pencereler en yüksek skordan başlayarak büyükten küçüğe sıralanmış ve iteratif olarak en yüksek skora sahip pencere yüz olarak kabul edilmiş ve onunla aynı özellikleri gösteren diğer pencereler elenmiştir. Şekil 4.1’de görüldüğü gibi arka-plana ait başka pencerelerde ardışıl sınıflandırıcılar tarafından yüz olarak döndürülmesine rağmen yüze benzemeyen bölgelerde döndürülen pencere sayısı oldukça azdır. Bu nedenle yüzün olduğu bölgelerde genel olarak fazla sayıda pencere yüz olarak döndürüldüğü için bu bilgiler kullanılarak, arka-plandan gelen bazı yanlış sınıflandırılmalar elenebilir. Çalışma sırasında ardışıl sınıflandırıcının döndürdüğü bir pencereyle aynı özellikleri gösteren (örtüşen) en az 3 pencere daha bulunmuyorsa bu pencere yüz olarak kabul edilmeyerek elenmiştir. Bu da bizim yüz olmadığı halde yüz olarak döndürülen birçok pencereyi (false positive) elememize imkan tanımıştır.

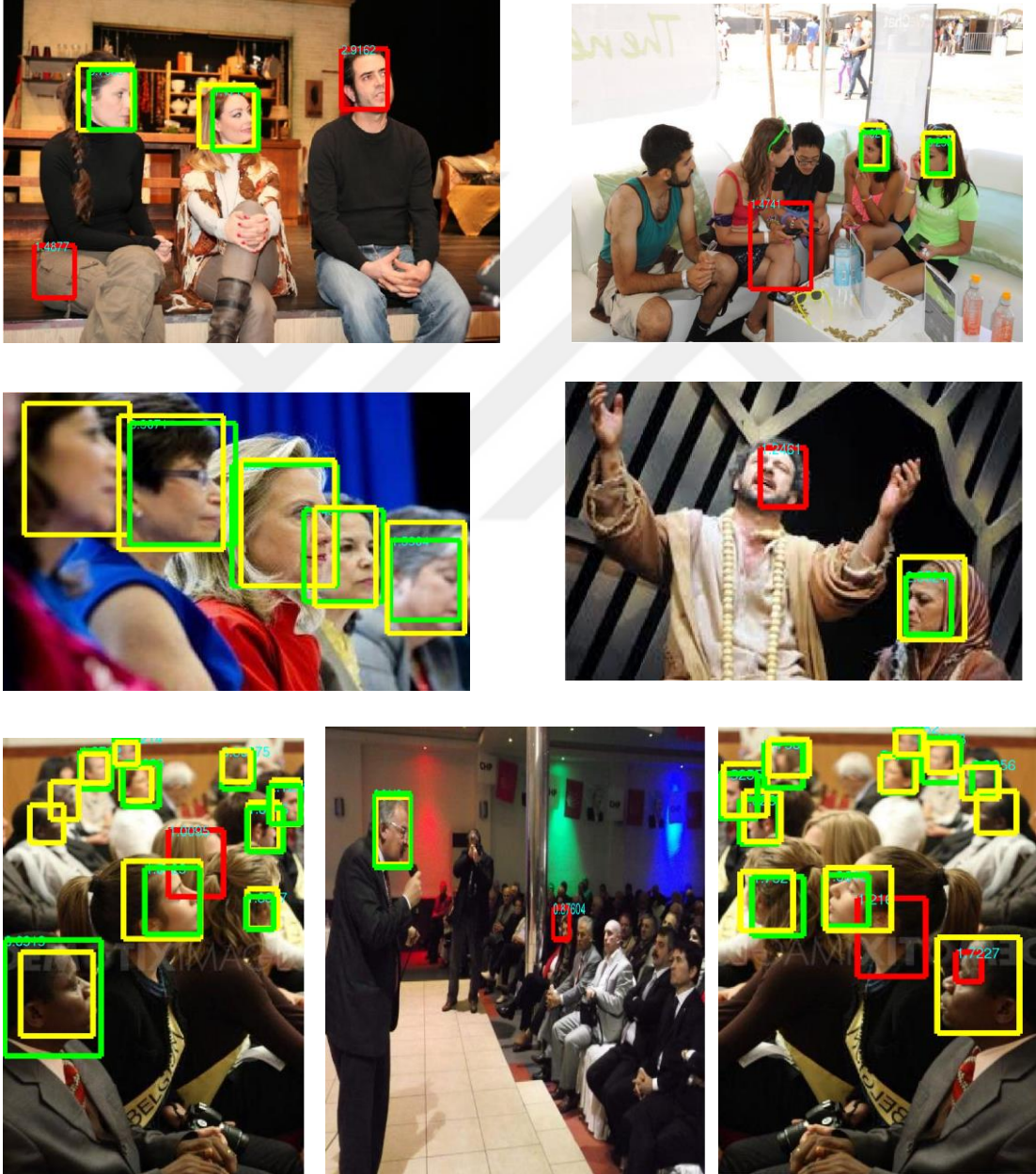


Şekil 4. 1. Maksimum olmayanı bastırma uygulanmamış yüz sezme algoritması çıktıları



Şekil 4. 2. Sağ ve sol profil yüz sezme sistemi tarafından başarı ile bulunan örnekler

Şekil 4.2’de tez kapsamında sağ ve sol profil yüz sezimi için geliştirilen sistemin 239 farklı resim ve 527 profil imgesi içeren test veri tabanı üzerinde uygulanması sonucu elde edilen çıktılardan bazıları gösterilmektedir. Şekil üzerindeki “Sarı renkli çerçeveler” elle işaretlenen “GroundTruth” doğrulama kısmındaki doğru konumları (koordinatları) yeşil çerçeveler ise sistem tarafından döndürülen ve PASCAL VOC ölçütü kullanılarak doğru olarak kabul edilen konumları göstermektedir.



Şekil 4. 3. Sağ ve sol profil yüz sezme sistemi tarafından hatalı bulunan örnekler

Şekil 4.3'te yüz bulma sisteminin hata yaptığı bazı örnekler gösterilmektedir. “Sarı çerçeveler” elle işaretlenen doğru konumları “yeşil çerçeveler” sistem tarafından döndürülen ve PASCAL VOC ölçütü kullanılarak doğru olarak kabul edilen konumları, “kırmızı çerçeveler” ise PASCAL VOC ölçütüne göre yanlış olarak değerlendirilen konumları göstermektedir. Görüldüğü gibi sistem cepheden baş yönelimi fazla yüzleri, diğer nesnelere tarafından engellenen yüzleri kaçırabilmekte ve cepheden çekilen yüz resimlerini profil resmi olarak algılayabilmektedir. Ayrıca bazı örneklerde de arka-planda yüzü andıran bölgeler yanlış yüz resmi olarak döndürülmüştür (kırmızı çerçevelerle belirtilen bölgeler).

Eğitim kısmında elde edilen sağ ve sol profil sınıflandırıcısının geliştirilmesi için test veri tabanı üzerinde ardışıl sınıflandırıcı parametreleri değiştirilerek 7 farklı test sınıflandırıcısı denenmiştir. En iyi sınıflandırıcıyı bulabilmek için sistemin katmanlarında yer alan Doğrusal Destek Vektör Makinelerinin (DDVM) bias parametresi “b”, Doğrusal Hiper-Küre sınıflandırıcısının (LHS) ve Doğrusal Olmayan Hiper-Küre sınıflandırıcısının (NHS) yarıçap (radius) “r” değerleri değiştirilerek en iyi sonucu veren sınıflandırıcı sistemi bulunmaya çalışılmıştır.

4.1 Sağ Profil Yüz Sezme Test Sonuçları

Eğitim kısmında elde edilen sağ profil sınıflandırıcısının (DDVM) bias parametresi “b”, değeri: 1.4147, (LHS) “r” değeri: 2.1423 ve (NHS) “r” değeri: 0.9898 olarak bulunmuştur. Bu değerler Çizelge 4.2, 4.4, 4.6, 4.8, 4.10, 4.12, 4.14'te “Orijinal Değer” sütununda gösterilmiştir.

Çalışma kapsamında eğitim kısmında elde edilen sınıflandırıcısının farklı “b” ve “r” değerlerine göre yüz sezme sonuçlarının nasıl değiştiğini gözlemlemek ve başarı oranının test etmek amacıyla 7 farklı test işlemi gerçekleştirilmiştir. Bu test işlemlerine sırasıyla Test 1-7 aralığında numara verilerek her bir test işleminde (DDVM), (LHS) ve (NHS) sınıflandırıcıları için farklı “b” ve “r” parametreleri kullanılmıştır. Çizelge 4.2'den başlayarak 4.15'e kadar bu test işlemleri kapsamında kullanılan “b” ve yarıçap “r” değerleri ile bu değerlere bağlı olarak elde edilen test sonuçları gösterilmektedir.

Çizelgelerde verilen test sonuçlarına göre en yüksek mAP değerini veren sınıflandırıcı en iyi sınıflandırıcı olarak seçilmiştir.

Çizelge 4. 2. Test 1 Sağ profil yüz sezme kullanılan değerler

| TEST 1 (sağ profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,8800 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1375 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9895 | r=0.9898 |

Çizelge 4. 3. Test 1 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 549 | 113 | 414 | 0.7856 | 0.6838 |

Çizelge 4. 4. Test 2 sağ profil yüz sezme kullanılan değerler

| TEST 2 (sağ profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,7800 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1370 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9895 | r=0.9898 |

Çizelge 4. 5. Test 2 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 447 | 136 | 391 | 0.7419 | 0.6492 |

Çizelge 4. 6. Test 3 sağ profil yüz sezme kullanılan değerler

| TEST 3 (sağ profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,8890 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1368 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9895 | r=0.9898 |

Çizelge 4. 7. Test 3 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TIS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 571 | 109 | 418 | 0.7932 | 0.6872 |

Çizelge 4. 8. Test 4 sağ profil yüz sezme kullanılan değerler

| TEST 4 (sağ profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,8875 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1370 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9896 | r=0.9898 |

Çizelge 4. 9. Test 4 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|----------------------|------------|
| 239 | 527 | 695 | 93 | 434 | 0.8236 | 0.6992 |

Çizelge 4. 10. Test 5 sağ profil yüz sezme kullanılan değerler

| TEST 5 (sağ profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,9800 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1410 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9896 | r=0.9898 |

Çizelge 4. 11. Test 5 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| 239 | 527 | 1330 | 73 | 454 | 0.8615 | 0.7094 |

Çizelge 4. 12. Test 6 sağ profil yüz sezme kullanılan değerler

| TEST 6 (sağ profil yüz sezme) | | |
|--|------------------|----------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,9050 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1367 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9895 | r=0.9898 |

Çizelge 4. 13. Test 6 sağ profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| 239 | 527 | 620 | 110 | 417 | 0.7913 | 0.6793 |

Çizelge 4. 14. Test 7 sağ profil yüz sezme kullanılan değerler

| TEST 7 (sağ profil yüz sezme) | | |
|--|------------------|----------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= 0,8890 | 1.4147 |
| Doğrusal Hiper-Küre (LHS) | r=2.1371 | r= 2.1423 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9896 | r=0.9899 |

Çizelge 4. 15. Test 7 sağ profil yüz sezme test sonuçları

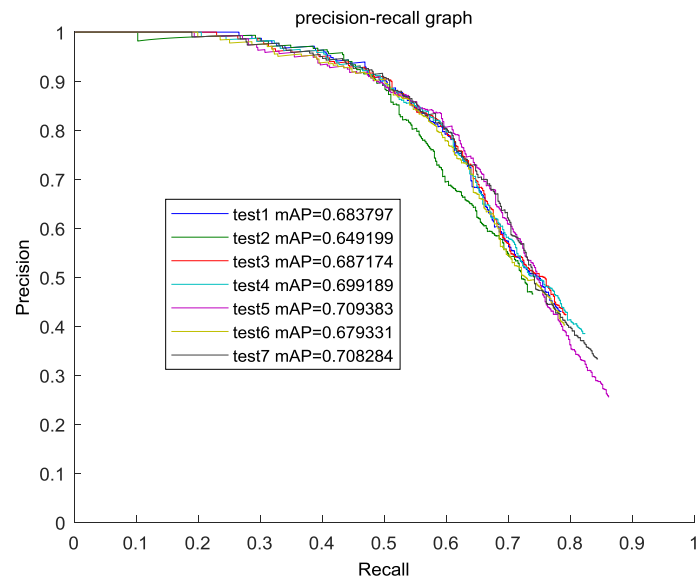
| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| 239 | 527 | 891 | 82 | 445 | 0.8444 | 0.7083 |

Çizelge 4.16’da, sağ profil yüz sezme test sonuçlarının genel gösterimi yer almaktadır. Sarı renkle boyalı satırlar en iyi 3 sonucu veren test sonuçlarını göstermektedir.

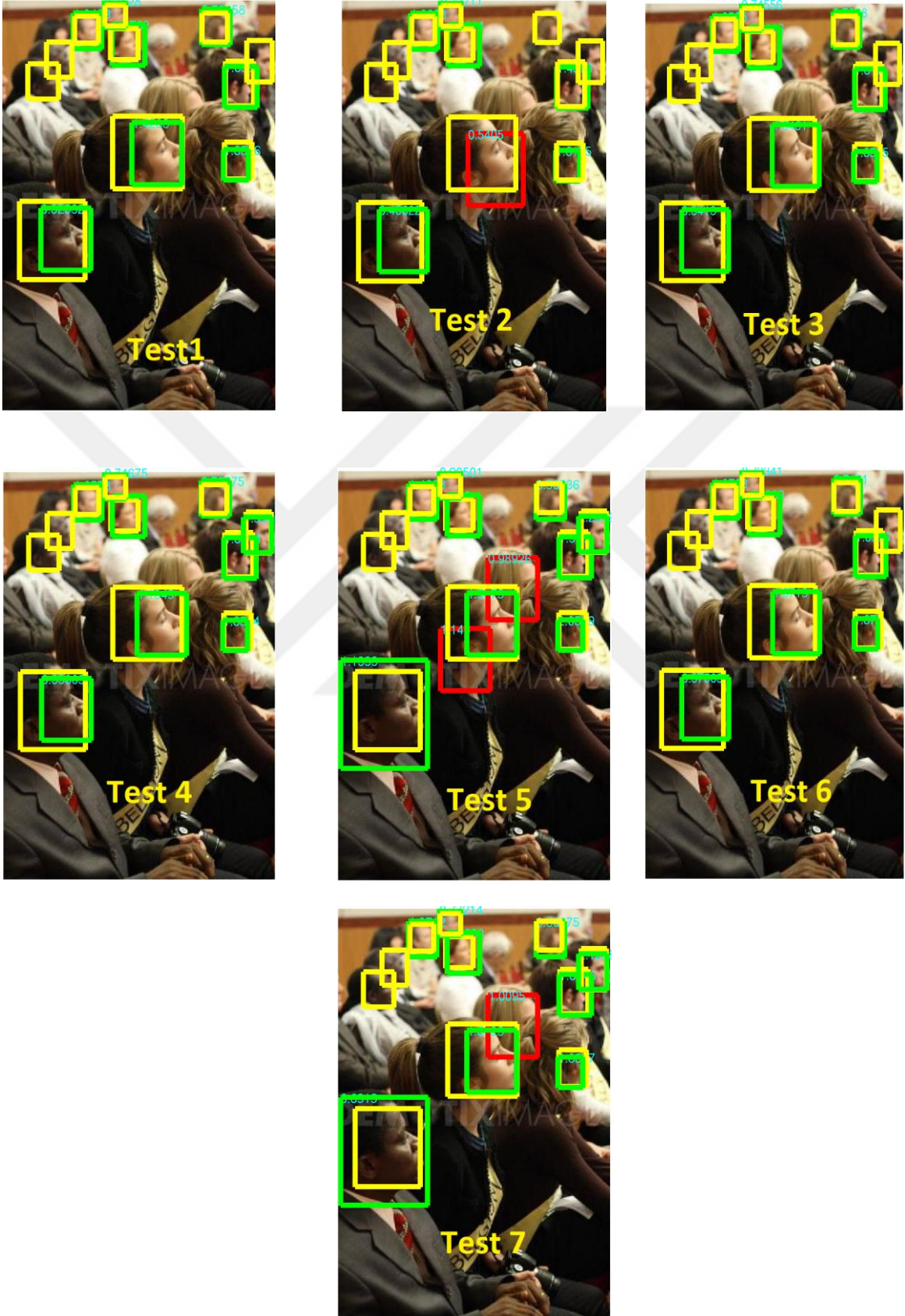
Çizelge 4. 16. Sağ profil yüz sezme test sonuçları

| TEST NO | Test Resim Sayısı | Toplam İmge Sayısı (TIS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|---------|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| TEST 1 | 239 | 527 | 549 | 113 | 414 | 0.7856 | 0.6838 |
| TEST 2 | 239 | 527 | 447 | 136 | 391 | 0.7419 | 0.6492 |
| TEST 3 | 239 | 527 | 571 | 109 | 418 | 0.7932 | 0.6872 |
| TEST 4 | 239 | 527 | 695 | 93 | 434 | 0.8235 | 0.6992 |
| TEST 5 | 239 | 527 | 1330 | 73 | 454 | 0.8615 | 0.7094 |
| TEST 6 | 239 | 527 | 620 | 110 | 417 | 0.7913 | 0.6793 |
| TEST 7 | 239 | 527 | 891 | 82 | 445 | 0.8444 | 0.7083 |

Şekil 4.4’te 7 adet sağ profil test sonucunun Matlab yardımıyla çizdirilmiş “Precision-Recall” Eğrileri grafiği, Şekil 4.5’te ise test veri tabanındaki aynı örnek üzerinde 7 farklı test sonuçları gösterilmektedir.



Şekil 4. 4. Sağ profil test 1-7 Precision-Recall Eğrileri grafiği



Şekil 4. 5. Aynı örnek üzerindeki sağ profil test 1-7 çıktıları

4.2 Sol Profil Yüz Sezme Test Sonuçları

Sağ profil yüz sezme kısmında uygulanan işlemlerin aynısı bu kısımda da uygulanmıştır. Eğitim kısmında elde edilen sol profil sınıflandırıcının (DDVM) bias parametresi “b”, değeri: -0.2337, (LHS) “r” değeri: 2.3977 ve (NHS) “r” değeri: 0.9906 olarak bulunmuştur. Bu değerler sırasıyla Çizelge 4.17, 4.19, 4.21, 4.23, 4.25, 4.27, 4.29’da sağ profilde olduğu gibi “Orijinal Değer” sütununda gösterilmiştir.

Çalışma kapsamında eğitim kısmında elde edilen sınıflandırıcının farklı “b” ve “r” değerlerine göre yüz sezme sonuçlarının nasıl değiştiğini gözlemlemek ve başarı oranının test etmek amacıyla sağ profil kısmında da anlatılan şekilde 7 farklı test işlemi gerçekleştirilmiştir. Bu test işlemlerine sırasıyla Test 1-7 aralığında numara verilerek her bir test işleminde (DDVM), (LHS) ve (NHS) sınıflandırıcıları için farklı “b” ve “r” parametreleri kullanılmıştır. Çizelge 4.17’den başlayarak 4.30’a kadar bu test işlemleri kapsamında kullanılan “b” ve yarıçap “r” değerleri ile bu değerlere bağlı olarak elde edilen test sonuçları gösterilmektedir. Bu test sonuçlarına göre en yüksek mAP değerini veren sınıflandırıcı en iyi sınıflandırıcı olarak seçilmiştir.

Çizelge 4. 17. Test 1 sol profil yüz sezme kullanılan değerler

| TEST 1 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= -0,4500 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r=2.3843 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9902 | r=0.9906 |

Çizelge 4. 18. Test 1 sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 533 | 140 | 387 | 0.7343 | 0.6612 |

Çizelge 4. 19. Test 2 sol profil yüz sezme kullanılan değerler

| TEST 2 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= - 0,5500 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3848 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9903 | r=0.9906 |

Çizelge 4. 20. Test 2 sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 922 | 113 | 414 | 0.7856 | 0.6916 |

Çizelge 4. 21. Test 3 sol profil yüz sezme kullanılan değerler

| TEST 3 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= - 1,3000 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3910 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9905 | r=0.9906 |

Çizelge 4. 22. Test 3 Sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 958 | 91 | 436 | 0.8273 | 0.6715 |

Çizelge 4. 23. Test 4 sol profil yüz sezme kullanılan değerler

| TEST 4 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= - 0,4500 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3800 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9903 | r=0.9906 |

Çizelge 4. 24. Test 4 sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 729 | 116 | 411 | 0.7799 | 0.6930 |

Çizelge 4. 25. Test 5 sol profil yüz sezme kullanılan değerler

| TEST 5 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= - 0,8500 | - 0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3830 | r = 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9904 | r = 0.9906 |

Çizelge 4. 26. Test 5 sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 746 | 118 | 409 | 0.7761 | 0.6751 |

Çizelge 4. 27. Test 6 sol profil yüz sezme kullanılan değerler

| TEST 6 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= -1,2000 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3900 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9904 | r= 0.9906 |

Çizelge 4. 28. Test 6 sol profil yüz sezme test sonuçları

| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 768 | 106 | 421 | 0.7989 | 0.6751 |

Çizelge 4. 29. Test 7 sol profil yüz sezme kullanılan değerler

| TEST 7 (sol profil yüz sezme) | | |
|--|-------------------------|-----------------------|
| Sınıflandırıcı Katmanı İsmi | Kullanılan Değer | Orijinal Değer |
| Doğrusal Destek Vektör Makineleri (DDVM) | b= - 0,4500 | -0.2337 |
| Doğrusal Hiper-Küre (LHS) | r= 2.3825 | r= 2.3977 |
| Doğrusal Olmayan Hiper-Küre (NHS) | r= 0.9904 | r= 0.9906 |

Çizelge 4. 30: Test 7 sol profil yüz sezme test sonuçları

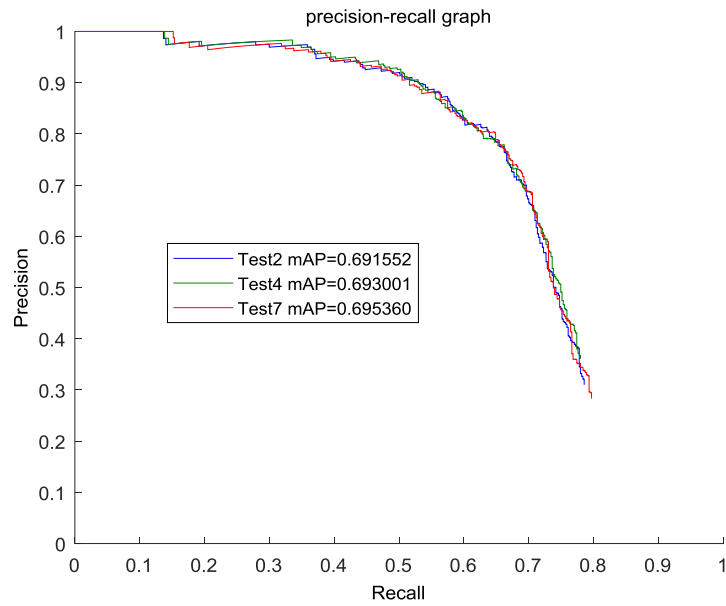
| Test Resim Sayısı | Toplam İmge Sayısı (TİS) | FP ALL | NM ALL | TP ALL | TP/TİS Oranı | mAP |
|--------------------------|---------------------------------|---------------|---------------|---------------|---------------------|------------|
| 239 | 527 | 1065 | 107 | 420 | 0.7970 | 0.6953 |

Çizelge 4.31'de ise sol profil yüz sezme test sonuçlarının genel gösterimi yer almaktadır. Sarı renkle boyalı satırlar en iyi 3 sonucu veren test sonuçlarını göstermektedir.

Çizelge 4. 31. Sol profil yüz sezme test sonuçları

| TEST NO | Test Resim Sayısı | Toplam İmge Sayısı (TIS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|---------|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| TEST 1 | 239 | 527 | 533 | 140 | 387 | 0.7343 | 0.6612 |
| TEST 2 | 239 | 527 | 922 | 113 | 414 | 0.7856 | 0.6916 |
| TEST 3 | 239 | 527 | 958 | 91 | 436 | 0.8273 | 0.6715 |
| TEST 4 | 239 | 527 | 729 | 116 | 411 | 0.7799 | 0.6930 |
| TEST 5 | 239 | 527 | 746 | 118 | 409 | 0.7761 | 0.6751 |
| TEST 6 | 239 | 527 | 768 | 106 | 421 | 0.7989 | 0.6751 |
| TEST 7 | 239 | 527 | 1065 | 107 | 420 | 0.7970 | 0.6954 |

Şekil 4.6’da 3 farklı en iyi sonuç veren (Test 2-4-7) sol profil test sonucunun Matlab yardımıyla çizdirilmiş “Precision-Recall” Eğrileri grafiği, Şekil 4.7’de ise test veri tabanındaki aynı örnek üzerinde 7 farklı test sonuçları gösterilmektedir.



Şekil 4. 6. Sol profil test 2-4-7 Precision-Recall Eğrileri grafiği



Şekil 4. 7. Aynı örnek üzerindeki sol profil test 1-7 çıktıları

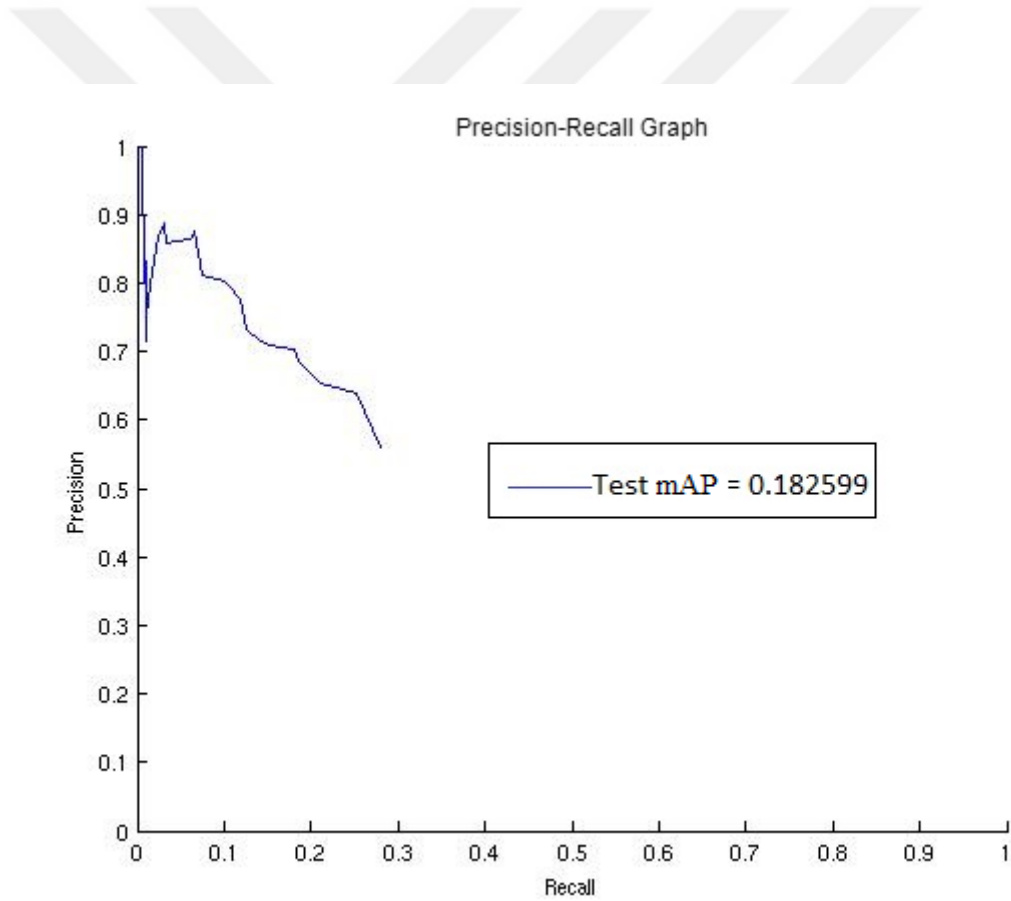
4.3 Viola&Jones Profil Yüz Sezme Test Sonuçları

Çalışmanın bu bölümünde OPENCV ile birlikte gelen Viola&Jones tarafından önerilen ardışıl sınıflandırıcıdan 20x20 profil yüz sezme algoritmasının hazır kaynak kodu olan “haarcascade_profileface.xml” test veri tabanı üzerinde test edilmiştir. Bu kaynak kodu tek parça halinde olup sağ ve sol profil yüz sezme algoritmaları içerisinde yer almaktadır. Daha sonra PASCAL VOC ölçütü kullanılarak doğru ve yanlış konumlar olarak belirlenen pencerelerin skorları kullanılarak Precision-Recall eğrileri elde edilmiş ve başarımlarını performans olarak bu eğrilerden elde edilen ortalama kesinlik skorları (average precision – mAP) kullanılmıştır.

Çizelge 4.32’de Viola&Jones profil yüz sezme test sonuçları, Şekil 4.8’de ise bu sonuçlar ve Matlab yardımıyla çizdirilmiş “Precision-Recall” Eğrileri grafiği, gösterilmektedir.

Çizelge 4. 32. Viola&Jones profil yüz sezme test sonuçları

| TEST NO | Test Resim Sayısı | Toplam İmge Sayısı (TIS) | FP ALL | NM ALL | TP ALL | TP /TIS Oranı | mAP |
|---------|-------------------|--------------------------|--------|--------|--------|---------------|--------|
| TEST | 239 | 527 | 246 | 375 | 152 | 0.2884 | 0.1826 |



Şekil 4. 8. Viola&Jones Precision-Recall Eğrileri grafiği



Şekil 4. 9. Viola&Jones profil yüz sezme test çıktıları

Şekil 4.9’da Viola&Jones tarafından önerilen profil yüz sezme algoritmasının test çıktıları görülmektedir. “Sarı çerçeveler” elle işaretlenen doğru konumları “yeşil çerçeveler” sistem tarafından döndürülen ve PASCAL VOC ölçütü kullanılarak doğru olarak kabul edilen konumları, “kırmızı çerçeveler” ise PASCAL VOC ölçütüne göre yanlış olarak değerlendirilen konumları göstermektedir.

4.4 Profil Yüz Sezme Yöntemlerinin Karşılaştırılması

Bölüm 4.1, 4.2 ve 4.3'te elde edilen sonuçların karşılaştırılması Çizelge 4.33'te gösterilmektedir. Değerler ortalama kesinlik skorları olup Precision-Recall eğrilerinin altında kalan alanların hesaplanmasıyla bulunmuştur.

Sağ profil yüz sezme ortalama kesinlik skoru için test aşamasında en iyi sonucu veren "Test 5" ($mAP=0.7094 = \% 70.9$), Sol profil yüz sezme için ise "Test 7" ($mAP=0.6954 = \%69.5$) sınıflandırıcısı kullanılmıştır.

Karşılaştırma tablosundan da görüldüğü üzere çalışma kapsamında geliştirdiğimiz ardışıl yüz sezme yöntemi ile bir diğer ardışıl sınıflandırıcı olan Viola & Jones yüz sezme yöntemine göre çok daha başarılı sonuçlar elde edilmiştir.

Çizelge 4. 33. Profil yüz sezme yöntemlerinin karşılaştırılması

| Yöntemler | Ortalama Kesinlik Skorları-Average Precision % |
|---|--|
| Sağ Profil Yüz Sezme (Geliştirilen Ardışıl Sınıflandırıcı) | 70.9 |
| Sol Profil Yüz Sezme (Geliştirilen Ardışıl Sınıflandırıcı) | 69.5 |
| Viola & Jones | 18.3 |

5. SONUÇ VE ÖNERİLER

Bu çalışmada dijital imgelerdeki yüzleri bulmak için 3 katmanlı bir ardışıl sınıflandırıcı sistemi geliştirdik. Ardışıl sınıflandırıcının ilk katmanında DDVM sınıflandırıcısı kullanılırken, ikinci katmanda Doğrusal Hiper-Küre sınıflandırıcısı kullanılmıştır. Son katmanda ise doğrusal sınıflandırıcılara oranla daha yavaş fakat başarımı yüksek olan Doğrusal Olmayan Hiper-Küre sınıflandırıcısı kullanılmıştır. Doğrusal Olmayan Hiper-Küre sınıflandırıcısı doğrusal olmayan DVM sınıflandırıcısından çok daha az sayıda destek vektör döndürdüğünden sistemi hızlandırmak adına bu çalışmada Doğrusal Olmayan DVM sınıflandırıcısı yerine tercih edilmiştir. Ancak bu yöntemin başarılı bir şekilde çalışması için örnek uzay boyutuna oranla çok daha fazla sayıda örneğe ihtiyaç vardır. Günümüzde internet ortamındaki resim sayısı dikkate alındığında görsel nesne gruplarına ait çok sayıda örnek toplamak mümkündür ve geliştirdiğimiz yöntem yüz sezme uygulamaları için son derece elverişlidir.

Geliştirdiğimiz yüz sezme sisteminde iyileştirme yapmak ve bu çalışmada test edilen veri tabanındaki başarı oranlarını arttırmak mümkündür. Profil yüz ile cepheden çekilmiş açılı yüzlerin ayırımı yapmak oldukça zordur. Çalışma kapsamında günlük hayattan çekilmiş web ortamında resimler kullanılarak oluşturulan test veri tabanı içerisinde ayırt edilmesi zor bu şekilde birçok imge bulunmaktadır. Bu nedenle test veri tabanının oluşturulması sırasında profil yüzler tek gözün gözükmeye prensibine göre sağ ve sol profil olarak kesilerek etiketlenmiştir. Geliştirdiğimiz sistem cepheden çekilmiş açılı (profil yüze benzer) birçok imgeyi de profil yüz olarak tespit etmiş ancak sistemin başarı oranı hesaplamasına bu imgeler dahil edilmemiştir. Ayrıca test veri tabanının hazırlanması sırasında özellikle kalabalık insan topluluklarını içeren resimlerdeki yanlış etiketlemelerden dolayı, sistemin başarı oranının tespiti sırasında yanlış olarak bulunan imgelerin bazılarının aslında gerçekten profil yüz imgesi olduğu tespit edilmiştir. Yukarıda açıklanan profil yüz-açılı cepheden yüz ayırımı ve yanlış etiketleme yaklaşık olarak başarı oranını %10 oranında etkilemektedir. Bu nedenle etiketleme işlemi sırasında cepheden çekilmiş yüzlerin belirli bir açığa kadar profil yüz olarak kabul edilmesi ve kalabalık insan topluluklarını içermeyen daha basit bir veri tabanı kullanıldığında, geliştirdiğimiz sistemin daha iyi sonuçlar vermesi beklenmektedir.

KAYNAKLAR DİZİNİ

- Ahonen, T., Hadid, A., Pietikainen, M., 2006, Face description with local binary patterns: application to face recognition, *IEEE Transactions on PAMI*, 28 (12), 2037-2041.
- Aiping, C., Lian, P., Yaobin, T., Ning, N., 2010, Second International Workshop on Education Technology and Computer Science: face detection technology based on skin color segmentation and template matching, Wuhan, China, p. 708-711.
- Anagnostopoulos, C., Anagnostopoulos, I., Vergados, D., Papaleonidopoulos, E., Kayafas, V., Stasinopoulos G., Stasinopoulos Loumos., 2002, *IEEE MELECON 2002: a probabilistic neural network for face detection on segmented skin areas based on fuzzy rules*, Cairo, Egypt, p. 493-497.
- Anijiantis, D., Dernzatas, E., Kukkinakis, G., 1999, The 6th IEEE International Conference on Electronics, Circuits and Systems: a neural network method for accurate face detection on arbitrary images, Pafos, Cyprus, p. 109-112.
- Aziz, K.A.A., Ramlee, R.A., 2009, International Conference of Soft Computing and Pattern Recognition: face detection using radial basis function neural networks with variance spread value, Malacca, Malaysia, p. 399-403.
- Bayhan, M.Ş., Gökmen, M., 2008, 23rd International Symposium on Computer and Information Sciences ISCIS'08: scale and pose invariant real-time face detection and tracking, Istanbul, Turkey, p. 1-6.
- Berber, M.A., Kelash, H.M., Kandeel, A.A., 2006, Geometric Modeling and Imaging-New Trends (GMAI'06): faces and facial features detections in color images, London, UK, p. 209-214.
- Burges, C. J. C., 1996, International Conference on Machine Learning: simplified support vector decisions.
- Chen, W., Sun, T., Yang, X., Wang, L., 2009, The Ninth International Conference on Electronic Measurement & Instruments ICEMI '2009: face detection based on half face-template, Beijing, China, p. 4-54-4-59.
- Cortes, C. ve Vapnik, V., 1995, Support-Vector network, *Machine Learning*, 20(3), 273–297.
- Çevikalp, H., Akyüz, O., Usanmaz, Güvenç., 2012, Dijital imgelerdeki nesnelerin konumlarını bulmak için yeni bir yöntem, URL: http://mlcv.ogu.edu.tr/pdf/siu_2012.pdf, erişim tarihi: 27.04.2018
- Çevikalp, H., Triggs, Bill., 2012, Efficient object detection using cascades of nearest convex model classifiers, URL: <http://mlcv.ogu.edu.tr/pdf/cvpr2012.pdf>, erişim tarihi: 27.04.2018

KAYNAKLAR DİZİNİ (devam)

- Dalal, N., Triggs, B., 2005, IEEE Society Conference on Computer Vision and Pattern Recognition (CVPR): histograms of oriented gradients for human detection.
- Felzenszwalb, P., Girshick, R.B., McAllester, D., Ramanan, D., 2010, Object detection with discriminatively trained part based models, IEEE Transactions on PAMI, 32 (9), 1627-1645.
- Fletcher, Tristan., 2008, Support vector machines explained, URL: https://cling.csd.uwo.ca/cs860/papers/SVM_Explained.pdf, erişim tarihi: 27.04.2018
- Guo, H., Yu, Y., Jia, Q., 2010, 3rd International Congress on Image and Signal Processing (CISP2010): face detection with abstract template, Yantai, China, p. 129-134.
- He, D.C., ve Wang, L., 1990, Texture unit, texture spectrum, and texture analysis, IEEE Transactions on Geoscience and Remote Sensing, 28(4), 509-512.
- Huang, D., Lin, T., Ho, C., Hu, W., 2010, Fourth International Conference on Genetic and Evolutionary Computing: face detection based on feature analysis and edge detection against skin color-like backgrounds, Shenzhen, China, p. 687-690.
- Jee, H., Lee, K., Pan, S., 2004, ISSNIP 2004: eye and face detection using svm, Melbourne, Australia, p. 577-580.
- Jin, H., Liu, Q., Lu, H., 2004, Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04): face detection using one-class-based support vectors, Seoul, Korea, p.457-462.
- Kherchaoui, S., Houacina, A., 2010, International Conference on Machine and Web Intelligence: face detection based on a model of the skin color with constraints and template matching, Algiers, Algeria, p. 469-472.
- Kun, Y., Hong, Z., Ying-jie, P., 2006, IEEE International Conference on Information Acquisition: human face detection based on SOFM neural network, Weihai, Shandong, China, p. 1253-1257.
- Li, Z., Xue, L., Tan, F., 2010, IEEE International Conference on Progress in Informatics and Computing: face detection in complex background based on skin color features and improved adaboost algorithm, Shanghai, China, p. 723-727.
- Lin, C., 2005, IEEE International Conference on Information Acquisition: face detection by color and multilayer feedforward neural network, Hong Kong and Macau, China, p.518-523.

KAYNAKLAR DİZİNİ (devam)

- Liu, Wei., Anguelov, Dragomir., Erhan, Dumitru., Szegedy, Christian., Reed, Scott., Fu, Cheng-Yang, Berg, C. Alexander., 2016, European Conference on Computer Vision: ssd: single shot multibox detector, Springer, Cham, p. 21-37.
- Liu, X., Geng, G., Wang, X., 2010, Sixth International Conference on Natural Computation (ICNC 2010): automatically face detection based on bp neural network and bayesian decision, Shandong, China, p. 1590-1594.
- Mohan, A.R., Sudha, N., 2009, IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009): fast face detection using boosted eigenfaces, Kuala Lumpur, Malaysia, p. 102-1006.
- Ojala, T., Pietikainen, M., Harwood, D., 1996, A comparative study of texture measures with classification based on featured distributions, Pattern Recognition, 29(1), 51-59.
- Öğüdücü, Gündüz, Ş., 2007, İTÜ Veri Madenciliği Ders Notları, <http://www.ninova.itu.edu.tr/tr/dersler/bilisim-enstitusu/195/bbl-606/ekkaynaklar?g8396>, erişim tarihi: 27.04. 2018.
- Peer, P., Kovac, J., Solina, F., 2003, The 2nd IEEE International Conference on Information Management and Engineering (ICIME): robust human face detection in complicated color images, Chengdu, China, p. 218-221.
- Qiang-rong, J., Hua-lan, L., 2010, The 2nd IEEE International Conference on Information Management and Engineering (ICIME): robust human face detection in complicated color images, Chengdu, China, p. 218-221.
- Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016, IEEE Conference on Computer Vision and Pattern Recognition: you only look once: unified, real-time object detection, p. 779-788.
- Ren, S., He, K., Girshick, R., Sun, J., 2015, Faster R-CNN: towards real-time object detection with region proposal networks, Advances in Neural Information Processing Systems, p.91-99.
- Ruan, J., Yin, J., 2009, International Conference on Communication Software and Networks: face detection based on facial features and linear support vector machines, Macau, China, p. 371-375.
- Sarode, N.S., Patil, A., 2013, Iris recognition using LBP with classifiers-KNN and NB, Index Copernicus Value.
- Seo, K., Kim, W., Oh, C., Lee, J., 2002, IEEE International Symposium on Industrial Electronics (ISIE 2002): face detection and facial feature extraction using color snake, L'Aquila, Italy, p.457-462.

KAYNAKLAR DİZİNİ (devam)

- Shavers, C., Li, R., Leiby, G., 2006, 38th Southeastern Symposium on System Theory: an svm-based approach to face detection, Tennessee Technological University, Cookeville, TN, USA, p. 362-366.
- Tax, D. M. J., Duin, R. P.W., 2004, Support vector data description, *Machine Learning*, 54,45-66.
- Tayyab, M., Zafar, M.F., 2009, International Conference on Emerging Technologies: face detection using 2d-discrete cosine transform and back propagation neural network, Islamabad, Pakistan, p. 35-39.
- Tsai, C.C., Cheng, W.C., Taur, J.S., Tao, C.W., 2006, IEEE International Conference on Systems, Man and Cybernetics: face detection using eigenface and neural network, Taipei, Taiwan, p. 4343-4347.
- Vapnik, V.N., 2000, *The Nature of Statistical Learning Theory*, 2. Baskı, Springer-Verlag, New York.
- Viola P., Jones M. J., 2004, Robust real-time face detection, *International Journal of Computer Vision*, 57(2), 137-154.
- Wang, J., Yang, H., 2008, Congress on Image and Signal Processing: face detection based on template matching and 2DPCA algorithm, Hainan, China, p. 575-579.
- Wang, W., Gao, Y., Hui, S.C., Leung, M.K., 2002, 9th International Conference on Neural Information Processing (ICONIP'02): a fast and robust algorithm for face detection and localization, Orchid Country Club, Singapore.
- Wang, X., Han, T.X., Yan, S., 2009, International Conference on Computer Vision: a HOG-LBP human detector with partial occlusion handling.
- Zhao, L., Sun, X., Xu, X., 2006, ICSP 2006: face detection based on facial features, Guilin, China.
- Zhengchun, Ye., Hongji, Lin., 2007, International Conference on Wavelet Analysis and Pattern Recognition: face detection based on scnn and wavelet invariant moment in color image, Beijing, China, p. 783-787.
- Zhi-fang L., Zhi-sheng Y., Jain, A.K., Yun-qiong, W., 2003, The Fifth International Conference on Computational Intelligence and Multimedia Applications (ICCIMA'03): face detection and facial feature extraction in color image, Xi'an, China, p.126-130.

EK AÇIKLAMALAR**Sayfa**

| | |
|--|----|
| Ek Açıklama-A: Yerel İkili Örüntüler (YİÖ) Matlab Kodları | 65 |
| Ek Açıklama-B: Yönlü Gradyan Histogramları (YGH) Matlab Kodları | 68 |
| Ek Açıklama-C: (YİÖ+YGH) Matlab Kodları | 71 |
| Ek Açıklama-D: Sınıflandırma Eğitimi İçin Kullanılan Matlab Kodları | 72 |



Ek Açıklama-A: Yerel İkili Örüntüler (YİÖ) Matlab Kodları

```

clear all;
close all; clc;
% Extracting LBP features of Positive face images
PositiveSamples=zeros(236,21180);
PositiveSamples2=zeros(236,21180);
mapping=getmapping(8,'u2');
rf=35;
count=0;
for i=1:21180, % 21180 is the Number of Positive Images in Database
    si=int2str(i);
    filename=['/file_name/' si '.png'];
    fid=fopen(filename);
    if fid~-1,
        im=imread(filename);
        ss=size(im);
        if length(ss)==3,
            im=rgb2gray(im);
        end
        [r,c]=size(im);
        rr=rf/r;
        im=imresize(im,rr,'bilinear');
        %immirror=im_mirror(im);
        [max(max(im)) min(min(im))]
        [rnew cnew]=size(im);
        count=count+1;
        % Working on normal image
        im = double(im);
        imfeature=mlcv_lbp(im,1,8,mapping);
        imfeature=double(imfeature);
        featvec2=mlcv_pyramid(imfeature, 59, [2,2]);
        featvec=hist_pyramidlbp(imfeature)';
    end
end

```

```

% sum(abs(feavec2-feavec))
    PositiveSamples(:,count)=feavec;
    PositiveSamples2(:,count)=feavec2;
    else
        continue;
    end
    fclose(fid);
end
save Positive_YIO PositiveSamples

sumOfPositiveSamples = sum(sum(PositiveSamples))/(count)

nAn =find(isnan(PositiveSamples)==1)

fprintf('\n Task is accomplished');

```

```

clear all;
close all; clc;

% Extracting LBP features of nonface images (negatives)
NegativeSamples=zeros(236,170000);
mapping=getmapping(8,'u2');
rf=35;
count=0;
for i=1:170000, % 170000 is the Number of Negative Images in Database
    si=int2str(i)
    filename=['/file_name/' si '.png'];
    fid=fopen(filename);
    if fid~-1,
        im=imread(filename);
        ss=size(im);
        if length(ss)==3,
            im=rgb2gray(im);
        end
    end
end

```

```

[r,c]=size(im);
rr=rf/r;
im=imresize(im,rr,'bilinear');
im = double(im);
%[max(max(im)) min(min(im))]
[rnew cnew]=size(im);
% Working on normal image
imfeature=mlcv_lbp(im,1,8,mapping);
imfeature=double(imfeature);
featvec=hist_pyramidlbp(imfeature)';
% featvec2=mlcv_pyramid(imfeature, 59, [2,2]);
count=count+1;
NegativeSamples(:,count)=featvec;
% [sum(featvec) sum(featvec2)]
% [tt]=lbp(im,1,8,mapping,"");
% result_new=hist_pyramidlbp(tt)';
% erterm(i)= max(max(featvec-result_new));
else
continue;
end
fclose(fid);
end
save Negative_YIO NegativeSamples
fprintf('\n Task is accomplished');

```

Ek Açıklama-B: Yönlü Gradyan Histogramları (YGH) Matlab Kodları

```

clear all;
close all; clc;
% Extracting HOG features of Positive face images
PositiveSamples=zeros(384,21180);
rf=35;
count = 0;
for i=1:21180, % 21180 is the Number of Positive Images in Database
    si=int2str(i);
    filename=['/file_name/' si '.png'];
    fid=fopen(filename);
    if fid~-1,
        im=imread(filename);
        ss=size(im);
        if length(ss)==3,
            im=rgb2gray(im);
        end
        [r,c]=size(im);
        rr=rf/r;
        im=imresize(im,[35 28],'bilinear');
        immirror=im_mirror(im);
        %[max(max(im)) min(min(im))]
        [rnew cnew]=size(im);
        % Working on normal image
        im1 = zeros(size(im,1),size(im,2),3);
        im1(:,:,1) = im(:,:,);
        im1(:,:,2) = im(:,:,);
        im1(:,:,3) = im(:,:,);
        im1 = double(im1);
        featvec=features(double(im1),6);
        ss=size(featvec);

```

```

if (sum(sum(sum(featvec))))>=10^-1,
    featvec=featvec*ss(1)*ss(2)/(sum(sum(sum(featvec))));
end
featvec=reshape(featvec,prod(size(featvec)),1);
count = count + 1;
PositiveSamples(:,count)=featvec;

%imgrad = mlcv_gradient_grayscale(double(im),6);
%featvec=mlcv_hog_normalized(imgrad,6);
else
continue;
end
fclose(fid);
end
save Positive_YGH PositiveSamples
sumOfPositiveSamples = sum(sum(PositiveSamples))/(count-1)
nAn =find(isnan(PositiveSamples)==1)
fprintf('\n Task is accomplished');

```

```

clear all;
close all; clc;
% Extracting HOG features of nonface images (negatives)
NegativeSamples=zeros(384,170000);
rf=35;
count = 0;
for i=1:170000, % 170000 is the Number of Negative Images in Database
    si=int2str(i);
    filename=['/file_name/' si '.png'];
    fid=fopen(filename);
    if fid~-1,
        im=imread(filename);
        ss=size(im);

```

```

if length(ss)==3,
    im=rgb2gray(im);
end
[r,c]=size(im);
rr=rf/r;
im=imresize(im,[35 28],'bilinear');
[max(max(im)) min(min(im))]
[rnew cnew]=size(im);
% Working on normal image
im1 = zeros(size(im,1),size(im,2),3);
im1(:,:,1) = im(:,:,1);
im1(:,:,2) = im(:,:,2);
im1(:,:,3) = im(:,:,3);
im1 = double(im1);
featvec=features(double(im1),6);
ss=size(featvec);
if (sum(sum(sum(featvec))))>=10^-3,
    featvec=featvec*ss(1)*ss(2)/(sum(sum(sum(featvec))));
end
featvec=featvec*ss(1)*ss(2)/(sum(sum(sum(featvec))));
featvec=reshape(featvec,prod(size(featvec)),1);
count = count + 1;
NegativeSamples(:,count)=featvec;
else
    continue;
end
fclose(fid);
end
NegativeSamples=NegativeSamples(:,1:count);
save Negative_YGH NegativeSamples
fprintf('\n Task is accomplished');

```

Ek Açıklama-C: (YİÖ+YGH) Matlab Kodları

```
clear all;
close all; clc;
load Positive_YIO % Loading obtained Positive YIO (LBP) Features
LBP=PositiveSamples;
load Positive_YGH % Loading obtained Positive YGH (HOG) Features
HOG=PositiveSamples;
PositiveSamples=[LBP;HOG];
save Positive_YIO_YGH PositiveSamples
```

```
clear all;
close all; clc;
load Negative_YIO % Loading obtained Negative YIO (LBP) Features
LBP=NegativeSamples;
load Negative_YGH % Loading obtained Negative YGH (HOG) Features
HOG=NegativeSamples;
NegativeSamples=[LBP;HOG];
save Negative_YIO_YGH NegativeSamples
```

Ek Açıklama-D: Sınıflandırma Eğitimi İçin Kullanılan Matlab Kodları

```

clear all;
close all; clc;
TrainData=[];
TrainLabels=[];
% Training Linear SVM classifier and Computing Classification Rates
nsTrain=[];
filename=['/Dosya_yolu_YIÖ+YGH/ Positive_LBP_HOG_PositiveSamples.mat'];
% Loading positive and negative samples to the matlab workspace
load(filename)
%Loading PositiveSamples
[d n1]=size(PositiveSamples);
filename=['/Dosya_yolu_YIÖ+YGH/ Negative_LBP_HOG_NegativeSamples.mat'];
load(filename)
%Loading NegativeSamples
[d n2]=size(NegativeSamples);
% Creating Training and Test Sets
TrainData=[PositiveSamples NegativeSamples];
TrainLabels=[ones(1,n1) -ones(1,n2)];
clear PositiveSamples NegativeSamples
[d,ntrain]=size(TrainData);
%[d,ntest]=size(TestData); %par_vec=[0.15 0.18 0.2 0.22 0.23 0.24 0.25 0.27 0.28]
kernel_parameter = 0.7;
[hyperplane1 hyperplane2 linear_sphere nonlinear_sphere] = train_classifiers(TrainData,
TrainLabels, kernel_parameter, 0);
clear TrainData
face_model.linear_svm = hyperplane1;
face_model.hyperplane = hyperplane2;
face_model.linear_hs = linear_sphere;
face_model.nonlinear_hs = nonlinear_sphere;
save right_profile_face_model
fprintf('\nFace Training is finished\n');

```

```

function [hyperplane hyperplane2 sphere nonlinear_sphere] = train_classifiers(TrainData,
TrainLabels, kernel_parameter, linear_flag)
if nargin<4
    linear_flag=0;
end
n1=length(find(TrainLabels==1));
n2=length(find(TrainLabels==-1));
% Training Linear SVM
svmC=1;
[W,W0,stat] = svmocas(TrainData,1,TrainLabels,svmC);
hyperplane.w=W;
hyperplane.b=W0;
ll=find(TrainLabels==1);
pos_data=TrainData(:,ll);
pos_dist=pos_data'*W;
[val ind]=sort(pos_dist,'descend');
hyperplane.b=-val(round(n1*0.96))-10^-6;
Cplus=3;
Cminus=1;
Delta=0.3;
[w2 b2] = hyperplane_fit_robust2 (TrainData,TrainLabels,Cplus,Cminus,Delta);
hyperplane2.w=w2;
hyperplane2.b=b2;
% Training Linear Hypersphere
kernel_option=[];
kernel_option.type='linear';
kernel.type=0;
sphere=find_bounding_sphere_TC_new(TrainData,TrainLabels,kernel,kernel_option,[0.2
0.1]);
for ii=1:(n1+n2),
    dd(ii)=norm(sphere.center-TrainData(:,ii));
end

```

```

[val ind]=sort(dd(1:n1));
sphere.radius_original=sphere.radius;
sphere.radius=val(round(n1*0.96));
estimated_labels=-ones(1,n1+n2);
ll=find(dd<=sphere.radius);
estimated_labels(ll)=1;
classification_acuracy=length(find(estimated_labels==TrainLabels))*100/(n1+n2);
if linear_flag == 0
    % Training Nonlinear Hypersphere
    kernel_option=[];
    kernel_option.type='gaussian';
    kernel_option.par=kernel_parameter;
    kernel.type=2;
    kernel.gamma=1/(2*(kernel_option.par)^2);
    nonlinear_sphere =
    train_nonlinear_hs_classifier(TrainData,TrainLabels,kernel,kernel_option,[0.20 0.10]);
else
    nonlinear_sphere=[];
end
%Forming classifier_model from model after training
load right_profile_face_model
classifier_model = construct_new_model (face_model)
% classifier_model =
%   LSVM: {[1x1 struct]}
%   LHS: [1x1 struct]
%   NHS: [1x1 struct]
%   mapping: [1x1 struct]
save right_profile_tez classifier_model

```