

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

ARAÇ RENK TANIMA SİSTEMİ

**YÜKSEK LİSANS TEZİ
Erida DULE**

Anabilim Dalı : Bilgisayar Bilimleri

Programı : Bilgisayar Bilimleri

Tez Danışmanı: Prof. Dr. Muhittin GÖKMEN

MAYIS 2010

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ BİLİŞİM ENSTİTÜSÜ

ARAÇ RENK TANIMA SİSTEMİ

YÜKSEK LİSANS TEZİ

Erida DULE

(704061025)

Tezin Enstitüye Verildiği Tarih : 07 Mayıs 2010

Tezin Savunulduğu Tarih : 10 Haziran 2010

Tez Danışmanı : Prof. Dr. Muhittin GÖKMEN (İTÜ)

Eş Danışman :

**Diğer Jüri Üyeleri : Prof. Dr. Coşkun SÖNMEZ (YTÜ)
Yrd.Doç.Dr. Mustafa Ersel KAMAŞAK
(İTÜ)**

MAYIS 2010

Sevgili Aileme,

ÖNSÖZ

Öncelikle, bu çalışma sırasında yardımlarını ve desteklerini benden esirgemeyen tüm sevdiklerime çok teşekkür ediyorum.

Danışmanım ve sevgili hocam Prof. Dr. Muhittin Gökmen'e, yüksek lisans eğitimim boyunca verdiği destekler ve yardımlar ile çok değerli tavsiyeleri ve yönlendirmeleri için teşekkürü bir borç bilirim. Projeye katkılarında dolayı fikirlerini benden esirgemeyen Abdülkerim Çapar ve M. Sebul Beratoğlu'na çok teşekkür ederim. Ayrıca, hocam Yrd. Doç. Dr. Turgay Altılar'a yol göstericiliği, ilgisi ve yardımlarından dolayı minnettarım.

Bu zorlu süreçte her anlamda yanımda olan "bizim tayfa" mensuplarına, Oky'e, Aslıhan'a, kardeşlerim Jerina, Denisa ve Tunç'a, ev arkadaşlarıma ve tüm dostlarıma teşekkür ediyorum.

Ve son olarak hayatım boyunca yanımda olan aileme sonsuz sevgilerinden ve desteklerinden dolayı teşekkür ederim.

Mayıs 2010

Erida Dule

İÇİNDEKİLER

Sayfa

ÖNSÖZ.....	v
İÇİNDEKİLER	vii
KISALTMALAR	ix
ÇİZELGE LİSTESİ.....	xi
ŞEKİL LİSTESİ.....	xiii
ÖZET.....	xv
SUMMARY	xvii
1. GİRİŞ	1
1.1 Daha Önceki Çalışmalar.....	2
1.2 Tezin Organizasyonu.....	4
2. ARAÇ RENK TANIMA VE VERİ KÜMESİ	5
2.1 Araç Renk Tanıma Sistemi	5
2.2 Veri Kümesi	5
3. EN UYGUN KARE SEÇİMİ (EUKS).....	7
4. RENK TANIMA BÖLGESİNİN (ROI) BELİRLENMESİ	9
4.1 Kaput Parçası ROI.....	9
4.2 Yarım Araç ROI.....	11
5. ÖZNETELİK SEÇİMİ VE ÖZNETELİK VEKTÖRÜ OLUŞTURMA	13
5.1 Kullanılan Renk Bileşenleri	13
5.1.1 RGB renk uzayı.....	13
5.1.2 Normalize RGB renk uzayı.....	13
5.1.3 HSV renk uzayı.....	14
5.1.4 L*a*b renk uzayı.....	15
5.1.5 L-color renk uzayı.....	16
5.1.6 Otha renk uzayı	18
5.1.7 Kromatit renk uzayı	18
5.2 Öznitelik Vektörü Oluşturma Yöntemleri.....	20
5.2.1 Histogram tabanlı öznitelik oluşturulma yöntemi.....	20
5.2.2 Ekli Histogram tabanlı öznitelik oluşturulma yöntemi	21
5.2.3 Piksel tabanlı öznitelik oluşturulma yöntemi.....	22
5.2.3.1 Orta değer-piksel öznitelik vektörü oluşturma yöntemi	23
5.2.3.2 Çoğul-piksel öznitelik vektörü oluşturma yöntemi	23
6. SINIFLANDIRMA YÖNTEMLERİ.....	25
6.1 K En Yakın Komşu (K Nearest Neighbour : K-NN)	25
6.2 Şablon Eşleme (Template Matching : TM).....	26
6.3 Yapay Sinir Ağları (Artificial Neural Networks - ANN).....	26
6.4 Destek Vektör Makineleri (Support Vector Machine – SVM)	28
6.5 Karar Ağacı Oluşturma	30
6.5.1 Tahmini yaklaşım.....	31
6.5.2 Hesaplamalı yaklaşım	32

7. YAPILAN TESTLER VE SONUÇLARI.....	35
7.1 Renk Tanıma Başarısını Düşüren Sebepler	36
7.2 En İyi Öznitelik Vektörünün Elde Edilmesi İçin Yapılan Testler.....	36
7.2.1 Piksel tabanlı öznitelik vektörü testleri	36
7.2.2 Histogram tabanlı öznitelik vektörü testleri	40
7.2.3 Önceki çalışmalarla en iyi öznitelik vektörünü karşılaştırma	46
7.3 Tahmini Ağaç Oluşturma ve Test Sonuçları	47
7.3.1 Orta Değer Ağacı test sonuçları	49
7.3.2 Histogram Ağacı test sonuçları	51
7.3.3 Ekli Histogram Ağacı test sonuçları.....	54
7.4 Hesaplamalı Karar Ağacı Öğrenmesi ve Bazı Testler.....	56
8. SONUÇLAR VE İLERİKİ ÇALIŞMALAR.....	63
KAYNAKLAR.....	65
ÖZGEÇMİŞ.....	69

KISALTMALAR

ANN	: Artificial Neural Network (Yapay Sinir Ağları)
ART	: Araç Renk Tanıma
ETS	: Entegre Trafik Sistemi
EUKS	: En Uygun Kare Seçme algoritması
HSV	: Hue Saturation Value renk uzayı
I1I2I3	: Kanade'nin makalesindeki renk uzayı
ID3	: Iterative Dichotomiser 3
K-NN	: K Nearest Neighbours (K En Yakın Komşu)
L-color	: L renk uzayı
L1L2L3	: L renk uzayı bileşenleri
RGB	: Red Green Blue (Kırmızı Yeşil Mavi)
ROI	: Region of Interest (seçilen bölge)
TM	: Template Matching (Şablon Eşleme)
SVM	: Support Vector Machine
WTS	: Weights of Synapses (bağlantı ağırlıkları)

ÇİZELGE LİSTESİ

Sayfa

Çizelge 7.1 : Yanlış sınıflandırmanın sebepleri.....	36
Çizelge 7.2 : TM sınıflandırıcısı için en iyi Piksel Tabanlı Öznitelik Vektör sonuçları.....	37
Çizelge 7.3 : TM ve KNN sınıflandırıcıları için en iyi Piksel Tabanlı Öznitelik Vektörü.....	38
Çizelge 7.4 : KNN sınıflandırıcısında k değıştikçe farklı renk uzaylarının başarı oranları.....	38
Çizelge 7.5 : ANN sınıflandırıcısı için en iyi Piksel Tabanlı Öznitelik Vektörü.....	40
Çizelge 7.6 : Farklı boyut normalizasyonu, birim vektör ve bin sayısı parametreleri TM ve ANN sınıflandırıcıları için başarı sonuçları.....	41
Çizelge 7.7 : TM- 8 ve 32 binlik histogram tabanlı öznitelik vektörleri sonuçları Norm: Normalizasyon.....	44
Çizelge 7.8 : ANN: 8, 16 ve 32 binlik histogram tabanlı öznitelik vektörleri sonuçları.....	45
Çizelge 7.9 : ANN: en iyi 5 tane histogram tabanlı öznitelik vektörü ve parametreleri.....	46
Çizelge 7.10 : Histogram özniteliđi ile 5 renkli veri tabanı karşılaştırmalı sonuçları.....	47
Çizelge 7.11 : Orta değer ağacı için renk tanıma sonuçları.....	50
Çizelge 7.12 : Histogram ağacı için renk tanıma başarı sonuçları.....	53
Çizelge 7.13 : Ekli histogram ağacı için renk tanıma başarı sonuçları.....	55
Çizelge 7.14 : Kırmızı- Diğer eğitim kümesi karıştırma matrisi.....	57
Çizelge 7.15 : Kırmızı - Diğer Test kümesi karıştırma matrisi.....	57
Çizelge 7.16 : Bir rengi diğerlerinden en iyi ayıran öznitelik vektörleri.....	58
Çizelge 7.17 : Eğitim kümesi ANN sınıflandırıcısı için karıştırma matrisi.....	60
Çizelge 7.18 : Eğitim kümesi TM sınıflandırıcısı için karıştırma matrisi.....	61

ŞEKİL LİSTESİ

Sayfa

Şekil 2.1 : ETS sisteminin akış diyagramı.....	5
Şekil 4.1 : Kaput Parçası ROI.....	11
Şekil 4.2 : Yarım Araç ROI.....	11
Şekil 5.1 : HSV renk uzayının konik gösterimi.....	14
Şekil 5.2 : İki boyutlu histogramın tek boyutlu öznitelik vektörüne çevrilmesi.	21
Şekil 5.3 : Ekli histogram öznitelik vektörü için oluşturulan üst ve alt bölgeler.....	22
Şekil 6.1 : Çok katmanlı Yapay Sinir Ağı.	27
Şekil 6.2 : Destek Vektör Makineleri - Hiperdüzlemlerle sınıflandırma.....	28
Şekil 6.3 : Tahmini yaklaşım ile oluşturulan karar ağacı.	31
Şekil 7.1 : k değıştikçe KNN modelinin başarısını gösteren grafik.....	39
Şekil 7.2 : Pksel öznitelik vektörleri ve her düğümdeki sınıflandırma başarıları.....	49
Şekil 7.3 : Histogram öznitelik vektörleri ve her düğümdeki sınıflandırma başarıları.....	52
Şekil 7.4 : Ekli histogram öznitelik vektörleri ve her düğümdeki sınıflandırma başarıları	54

ARAÇ RENGİ TANIMA SİSTEMİ

ÖZET

Son yıllarda, teknolojinin gelişmesiyle uzaktan algılama ve tanıma sistemleri hayatımızın her alanına girmeye başladı. Çoklu ortamların giderek yaygınlaşması ve ucuzlamasıyla birlikte renk tanıma ile ilgili çalışmalar önem kazanmıştır. Çünkü, renk, resim ve video gibi görsel öğelerde nesnelere ayırt etmek için önemli bir özelliktir. Ancak, renk ışık kaynağının açısına ve şiddetine, gölgeye, hava koşullarına ve kameranın bakış açısına göre değiştiği için görüntüleri gösteren sabit bir eleman değildir. Bu nedenle dış ortam görüntülerinde yansıyan yüzeyli nesnelere renklerinin tanınması zor ve ilgi çekici bir problemdir.

Bu çalışmada dış ortamda ve hareket halindeki araçların renklerini tanıyan gerçek zamanlı bir sistem geliştirilmeye çalışılmıştır. Sistemi gerçekleştirmek için farklı renk tanıma bölgeleri, farklı öznitelikler ve farklı sınıflandırma yöntemleri kullanılmış veya geliştirilmiş, aralarındaki en iyi kombinasyon bulunmaya çalışılmıştır. Sistem yedi farklı rengi tanımak için tasarlanmıştır: *beyaz, gri, kırmızı, mavi, sarı, siyah ve yeşil*.

Araç görüntülerinden meydana gelen veri tabanının oluşturulması için otoban kenarına kameralı bir bilgisayar düzeneği kurulmuş ve yoldan geçen hareket halindeki araçların görüntüleri çekilmiştir. Bu video görüntülerinin her bir karesi plaka tanıma sistemi kullanılarak elekten geçirilmekte ve plakalı araçların olduğu görüntüler seçilmektedir. Böylece görüntülerdeki cismin bir araç olduğu %99 oranında saptanmış olmaktadır. Renk tanıma sistemi bu işlemde sonra devreye girmektedir.

İlk olarak, aynı araca ait peş peşe görüntülerden araç renk tanıma (ART) sisteminde kullanılacak olan görüntüye “Uygun Kare Seçimi” algoritması ile karar verilmektedir. Uygun kare seçimi yapıldıktan sonra, seçilen görüntüdeki aracın renginin belirlenmesi için uygun bölgenin (ROI : Region of Interest) seçilmesi gerekmektedir. ROI seçimi için üç yöntem denemiş, iki yöntemde karar kılınmıştır: kaput parçası ROI ve yarım araç ROI. Kaput parçası ROI betik (piksel) seçerek oluşturulan öznitelik vektörleri için, yarım araç ROI ise histogram tabanlı öznitelik vektörlerinin oluşturulması için daha uygun yöntemlerdir.

Bu çalışmada üç farklı öznitelik vektörü oluşturma yöntemi geliştirilmiştir. Bunlar histogram tabanlı öznitelik vektörü, ekli histogram tabanlı öznitelik vektörü ve piksel tabanlı öznitelik vektörü oluşturma yöntemleridir. Öznitelik vektörleri, renklerin ifade edildiği farklı renk uzayı bileşenlerinden oluşturulmaktadır. Piksel tabanlı yöntemde her bir öznitelik, rengin sayısal ortamda ifade edildiği bileşenleri cinsinden (RGB) ve bu bileşenlerin normalize rgb, HSV, L*a*b, L-color gibi farklı renk uzaylarına çevrilmesiyle elde edilen sayısal değerini ifade etmektedir. Histogram tabanlı öznitelik oluşturma tekniğinde ise, renk uzayı bileşenlerinden bir ve iki boyutlu normalize histogramlar oluşturulmaktadır. Öznitelik vektörleri ise bir yada birden çok histogramın art arda sıralanması ile elde edilmektedirler.

Araç renk tanıma sisteminde son olarak öznitelik vektörlerinin sınıflandırması algoritması yer almaktadır. Bu aşamada Şablon Eşleştirme (TM), K En Yakın Koşu (K-NN), Yapay Sinir Ağları (ANN) ve Destek Vektör Makineleri (SVM) sınıflandırma modelleri araştırılmış, farklı koşullarda yapılan testler sonucunda en iyi performansı veren sınıflandırma yöntemi seçilmiştir.

Sınıflandırma aşamasında bir ileri adım olarak Karar Ağacı oluşturulmuştur. Karar ağacı oluşturulurken iki yaklaşım izlenmiştir: Biri tahmini yaklaşım ile ağaç oluşturma diğeri ise hesaplamalı yaklaşım ile eğitim kümesi kullanılarak Iterative Dichotomiser 3 (ID3) algoritması yardımıyla karar ağacının oluşturulması. Tahmini karar ağacı yönteminde ağacı oluştururken izlenen yaklaşım, renkleri en farklı özelliklerine göre ayırmak olarak söylenebilir. Ağacın kökünden başlanarak her karar düğümünde her sınıflandırma yönteminin, her bir öznitelik vektörü oluşturma yöntemi ve her bir öznitelik vektörü için sınıflandırma oraları hesaplandı. Testler sonucunda yukarıda sıralanan tüm yöntemlerin birbirleriyle nasıl çalıştıkları ve performansları elde edilmiş oldu.

Yapılan testler sonucunda piksel tabanlı öznitelik vektörleri kullanılarak elde edilen en yüksek başarı oranı ANN sınıflandırıcısı ve “*Rn Gn H S L b I1 I2 L1 L3 ç1*” öznitelik vektörü ile %82.1 olarak bulunmuştur. Histogram tabanlı öznitelik vektörlerinin sınıflandırılmalarıyla yapılan testlerde de yine ANN modeli diğer sınıflandırıcılardan daha iyi performans sağlamış ve “*RnGn SH Gray1 L-b L1-L2 L1-L3*” histogram öznitelik vektörü yarım araç ROI’de %83.89 başarı oranını yakalamıştır. Oluşturulan histogramların normalize edilmeleri ve bin sayısındaki değişiklik renk tanıma performansını önemli ölçüde arttırdığı, öznitelik vektörünün birim vektör haline getirilmesinin ise renk tanıma performansını değiştirmedığı gözlenmiştir.

Tahmini karar ağacının oluşturulmasıyla yapılan demeler sonucunda öznitelik vektörlerinin türüne göre oluşturulan üç farklı karar ağaçlarının başarı oranları sırasıyla histogram ağacı (histogram öznitelik vektörlerin sınıflandırılması), ekli histogram ağacı ve orta değer ağacının (piksel öznitelik vektörlerinin sınıflandırılması) performansları olarak sıralanmaktadır. Karar ağaçlarının her düğümünde en iyi renk tanıma sonucunun veren farklı öznitelik vektörü ve sınıflandırma modeli çiftleri elde edilmiş, histogram ağacında kök düğümünde $R \neq G \neq B$ (renkli) ve $R=G=B$ (renksiz) olarak ayrıştırılan örnekleri en iyi ayıran öznitelik vektörü ve sınıflandırıcı çifti “*RnGn SV Gray1 Gray2 a-b L1-L2 L1-L3*” : TM iken, $R \neq G \neq B$ düğümünde örnekleri kırmızı, sarı ve mavi-yeşil renk sınıflarına en iyi ayıran öznitelik ve sınıflandırıcı çifti ise “*RnGn SH Gray2 a-b L-a L1-L3*” ve ANN modelidir. Histogram tahmini ağacı ile sınıflandırma yapılırsa sistemin toplam başarısı %87,828 olarak elde edilmektedir.

Farklı sınıflandırma modeli ve öznitelik vektörleri ile oluşturulan tahmin ağaç sınıflandırma yöntemi daha yüksek başarı oranları vermesine rağmen, gerçek zamanlı bir sistemde oluşan zaman kısıtı nedeniyle tercih edilmeyebilir. Bu durumda histogram tabanlı öznitelik vektörü “*RnGn SH Gray1 L-b L1-L2 L1-L3*”, yarım araç ROI veri kümesi kullanılarak ANN sınıflandırıcısı araçların renk tanıma işlemi için seçilmelidir.

VEHICLE COLOR RECOGNITION SYSTEM

SUMMARY

As technology has improved, remote detection and recognition systems have become a part of our lives in last years. As long as multimedia environments have widespread and become less expensive, studies about color recognition issue have gained in importance. It is wellknown that the color is a property, which makes objects distinguishable in visual products, such as photos and videos. Despite of this; it varies according to the shadow, the power and the position of the light, the weather conditions and the perspective of the camera. Therefore, color is not a stable entity, which displays outdoor images. Since the color varies especially outdoor and reflexive surfaces, it is not as easy as it is assumed to distinguish outdoor objects' color.

In this study, it is aimed to develop a real time system, which identifies the color of moving and outdoor objects. In order to implement the best performed system, different region of interest (ROI), feature selection, feature vector construction, and classification methods are utilized and their success rates are tested. Moreover, it is attempted to find the best combination of all of these techniques. The system has designed to recognize seven colors: *white, grey, red, blue, yellow, black, and green.*

The digital moving car images in the database are retrieved from a mechanism containing a computer and a video camera, which has been settled near the motorway. Every frame in the video is passed trough a sieve using plate recognition system and car images including their plates are chosen. By means of the plate, the object in the images is determined 99% as a car. As soon as plate recognition process ends, vehicle color recognition system begins to proceed using the plate as start reference.

Appropriate Frame Selection algorithm chooses the most suitable image among numerous frames of the same car, which sieved before by plate recognition system. After the frame decision made, a colorful ROI is needed to be selected over the car image. There are integrated two techniques of ROI selection. One method is "hood piece ROI" which is used with pixel based feature selection, and the other one is "half ROI" for histogram based feature selection.

In this work, two different feature vector construction methods are implemented. The first one is histogram based feature vector construction method; the second one is pixel based feature vector construction method whose feature selection is made in two different ways. Features in the feature vector can be either the color values of the median pixel in the ROI image or just an ordinary pixel taken inside the ROI. The features are numerical color component values which are obtained from the transformation of RGB color space to different color spaces such as normalized RGB, HSV, L*a*b, L-color. In the histogram based method, feature vectors are one or two dimensional histograms which are formed over the selected ROI using different color components. On the other hand, in the pixel based method, what

forms the feature vector is the alignment of different color component values derived from RGB values a single pixel.

In the classification step of this study, K Nearest Neighbor (K-NN), Artificial Neural Network (ANN) and Support Vector Machines (SVM) models are used for searching the best performances of various method combinations. Furthermore, an answer of the question “Can a greater success be possible by integrating different circumstances that give the highest 10 successes which are applied by Classifying Neural Network?” is investigated by applying different weights. Eventually, this method is not considered as successful as it was expected to be. Subsequently, it is constructed a decision tree structure which is obtained with approximate approaches. Here also, at each branch of the tree, it is searched for the best performed combination of classification method, feature vector construction, ROI and feature selection methods.

1. GİRİŞ

Günümüzde dış alanlarda, yollarda, caddelerde, merkezi yerlerde kameralı güvenlik sistemlerine çokça rastlamaktayız. Bu sistemlerin kullanım amaçlarından bir tanesi trafikte giden araçların kimliklerini saptamaktır. Böylece oluşabilecek bir güvenlik tehdidinde araç tespit edilerek gerekli birimlere iletilmektedir. Kameralı trafik güvenlik sistemleri trafik suçlarını ve suçlularını tespit etmek amacıyla da yaygın olarak kullanılmaktadır. Özellikle hırsızlık, kaçakçılık gibi suç olaylarında insanlar araçlarına sahte plaka takma veya araçlarının renklerini değiştirme gibi yollara başvurabiliyorlar. Dolayısıyla araçların eşgalini verirken renk bilgisi de çok önemlidir. Bu nedenle araçların renklerinin tanınması o anda geçen aracın hangi renk olduğunun bilinmesine ve bu gibi olayların açığa çıkarılmasında büyük önem taşımaktadır.

Renk değişken bir elemandır. Işık kaynağının şiddetine, yönüne, yüzeyin yansıtma katsayısına, gölgeye bağlı olarak değişir. Dış ortamdan elde edilen görüntülerde aynı cisim günün farklı saatlerinde farklı renkte görünebilir. Üzerine güneş geldiyse parlak, gölge geldiyse koyu görünebilir. Hava kapalıysa koyu, açık ise açık renkte görünebilir. Araçlar da dışarıda ve hareket halinde oldukları için elde edilen görüntülerde renkleri değişiklik gösterecektir. O nedenle araçlarda renk tanıma olayı düşünüldüğü kadar kolay değildir. Bu alanda yapılan daha önceki çalışmaların sayısının azlığının bir nedeni de budur.

Araçlarda renk tanıma (ART) üzerine daha önce yapılmış çalışmalarda gerçekçi veriler yerine daha düzgün ve özenle seçilmiş, güneşli havada çekilmiş araç resimleri kullanılmış olduğu için çıkan sonuçlar gerçek hayata çok uygun değil. Bu çalışmada ise, görüntü veri tabanı otoban kenarına kurulan kameralı system aracılığıyla yoldan geçen araçlardan günün farklı saatlerinde ve farklı hava koşullarında çekilen görüntülerden elde edilmiştir. Veri tabanı sistemin kullanılacağı ortamdan alındığı için çalışmada alınan sonuçlar ile ART sistemi hayata geçirildiğinde elde edilecek sonuçlar çok farklı olmayacaktır.

ART sistemi Plaka Tanıma ve Araç Marka Tanıma sistemlerinin de aralarında bulunduğu gerçek zamanlı bir sistemin parçası olarak tasarlanmıştır. O nedenle renk tanıma sisteminde kullanılacak algoritmalar görece hızlı olmalıdır. Geliştirilen algoritmalarda hem hız hem de doğruluk payı performanslarına dikkat edilmiştir.

Bu çalışmada amaç, renkli bir görüntüden aracın rengini bulup seçilmiş yedi renk sınıfından birine karar vermek için en iyi sonucu veren ROI, öznitelik vektörü oluşturma tekniği, öznitelik seçimi ve sınıflandırma yöntemini bulmaktır. Çeşitli renk bileşenlerinden (HSV, normalize RGB, L^*a^*b , ks_1 ve ks_2 , ... vs.) oluşan farklı öznitelik kümelerinin resim üzerinde farklı bölge seçimi ve farklı sınıflandırıcı yöntemleriyle (K-NN, Yapay Sinir Ağları, SVM, karar ağacı) kullanıldıklarında nasıl sonuç verdikleri test edilmiştir. Yapılan çalışmada dış ortamın araba rengi üzerindeki etkisini azaltmak için en uygun kare bulma algoritması kullanılarak peş peşe gelen aynı araca ait karelerden en uygunu seçilmektedir.

1.1 Daha Önceki Çalışmalar

Gerçek zamanlı kamera kullanımına dayalı takip ve izleme sistemlerinin giderek yaygınlaşmasıyla birlikte *araç rengi tanımlama* konusu da daha önemli bir hal almıştır. Bu tür uygulamalarda araç renklerini birbirinden ayırt etmek çok önemli olduğundan araç renklerinin sınıflandırılması çalışmaları da dolaylı olarak önemini korumaktadır.

Literatüre baktığımız zaman renk tanıma, görüntü işleme ve kapalı veya açık alandaki imgelerin tanımlanması konusunda pek çok araştırma göze çarpmaktadır. Ne var ki araç rengi tanıma alanında yapılan çalışma sayısı gayet kısıtlıdır. Ayrıca, yapılan bu çalışmaların birçoğu sadece görüntüleri türlerine göre sınıflandırmaya yoğunlaşmış ve dolayısıyla bakış açısı, ışıklandırma ve de ölçek çeşitliklerinden doğan farklılıklar ile ilgilenmemektedirler [1]. Bununla birlikte imgelerde oluşan gürültüleri önlemeyi amaçlayan çalışmalar da mevcuttur [2]. Bazı yaklaşımlar, peş peşe gelen görüntülerin üzerindeki renk farklılıklarını kullanarak çevredeki değişimleri algılamaya çalışmışlardır. Örneğin, bir yolda hareket halindeki araçları belirlemek için yolun renginin referans alınarak yol rengi dışında bir renge sahip araçlar ayırt edilmek istenmiştir [3]. Bir başka örnek ise, aynı yeri gösteren görüntülerde hareket halindeki objeler, aynı alan üzerindeki aydınlık ve gölgelenmiş bölgelerdeki farklılıklara bakılarak ayırt edilmektedirler [4].

Diğer belli başlı arařtırmalar ise kapalı alandaki görüntüler üzerinde çalışarak sentetik ve tam mat (lambertian) yüzeylerdeki deneymelerle ilgilenirken, bu çalışmanın kapsamı içerisinde yer alan yüzey yansımaları olgusunu göz ardı ederler.

B.D. Zari'tin arařtırmasında yine renk histogramları, bu kez, beş farklı renk uzayı üzerinde kullanılmıştır. Bu çalışma, renk uzaylarını ve iki farklı sınıflandırma metodunu Bayes yoğunluk sınıflandırması teoremi ile başvuru çizelgesi (*look up table*) yöntemlerini karşılařtırdığı için [5] bizim arařtırmamız açısından önemlidir. Baktığımız zaman, çalışmamıza en benzer arařtırmalardan biri de açık alandaki araçların renklerini altı farklı kategoride inceleyen Hsegawa ve Kanade nin arařtırmasıdır. Bu arařtırmada, renk grupları tek başına renklerden oluşmayıp, koyu mavi, siyah ve koyu gri gibi benzer renklerin aynı grupta birleřtirilmesiyle oluşmaktadır. Bu yaklaşım sistemin başarı oranını arttırmasına rağmen, tam bir araç rengi vermediği için soruna tam bir yanıt sağlayamamaktadır. RGB renk uzayından lineer olarak çevrilmiş bir renk uzayını (I1 I2 I3) kullanan Hsegawa ve Kanade, aynı zamanda K-NN sınıflandırıcısına benzer bir sınıflayıcı olarak kullanırlar. Kanade sınıflandırma yöntemi olarak adlandırdığımız bu sınıflandırıcı, en yakın k elemanların çoğunun uzaklığına bakmak yerine, k en yakın elemanının her sınıf için ayrı ortalamasını alıp bunu bir seçme kriteri olarak kullanır [6].

Merler ise tanımladığı renklerin dağılımlarını göstermek için iki renk uzayı tanımlar, normalleřtirilmiş RG ve CbCr. Onun yönteminde ise KNN sınıflandırıcısı ve bir ön işleme adımı olarak imge bölütleme (segmentasyon) algoritması kullanılır [7]. Renk uzaylarının farklı betimlemesinin haricinde bu çalışmada bizim ilgimizi çekebilecek araç tanıma problemiyle bağlantılı başka bir nitelik bulunmamaktadır.

Araç renk tanıma temasıyla ilintili bahsedeceğimiz son çalışma ise HSV renk uzayından faydalanan Beak ve Park'ın arařtırmasıdır. Bu yaklaşımda, araç resimlerinin iki boyutlu normalleřtirilmiş H-S histogramı oluşturup SVM ile sınıflandırırlar. Araç renklerini kırmızı, mavi, siyah, beyaz ve sarı olmak üzere beş farklı renk sınıfında toplar. Renkler sınıflandırılırken en çok karışıklığa sebep olan gri rengini gerçek görüntülerdeki ışık ve gölgeler sebebiyle sınıflandırma dışı tutmuşlardır [8]. Bizim çalışmamızda ise, burada kullanılan beş rengin yanı sıra gri ve yeşil renk için de birer sınıf oluşturulup ideal performansa ulařılmaya çalışılmıştır.

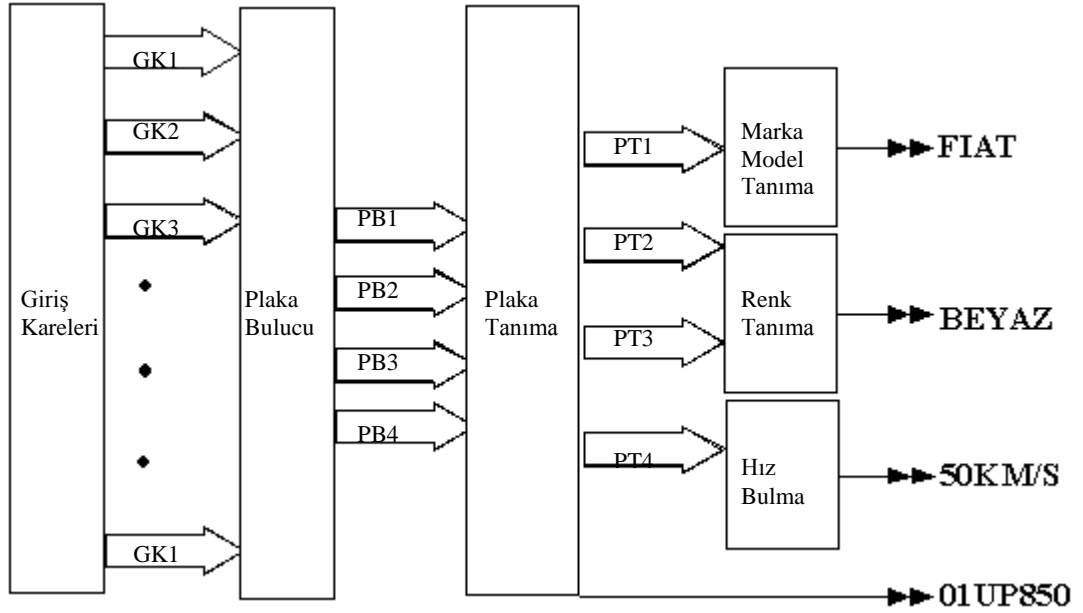
1.2 Tezin Organizasyonu

Bu bilimsel incelemede, Araç Renk Tanıma sisteminin ana hatları belirtilmiş ve kullanılan algoritmalar, yöntemler, yapılan testler ayrıntılı olarak açıklanmıştır. Bölüm 2’de ART’nin genel yapısı ve veri kümesi anlatılmaktadır. Aracın rengini etkileyen dış etkenleri en aza indirmek için kullanılan en uygun kare seçme (EUKS) algoritması Bölüm 3’te verilmektedir. Bundan sonra Bölüm 4’te araç üzerinde en uygun renk tanıma alanının (ROI) bulunmasına ilişkin yöntemler aktarılmaktadır. 5. Bölüm geliştirilen farklı öznelik vektörü oluşturulma yöntemlerini açıklamaktadır. Kullanılan sınıflandırma yöntemleri hakkında bilgiler ve nasıl kullanıldıkları Bölüm 6’da verilmektedir. Bölüm 7’de yapılan testleri ve başarı sonuçlarının verilir önceki çalışmalarla karşılaştırmaların yapıldığı bölümdür. Son olarak da 8. Bölüm’de tez sonuçlandırılmıştır.

2. ARAÇ RENK TANIMA VE VERİ KÜMESİ

2.1 Araç Renk Tanıma Sistemi

Araç renk tanıma sistemi Entegre Trafik Sisteminin (ETS) bir modülü olarak tasarlanmıştır. ETS plaka tanıma, araç marka tanıma, hız bulucu ve araç renk tanıma sistemlerinden oluşmaktadır [9]. ART sisteminin çalışması için plakası görünen araç görüntüsü, plaka koordinatlarının bulunduğu bir dosya ve sınıflandırmanın yapılacağı renk sınıflarının adlarının olduğu cls uzantılı bir dosya gerekmektedir. ART sistemi çıkış olarak yakaladığı aracın rengini cls uzantılı dosyada bulunan yedi renk sınıfından birini vermektedir. Bu renkler: beyaz, gri, kırmızı, mavi, sarı, siyah ve yeşildir. Şekil 2.1 ETS sistemine genel bir bakış atmamıza yarayacaktır.



Şekil 2.1 : ETS sisteminin akış diyagramı

2.2 Veri Kümesi

Renk tanıma sisteminde kullanılan veri kümesi otoban kenarına kurulan bir kameralı sistemden elde edilmektedir.

Video olarak alınan veri plaka bulucu ve plaka tanıma modülleri sayesinde karelere ayrılıp plakasız resimler ayıklandıktan sonra aynı araca ait olan resimler belirlenerek ART modülünün giriş parametreleri olarak kaydedilmektedir. Çekimler birkaç farklı yerde, günün farklı saatlerinde ve farklı hava şartlarında yapılmıştır. Veri kümesinde araba, otobüs ve kamyonet gibi birden fazla çeşit, plakalı araç görüntüleri bulunmaktadır. Resimlerin plakalı olmalarının sebebi, ROI seçerken plaka koordinatlarının referans alınmasıdır.

Veri kümesi eğitim ve test kümesi olmak üzere ikiye ayrılmıştır. Her renk sınıfı için yaklaşık 180 eğitim görüntüsü bulunmaktadır. Test kümelerinden bir tanesi her renk sınıfı için eşit sayıda (100'er) 700 resimden, diğeri de araç renklerinin gerçek hayatta karşılaşıma oranına göre ayarlanıp toplam 1000 görüntüden oluşmaktadır. Örneğin, beyaz arabaların oranı %30 olduğu için test kümesinde beyaz araçların karşılığı 300 görüntü iken %10 oranında olan kırmızı araçların karşılığı ise 100 test görüntüsüdür. Ayrıca eğitim ve test kümeleri her bir farklı ROI seçimi için de hazırlanmıştır. İlk eğitim-test kümesi çifti tüm araç görüntüsü, ikinci çift yarım araç görüntüsü, son küme çifti ise araç kaputunun renkli bölgesinden alınmış renkli resim parçalarını içermektedir.

3. EN UYGUN KARE SEÇİMİ (EUKS)

Bir aracın rengini tanımak için o araca ait peş peşe gelen kamera karelerinden en uygun olanını seçme işlemini kapsamaktadır. Araç görüntüye girdiği anda ETS aynı araca ait görüntüleri kaydeder ve dizinlerini de v1 formatlı bir dosyaya yazar. Renk tanıma programı çalıştırıldığında bu dosyayı okur, görüntüleri aldıktan sonra ilk olarak “frame” seçme algoritmasını çalıştırır.

En uygun görüntü karesini seçerken art arda gelen iki kare arasındaki ışık şiddeti farkı hesaplanır. Eğer arada eşik değerinden büyük bir fark varsa ilk görüntü elenir. Araca ait her kareye aynı işlemler uygulanmakta ve iki kare arasındaki fark eşik değerinin altında olduğunda kareler aday görüntüler olarak seçilir. Böylelikle kameradan veya ortamdan kaynaklanan bazı ani ışık değişimlerinden programımızı soyutlamış oluyoruz. Bunun yanında, daha iyi sonuç almak için, görüntü içindeki aracın yeri de önem taşımaktadır. Bir araca ait birden fazla kare seçilmişse aracın mümkün olduğunca yakın, plakayı da içeren bir görüntüsünün seçilmesi gerekmektedir. Böylece rengin tanınacağı bölge renk açısından daha belirgin (renk) ve daha büyük olur. Kullanılan EUKS algoritması aşağıda verilmiştir.

Method LoadGoodFrames(PathImageList)

Begin

```
vFrame = new vector();
```

```
For i = 0 to size of vPathImage do
```

```
    pathRec2File = changeFileExt(PathImageList[i], "rec2");
```

```
    TFrame pFrame = new TFrame;
```

```
    pFrame.pathImage = PathImageList[i];
```

```
    pFrame.index = i;
```

```
    pFrame.valid = fileExist(pathRec2File);
```

```
    Push pFrame to vFrame;
```

```
    if pFrame is valid
```

```
        then
```

```
loadRectangle(pathRec2File, pFrame.rectPlate);
pFrame.Height = pFrame.y;
/* Yeterli ALAN kistası */
if (pFrame.Height < (height of plate)*katParam)
then
    pFrame.valid = false;
    continue;
End
    pFrame.image = cvLoadImage(PathImageList[i]);
End
End
return vFrame;
End
```

4. RENK TANIMA BÖLGESİNİN (ROI) BELİRLENMESİ

Araç üzerinde rengin tanınacağı bölgeleri belirlemek için görüntüde araç üzerinde bir başlangıç noktasına ihtiyacımız vardır. Referans noktası olarak da, daha önce çalışan Plaka Tanıma programından elde edilen plaka koordinatları kullanılmaktadır [9]. Üç farklı renk tanıma için ilgili bölge (region of interest – ROI) seçimi yapılmıştır. Tüm araç, yarım araç ve renkli kaput parçası olarak adlandırılan bu üç ROI seçiminden ilki yapılan denemelerde pek başarılı çıkmadığı için burada sadece yarım araç ROI ve kaput parçası ROI'den bahsedilecektir.

4.1 Kaput Parçası ROI

Aracın renkli kaput kısmından alınan yeterince büyük ve renk değişiminden uzak parçadır. Kaput üzerinde uygun renkli parçayı bulmak için standart sapmalı [10] bölge seçimi algoritması kullanılmaktadır. Algoritma akışı aşağıdaki gibidir.

Elimizdeki koordinat bilgileri ile aracın plakasının sol üst köşesine gelinmektedir. Bundan sonra yapılan ilk işlem aracın görüntüde yeterince yer kaplayıp kaplamadığına bakmak olmaktadır. Bunu hesaplamak için de plaka boyutları referans olarak kullanılmaktadır. Örneğin, görüntü plakanın üst sınırından plaka yüksekliğinin en az 4 ile 5 katı kadar yukarısında bitme koşulu aranmaktadır. Böylece, renk bilgisinin alınacağı kaput bölgesine erişme olasılığının çok düşük olacağı veya bölgenin çok küçük olacağı aksi durumlar en aza indirgenmeye çalışılmaktadır.

Plaka bölgesinin hemen üstünden başlayarak bir plaka bölgesi büyüklüğünde bölge belirleniyor. Daha sonra bu bölgenin sağından ve solundan yüksekliği aynı genişlikleri plaka genişliğinin 0.65'i kadar olan ortadaki bölgeye bitişik iki bölge belirlenmektedir.

En sağdaki bölgeden başlayarak sırasıyla her bölgenin varyansını (siyah beyaz resminin) hesaplayıp her seferinde bu bölgeleri ikişer satır yukarıya kaydırmak suretiyle varyansları hesaplanır ve saklanır. Daha sonra bu varyanslardan bir “otsu” eşik değeri hesaplanır. Araç tekrar yukarıda bahsedildiği biçimde taranmaya başlanır.

Eğer bulunan eşik değerinden küçük art arda iki eşik değeri bulunursa seçilecek ROI'nin alt sınırı tarama bölgesinin alt sınırı olarak belirlenir. ROI'nin üst sınırı ise tekrar yüksek eşik değerlerine ulaşıldığında bulunmuş olur. Böylece kaput üzerinde 3 tane dikdörtgensel bölge elde edilir. Son versiyonda seçilen 3 ROI'den en büyük ve sınırları en uygun yerde olanı seçilip üzerinde renk tanıma işlemi yapılmaktadır. Aşağıda kaput parçası bölgesini seçmek için geliştirdiğimiz algoritma yer almaktadır.

```
/**/  
Method FindSmoothROI ( Im, pt )  
Begin  
    tmpROI[0] = getInitialRect( Im, pt );  
    ROI = tmpROI[0];  
    integer i = 0;  
    while tmpROI[i] not on the top of the image do begin  
        stdDev[i] = StandartDeviation( tmpROI[i] );  
        tmpROI[i+1] = SlideUpBy2Pixel( tmpROI[i] );  
        i++;  
    End  
    threshold = findOtsuThr( stdDev);  
    count = 0;  
    For standart deviation i=0 to size(stdDev) do  
        if stdDev[i] > threshold  
        then if count is 0  
            then count++;  
            else if count is 1  
                then ROI.bottomY = tmpROI[i].bottomY;  
            else ROI.upperY = tmpROI[i].upperY;  
            break;  
    End  
    return ROI;  
End
```

Yukarıda bahsedilen ROI seçim yöntemi piksel piksel ve orta değer bulma öznelik vektörü oluşturma yöntemlerinde kullanılmaktadır. Şekil 4.1'de konu edilen küçük parça ROI gösterilmektedir.



Şekil 4.1: Kaput Parçası ROI

4.2 Yarım Araç ROI

Yarım araç ROI seçimi için geliştirilen algoritma aracın plaka boyu ile orantılı sınır yerlerinin belirlenmesinden ibarettir. Algoritma EUKS algoritmasının seçtiği araç görüntüsü üzerinde plakanın sol üst köşesinden başlayıp plaka boyunun yarısı kadar aşağıya, iki katı kadar yukarı ve bir boy sola iki katı kadar da sağa giderek ROI'nin sınırlarını belirler [11]. Böylece elde edilen ROI plakanın biraz altından başlayıp yukarıdan ön camın başladığı kaputun bittiği yer, sağdan ve soldan ise aracın sınırları içinde kalan Şekil 4.2'deki gibi bir bölgedir. Yarım araç ROI histograma dayalı öznitelik vektörler oluşturmada kullanılır.



Şekil 4.2 : Yarım Araç ROI

5. ÖZİNİTELİK SEÇİMİ VE ÖZİNİTELİK VEKTÖRÜ OLUŞTURMA

5.1 Kullanılan Renk Bileşenleri

ART sisteminde farklı özellikleri dolayısıyla birçok renk uzayı ve bileşenleri kullanılmıştır. Pikseller sayısal ortamda genellikle RGB renk uzayı ile ifade edilirler. RGB renk uzayı bileşenlerinin dönüştürülmesiyle elde edilen renk uzayları şöyledir: normalize RGB [7, 12], HSV [8, 13, 5], L*a*b [5], Otha renk uzayı (I1 I2 I3) [14, 10], L-color renk uzayı [2], ve kromatik renk uzayı ($\xi_1 \xi_2$) [1].

Yapılan testlerden sonra en başarılı öznitelik dizimi ortaya çıkacak ve öznitelik vektörünü en iyi sonuç veren renk bileşenlerinden biri, birkaçı ya da hepsi oluşturabileceklerdir. Öznitelik vektörü en iyi başarı sonucunu veren renk uzayı kombinasyonunu içermelidir.

5.1.1 RGB renk uzayı

RGB (red, green, blue), sayısal ortamda renkleri ifade etmek için kullanılan en yaygın renk uzayıdır. R kırmızıyı (255, 0, 0), G yeşili (0, 255, 0), B de mavi (0, 0, 255) rengini ifade eder. Renkli görüntü üzerindeki bir piksel R, G ve B değerlerini taşıyan üç elemanlı bir dizi oluşturur. Her bir renk bileşeni [0, 255] aralığında birer doğal sayı değerleri alarak 2^{24} tane farklı renk oluşturur. RGB renk uzayı ile temsil edilen renkler yansıma, bakış açısı, cismin şekli ve görüntünün çekildiği cihaza bağlı olarak değişiklik gösterirler [3]. RGB bunlar gibi renk tanıma uygulamasını etkileyebilecek parametrelerden bağımsız değildir.

5.1.2 Normalize RGB renk uzayı

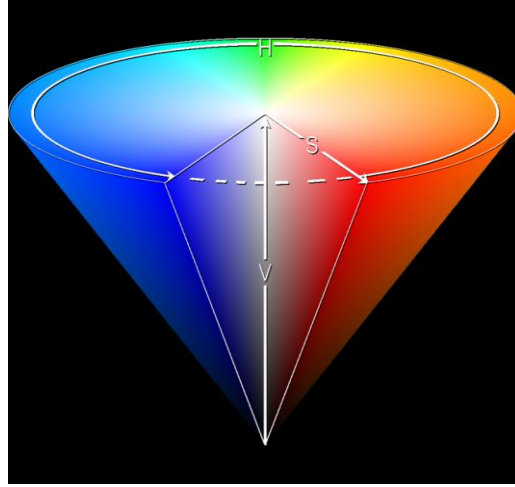
Normalize RGB renk uzayı, RGB uzayının renk bileşenlerinin normalize edilmesiyle oluşturulmaktadır. Her bir renk bileşeni değeri tüm bileşen değerlerinin toplamına bölünerek elde edilir [15]. Böylelikle, normalize RGB (nRGB) renk bileşenleri [0, 1] aralığında değerler alırlar. RGB renk uzayı için renk bileşenlerinin değerleri r , g , ve b olsun, nRGB değerleri nR , nG ve nB (5.1) formülünde gösterildiği gibi elde edilirler.

$$nR = \frac{r}{r+g+b}, \quad nG = \frac{g}{r+g+b}, \quad nB = \frac{b}{r+g+b} \quad (5.1)$$

Normalize RGB renk uzayı yansıma, bakış açısı ve cisim şekillerinden kaynaklanan renk değişikliklerinden bağımsızdır [12]. Sistemimizde yapılan testler sonucu, sadece nR ve nG bileşenlerinin kullanılmasına karar verilmiştir.

5.1.3 HSV renk uzayı

HSV (Hue, Saturation, Value) renk uzayının renk bileşenlerinden H renk özü, S doygunluk ve V de parlaklık ya da koyuluk-açıklık oranı olarak tarif edilmektedir. İnsan gözüne benzer bir renk algılayışı ifade etmek için tanımlanmıştır. HSV, aynı zamanda renk tanıma için en uygun renk uzaylarından biridir [5]. Çünkü, RGB'nin aksine HSV renk uzayında aydınlık değeri ayrı ifade edildiği için resimde ne kadar parlaklık ya da gölge olursa olsun H ve S değerlerinden gerçek renge ulaşmak mümkün olmaktadır. Yalnız HSV cihaz bağımsız bir renk uzayı değildir. Farklı bir aygıttan elde edilen aynı görüntü değer olarak farklılık gösterebilmektedir [8].



Şekil 5.1: HSV renk uzayının konik gösterimi

HSV, RGB renk uzayının renk bileşenlerinden doğrusal olmayan bir yöntemle elde edilmiştir. Bu durum, aşağıdaki (6.2), (6.3) ve (6.4) dönüşüm formüllerinde de açıkça görülmektedir.

Renk özü (H); rengin dalga boyu değerini taşıyarak rengin kimliğini tayin eder. Dalga uzunluğunu verdiği için açısal bir değerdir ve 0° ile 360° arasında değer alır.

Doymunluk (S); renk oranını belirtir. S değeri [0, 100] aralığında deęiřir. S bileřeni bydke renk de daha canlı tonlarda grnrken, doymunluk oranı dřtke renk gri tonlara yaklařır.

Parlaklık (V); rengin koyuluk aıklık, bařka bir deyiřle aydınlık oranını belirler. V'nin deęiřim aralıęı da [0, 100] 'dir.

H, S, ve V renk bileřenleri RGB renk uzayından $H \in \{0, 360\}$ ve $S, V, R, G, B \in \{0, 1\}$ olarak alınarak hesaplanmaktadır [13].

$$H = \arctan\left(\frac{\sqrt{3}(G - B)}{(R - G) + (R - B)}\right) \quad (5.2)$$

$$S = 1 - \frac{\min(R, G, B)}{R + G + B} \quad (5.3)$$

$$V = \max(R, G, B) \quad (5.4)$$

5.1.4 L*a*b renk uzayı

İnsanlarda renk algısının, retinaya dřen grlebilir ıřık ıřınlarının (400mm-700mm dalga boyu aralıęı) oranına dayanmakta olduęu bilgisine baęlı olarak ve retinada  tr renk algılayıcı hcre (photo-receptor cone cells) bulunmasından yola ıkarak, her biri farklı renk tayfı eęrisine karřılık gelmek zere,  sayı ile tm renklerin tamamen ifade edilebileceęi kanısına varılmıřtır. Bu ifade biimi renk bilgisinin yaratılması ve grselleřtirilmesi aısından pek ok uygulama iin kullanım kolaylıęı saęlamıřtır. Nasıl ki RGB renk uzayı televizyon monitr gibi grnt cihazlarında yaygın olarak kullanılıyorsa, renkler arasındaki fark edilebilir benzerlikleri olabildięince ayırt edebilmeye ynelik L*a*b renk uzayı tasarlanmıřtır [14]. L*a*b renk uzayında, tek ve yakın renkler arasında algılanan farklar, renk koordinatlarının klit mesafelerine karřılık gelir.

RGB'den L*a*b renk uzayına gerekleřtirilen doęrusal olmayan dnřmler ařaęıdaki denklemler ile tanımlanabilir;

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (5.5)$$

$$f(t) = \begin{cases} t^{1/3} & \text{eger } Y/Y_n > 0.008856 \\ 7.787t + 16/116 & \text{aksihalde} \end{cases}, \quad (5.6)$$

$$L^* = \begin{cases} 116(Y/Y_n)^{1/3} - 16 & \text{eger } Y/Y_n > 0.008856 \\ 903.3(Y/Y_n) & \text{aksihalde} \end{cases}, \quad (5.7)$$

$$a^* = 500(f(X/X_n) - f(Y/Y_n)), \quad (5.8)$$

$$b^* = 200(f(Y/Y_n) - f(Z/Z_n)) \quad (5.9)$$

5.1.5 L-color renk uzayı

Bir görüntüdeki renkleri, n tane renk içeren ayrık bir renk uzayı ile haritalandırsak elde edeceğimiz renk histogramı $H(I)$, $(h_{c_1}, h_{c_2}, h_{c_3}, \dots, h_{c_n})$ biçiminde, her elemanı görüntünün c_j rengini içerip içermeme olasılığını temsil eden bir vektör olur. En yaygın kullanılan renk histogram metrikleri L_1 ve L_2 'dir [2]. 1995 yılında Hafner tarafından renk benzerliği konusunda insan yargısına yakınlık gösteren daha gelişmiş bir yöntem ortaya atıldı. Bu metoda göre farklı renkler arasındaki benzerlikleri eşleştiren ve renk histogram binleri tarafından temsil edilen karesel bir uzaklık ölçümü L_q ifade edildi:

$$L_q(x, y) = (x - y)^t * A * (x - y),$$

Burada, x ve y renk histogramları; a_{ij} i, j renk binleri arasındaki benzerliği ifade etmek suretiyle, $A = [a_{ij}]$ 'dir.

Renklerin renk histogramlarına dağılımı konusundaki açıklama eksikliğini tespit eden Huang, 1997’de, görüntü düzeltmeye yönelik “renk korelogramı” adıyla yeni bir renk niteliği önerdi. Bu niteliğin görevi renk çiftlerinin uzaysal yaklaşmasının resim içindeki uzaklığa göre değişimini karakterize etmektir.

Bir resmin korelogramı, renk çiftleri ile numaralanmış bir tablo olup; bu tablonun $\langle c_i, c_j \rangle$ renk piksel ikilisi için k . elmanı, c_i renk pikselinden k değeri kadar uzakta c_j renk pikselinin bulunma olasılığını gösterir.

$$Cg_{c_i, c_j}^k(I) = \Pr_{p_1 \in I_{c_i}} [p_2 \in I_{c_j} \mid |p_1 - p_2| = k] \quad (5.10)$$

Bu denklemde, p_1 ve p_2 resimdeki pikselleri temsil eder ve I_p ’nin piksel rengini göstermesi kaydıyla $I_c = \{p \mid I(p) = c\}$ ’dir. Genellikle renk korelogramının boyutları çok büyük olduğu için renk otokorelogram kullanılır. Bu nitelik sadece eş renkler arasındaki uzaysal yaklaşmayı kaydeder.

1999 yılında Gevers ve Smeulders’in analizleri sonucu değişken aydınlatmalı ortamlar için yeni bir renk modeli kullanılmaya başlandı, bu renk modelinin beyaz ışık kaynağı ile aydınlatılmış ortamlarda kullanımının iyi sonuç verdiği tespit edildi [2].

Bu modele göre RGB uzayından L-color renk uzayına geçiş şöyle ifade edilmektedir:

$$I_1(R, G, B) = \frac{(R - G)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2} \quad (5.11)$$

$$I_2(R, G, B) = \frac{(R - B)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2} \quad (5.12)$$

$$I_3(R, G, B) = \frac{(G - B)^2}{(R - G)^2 + (R - B)^2 + (G - B)^2} \quad (5.13)$$

5.1.6 Otha renk uzayı

Otha'nın dış ortam görüntülerinde nesnelere daha kolay algılamak ve ayırt etmek için tanımlanmış olduğu bir renk uzayıdır [16]. Hesagawa ve Kanade araç renklerinin sınıflandırmasıyla ilgili yaptıkları çalışmada Otha'nın renk uzayını kullanmışlardır [17]. Bu renk uzayı I_1 I_2 I_3 bileşenlerinden oluşmaktadır. Renk bileşenlerinin aralarında istatistiksel ilinti olmaması ve aynı zamanda dış ortamlarda ve ışığın kontrol edilemediği ortamlarda aydınlatma değişimlerinden bağımsız olmaları, bu renk uzayını bilgisayarla görü dünyasında tercih edilir kılmaktadır [18]. RGB'den Otha renk uzayına aşağıdaki (5.14), (5.15) ve (5.16) denklemlerinde görüldüğü gibi lineer olarak geçiş yapılmaktadır.

$$I_1 = \frac{R + G + B}{3} \quad (5.14)$$

$$I_2 = \frac{R - B}{2} \quad (5.15)$$

$$I_3 = \frac{2R - G - B}{4} \quad (5.16)$$

5.1.7 Kromatit renk uzayı

Rengin parlaklık özelliğinden bağımsız olarak, renk kalitesi sınıflandırmasını rengin renk aralığına dayandırıp nesnel bir şekilde niteleyen "kromatiklik" kavramı, renk biliminde ışık kaynağının ya da görüntü yüzeyinin nötr referans noktası olan beyaz noktayı niteler. Kromatit renk uzayı, aydınlatma pozisyonuna bağlı olarak görüntüdeki gelen renk dağılımında meydana gelen değişiklikleri azaltır.

Örnek bir iki boyutlu kromatit renk uzayı $x = [x_1, x_2]^T$, algılayıcı tepkilerinin aşağıdaki gibi normalize edilmesiyle oluşturulur:

$$x_1 = \frac{q_R}{q_R + q_G + q_B} \quad x_2 = \frac{q_G}{q_R + q_G + q_B} \quad (5.17)$$

Bu renk uzayı sensör tepki oranlarının logaritmalarına ve aydınlatmadan kaynaklanan kromatit deformasyonuna dayandırılır. Kromatit imzaların bozulması transformasyon bağlantıları tarafından yönetilir. Köşegen matris dönüşümüne dayalı renk tutarlılığı ve etkileri bir grup araştırmacı tarafından araştırıldığında, ışıklandırma değişiminin aydınlatma ve yansıma için gerekli kısıtlara dayandıran düşük köşegen matris dönüşümü şeklinde modellenebileceği gösterildi [1]. Bu model olmaksızın köşegen matris dönüşümüne yaklaşmanın yolu dar renk filtreleri kullanmak olabilirdi. Eğer $Q_k(\lambda)$ 'nın filtre fonksiyonu ve $\Delta Q_k(\lambda)$ 'nın λ_k dalga boyu etrafında dar bir bant genişliğine sahip olduğu, $e(\lambda)$ ve $s^p(\lambda)$ 'nin hemen hemen filtre bant genişliği içinde sabit olduğu kabul edilirse, Q_m filtre karşılığının büyüklüğü ve q^p ve \tilde{q}^p sensör tepkileri arasındaki ilişki olmak kaydıyla:

$$q_k \approx [n_s \cdot n^p] e(\lambda_k) s^p(\lambda_k) Q_m \Delta Q_k \quad , \quad (5.18)$$

$$\tilde{q}^p = r^p D q^p \quad (5.19)$$

Köşegen elemanları ise şöyledir:

$$D_{00} = \tilde{e}(\lambda_0)/e(\lambda_0), \quad D_{11} = \tilde{e}(\lambda_1)/e(\lambda_1), \quad D_{22} = \tilde{e}(\lambda_2)/e(\lambda_2) \quad (5.20)$$

Köşegen matris dönüşümü bir çok renk sabitleme algoritması için başlangıç noktası olarak kullanılmaktadır. Köşegen dönüşüm koşulu altında, Berwick aydınlatma ve yansımadan bağımsız yeni bir kromatik renk uzayı $\xi = [\xi_1, \xi_2]^T$ ileri sürmektedir. Aşağıdaki formül 5.18'de verilen dönüşüme göre, aydınlatma rengi ne olursa olsun görüntüdeki cisimlerin renklerinin değişmemesi beklenmektedir.

$$\xi_1 = \ln\left(\frac{q_R}{q_G}\right), \quad \xi_2 = \ln\left(\frac{q_B}{q_G}\right) \quad (5.21)$$

Farklı aydınlatma rengi altında bir nesnenin kromatik dağılımları aynı görünmelidir.

Aydınlık, $\eta = \left[\ln \frac{D_{00}}{D_{11}}, \frac{D_{22}}{D_{11}} \right]^T$ olduğu durumlarda ξ renk uzayı $\tilde{\xi} = \xi + \eta$

dönüşümünü sağlar. Bu renk uzayı nesnelerin tanımlanması ve aydınlık tahmini için kullanılmaktadır.

Denklem 5.21 ile iki boyutlu kromatit uzayı (5.17) arasındaki ilişki aşağıdaki 5.22 dönüşümleri ile verilmektedir [1].

$$\xi_1 = \ln\left(\frac{x_1}{x_2}\right), \quad \xi_2 = \ln\left(\frac{1-x_1-x_2}{x_2}\right) \quad (5.22)$$

5.2 Öznitelik Vektörü Oluşturma Yöntemleri

Sistemimizde histogram tabanlı ve piksel tabanlı olmak üzere iki ana öznitelik oluşturma algoritması bulunmaktadır. Piksel tabanlı yöntemini de orta değer pikseli bulma yöntemi ve piksel-piksel sınıflandırma yöntemi olarak iki ana başlıkta inceleyebiliriz.

5.2.1 Histogram tabanlı öznitelik oluşturma yöntemi

Görüntüler sayısal ortamda genellikle 24 bitlik RGB renk uzayıyla ifade edildikleri gibi, RGB'den farklı özelliklerde çeşitli renk uzayları ile de temsil edilebilirler (HSV, Lab, L-color gibi). Bu çalışmada, RGB renk uzayında olan görüntüler özellikleri ve dönüşüm denklemleri bölüm 5.1'de anlatılan renk uzaylarına çevrilip, bu renk uzayı bileşenlerinden oluşan iki boyutlu histogramlar oluşturulmaktadır. Histogramlar 8x8, 16x16, 32x32 bölmelerden (bin) oluşmaktadır. Histogramlar aşağıdaki renk uzayı bileşenleri çiftlerinden oluşmaktadır:

İki boyutlu: Rn-Gn, H-S, S-V, a-b, L-a, L-b, L1-L2, L1-L3

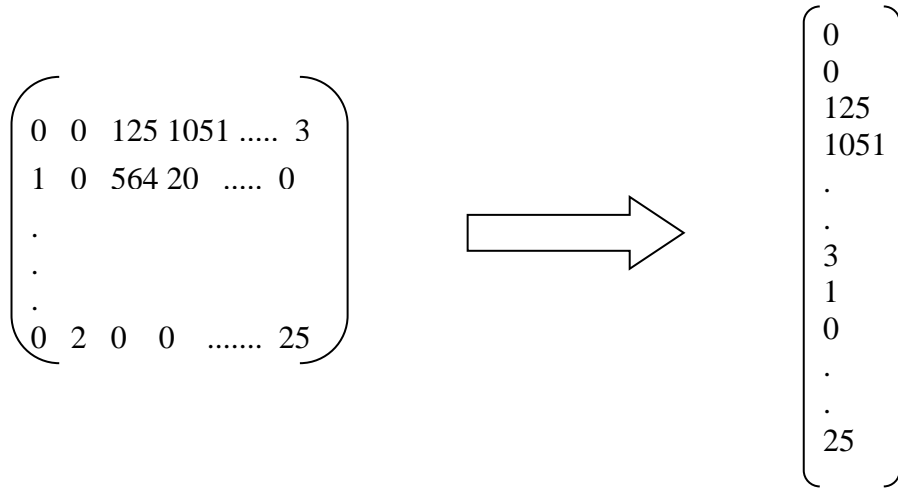
Bir boyutlu: Gray1, Gray2

Gray1 ve Gray2 bileşenleri aşağıdaki denklemlerde verilmiştir.

$$Gray_1 = \frac{R + G + B}{3 * 256} \quad (5.23)$$

$$Gray_2 = \frac{0.299R + 0.587G + 0.114B}{256} \quad (5.24)$$

Bir veya birden fazla histogramın tek boyutlu diziye çevrilmesi ile art arda sıralanması ile öznitelik vektörü elde edilmektedir.



Şekil 5.2: İki boyutlu histogramın tek boyutlu öznitelik vektörüne çevrilmesi

İki boyutlu normalize histogramların tek boyutlu öznitelik vektörlerine çevrilmesi gerekmektedir. İki boyutlu dizi Şekil 5.2’de gösterildiği gibi satır satır dizilerek vektör haline getirilmektedir. Histogram tabanlı öznitelik vektörlerinin oluşma adımları aşağıdaki algoritma ile gösterilebilir.

5.2.2 Ekli Histogram tabanlı öznitelik oluşturma yöntemi

Ekli histogram öznitelik vektörü oluşturma yöntemi histogram tabanlı öznitelik vektörü oluşturma yöntemine çok benzemektedir. Bu yöntemde de yarım araç ROI üzerinden öznitelik vektörleri oluşturmada, öznitelik vektörleri aynı renk bileşenlerinden elde edilen 1 ve 2 boyutlu histogramların peş peşe sıralanmasıyla oluşturulmaktadır.

Ekli histogram modeli histogram tabanlı öznitelik oluşturmada farklı olarak, yarım araç ROI’lerde yukarıdan başlayarak kaputun bitimine kadar olan alanı ve kaputun bitimi ile resmin sonuna (plakanın bitimine) kadarki alanlardan ayrı ayrı histogramları hesaplamaktadır. Üst ve alt bölgelere ait histogramlar önce üst bölgenin histogramları sonra alt bölgeden elde edilen histogramlar olmak üzere art arda eklenirler. Bu nedenle bu yönteme “Ekli Histogram” adı verilmiştir. Şekil 5.3’te üst ve alt bölgeler gösterilmiştir.



Şekil 5.3: Ekli histogram öznitelik vektörü için oluşturulan üst ve alt bölgeler

Ekli histogram tabanlı modelinin geliştirilmesinin amacı, daha renkli olan aracın kaput kısmı ile ızgara kısmını birbirinden ayırmak ve renklerin birbirini ezmesini engellemektir. Ekli histogram tabanlı öznitelik vektörleri histogram tabanlı öznitelik vektörlerinin iki katı uzunluğundadır. O nedenle sınıflandırma işlemi yapılırken sistemi biraz yavaşlatabilir.

5.2.3 Piksel tabanlı öznitelik oluşturulma yöntemi

Görüntü üzerinde renk tanımanın yapılacağı bölgeler (ROI) belirlendikten sonra, bu bölgelerde yer alan piksel değerleri RGB renk uzayından farklı renk uzaylarına çevrilmekte ve bu renk uzayı bileşenlerine karşılık düşen piksel değerleri art arda sıralanarak öznitelik vektörünü oluşturmaktadırlar. Burada pikselleri seçerken iki farklı yöntemden biri uygulanabilmektedir. A'nın RGB renk öğelerini temsil eden görüntü olduğunu kabul edelim. Öznitelik vektörünün de HSV renk uzayının H, S ve V değerlerinden oluştuğunu farz edersek, X öznitelik vektörü aşağıdaki gibi bulunur:

```
Method CreatePixelFV( A, X ) Begin  
    r = A[p-1]R ;  
    g = A[p-1]G ;  
    b = A[p-1]B ;  
    RGBToHSV(r, g, b, H, S, V, 0);  
    Push H to X;  
    Push S to X;  
    Push V to X;  
End
```

5.2.3.1 Orta değer-piksel öznitelik vektörü oluşturma yöntemi

Orta değer-piksel yönteminde, üzerinde çalışılacak ROI'deki tüm pikseller kullanılacak renk bileşenlerine çevrilir. Her renk bileşeninin ayrı ayrı orta değeri alınır. Bu farklı renk bileşenlerinden elde edilen orta değer değerleri öznitelik vektörünü oluşturur. Bu yöntemde sadece bir kere öznitelik vektörü oluşturulup sınıflandırma algoritmasına gönderilir.

```
/*Orta değer-piksel öznitelik vektörü*/
Method ORTA DEĞER_PIKSEL( A, X)
Begin
    For each pixel  $p$  in the image do begin
         $r = A[p-1]_R$  ;     $g = A[p-1]_G$  ;     $b = A[p-1]_B$  ;
         $R_n = r / \max(1, b + g + r)$  ;
         $G_n = g / \max(1, b + g + r)$  ;
        RGBToHSV( $r, g, b, H, S, V, 0$ ) ;
         $B_{R_n}[p-1] = R_n$  ;
         $B_{G_n}[p-1] = G_n$  ;
         $B_H[p-1] = H$  ;     $B_S[p-1] = S$  ;     $B_V[p-1] = V$  ;
    End
    For component  $c = 0$  to compSize do
        sort(  $B[c]$  ) ;
         $m = B[c][p/2]$  ;
        Push  $m$  to  $X$  ;
End
```

5.2.3.2 Çoğul-piksel öznitelik vektörü oluşturma yöntemi

Çoğul-piksel yönteminde ROI'de bulunan her piksel farklı renk uzaylarına çevrilir ve her piksel için ayrı öznitelik vektörü oluşturur. Böylece her piksel için sınıflandırma yapılır ve aracın rengi piksellerin çoğunluğunun ait olduğu renk olarak belirlenir.

6. SINIFLANDIRMA YÖNTEMLERİ

Renk tanıma sisteminin en önemli parçası kullanılacak sınıflandırıcıya karar vermek ve farklı sınıflandırma yolları aramaktır. Bu çalışmada öznitelik seçimi ile beraber hangi sınıflandırma yönteminin en iyi sonucu vereceğini bulmak gerekti. Bunun için bir dizi test yapılmış ve başarı sonuçları alınmıştır. Kullanılan sınıflandırma yöntemleri: K-NN, Şablon Eşleme (TM), Yapay Sinir Ağları (ANN) ve Destek Vektör Makinesi (SVM) sınıflandırıcıları kullanılmıştır. Ayrıca, Voting kaynaştırma yöntemi ile en iyi sonuçlar arasında kaynaştırma yapılmış fakat başarılı olunamamıştır. Son olarak da karar ağacı oluşturulmuş ve dallanmalarda kullanılacak sınıflandırıcı ile öznitelik vektörlerine karar verilmiştir.

6.1 K En Yakın Komşu (K Nearest Neighbour : K-NN)

K-NN sınıflandırma yöntemi sık kullanılan kolay geliştirilebilen bir yöntemdir [19]. Sınıflandırılmak istenen bir x elemanımız olsun. Bu x elemanın sınıflandırılacağı N adet sınıfımız olsun. Her bir sınıf için birer eğitim kümesi oluşturulur. x elemanın eğitim kümelerindeki her bir elemana olan uzaklığı hesaplanır ve x elemanın en yakın olduğu k tane eleman seçilir. Bu k en yakın komşunun çoğunluğu hangi sınıfa aitse x elemanı da o C sınıfına atanır. Elemanlar arasındaki uzaklık Euler Uzaklığı (L2) ve mutlak uzaklık (L1) gibi yöntemlerle hesaplanır. Bu uzaklıkların hesaplanması (6.1) ve (6.2) formüllerinde gösterilmiştir [14].

$$L1 = \sum_1^N |t_i - x_i| \quad (6.1)$$

$$L2 = \sum_1^N (t_i - x_i)^2 \quad (6.2)$$

Yukarıdaki formüllerde bir t eğitim kümesindeki her bir elemanı, x sınıflandırılacak elemanı, N bir elemanın öznitelik vektöründeki öznitelik sayısı, i ise sayacıdır.

6.2 Şablon Eşleme (Template Matching : TM)

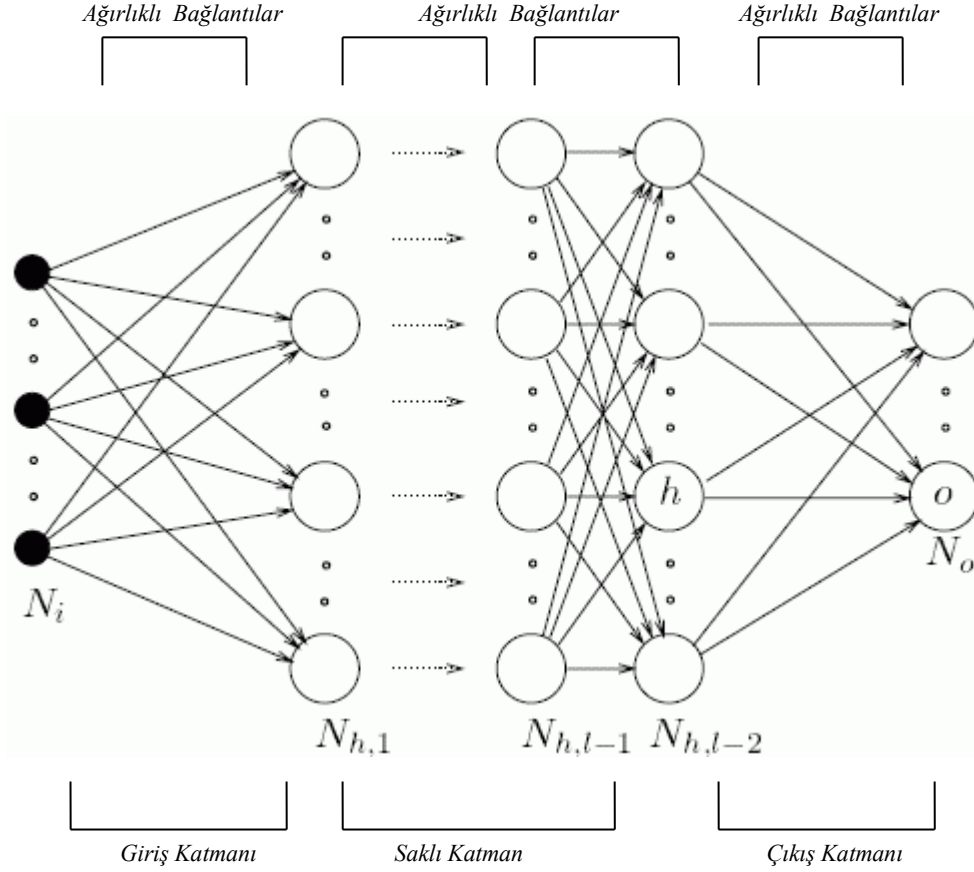
TM sınıflandırma yöntemi sık kullanılan, kolay geliştirilebilen bir yöntemdir. TM sınıflandırıcısı, sınıflandırılacak elemana en yakın elemanı bulmayı amaçlar. K-NN sınıflandırıcısından tek farkı eğitim kümesindeki elemanların k tanesine değil en yakın bir tanesine bakmasıdır [20].

Sınıflandırılmak istenen bir x elemanımız olsun. Bu x elemanının sınıflandırılacağı N adet sınıfımız olsun. Her bir sınıf için birer eğitim kümesi oluşturulur. x elemanının eğitim kümelerindeki her bir elemana olan uzaklığı hesaplanır ve x elemanının en yakın eleman seçilir. x elemanının sınıfı bulunan en yakın komşunun sınıfı olarak atanır. Elemanlar arasındaki uzaklık Euler uzaklığı (L2) ve mutlak uzaklık (L1) gibi yöntemlerle hesaplanır.

6.3 Yapay Sinir Ağları (Artificial Neural Networks - ANN)

Yapay Sinir Ağları, insan beyninin bilgi işleme teknolojisinden esinlenerek geliştirilmiş bir bilişim teknolojisidir. ANN ile basit biyolojik sinir sisteminin çalışma şekli simüle edilir. ANN ağırlıklandırılmış şekilde birbirlerine bağlanmış birçok işlem biriminden (nöronlar) oluşan matematiksel sistemlerdir. ANN mimarisinin temel işlem birimi olan nöronlar tek başlarına hiçbir şey ifade etmezler ve sık sık transfer fonksiyonu olarak anılan denklemlerdir. Tüm nöronlar bir sonraki katmandaki nöronlarla bağlantı içindedir ve bu bağlantılarla işlem yapabilmektedirler. Bu işlem birimi, diğer nöronlardan sinyalleri alır, bunları birleştirir, dönüştürür ve sayısal bir sonuç ortaya çıkartır [21]. Genelde, işlem birimleri kabaca gerçek nöronlara karşılık gelirler ve bir ağ içinde birbirlerine bağlanırlar, bu yapı da sinir ağlarını oluşturmaktadır.

En yaygın olarak kullanılan ANN'lerden biri olan Çoklu Katman Algılama (MLP), girdi katmanı, saklı katman ve çıktı katmanı olmak üzere üç farklı katmandan oluşmaktadır. MLP, üç çeşit farklı sayıdaki nörona sahip tabak serilerinden oluşmaktadır. Girdi katmanı ağ alıcısı, saklı katman aracı ve çıktı katmanı ise sistem vericisidir. Girdilerle çıktılar arasındaki bağlantı doğrusal olmadığında saklı katman, çıktılarının genelleştirilmesini sağlar [22]. Ayrıca işlem birimleri olarak da adlandırılan farklı sayıdaki nöronlar her katmanda yer alır ve birbirlerine içsel ağırlıklarla bağlıdır. MLP ağ modelinin genel mimari yapısı Şekil 6.1 'de gösterilmiştir.

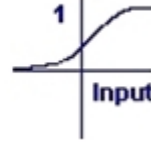


Şekil 6.1: Çok katmanlı Yapay Sinir Ağı [23]

ANN'ler pratik kullanımda bir çok avantajlara sahiptir. ANN sistemleri örneklerden öğrenme ve sonuçları genelleştirip başka datalar oluşturma yeteneğine sahiptir. Girdi ve çıktıları eşleştirmek için herhangi bir modelleme veya programlamaya ihtiyacı yoktur. Ayrıca, bunlar eksik data ve gürültü ile çalışabilir. Bunun yanında karar vericiye zaman ve maliyet avantajı sağlarlar. ANN metodu , bilgi yönetimi karmaşık ve bilgiye dayalı uzman kararı zaman aldığıda seçilir [24]. Tüm bunlar dikkate alındığında, ANN'ler mühendislik ve yönetim alıştırmaları uygulamalarında geniş yer tutar.

Sistemimizde kullandığımız ANN için bazı parametrelerin belirlenmesi gerekti. Kullandığımız ANN'in 100 gizli düğüm sayısı, değişken giriş ve 7 çıkış sayısına (renk sınıf sayısı) sahip bir yapay sinir ağıdır. 20 iterasyonda eğitim kümesi kullanılarak bağlantı ağırlıkları (weighth of synapses – wts) elde edilir. Düğümlerde geçiş fonksiyonu olarak Şekil 6.2'de gösterilen "sigmoid" fonksiyonu kullanılmaktadır.

$$f(x) = \frac{1}{1 + e^{-x}}$$

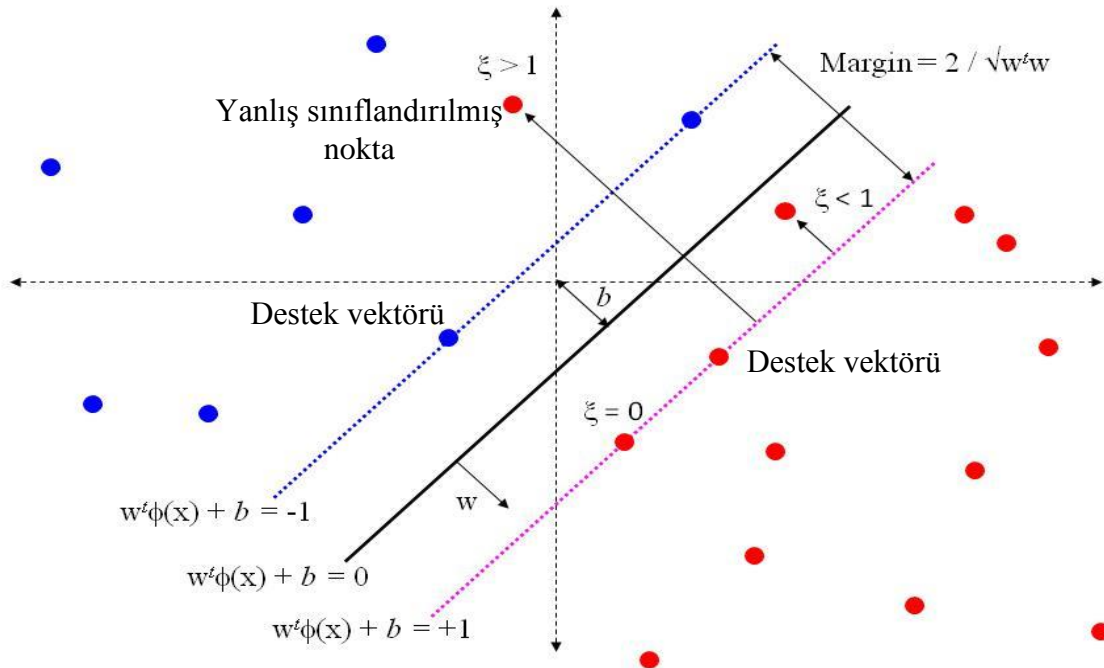


Sigmoid fonksiyonu

6.4 Destek Vektör Makineleri (Support Vector Machine – SVM)

Destek vektör makineleri (SVM) sınıflandırma ve regresyon için kullanılan bir dizi öğrenme metodlarından oluşur. Daha açık söylemek gerekirse, iki kategoriden biri ile etiketlenmiş bir veri kümesinde bir SVM eğitim algoritması yeni bir örneğin birinci kategoriye mi yoksa diğerine mi denk düştüğünü söyleyen bir model geliştirir. Bu öyle bir model ki, uzayda yer kaplayan noktalar olarak ifade edilen örnekler farklı kategorideler ise aralarında mümkün olduğunca geniş bir boşluk bulunmaktadır [25]. Böylece ikiden fazla kategori için bir dizi hiperdüzlem oluşmuş olur. İyi bir sınıflandırma için veri noktaları kümeleri arasındaki mesafe ne kadar büyük ise hata payı da o kadar düşer [27].

Vladimir Vapnik 1979’da Destek Vektör Makinelerini(SVM) icat etmiştir [28]. En basit olarak, doğrusal form bir SVM, olumsuz örnek bir dizi ile olumlu örnek bir diziyi maksimum farkla ayıran hiperdüzlemdir (bk. Şekil 6.3).



Şekil 6.2: Destek Vektör Makinesi – Hiperdüzlemlerle sınıflandırma [29]

Doğrusal durumda marjin, negatif ve pozitif örneklerin hiperdüzleme en yakın mesafesi olarak tanımlanır. Doğrusal SVM çıktısı olan kenar sınırı aşağıdaki formülle ifade edilir:

$$u = \bar{w} \bullet \bar{x} - b \quad (6.3)$$

Burada w , hiperdüzlemin normal vektörü ve x de girdi vektörüdür. Ayrık hiperdüzlem, $u = 0$ düzlemidir. En yakın nokta $u = \pm 1$ düzlemidir. Buna göre kenar mesafesi m formül 6.4 gibi hesaplanır ve 6.5 denklemindeki gibi maksimize edilir.

$$m = \frac{1}{\|w\|_2} \quad (6.4)$$

$$\min_{w,b} \frac{1}{2} \|\bar{w}\|^2, \quad y_i = (\bar{w} \bullet \bar{x}_i - b) \geq 1, \forall i \quad (6.5)$$

Burada x_i , eğitim kümesinin i 'inci örneği ve y_i de i 'inci eğitim örneğine ait SVM'in doğru bir çıktısıdır.

Y_i değeri pozitif örnekler sınıfında +1 ve negatif örneklerde -1 dir. Lagrange yöntemi kullanılarak bu eniyileme problemi ikili forma dönüştürülebilir ki bu, amaç fonksiyonu Ψ sadece bir dizi α_i Lagrange çarpanlarına bağlı olan bir Kuadratik Programlama (Quadratic Programming – QP) problemine dönüşür. Amaç fonksiyonu N eğitim örneklerinin sayısı olmak üzere 6.6 denklemleri ile bulunur.

$$\min_{\bar{\alpha}} \psi(\bar{\alpha}) = \min_{\bar{\alpha}} \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N y_i y_j (\bar{x}_i \bullet \bar{x}_j) \alpha_i \alpha_j - \sum_{i=1}^N \alpha_i \quad (6.6)$$

Yukarıdaki denklemin eşitlikli kısıtları (6.7) ,

$$\alpha_i \geq 0, \forall i, \quad (6.7)$$

ve lineer eşitlik kısıtı denklemleri (6.8) bulunmaktadır.

$$\sum_{i=1}^N y_i \alpha_i = 0, \quad (6.8)$$

Burada her bir Lagrange çarpanı ve her eğitim örneği arasında bire bir bağıntı mevcuttur. Lagrange çarpanı belirlendikten sonra, normal vektörü \vec{w} ve eşik değeri b Lagrange çarpanından 6.9'daki gibi türetilir:

$$\vec{w} = \sum_{i=1}^N y_i \alpha_i \vec{x}_i, \quad b = \vec{w} \bullet \vec{x}_k - y_k \text{ bazı } \alpha_k > 0 \text{ için} \quad (6.9)$$

\vec{w} değeri kullanılmadan önce denklem (6.9) dan eğitim verileri ile hesaplanabildiği için, doğrusal bir SVM değerlendirmek için gerekli olan hesaplama miktarı , sıfırdan farklı destek vektörlerinin sayısına sabitlenir.

Tabi ki, tüm veri kümeleri doğrusal ayrılabilir değildir. Pozitif örneklemeleri negatiften ayıran hiperdüzlemler olmayabilir. Yukarıdaki denklemde, pozitif ve negatiflerin tam olarak ayrılamayacağı bir durum için sonsuz bir çözüme karşılık gelebilir. Fakat, 1995'te Cortes & Vapnik [28] orijinal eniyileme açıklaması 6.5'e bir değişiklik önermiştir, ki bu örnek başarısızlığı ile doğru kenar uzaklığına ulaşabilmeye imkan kılar. Bu değişiklik:

$$\min_{\vec{w}, b, \xi} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^N \xi_i, \quad y_i = (\vec{w} \bullet \vec{x}_i - b) \geq 1 - \xi_i, \forall i \quad (6.10)$$

Burada ξ_i marjın hatalarına olanak tanıyan ihmal değişkenler, C ise geniş marjinleri küçük marjın hataları sağlayan parametredir. Bu yeni eniyileme problemi ikili forma aktarıldığında, (6.7) ve (6.8) deki sınır değerleri aşağıdaki (6.11) gibi değişir.

$$0 \leq \alpha_i \leq C, \quad \forall i \quad (6.11)$$

Bu aşamadan sonra ξ_i değişkeni ikili formülasyonda bulunmamaktadır. SVM'ler doğrusal olmayan sınıflandırıcılar arasında genelleştirilebilir. Doğrusal olmayan SVM çıktısı yine Lagrange çarpanları ile hesaplanabilir [30]. Fakat burada bundan bahsedilmeyecektir.

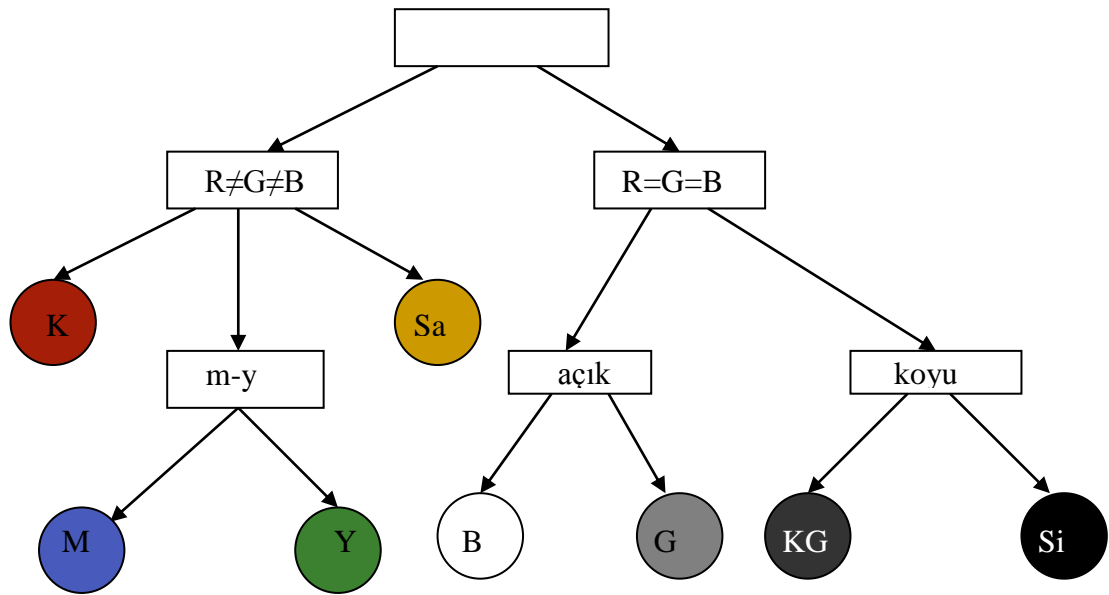
6.5 Karar Ağacı Oluşturma

Karar Ağacı olası tüm eylemlerin yönlerini, eylemlerin yönlerine etkisi olabilecek tüm olası faktörleri ve tüm bu faktörlere dayanan her bir olası sonucu, verilere bağlı olarak değerlendiren iş kolaylaştırıcı bir karar mekanizmasıdır [31].

Renk tanıma analizinde farklı renkleri farklı sınıflandırıcılar ve farklı öznitelik vektörleri ile sınıflandırmaya kalkarsak daha başarılı bir sonuç elde edilir mi diye oluşturulmuştur. Aynı zamanda, ID3 algoritması kullanarak en kısa karar ağacını oluşturup doğru dallanmaya doğru öznitelik – sınıflandırıcı çiftlerini yerleştirip daha yüksek test başarısı araştırıldı. Ağaç oluşturmak için aşağıdaki iki yaklaşım kullanılmıştır.

6.5.1 Tahmini yaklaşım

Tahmini yaklaşımda, renkler her seviyede benzer özelliklerine göre dallandırılmışlardır. Örneğin ilk seviyede $R \neq G \neq B$ (renkli) ve $R=G=B$ (renksiz) olmak üzere iki dala ayrıldı. İkinci seviyede $R=G=B$ karar düğümü kırmızı, sarı yapraklarına ve mavi-yeşil düğümüne, $R \neq G \neq B$ karar düğümü ise açık ve koyu düğümlerine ayrılmıştır. Son seviyede de mavi-yeşil düğüm mavi ve yeşil yapraklarına, açık düğümü beyaz ve gri, koyu düğüm de siyah ve koyu gri yapraklarına dallanmaktadır. Aşağıda Şekil 6.2’de yapılan dallanma açıkça görülmektedir.



Şekil 6.3: Tahmini yaklaşım ile oluşturulan karar ağacı

Her düğümde (ayırma işlemi öncesi), her ROI, her öznitelik ve her sınıflandırma yöntemi için renkler şekilde gösterildiği gibi sınıflandırılmıştır. Elde edilen sınıflandırma başarıları sonucunda her dallanma için en iyi piksel ve histogram tabanlı öznitelik vektörlerine ulaşılmaktadır.

Tahmini karar ağacı yönteminde bir rengin C_i sınıflandırılma başarısı $P(C_i)$, renk düğümünden başlayarak köke ulaşana kadar karşılaşılan her seviyede (L_j) dallanma düğümlerindeki B_j tanıma başarıları $P(B_j)$ birbirleriyle çarpılarak (6.12)'deki gibi elde edilir [61].

$$P(C_i) = \prod_{j=1}^L P(B_j) \quad (6.12)$$

Sistemin toplam başarısı $P(C)$ ise her bir rengin renk tanıma oranlarının ağırlıklı ortalaması olarak bulunur.

$$P(C) = \frac{1}{N} \sum_{i=1}^N P(C_i) \quad (6.13)$$

6.5.2 Hesaplamalı yaklaşım

Hesaplamalı yaklaşımda karar ağacını oluşturmak için sezgisel yaklaşımlar değil eğitim kümesi ile karar ağacı öğrenmesi uygulanmaktadır. Bunun için Iterative Dichotomiser 3 (ID3) algoritması kullanılır. Bu çalışmada B. J. Ross Quinlan ID3'ü ilk olarak Sydney Üniversitesi'nde geliştirmiştir [32]. ID3, Kavram Öğrenme Sistemi algoritmasına (Concept Learning System - CLS) dayanır. Temel CLS algoritması bir C eğitim kümesine uygulanırsa [31]:

Adım 1: C kümesindeki tüm örnekler pozitif (+) ise; EVET düğümünü yarat ve dur. C 'deki tüm örnekler negatif (-) ise, HAYIR düğümünü yarat ve dur. İki durum da sağlanmıyorsa, v_1, \dots, v_n değerleri ile bir F özelliği seç ve bir karar düğümü yarat.

Adım 2: C içindeki tüm örnekleri V 'nin değerleriyle uyumlu bir şekilde C_1, C_2, \dots, C_m şeklindeki alt kümelerine ayır.

Adım 3: Algoritmayı özyinelemeli olarak tüm C_i alt kümelerine uygula

ID3 algoritması eklediği buluşsal yöntem ile CLS algoritmasını iyileştirmektedir. ID3 algoritması, eğitim örneklerinin öznitelikleri arasında tarama yaparak verilen örneği en iyi şekilde ayıran özniteliği açığa çıkartır. Eğer bu öznitelik eğitim kümesini kusursuz olarak sınıflandırıyorsa, ID3 durur; değil ise ayrıştırılmış n (n : bir

özniteliğin alabileceği olası değerlerin sayısı) alt kümenin üzerine özyinelemeli olarak koşarak bunlara ait “en iyi” özniteliği bulana kadar koşar. Algoritma “greedy” bir arama kullanır. Böylelikle, algoritma, en iyi özniteliği seçtikten sonra önceki seçimlerine tekrar bakmaz [31].

ID3 artan olmayan bir algoritmadır, yani sınıflarını sabit bir eğitim örnekleri kümesinden türetir. Artan bir algoritma ise, mevcut kavram tanımını, gerektiği durumda yeni bir örnekleme ile düzeltir. ID3 tarafından yaratılan sınıflar tümevarımsaldır. Bunun anlamı, küçük bir eğitim kümesi için ID3 tarafından yaratılan sınıfların ileride oluşacak tüm örnekler için de çalışması beklenir. Bilinmeyenlerin dağılımı test edilen vakalarla aynı olmalıdır. Tüme varımsal sınıflarının her vaka için çalışması kesin olarak ispatlanamaz çünkü sonsuz sayıda örneği sınıflandırıyor olabilirler [33]. Unutulmamalıdır ki, ID3 (ya da herhangi bir tüme varımsal algoritma) veriyi yanlış sınıflandırabilir.

Karar ağaçları genellikle Evet – Hayır (iki sınıflı) problemlerine çözüm bulmak için üretilirler. Bizim sistemimizdeki karar ağacının çıktısı 7 renk sınıfıdır. O nedenle, her karar düğümündeki dallanma sayısı 7 olacaktır.

ID3 hangi özneniteğin *en iyi* olduğuna bir istatistiksel özellik olan **bilgi kazancını** kullanarak karar vermektedir. Kazanç, seçilen bir özniteliğin eğitim örneklerini hedeflenen sınıflara ne kadar iyi şekilde ayırabildiğini ölçer. En fazla bilgi içeren, sınıflandırmaya en çok yarayan bilgileri içeren öznitelik seçilir. Kazancı tanımlayabilmek için, bilgi teorisinden entropi kavramını ödünç alıp tanımlamamız gerekmektedir. Entropi, bir özniteliğin içerdiği bilgi miktarını ölçer.

C sayıda çıktısı olan bir S topluluğunu ele alalım; $p(i)$ S 'nin i . sınıfına ait olma oranını, C' lerin toplamı ve $\log_2 2$ tabanında logaritma olmak üzere

$$E(S) = \sum_1^N - p(i) * \log_2 p(i) \quad (6.14)$$

Burda S 'nin tek bir öznitelik değil, tüm örnek kümesi olduğuna dikkat edilmelidir. Eğer S 'nin tüm elemanları aynı sınıfa ait ise (veri kusursuz şekilde sınıflandırılmışsa) Entropi 0'dır. Entropinin değer aralığı [0 1] arasında değişmektedir. Entropi 0 ise veri kümesi kusursuz sınıflandırılmış, entropi 1 ise veri kümesi tamamıyla rastlantısalıdır.

S örnek kümesinin A özniteliği üzerindeki Bilgi Kazancı (Information Gain) olan $IG(S,A)$ aşağıdaki şekilde (6.15) tanımlanır [32]:

$$IG(S, A) = E(S) - \sum (|S_v|/|S|) * E(S_v) \quad (6.15)$$

Her bir öznitelik için, kazanç hesaplanarak, en yüksek kazanç karar düğümünde kullanılır. Bunun sonucunda karar düğümündeki örneklerin tamamının doğru sınıflara atanmış olup olmadıklarına karar verilir.

ID3 bir çok ticari uygulama ve tüme varımsal pakette yerini almıştır. Bunlardan bazıları: tıbbi teşhis, kredi risk değerlendirme, cihaz arızalarını ve sebeplerini araştırma, soya fasulyesi hastalıklarını sınıflandırma ve web arama sonuçlarının sınıflandırmaya yönelik uygulamalardır [31].

7. YAPILAN TESTLER VE SONUÇLARI

Bu çalışmada, araç renk tanımada en iyi sonucu veren ROI, öznitelik seçimi, öznitelik vektörü oluşturma yöntemi ve sınıflandırıcı kombinasyonunun elde edilmesi amaçlanmıştır. Böyle bir kombinasyon bulunabilir mi diye, eğitim ve test kümelerini kullanarak çok sayıda test yapılarak bu problem araştırılmıştır. Elimizde

- iki çeşit ROI
- piksel tabanlı öznitelik vektörü için 16 adet öznitelik elemanı
 - $R_n, G_n, H, S, V, L, a, b, I_1, I_2, I_3, L_1, L_2, L_3, \xi_1$ ve ξ_2
- histogram tabanlı öznitelik vektörü için 2 adet tek boyutlu, 8 adet de iki boyutlu histogramları
 - Gray1, Gray2 → tek boyutlu histogram
 - $R_n-G_n, H-S, H-V, L-a, L-b, a-b, L_1-L_2, L_1-L_3$ → iki boyutlu histogram
- dört adet sınıflandırma yöntemi
 - KNN, TM, ANN ve SVM

vardır. Bu nedenle bütün denemeleri yapmak üç ay sürmüştür.

Eğitim ve test kümeleri iki ayrı ROI seçimi (kaput parçası ROI, yarım araç ROI) için ayrı ayrı eğitim ve test kümeleri oluşturulmuştur. Farklı öznitelik seçimi için farklı veri kümeleri kullanılmaktadır. Piksel tabanlı öznitelik vektörü oluştururken “kaput parçası ROI” yöntemi ile oluşturulan eğitim ve test görüntüleri kullanılırken, histogram tabanlı öznitelik vektörü oluştururken “yarım araç ROI” bölge seçme yöntemi kullanılmaktadır.

7.1 Renk Tanıma Başarısını Düşüren Sebepler

Araç renk tanıma sistemi ile araçları sınıflandırırken renk tanıma başarısını düşüren çeşitli problemlerle karşılaşılmaktadır. Bu problemlerin ne kadarının hangi sebepten kaynaklandığını anlamak ve akabinde problemlere yönelik çözümler üretmek için bir araştırma yapılmıştır. Bu araştırmaya göre önceden etiketlenmiş 1200 resim test edilerek bunlardan 319 tane araç yanlış sınıflandırılmıştır. Çizelge 7.1’de de görüldüğü gibi araçların 18 tanesi yanlış veya yetersiz bölge seçiminden, 4 tanesi yanlış etiketlenmiş olmaktan, 60 tanesi renginin gözle bile ayırt edilmeyecek durumda olması, 237 tanesi de görüntülerin kamera açısı, ışık, gölge, toz vb. gibi etkenlerden etkilenmesi sebebiyle yanlış sınıflandırılmışlardır.

Çizelge 7.1 : Yanlış sınıflandırmanın sebepleri

	Yanlış Bölge Seçimi	Renk değişimleri	
		Renk (*) – gözle bile ayırt edilemeyenler	Işık- gölge vs. kaynaklı
Adet	18	60	237
Oran	% 5.714	% 19.047	% 75.238

Problem kaynaklarından yanlış bölge seçimini minimuma indirmek için EUKS algoritması kullanılmaktadır. EUKS peş peşe gelen aynı araca ait görüntülerin optimum aydınlıkta olanını da seçmesi açısından ışık-gölge kaynaklı hataların azaltılmasında da fayda sağlamaktadır. Üçüncü etken kaynaklı hataları azaltmanın başka yolları da yansıma ve bakış açısından bağımsız öznitelik vektörleri oluşturmaktır. Bunun için, birçok renk uzayını kullanarak en iyi sonuç veren öznitelik vektörlerini aşağıda anlatılan testlerle elde etmeye çalıştık. 2. Sebepten kaynaklanan hatalı sınıflandırmaları minimuma indirmek ise pek olası değildir.

7.2 En İyi Öznitelik Vektörünün Elde Edilmesi İçin Yapılan Testler

7.2.1 Piksel tabanlı öznitelik vektörü testleri

Piksel tabanlı öznitelik vektörü oluşturmak için elimizde 16 renk bileşeni vardır. Elimizdeki bu on altı renk bileşeninden biri, bazıları veya hepsi bir araya gelerek, en iyi renk tanıma başarısını veren [1 16] aralığında değişiklik gösteren bir öznitelik vektörü araştırılmaktadır.

Bu öznitelik vektörünün hangisi olduğunu bulmak için 16 renk bileşeninin 2^{16} kombinasyonu oluşturulmuş ve bu öznitelik vektörleri için tek tek hem TM hem de ANN sınıflandırıcıları kullanılarak renk tanıma başarıları hesaplanmıştır. En yüksek başarılı öznitelik vektörlerinin, $k = 3$ için KNN sınıflandırıcısı için de doğru sınıflandırma oranları elde edilmiştir.

Çizelge 7.1’de piksel tabanlı öznitelik vektörü yöntemi, kaput parçası ROI veri kümesi ve TM sınıflandırıcısı kullanılarak elde edilen en iyi on beş renk tanıma başarı oranlarını göstermektedir. TM sınıflandırıcısında uzaklık hesabı olarak Euler Uzaklığı (L2) kullanılmıştır. Çizelgede de görüldüğü üzere, TM sınıflandırıcısı için en yüksek başarılı sonucu %79,2 ile iki öznitelik vektörü paylaşmaktadır. Bunlar “ $G_n V a b I_1 I_3 \xi_2$ ” and “ $R_n G_n V L a b I_1 I_2 I_3 \xi_2$ ” dizilimindeki piksel tabanlı yaratılmış öznitelik vektörleridir.

Çizelge 7.2 : TM sınıflandırıcısı için en iyi Piksel Tabanlı Öznitelik Vektör sonuçları.

ÖV Metodu	TM	Eğitim Kümesi	Öznitelik Vektörü
pixel	79,2	Kaput Parça	$R_n G_n V L a b I_1 I_2 I_3 \xi_2$
pixel	79,2	Kaput Parça	$G_n V a b I_1 I_3 \xi_2$
pixel	79,1	Kaput Parça	$R_n G_n V a I_1 I_2 I_3 \xi_2$
pixel	79	Kaput Parça	$R_n V a I_1 I_2 I_3 \xi_2$
pixel	78,9	Kaput Parça	$R_n V L a b I_1 I_2 I_3 \xi_2$
pixel	78,9	Kaput Parça	$R_n G_n V a b I_1 I_3 \xi_2$
pixel	78,9	Kaput Parça	$V a b I_1 I_3 \xi_2$
pixel	78,9	Kaput Parça	$G_n V a I_1 \xi_2$
pixel	78,9	Kaput Parça	$G_n V a I_3 \xi_2$
pixel	78,9	Kaput Parça	$G_n V a I_1 I_3 \xi_2$
pixel	78,9	Kaput Parça	$R_n V a b I_1 I_3 \xi_2$
pixel	78,9	Kaput Parça	$R_n G_n V a b I_1 I_2 I_3 \xi_2$
pixel	78,9	Kaput Parça	$G_n V b I_1 I_2 I_3$
pixel	78,8	Kaput Parça	$V a I_1 I_3 \xi_2$
pixel	78,8	Kaput Parça	$R_n V a b I_1 I_2 I_3 \xi_2$
pixel	78,8	Kaput Parça	$G_n V a b I_1 I_2 I_3 \xi_2$
pixel	78,8	Kaput Parça	$R_n G_n V L a I_1 I_2 I_3 \xi_2$

Çizelge 7.3’de KNN k=3 için en başarılı on sonucu veren piksel tabanlı öznitelik vektörleri karşılık düşen TM başarı oranlarıyla birlikte verilmiştir. Çizelgede de görüldüğü üzere k = 1 için elde edilen KNN başarı sonuçları (TM) k=3 için elde edilen sonuçlardan en iyi başarı sonucu hariç daha yüksektir. 3NN için en iyi sonuç %78.50 başarı oranıyla “*Gn L b I1 I2 I3*” öznitelik vektörü tarafından alınmıştır.

Çizelge 7.3 : TM ve KNN sınıflandırıcıları için en iyi Piksel Tabanlı Öznitelik Vektörü.

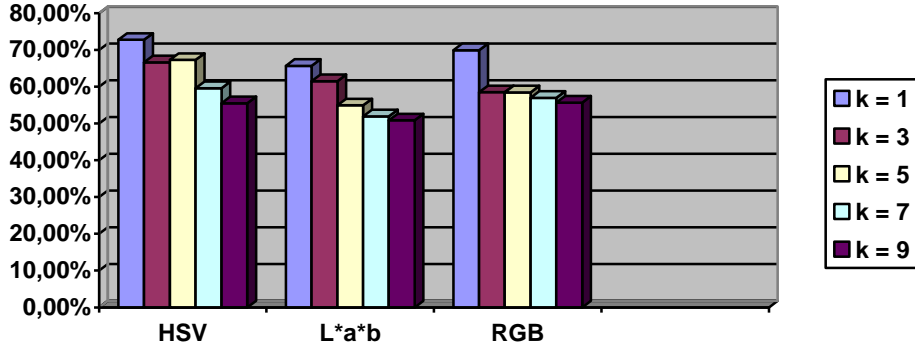
ÖV Metodu	KNN Performans	TM Performans	Eğitim kümesi	Özniteik Vektörü
Piksel	78.50	78.40	Kaput parça	Gn L b I1 I2 I3
Piksel	78.40	78.40	Kaput parça	Rn Gn L b I1 I2 I3
Piksel	78.10	78.4	Kaput parça	Rn L b I1 I3
Piksel	78.00	78.6	Kaput parça	V a b I1 I3
Piksel	77.80	78.9	Kaput parça	Gn V b I1 I2 I3
Piksel	77.80	78.5	Kaput parça	Rn V a b I1 I3
Piksel	77.80	78.5	Kaput parça	Rn V a b I1
Piksel	77.80	78.5	Kaput parça	Rn V a b I3
Piksel	77.60	78.4	Kaput parça	Gn V a b I1 I3
Piksel	77.50	78.6	Kaput parça	Rn Gn V L a b I1 I3

Daha önceden yapılan denemelerde KNN sınıflandırıcısının k=1 için alınan sonuçların k = 3 , k = 5 ve k = 7 den daha başarılı olduğu görülmüştür. Bu nedenle çizelge 7.2’de sadece k = 1 ve k = 3 ve için sonuç bulunmaktadır. Aşağıda listelenen çizelge 7.4’te ise KNN sınıflandırıcısında k değerini belirlemek için HSV ve L*a*b renk uzayları üzerinde k = {1, 3, 5, 7, 9} için yapılmış testlerin sonuçları yer almaktadır.

Çizelge 7.4 : KNN sınıflandırıcısında k değıştikçe farklı renk uzaylarının başarı oranları

K	HSV	L*a*b	RGB
1	%72,777	%65,706	%69,921
3	%66,616	%61,534	%58,515
5	%67,307	%54,954	%58,506
7	%59,583	%51,924	%57,032
9	%55,515	%50,865	%55,719

Aşağıdaki grafikte, çizelgede gösterilen değerlerin grafik olarak gösterilmesidir. Yatayda her renk uzayı (HSV, Lab, RGB) için 5 ayrı k değeri sütun olarak temsil edilmektedir. KNN modelinin k değerlerine denk düşen sınıflandırma başarıları dikeyde gösterilmiştir.



Şekil 7.1: k değıştikçe KNN modelinin başarısını gösteren grafik

Piksel tabanlı öznitelik vektörlerinin ANN sınıflandırıcısı ile elde edilen başarı sonuçları Çizelge 7.5'te en yüksekten başlayarak en düşüğe doğru sıralanmışlardır. ANN sınıflandırma yönteminde "*Rn Gn H S L b I₁ I₂ L₁ L₃ ξ₂*" öznitelik vektörünün en yüksek başarı oranı %82,1'i sağladığı görülmektedir. Belirtilen sonuçlar baz alınarak, piksel tabanlı öznitelik vektörleri için ANN sınıflandırma yönteminin araç renk tanımada TM sınıflandırıcısından daha başarılı olduğu söylenebilir.

Çizelge 7.5 : ANN sınıflandırıcısı için en iyi Piksel Tabanlı Öznitelik Vektörü

ÖV Metodu	ANN	Eğitim kümesi	Öznitelik Vektörü
Piksel	82,1	Kaput Parça	Rn Gn H S L b I1 I2 L1 L3 k1
Piksel	81,9	Kaput Parça	Rn H S L a I1 I2 L1 L3 ksi2
Piksel	81,8	Kaput Parça	H S L b L1 ksi1
Piksel	81,7	Kaput Parça	H S L a L1 L3 ksi1
Piksel	81,7	Kaput Parça	Rn Gn H S L b I1 L1 L3 k1
Piksel	81,7	Kaput Parça	Rn Gn H S L b I3 L1 L3 k1
Piksel	81,7	Kaput Parça	Rn Gn H S L a b I1 L1 L3 k2
Piksel	81,7	Kaput Parça	Rn Gn H S L a b I3 L1 L3 k2
Piksel	81,7	Kaput Parça	Rn H S L I1 I3 L1 L3 ksi2
Piksel	81,7	Kaput Parça	Rn Gn H S L a b I1 I3 L1 L3 k2
Piksel	81,6	Kaput Parça	Gn H S L I1 L1 L3 k2
Piksel	81,6	Kaput Parça	H L a b I1 L1 L3 k1
Piksel	81,5	Kaput Parça	Rn Gn H V a b I1 I2 k2
Piksel	81,4	Kaput Parça	H S V L b L1 L3
Piksel	81,4	Kaput Parça	Rn H S L I1 L1 L2 L3

7.2.2 Histogram tabanlı öznitelik vektörü testleri

Histogram tabanlı öznitelik oluşturma yönteminde piksel tabanlı öznitelik vektörü tekniğinden farklı olarak, elimizde farklı renk bileşenlerinden oluşturulan 10 adet histogram bulunmaktadır. Bu bir ve iki boyutlu histogramların bir veya birkaçının art arda sıralanmasıyla 2^{10} tane farklı öznitelik vektörü elde edilebilmektedir. Aynı zamanda, histogramların bin sayısını değişken tutarak her birinden bileşen başına 8, 16 ve 32 bin gelecek şekilde histogramlar oluşturulmuştur. Örnek vermek gerekirse, $L*a*b$ renk uzayının L ve a bileşenlerinden iki boyutlu L-a histogramını elde etmek istediğimizde her boyutu 8, 16 ve 32 parçaya ayırarak aynı özellikleri taşıyan fakat değişik boyutlarda 3 ayrı histogram oluşturmuş olmaktadır. Oluşan öznitelik vektörleri en küçük histogram boyu olan 8 ile tüm en büyük boyutlu histogramların bir araya gelmesiyle $2^{13}+2^6$ ($8 \times 32^2 + 2 \times 32 = 2^{13}+2^6$) değeri arasında bir uzunluğa sahip olurlar. Yapılan çalışmada, oluşturulması mümkün olan tüm öznitelik vektörlerinin, yukarıda belirtilen, üç ayrı bin boyutu için de renk tanıma başarı oranları hesaplanmıştır.

Buna göre, en yüksek başarı sonucunu veren histogram tabanlı öznitelik vektörünü bulmak için, test algoritması 2^{10} defa TM ve ANN modelleri için ayrı ayrı çalıştırılmıştır. Öznitelik vektörlerinin sayısı çok olduğu için SVM ve KNN (k=3) sınıflandırıcıları sadece en yüksek başarılı en fazla 20 özniteliği sınıflandırmışlardır.

Yarım araç ROI veri kümesi histogram tabanlı öznitelik vektörü oluşturmak için kullanılmıştır. Yapılan denemelerde histogramların bin sayısı, histogramların normalize edilmesi, ve öznitelik vektörünün birim vektöre çevrilip çevrilmemesi olmak üzere üç ek koşul için başarı sonuçlarının nasıl değiştiği araştırılmıştır. Bu koşullar deneylerde histogram tabanlı öznitelik vektör testlerinde giriş parametresi olarak kullanılmışlardır. Çizelge 7.6, TM sınıflandırıcısı için en yüksek başarıyı veren histogram özniteliğinin yukarıda belirtilen boyut normalizasyonu, birim vektör oluşu ve bin sayısı parametrelerinden nasıl etkilendikleri gösterilmeye çalışılmıştır. Aşağıda hem TM hem de ANN için başarı oranlarının nasıl değiştikleri görülmektedir.

Çizelge 7.6 : Farklı boyut normalizasyonu, birim vektör ve bin sayısı parametrelerinin TM ve ANN sınıflandırıcıları için başarı sonuçları

Boy Norm	Bin Sayısı	Birim Vektör	TM	ANN	Öznitelik vektörü
0	8	0	82,016	46,739	RnGn SV Gray2 L-b L1-L2
0	8	1	82,016	46,739	RnGn SV Gray2 L-b L1-L2
1	8	0	83,3	82,016	RnGn SV Gray2 L-b L1-L2
1	8	1	83,3	82,016	RnGn SV Gray2 L-b L1-L2
0	32	0	79,15	50,593	RnGn SV Gray2 L-b L1-L2
0	32	1	79,15	50,593	RnGn SV Gray2 L-b L1-L2
1	32	0	81,522	81,719	RnGn SV Gray2 L-b L1-L2
1	32	1	81,522	81,719	RnGn SV Gray2 L-b L1-L2
0	8	0	79,051	61,462	Gray2 a-b L-a L-b L1-L2
0	8	1	79,051	61,462	Gray2 a-b L-a L-b L1-L2
1	8	0	79,249	80,237	Gray2 a-b L-a L-b L1-L2
1	8	1	79,249	80,237	Gray2 a-b L-a L-b L1-L2
0	32	0	80,039	58,597	Gray2 a-b L-a L-b L1-L2
0	32	1	80,039	58,597	Gray2 a-b L-a L-b L1-L2
1	32	0	81,225	83,004	Gray2 a-b L-a L-b L1-L2
1	32	1	81,225	83,004	Gray2 a-b L-a L-b L1-L2

Histogram boyu normalizasyonu ve birim vektörü etkenleri algoritmada mantıksal değerler olarak ifade edildikleri için çizelgede de işlem uygulanmışsa 1, uygulanmamışsa da 0 değerleri ile ifade edilmektedirler. “*RnGn SV Gray2 L-b L1-L2*” öznitelik vektörünün 8 binlik ve 32 binlik histogramlar için ayrı ayrı sonuçlar alınmıştır. Buna göre, öznitelik vektörlerin birim vektöre dönüştürülmesi iki sınıflandırıcı için de hiçbir etki yaratmazken, normalize edilmiş histogramlardan oluşan öznitelik vektörleri normalize edilmemiş hallerine oranla daha başarılı sonuçlar vermektedirler. Özellikle, kullanılan sınıflandırıcı modeli ANN sınıflandırıcısı ise histogramların mutlaka normalize edilmesi gerekmektedir, çünkü normalize edilmiş ile edilmemiş öznitelik vektörü arasında %40'lara varan bir farktan söz edilmektedir. TM sınıflandırıcısında ise, bu fark ANN modeline göre çok düşüktür ve histogramların normalize edilmiş olması ancak %2 – %3 oranında bir başarı artışı sağlamaktadır.

Histogramların bin sayısının da renk tanıma başarısını etkilediği gözlenmektedir. Öyle ki, boy normalizasyonunun 1 ve birim vektör 0 olduğu satırlara bakarsak, 8 binlik histogramlardan oluşan öznitelik vektörlerinin TM ve ANN başarıları sırasıyla %83.3, %82.16 iken 32 binlik histogramlardan oluşan aynı öznitelik vektörünün TM ve ANN başarı oranları %81,52 ve %82,719 dur. Yapılan tüm testler göz önüne alındığında 8 binlik öznitelik vektörlerin 16 ve 32 binlik histogramlı öznitelik vektörlerine göre daha yüksek başarı oranları verdikleri saptanmıştır.

Çizelge 7.7, histogram tabanlı öznitelik vektörlerinin TM (KNN, k=1) sınıflandırıcısından geçirilmesiyle elde edilen renk tanıma başarı oranlarını belirtmektedir. Çizelgede, özvektörlerin farklı histogram bin sayıları için sonuçlar yer almaktadır. TM sınıflandırma modeli için en iyi sonucu veren öznitelik vektörü %83.3 başarı oranı ile 8 binlik histogramlardan oluşan “*RnGn SV Gray2 L-b L1-L2*” öznitelik vektörüdür. Aslında, çizelgede de görüldüğü gibi, aynı başarı oranını veren 4 tane öznitelik vektörü vardır. En iyi olarak bunun seçilmesinin sebebi diğer öznitelik vektörlerine nazaran boyunun daha kısa olmasındandır. ART modeli gerçek zamanlı bir sistem içinde çalışacağından dolayı, renk tanıma işleminin optimum düzeyde hızlı çalışıp doğru sonuç vermesi çok önemlidir.

Histogram tabanlı öznitelik vektörlerinin ANN sınıflandırıcı modeli tarafından da farklı bin sayıları için performans değerleri elde edilmiştir. Bu performans değerleri, karşılık gelen öznitelik vektörü 8 binlik histogram tabanlı öznitelik vektörleri için en iyi 15 tanesi ve bu özniteliklere karşılık gelen 16 ve 32 binlik histogram öznitelik vektörlerinin ANN başarıları Çizelge 7.8’de gösterilmiştir. En iyi renk tanıma başarısını veren “*SV SH Gray2 a-b L-a L-b*” öznitelik vektörü araçların renklerini **%83,5** oranında doğru tanımıştır. Bu çizelgeden de görüleceği gibi histogram sayısının 8 olmasının renk tanıma başarısına katkısı 16 ve 32 bine göre daha fazladır. Aşağıda yer alan sonuçlar normalize edilmiş öznitelik vektörlerinin sınıflandırılmasından elde edilmişlerdir. Yani boyut normalizasyonu parametresi 1 olarak alınmıştır. Ayrıca, ANN modelinde gizli katman için düğüm sayısı 10 olarak belirlenmiştir.

Çizelge 7.7 : TM- 8 ve 32 binlik histogram tabanlı öznelik vektörleri sonuçları
Norm: Normalizasyon

ÖV Metodu	Norm	Bin Sayısı	KNN (TM)	Öznelik Vektörü
histogram	1	8	83,3	RnGn SV SH Gray1 Gray2 a-b L-b L1-L2 L1-L3
histogram	1	8	83,3	RnGn SV Gray1 Gray2 L-b L1-L2 L1-L3
histogram	1	8	83,3	SV Gray1 Gray2 L-b L1-L2 L1-L3
histogram	1	8	83,3	RnGn SV Gray2 L-b L1-L2
histogram	1	8	83,202	RnGn SV SH Gray1 Gray2 L-a L1-L2 L1-L3
histogram	1	8	83,004	RnGn SH Gray1 Gray2 L-a L-b L1-L2 L1-L3
histogram	1	8	83,004	SV SH Gray1 Gray2 a-b L-b L1-L2 L1-L3
histogram	1	8	83,004	SV Gray2 L-b L1-L2
histogram	1	8	82,905	RnGn SV SH Gray1 Gray2 L-b L1-L2 L1-L3
histogram	1	8	82,905	SV SH Gray1 Gray2 L-a L1-L2 L1-L3
histogram	1	8	82,905	SV Gray1 L-b L1-L3
histogram	1	8	82,905	RnGn SV SH Gray1 L-a L1-L3
histogram	1	8	82,905	SV SH L-a L-b L1-L2
histogram	1	8	82,806	RnGn SV SH Gray1 Gray2 a-b L-a L-b L1-L2 L1-L3
histogram	1	8	82,806	RnGn SV SH Gray1 Gray2 L-a L-b L1-L2 L1-L3
histogram	1	32	83,004	RnGn SH Gray1 L-a L-b
histogram	1	32	82,905	RnGn SH L-a
histogram	1	32	82,905	RnGn SV SH L-b
histogram	1	32	82,905	SH Gray1 L-a L-b
histogram	1	32	82,806	RnGn L-b
histogram	1	32	82,806	RnGn SV Gray1 L-b
histogram	1	32	82,806	SH Gray1 L-b
histogram	1	32	82,708	RnGn SH Gray1 L-b
histogram	1	32	82,708	RnGn SV Gray1 Gray2 L-b
histogram	1	32	82,708	SV SH L-a L1-L2
histogram	1	32	82,609	SV SH L-b
histogram	1	32	82,609	RnGn Gray2 L-b
histogram	1	32	82,51	RnGn SV Gray1 Gray2 a-b
histogram	1	32	82,51	RnGn SV L-b
histogram	1	32	82,411	RnGn SV Gray2

Çizelge 7.8 : ANN: 8, 16 ve 32 binlik histogram tabanlı öznitelik vektörleri sonuçları
*HST: histogram

ÖV Metodu	Boyut Norm	8 bin	16 bin	32 bin	Öznitelik Vektörü
HST	1	83,50	82,12	81,82	SV SH Gray2 a-b L-a L-b
HST	1	83,34	81,82	81,72	RnGn SH Gray1 L-a L-b L1-L2
HST	1	83,30	81,52	82,21	RnGn SV Gray1 a-b L-a L-b
HST	1	83,30	81,92	82,21	SV Gray1 a-b L-a L-b L1-L2
HST	1	83,00	83,00	77,30	SV SH Gray2 L-a L-b L1-L3
HST	1	82,91	81,92	81,72	SV SH Gray1 Gray2 a-b L-b L1-L2
HST	1	82,81	82,02	81,62	SV SH Gray1 L-a L-b
HST	1	82,81	81,03	81,52	RnGn SV SH Gray1 Gray2 L-a L-b
HST	1	82,81	82,31	81,72	RnGn SV SH Gray1 Gray2 a-b L-b L1-L2
HST	1	82,81	82,00	81,72	SV SH Gray2 L-a L-b L1-L2
HST	1	82,81	82,81	81,81	SV SH Gray1 Gray2 a-b L-a L-b L1-L3
HST	1	82,81	82,41	78,00	SV SH Gray1 Gray2 L-a L-b L1-L2 L1-L3
HST	1	82,81	82,71	81,42	SV L-a L-b L1-L2
HST	1	82,71	83,00	82,61	RnGn SH Gray1 Gray2 a-b L-a L-b L1-L2
HST	1	82,61	81,92	82,02	RnGn SH Gray2 L-b L1-L2

Yukarıdaki çizelgede ANN sınıflandırıcısının saklı katmanında düğüm sayısı 10 alınarak elde edilen başarı sonuçlarıdır. Düğüm sayısını değiştirerek daha iyi sonuçlar alır mıyız sorusuna cevap bulmak için histogram öznitelik vektörleri, bin sayıları 8 ve ANN saklı katmanında 12 düğüm olan koşullarda tekrar sınıflandırıldılar. Elde edilen en yüksek 5 başarı oranı Çizelge 7.9’da gösterilmiştir. “**RnGn SH Gray1 L-b L1-L2 L1-L3**” öznitelik vektörü 12 düğüm ile oluşturulan ANN sınıflandırıcısında %83.89 başarı sonucunu sağlamıştır.

Çizelge 7.9 : ANN: en iyi 5 tane histogram tabanlı öznitelik vektörü ve parametreleri

Norma lize	Bin sayısı	ANN Düğüm sayısı	TM	ANN	Eğitim kümesi	Öznitelikler
1	8	12	82,51	83,893	Yarım Araç	RnGn SH Gray1 L-b L1-L2 L1-L3
1	8	12	83,004	83,795	Yarım Araç	RnGn SH Gray1 Gray2 L-a L-b L1-L2 L1-L3
1	8	12	80,731	83,597	Yarım Araç	RnGn SV SH Gray2 a-b L-a L-b
1	8	12	82,312	83,597	Yarım Araç	SV SH Gray2 L-a L-b L1-L3
1	8	12	82,411	83,498	Yarım Araç	RnGn SV Gray2 L-a L-b L1-L2 L1-L3

Yukarıdaki verilerden de anlaşılacağı üzere histogram öznitelik seti ile ANN birlikte kullanıldıkları zaman diğer dört kombinasyondan daha iyi sonuç elde edilmektedir. Ancak TM – histogram özniteliği ikilisi ile arasında %0.5 gibi düşük bir başarı oranı farkı vardır. Fakat yine de ANN ile sınıflandırma yaparken (test), sınıflandırma işlemi TM'ye göre daha hızlı olduğu için ART sisteminde kullanılmak üzere bu sınıflandırıcı – öznitelik ikilisi tercih edilmelidir.

7.2.3 Önceki çalışmalarla en iyi öznitelik vektörünü karşılaştırma

Beak, Park ve Kim'in yapmış oldukları araç renk sınıflandırma çalışmasında (2007) öznitelik vektörü olarak H-S (Hue - Saturation) iki boyutlu histogramı kullanılmış ve SVM ile sınıflandırma yapılmıştır. Kullandıkları veri tabanı sadece beş renkten (kırmızı, mavi, sarı, beyaz, siyah) oluşmakta, bunların arasında birbiriyle karıştırılabilecek gri gibi ara renkler bulunmamaktadır. Her renk için 100 adet araçların önden çekilmiş görüntüleri kullanılmıştır. Bu çalışmada elde edilen sonuç %94.92 gibi yüksek bir başarıdır [8].

Kullandıkları veri tabanına erişemediğimiz için kendi “yarım araç ROI” veri tabanımızdan onların da kullandıkları 5 renge ait eğitim kümesi ve 100'er parçalık test kümesi oluşturuldu. Makalede kullanılan algoritmanın aynısı bu veri tabanı üzerinde denendi ve **%85.15** oranında başarılı sonuçlar elde edilmiştir. Sadece veri tabanındaki değişiklik renk tanıma başarısında %10 değerinde bir düşüş yaşatmıştır.

Aynı veri tabanını ve SVM sınıflandırıcısıyla bu sefer de HS histogramı yerine bizim bulduğumuz en iyi histogram öznitelik vektörünü “*RnGn SV Gray1 Gray2 L-b L1-L2 L1-L3*” kullandığımız zaman elde edilen başarı **%89.31** oluyor.

Dolayısıyla, bu sonuçlara dayanarak, yaptığımız öznitelik seçimi çalışmalarının başarılı olduğunu, daha iyi bir öznitelik vektörü seçmiş olduğumuzu kanıtlamış olmaktadır. SVM ile yapılan sınıflandırmalar bir de TM ve ANN ile yapılmış ve iki durumda da TM ile daha yüksek başarı oranı yakalanmıştır. Çizelge 7.10, yapılan bu ve birkaç tane daha karşılaştırma test sonuçlarını göstermektedir.

Çizelge 7.10 : Histogram özniteliği ile 5 renkli veri tabanı karşılaştırmalı sonuçları

Veri Tabanı	Öznitelik vektörü	Histogram (TM)	Histogram (ANN)	SVM	Makale Sonucu
5 renk 100er VT	SH	%86.535	%85.941	%85.14	%94.92
5 renk oranlı VT	SH	%87,975	%86,392	%85.12	
5 renk 100er VT	En iyi	%88.515	%86.535	%89.30	
7 renk 100er	En iyi	%83,3	%83,893	%83.92	
7 renk oranlı	En iyi	%82.51	%81.423	%83.00	

7.3 Tahmini Ağaç Oluşturma ve Test Sonuçları

Tahmini yaklaşım ile karar ağacı oluşturmak için yapılan denemelerde belirlenen sınıflandırma çıkış sayısı için SVM, ANN, ve TM sınıflandırıcıları en iyi sonuç veren öznitelik vektörlerini bulmak için her düğümde öznitelik sayısı kadar çalıştırıldılar. Elde edilen sonuçlara göre ağaç yapısında da histogram öznitelik vektörlerinin ortadeğer-piksele göre nispeten daha başarılı oldukları gözlemlendi. ANN ve TM sınıflandırıcıları SVM ye göre daha başarılı sonuçlar vermişlerdir. Ayrıca “renkli” dallanmalardaki renk sınıflandırmadan daha iyi performans elde edilmiştir. Şekil 7.2, 7.3 ve 7.4’te elde edilen sonuçlar daha ayrıntılı olarak incelenebilir.

Tahmini ağaç yönteminde sistemin toplam başarısını hesaplamak istenirse, sınıflandırma sonunda her yapraktaki eleman sayısını bir üst seviyedeki anne düğümdeki sınıflandırma başarısıyla çarpılıp köke ulaşana kadar her dallanma düğümünde aynı işlem uygulanır. Bir yapraktan köke ulaşmak o rengin sınıflandırma başarısını vermektedir.

Tüm yapraklardan köke ulaşılnca, yapraklardan köke gelene kadar elde edilmiş tüm oranlar toplanır ve toplam, sınıflandırılan örnek sayısına bölünür. Böylece sistemin genel başarısı elde edilmiş olur.

Örneğin, median ağacında ANN sınıflandırıcısı için sistemin toplam sınıflandırma başarısını hesaplayalım. Tüm yapraklardaki örnek sayısının eşit olduğunu varsayarsak işimiz epey kolaylaşır. Buna göre,

$$P(k) = P(sa) = \frac{99.14}{100} \times \frac{94.37}{100} = \%93.5584$$

$$P(m) = P(y) = \frac{99.14}{100} \times \frac{94.37}{100} \times \frac{93.33}{100} = \%87.318$$

$$P(b) = P(ag) = \frac{94.37}{100} \times \frac{85.03}{100} \times \frac{91.62}{100} = \%73.5184$$

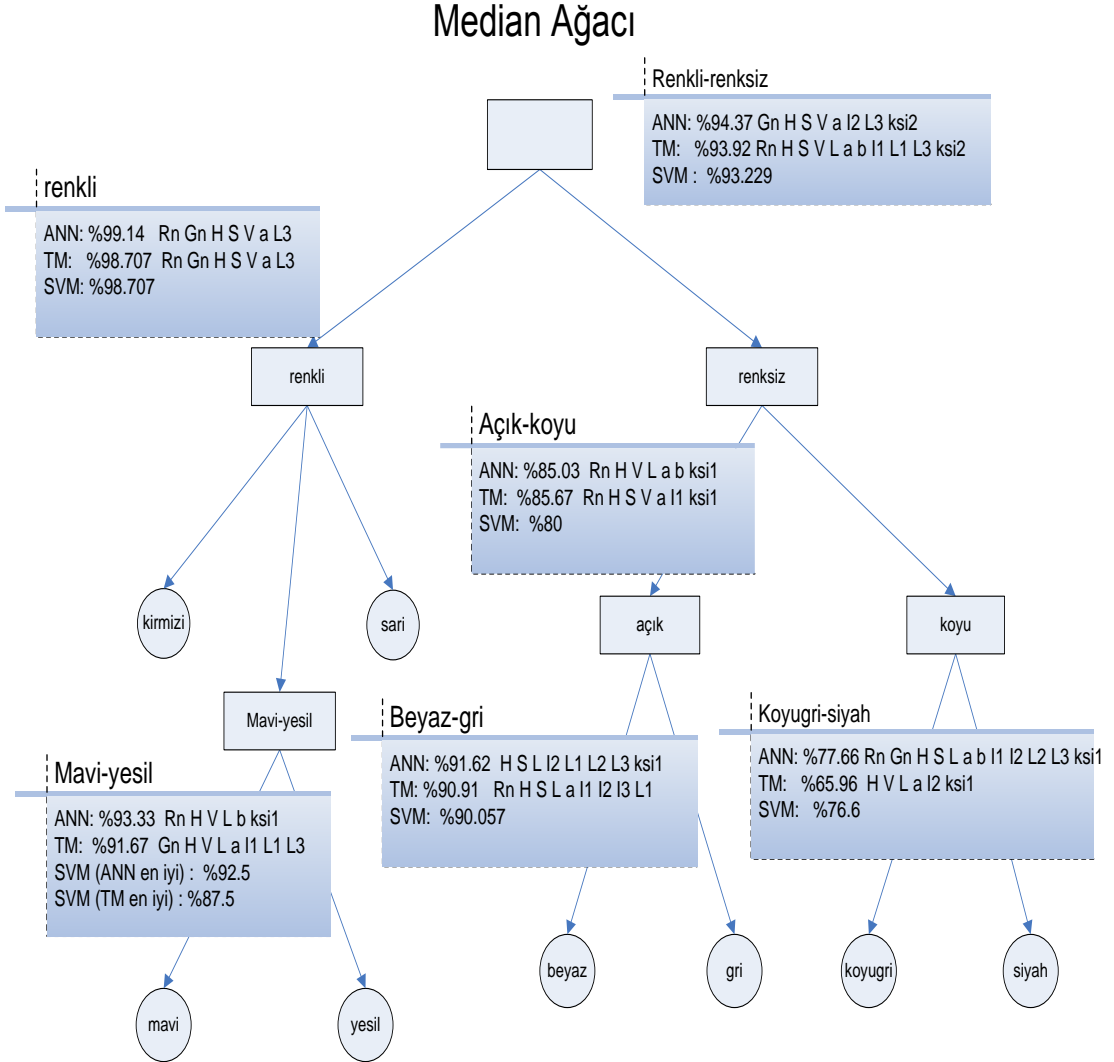
$$P(s) = P(kg) = \frac{94.37}{100} \times \frac{85.03}{100} \times \frac{77.66}{100} = \%62.2684$$

$$\text{Toplam Başarı} = \frac{2 \times 93.5584 + 2 \times 87.318 + 2 \times 73.5184 + 2 \times 62.2684}{8}$$

$$\text{Toplam Başarı} = 79,1658$$

olarak hesaplanır. Aynı şekilde TM sınıflandırıcısı için karar ağacındaki başarı hesaplandığında %75.98 oranını verdiği görülmüştür. Her düğümde en başarılı yöntem alındığında ANN ve TM sınıflandırıcılarına göre daha yüksek bir renk sınıflandırma başarısı elde edilmiştir: %79.434.

7.3.1 Orta Değer Ağacı test sonuçları



Şekil 7.2 : Piksel öznitelik vektörleri ve her düğümdeki sınıflandırma başarıları

Karar ağacını oluştururken *koyu gri* ve *siyah* yaprakları yerine *koyu* düğümünü dallandırmadan bir yaprak oluşturduğumuzda sistemin sınıflandırma başarısında büyük artış gözlenmektedir. Böylece elimizde yine 7 sınıf olmakta ve çok yakın renkteki *koyu gri* ve *siyah* araçlar aynı sınıfta toplanmaktadır. Bu şekilde oluşturulmuş bir ağaçta elde edilen sınıflandırma başarıları ANN, TM, SVM ve her düğümde en iyi başarıyı alarak hesaplanan renk tanıma başarı oranları sırasıyla %84.15, %83.16, %80.454 ve %84.39 olarak bulunmuştur. Ayrıntılı sonuçlar Çizelge 7.11’de gösterilmiştir. Çizelgedeki listelenen sonuçlar yüzde başarıları ifade etmektedirler. Piksel tabanlı (Median) öznitelik vektörü oluşturma yöntemi için oluşturulan tahmini ağacın her dallanmada elde edilen sınıflandırma oranları Şekil 7.2’de verilmiştir.

Çizelge 7.11: Orta değer ağacı için renk tanıma sonuçları

Kategori	TM	ANN	SVM	Karma
Kırmızı – Sarı	92.706	93.558	92.024	93.558
Mavi – Yeşil	84.983	87.318	85.122	87.318
Beyaz – Gri	73.147	73.518	67.167	74.072
Siyah - Koyu Gri	53.072	62.268	57.131	62.786
Koyu (*)	80.461	80.243	74.583	80.847
TOPLAM	75.977	79.166	75.361	79.434
TOPLAM (*)	83.162	84.148	80.458	84.392

Yukarıdaki çizelgede tahmini ağaç yönteminde “Median piksel” öznitelik seçim yöntemi ile elde edilen yüzde başarı oranları listelenmiştir. ART sonuçları TM, ANN, SVM ve Karma sınıflandırma yöntemleri kullanılarak hesaplanmıştır. “Karma” diye isimlendirilen yöntem, ağacın her dallanmasında elde edilen en iyi sınıflandırma başarılarının seçilmesinden başka bir şey değildir. Her düğümde en iyi sonucu veren sınıflandırma yönteminin seçilmesiyle tek sınıflandırıcı kullanmaktan daha iyi sonuçlar elde edilmiştir.

Örneğin, *beyaz* ve *gri* renklerinin sınıflandırılması ağaç üzerinde üç düğümden geçerek (kök düğümü, renksiz düğümü, açık düğümü) elde edilmektedir. Kök düğümündeki sınıflandırıcıların renkleri ayırma başarıları TM, ANN ve SVM sınıflandırıcıları için sırasıyla %93.92, %94.37 ve %93.229 olarak hesaplanmıştır. Karma sınıflandırma hesabı için ilk etapta %94.37 en yüksek başarısı ile ANN sınıflandırıcısı seçilmektedir. İkinci adımda, ilk adımdaki gibi TM, ANN ve SVM

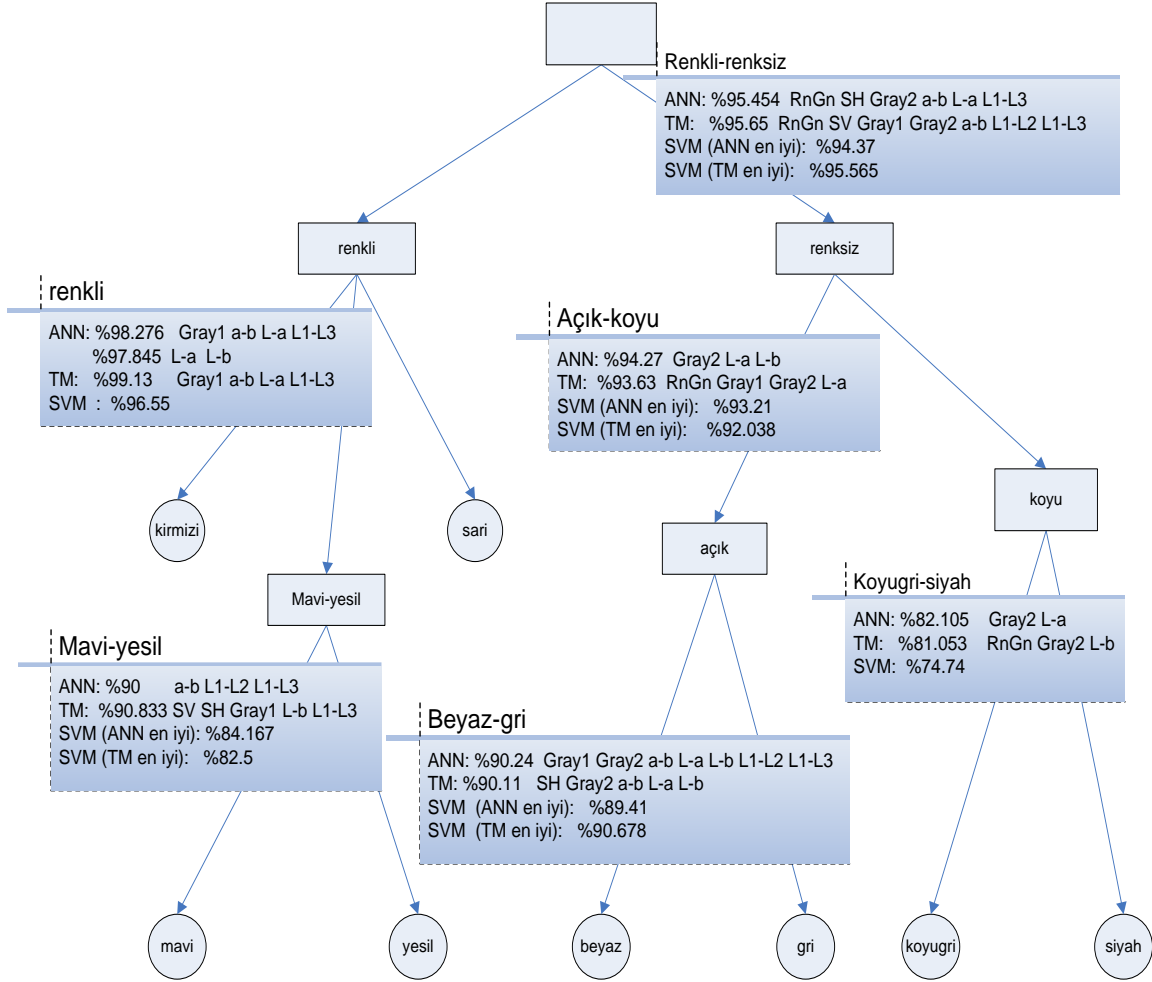
sınıflandırma yöntemleri arasında en iyi renk tanıma sonucu ve sınıflandırıcısı Karma yöntemi için kullanılmaktadır. Burada en yüksek sonuç %85,67 ile TM modeline aittir. Üçüncü ve son adımda da en yüksek başarı oranını %91.62 ile ANN sınıflandırıcısı vermiştir. *Beyaz* ve *gri* renklerinin Karma yöntemi ile bulunan %74.072 başarı sonucu yukarıda belirtilen her düğümdeki en yüksek başarılarının, diğer sınıflandırma yöntemlerinde olduğu gibi, çarpımlarıyla elde edilmektedir. Diğer renklere ve toplam başarıya bakıldığında Karma yönteminin performansının daha yüksek olduğu açıkça görülmektedir.

Çizelgede (*) ile işaretlenmiş “Koyu” kategorisi, yukarıda da belirtildiği gibi *koyu* düğümünün *siyah* ve *koyu gri* renklerinin bir daha sınıflandırıcıdan geçirilmediği ve bir kategoride ele alındığı, ağacın da 8 değil 7 sınıfa sahip olduğu durumu göstermektedir. TOPLAM(*) da aynı şekilde *kırmızı*, *sarı*, *mavi*, *yeşil*, *beyaz*, *gri* ve *koyu* renk sınıflarının olduğu tahmini ağaç için bulunan ağacın toplam başarılarını göstermektedir.

7.3.2 Histogram Ağacı test sonuçları

Oluşturduğumuz tahmini ağaçta histogram tabanlı öznitelik vektörlerinin sağlayacakları performansları görmek için de testler yapılmıştır. Orta değer ağaç yönteminde olduğu gibi ağacın her düğümü için, oluşabilecek tüm histogram öznitelik vektörleri (2^{10}) için TM, ANN ve SVM başarıları hesaplanmıştır. Şekil 7.3'te tahmini ağacın en iyi toplam başarısını elde etmek için hesaplanan, her düğümdeki en yüksek performanslar ve bu sonuçları sağlayan sınıflandırıcı yöntemleri ile öznitelik vektörleri gösterilmektedir.

Histogram Ağacı



Şekil 7.3 : Histogram öznetelik vektörleri ve her düğümdeki sınıflandırma başarıları

Histogram öznetelik vektörleri kullanılarak yapılan araç renk tanıma testlerinde sistemin renkli düğümünde kırmızı, sarı ve mavi-yeşil renklerini an az hata ile sınıflandırdığını görmek mümkündür (ANN: %93.76, TM: %99.13). Ayrıca, araç resimlerini “renksiz” düğümünde açık ve koyu olarak sınıflandırırken de oldukça başarılı olduğu gözlenmiştir (ANN: %94.27, TM: %93.63). Diğer taraftan, “koyu” düğümündeki örneklerin koyu gri ve siyah renklerine sınıflandırılması en başarısız performans oranlarını göstermektedir (ANN: %82.105, TM: %81.053). Bu nedenle sistemin başarısını artırmak için en başarısız sınıflandırma sonucunu veren ve birbirine çok yakın renkleri barındıran “koyu” düğümünü yaprağa dönüştürerek

yeni bir tahmini ağaç da elde edilmiştir. Çizelge 7.12’de hem yukarıda bahsedilen iki farklı ağaç için sistemin toplam sınıflandırma başarıları hem de her bir araç renginin doğru sınıflandırma oranları gösterilmektedir.

Çizelge 7.12: Histogram ağacı için renk tanıma başarı sonuçları

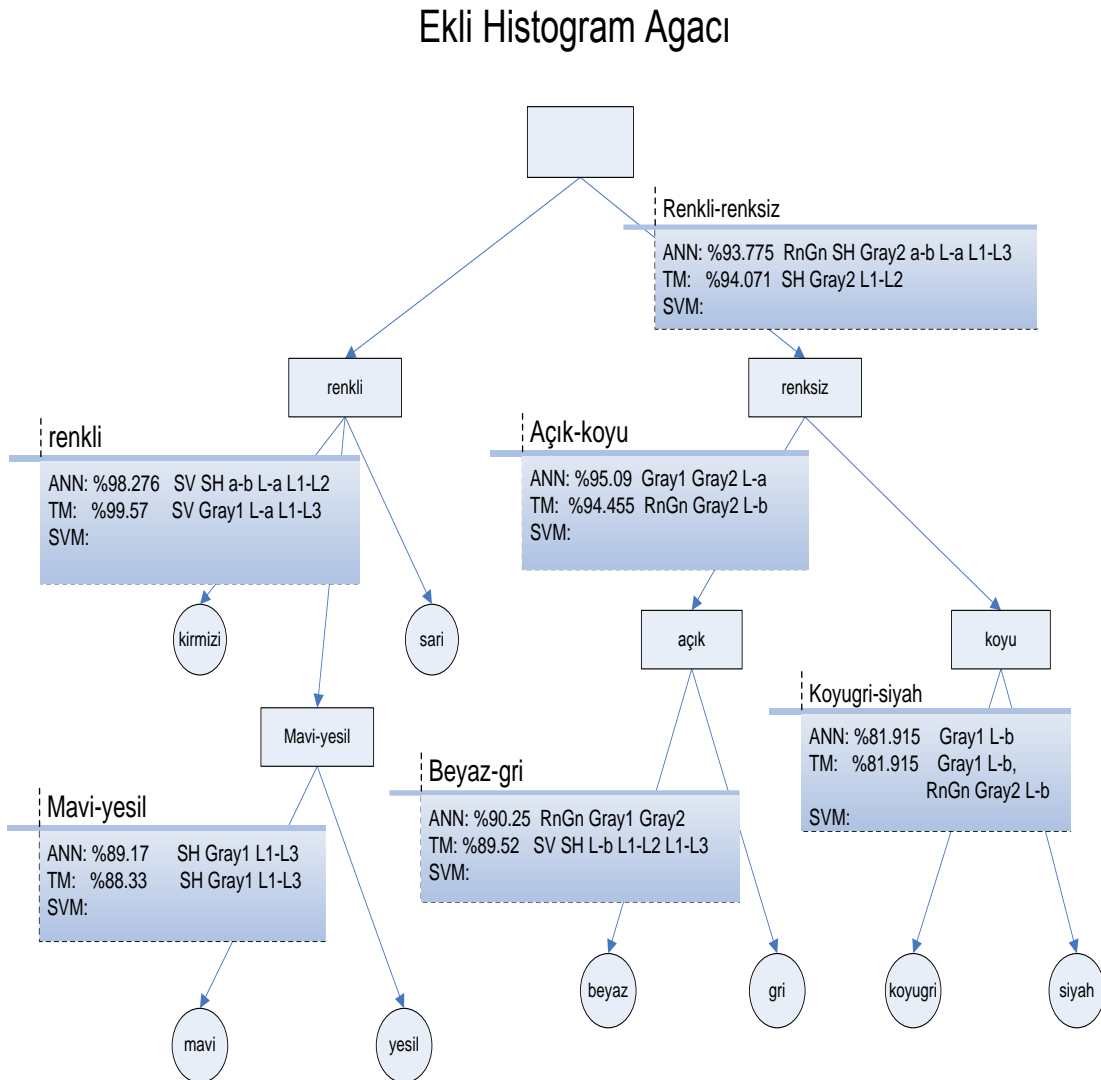
Ağaç	TM	ANN	SVM	Karma
Kırmızı – Sarı	%94.818	%93.808	%91.303	%94.818
Mavi – Yeşil	%86.126	%84.428	%76.847	%86.126
Beyaz – Gri	%80.70	%81.202	%81.691	%81.369
Siyah - Koyu Gri	%72.589	%73.882	%67.333	%74.034
Koyu (*)	%89.557	%89.984	%89.076	%90.169
TOPLAM	%83.56	%83.33	%79.293	%84.086
TOPLAM(*)	%87.55	%86.98	%84.148	%87.828

Çizelgede (*) işaretli satırlar *koyu* düğümünün bir kategori olarak alındığı zamanki *koyu* renginin (yaprak) tüm sınıflandırıcılara göre sınıflandırma oranlarını ve sistemin toplam başarısını ifade etmektedir. Yapılan testler sonucunda elde edilen çizelgedeki veriler kimi renklerin ANN modeli ile kimi renklerin de TM modeli ile daha iyi sınıflandırıldıklarını göstermektedirler. Bu nedenle bazı düğümlerde TM bazı düğümlerde de ANN modeli ile sınıflandırma yaparak toplamda %87.828 sınıflandırma başarısını elde etmek mümkündür. Sadece TM sınıflandırıcısı kullanıldığında sistemin başarısı %87.55 olarak bulunmaktadır. Çizelgede de görüldüğü gibi *siyah* ve *koyu gri* renklerini ayırmak yerin onları bir üst seviyedeki kategorilerinde bırakmak TM’de %4, ANN’de %3.65, SVM’de %5, Karma modelinde de %3,74’lük bir performans artışı sağlamıştır.

Histogram tahmini ağacı ile orta değer tahmini ağaçlarının performanslarını karşılaştırdığımızda, “Yarım Araç ROI” veri tabanı kullanılarak elde edilen histogram öznitelik vektörleri ile oluşturulan histogram ağacının neredeyse her renk için ve genel başarıda da daha yüksek sonuçlar verdiğini görmekteyiz. Orta değer tahmini ağacında sadece *mavi* ve *yeşil* renklerin sınıflandırma başarı oranları (%87.318) histogram tahmini ağacında elde edilen bu renklerin sınıflandırma başarılarından daha yüksektir.

7.3.3 Ekli Histogram Ağacı test sonuçları

Ekli histogram öznitelik vektörleri ile tahmini ağaç üzerindeki son testler yapılmıştır. Ekli histogram tahmini ağacında yapılan testler histogram tahmini ağacındaki ile benzerlik göstermektedir. Bu testlerde de, ağacın her düğümünde 2^{10} adet öznitelik vektörünün performansları TM ve ANN sınıflandırıcıları için hesaplanmış ve en yüksek değerleri veren öznitelikler seçilmiştir. Diğer testlerden edinilen verilere göre SVM modelinin TM ve ANN karşısında daha başarısız olduğu görüldüğü için bu testlerde yer almamıştır. Şekil 7.4'teki ağaçta ekli histogram öznitelik vektörlerinin tahmini ağacın her düğümünde elde edilen doğru sınıflandırma başarıları gösterilmektedir.



Şekil 7.4 : Ekli Histogram öznitelik vektörleri ve her düğümdeki sınıflandırma başarıları

Ekli histogram Tahmini ağacında da en yüksek sınıflandırma başarısının alındığı düğüm “renkli” düğümü (ANN: %98.276, TM: %99.57), en düşük başarı sonucunun elde edildiği düğüm de *siyah* ve *koyu gri* renklerini ayıran “*koyu*” düğümüdür (ANN ve TM: %81.915). *Koyu* düğümünün sınıflandırma başarısı ile kastedilen sadece *koyu* kategorisindeki örneklerin *siyah* ve *koyu gri* renklerine doğru ayrılma oranıdır, “*kök*”ten başlayarak “*renksiz*” düğümünden geçerek “*koyu*” düğümüne gelene kadar yapılan sınıflandırmalar hesaba katılmamaktadır. Bunlar hesaba katılarak (kökten başlanarak yol üzerindeki her düğümün sınıflandırma oranları çarpılarak) elde edilen *siyah* ve *koyu gri* renk tanıma başarılarının ANN için %73.044, TM için ise %72.789 olarak bulunmuştur. *Siyah* ve *koyu gri* renkleri birbirine yakın renklendir. Ayrıca görüldüğü üzere birbirlerinden ayrılma oranları diğer renklere göre düşük olduğu için bir de bu renkler aynı sınıfta (*koyu*) olacak şekilde tahmini ağacın toplam başarısı hesaplanmıştır. Çizelge 7.13’te bu sonuç, TOPLAM(*) satırında her sınıflandırma modeli için verilmiştir. *Koyu*(*) satırında da *koyu* renginin sınıflandırıcılara göre sınıflandırma başarıları görülmektedir.

Çizelge 7.13: Ekli histogram ağacı için renk tanıma başarı sonuçları

Ağaç	TM	ANN	Karma
Kırmızı – Sarı	%93.667	%92.158	%93.667
Mavi – Yeşil	%82.736	%82.178	%83.522
Beyaz – Gri	%79.543	%80.477	%80.731
Siyah – Koyu Gri	%72.785	%73.044	%73.275
Koyu (*)	%88.855	%89.171	%89.452
TOPLAM	%82.183	%81.964	%82.799
TOPLAM(*)	%85.820	%85.517	%86.470

Çizelgedeki oranlar her renk sınıfının TM, ANN ve bu sınıflandırıcılardan her düğümdeki daha yükseklerinin alınarak hesaplanan Karma Modeli için renk tanıma başarıları listelenmiştir. Toplam renk başarılarına baktığımızda oranların küçükten büyüğe doğru ANN (%81.964), TM (%82.183) ve Karma (%82.799) sınıflandırıcı modellerinin verdikleri sonuçlar olarak sıralanmaktadırlar. Bu sonuçlar Şekil 7.4’te gösterilen sekiz yaprağı olan tahmini ağaç için hesaplanan toplam renk tanıma başarı

oranlarıdır. Yedi renk çıktısı (*kırmızı, sarı, mavi, yeşil, beyaz, gri ve koyu*) olan tahmini ağaç için elde edilen toplam renk tanıma sonuçları ise sırasıyla ANN: %85.517, TM: 85.820 ve Karma: %86.470 oldukları görülmektedir.

Tahmini ağaç üzerinde yapılan testler sonucunda, Orta değer Ağacının Histogram ve Ekli Histogram Ağaçlarına göre daha düşük performans gösterdiği gözlemlenmiştir. Testlere başlarken Ekli Histogram Ağacı testlerinin Histogram Ağacı testlerinden daha başarılı olabileceklerini düşünmemize rağmen testler bunun tam tersi sonuçlar vermişlerdir. Sınıflandırıcılara bakıldığında da Orta değer ağacında ANN sınıflandırma modelinin, Histogram ağaçlarında da “renkli” renklerini sınıflandırmada TM, “renksiz” renklerinde de ANN modellerinin daha iyi oldukları gösterilmiştir. Diğer yandan, SVM modeli TM ve ANN sınıflandırma yöntemlerine göre daha düşük sınıflandırma oranları vermiştir.

Sekiz yapraklı (8 renk sınıfı) tahmini kara ağacı modelinde en yüksek renk tanıma başarısı olan %84.86 oranı Histogram ağacında Karma modeli ile hesaplanmıştır. Bundan daha yüksek bir sınıflandırma oranı yine Histogram ağacında yedi yapraklı (7 renk sınıfı) tahmini ağaç için hesaplanan %87.828’lik Karma modeli başarı oranıdır.

7.4 Hesaplamalı Karar Ağacı Öğrenmesi ve Bazı Testler

Karar ağacını oluşturmak için belirli karar koşullarından yola çıkarak hangi koşulu ağaç üzerinde nerede kullanmak gerektiğini bulmak için ID3 algoritmasından yararlanılmaktadır. Bu yöntem ile en uygun “Hesaplamalı Karar Ağacı” elde edilebilmektedir. Hesaplamalı karar ağacını oluşturabilmek için araçların renklerini etkileyebilecek i) hava koşulları, ii) kamera açısı ve iii) farklı öznitelik vektörleri gibi koşullardan yararlanılmalıdır [3].

Elimizde çok fazla öznitelik vektörü olduğu için öznitelik vektörlerinde sadeleştirmeye gidilmesi gerekti. Karar ağacı oluşturulurken her düğümde herhangi bir renk sınıfındaki örneklerin tümünün doğru sınıfa atanmış olmasına bakılmaktadır. Bu nedenle hesaplamalı karar ağacında kullanılacak öznitelik vektörlerini seçerken bir rengi diğerlerinden en iyi ayıran öznitelik vektörlerinin seçilmesine dikkat edilmiştir. Örneğin, TM sınıflandırma yöntemi kullanılarak kırmızı rengini diğer

renklerden en iyi “*SV Gray1 Gray2 L-a L1-L2*” öznitelik, beyaz rengini de “*RnGn SV SH L-b L1-L2 L1-L3*” öznitelik vektörü diğer renklerden ayırmaktadır.

Histogram tabanlı öznitelik vektörleri hem normal sınıflandırmada hem de Tahmini Karar Ağacı modelinde daha yüksek renk sonuçlar verdiği için, Hesaplamalı Ağaç oluşturmak için yapılan testlerde histogram tabanlı öznitelik vektörleri kullanılmıştır. Bu aşamada yapılan testler için araç veri tabanı renk ve diğerleri olarak 7 farklı şekilde eğitim ve test kümeleri olarak düzenlenmiştir. Bir rengi geri kalan renklerden en iyi ayıran öznitelik vektörünü bulmak için oluşturulan yeni veri kümesinde iki sınıf (renk ve diğerleri) bulunmaktadır. Örneğin, kırmızı renk için oluşturulan veri kümesinde, *kırmızı* renkli araçlar *kırmızı* sınıfında, kırmızıdan başka renge sahip araçlar ise *diğer* sınıfında gruplanmışlardır. Aşağıdaki çizelgelerde *kırmızı* ve *diğer* olarak yapılan sınıflandırma için başarı sonuçları gösterilmiştir. Ayrıca, “*SH Gray2 L1-L2*” öznitelik vektörü ile elde edilen eğitim ve test karıştırma matrisi yer almaktadır (sırasıyla Çizelge 7.14 ve Çizelge 7.15).

Çizelge 7.14: Kırmızı- Diğer eğitim kümesi karıştırma matrisi

ANN Eğitim	Kırmızı	Diğer		Başarı
Kırmızı	336	0		%100
Diğer	0	600		%100
Toplam				%100

Çizelge 7.15: Kırmızı - Diğer Test kümesi karıştırma matrisi

ANN Test	Kırmızı	Diğer		Başarı
Kırmızı	96	4		%96
Diğer	0	600		%100
Toplam				%99.429

Yedi renk, 2^{10} adet öznitelik vektörü ve TM ile ANN sınıflandırıcıları kullanılarak yapılan toplam 7×2^{10} sınıflandırma sonucunda her bir rengi en iyi sınıflandıran öznitelik vektörleri bulunmuştur. Çizelge 7.16’da gösterilmiştir.

Çizelge 7.16: Bir rengi diğerlerinden en iyi ayıran öznitelik vektörleri

Renk	Yöntem	Başarı	Öznitelik
Beyaz	TM	%96.059	RnGn SV SH L-b L1-L2 L1-L3
	ANN	%95.833	RnGn L-b L1-L3 RnGn L-b L-a L1-L3
Gri	TM	%90.341	RnGn SV Gray1 L-b RnGn SV Gray2 L-b
	ANN	%89.773	SV Gray2 a-b L-a
Kırmızı	TM	%99.841	SV Gray1 Gray2 L-a L1-L2
	ANN	%99.365	SH Gray2 L1-L2 RnGn SH Gray2 L1-L2 L-a L1-L2 SV L-a L1-L2 SV Gray1 L-a L1-L2 SV Gray2 L-a L1-L2 L-a L1-L3 SV L-a L1-L3 L-a L1-L2 L1-L3 SV L-a L1-L2 L1-L3
Mavi	TM	%82.54	RnGn SH a-b L-b L1-L2 SH a-b L-a L-b L1-L2 RnGn SH a-b L-a L-b L1-L2
	ANN	%84.127	SH RnGn SV RnGn a-b RnGn Gray1 RnGn Gray1 Gray2

			RnGn Gray2 L-b Gray1 L-b Gray2 L-b RnGn L-b a-b L1-L3 L1-L2 RnGn SV Gray2 RnGn L-a SV SV Gray1
Sarı	TM	%97.623	RnGn SH L1-L2
	ANN	%97.623	RnGn SH L1-L3
Siyah	TM	%90.348	RnGn SV Gray2 L1-L2 RnGn SV SH Gray2 L1-L2 L1-L3 RnGn SV SH Gray1 Gray2 L-b L1-L2
	ANN	%91.772	RnGn SV L1-L2
Yeşil	TM	%94.286	RnGn SH SV L1-L3 RnGn SH Gray1 L1-L3
	ANN	%95.571	SH L1-L3

Yukarıda belirtilen öznitelik vektörlerinden başka bir de Çizelge 7.7 ve Çizelge 7.9'daki sonuçlara göre histogram öznitelik vektörlerinden en yüksek başarı oranını sağlayan 5'er tane öznitelik vektörü seçilmiştir. Seçilen öznitelik vektörleri kullanılarak eğitim kümesi sınıflandırılarak Entropi (E) ve Bilgi Kazancı (IG) değerleri hesaplanmıştır. Çizelge 7.17'de da görüldüğü üzere yedi kategoriden sadece “kırmızı” kategorisine ait olarak bulunan örnekler homojendir. Çizelgede “SV

SH Gray2 L-a L-b L1-L3” öznitelik vektörünün kullanıldığında elde edilen doğru ve yanlış bulunan araç resimlerinin sayıları gösteren karıştırma matrisi verilmiştir.

Çizelge 7.17: Eğitim kümesi ANN sınıflandırıcısı için karıştırma matrisi

ANN TRAIN	Beyaz	Gri	Kırmızı	Mavi	Sarı	Siyah	Yeşil	Başarı
Beyaz	325	4	0	0	0	1	0	%98.48
Gri	2	318	0	4	0	4	2	%96.36
Kırmızı	0	0	336	0	0	0	0	%100
Mavi	1	2	0	228	0	0	0	%98.7
Sarı	0	0	0	0	106	0	0	%100
Siyah	0	3	0	4	1	211	0	%96.35
Yeşil	0	0	0	2	0	0	99	%98.02
Toplam								%98.15

TM sınıflandırıcısı ile eğitim kümesi sınıflandırıldığında her öznitelik vektörü için %100 bir sınıflandırma başarısı elde edildiği için (Çizelge 7.18), en uygun ağacı oluştururken TM sınıflandırıcısı kullanılmayacaktır.

Çizelge 7.18: Eğitim kümesi TM sınıflandırıcısı için karıştırma matrisi

TM TRAIN	Beyaz	Gri	Kırmızı	Mavi	Sarı	Siyah	Yeşil	Başarı
Beyaz	330	0	0	0	0	0	0	%100
Gri	0	330	0	0	0	0	0	%100
Kırmızı	0	0	336	0	0	0	0	%100
Mavi	0	0	0	231	0	0	0	%100
Sarı	0	0	0	0	106	0	0	%100
Siyah	0	0	0	0	0	219	0	%100
Yeşil	0	0	0	0	0	0	101	%100
Toplam								%100

Öznitelikler belirlendikten sonra hava durumu koşulları için veri kümesinin bulutlu ve açık havalar için tekrar düzenlenmesi gerekiyordu. Sonuç olarak ID3 algoritması ile karar ağacı eğitilememiştir. Bu kısım ileriye dönük bir çalışmaya rehberlik etmesi amacıyla yazılmıştır.

8. SONUÇLAR VE İLERİKİ ÇALIŞMALAR

Bu çalışmanın sonunda ulaşılmak istenen, özellik vektörü oluşturma, ROI seçme ve sınıflandırma metotlarının bir potada eritilip araç renk tanıma (ART) problemine çözüm olacak en başarılı yöntemi bulmaya çalışmaktır. Araç renk tanıma sistemi giriş olarak renkli araç görüntülerini alarak araçları siyah, gri, beyaz, kırmızı, yeşil, mavi ve sarı olmak üzere yedi renkten birinden bir sonuç tahmini yapar. Aynı zamanda, ART sisteminin renk tanıma başarısını artırmak için karşılaşılan yansıma, gölge, aracın görüntüde ne kadar yer tuttuğu gibi problemlere de çözüm getirilmeye çalışılmıştır. Bunun için en uygun kare seçimi, ROI seçimi, farklı renk uzayları seçimi ve öznitelik vektörü oluşturma yöntemleri geliştirilip denenmiştir. En yüksek sınıflandırma başarısını veren renkler kırmızı ve sarı, en düşük olanlar ise gri ve yeşildir. Yeşil araç sayısı az olması nedeniyle, asıl süregelen problem gri araçların beyaz araçlarla karışması olmuştur.

Yapılan çalışmalar gösteriyor ki, histogram öznitelik vektörü oluşturma yöntemi ve ANN sınıflandırıcısını kullanarak en yüksek başarılı sonuç elde edilebilir. Ayrıca, bulunan öznitelik vektörünün daha önce yapılmış araştırmalar ile karşılaştırıldığında, onlardan daha iyi sonuç verdiği de görülmüştür.

İleriki çalışma olarak tam olarak gerçekleyemediğimiz ID3 algoritması ile bir karar ağacı oluşturarak renkleri sınıflandırmayı verebiliriz. ID3 algoritmasını araç renk tanıma üzerinde uygulamak için, veri kümelerini hava koşullarına, araç tiplerine göre vs. ayırarak dallanma kriterleri oluşturulabilir. Bizim veri kümemiz baştan başka şekilde oluşturulduğu için ID3 algoritması gerçekleşmesi ertelenmiştir.

KAYNAKLAR

- [1] **D. Berwick, and S.W. Lee.** A chromaticity space for specularity, illumination color- and illumination pose- invariant 3D object recognition. *Proceedings International Conference on Computer Vision*, 1998, p. 165-170. Retrieved December 18, 2009, from <http://citeseer.ist.psu.edu/berwick98chromaticity.html>
- [2] **N. Sebe, M.S. Lew.** Color-based retrieval. *Pattern Recognition Letters*, 22:223-230, Elsevier Science B.V., 2001.
- [3] **S.D. Buluswar and B.A. Darper.** Color Recognition in Outdoor Images. In *IEEE International Conference on Computer Vision*, January 1998.
- [4] **Y. Tsin, R.T. Collins, T. Kanade, and V. Ramesh.** Bayesian color constancy for outdoor object recognition. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR-01)-Volume 1*, p. 1132, 2001.
- [5] **B.D. Zarit, B.J. Super, F.K.H. Quek.** Comparison of five color models in skin pixel classification. In: *Proc. Int. Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, p. 58-63, 1999.
- [6] **D.Chai and A. Bouzerdoum.** A Bayesian approach to skin color classification using YCbCr color space. In: *Proc. TENCON 2000 - Volume 2*, p. 421-423, September 2000.
- [7] **M. Merler.** Car Color and Logo Recognition, 2006. Retrieved June 15, 2008, from <http://www-cse.ucsd.edu>
- [8] **N. Beak, S.-M. Park, K.-J. Kim, and S.-B. Park.** Vehicle color classification based on the support vector machine method. *D.-S. Huang, L.Heutte, and M. Loog(Eds): ICIC 2007, CCIS 2*, p. 1133 – 1139, 2007. Springer-Verlag Berlin Heidelberg, 2007.
- [9] **F.Kahraman, M.Gökmen.** Bakış Açısından Bağımsız Gürbüz Plaka Tanıma Sistemi, IEEE 13. Sinyal İşleme ve İletişim Uygulamaları Kurultayı, 16-18 Mayıs 2005, Kayseri
- [10] **A.W. Moore.** Learning with Maximum Likelihood. September 2001. Retrieved May 02, 2007, from <http://www.autonlab.org/tutorials/mle13.pdf>
- [11] **V.S. Petrovic, T. Cootes.** Analysis of Features for Rigid Structure Vehicle Type Recognition. In: *British Machine Vision Conference (BMVC2004)*, Kingston, September 2004.

- [12] **M. Swain and D. Ballard.** Color indexing. *International Journal of Computer Vision*, 7(11), November 1991.
- [13] **R.E. Woods, R.C. Gonzalez, S.L. Eddins,** Color Image Processing (Ch 6), *Digital Image Processing Using Matlab*, Pearson Prentice Hall, Upper Saddle River / New Jersey, 2004.
- [14] **Y. Rubner, J. Puzicha, C. Tomasi, and J.M. Buhmann.** Empirical evaluation of dissimilarity measures for color and texture. In: *Seventh International Conference on Computer Vision (ICCV'99)* - Volume 2, p. 1165, 1999.
- [15] **M.S. Drew, J. Wie, Z.N. Li.** Illumination-Invariant Color Object Recognition via Compressed Chromaticity Histograms of Color-Channel-Normalized Images. In: *International Conference on Computer Vision (ICCV)*, p. 533, IEEE Computer Society. 1998.
- [16] **Y. Otha,** Knowledge-Based Interpretation of Outdoor Natural Color Scenes. Morgan Kaufman Publishers, Inc. Palo Alto, CA., 1985.
- [17] **O. Hasegawa and T. Kanade.** Type classification, color estimation, and specific target detection of moving targets on public streets. In *Machine Vision and Application*, 16: 116-121, 2005. Springer-Verlag, December 2004
- [18] **M. Alencastre-Miranda, L. Munoz-Gomez, R. Swain-Oropeza Super, C. Nieto-Granda.** Color-Image Classification using MRFs for an Outdoor Mobile Robot. In *Journal of Systemics, Cybernetics and Informatics*-Volume 3, p. 52-59, 2006.
- [19] **J.F. Barbe, A.H. Tewfik, A.B. Khodursky.** Transcription Factor Discovery Using Support Vector Machines and Heterogeneous Data. In: *GENSIPS 2007, IEEE International Workshop On*. Ekim 2007.
- [20] **G.Guo, H.Wang and D.Bell.** KNN Model-Based Approach in Classification. *On The Move to Meaningful Internet Systems*, pp. 986-996, Springer Berlin/Heidelberg, October 2003.
- [21] **S. Kahraman, O. Gunaydin, M. Alber and M. Fener,** Evaluating the strength and deformability properties of misis fault breccia using artificial neural networks. *Expert Systems with Applications*, 36: 6874–6878, 2009.
- [22] **E. Koskivaara.** Neural networks in analytical review procedures. *Managerial Auditing Journal*, 19(2), 191–223, 2004.
- [23] **M. Firat.** Image Recognition with Neural Networks. October 2007. Retrieved March 12, 2010, from <<http://www.codeproject.com/KB/cs/BackPropagationNeuralNet.aspx>>
- [24] **P. Gupta and N. K. Sinha.** An improved approach for nonlinear system identification using neural networks. *Journal of the Franklin Institute*, 336(4):721-734, 1999.

- [25] **C. Cortes, V.Vapnik**, Support Vector Networks, *Machine Learning*, 20:273-297, 1995.
- [26] **J. Johnson, and P. Picton**. *Mechatronics: Designing Intelligent Machines, Concepts in Artificial Intelligence*. Vol 2, Butterworth-Heinemann, 1995.
- [27] **A.W. Moore**. *Support vector machines* (Tutorial-slides). November 2001. Retrieved December 14, 2007, from <http://www.autonlab.org/tutorials/svm15.pdf>
- [28] **J. C. Platt**. Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines. In: *Advances in Kernel Methods - Support Vector Learning*, MIT Press, January 1998.
- [29] **Url-1** The Standars SVM Formulation. Retrieved April 04, 2008, from <<http://research.microsoft.com/en-us/um/people/manik/projects/trade-off/svm.html>>
- [30] **T. Joachims**. 11 in: Making large-Scale SVM Learning Practical. *Advances in Kernel Methods - Support Vector Learning*, B. Schölkopf and C. Burges and A. Smola (ed.), MIT Press, 1999.
- [31] **The McGraw-Hill Companies**, 1997. *Machine Learning*, Inc, International Edition.
- [32] **A. Colin**. *Building Decision Trees with the ID3 Algorithm*. Dr. Dobbs Journal, June 1996.
- [33] **J. R. Quinlan**. *C4.5: Programs for Machine Learning*. Morgan-Kaufmann Publishers, San Mateo, CA, 1993.

ÖZGEÇMİŞ



- Ad Soyad:** Erida DULE
- Doğum Yeri ve Tarihi:** Korçe / Arnavutluk – 26.06.1984
- Adres:** Silivri / İstanbul
- Lisans Üniversitesi:** İstanbul Teknik Üniversitesi
- Yayın Listesi:** **E. Dule**, M. Gökmen, S.Beratoğlu. A Convenient Feature Vector Construction for Vehicle Color Recognition, In: *WSEAS*. Iasi Romania, June 2010.