



**T.C.**  
**KONYA TEKNİK ÜNİVERSİTESİ**  
**LİSANSÜSTÜ EĞİTİM ENSTİTÜSÜ**



**GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE DİŞ**  
**RENK SKALASININ BELİRLENMESİ**  
**Furkan PEHLİVAN**

**YÜKSEK LİSANS TEZİ**

**ELEKTRİK-ELEKTRONİK MÜHENDİSLİĞİ**  
**Anabilim Dalı**

**Haziran-2022**  
**KONYA**  
**Her Hakkı Saklıdır**

## TEZ KABUL VE ONAYI

Furkan PEHLİVAN tarafından hazırlanan “GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE DİŞ RENK SKALASININ BELİRLENMESİ” adlı tez çalışması 21/06/2022 tarihinde aşağıdaki jüri tarafından oy birliği / oy çokluğu ile Konya Teknik Üniversitesi Lisansüstü Eğitim Enstitüsü Elektrik – Elektronik Mühendisliği Anabilim Dalı’nda YÜKSEK LİSANS/DOKTORA TEZİ olarak kabul edilmiştir.

### Jüri Üyeleri

### İmza

#### Başkan

DOÇ. DR. ŞABAN ÖZTÜRK

.....

#### Danışman

Dr. Öğr. Üyesi MÜCAHİD BARSTUĞAN

.....

#### Üye

Dr. Öğr. Üyesi UMUT ÖZKAYA

.....

Yukarıdaki sonucu onaylarım.

Prof. Dr. Saadettin Erhan KESEN  
Enstitü Müdürü

## TEZ BİLDİRİMİ

Bu tezdeki bütün bilgilerin etik davranış ve akademik kurallar çerçevesinde elde edildiğini ve tez yazım kurallarına uygun olarak hazırlanan bu çalışmada bana ait olmayan her türlü ifade ve bilginin kaynağına eksiksiz atıf yapıldığını bildiririm.

## DECLARATION PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

İmza  
Furkan PEHLİVAN

Tarih:

## ÖZET

### YÜKSEK LİSANS

## GÖRÜNTÜ İŞLEME TEKNİKLERİ İLE DIŞ RENK SKALASININ BELİRLENMESİ

**Furkan PEHLİVAN**

**Konya Teknik Üniversitesi  
Lisansüstü Eğitim Enstitüsü  
Elektrik-Elektronik Mühendisliği Anabilim Dalı**

**Danışman: Dr. Öğr. Üyesi MÜCAHİD BARSTUĞAN**

**2022, 83 Sayfa**

**Jüri**

**DOÇ. DR. ŞABAN ÖZTÜRK  
Dr. Öğr. Üyesi MÜCAHİD BARSTUĞAN  
Dr. Öğr. Üyesi UMUT ÖZKAYA**

Diş hekimliğinde önemli bir yere sahip olan diş renginin belirlenmesi ve komşu dişlere renk bakımından uygunluğu hatasız veya yüksek doğruluk oranına sahip olması gerekmektedir. Çünkü dişler insanların en çok dikkatini çeken alanlardan birisidir. Renk belirlenirken kullanılan yöntemler, çevresel etkiler, göz yorgunluğu ve birçok etkenden dolayı göz ile yapılan renk belirleme işlemi yapay diş ile doğal diş arasında renk farklılığına neden olmakta ve komşu dişler arasında uyumsuzluk yaratmaktadır. Bu tür durumlar için birçok elektronik alet geliştirilmiş ve hataların azaltılması amaçlanmıştır. Bu elektronik aletlerin maliyeti, boyutları ve kullanım alanları ve doğruluğuna göre birçok modeli vardır. Bu tezde sabit ve kaliteli ışık kaynağı kullanarak kamera vasıtasıyla elde edilmiş görüntülerin bilgisayar ortamında bazı görüntü işleme algoritmalarından geçirilerek VITA Toothguide 3D-MASTER diş renk skalasına göre yüksek doğrulukta sonuç üretmesi sağlanmaya çalışılmıştır. Bu maksatla birden fazla fotoğraf elde edilmiş ve elde edilen görüntülerde oluşan çevresel faktörler en aza indirilmiş ve ağız içerisinde bulunan dişlerin renk haritalaması yapılmıştır. Skalada alınan veriler ile elde edilen görüntülerin renk ortalamaları karşılaştırılarak sonuç üretilmiştir.

**Anahtar Kelimeler:** Görüntü İşleme, Diş Haritalama, Renk Tespiti, Derin Öğrenme, K-Means algoritması

**ABSTRACT**

**MS THESIS**

**DETERMINATION OF TOOTH COLOR SCHEDULE BY IMAGE  
PROCESSING TECHNIQUES**

**Furkan PEHLİVAN**

**Konya Technical University  
Institute of Graduate Studies  
Department of ELECTRICAL ELECTRONICS ENGINEERING**

**Advisor: Asst. Prof. Dr. MÜCAHİD BARSTUĞAN**

**2022, 80 Pages**

**Jury**

**Assoc. Dr. ŞABAN ÖZTÜRK  
Asst. Prof. Dr. MÜCAHİD BARSTUĞAN  
Asst. Prof. Dr. UMUT ÖZKAYA**

It is necessary to determine the color of the teeth, which has an important place in dentistry, and its compatibility with neighboring teeth in terms of color should be error-free or have a high accuracy rate. Because teeth are one of the areas that attract the most attention of people. Due to the methods used in determining the color, environmental effects, eye fatigue and many factors, the color determination process with the eye causes a color difference between the artificial tooth and the natural tooth and creates incompatibility between the neighboring teeth. Many electronic devices have been developed for such situations and it is aimed to reduce errors. There are many models of these electronic devices according to their cost, size and usage areas and accuracy. In this thesis, it has been tried to produce high accuracy results according to the VITA Toothguide 3D-MASTER tooth color scale by passing the images obtained by the camera using a fixed and high quality light source through some image processing algorithms in the computer environment. For this purpose, more than one photograph was obtained and environmental factors in the obtained images were minimized and color mapping of the teeth in the mouth was made. The result was produced by comparing the color averages of the images obtained with the data taken in the scale.

**Keywords:** Image Processing, Tooth Mapping, Color Detection, Deep Learning, K-Means algorithm

## ÖNSÖZ

Çalışmalarımda yardımını esirgemeyen danışmanım Konya Teknik Üniversitesi Elektrik-Elektronik Mühendisliği öğretim üyesi Dr. Öğr. Üyesi Mücahid BARSTUĞAN, ikinci danışmanım Selçuk Üniversitesi Diş Hekimliği doktorlarından Prof. Dr. Serhan AKMAN' a ve bölümümüz tüm değerli öğretim elemanlarına teşekkürü borç bilirim.

Furkan PEHLİVAN  
KONYA-2022



# İÇİNDEKİLER

<b>ÖZET</b> .....	<b>iv</b>
<b>ABSTRACT</b> .....	<b>v</b>
<b>ÖNSÖZ</b> .....	<b>vi</b>
<b>İÇİNDEKİLER</b> .....	<b>vii</b>
<b>SİMGELER VE KISALTMALAR</b> .....	<b>ix</b>
<b>1. GİRİŞ</b> .....	<b>1</b>
<b>2. KAYNAK ARAŞTIRMASI</b> .....	<b>3</b>
<b>3. DIŞ RENGİ TESPİTİ</b> .....	<b>7</b>
3.1. Metamerizm .....	9
3.2. Diş Rengi Seçiminde Kullanılması Gereken Prensipler .....	10
3.3. Diş Skalaları .....	11
3.4. Diş Rengi Belirlemede Kullanılan Elektronik Aletler .....	14
3.4.1. RGB Cihazlar .....	14
3.4.2. Spektrofotometre .....	15
3.4.3. Spektroradyometre .....	17
3.4.4. Kolorimetre .....	18
<b>4. MATERYAL VE YÖNTEM</b> .....	<b>19</b>
4.1. Işık ve Elektromanyetik Spektrum İlişkisi.....	19
4.2. Renk Modelleri .....	23
4.2.1. Munsell Renk Sistemi .....	23
4.2.2. CIELab Renk Sistemi.....	24
4.2.3. RGB Renk Uzayı .....	24
4.2.3. HSV Renk Uzayı.....	25
4.2.4. RGB Görüntüyü HSV görüntüye dönüştürme .....	26
4.3. HSV Görüntüyü RGB görüntüye dönüştürme .....	27

4.3. YOLO Algoritması .....	28
4.2. Dijital Görüntü İşleme Teknikleri.....	31
4.2.1. Görüntü pekiştirme .....	34
4.2.2 Görüntü filtreleme.....	35
4.2.3. Morfolojik işlemler .....	39
4.2.4. K-Means algoritması.....	42
<b>5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA.....</b>	<b>44</b>
5.2. Uygulamada Kullanılan Kamera ve Lens Özellikleri .....	53
<b>6. SONUÇLAR VE ÖNERİLER .....</b>	<b>56</b>
6.1 Sonuçlar .....	56
6.2 Öneriler .....	57
<b>KAYNAKLAR .....</b>	<b>58</b>
<b>EKLER.....</b>	<b>61</b>



## SİMGELER VE KISALTMALAR

### Simgeler

W : Watt

Lm : Lümen

Nm : Nanometre

K : Kelvin

HZ : Hertz

$\Delta E$  : CIELab renk sistemine göre iki renk arasındaki farklılık

### Kısaltmalar

CIE : Uluslararası Aydınlatma Komisyonu

CMY : Cyan (Cam göbeği), magenta, sarı

CMYK : Cyan (Cam göbeği), magenta, sarı, siyah)

RGB : Kırmızı, Yeşil, Mavi

HSV : Renk Tonu

BoS : Bag of Specials

BoF : Bag of Freebies

ALU : Aritmetik mantık Birimi

GPU : Grafik işlemci birimi

FPS : Saniyedeki kare sayısı

## 1. GİRİŞ

İnsanlar günlük yaşamlarında etrafındaki insanlar ile sürekli iletişim halinde olmakla beraber mutlu anlarında da birbirlerine gülümsemektedir. Gülümsemeyle beraber ön üst dişler dikkat çekmektedir. Bu yüzden diş sağlığının yanı sıra diş temizliği ve dişlerin hoş görünümü de ön plana çıkmaktadır. Dişlerde oluşacak hastalık vb. sebeplerden dolayı oluşacak kötü görünümü gidermek ve dişler arasında renk ahengini sağlayabilmek için estetik operasyonlar ve beyazlatma gibi müdahaleler yapılmaktadır[1]. Bu müdahalelerin başarılı olması durumunda insanlar kendini daha iyi ve özgüvenli hissetmektedirler.

Bu müdahaleler arasında diş hekimini en çok zorlayan meselelerden biride renk seçimidir. Diş rengini bulabilmek için önce dişin rengini ve o dişin komşu dişler ile renk ilişkisini tespit edebilmek gerekmektedir. İşlem uygulanacak dişin rengini belirlemede birçok etken devreye girmektedir. Bu etkenler arasında renk tespiti yapacak kişi veya alet, ortam ışığı ve rengi oluşturan materyaller bulunmaktadır[2].

Elektronik aletler ile diş rengi tespit etmek sübjektif bir yaklaşım olan göz ile renk tespit etmeye göre daha objektif bir yaklaşımdır. Uzmanlar diş rengi belirlemede hem görsel hem de elektronik ekipman destekli yapılmasını önermişlerdir[3]. Renk tespitinde en fazla kullanılan yöntem işlem uygulanacak diş ile kullanılacak olan skaladaki protez dişin göz yardımıyla tespit edilmesidir. Buradaki yanıltmalar dış faktörlerin yanında hekimin yaşı, göz yorgunluğu, renk körlüğü ve kişinin fizyolojik farklıları etki etmektedir[4]. Bu durumlardan dolayı spektrometre, spektrofotometre, kolorimetre, kamera gibi elektronik ekipmanlar geliştirilmiş ve diş hekiminin hizmetine sunulmuştur. Bu elektronik ekipmanların avantajları ve dezavantajları bulunmaktadır.

Diş rengini tespit etmede genel olarak Munsell renk sistemi ve CIE renk sistemi ve HSV renk sistemi kullanılmıştır. Munsell renk sistemi niteliksel özellikler taşıdığı için kliniklerde, CIE renk sistemi ise niceliksel özellik taşıdığı için genellikle araştırmalarda kullanılmaktadır[5].

İşlem uygulanacak dişin eski haline uygun ve yanındaki dişlerle uyumlu olmasını sağlamak için belirli renk skalaları kullanılmaktadır. Vitapan klasik renk skalası, Vita 3D-Master renk skalası, Chromascop renk skalası, Biodent renk skalası en fazla kullanılan renk skalaları arasındadır[3].

Bu tezde uygun kořullar altında çekilen diş görüntüleri bilgisayar ortamında işlenmiş ve Vita 3D-Master renk skalası göre eşleştirilmiştir. Eşleştirme sonucu elde edilen detaylı renk bilgisiyle hekimin işinin kolaylaştırılması sağlanmıştır.



## 2. KAYNAK ARAŞTIRMASI

Dijital görüntü işleme yöntemlerine ilgi iki temel uygulama alanı üzerinde bulunmaktadır. Bunlar: insan yorumu için resimsel bilgilerin geliştirilmesi ve resimsel bilgilerin depolanması, iletilmesi, çıkarılması gibi görevler için görüntü verilerinin işlenmesidir. Bir görüntü iki boyutlu bir işlev olarak  $f(x, y)$  şeklinde tanımlanabilir, burada  $x$  ve  $y$  düzlemsel koordinatlarıdır ve herhangi bir  $f$  koordinat çiftindeki  $(x, y)$  genliği, yoğunluğu veya o noktada görüntünün gri düzeyini göstermektedir. Dijital görüntü işleme alanı, dijital görüntülerin bir dijital bilgisayar vasıtasıyla işlenmesini ifade eder. Dijital bir görüntünün, her biri belirli bir konuma ve değere sahip olan sınırlı sayıda öğeden oluşur. Bu öğelere resim, görüntü, pels ve piksel denir. Piksel, dijital görüntünün öğelerini göstermek için en yaygın kullanılan terimdir. Anlamlı görüntü işleme görevlerini yerine getirecek kadar güçlü ilk bilgisayarlar 1960'ların başında ortaya çıktı. Dijital görüntü işleme, makinelerin kullanılabilirliği ve bu dönemde uzay programının başlangıcına ile başlamaktadır. Uzay uygulamalarına paralel olarak, dijital görüntü işleme teknikleri tıbbi görüntüleme, uzak dünya kaynakları gözlemleri ve astronomi ve birçok alanlarda da kullanılmaya başlanılmıştır (Gonzalez ve Woods, 2018).

Pandey ve Travieso (2017), "Processing Based Dental Image Analysis Using Automatic Gaussian Fitting Energy and Level Sets" adlı çalışmalarında 40 adet diş röntgeni görüntüleri üzerinde otomatik gauss dağılımı ile seviye setleri kullanarak dental görüntü analizi yapmışlardır. Bu çalışmada kistlerin ve iltihaplı lezyonların genellikle diş kökü çevresinde meydana geldiğini ve radyografilerdeki bu alanların genellikle gürültüye, zayıf kontrasta ve çok düzensiz aydınlatmaya maruz kaldığını belirtilmiştir. Kullandığı görüntülerin yerel yoğunlukları ve diş konturlarının doğru segmentasyonu için seviye ayarlama fonksiyonu ile birleştiren gauss dağılımları ile tanımlamış ve çalışmanın gerçek zamanlı çalıştığını gösterilmiştir.

Díaz, Morales ve Torres (2016), "Study Of The Temporal Evolution Of Whitening In Teeth Undergoing Mouthwashes Using Digital Image Processing" adlı çalışmalarında dişlerin yaş, cinsiyet, ırk gibi belirli faktörlere bağlı olarak birden fazla tonu ve rengi olduğunu aynı zamanda dişlerin ilaçlar ve kimyasal maddelere karşı hassas olduğu bunun da özellikle diş gelişimi aşamasında hem diş yapısının hem de diş renginin etkilenmesine yol açtığını ifade etmektedir. Bu nedenle ağız gargarası kullanımından kaynaklanan dişlerde ağarma derecesini analiz eden görüntü işleme algoritması tasarlamış ve

uygulamıştır. Bu algoritma öncelikle bir insan dişi görüntüsü yakalamış bu görüntüyü sayısallaştırıp gri tonlamalı ön işleme gerçekleştirmiştir. İşlenen görüntü üzerinde histogram analizi uygulamıştır. Çalışma sonucunda belirli zaman aralıklarında kimyasal çözelti içerisinde bırakılan dişler dijital olarak analiz edilmiştir.

Abdallah ve Alqahtani (2019), “Research in Medical Imaging Using Image Processing Techniques” adlı çalışmalarında görüntü işleme teknikleri ile tıbbi görüntüleme araştırmaları yapmışlardır. Bu çalışmada genel olarak medikal çalışmalarda kullanılan x-ray, CT gibi görüntüleme teknikleri görüntü işleme de kullanılan segmentasyon ve kenar tespit algoritmaları hakkında bilgi verilmiştir.

Setianingrum, Rini ve Hakiem (2017), “Image segmentation using the Otsu method in Dental X-rays,” adlı çalışmalarında görüntü işleme teorisine dayalı diş röntgenlerini kullanarak sağlık koşullarını belirlemeyi amaçlamışlardır. Bu çalışma için 10 farklı hastadan 40 adet diş röntgeni elde ederek sağlıklı ve hastalıklı dişler olarak gruplandırmıştır. Görüntü kalitelerinde iyileştirme, kontrast ve parlaklık ayarlaması yapıp otsu yöntemi kullanarak görüntü bölütleme işlemi uygulamıştır. Otsu sonucunun eşik değerlerini ayarlanmasının diş çürümesinin belirlenmesinde etkisi olduğunu göstermiştir.

Lehmann, Troeltsch, Spitzer (2002), “Image processing and enhancement provided by commercial dental software programs” adlı çalışmalarında Almanya’nın Köln şehrinde yapılan 28. Uluslararası Dental Show isimli programa katılan dijital dental görüntüleme de kullanılan çeşitli ticari yazılımları incelenmiştir ve algoritmalar hakkında bilgi verilmiştir. Görüntü işlemenin standartlaştırılmış terminolojisi ve artırılmış işlevselliği diş hekimliği mesleğine sunulmasını istenmiştir.

Restoratif diş hekimliğinde en zor aşamalardan biri dişlerin doğal yapısına uygun değişimlerin yapılmasıdır. Estetik restoratif diş hekimliğinde renk seçimi yapabilmek için renk bilimi iyice kavranmalıdır. Diş renginin belirlenmesi uzman görüşü ve bilgisayar destekli sistemler olarak iki kategoride incelenebilir. (Kahramanoğlu ve Özkan, 2013). Estetik restorasyonun başarılı olabilmesi için etrafında bulunan diğer dişler ile renk farklılığının farklı ortamlarda gözle algılanamayacak biçimde olmalıdır. Diş renginin belirlenmesinde dental renk skalaları kullanılmakta ve genellikle renk seçiminde kullanılacak materyale ait renk skalası tercih edilmelidir (Saraç ve Yüzbaşıoğlu, 2006). Bir cisim ısıtıldığında renk değişimi gözlenir çünkü renkler arası geçişlerde sıcaklık değişimi olur bu yüzden her renge ait karakteristik bir renk sıcaklığı bulmaktadır. Bu

sıcaklık değeri Kelvin (K) olarak ifade edilmiştir ve gün ışığının renk sıcaklığı 5500 K'dir. Renk sıcaklığında eşit miktarda kırmızı, mavi ve yeşil renk bulunmaktadır ve bu renkler beyaz rengi oluşturmaktadır. Gün ışığı zamana, hava durumuna ve kirliliğe bağlı olarak farklı dalga boylarında yayılmasına sebep olmaktadır (Müdüroğlu, Kıvrak ve Nalçacı, 2018). Herhangi bir yüzeyde rengin tespit edilebilmesi için ışığa ihtiyaç vardır. Bir cisim üzerine düşen ışığın bir kısmı yüzey tarafından absorbe edilir. Absorbe edilmeyen diğer kısım göz tarafından algılanır ve beyin tarafından belirli bir renk olarak tanınır. Buradaki absorbe miktarı metalin yapısı ile ilgilidir. Cismin yüzeyinden yansıyan dalga boyundaki ışınlar rengin seçilmesini sağlar. Dişlerin doğal renk görünümünü sağlama bilmek için floresans, opalesans ve translusens adlı optik özellikleri bilmek gerekmektedir. Floresans (ışılma), bir cismin üzerine düşen ışığın absorbe edilmesi ve sürekli olarak uzun dalga boyları halinde yansıtılmasıdır. Opalesans, bir cisim üzerinde ışık yansıdığı anda bir renk, cismin içinden geçen ışığın farklı bir renk olarak görülmesi olayıdır. Translusensi ise ışığın geçirgenlik yani saydamlığının derecesini ifade etmektedir (Doğan ve Yüzügüllü, 2011).

Işık farklı dalga boylarında oluşmaktadır ve aynı farklı yerlerden farklı renklerde görülebilir bu olaya metamerizm denmektedir. Diş üzerine ışık düştüğü zaman; ışığın diş dokuları boyunca iletimi, ışığın doğrudan yüzeyden yansması, yüzeyde dağılmış ışığın yansması ve diş dokuları içinde ışığın absorbe edilmesi veya saçılması gibi dört temel olay yaşanmaktadır. Dişin dişeti kenarlarına yakın kısımları, dentine daha yakın komşulukta olması nedeniyle daha koyu görünür. Çoğu kişide kanin dişleri santral ve lateral dişlere kıyasla daha koyu, geç insanlarda primer dentisyon nedeniyle karakteristik olarak daha açık görülmektedir (Çağlar ve Yamanel, 2007).

Singried Forsius 1611 yılında rengin üç boyutlu olduğunu göstermiştir. Rengin bu özelliğini açıklamak için günümüze kadar birçok yaklaşım tanımlanmıştır. Bu yaklaşımlar içerisinde Munsell ve CIE (Commission Internationale de Lieclairaęe) renk sistemleri uluslararası kabul edilebilirliği ve pratik kullanılabilirliği açısından en çok tercih edilen sistemlerdir. Munsell renk sistemi Albert H. Munsell tarafından rengin üç boyutunu hue (renk tonu), value (renk değeri) ve chroma (renk yoğunluğu) olarak 1905 yılında tanımlamıştır. Hue, rengi diğer renklerden ayıran özelliktir. Value, rengin açıklık-koyuluk derecesi veya bir cismi parlaklığını gösteren özelliktir. Chroma ise güçlü bir rengin zayıf renklerden ayrılmasını sağlayan ve doygunluk derecesini gösteren özelliktir. Yoğunluk Vita Renk skalasında A (kırmızımsı kahverengi), B (kırmızımsı sarı), C (Gri), D

(Kırmızımsı gri) olmak üzere dört temel renge ayrılır. A renginin beş tonu diğer renklerin ise dört ayrı tonu bulunmaktadır (Kurt ve Bal, 2016).

İlk renk sistemi CIE tarafından 1931 yılında tanımlanmış ve prensipleri değiştirilmeden sürekli geliştirilmiştir. Bu sisteme göre X, Y ve Z parametreleri kullanılmış ve gözlemci tarafından spektral cevap fonksiyonları tanımlanmıştır. CIE chromaticity diyagramı kesin renk tanımı içinde kullanılmaktadır.  $L^*$ ,  $a^*$  ve  $b^*$  değerleri üç uyaranlı x, y, z değerlerinden hesaplanmaktadır. CIE sisteminde üç boyutlu renk aralığının yaklaşık olarak düzenlemesi diğer sistemlere göre avantaj sağlamaktadır. CIE koordinatlar, renk tonuna bağlı olmaksızın, iki rengin şiddeti arasındaki mesafenin algı şiddetine oranlanması yolu ile daha eşdeğer bir renk skalasına getirilmesini öneren bir komisyondur. Bu sistemde bazı genel bilgiler harfler halinde kodlanmış ve renk farklılığı ( $\Delta E$ ) formülü belirlenmiştir (Bayındır ve Wee, 2006).

Keyf, Uzun ve Altunsoy (2009), “Diş Hekimliğinde Renk Seçimi” adlı çalışmada estetik restoratif diş hekimliğinde doğal dişe uygun renk seçimi için maddeler halinde uyulması gereken genel prensiplerden bahsetmişlerdir.

Öngül, Çelik, İlbey ve Şermet, (2013), “Türk Toplumundaki Genç Bireylerde Diş renk Dağılımının İncelenmesi” adlı çalışmalarında çürük ve restorasyon olmayan ve son bir yıl içinde beyazlatma uygulanmış ön santral, leteral ve kanin dişleri bulunan 18-22 yaş grubunda bulunan 164 kişinin dişleri üzerinde çalışmada bulunmuş ve Vita Klasik ve 3D Master skalası ile CIE renk sistemine göre yaptıkları çalışmada genç bireylerin Vita skalasına göre B ve C tonlardan olduğu D tonuna hiç rastlanmadığını belirlemişlerdir. Ayrıca CIE sistemine göre kanin dişlerinin diğer dişlere kıyasla,  $L^*$  değerlerinin daha düşük,  $a^*$  değerlerinin daha yüksek olduğu tespit edilmiştir.  $L^*$  değerleri cinsiyete göre değerlendirildiğinde; bayanların erkeklere göre santral dişlerinde  $L^*$  parlaklık değerinin daha fazla olduğu saptanmıştır ( $p < 0.05$ ). Tüm bireylerdeki en düşük  $\Delta E$  değerleri santral-lateral dişleri arasında, en yüksek değerler ise santral ve kanin dişleri arasında olduğu tespit edilmiştir. Bu çalışma sonucunda genç bireylerde anterior bölge restorasyonlarında tek bir renk seçilmesi doğal bir estetiğin elde edilmesini zorlaştıracığı da belirtilmiştir.

### 3. DIŐ RENGİ TESPİTİ

Renk genel olarak objeleri birbirinde ayırmamıza yardım eden gözde oluşan duyumdur. Renk algısında rol oynayan üç önemli faktör vardır. Bu faktörler rengi tanımlayacak olan gözlemci, rengi yansıtacak olan obje ve objenin üstüne düşen ışık kaynağıdır. Rengin algılanabilmesi için ilk önce objenin üzerine ışık düşer ve obje bu ışığı yansıtır gözlemci ise yansıyan ışığı tanımlar[6].

Işık kaynağı rengi oluşturan en önemli bileşendir. Işık kaynağı olmadan renk oluşmaz. Dalga boylarından oluşan ışık kırıldığında spektrum oluşur. Oluşan elektromanyetik spektrumdan gamma ışınları, radyo dalgaları, mor ötesi, kızıl ötesi dalgaları ve görünür meydana gelmektedir. Elektromanyetik dalgalardan oluşan görünür ışık 400-700 nm dalga boyu aralığında bulunan ve insan gözü ile algılanabilen dalga boyunu tanımlar. Manyetik ve elektrik alanın birbirine dik olarak salınmasıyla oluşan elektromanyetik dalganın küçük bir bölümü insan gözü ile görülebilir. Işık temel olarak kırmızı, mavi, yeşil renkler ile oluşur ve diğer tüm renkler bu üç rengin belirli oranlarda karışmasıyla elde edilir. Objeden yansıyan rengi düzgün bir şekilde belirleyebilmek için ideal bir ışık kaynağı gerekmektedir ve bu ideal ışık kaynağının özellikleri şu şekilde olmalıdır:

- Işık kaynağı tüm spektrumları içermelidir.
- Yayılan ışın yoğunluğu yüksek olmalıdır.
- Gözü rahatsız eden renk yaymamalıdır.
- Kalitesi ve kantitesi zamana ve yere göre değişmemelidir.

Işık yoğunluğu düşük ise ince detayların algılanması zorlaşır eğer ışık yoğunluğu yüksek ise rengin algılanmasını zorlaştırır. Renk seçimini doğru yapılabilmesi için ışık yoğunluğunun 800 ile 2700 lux arasında bulunan ışık kaynakları tercih edilmelidir. Dental unit ışıkları yüksek oranda kırmızı-sarı renkleri düşük oranda mavi renkleri içerdiği için kullanılmamalıdır. Genel olarak gün ışığı rengi veren floresan veya akkor ışık kaynakları tercih edilmelidir. Bir ışık kaynağının kalitesi yaydığı ışığın renk sıcaklığına bağlıdır ve birimi Kelvin'dir. Uluslararası Aydınlatma Komisyonu CIE, spektral güç dağılımına göre ışık kaynaklarını isimlendirmiştir. Bunlardan genel olarak kullanılanları şu şekildedir:



A: 2856° Kelvin renk sıcaklığında genel olarak ev ortamında kullanılan akkor tipi ışık kaynağı.

D50: Gün doğumu ya da gün batımını temsil eden 5000° Kelvin renk sıcaklığındaki ılık gün ışığıdır.

D65: Gün ışığı 6504° Kelvin renk sıcaklığını ifade eder. Aynı zamanda D55 5500° Kelvin, C 6774° Kelvin, D75 ise 7500° Kelvin renk sıcaklıkları da gün ışığının renk sıcaklığını belirtmektedir.

F2: Soğuk beyaz ışık olarak adlandırılan florasan ışık kaynağı 4230° Kelvin renk sıcaklığındadır.

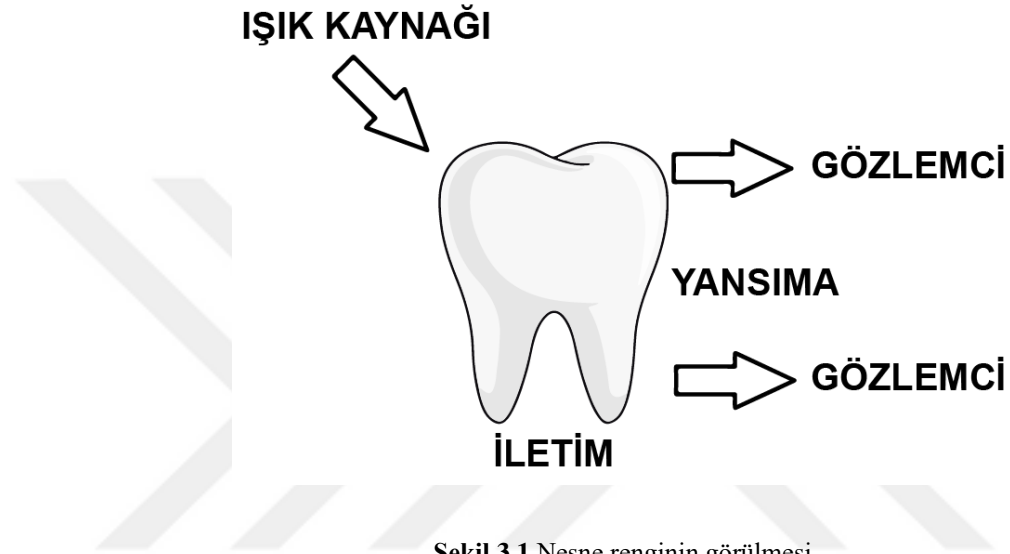
F7: Gün ışığını temsil eden geniş bantlı florasan ışığıdır.

Bu isimlendirmelere göre diş rengi tespitinde genel olarak A, D50 ve D65 gruplandırmasında bulunan ışık kaynakları tercih edilmiştir. Eğer ışık kaynağı olarak güneş ışınları kullanılacaksa değerlendirmenin öğlen saatlerinde yapılması daha doğru bir sonuca ulaşılması için gerekli olduğu düşünülmektedir[1].

Renk, ışığın obje üzerinde yansımaları, geçmesi veya absorbe edilmesiyle oluşmaktadır. Objeye aynı renkte olan ışık dalga boyları yansır diğer dalga boyları ise obje tarafından absorbe edilir. Objeler ışığın tamamını absorbe ederse siyah renk, tamamını yansıtırsa temel renklerin hepsini içeren beyaz renk oluşur. Opaklık, translüent ve transparan materyaller olarak ifade edilen bu nesnelere ışığı yansıtması ve absorbe etmesine göre gruplandırılmıştır. Işığın bir kısmını absorbe eden bir kısmını ise yansıtan objeler opak materyaller, ışığın bir kısmını geçiren objelere translüent materyaller, ışığın tamamını geçiren objelere ise transparan materyaller denmektedir. Değişik yapıda bulunabilen dişler translüent yapıdadır. Diş rengini veren dentindir. Dentin kalınlığına ve mine şeffaflığına göre değişen bu renk dentinde toplanır ve mineden yansımalarıyla elde edilir.

İnsan bir objeye baktığı zaman objeden yansıyan ışık düz bir hat şeklinde göze ulaşır. Yansıyan bu ışık kornea, pupila ve göz merceğinden geçer. Göz merceği ve kornea ya ulaşan ışık kırılarak retina üzerinde toplanması sağlanır. Retina üzerinde bulunan fotoreseptörler yansıyan ışığı elektrik akımına dönüştürerek optik sinirler vasıtasıyla elektrik akımının beyne ulaşması sağlanır. Beyinde gelen bu akım sinyallerini cismin diğer özellikleriyle birlikte görüntüye çevirir. Fotoreseptörlerin ışığı daha iyi

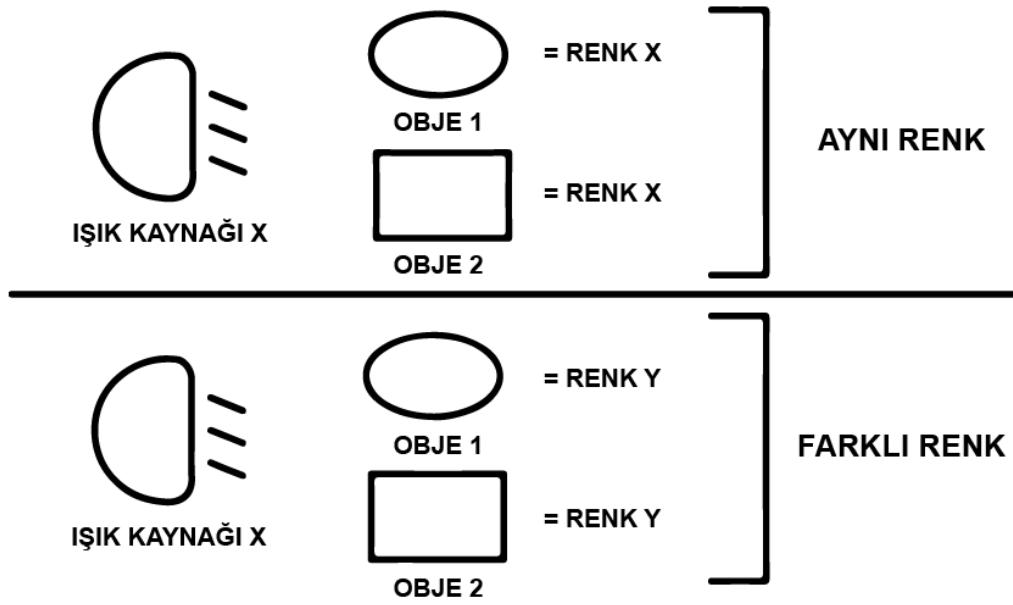
algılayabilmesi için bol ışık kaynağı ve dar açılı alanların kullanılması gerekmektedir. Gözün bir renk ile sürekli uyarılması gözün yorulmasını ve göz tepkisinin azalmasını tetiklediği için rengin görülmesinde optik sinirlerin tepkisi büyük rol oynamaktadır. Renk algılayan mekanizmaların herhangi bir bölgesinde oluşacak bozukluk renk körlüğüne sebep olmaktadır. Bu yüzden farklı kişiler aynı rengi farklı renklerde algılayabilmektedirler. Şekil 3.1’ de gözlemcinin nesneyi nasıl gördüğü gösterilmiştir[5].



Şekil 3.1 Nesne renginin görülmesi

### 3.1. Metamerizm

Işık kaynağının özellikleri, çevrede bulunan renkler ve objelerin özellikleri gibi çevresel faktörler renk seçimini etkileyen önemli faktörlerdir fakat aynı ışık kaynağı ile aynı renkte gözükten objeler farklı ışık kaynakları altında farklı renklerde görülebilmektedir. Metamerizm adı verilen bu olay renk secimi yaparken göz önünde bulundurulması gereken bir özelliktir. Bu yüzden renk belirlerken standart ve verimli bir ışık kaynağı tercih edilmelidir. Gün ışığında mevsimlerin ve hava şartlarının etkisi yüksek olduğu için standart bir gün ışığına sahip olmak mümkün değildir. Farklı ışık kaynakları farklı renk dalgaları yayacağından objeden yansıyan ışığında dalga aralığı farklı olacaktır bu sebepler standart bir ışık kaynağı elde edebilmek için renk ısı, spektral yansım eğrileri ve Color Rendering Index gibi parametreleri göz önüne alarak ışık kaynaklarında standart gün ışığı rengi elde edilmeye çalışılmıştır. Şekil 3.2’ de metamerizm gösterilmiştir[4].



Şekil 3.2 Metamerizm

### 3.2. Diş Rengi Seçiminde Kullanılması Gereken Prensipler

Ortam bulunan ışık kaynağı, kullanılan skala ve çevre şartlarına bağlı olarak diş rengi belirlenirken uygulanması gereken bazı prensipler vardır. Bu prensipleri uygulamak renk doğruluğunu ve tekrarlanabilirliğini olumlu ölçüde arttıracak adımlardır. Bu adımlar şöyledir[1];

1. Retinanın renge daha duyarlı olabilmesi için hasta ve hekimin göz seviyesinde bulunmalıdır.
2. Renk tespiti yapılacak olan dişin yüzeyi temiz olmalı kendi rengini yansıtabilmelidir.
3. Diş tespiti yapılacak olan mekânın duvar rengi gözü az yoran renklerde ve diş rengini tespit etmemizi zorlaştıracak renklerde olmamalıdır. Bunun için önerilen renk gri tonlarıdır.
4. Renk belirlemede kullanılan diş renk skalasında dişe en yakın renk seçildikten sonra doğal dişin ve yapay dişin kesici kenarları birbirine tutularak doğrulama veya deneme yapılmalıdır.

5. Mekân dışarıdan gelen güneş ışını vs. ışık kaynaklarından izole edilmeli ve uygun bir ışık kaynağıyla renk tespiti yapılmalıdır.
6. Hastanın üstü gri bir önlük ile kapatılmalıdır.
7. Dental ünit ışığı kullanılmamalı ışığı kullanılmamalı bunun yerine 5500 K civarında renk sıcaklığına sahip akkor veya floresan ışık kaynakları tercih edilmelidir.
8. Renk secimi tedavi öncesi yapılmalıdır.
9. Dişlere 5 saniyeden fazla bakılması ana renkte sapmalara neden olmakta ve diş rengi ölçümlerini çok fazla tekrarlamak göz yorgunluğuna sebebiyet verdiği için ilk ölçümleri seçmek daha faydalı bulunmaktadır.
10. Diş rengi belirlemede zorlanılıyorsa turuncu veya sarı renkleri temsil eden gruplara bakmak faydalıdır çünkü diş renkleri büyük oranda bu renk aralığı arasındadır.
11. Işığın farklı açılardan diş üzerinden nasıl yansıdığını belirlemek için renk tespiti yapılacak dişe farklı yönlerden bakılmalıdır.

Sıralanan bu kurallar rengi belirleyecek olan hekime ve kamera vasıtasıyla renk tespit eden sistemlerin kararlılığına yardımcı olacak ve sabit kriterler belirleyecektir.

### 3.3. Diş Skalaları

Diş rengi belirlemede renk referansı olarak kullanılan renk skalaları birçok firma tarafından farklı sayılarda örnekler, farklı doygunluk ve parlaklık ayarında kendilerine has standartlaştırılmış yapay diş örnekleridir. Daha öncede bahsedildiği gibi aydınlatma türü ve gücü, göz yorulması gibi koşullar renk belirlemesini zorlaştıran etkenlerdir. Bu skalalar sayesinde hata oranlarını azaltılmak daha doğru sonuçlar elde edilmek istenmiştir. Diş hekimliğinde daha çok Vita renk skalaları kullanılmaktadır ve bu skalalar doğal dişlere göre düşük doygunluk ve yüksek parlaklığa sahip skalalardır. Genel olarak Vita Classical ve VITA Toothguide 3D-MASTER ve Ivoclar Chromascop en çok kullanılan skalalardır. Bu skalalar aynı yerde üretilseler dahi bazen çok ufak renk farklılıkları oluşmaktadır. Bu yüzden diş skalalarının önemi yanı sıra renk tespiti yapacak hekiminde tecrübesi önemli faktörler arasındadır.

Vita Classical skalasında bulunan 4 farklı grupta bulunan yapay dişlerin renk tonunu ifade eden harfler Tablo 3.1 de ifade edilmiş Şekil 3.3’ de ise skalanın görünümü gösterilmiştir[22].

**Tablo 3.1** Vita Classical renk tablosu

Harf Kod Aralığı	Rengi
A1 -A4	Turuncu (kırmızımsı-kahverengi)
B1-B4	Sarı (Kırmızımsı-sarımsı)
C1-C4	Grimsi tonlar
D1-D4	Kırmızımsı-gri



**Şekil 3.3** Vita Classical Renk skalası

VITA Toothguide 3D-MASTER skalasında 29 adet örneklenmiş yapay diş rengi bulunmaktadır. 3 farklı harf kodu ve 5 farklı sayı kodu ile eşlenerek renk kodları oluşturulmuştur. Bu skala diğer skalalara göre daha fazla örnek ve renk vardır. Genel olarak L sarı tonları, M gri tonları, R ise turuncu renkleri ifade ederken sayılar ise renk doygunluğunu ifade etmektedir. Şekil 3.4’ de skalanın görünümü gösterilmiştir.

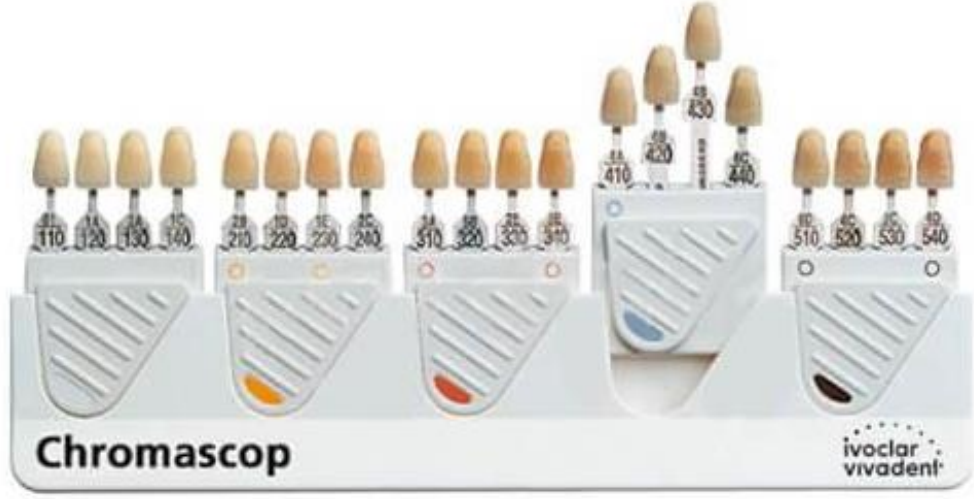


**Şekil 3. 4** VITA Toothguide 3D-MASTER skalası

Ivoclar Chroma renk skalasında 5 grupta toplam 20 adet diş renk örneği kullanılmıştır. Bu renkleri ifade etmek için sayısal kodlama kullanılmıştır. Tablo 3.2 de sayısal kodlamanın renk karşılığı gösterilmiştir. Şekil 3.5’ de Ivoclar Chromascop skalasını görünümü gösterilmiştir[23].

**Tablo 3.2** Ivoclar Chromascop Renk Tablosu

Sayısal Kod	Renk
100	Beyaz
200	Sarı
300	Turuncu
400	Gri
500	Kahverengi



Şekil 3.5 Ivoclar Chromascop skalası

### 3.4. Diş Rengi Belirlemede Kullanılan Elektronik Aletler

Kanada menşeli Cortex Machina şirketi 1990 yılında dijital renk secim sistemlerini piyasaya sürmüştür. Bu makinelerde şık kaynağının etkisini kaldırmış hızlı ve güvenilir sonuçlar elde etmiştir. Bu renk ölçüm cihazlarını diş yüzeyinin ufak bir bölümde ölçüm yaparak ve bütün diş yüzeyinde ölçüm yaparak iki kategoride renk tarama işlemi yapmışlardır. Genel olarak bu tip makinelerde CIELab renk sistemi kullanılmıştır. Renk ölçüm cihazlarında güvenilirlik, tekrarlanabilirlik ve doğruluk en önemli parametrelerdir. Dijital renk seçimde RGB cihazlar, spektrofotometreler, kolorimetreler ve spektroradyometreler kullanılmaktadır.

#### 3.4.1. RGB Cihazlar

RGB cihazlar kameralar vasıtasıyla elde edilen video veya fotoğraf görüntüleri üzerinden renk tespiti yapan cihazlardır. Laboratuvar ortamında hızlı ve kullanışlı olmaları sebebiyle kullanılmaktadır ancak bir ölçüm cihazı olmadığı için subjektif değerlendirmeye açık cihazlardır. Standart ışıklandırma ve doğru açıyla çekinilen görüntülerde doğru sonuçlar alınabilmektedir.

### 3.4.2. Spektrofotometre

Spektrofotometreler, bir nesnenin yansıttığı veya geçirdiği görünebilir elektromanyetik dalgayı ölçebilen bir cihazdır. Spektrofotometre içerisine konulan nesneye farklı açılarda ışına maruz bırakarak nesne rengini tespit etmeye çalışır. Bu işlemde beyaz ışığı 10-20 nm dalga boylarına ayran yarıklar veya prizmalar kullanarak yapar. Bir spektrofotometre sadece küçük bir alanı tarayarak yapar fakat teknolojinin ilerlemesi ve ihtiyaçların doğmasıyla birlikte tüm alanı tarayan spektrofotometrelerde yapılmıştır[5].

Taşınabilir spektrofotometrelerden olan Vita Easyshade cihazı fiber optik kabloyla bir el parçasına bağlıdır. Bu el parçasının ucunda 5mm çapında halojen bir ışık kaynağı bulunmaktadır. Yüksek doğruluk kapasitesine sahip bu cihaz Vita 3D master skalasıyla uyumlu sonuçlar üretmektedir. Diş hekimliğinde kullanımı oldukça popüler olan Vita grubu cihazları genel olarak CIELab renk sistemini kullanır ve Vita renk skalasına göre sonuç vermektedir. Şekil 3.6’ da VITA Easyshade V cihazı gösterilmiştir[23].



Şekil 3.6 VITA Easyshade V Spektrofotometre



2006 yılında Japonya menşeli CrystalEye isimli cihaz endüstride kullanılmaya başlanmıştır. Bu cihaz ağız içi fotoğrafını çeker ve yapılan diş protezini yapay bir çene modeli üzerinde fotoğraflar. Yapılan çalımlarda doğal diş rengi ile renk uyumu yakaladığı ve CIELab renk sistemine göre daha düşük  $\Delta E$  değerine sahip olduğu tespit edilmiştir[5]. Şekil 3.7’ de Crystaleye Spektrofotometresi gösterilmiştir[24].



Şekil 3.7 Crystaleye Spektrofotometre

Spektrofotometrik analiz ile dijital görüntüleme analizini birleştiren SpecroShade isimli cihaz İsviçre’ de icat edilmiştir. Bütün diş üzerinde ölçüm yapan bu sistem spektrofotometreye bağlı halojen ışık kullanan dijital kamerayla ölçüm yapmaktadır. Yüksek doğruluk oranına sahip bu cihaz taşınabilir değildir ve optik başlığı oldukça büyük ve maliyetli olduğu için dezavantajlı modeldir. Şekil 3.8’ de SpecroShade Spektrofotometresi gösterilmiştir[25].



Şekil 3.8 SpectroShade Spektrofotometre

### 3.4.3. Spektrometre

Toplayıcı optik, monokromatör, dedektör ve okuyucu bölümlerinden oluşan spektrometreler spektral güç dağılımını ölçmek için tasarlanmıştır. Diş üzerinden yansıyan elektromanyetik dalgayı toplayıcı optik vasıtasıyla monokromatör uygun ışın demeti şeklinde iletilmektedir. Monokromatör gelen ışın demetini prizmadan geçirerek farklı dalga boylarında ve açılarda dağıtır. Burada dar bir band aralığın ışın dağıtılır. Odaklayıcı optiğe gelen ışın demeti daraltılarak dedektöre ulaştırılır. Dedektör daraltılan elektromanyetik gücü algılayarak okuyucuya elektriksel sinyal gönderir. Spektrometrelerde dalga boylarını ayırmak için difraksiyon kafesleri kullanılmaktadır fakat hiçbir zaman bu dağıtma işlemi doğru bir şekilde yapılamamaktadır. Bunun nedeni saçılması istenen dalga boyuna komşu dalga boylarının yansması istenilen saçılma şeklini engellemektedir. Bu olay spektral saçılma olarak da adlandırılmaktadır ve kafesler, aynalar veya monokromatörün neden olduğu düşünülmektedir[27].

Renk ölçümünde kullanılan spektrometrelerin toplayıcı optiğinde teleskop tip lens bulunur ve nesne üzerindeki bir noktadan elektromanyetik güç ölçülerek renk tespiti yapılır. Bu tip cihazlara tele-spektrometre adı verilmektedir.

Diş yüzeylerinin yapısı itibarıyla düz olmaması ve translusent yapısı nedeniyle diğer renk ölçüm cihazlarına göre daha az kullanılmaktadır.

### 3.4.4. Kolorimetre

Diş hekimliğinde sıklıkla kullanılan kolorimetreler standart renk kalibrasyonuna dayanarak diş rengini analiz eden cihazlardır. Diş üzerinden yansıyan ışığın yoğunluğunu kırmızı, yeşil ve mavi filtrelerden geçirerek ölçüm yapmaktadır. İçerisinde ışığın spektral özelliklerini sınırlayan silikon fotodiyot bulunduran bu cihazlar gelen ışığı filtreler. Detaylı ölçüm yapmazlar.

Diş hekimliğinde sıklıkla kullanılan modelleri arasında Shade Eye NCC marka cihazın taşınabilir ucu 3mm çapında olup dişe temas ettirerek diş üzerine ışık gönderilir ve diş yüzeyinden yansıyan ışığı toplar. Toplanan ışık detektörlere iletilerek filtrelerden geçirilir. Hafızasında bulunan örneklere göre rengi en yakın örneğe göre sonuç verir. Şekil 3.9’ da Shade Eye NCC kolorimetresi gösterilmiştir[26].



Şekil 3.9 Shade Eye NCC kolorimetre

## 4. MATERYAL VE YÖNTEM

Diş renginin görüntü işleme yolları ile tespit edilebilmesi için ışık ve renk bilimini temel olarak kavranması ve diş görüntü üzerindeki etkilerinin iyi tespit edilmesi gerekmektedir. Aynı zaman bu renklerin bilgisayar ortamında nasıl algılandığı ve görüntüler üzerinde yapılacak olan manipülasyon işlemlerinde ne ve hangi sırada olması gerektiği bilinmelidir.

### 4.1. Işık ve Elektromanyetik Spektrum İlişkisi

1666'da Sir Isaac Newton, bir güneş ışınını cam prizmadan geçirdiğinde, ortaya çıkan ışığın beyaz olmadığını ve spektrumun bir uçtan menekşe diğer uçtan kırmızıya kadar değişen bir renk spektrumu oluştuğunu keşfetti. Renk spektrumu menekşe(mor), mavi, yeşil, sarı, turuncu ve kırmızı olmak üzere altı geniş bölgeye ayrılır. Renkler tam olarak görüntülendiğinde, spektrumdaki hiçbir renk aniden bitmez; bunun yerine, her renk bir sonrakine sorunsuzca karışır. Temel olarak, insanların ve diğer bazı hayvanların bir nesnede algıladığı renkler, nesneden yansıyan ışığın doğası tarafından belirlenir. Görünür ışık elektromanyetik spektrumdaki diğer bantlara göre daha dar bir frekans bandında bulunmaktadır. Görünür ışıkta algıladığımız renk aralığı elektromanyetik spektrumun küçük bir kısmıdır. Spektrumun bir ucunda dalga boyları görünür ışıktan milyarlarca kat daha uzun olan radyo dalgaları bulunur. Spektrumun diğer ucunda, görünür ışıklardan milyonlarca kez daha küçük dalga boylarına sahip gama ışınları bulunur.

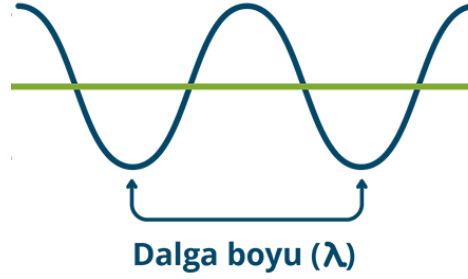
Elektromanyetik spektrum dalga boyu, frekans veya enerji olarak 4.1 denkleminde ifade edilmiştir. Dalga boyu ( $\lambda$ ) ve frekans ( $\nu$ ) ifadesi ile ilişkilidir. Burada  $C$  ışık hızıdır ( $2.998 * 10^8$  m / s).

$$\lambda = C/\nu \quad (4.1)$$

Elektromanyetik spektrumun enerjisi 4.2 denklemindeki ifade ile verilir.

$$E = h \times \nu \quad (4.12)$$

Burada  $h$  Planck'ın sabitidir. Frekans Hertz (Hz) cinsinden ölçülür, bir Hz saniyede bir sinüzoidal dalganın bir döngüsüne eşittir. Enerji birimi elektron voltur.



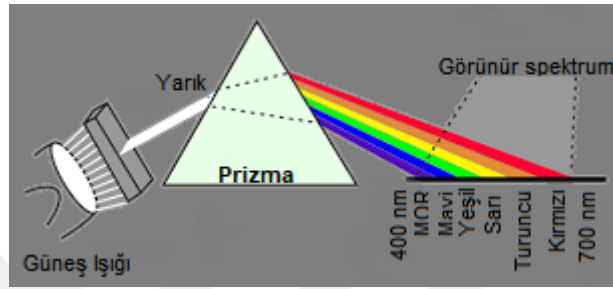
Şekil 4.1 Elektromanyetik dalga

Elektromanyetik dalgalar, dalga boyu  $\lambda$  ile sinüzoidal dalgalar yaymak olarak Şekil 4.1'deki gibi görselleştirilebilir veya her biri dalga benzeri bir desende seyahat eden ve ışık hızında hareket eden kütleli parçacık akışı olarak düşünülebilir[28]. Her kütleli parçacık, foton adı verilen belirli bir miktarda (veya demet) enerji içerir. Denklem 4.2'ye göre en düşük enerjili fotona sahip radyo dalgalarıdır ve sırasıyla mikrodalgalar, kızılötesi, görünür ışık, mor ötesi, X-ışınları ve gama ışınlarıdır. Bir elektromanyetik spektrum da en yüksek enerjili fotonlara sahip olan gamma bantıdır. Yüksek enerjili elektromanyetik radyasyon, özellikle X-ışını ve gama ışını bantlarında canlı organizmalar için zararlıdır. Işık, göz tarafından algılanabilen bir tür elektromanyetik radyasyondur.

Bir nesnede algılanan renkler, nesnenin yansıdığı ışığın doğası tarafından belirlenir. Görünür tüm dalga boylarında nispeten dengeli ışığı yansıtan bir gövde, gözlemciye beyaz görünür. Bununla birlikte, görünür spektrum sınırlı bir band aralığında yansıtmayı destekleyen bazı renk tonları sergiler. Örneğin, yeşil nesnelere, diğer dalga boylarındaki enerjinin çoğunu emerken, esas olarak 500 ila 570 nm aralığında dalga boylarına sahip ışığı yansıtır. Rengi geçersiz olan ışığa tek renkli (veya akromatik) ışık denir. Tek renkli ışığın tek özelliği yoğunluğudur. Monokromatik ışığın yoğunluğunun siyahtan grilere ve son olarak beyaza göre değiştiği algılandığından, gri seviyesi terimi genellikle monokromatik yoğunluğu belirtmek için kullanılır. Siyahtan beyaza monokromatik ışığın değer aralığı genellikle gri skala olarak adlandırılır ve tek renkli görüntülere genellikle gri tonlamalı görüntüler denir.

Işık akromatikse (renk boşluğu), tek özelliği yoğunluğu veya miktarıdır. Kromatik ışık elektromanyetik spektrumu yaklaşık 400 ila 700 nm arasında kapsar. Şekil 4.2' de

yarıktan geçirilen güneş ışığının görünür ışık bandı gösterilmiştir[2]. Kromatik ışık kaynağının kalitesini tanımlamak için kullanılan üç temel bulunmaktadır. Bunlar radiance, luminance ve parlaklıktır. Radiance, ışık kaynağından akan toplam enerji miktarıdır ve genellikle watt (W) olarak ölçülür. Luminance, Lümen (lm) cinsinden ölçülen parlaklık birimidir ve bir gözlemcinin bir ışık kaynağından algıladığı enerji miktarının bir ölçüsüdür. Parlaklık ölçülmesi neredeyse imkânsız olan öznel bir tanımlayıcıdır. Akromatik yoğunluk kavramını somutlaştırır ve renk hissini açıklamada anahtar faktörlerden biridir.



Şekil 4.2 Görünür ışık spektrumları

CIE Kırmızı (R), yeşil (G) ve mavi (B) renklerini standartlaştırmak amacıyla, 1931'de üç ana renge aşağıdaki özel dalga boyu değerlerini belirlemiştir. Bu dalga boyları mavi = 435,8 nm, yeşil = 546,1 nm ve kırmızı = 700 nm'dir. Standartlaştırma amacıyla üç spesifik birincil renk dalga boyu tanımlanmasının, RGB bileşenlerinin tüm spektrum renklerini üretebileceği anlamına gelmemektedir. Birincil renkler ışığın ikincil renklerini (macenta (kırmızı + mavi), cyan (yeşil + mavi) ve sarı (kırmızı + yeşil) üretmek için birlikte eklenebilir.

Ton, ışık dalgalarının bir karışımındaki baskın dalga boyu ile ilişkili bir özelliktir. Renk tonu, bir gözlemci tarafından algılanan baskın rengi temsil eder. Bu nedenle, bir nesneyi kırmızı, turuncu veya sarı olarak adlandırdığımızda, tonuna atıfta bulunuyoruz. Doygunluk, bir renk tonu ile karıştırılan nispi saflık veya beyaz ışık miktarını ifade eder. Saf spektrum renkleri tamamen doygundur. Pembe (kırmızı ve beyaz) ve lavanta (mor ve beyaz) gibi renkler daha az doygun olur, doygunluk derecesi eklenen beyaz ışık miktarı ile ters orantılıdır. Birlikte alınan renk tonu ve doygunluğa renklilik denir ve bu nedenle bir renk, parlaklığı ve renkliliği ile karakterize edilebilir. Belirli bir rengi oluşturmak için gereken kırmızı, yeşil ve mavi miktarlarına tristimulus değerleri denir ve sırasıyla  $X$ ,  $Y$  ve  $Z$  olarak gösterilir. Daha sonra bir renk, trikromatik katsayıları ile belirtilir.

$$x = \frac{X}{X + Y + Z} \quad (4.3)$$

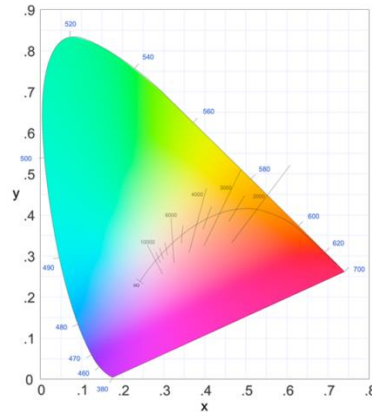
$$y = \frac{Y}{X + Y + Z} \quad (4.4)$$

$$z = \frac{Z}{X + Y + Z} \quad (4.5)$$

Denklem 4.3, 4.4 ve 4.5' den denklem 4.6 elde edilir

$$x + y + z = 1 \quad (4.6)$$

Görünür spektrumdaki herhangi bir ışık dalga boyu için, o dalga boyuna karşılık gelen rengi üretmek için gerekli tristimulus değerleri doğrudan kapsamlı deneysel sonuçlardan derlenmiş eğrilerden veya tablolardan elde edilebilir. Renkleri belirtmek için başka bir yaklaşım, renk kompozisyonunu  $x$  (kırmızı) ve  $y$  (yeşil) fonksiyonu olarak gösteren CIE renklilik diyagramını kullanmaktır. Şekil 4.3' de gösterilen eşit enerji noktası, üç ana rengin eşit bölüntüsüne karşılık gelir; beyaz ışık için CIE standardını temsil eder. Renklilik çizelgesinin sınırında bulunan herhangi bir nokta tamamen doymuştur. Bir nokta sınırı terk edip eşit enerji noktasına yaklaştıkça, renge daha fazla beyaz ışık eklenir ve daha az doymuş hale gelir. Eşit enerji noktasındaki doymuluk sıfırdır.



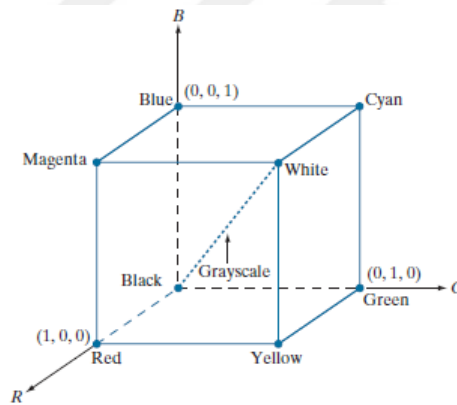
Şekil 4.3 CIE Renk Modeli

## 4.2. Renk Modelleri

Bir renk modelinin amacı, renklerin standart bir şekilde belirlenmesini kolaylaştırmaktır. Renk modeli bir koordinat sisteminin ve bu sistem içindeki bir alt uzayın ayırt edici özelliklerini belirtmektedir. Dijital görüntü işleme açısından, uygulamada en sık kullanılan donanım odaklı renk modelleri;

- Renkli monitörler için RGB model tercih edilmektedir
- Renkli baskı için CMY ve CMYK modelleri tercih edilmektedir
- İnsanların rengi tanımlama ve yorumlama şekline çok yakın olan HSV modeli tercih edilmektedir.

Ana diyagonal boyunca noktalar, başlangıç noktasından siyahtan noktaya (1, 1, 1) beyaza kadar gri değerlere sahip olan RGB renk küpü şeması Şekil 4.4' de gösterilmiştir[11].



Şekil 4.4.10 RGB renk küpü şeması

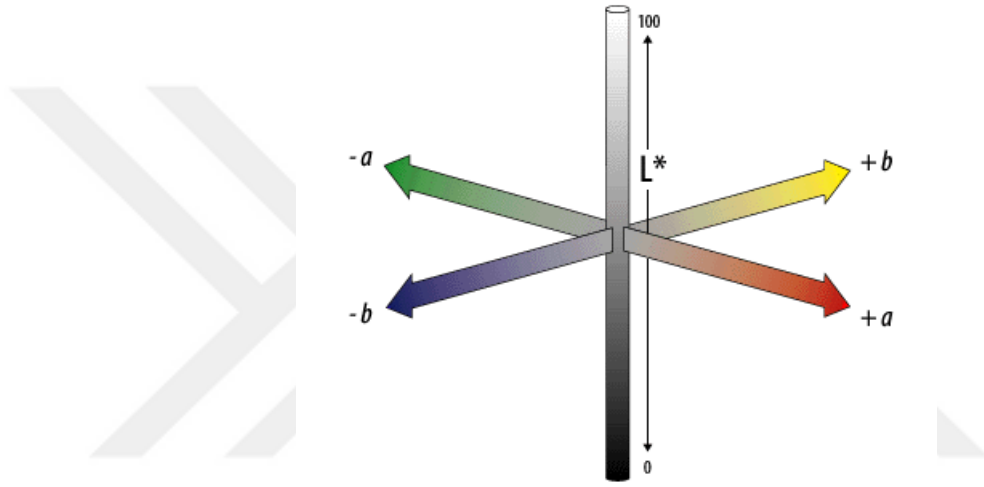
### 4.2.1. Munsell Renk Sistemi

Munsell renk sistemi 1930 yıllarında Amerika Birleşik Devletleri Tarım Bakanlığın' da çalışan Albert H. Munsell tarafından toprak araştırmaları esnasında oluşturulmuş bir renk sistemidir. Üç ana özelliği olan hue, value ve chroma ile tanımlanmaktadır. Hue, birincil renkler olan kırmızı, mavi, yeşil gibi renkleri ayırt etmemize olanak sağlamaktadır. Value tespit edilen rengin açıklık veya koyuluk oranlarını tespit etmemizi sağlamaktadır. Chroma ise renk doygunluğunu belirlemektedir. Munsell renk sistemi silindirik koordinat üzerinde hue, value ve chroma özelliklerine göre gösterimi yapılmaktadır[3]. Aynı zamanda hue ana renk, value parlaklık ve chroma yoğunluk olarak adlandırılmaktadır.



#### 4.2.2. CIELab Renk Sistemi

CIELab renk sistemi küresel koordinat sisteminde üç boyutlu ifade edilen renk sistemidir ve üç ana bileşenden oluşmaktadır. Bu bileşenler L, a, ve b ile adlandırılmaktadır. L koordinatı Munsell renk sistemindeki value değerini yani parlaklık değerini temsil eder. Bu değer sıfır ise renk siyah yüz ise renk beyazdır. A koordinatının pozitif kanadı kırmızı, negatif kanadı ise yeşil rengin doygunluğunu ifade etmektedir. B koordinatının pozitif kanadı sarı, negatif kanadı ise mavi rengin doygunluğunu ifade etmektedir.



Şekil 4.6 CIELab renk sistemi

Şekil 15' de gösterilen küresel koordinat sisteminde bileşenlerin kesiştiği nokta bir rengi tanımlanmaktadır. İki renk arasındaki farklılık denklem 4.6 'de gösterildiği gibi hesaplanmaktadır[3].

$$\Delta E_{1-2} = [(\Delta L)^2 + (\Delta a)^2 + (\Delta b)^2]^{1/2} \quad (4.7)$$

#### 4.2.3. RGB Renk Uzayı

Kartezyen koordinat sistemine dayanan RGB renk uzayında her renk kırmızı, yeşil ve mavi olan birincil spektral bileşenlerinde bulunmaktadır. RGB renk alt uzayı, RGB birincil değerlerinin üç köşede olduğu, Şekil 4.7' da gösterilen küptür, ikincil renkler cyan, macenta ve sarı diğer üç köşededir. Siyah renk orijindedir ve beyaz renk orijinden

en uzak köşededir[27]. RGB renk uzayında, gri tonlama (eşit RGB değerlerine sahip noktalar), bu iki noktayı birleştiren çizgi boyunca siyahtan beyaza uzanır. RGB renk uzayında bulunan farklı renkler, küpün üzerindeki veya içinde bulunan noktalardır. Aynı zamanda orijinden uzanan vektörlerle tanımlanır. Bu renk uzayındaki tüm R, G ve B değerlerinin  $[0, 1]$  aralığında olduğu varsayılmaktadır.

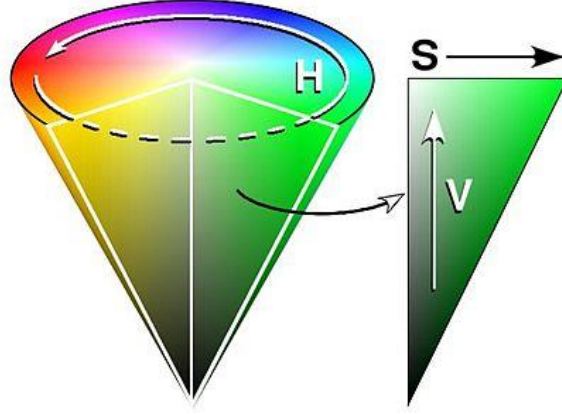


Şekil 4.7 RGB Renk Uzayı

RGB renk uzayı bilgisayar grafiklerinde en çok kullanılan renk sistemidir fakat görüntü işleme için açısından etkili bir yol değildir. Çünkü bir pikselin renk yoğunluğu değiştirilmek istendiğinde o pikselin bellekteki RGB değerleri ayrı ayrı okunur ve istenilen yoğunluğa göre işlem yapıldıktan sonra bellekteki yerine RGB değerleri ayrı ayrı yazılmalıdır. Bu yüzden farklı renk uzayları kullanarak bunun gibi parlaklık ve yoğunluk değiştirme işlemleri daha hızlı ve kolay yapılabilmektedir.

#### 4.2.3. HSV Renk Uzayı

HSV renk uzayı Alvy Ray Smith tarafından 1978 yılında RGB uzayına göre insan gözü düzeneğine daha yakın yapı oluşturulmak için tanımlanan renk sistemidir. Bu tezde kullanılan veriler için daha uygun olan bu renk sistemi hue, satürasyon ve value nitelikleri ile tanımlanmaktadır[6]. Hue değeri rengi belirler, satürasyon değeri doygunluğu temsil eder ve value değeri ise parlaklığı temsil etmektedir. HSV renk modelinde her renk tonunun renkleri, siyah ile beyaz arasında değişen nötr renklerin merkezi ekseni etrafında Şekil 4.8' de de görüldüğü gibi radyal bir dilim halinde düzenlenir[30].



Şekil 4.811 HSV Renk uzayı

#### 4.2.4. RGB Görüntüyü HSV görüntüye dönüştürme

R, G, B değerleri 0 ile 1 arasında ve H değeri 0 ile 360 arasında olan RGB ve HSV renk uzaylarında dönüşüm Şekil 4.9' da gösterilmiştir[6].

$$\text{MAX} = \max\{R, G, B\}, \text{MIN} = \min\{R, G, B\}$$

$$H = \begin{cases} \text{Tanımsız,} & \text{eğer MAX} = \text{MIN} \\ 60 \frac{G-B}{\text{MAX}-\text{MIN}} + 0, & \text{eğer MAX} = R \text{ ve } G \geq B \\ 60 \frac{G-B}{\text{MAX}-\text{MIN}} + 360, & \text{eğer MAX} = R \text{ ve } G < B \\ 60 \frac{B-R}{\text{MAX}-\text{MIN}} + 120, & \text{eğer MAX} = G \\ 60 \frac{R-G}{\text{MAX}-\text{MIN}} + 240, & \text{eğer MAX} = B \end{cases} \quad (4.9)$$

$$S = \begin{cases} 0, & \text{eğer MAX} = 0 \\ 1 - \frac{\text{MIN}}{\text{MAX}}, & \text{değilse} \end{cases}$$

$$V = \text{MAX}$$

Şekil 4.9 RGB-HSV dönüşüm formülleri

### 4.3. HSV Görüntüyü RGB görüntüye dönüştürme

R, G, B değerleri 0 ile 1 arasında ve H değeri 0 ile 360 arasında olan RGB ve HSV renk uzaylarında dönüşüm Şekil 4.10' da gösterilmiştir[6].

$$H_i = \left[ \frac{H}{60} \right] \text{mod } 6$$

$$f = \frac{H}{60} - H_i$$

$$p = V(1 - S)$$

$$q = V(1 - fS)$$

$$t = V(1 - (1 - f)S)$$

$$\text{eğer } H_i = 0 \rightarrow R = V, G = t, B = p \quad (4.10)$$

$$\text{eğer } H_i = 1 \rightarrow R = q, G = V, B = p$$

$$\text{eğer } H_i = 2 \rightarrow R = p, G = V, B = t$$

$$\text{eğer } H_i = 3 \rightarrow R = p, G = q, B = V$$

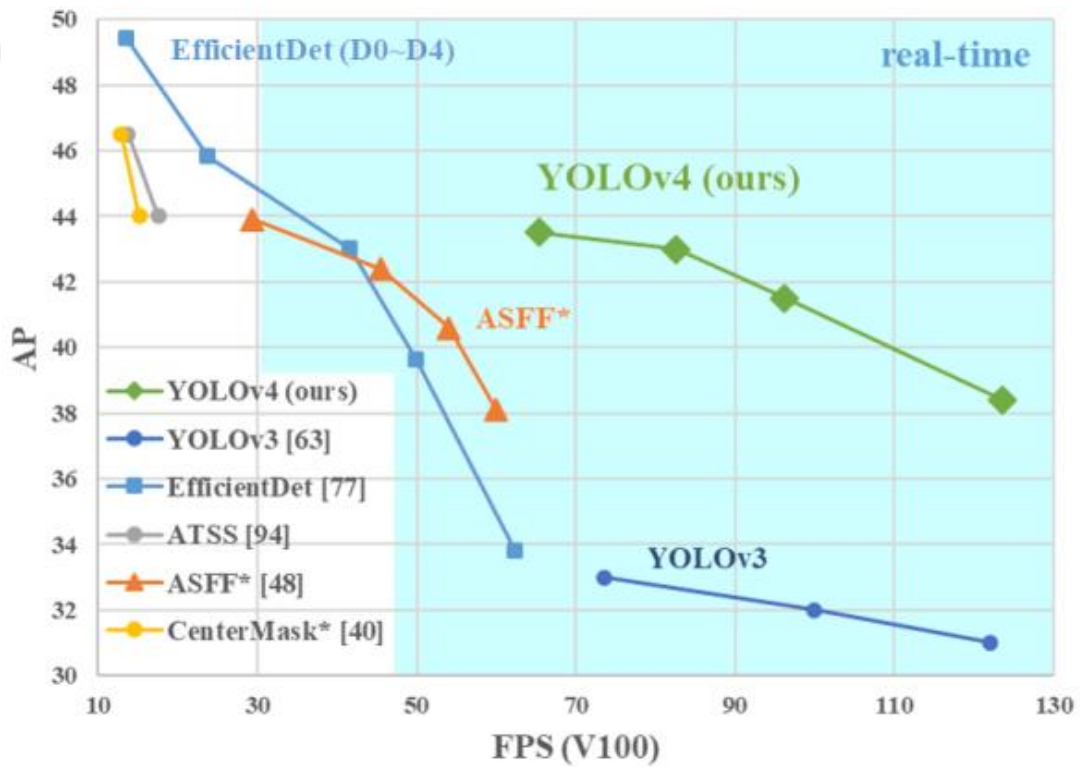
$$\text{eğer } H_i = 4 \rightarrow R = t, G = p, B = V$$

$$\text{eğer } H_i = 5 \rightarrow R = V, G = p, B = q$$

Şekil 4.10 HSV-RGB Dönüşüm Formülleri

### 4.3. YOLO Algoritması

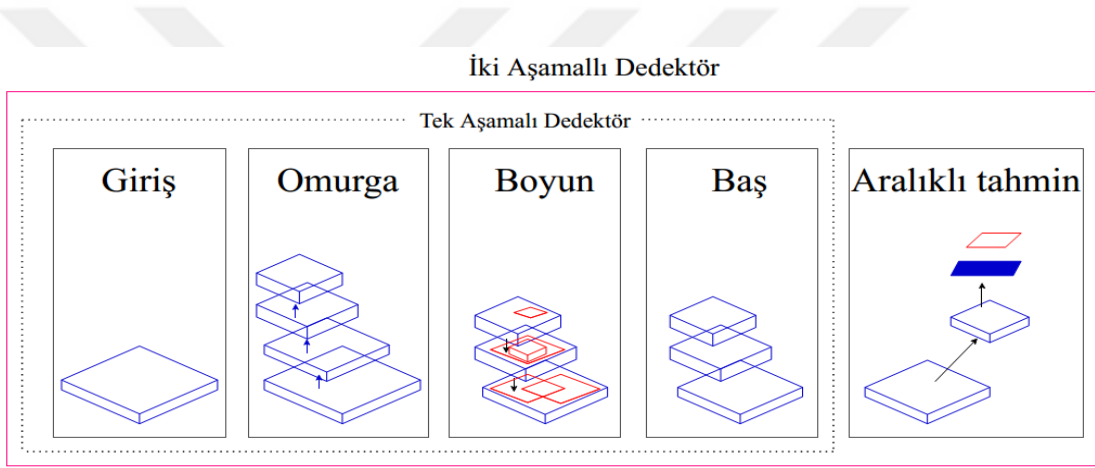
YOLO (You Only Look Once) algoritması sadece bir kere bak mantığıyla Joseph Redmon tarafından düşük seviyeli diller ile yazılmış bir derin öğrenme algoritmasıdır. Yıllar içerisinde birden fazla versiyonu çıkarılarak algoritmaya verilen görüntülerde gerçek zamanlı olarak obje tespitini hızlı ve doğru biçimde yapılması hedeflenmiştir. Bunu da görüntünün tamamını tek seferde sinir ağından geçirerek yapmaktadır. YOLO tek bir sinir ağı kullanarak nesnelerin hem sınıflandırılmasını hem de tahminini gerçekleştirmek için multi-step işlemini birleştirir. Bunun sonucunda nesne algılama performansı büyük ölçüde optimize edilmiş olur[31].



Şekil 4.11 YOLOv4' ün diğer nesne tespit algoritmalarıyla MS COCO veri seti üzerindeki performans karşılaştırılma grafiği

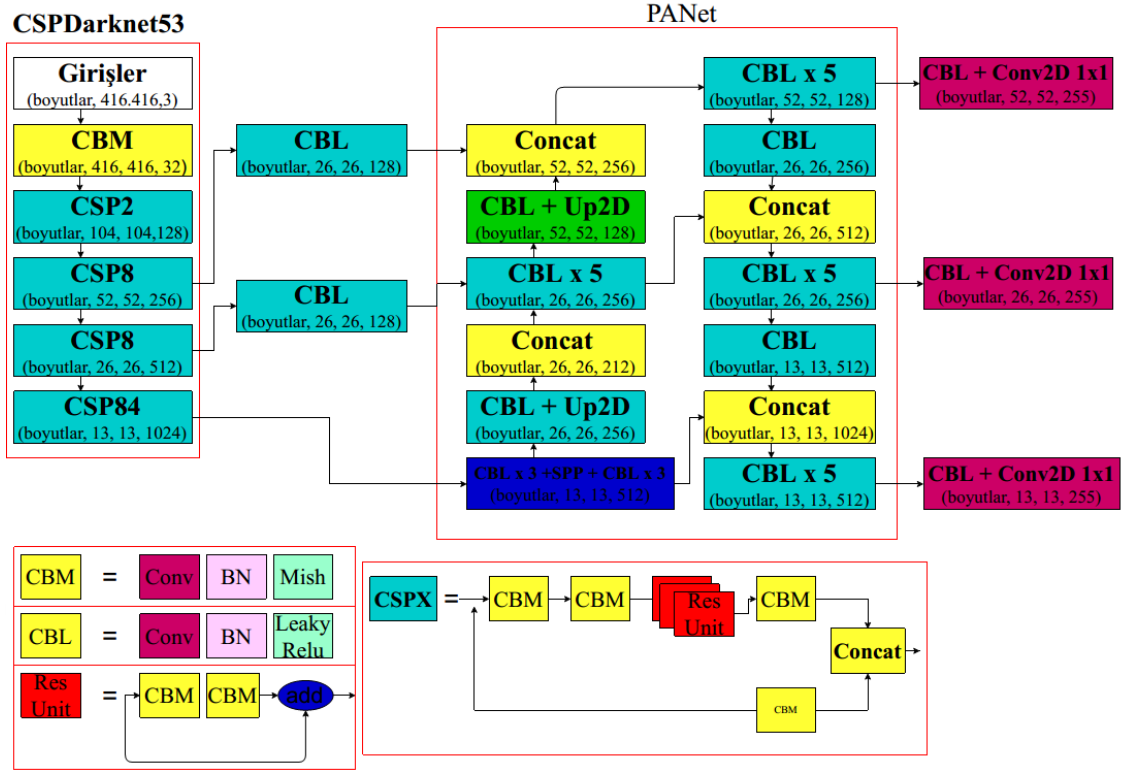
Genel olarak YOLO algoritmasına verilen görüntü istenilen boyutlarda ızgaralara bölünür ve her ızgara kendi içerisinde nesne olup olmadığını tahmin etmeye çalışır. Tahmin işlemini yerine getirirken her ızgara için güvenlik puanlaması oluşturur. En düşük güven düzeyi "0.0", en yüksek güven düzeyi ise "1.0" olarak belirlenmiştir. Izgara içerisinde nesne olduğu düşünülüyorsa merkez noktasının kendi içinde olup olmadığı tahmin edilmeye çalışılır. Nesnenin merkez noktasına sahip olduğuna karar veren ızgara

o nesnenin sınıfını, yüksekliğini ve genişliğini bulduktan sonra tahmin edilen nesnenin etrafında sınırlayıcı kutu çizer. Burada birden fazla sınırlayıcı kutu çizimi oluşmuş olabilir fakat daha önce tespit edilen güvenlik düzeyi puanlarına göre Non-max Suppression algoritması kullanılarak sadece güvenlik düzeyi en yüksek sınırlayıcı kutu görüntü üzerinde gösterilebilir. YOLOv4 algoritması diğer YOLO algoritmalarından farklı olarak 2 aşamada gerçekleştirilmektedir ve Şekil 4.12’de bu aşamalar gösterilmektedir. Birinci aşamada giriş görüntüsü, omurga, boyun ve baş bulunurken ikinci aşamada tahmin olayı gerçekleştirilmektedir. Şekil 4.11’ de görüldüğü gibi YOLOv4 algoritması diğer algoritmalara göre yüksek FPS ve performans sağladığı gösterilmektedir. YOLOv4, YOLOv3 e göre %10 daha fazla performansa %12 daha yüksek FPS de sonuç verir[13].



Şekil 4.12 YOLOv4 iki aşamalı yapısı

YOLOv4 algoritması ile nesne tespitinde optimum hız ve nesne algılama doğruluğu hedeflenmiştir. Bu algortmada özellik çıkarma görevi (omurga) CSPDarknet53, özellik toplayıcı olarak (boyun) olarak SPP ve PANet ve nesne detektörü (baş) olarak YOLOv3 kullanılmaktadır. Hızlı nesne tespiti yaparken doğruluğu arttırmak için BoF (Bag of Freebies) ve BoS (Bag of Specials) adı verilen araç seti kullanılmıştır. Şekil 4.13’de YOLOv4 algoritmasının ağ mimarisi gösterilmiştir.



Şekil 13 YOLOv4 Ağ Mimarisi

BoF doğruluğu arttırmak eğitim sürecinde veri arttırma, sınıf dengesizliği, yumuşak etiketleme gibi işlemlerle iyileştirmeler yapmaktadır. BoS ise performansı arttırmak için alan arttırılması, dikkat kullanımı, FPN gibi özellik entegrasyonunu ve maksimum olmayan bastırma gibi işlemler gerçekleştirir. Nesne tespitinde doğruluğu arttırmak performansı arttırmak değildir bu yüzden BoF ve BoS farklı iki olay ile ilgilenmektedir.

Omurga kısmında temel görevi özellik çıkartma görevi olan CSPDarknet53 derin öğrenme sınıflandırıcı kullanılmıştır. Boyun kısmında PANet ve SPP kullanılmıştır ve boyun kısmı BOS' un alt kümesidir. Burada farklı aşamalardan özellik haritaları toplanır yani boyun kısmı bir nevi özellik toplayıcıdır. Kafa kısmında YOLOv3 kullanılır ve nesnenin nerede bulunabileceğini tahmin eder ama bunu diğer bölümlerle paylaşmaz yani bu kısım bir nevi nesne detektörü görevi üstlenmektedir. Anchor-based ve anchor-free detektörlere ayrılan iki aşamalı detektörler ve bir aşamalı detektörler kullanılmaktadır. YOLOv4 'ün omurgayı oluşturan kısımlarında farklı tercihler yapılabilir yapılan bu tercihler ağ mimarisini değiştirerek daha farklı sonuçlar elde etmemize neden olabilmektedir.

YOLOv4 algoritmasında giriş görüntüsü  $416 * 416$  boyutun da ve 3 kanallı olmak zorundadır. Bu kural ışığında her görüntünün etiketleme işlemi yapılmalı ve bazı parametreler ayarlanarak algoritmaya sunulmalıdır.

## 4.2. Dijital Görüntü İşleme Teknikleri

Dijital görüntü işleme yöntemleri genel olarak insan yorumu için resimsel bilgilerin geliştirilmesi, resimsel bilgilerin depolanması, iletilmesi ve çıkarılması gibi görevler için kullanılmaktadır. Görüntü işleme, görüntüyü manipüle etme işlemidir, böylece sonuç belirli bir uygulama için orijinalden daha uygun olarak hazırlanır. Dijital görüntü işleme alanı bir matematik temeli üzerine inşa edilmiş olsa da insan sezgisi ve analizi genellikle bir tekniğin diğerine karşı seçiminde rol oynar ve bu seçim genellikle özne ve görsel yargılara dayanarak yapılır.

Sıradan bir fotoğraf kamerasında lensin sabit odak uzaklığı vardır. Çeşitli mesafelere odaklanma, filmin bulunduğu mercekle görüntüleme düzlemi arasındaki mesafeyi değiştirerek sağlanır. İnsan gözünün çalışma prensibinin tersi şeklinde çalışır.

Bir görüntü iki boyutlu bir işlev olarak  $f(x, y)$  şeklinde denklem 4.9' da ki gibi tanımlanmaktadır. Burada  $x$  ve  $y$  düzlemsel koordinatlarıdır ve herhangi bir  $f$  koordinat çiftindeki  $x$ ,  $y$  genliği, yoğunluğu veya o noktada görüntünün gri düzeyidir. Bir görüntünün dijital görüntü olabilmesi için  $x$ ,  $y$  ve  $f$ 'n yoğunluk değerlerinin tümü sonlu ve ayrık olması gerekmektedir[30]. Dijital görüntü işleme alanı, dijital görüntülerin bir dijital bilgisayar vasıtasıyla işlenmesini ifade eder.

Dijital görüntüler elde etmek iki aşamalı sistem olup bunlardan birincisi görüntülemek istediğimiz nesnenin yaydığı enerjiye cevap veren fiziksel bir sensör ikincisi ise sayısallaştırıcı olarak adlandırılan sensörün çıktısını dijital forma dönüştüren bir cihazdır[30]. Özelleştirilmiş görüntü işleme donanımı genellikle sayısallaştırıcıdan ve tüm görüntülerde paralel olarak aritmetik ve mantıksal işlemler gerçekleştiren aritmetik mantık birimi (ALU) gibi diğer ilkel işlemleri gerçekleştiren donanımdan oluşur. ALU, bir bilgisayarın işleyemediği hızlı veri çıkışları gerektiren işlemleri gerçekleştirir. Çok fazla matris işlemleri gerektiren görüntü işleme sistemlerinden birden fazla GPU kullanımı yaygındır.



Dijital bir görüntünün, her biri belirli bir konuma ve değere sahip olan sınırlı sayıda öğeden oluşur. Bu öğelere resim öğeleri, görüntü öğeleri, pels ve piksel denir. Piksel, dijital görüntünün öğelerini göstermek için en fazla kullanılan terimdir. Her piksel 8 bit yani 1 byte uzunluğunda olup 512\*512 piksel büyüklüğünde kaydedilecek bir görüntü dosyası için toplamda 512 kilobayt bellek alanı gerekmektedir. Bu yüzden birden fazla görüntü için yeterli bellek sağlamak gerekmektedir. Bu alanı sağlama için görüntü işleme uygulamaların kısa süreli bellek, çevrimiçi bellek ve arşivsel bellek kullanılmaktadır. İşlem sırasında ihtiyaç duyulan kısa süreli bellek iki yöntem ile yapılmakta olup bunlardan birincisi rastgele erişimli bellek (RAM) kullanılarak ikincisi ise frame buffer olarak bilinen özel devreler ile yapılmaktadır. Bu yöntemler ile bir ya da birden fazla görüntüye saniyede 30 kare kullanılarak ulaşılabilir. Çevrimiçi bellekler CD ve DVD gibi manyetik veya optik medya diskleridir. Sık erişim için çevrimiçi bellek, az erişim için arşivsel bellek kullanılmaktadır. Görüntüleri kaydetmek için basılı kopya cihazları arasında lazer yazıcılar, film kameraları, ısıya duyarlı cihazlar, mürekkep püskürtmeli birimler ve optik ve CD-ROM diskler, ağ ve bulut iletişimi gibi dijital birimler bulunur.

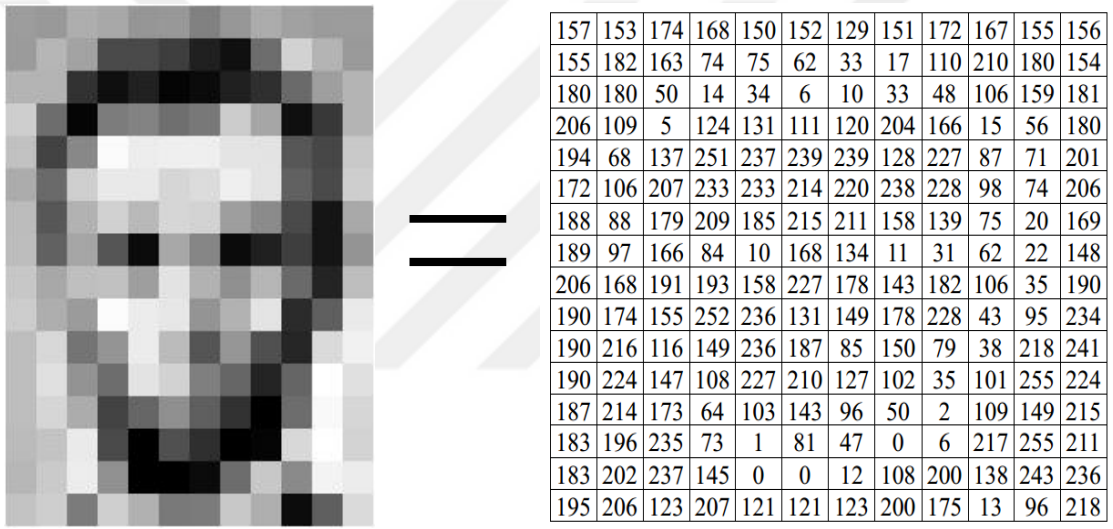
Sürekli görüntüyü M satırları ve N sütunları içeren  $f(x, y)$  dijital görüntüsünde örneklersek  $x, y$  ayrık koordinatlardır. Denklem 4.9' da başlangıçtaki dijital görüntü  $f(0,0)$  ve ilk satırdaki sonraki koordinatlardaki değeri  $f(0,1)$ ' dir[30]. Buradaki  $(0, 1)$  notasyonu, ikinci örneği ilk sıra boyunca göstermek için kullanılmaktadır. Bu değerler görüntü örneklendiğinde fiziksel koordinatların değerleri olduğu anlamına gelmez.

$$f(x, y) = \begin{matrix} f(0,0) & f(0,1) \dots & f(0, N - 1) \\ \vdots & \vdots & \vdots \\ f(M - 1,0) & f(M - 1,1) \dots & f(M - 1, N - 1) \end{matrix} \quad (4.9)$$

Bu 4.9 denkleminin sağ tarafı, gerçek sayılar dizisi olarak temsil edilen dijital bir görüntüdür ve piksel olarak ifade edilir. Piksel dizinin sabit bir koordinat çiftindeki değerleridir ve Şekil 23'de gösterildiği gibi bir piksel 0 ile 255 arasında değişen sayı değerlerini alabilir. Dijital bir görüntüyü geleneksel matris biçiminde de temsil denklem 4.10' da gösterilmiştir[30].

$$A = \begin{bmatrix} a_{0,0} & \cdots & a_{0,N-1} \\ \vdots & \ddots & \vdots \\ a_{M-1,0} & \cdots & a_{M-1,N-1} \end{bmatrix} \quad (4.10)$$

Bir görüntüleme sisteminin değer aralığını, ölçülebilir maksimum yoğunluğun sistemdeki saptanabilir minimum yoğunluk seviyesine oranı olarak tanımlanır. Değer aralığı bir görüntünün sahip olabileceği en düşük ve en yüksek yoğunluk seviyelerini belirler. Bir görüntüdeki en yüksek ve en düşük yoğunluk seviyeleri arasındaki yoğunluk farkı kontrast olarak ifade edilir. Şekil 4.14’ de 16\*12 boyutundaki bir görüntünün her piksel için piksel değeri gösterilmiştir[30]. Ayrıca bu piksel değerlerinin yan yana gelmesiyle anlamlı görsel veri oluşumunu göstermektedir.



Şekil 4.14 Bir görüntü üzerindeki piksel gösterimi

Dijital bir görüntüyü saklamak için gereken bit sayısı denklem 4.11 gösterilmiştir[31].

$$b = M \times N \times k \quad (4.11)$$

Satır sayısı sütun sayısına eşit ise bit sayısı denklem 4.12 ‘deki gibi hesaplanabilir[30].

$$b = N^2 \times k \quad (4.12)$$

Renkli görüntüleri sayısal sistemlerde ifade etmek için birden fazla renk uzayları kullanılmaktadır. Genel olarak kullanılan RGB renk uzayın her pikselde üç ana renkten hangi miktarda barındığını gösterir. Bu sistemde de oluşturulan matris 3 boyutlu ve her bir ana renk 8 bit uzunluğunda olup toplamda 24 bit uzunluğundadır. Bir görüntü üzerindeki toplam nokta sayısıdır yani görüntüdeki satır ve sütun sayısı 512 ise görüntünün çözünürlüğü 512\*512 olarak kullanılır. Yüksek çözünürlüklü görüntüler sıkıştırılmadan depolanırsa çok fazla yer kaplamaktadır. Bu gibi durumlarda JPG, PNG gibi sıkıştırma formatları kullanılmaktadır[30].

#### 4.2.1. Görüntü pekiştirme

Bilgisayara veya insana yönelik olmak üzere uygulanan görüntü pekiştirme algoritmaları spesifik bir uygulama alanı için görüntülerin daha uygun ve anlaşılır hale getirmesini sağlamaktadır. Görüntü pekiştirme uygulamaları nokta ve maske işleme teknikleri olarak iki kategoride incelenmektedir. Görüntü düzlemi olarak adlandırılan teknik bir görüntü üzerindeki piksel kümesine verilen addır ve bu teknik pikseller üzerinde işlem yapmaktadır.

Temel olarak uygulanan piksel işlemleri denklem 4.13 eşitliğinde gösterilmiştir[30].

$$g(x, y) = T[f(x, y)] \quad (4.13)$$

Görünü düzlemi üzerinde yapılan temel işlemler denklem 4.13'de gösterilmiştir. Bu denkleme göre  $f(x, y)$  bir giriş görüntüdür. Denklemin diğer tarafında bulunan T ise f üzerinde  $(x, y)$  koordinatlarının komşuluğunda tanımlı operatördür. Görüntü üzerinde merkezi  $x, y$  olarak tanımlanan bir dikdörtgen belirlenir bu dikdörtgenin boyutları kişiye bağlı olup 1x1, 2x2, ... gibi değerler alabilmektedir. Dikdörtgen boyutu 1x1 olarak tanımlanan bir işleminde g'nin alacağı değeri f in  $(x,y)$  noktasındaki değerine göre belirlenir ve belirlenen bu değer ile nokta işleme tekniği kullanılmaktadır. Eğer dikdörtgen boyutuna 5 \* 5, 7 \* 7, 3 \* 5 gibi değerler verilirse filtre veya maske olarak adlandırılır. Filtre kullanılan uygulama sonucunda g' nin alacağı değer filtre matrisinin katsayılarına bağlı ve uygulamaya maske işleme tekniği denmektedir. Bu işlemler ile görüntü üzerinde parlaklık ayarı, kontrast ayarı, eşikleme gibi birçok işlem yapılmaktadır.

## 4.2.2 Görüntü filtreleme

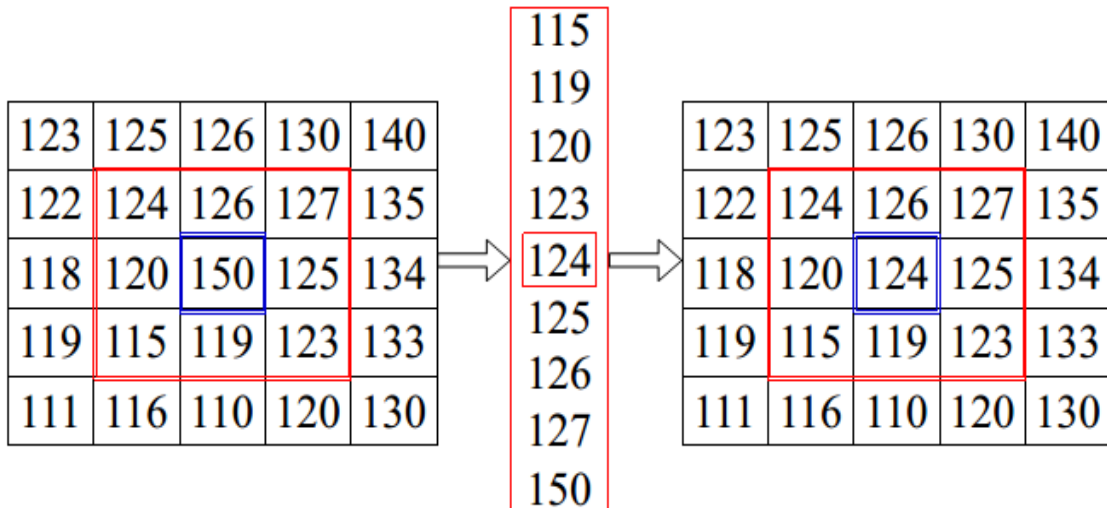
Temel olarak görüntü filtreleme uygulamaları giriş görüntü üzerinde dolaştırılan katsayılı bir filtre matrisinin giriş görüntüsü üzerindeki pikselleri farklı tekniklerle yeniden hesaplayıp giriş görüntüsüne benzer bir görüntü çıkartmasıdır[30]. Bu tekniklerin genel amacı görüntü üzerinde yumuşatma keskinleştirme, kenar saptama gibi efektler uygulayıp giriş görüntü üzerinde belirli nesne özelliklerini ön plana çıkartmaktır.

$$f(x,y) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j) \times f(x-i,y-j) \quad (4.11)$$

Bu filtrelerin yaptıkları işlere ve hesaplamalarına bağlı olarak birçok türü bulunmaktadır.

### 4.2.2.1. Medyan filtresi

Görüntü üzerinde gezdirilen filtrenin boyuna göre filtre içerisinde bulunan piksellerin değerleri sıralanır ve ortanca değer merkez piksele yazılır[15].



Şekil 4.1612 Medyan filtre uygulaması

Şekil 4.16' da bir giriş verisine 3\*3 boyutundaki filtre ile yapılan medyan filtreleme uygulaması gösterilmiştir[30].

$$f(x, y) = \text{median}_{(r,c) \in S_{x,y}} \{g(r, c)\} \quad (4.14)$$

Denklem 4.14'de medyan filtresinin matematiksel hesabı gösterilmiştir.

#### 4.2.2.2. Maks ve Min filtreleri

Bir giriş görüntüsü üzerindeki en parlak noktaları bulmaya veya parlak alanlar ile yan yana olarak karanlık bölgeleri aşındırmak için kullanılan Maks filtresinin matematiksel hesabı denklem 4.15'te verilmiştir[30].

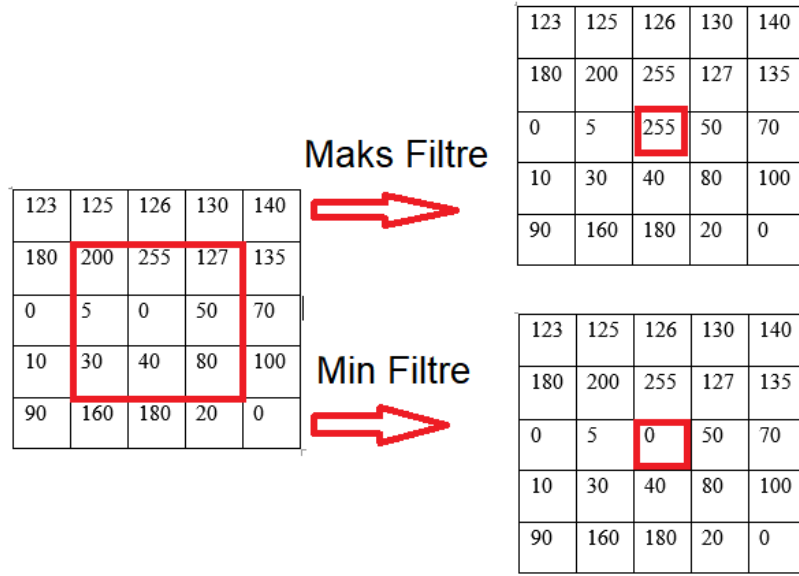
$$f(x, y) = \max_{(r,c) \in S_{x,y}} \{g(r, c)\} \quad (4.15)$$

Bu denkleme göre filtre içinde bulunan piksel değerlerinden en yüksek değere sahip olan piksel değeri merkez piksel değeri olarak atanır.

Bir giriş görüntüsü üzerindeki en karalık noktaları veya karalık alanlara ile yan yana olan parlak bölgeleri aşındırmak için kullanılan filtreye Min filtresi denmektedir.

$$f(x, y) = \min_{(r,c) \in S_{x,y}} \{g(r, c)\} \quad (4.16)$$

Denklem 4.16 gösterilen matematiksel ifadede bir görüntü üzerinde dolaştırılan filtre içinde bulunan piksel değerlerinden en küçük değere sahip piksel değeri merkez piksel değeri olarak atanır[30]. Şekil 4.17' de maks ve min filtre ile ilgili görsel işlem verilmiştir.



Şekil 4.17 Maks ve Min filtre örneği

#### 4.2.2.3 Ortalama filtresi

Mean filtresi olarak da bilinen bu filtre giriş görüntüsü üzerindeki pikselin komşu pikseller ile arasındaki değer değişim miktarını minimuma indirerek görüntüyü yumuşatır ve görüntü üzerindeki gürültü azaltmaktadır. Ortalama filtresi, bir görüntünün her bir piksel değerini komşularının ve kendisinin dahil olduğu ortalama değer ile değiştirmektedir. Bu durum, çevresindekileri temsil etmeyen piksel değerlerinin ortadan kalkmasını sağlar. Ortalama filtresi bir çekirdek şablon (kernel) temeline dayanan konvolüsyon filtresidir. Genellikle  $3 \times 3$  kare çekirdek şablon ancak büyük görüntülerde ve bazı yumuşatma işlemlerinde daha büyük şablonlar kullanılabilir. Şekil 4.18’de kernel katsayısı  $1/9$  olan  $3 \times 3$ ’lük filtrenin bir görüntüye uygulanması ve sonucu gösterilmektedir[30].

Şekil 4.18  $3 \times 3$  Ortalama filtresi

#### 4.2.2.4. Gauss filtresi

Gauss bir giriş görüntüsü üzerinde bulunan gürültüyü ortandan kaldırmak ve görüntüyü bulanıklaştırmak için kullanılan filtreleme yöntemidir. Bu filtre istenilen görüntüyü elde etmek için ilgili bölgenin her pikselinde dolaştırılan tek boyutlu simetrik bir kernel olarak uygulanır[30]. Kernel, dışardan çekirdeğin merkezine doğru olan piksellerden son değere doğru daha fazla ağırlığa sahip olması nedeniyle, şiddetli renk değişimine yani kenarlara karşı zorlanmaz.

$$f(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (4.17)$$

Denklem 4.17’de gauss filtresinin bir görüntü üzerine uygulandığı yapılan matematiksel işlemi göstermektedir. Burada  $x$  ve  $y$  görüntünün koordinat değerleridir. Görüntünün filtre boyutlarındaki kısmının standart sapması  $\sigma$  olarak belirtilmektedir.



Şekil 4.19 Gauss filtresi

Şekil 4.19’ da standart sapması 2 olan 3\*3’ lük kernel uygulaması gösterilmektedir[30].

### 4.2.3. Morfolojik işlemler

Biyoloji de hayvan ve bitkilerin şekil ve yapısı ile ilgilenen bir alan morfoloji görüntü işlemede is sınırlar, iskeletler ve dış bükey gövde gibi görüntü bileşenlerinin ayrıştırılması amacıyla kullanılan araçtır. Bu araç içerisinde morfolojik filtreleme, inceltme ve budama gibi birçok teknik analiz bulunmaktadır.

İki boyutlu sayı olan  $Z^2$  bir elemanı olarak tanımlanan görüntüler sadece siyah ve beyaz renk değerinden oluşan ve bir elemanı iki boyutlu vektör ile gösterilen ve koordinat sisteminde denklem 4.9 da ki gibi gösterilen  $f(x,y)$  olarak temsil edilen siyah piksellerden oluşan kümelerdir.  $Z^3$  'ün bir elemanı olarak temsil edilen gri ölçekli görüntülerde aynı şekilde tanımlanmaktadır[30]. Bu uzayın diğer uzaydan farkı her elemanın  $x$  ve  $y$  bileşeni koordinatları gösterirken diğer bir bileşeninde görüntünün gri seviyesini göstermesidir. Uzay derecesi arttıkça bileşen sayısı ve karmaşıklığı artmakta içerisinde renk ve zaman değişimi gibi ek özellikler barındırmaktadır.

Temelde birçok morfolojik yöntem aşındırma ve yayma işlemine dayanmaktadır. Açma ve kapa işlemi gibi morfolojik işlemler esasında görüntüye yapı elemanına yayma işlemi uyguladıktan sonra aşındırma işlemi uygulamaktadır. Bu ve bunun gibi morfolojik işlemler genel olarak yayma ve aşındırma işleminin sırasını büyüklüğü değiştirerek farklı varyasyonlar sonucu görüntü bileşenlerini ayrıştırmaktadır.

#### 4.2.3.1. Yayma İşlemi

Dilatasyon olarak da bilenen bu işlem ikili imgedeki nesneyi büyüten ya da kalınlaştıran morfolojik bir işlemdir. Bu işlemin nasıl yapılacağını yapı elemanı belirlemektedir.

Denklem 4.18' de  $A$  ve  $B$   $Z^2$  de tanımlı iki küme olması durumunda  $A$ 'nın  $B$ 'ye yayılmasını ifade etmektedir[30].

$$A \oplus B = \{z | (B')_z \cap A \neq \emptyset\} \quad (4.18)$$

Bu eşitlik  $B$  yansımasının  $z$  noktasına göre dönüştürme işlemine dayanmaktadır. Bu manada  $A$ 'nın  $B$ 'ye göre yayılması  $z$  noktasına göre yer değiştirme kümesi oluşturacağından 4.18. denklemi 4.19 denklemi şeklinde de ifade edilebilir[30].



$$A \oplus B = \{z | [(B')^z \cap A] \subseteq A\} \quad (4.19)$$

Burada  $B$  elemanı yapı elemanı olarak kullanılmıştır. Yayma işlemi yukarıdaki denklemlerden de görüldüğü gibi tek bir eşitlikten oluşmaz. Denklem 4.19' de  $B$  katlama maskesi olarak kullanılmıştır. Bu eşitlikler birbirine benzeseler bile katlama işlemi aritmetik bir işlemken yayma işlemi küme işlemidir[17].



Şekil 4.20 Yayma işlemi

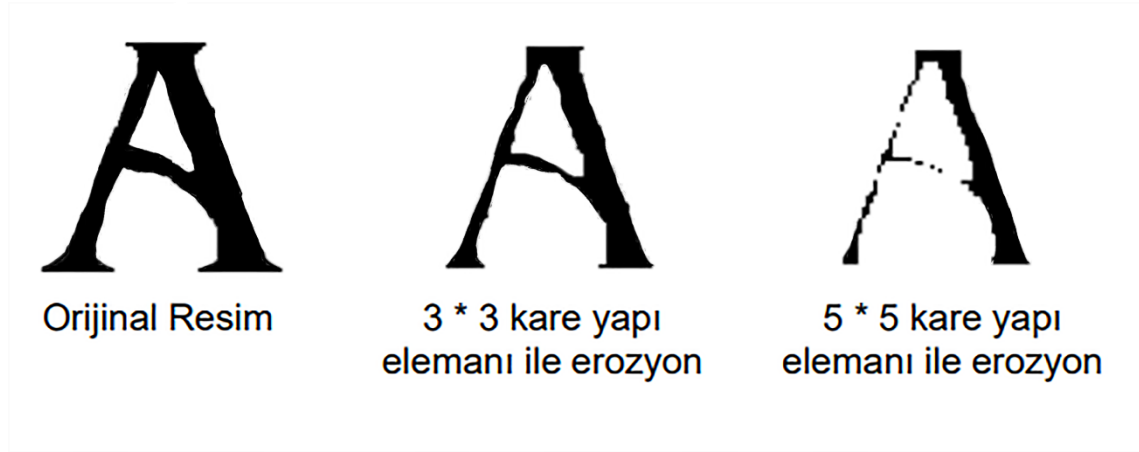
Şekil 4.20' de bir görüntünün farklı yapı elemanları ile yayma işlemini göstermektedir. Yapı taşı oranı büyüdükçe yayma işlemi de arttığı görülmektedir.

#### 4.2.3.2. Aşındırma İşlemi

Aşındırma işlemi genel olarak görüntüler üzerinde var olan gereksiz ve küçük detayları yok etmektedir. Erozyon olarak da nitelendirilen aşındırma işleminde  $A$  ve  $B Z^2$  de tanımlı ise  $A$ 'nın  $B$  ile aşındırılma işlemi denklem 4.20 de gösterilmiştir[30].

$$A \ominus B = \{z | (B)_z \subseteq A\} \quad (4.20)$$

Burada  $A'$  nın  $B$  ile aşındırılması ile  $B$ 'nin  $z$ 'ye göre dönüşüm kümesinin  $A$  tarafından kapsanan tüm  $z$  noktalarında oluşan kümesini kapsamaktadır. Bu eşitlik pratik olduğu için morfoloji uygulamalarında sık sık kullanılmaktadır[17].



Şekil 4.21 Aşındırma işlemi

Şekil 4.21' de aşındırma işlemi ile görüntü içindeki nesnenin inceltildiği görülmektedir. Bu inceltme işlemi yapı elemanın büyüklüğü ile değişmektedir. Kümesel tümlene ve yansıma açısından birbirinin aynısı olan yayma ve aşındırma işlemi matematiksel olarak denkliği denklem 4.21' de ifade edilmiştir[30].

$$(A \ominus B)^c = A^c \oplus B' \quad (4.21)$$

Bu eşitliği doğrulama adına aşındırma işlemi denklem 4.22 yeniden tanımlarsak:

$$(A \ominus B)^c = \{z | (B)_z \subseteq A\}^c \quad (4.22)$$

Eşitliğini keşfedebiliriz. Bu eşitlikte  $(B)_z$  kümesi  $A$  kümesi tarafından kapsandığından  $(B)_z \cap A^c = \emptyset$  olarak tanımlanacaktır ve bu tanımlamada içerisindeki  $z$ 'lerden oluşan kümenin tümleyeni  $(B)_z \cap A^c = \emptyset$  şartını sağlamalıdır[30].

$$(A \ominus B)^c = A^c \oplus B' = \{z|(B)_z \cap A^c = \emptyset\}^c \quad (4.23)$$

Denklem 4.23' de görüldüğü gibi eşitlik sağlanmış olur

#### 4.2.4. K-Means algoritması

K-means algoritması görüntü üzerinde bulunan renkleri anlamlı bölümlere ayırmak için kullanılan görüntü bölütleme algoritmasıdır. Uzamsal alan ve öznitelik uzay yöntemleri dahil olmaz üzeri uygulamanın amacına bağlı olarak çeşitli teknikleri bulunmaktadır. Özellik uzayı yöntemlerinden olan kümeleme tekniği pikselleri önceden tanımlanmış herhangi bir renk uzayında sınıflanmayı amaçlarken, uzamsal alan yöntemi olan bölgelerin büyütülmesi tekniği de pikselleri manipüle ederek bağlantılı alanları oluşturur[20]. Bu iki farklı teknik için de farklı algoritmalar geliştirilmiş olup bunlardan en yaygın olanı K-Means kümelemesidir.

Farklı renk skalalarında verimli çalışan bu algoritma RGB, YCbCr ve YUV gibi renk uzaylarında direk ham veri üzerinde yaygın olarak kullanılsa bilir insan algısına uygun değildir. CIE renk uzayında renk algılama biçimi tek olduğu için tercih edilmez[20]. İnsan algısına yakınlığıyla bilinen HSV renk uzayı bu algoritma ile daha iyi görüntü bölütleme sonuçları vereceği düşünülmüş ve kümeleme tekniği için geri ve renkli histogram kutularını birleştiren K-Means tekniği geliştirilmiştir[19]. Böylece K-Means algoritmasının işlem karmaşıklığı azaltılmış ve küme sayısı test edilmeden görüntü bölütleme sonuçlarına ulaşılmıştır.

Renkli bir görüntü üzerinde K-Means algoritmasını kullanabilmek için belirlemiş olduğumuz  $x_1, x_2, \dots, x_n$  renklerinden oluşan X kümesinin k adet alt kümeyle bölünmesi gerekmektedir. Bu renklerinin her birinin renk merkezine olan uzaklığı hesaplanıp k adet renk sınıf merkezi  $m_1(1), m_2(2), \dots, m_k(1)$  değerlerine rastgele olarak atanır. İki farklı renk olan **a** ve **b** renklerinin üç boyutlu renk uzayında birbirine olan uzaklığı 4.24 denkleminde gösterilmiştir[19].

$$d(a, b) = \| a, b \| = \sqrt{\sum_{i=1}^3 (a_i - b_i)^2} \quad (4.24)$$

Renk gruplarının  $k$  adet renk topaklarına dağılımı her pikselin kendisine en yakın olan renk topağına dâhil edilmesiyle oluşturulur[19].

Bütün  $i = 1, 2, \dots, k$  ve  $i \neq j$  için  $\| x - m_j(t) \| < \| x - m_i(t) \|$  ise  $x \in S_j(t)$  dir

Bu ifadede  $S_j(t)$ , merkezi  $m_j(t)$  olan renk topağını temsil etmektedir. Bu algorithmada her yeni üye için yeniden renk merkezleri oluşturulur ve yeni merkezlerde üyelerin renk ortalaması alınarak yapılır. Histogram bilgisinden yararlanarak ağırlıklı ortalamaları alınır ve yeni renk merkezlerini temsil etmek için kullanılır. Denklem 4.25' deki  $h_i$  sembolü  $i$  renginin histogramdan belirlenen rengin kullanım sıklığını,  $Th_j$  ise  $j$  renk grubunun kullanım miktarını göstermektedir[19].

$$m_j(t+1) = \frac{1}{Th_j} \sum_{x_i \in S_j(t)} h_i x_i \quad j = 1, 2, \dots, k \quad \forall x_i | x_i \in S_j(t) \quad (4.25)$$

Her renk gurubu  $m_j(t+1) = m_j(t)$  yakınsamışsa denklem 4.15 kullanılır[19].

$$Th_i(t) = \sum_{x_i \in S_j(t)} h_i \quad \forall x_i | x_i \in S_j(t) \quad (4.15)$$

## 5. ARAŞTIRMA SONUÇLARI VE TARTIŞMA

Görüntü işleme teknikleri ile diş rengi belirleme projemizde amacımız dişler arasında renk uyumu yakalamak olmuştur. Bu amaçla geliştirilmek istenen yazılımdan önce diş görüntü verisi ve renklerde bağlı kalınacak diş renk skalası gerekmektedir. Diş renk skalası için diş hekimi önerisi olarak VITA renk skalası kullanılmıştır. Gülüşümüzde en fazla dikkat çeken bölüm ağızımızın ortasında bulunan üst ön keser ve üst yan keser dişlerdir.

Kamera ile yapılan çalışmamızda tüm dişlerin yüzeyini bütünüyle algılayamayacağımız ve dikkat çeken kısmın üst ön keser ve üst yan keser dişler olmasında dolayı çalışmamız sadece bu 4 diş üzerinden sürdürülmüştür. Toplamda 507 adet görüntü elde edilmiştir. Yolo-V4 algoritması özelliği olduğu için görüntüler 416 \* 416 boyutlarında yeniden boyutlandırılmıştır ve görüntüleri daha anlaşılır olması için sıralı bir şekilde kaydedilmiştir. Görüntü sayısını artırmak ve eğitimin doğruluk oranını artırmak için eğitim verilerine data augmentation(veri arttırma) yöntemleri uygulanmıştır. Bu yöntem içerisinde görüntüler 90-180 derece döndürme, bulanıklaştırma, tuz-biber görüntüsü ekleme kontrast ayarları ile oynama gibi yöntemler kullanılarak görüntü sayısı arttırılmıştır. Daha sonra elimizde bulunan verilere etiketlenme işlemi yapılmıştır. Görüntüler üzerinde dişlerin bulunduğu bölgelerin koordinatları roboflow adlı uygulama ile tek tek kaydedilmiştir. Görüntülerin %70 eğitim verisi %20' si doğrulama verisi ve kalan %10' u ise test verisi olarak kullanılmıştır.

Eğitim aşamasına geçmek için öncelikle YOLO-V4 DarkNet modelini indirmek gerekmektedir. Darknet adlı dosyayı indirdikten sonra eğitim aşamasına geçmek için elde ettiğimiz görüntüleri ve etiket bilgilerini darknet içerisine atılması gerekmektedir.

Eğitimin daha kısa sürede bitmesi için Google Colap kullanılmıştır. Google Colap' ı kullanabilmek için ilk önce darknet isimli dosyayı Google Drive yüklemek gerekmektedir. Colap içerisinde Drive' a bağlandıktan sonra gerekli ayarlamalar yapılarak Google CPU üzerinden eğitime başlanmıştır. Eğitim yaklaşık 6 saat sürmüş, 3000 iterasyon ile %98,8 doğruluk oranına ulaşılmıştır. Eğitim sonucunda ağırlık ve cfg dosyası kaydedilmiştir. Eğitim bittikten sonra modelin hiç görmediği görüntüleri oluşturduğumuz eğitim sonucunda oluşturduğumuz ağırlık ve cfg dosyası üzerinden python3 programlama diliyle modelin doğruluğu test edilmiştir.



Şekil 5.1 YOLO sonuçları

Şekil 5.1’ de görüldüğü gibi rastgele seçilmiş görüntülerin etiketlenen diş olma olasılığı yüksek doğruluk oranlarında doğru bir şekilde tespit edilmiştir.

Klinik ortamında çektiğimiz görüntüleri YOLOv4 algoritmasından geçirdikten sonra ön 4 dişi çevreleyen dikdörtgenin kenar bilgileri alınmış ve bu çerçeve dahilinde işleme devam edilmiştir.

Çerçeve içerisinde bulunan görüntü kırılmıştır. Kırılmış görüntü HSV formatına çevrilmiştir. HSV formatındaki görüntü üzerinde filtreleme işlemleri uygulanmıştır. Damağı dişlerden ayırmak için elimizde bulunan görüntülerden ve skalalardan en geniş diş olma aralığı belirlendi ve işleme giren her görüntü için sadece H değeri 9-35 arasındaki, S değeri 50-180 arasındaki, V değeri 150-255 değer aralığındaki renkler aralığının gösterilmesi sağlanmıştır. Elde edilen bu HSV formatındaki görüntü 20 iterasyonda 1 doğruluk değerine sahip K-Means algoritmasına sokulmuştur. Burada 5 adet renk belirlenmiştir. Bu renklerden 1 tane siyah olarak damağı ve diş etini kırmızı, sarı, yeşil, mavi renkleri ise diş üzerinde bulunan rengi temsil etmektedir. Bu renkleri temsil eden piksel koordinatları hafızaya aktarılmıştır. Tek tek bu piksel değerlerinin HSV renk değeri ortalamaları alınmıştır. Bu ortalamalar diş skalası üzerindeki örnek dişlerin görüntülerinden alınan HSV renk ortalamaları ile karşılaştırılıp. Sonuç pyqt5 ara yüzü ile oluşturulan ekrana iletilmiştir. Ayrıca bu ara yüzün içerisinde kırılmış ağız görüntüsü ve renklendirilmiş diş görüntüsü de bulunmaktadır.

**Tablo 5.1** Vita Toothguide 3D-MASTER diş skalasının HSV renk uzayına göre renk ortalamaları

VITA Toothguide 3D-MASTER diş renk skalası Renk Kodu	Görüntülerin HSV Ortalaması
1M1	15.4, 28.4, 139.9
1M2	16.8, 41.1, 146.0
2L1,5	17.7, 45.7, 144.9
2L2,5	18.4, 62.4, 152.1
2M1	14.9, 33.0, 134.4
2M2	16.2, 51.0, 141.5
2M3	17.7, 69.2, 143.0
2R1,5	16.2, 47.1, 149.3

2R2,5	17.3, 62.7, 154.9
3L1,5	18.5, 54.9, 147.9
3L2,5	18.6, 74.3, 105.0
3M1	16.1, 42.0, 138.1
3M2	16.8, 57.0, 143.8
3M3	17.7, 73.4, 151.4
3R1,5	15.3, 57.5, 133.7
3R2,5	17.0, 79.3, 141.4
4L1,5	16.7, 66.8, 142.6
4L2,5	17.6, 85.8, 146.0
4M1	17.7, 73.3, 151.8
4M2	16.4, 68.1, 138.9
4M3	17.2, 89.0, 143.7
4R1,5	16.1, 66.4, 136.4
4R2,5	16.5, 86.4, 144.2
5M1	14.9, 65.9, 121.1
5M2	15.6, 88.5, 129.4
5M3	16.5, 118.4, 134.8

Tablo 5.1' de verilen tabloda VITA skalasından tek tek çekilmiş görüntülerin HSV ortalaması gösterilmektedir. Bu skaladan çekilen görüntülerde dişler tek tek arka plandan ayrıldıktan sonra diş üzerine düşen flaş etkisini azaltmak için H ve S değerini 5 değerinden küçük olan piksel değerlerini ortalamaya katmadan hesaplanmıştır.



Şekil 5.2 2M1 kodlu diş

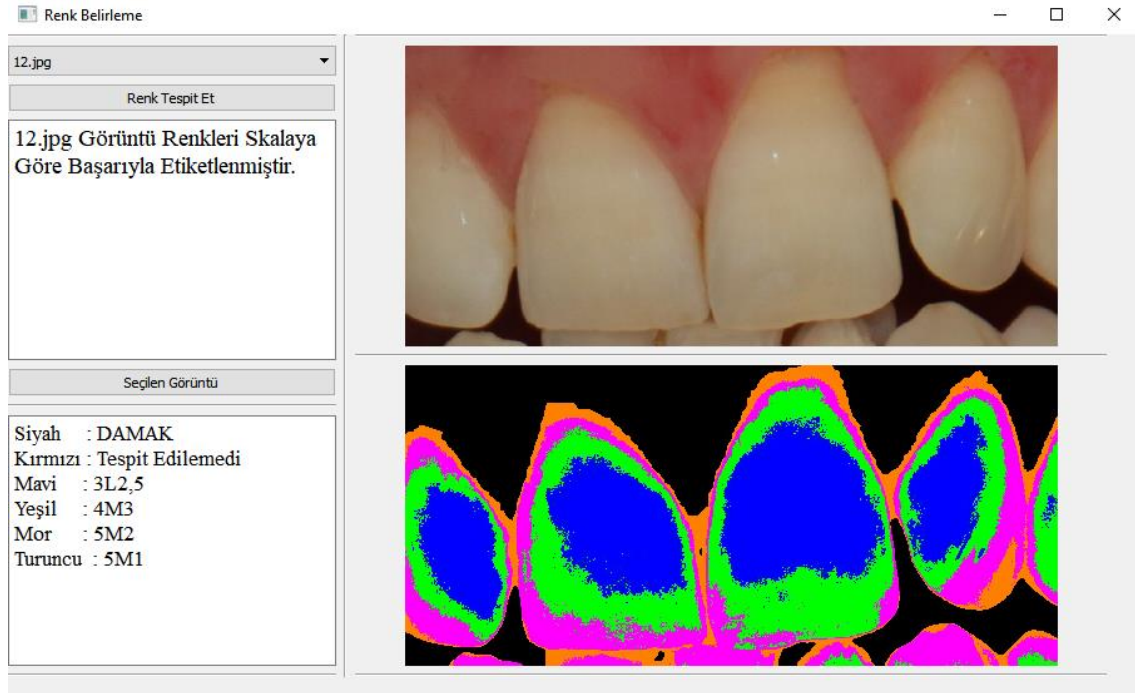


Şekil 5.3 4R2,5 kodlu diş



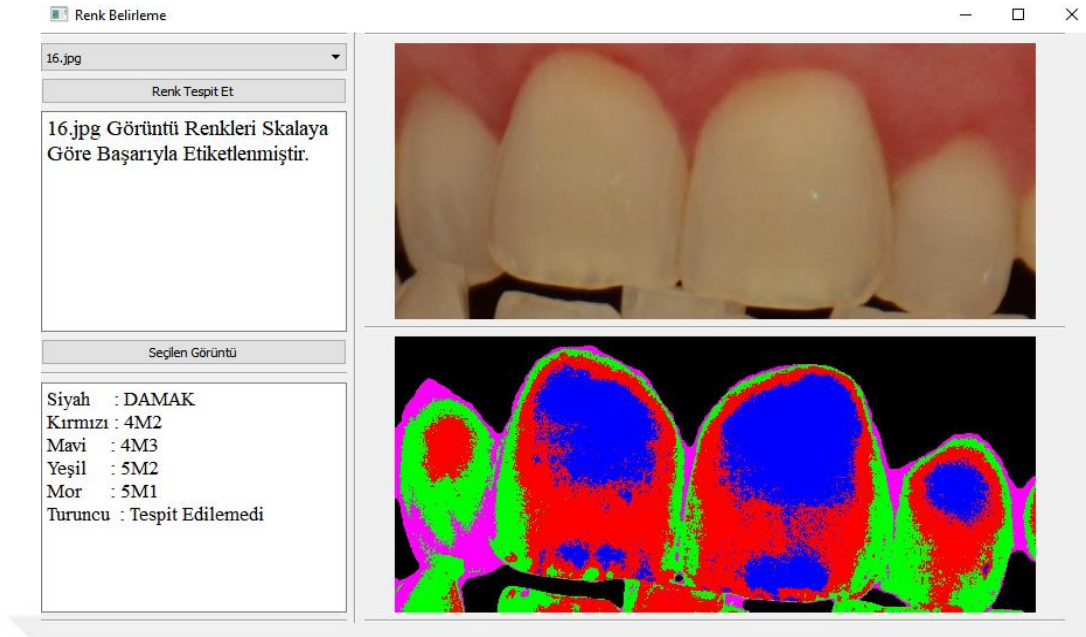
Şekil 5.2 ve Şekil 5.3’ de tek tek arka plandan ayrılmış diş skalası örneği gösterilmiştir. Diğer tüm skala dişler aynı şekilde ayarlanmıştır.

Toplamda 30 adet görüntü üzerinde yaptığımız çalışmalarda üst ön keser ve üst yan keser dişler VITA skalasına göre kodlanmıştır. Aşağıdaki şekillerde klinik ortamında çekilmiş 5 adet görüntünün kesilmiş hali ve renklendirilmiş hali GUI içerisinde gösterilmektedir.



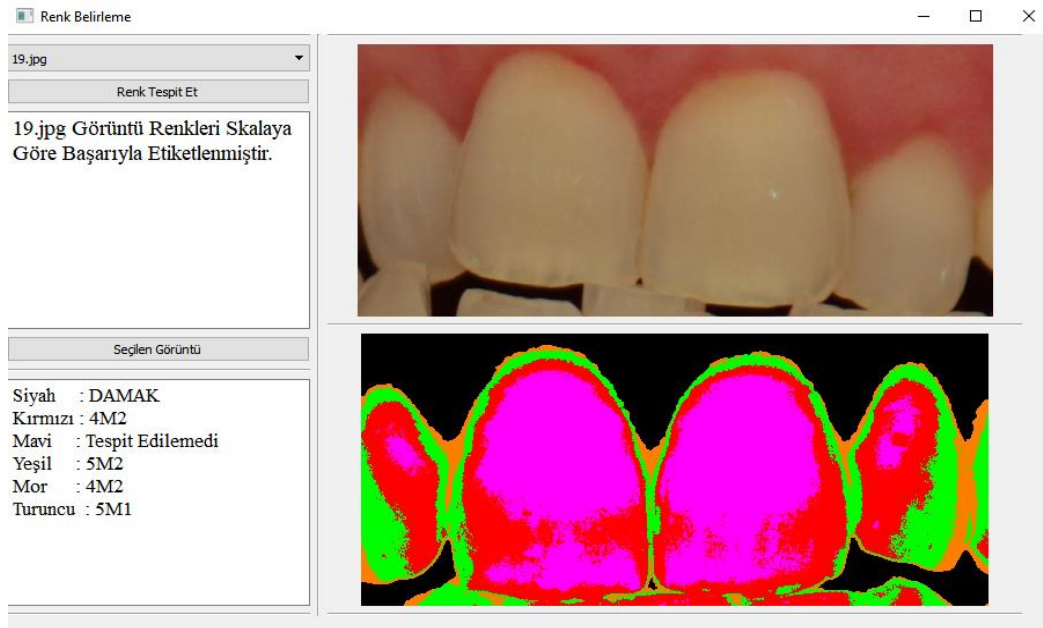
Şekil 5.4 Kümelenmiş görüntü

Şekil 5.4 de aynı kişiye ait orijinal fotoğrafın üst ön keser ve üst yan keser dişleri gösteren kesilmiş fotoğrafı ve kesilmiş bu fotoğraf üzerine uygulanan renk bulma algoritması sonucu çıkan görüntüdür. Bu görüntüde siyah diş etini göstermektedir. Kırmızı renk bulunmamaktadır. Mavi renk 3L2,5, yeşil renk 4M3, mor renk 5M2, turuncu renk 5M1 olarak etiketlenmiştir.



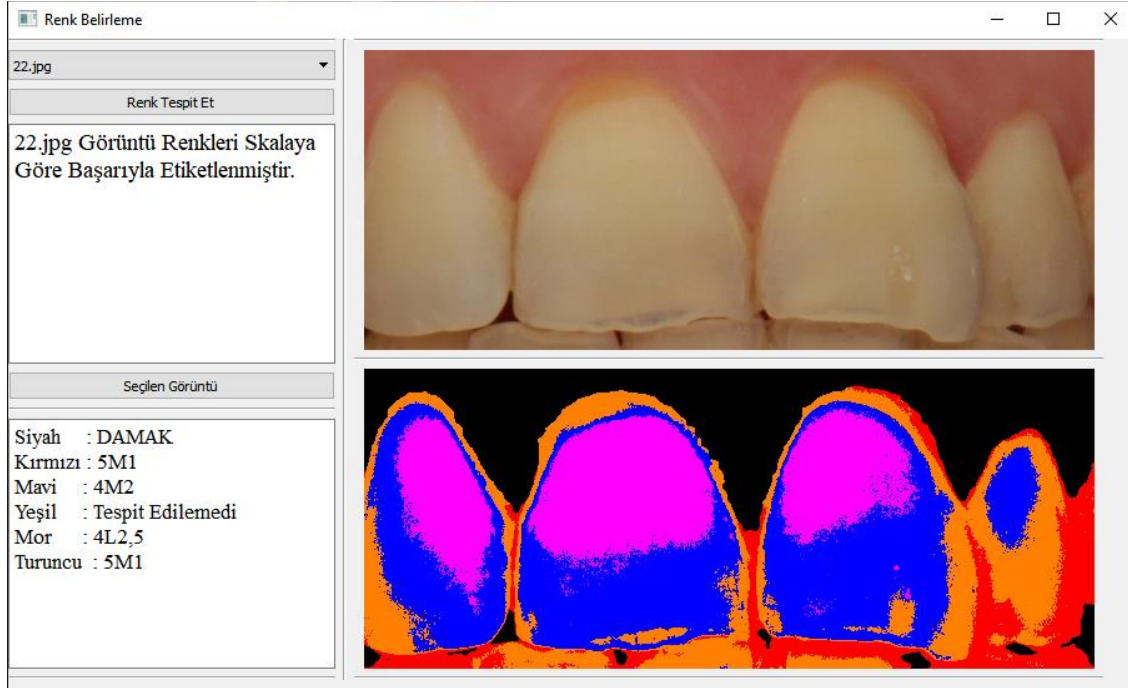
Şekil 5.5 Kümelennmiş ikinci görüntü

Şekil 5.5 de aynı kişiye ait orijinal fotoğrafın üst ön keser ve üst yan keser dişleri gösteren kesilmiş fotoğrafı ve kesilmiş bu fotoğraf üzerine uygulanan renk bulma algoritması sonucu çıkan görüntüdür. Bu görüntünün sol alt köşesinde renklerin vita skalasına göre karşılık geldiği renk kodları gösterilmiştir.



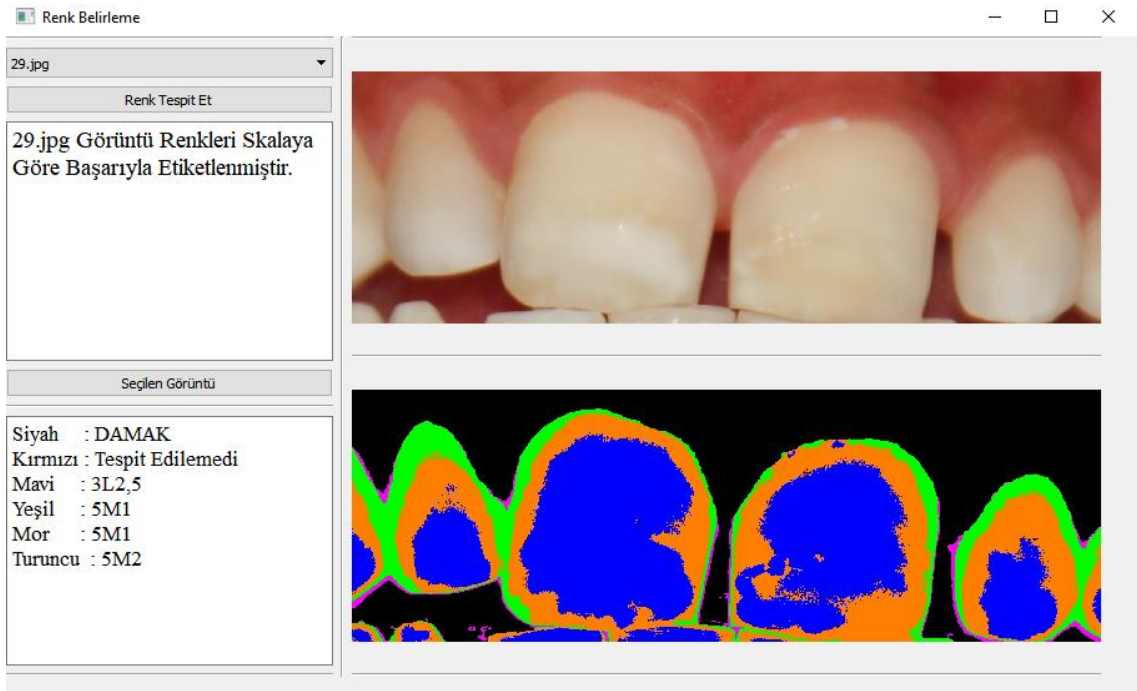
Şekil 5.6 Kümelennmiş üçüncü görüntü

Şekil 5.6' da aynı kişiye ait orijinal fotoğrafın üst ön keser ve üst yan keser dişleri gösteren kesilmiş fotoğrafı ve kesilmiş bu fotoğraf üzerine uygulanan renk bulma algoritması sonucu çıkan görüntüdür. Bu görüntünün sol alt köşesinde renklerin vita skalasına göre karşılık geldiği renk kodları gösterilmiştir.



Şekil 5.7 Kümelenmiş dördüncü görüntü

Şekil 5.7' de aynı kişiye ait orijinal fotoğrafın üst ön keser ve üst yan keser dişleri gösteren kesilmiş fotoğrafı ve kesilmiş bu fotoğraf üzerine uygulanan renk bulma algoritması sonucu çıkan görüntüdür. Bu görüntünün sol alt köşesinde renklerin vita skalasına göre karşılık geldiği renk kodları gösterilmiştir.



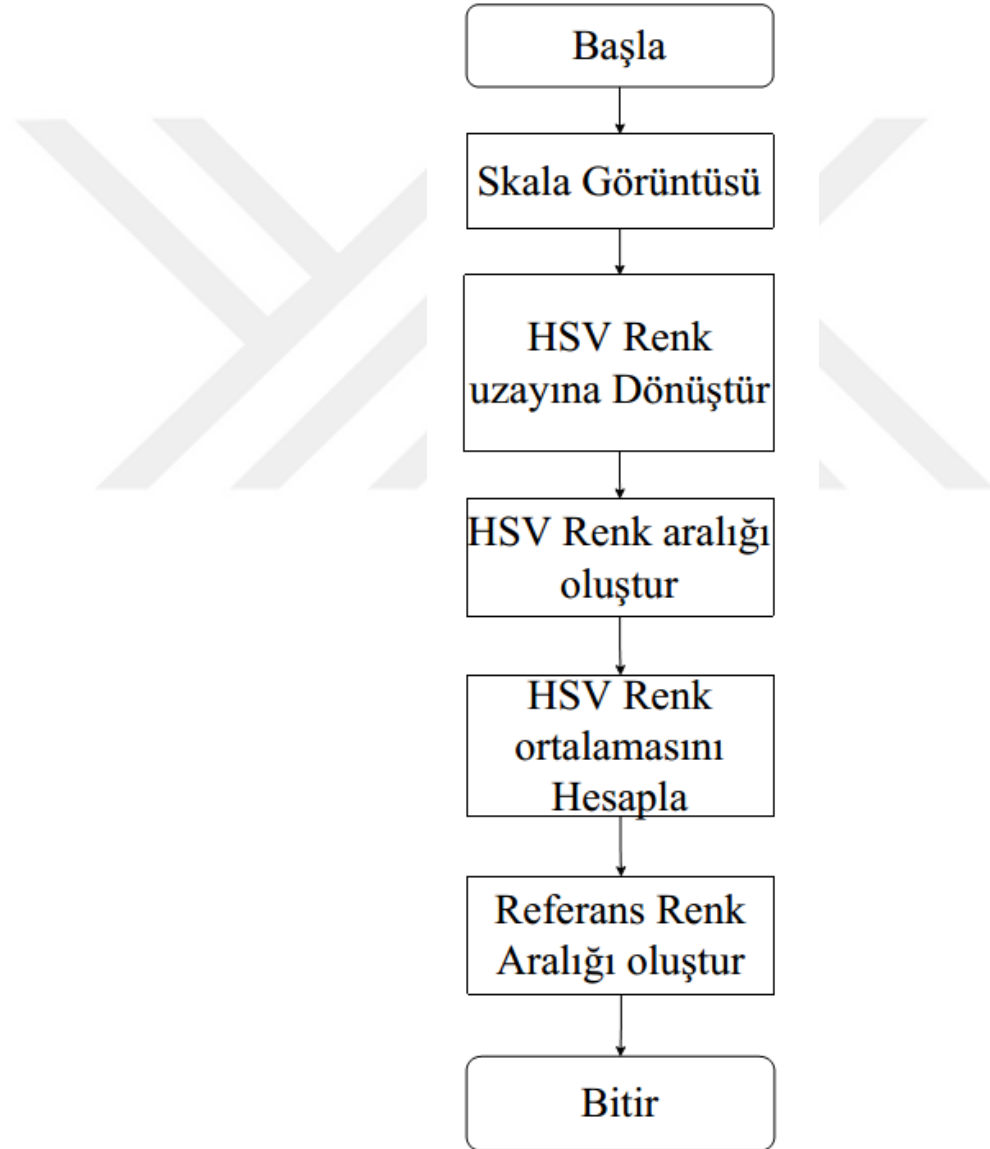
Şekil 5.8 Kümelenmiş beşinci görüntü

Şekil 5.8 de aynı kişiye ait orijinal fotoğrafın üst ön keser ve üst yan keser dişleri gösteren kesilmiş fotoğrafı ve kesilmiş bu fotoğraf üzerine uygulanan renk bulma algoritması sonucu çıkan görüntüdür. Bu görüntünün sol alt köşesinde renklerin vita skalasına göre karşılık geldiği renk kodları gösterilmiştir.

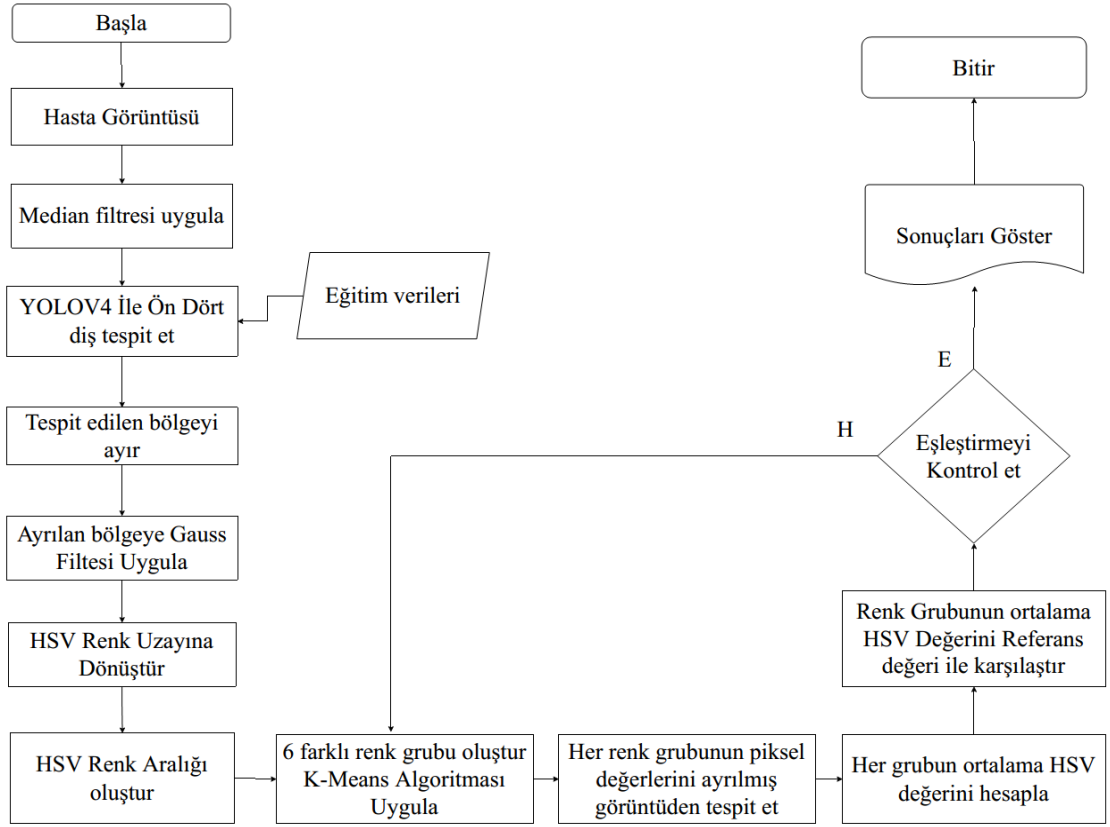
Bu görüntülerde en az bir adet renk yok olarak gösterilmiştir çünkü her görüntüde diş eti veya damak rengi özel olarak siyah görüntülenmesi istenilmiştir. Ayrıca 30 adet görüntü üzerinden rastgele olarak alınan bu görüntülerde görüldüğü gibi genel de renk dişin üst kısımlarına doğru değişmiş ve renk dağılımı arada fazla fark olmadan tespit edilmiştir.

Özet olarak bu tezde ilk önce ön dört diş YOLOV4 algoritması ile bulabilmek için görüntü toplanmıştır. Vita Skalasından alınan örneklerin HSV renk aralığı ortalama olarak belirlenmiştir. Şekil 5.9' de skaladan alınan örneklere uygulanan işlemin program akış şeması gösterilmiştir. YOLOV4 eğitiminden aldığımız ağırlık dosyasıyla klinik ortamın çekilmiş asıl görüntülerimizdeki ön dört diş tespit edilmiştir. Tespit edilen ön dört dişe filtreleme işlemi uygulanarak görüntüde oluşan parazitler engellenmeye çalışılmış ve kenar bilgileri ön plana alınmıştır. Filtrelenmiş görüntüye K-Means algoritması uygulanarak görüntüde bulunan dişler haritalanmıştır. Yeniden 5 farklı renk ile renklendirilen görüntünün piksel konumları belirlenerek her renge karşılık gelen piksel

değerinin ortalaması hesaplanmıştır. Daha önce belirlenen Vita skalasından alınan örneklerin HSV renk değeri aralıklarına göre etiketlenmiştir. Her zaman damak ve diş eti siyah renk ile gösterilmiştir. K-Means algoritmasında oluşturulan renk grupları rastgele oluşturulduğu için bir renk grubu fazladan oluşturulmuştur. Damak ve diş eti HSV renk değerini sağlayan renk grubu daha sonradan siyah renkle kaplanıp ortalama hesaplamasına sokulmamış ve skalaya göre etiketlenilmemiştir. Şekil 5.10' de program akış şeması gösterilmiştir.



Şekil 5.9 Vita Skalasına uygulanan program akış şeması



Şekil 5.10 Program Akış Şeması

## 5.2. Uygulamada Kullanılan Kamera ve Lens Özellikleri

Bu projede Canon EOS 650D marka kamera kullanılmıştır. Özellikleri ise Tablo-2' de gösterilmiştir.



Şekil 5.11 Canon EOS 650D

**Tablo 2** Canon EOS 650D Teknik Özellikleri

Özellikler	Açıklaması
<b>Çözünürlük</b>	18 Megapiksel
<b>Sensör</b>	22.3 x 14.9 mm APS-C CMOS
<b>Netlik</b>	9 noktalı netlik sistemi
<b>ISO</b>	100-12.800 hassasiyeti, ISO 25.600'e genişletilebilir
<b>Seri Çekim</b>	Saniyede 5 kare
<b>Video</b>	Full HD (1080p) saniyede 30 kare
<b>Ekran</b>	3 inç, 1 milyon piksel değişken açılı dokunmatik ekran
<b>İşlemci</b>	Digic 5
<b>Fotoğraf formatları</b>	JPEG ve RAW
<b>Ek özellikler</b>	HDMI çıkışı, Mikrofon girişi, LiveView, HDR çekim modu, EyeFi desteği
<b>Objektif uyumu</b>	EF ve EF-S serisi ile tam uyumlu
<b>Pil ömrü</b>	CIPA standartlarına göre 440 kare
<b>Ağırlık</b>	575 gram (pil ile sadece gövde)

Bu projede kamera lensi olarak Canon Macro Ring Lite MR-14EX ii markası kullanılmıştır. Şekil 5.11' de projede kullanılan kameranın görüntüsü gösterilmiştir. Bu lens ile çok yönlü aydınlatma yapabilen halka flaş ve aydınlatma kontrolü yapabilmek için çift flaş tüpü kullanılmıştır. Böylece kompozisyon ve odaklama kolaylaşmıştır. Çift flaş tüpün bağımsız şekilde ayarlanabilir ve altı adımlı oran sayesinde farklı aydınlatma şekilleri oluşturabilir. Gölgesiz efekt için iki tüpü eşit şekilde kullanılır veya bir tüpü diğerinden daha parlak yaparak dokuları vurgulayan, kabartma efektli bir aydınlatma elde edilir. Kılavuz numarası 14 (ISO 100, metre) ile elde edilen yüksek güç çıkışı, yakın çekim yaparken maksimum alan derinliği için küçük diyafram açıklıklarının kullanılmasına izin verir. Renk sıcaklığı bilgilerinin çekim sırasında fotoğraf makinesine geri aktarılması ve fotoğraf makinesinin otomatik beyaz dengesini optimize etmenize yardımcı olur. Şekil 5.12' de Canon Macro Ring Lite MR-14EX ii cihazı gösterilmektedir.



Şekil 5.12 Canon Macro Ring Lite MR-14EX ii





## 6. SONUÇLAR VE ÖNERİLER

### 6.1 Sonuçlar

Reostatif diş hekimliğinde renk belirleme analizi ve sonuçlarının değerlendirilmesi elektronik aletler kullanarak sübjektifliğini ve hataları ortadan kaldırmaktadır. Bu yüzden kliniklerde genel olarak spektrofotometre ve kolorimetre gibi doğruluğu yüksek ve objektif sonuç veren ekipmanlar kullanılmaktadır. Bu ekipmanlar aynı zamanda maliyetli, hassasiyet ayarı zor ve kısıtlı olduğu için alternatif olarak doğru sonuçlar verebilecek RGB cihazlara da ihtiyaç duyulmaktadır.

Bu tez kapsamında üst ön keser ve üst yan keser dişler klinik ortamında görüntüleyerek diş renk haritalanması VITA 3D Master skalasına göre etiketlenmiştir. Yapılan deneylerde özellikle ortam ışığına, çekim açısına ve uzaklığına dikkat edilmesi gerekmektedir. Bu bağlamda dışarıdan ışığa ihtiyaç duyulmaksızın kameraya bağlı olan halka flaş ışığı kullanılmış yaklaşık 1 metre uzaklıktan ve hasta dişlerini tam karşıdan görecektir şekilde görüntüler çekilmiştir. Aynı şekilde Vita 3D Master skalasında bulunan örneklerde tek tek görüntülenmiştir.

Klinik ortamında çekilen hasta görüntüleri ve skala örnekleri bilgisayar ortamında işlenerek önce YOLOv4 algoritmasından geçirilerek sadece ön dört dişin bulunduğu alan üzerinde işlem yapılmaya devam edilmiştir. Bu kesilen görüntü üzerinde bir çok filtre çeşidi denenmiş ve başarılı olan gauss filtresi kullanarak görüntü üzerindeki parazitler büyük ölçüde engellenmiş ve görüntü HSV renk uzayına döndürülmüştür. Görüntü üzerinde mekandan ve halka flaşdan kaynaklı oluşan yoğun beyaz ışığı engellemek için hue ve satürasyon değerleri her görüntüde ortak olacak şekilde belirli bir aralığa indirgenerek ortalama hesabına etkisi ortadan kaldırılmıştır. Kmeans algoritması ile birbirine yakın renk grupları haritalanarak piksel konumları hafıza ayrı ayrı tutularak HSV renk piksel ortalama hesabı yapılmış ve skalaya göre eşleştirilmiştir.

Skaladan elde edilen HSV renk ortalamalarına göre etiketlenen görüntülerde renk dağılımı arasında uçurum olmadan her defasında aynı sonuçlar üreten görüntüler elde edilmiştir. Göz ile yapılan sonuçlar ve bu tez kapsamında üretilen sonuçlar birbirine benzemektedir.

## 6.2 Öneriler

Öncelikle hastanın gerçek diş rengini değiştirecek kimyasal bir madde kullanmaması gerekmektedir. Hasta dişi kendini belli edebilecek şekilde temiz olmalıdır. Klinik ortamında çekilen hasta görüntülerinin ve skala görüntülerinin hem uzaklık hem de açı bakımında sabit olması gerekmektedir bu bağlamda trimpot vb. araçlarla kamera sabitlenmeli ve kamera ile hasta arasındaki mesafe sabit tutulmalıdır. El ile yapılan çekimlerde açı ve mesafe sürekli sabit kalmadığı için optimizasyon zorlaşmaktadır. Ayrıca orijinal renk bilgilerini ön planda tutan kamera lenslerinin kullanılması renk tespitinin doğruluk oranını da arttıracaktır. Skala üzerinden çekilen diş örnekleri tek tek çekildiği için arka plan rengi seçimi çok koyu veya çok açık olmayan renkler tercih edilmelidir. Genel olarak gri renkler kullanılmalıdır. Bu işlemler gerçekleştirildiğinde bilgisayar ortamında kullanılan filtrelerin ağırlık ve boyutlarının optimizasyonu da kolaylaşacak ve daha sağlıklı sonuçlar elde etmemizi sağlayacaktır. Aynı zamanda kullanılan bu filtrelerle beraber görüntü üzerinde morfolojik işlemler uygulamak her dişin kenar bilgilerini artırarak görüntüde bulunan tüm yakın renkleri gruplandırmadan her dişi kendi içerisinde değerlendirerek ve gruplandırarak daha doğru sonuçlar elde edebiliriz. Bu tez sonucunda üretilen sonuçların doğruluğunu daha iyi kanıtlayabilmek ve sonuçların doğruluğu arttırmak için objektif ve doğru sonuçlar veren cihaz sonuçları ile karşılaştırılması gerekmektedir.

**KAYNAKLAR**

- Çağlar A., Yamanel K., 2007, Diş renginin belirlenmesinde kullanılan yöntemler, ADO Klinik Bilimler Dergisi, 2(1), Ankara, 49-54
- Keyf F, Uzun G., Altınsoy S., 2009, Diş hekimliğinde renk seçimi, Hacettepe Diş Hekimliği Dergisi, 33(4), 52-58
- Kahramanoğlu E., Özkan Y.K., 2011, Esthetic and color in denstistry, Cumhuriyet Dent Journal, 16(4), 339-347
- Bayındır F., Wee A.G., 2006, The use of computer aided system in tooth shade-matching, Hacettepe Diş Hekimliği Fakültesi Dergisi, 30(3), Ankara, 40-46
- Doğan A., Yüzügüllü B., 2011, Recent technological developments in color selection, Atatürk Üniversitesi Diş Hekimliği Fakültesi Dergisi, 4, 65-72
- Akan A., 2011, Elektronik sistemlerle diş renginin tespit edilmesi, İstanbul Üniversitesi Fen Bilimleri Enstitüsü Yüksek Lisans Tezi, 78
- HACIHASANOĞLU H., 2005, Diş hekimliğinde kullanılan porselenlerin renk dağılımının ve renk seçme yöntemlerinin araştırılması, İstanbul Üniversitesi Sağlık Bilimleri Enstitüsü Yüksek Lisans Tezi, 55
- Kurt M., Bal Turhan B., Bal Cenkhan, 2016, Actual methods of color measurement: a systematic review, Türkiye Klinikleri Journal of Dental Sciences, 22(2), Ankara, 130-146
- Öngül D., Çelik B., İlbey D., Şermet B., 2013, Türk toplumundaki genç bireylerde diş renk dağılımının incelenmesi, İstanbul Üniversitesi Diş Hekimliği Fakültesi Dergisi, 47(2), İstanbul, 30-40
- Saraç Y.Ş., Saraç D., Yüzbaşıoğlu E., 2005, Üç farklı renk skalasının renk farklılıkları yönünden kolometrik olarak incelenmesi, Gazi Üniversitesi Diş Hekimliği Fakültesi Dergisi, 23(2), Ankara, 85-90
- Albayrak A., Bilgin G., Histopatolojik görüntülerde renk uzaylarının hücre bölütlenmesine etkileri, IEEE, Zonguldak
- Bhat V., Prasad K.D., Bhat A., 2011, Role of colors in proshodontics: Application of color sciencd in restorative dentistry, Indian Journal of Dental Research, 22(6), Hindistan, 804-809
- Bochkovskiy A., Wang C., Liao H.M, 2020, YOLOv4: Optimal Speed and Accuracy of Object Detection, arXiv:2004.10934 [cs.CV]
- Du J., 2008, Understanding of object detection based on CNN family and YOLO, Journal of Physics: Conference Series, 1004, Hong Kong

Aktan S., 2004, Distinguishable color analysis for egg yolk via digital image processing, 4. Ulusal Zootekni Bilim Kongresi

Datta S., Chaki N., 2015, Person Identification Technique Using RGB Based Dental Images, 14th Computer Information Systems and Industrial Management (CISIM), Poland, 169-180

Tambe S.B., Kulhare D., Nirmal M.D, Parajapati G., 2013, Image processing (IP) through erosion and dilations meethods, International Journal of Emerging Technology and Advanced Engineering, 3(7)

Zhue G., Piao Z., Kim S.C., 2020, Tooth detection and segmentation with mask R-CNN, 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIC), Japan

Chen T., Chen Y., Chien S., 2008, Fast image segmentation based on K-means clustering with histograms in HSV color space, 2008 IEEE 10th Workshop on Multimedia Signal Processing, Australia

Albayrak S., Karslıgil., 2001, Renkli görüntülerde ağırlıklı K-means yöntemi ile renk toplama, 9.Sinyal İşleme ve Uygulamaları Kurultayı -SİU, Lefkoşa, 1-5

<https://www.vita-zahnfabrik.com/en/VITA-Zahnfabrik-This-is-how-prosthetics-are-made-88384.html>

[https://www.ivoclar.com/en\\_us/shop/p/teeth/chromascopshadeguide/p/529479](https://www.ivoclar.com/en_us/shop/p/teeth/chromascopshadeguide/p/529479)

<https://www.dismat.com/dijital-cozumler/vita-easyshade-v/>

<https://www.olympus-global.com/en/news/2006b/nr061113crystale.html>

<https://spectroshade.us/products/spectroshade-micro-ii>

<https://www.natural-esthetics.com/index.php?lang=en&page=shade-eye-ncc-en.html>

Müdüroğlu R., Kıvrak T.Ç., Nağacı A., 2018, Renk belirlenmesinde kullanılan yöntem ve cihazlar, Cumhuriyet Dental Journal, 21(1), Ankara, 61-69

<https://bilimgenc.tubitak.gov.tr/makale/haberlesme-bilgi-uzak-mesafelere-nasil-gonderilir>

<https://www.pngegg.com/tr/png-tbpnn>

<https://stringfixer.com/tr/Hue>

Gonzalez R.C., Woods R.E., 2008, Digital Image Processing, Pearson, Fourth Edition, New York,

<https://ai.stanford.edu/~syeyung/cvweb/tutorial1.html>

<https://sh-tsang.medium.com/review-yolov4-optimal-speed-and-accuracy-of-object-detection-8198e5b37883>

<https://jonathan-hui.medium.com/yolov4-c9901eaa8e61>

<https://medium.com/data-folks-indonesia/step-by-step-to-understanding-k-means-clustering-and-implementation-with-sklearn-b55803f519d6>



## EKLER

**EK-1** VITA Toothguide 3D-MASTER skalasında bulunan dişlerin HSV uzayında renk ortalamasının alınması. “HSV\_ort\_color.py” dosyası

```

import cv2
import numpy as np
import os

mevcut_dizin = os.getcwd()
fotolar = []
for i in os.listdir(mevcut_dizin):
    if i.endswith(".png"):
        fotolar.append(i)
for n in fotolar:
    new_img = cv2.imread(n)
    hsv_frame = cv2.cvtColor(new_img, cv2.COLOR_BGR2HSV)
    new_gor = hsv_frame
    rows, cols, _ = new_gor.shape
    H = 0
    S = 0
    V = 0
    sayac = 0
    H_list = []
    S_list = []
    V_list = []
    HSV_list_son = []
    for i in range(0, rows, 1):
        for j in range(0, cols, 1):
            k = new_gor[i, j]
            # print(k[0],k[1],k[2])
            # if (177 < k[0] < 181) and (224 < k[1] < 228) and (235 < k[2] < 240):
            if (8 <= k[0] <= 33) and (20 <= k[1] <= 200) and (0 <= k[2] <= 230):
                H += k[0]
                S += k[1]
                V += k[2]
                # print(k[0],k[1],k[2])
                sayac += 1
    try:
        son_H = H / sayac
    except ZeroDivisionError:
        son_H = 0
    try:
        son_S = S / sayac
    except ZeroDivisionError:
        son_S = 0
    try:
        son_V = V / sayac
    except ZeroDivisionError:
        son_V = 0
    HSV_list = []
    HSV_list.append(round((son_H), 1))
    HSV_list.append(round((son_S), 1))
    HSV_list.append(round((son_V), 1))
    HSV_list_son.append(HSV_list)
    print(" {0} = {1}".format(n,HSV_list_son))

cv2.waitKey(0)
cv2.destroyAllWindows()

```

**EK-2** YOLO Algoritmasıyla üst ön keser ve üst yan keser dışların tespiti. “dis\_bul.py” dosyası.

```

import cv2
import numpy as np
import time

class yolo_dis_tespiti:
    # -*- coding: utf-8 -*-
    def dis_tespiti(self,model,frame):
        durum = 0
        self.frame = frame
        self.model =model
        frame_width = self.frame.shape[1]
        frame_height = self.frame.shape[0]
        frame_blob = cv2.dnn.blobFromImage(self.frame, 1 / 255, (416, 416), swapRB=True, crop=False)
        labels = ["Dis"]
        colors = ["0,0,255", "0,0,255", "255,0,0", "255,255,0", "0,255,0"]
        colors = [np.array(color.split(",")).astype("int") for color in colors]
        colors = np.array(colors)
        colors = np.tile(colors, (18, 1))
        layers = self.model.getLayerNames()
        output_layer = [layers[layer[0] - 1] for layer in self.model.getUnconnectedOutLayers()]
        self.model.setInput(frame_blob)
        detection_layers = self.model.forward(output_layer)
        ids_list = []
        boxes_list = []
        confidences_list = []
        for detection_layer in detection_layers:
            for object_detection in detection_layer:
                scores = object_detection[5:]
                predicted_id = np.argmax(scores)
                confidence = scores[predicted_id]
                if confidence > 0.70:
                    durum = 1
                    label = labels[predicted_id]
                    bounding_box = object_detection[0:4] * np.array(
                        [frame_width, frame_height, frame_width, frame_height])
                    (box_center_x, box_center_y, box_width, box_height) = bounding_box.astype("int")
                    # cv2.putText(frame, "("+str(box_center_x)+str(box_center_y)+")", (box_center_x+20,
                    box_center_y+20), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (255, 255, 255), 1)
                    self.start_x = int(box_center_x - (box_width / 2))
                    self.start_y = int(box_center_y - (box_height / 2))
                    ids_list.append(predicted_id)
                    confidences_list.append(float(confidence))
                    boxes_list.append([self.start_x, self.start_y, int(box_width), int(box_height)])
        max_ids = cv2.dnn.NMSBoxes(boxes_list, confidences_list, 0.5, 0.4)
        for max_id in max_ids:
            max_class_id = max_id[0]
            box = boxes_list[max_class_id]
            self.start_x = box[0]
            self.start_y = box[1]
            box_width = box[2]
            box_height = box[3]
            predicted_id = ids_list[max_class_id]
            label = labels[predicted_id]
            confidence = confidences_list[max_class_id]
            self.end_x = self.start_x + box_width
            self.end_y = self.start_y + box_height
            box_color = colors[predicted_id]

```

```
box_color = [int(each) for each in box_color]
label = "{: {:.2f}%".format(label, confidence * 100)
merkez_x = int(np.round(self.start_x + box_width / 2))
merkez_y = int(np.round(self.start_y + box_height / 2))
cv2.rectangle(self.frame, (self.start_x, self.start_y), (self.end_x, self.end_y), box_color, 1)
cv2.rectangle(self.frame, (self.start_x - 1, self.start_y), (self.end_x + 1, self.start_y - 30), box_color,
-1)
cv2.putText(self.frame, label, (self.start_x, self.start_y - 10), cv2.FONT_HERSHEY_SIMPLEX,
0.5, (255, 255, 255), 2)
```





**EK-2** Kesilmiş görüntüden renk tespit etme algoritması. “dis\_renk\_sonuc.py” dosyası.

```

import time, cv2, os
import numpy as np
import random
from dis_bul import yolo_dis_tespiti

class renk_bul():
    def color(self, belirlenen_goruntu):
        frame = cv2.imread("goruntuler/"+ belirlenen_goruntu+".jpg")
        self.yolo = yolo_dis_tespiti()
        self.yolo.dis_tespiti(model=cv2.dnn.readNetFromDarknet("yolov4-
detector.cfg","final.weights"), frame=frame)
        self.start_x, self.start_y, self.end_x, self.end_y = self.yolo.start_x, self.yolo.start_y,
self.yolo.end_x, self.yolo.end_y
        self.kesilmis_goruntu = frame[self.start_y + 5:self.end_y-15,self.start_x + 80:self.end_x-50]
        kes_im = self.kesilmis_goruntu.copy()
        img = self.kesilmis_goruntu.copy()
        hsv_frame = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        median = cv2.GaussianBlur(hsv_frame, (71, 41), cv2.BORDER_DEFAULT)
        lower_dis = np.array([9, 50, 150])
        upper_dis = np.array([37, 180, 255])
        dis_mask = cv2.inRange(median, lower_dis, upper_dis)
        dis = cv2.bitwise_and(img, img, mask=dis_mask)
        img_k = dis
        img1_cop = img_k.copy()
        Z = img_k.reshape((-1, 3))
        Z = np.float32(Z)
        criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 20, 1.0)
        K = 5
        ret, label, center = cv2.kmeans(Z, K, None, criteria, 10,
cv2.KMEANS_RANDOM_CENTERS)
        #print(center)
        for sira, i in enumerate(center):
            if i[0] == 0 and i[1] == 0 and i[2] == 0:
                self.k = sira
        center = [[255, 0, 0], [0, 255, 0], [0, 255, 255], [0, 0, 255], [255, 0, 255]]
        center[self.k] = [0, 0, 0]
        print(center)

```

```

center = np.uint8(center)
res = center[label.flatten()]
res2 = res.reshape((img_k.shape))
yesil = []
mavi = []
mor = []
sari = []
kirmizi = []
siyah = []
rows, cols, _ = res2.shape
print(rows,cols)
for i in range(0, rows, 1):
    for j in range(0, cols, 1):
        c = res2[i, j]
        if c[0] == 255 and c[1] == 0 and c[2] == 0:
            kirmizi.append([i, j])
        if c[0] == 0 and c[1] == 255 and c[2] == 0:
            yesil.append([i, j])
        if c[0] == 0 and c[1] == 0 and c[2] == 255:
            mavi.append([i, j])
        if c[0] == 255 and c[1] == 0 and c[2] == 255:
            mor.append([i, j])
        if c[0] == 255 and c[1] == 255 and c[2] == 0:
            sari.append([i, j])
        if c[0] == 0 and c[1] == 0 and c[2] == 0:
            siyah.append([i, j])
if len(yesil) > 0:
    try:
        self.renk_ort(img1_cop, yesil)
        self.yesil_H, self.yesil_S, self.yesil_V, self.yesil_scala = self.H_ort, self.S_ort,
self.V_ort, self.scala
    except:
        self.yesil_H, self.yesil_S, self.yesil_V, self.yesil_scala = 0, 0, 0, "Piksel değeri yok"
else:
    self.yesil_H, self.yesil_S, self.yesil_V, self.yesil_scala = 0, 0, 0, "Yesil Renk yok"

```

```
print("Yesil: ", round(self.yesil_H, 1), round(self.yesil_S, 1), round(self.yesil_V, 1),
self.yesil_scala)
```

```
if len(mavi)>0:
```

```
try:
```

```
self.renk_ort(img1_cop, mavi)
```

```
self.mavi_H, self.mavi_S, self.mavi_V, self.mavi_scala = self.H_ort, self.S_ort,
self.V_ort, self.scala
```

```
except:
```

```
self.mavi_H, self.mavi_S, self.mavi_V, self.mavi_scala = 0, 0, 0, "Piksel değeri yok"
```

```
else:
```

```
self.mavi_H, self.mavi_S, self.mavi_V, self.mavi_scala = 0, 0, 0, "Mavi Renk yok"
```

```
print("Mavi: ", round(self.mavi_H, 1), round(self.mavi_S, 1), round(self.mavi_V, 1),
self.mavi_scala)
```

```
if len(mor) > 0:
```

```
try:
```

```
self.renk_ort(img1_cop, mor)
```

```
self.mor_H, self.mor_S, self.mor_V, self.mor_scala = self.H_ort, self.S_ort, self.V_ort,
self.scala
```

```
except:
```

```
self.mor_H, self.mor_S, self.mor_V, self.mor_scala = 0, 0, 0, "Piksel değeri yok"
```

```
else:
```

```
self.mor_H, self.mor_S, self.mor_V, self.mor_scala = 0, 0, 0, "Mor Renk yok"
```

```
print("Mor: ", round(self.mor_H, 1), round(self.mor_S, 1), round(self.mor_V, 1),
self.mor_scala)
```

```
if len(sari)>0:
```

```
try:
```

```
self.renk_ort(img1_cop, sari)
```

```
self.sari_H, self.sari_S, self.sari_V, self.sari_scala = self.H_ort, self.S_ort, self.V_ort,
self.scala
```

```
except:
```

```
self.sari_H, self.sari_S, self.sari_V, self.sari_scala = 0, 0, 0, "Piksel değeri yok"
```

```
else:
```

```
self.sari_H, self.sari_S, self.sari_V, self.sari_scala = 0, 0, 0, "sari renk yok"
```

```
print("Sari: ", round(self.sari_H, 1), round(self.sari_S, 1), round(self.sari_V, 1),
self.sari_scala)
```

```
if len(kirmizi)>0:
```

```

        try:
            self.renk_ort(img1_cop, kirmizi)
            self.kirmizi_H, self.kirmizi_S, self.kirmizi_V, self.kirmizi_scala = self.H_ort,
self.S_ort, self.V_ort, self.scala
        except:
            self.kirmizi_H, self.kirmizi_S, self.kirmizi_V, self.kirmizi_scala = 0, 0, 0, "Piksel değeri
yok"
        else:
            self.kirmizi_H, self.kirmizi_S, self.kirmizi_V, self.kirmizi_scala = 0,0,0,"Kirmizi renk
yok"

        print("kirmizi: ", round(self.kirmizi_H, 1), round(self.kirmizi_S, 1), round(self.kirmizi_V,
1), self.kirmizi_scala)
        if len(siyah) > 0:
            self.renk_ort(img1_cop, siyah)
            self.siyah_H, self.siyah_S, self.siyah_V, self.siyah_scala = self.H_ort, self.S_ort,
self.V_ort, self.scala
        else:
            self.siyah_H, self.siyah_S, self.siyah_V, self.siyah_scala = 0,0,0,"siyah Bos"
        print("siyah: ", round(self.siyah_H, 1), round(self.siyah_S, 1), round(self.siyah_V, 1),
self.scala)
        cv2.destroyAllWindows()

        return self.kirmizi_scala, self.mavi_scala, self.yesil_scala, self.sari_scala, self.siyah_scala,
self.mor_scala

```

```
def renk_ort(self, image, array):
```

```
    H = 0
```

```
    S = 0
```

```
    V = 0
```

```
    sayac = 0
```

```
    H_ort = 0
```

```
    S_ort = 0
```

```
    V_ort = 0
```

```
    self.H_ort = 0
```

```
    self.S_ort = 0
```

```
    self.V_ort = 0
```

```
    self.scala = ""
```

```
    im = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
```

```
    for i in array:
```

```
l = im[i[0], i[1]]
if (9 <= l[0] <= 25) and (20 <= l[1] <= 130) and (90 <= l[2] <= 225):
    H += l[0]
    S += l[1]
    V += l[2]
    sayac += 1
if sayac != 0:
    self.H_ort = H/sayac
    self.S_ort = S/sayac
    self.V_ort = V/sayac
if self.H_ort < 15:
    if self.S_ort < 50:
        self.scala = "2M1"
    else:
        self.scala = "5M1"
if 15 <= self.H_ort < 16:
    if self.S_ort <= 42:
        self.scala = "1M1"
    if 42 < self.S_ort <= 65:
        self.scala = "3R1,5"
    else:
        self.scala = "5M2"
if 16.5 <= self.H_ort < 17:
    if self.S_ort <= 45:
        self.scala = "1M2"
    if 45 < self.S_ort <= 62:
        self.scala = "3M2"
    if 62 < self.S_ort <= 78:
        self.scala = "4L1,5"
    if 62 < self.S_ort <= 105:
        self.scala = "4R2,5"
    if 105 < self.S_ort <= 255:
        self.scala = "5M3"
if 16 <= self.H_ort < 17.5:
    if self.S_ort <= 43:
        self.scala = "3M1"
```

```
if 43 < self.S_ort <= 48:
    self.scala = "2R1,5"
if 48 < self.S_ort <= 58:
    self.scala = "2M2"
if 58 < self.S_ort <= 67:
    self.scala = "4R1,5"
if 68 < self.S_ort <= 255:
    self.scala = "4M2"
if 17 <= self.H_ort < 17.5:
    if self.S_ort <= 68:
        self.scala = "2R2,5"
    if 68 < self.S_ort <= 85:
        self.scala = "3R2,5"
    if 85 < self.S_ort <= 255:
        self.scala = "4M3"
if 17.5 <= self.H_ort < 18:
    if self.S_ort <= 54:
        self.scala = "2L1,5"
    if 57 < self.S_ort <= 72:
        self.scala = "2M3"
    if 72 < self.S_ort <= 73:
        self.scala = "3M3"
    if 73 < self.S_ort <= 82:
        self.scala = "4M1"
    if 82 < self.S_ort <= 255:
        self.scala = "4L2,5"
if 18 < self.H_ort:
    if self.S_ort <= 57:
        self.scala = "3L1,5"
    if 54 < self.S_ort <= 68:
        self.scala = "2L2,5"
    if 68 < self.S_ort <= 255:
        self.scala = "3L2,5"
if self.H_ort == 0 and self.S_ort == 0 and self.V_ort == 0:
    self.scala = "DAMAK"
```

**EK-2** PyQt5 ara yüz oluşturma kodu.

```

import sys
from PyQt5.QtCore import *
from PyQt5.QtGui import *
from PyQt5.QtWidgets import *
from dis_renk_sonuc import renk_bul

class MainWindow(QMainWindow):
    def __init__(self):
        super().__init__()
        self.ui = Ui_MainWindow()
        self.ui.setupUi(self)
        for i in range(1,45):
            self.ui.comboBox.addItem(str(i))
            self.ui.pushButton.clicked.connect(self.renk_sec)
        def renk_sec(self):
            goruntu = self.ui.comboBox.currentText()
            self.pixmap = QPixmap("goruntuler/"+ goruntu+".jpg")
            w = self.ui.asil_foto.width()
            h = self.ui.asil_foto.height()
            self.ui.asil_foto.setPixmap(self.pixmap.scaled(w,h,Qt.KeepAspectRatio))
            self.renk = renk_bul()
            x = self.renk.color(goruntu)
            self.pixmap_1 = QPixmap("renkli.jpg")
            w_1 = self.ui.renk_belirlenmis_foto.width()
            h_1 = self.ui.renk_belirlenmis_foto.height()
            self.ui.renk_belirlenmis_foto.setPixmap(self.pixmap_1.scaled(w_1,h_1,
Qt.KeepAspectRatio))
            self.ui.kirmizi_label.setText("Kirmizi= " + x[0])
            self.ui.sari_label.setText("Sari= " + x[3])
            self.ui.yesil_abel.setText("Yesil= " + x[2])
            self.ui.mavi_label.setText("Mavi = " + x[1])
            self.ui.siyah_label.setText("Mor = " + x[4])

app = QApplication(sys.argv)
ex = MainWindow()

```

```

ex.show()
sys.exit(app.exec_())
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(1920, 1080)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum,
QtWidgets.QSizePolicy.Maximum)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(MainWindow.sizePolicy().hasHeightForWidth())
        MainWindow.setSizePolicy(sizePolicy)
        MainWindow.setMinimumSize(QtCore.QSize(1920, 1080))
        MainWindow.setMaximumSize(QtCore.QSize(1920, 1080))
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.formLayout = QtWidgets.QFormLayout(self.centralwidget)
        self.formLayout.setObjectName("formLayout")
        self.verticalLayout = QtWidgets.QVBoxLayout()
        self.verticalLayout.setObjectName("verticalLayout")
        self.asil_foto = QtWidgets.QLabel(self.centralwidget)
        sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Minimum)
        sizePolicy.setHorizontalStretch(0)
        sizePolicy.setVerticalStretch(0)
        sizePolicy.setHeightForWidth(self.asil_foto.sizePolicy().hasHeightForWidth())
        self.asil_foto.setSizePolicy(sizePolicy)
        self.asil_foto.setMinimumSize(QtCore.QSize(1200, 500))
        self.asil_foto.setText("")
        self.asil_foto.setObjectName("asil_foto")
        self.verticalLayout.addWidget(self.asil_foto)
        self.horizontalLayout = QtWidgets.QHBoxLayout()
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.siyah_label = QtWidgets.QLabel(self.centralwidget)
        self.siyah_label.setMinimumSize(QtCore.QSize(200, 20))
        self.siyah_label.setMaximumSize(QtCore.QSize(200, 20))

```



```

self.siyah_label.setObjectName("siyah_label")
self.horizontalLayout.addWidget(self.siyah_label)
self.kirmizi_label = QtWidgets.QLabel(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum,
QtWidgets.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.kirmizi_label.sizePolicy().hasHeightForWidth())
self.kirmizi_label.setSizePolicy(sizePolicy)
self.kirmizi_label.setMinimumSize(QtCore.QSize(200, 20))
self.kirmizi_label.setObjectName("kirmizi_label")
self.horizontalLayout.addWidget(self.kirmizi_label)
self.sari_label = QtWidgets.QLabel(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum,
QtWidgets.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.sari_label.sizePolicy().hasHeightForWidth())
self.sari_label.setSizePolicy(sizePolicy)
self.sari_label.setMinimumSize(QtCore.QSize(200, 20))
self.sari_label.setObjectName("sari_label")
self.horizontalLayout.addWidget(self.sari_label)
self.yesil_abel = QtWidgets.QLabel(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum
, QtWidgets.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.yesil_abel.sizePolicy().hasHeightForWidth())
self.yesil_abel.setSizePolicy(sizePolicy)
self.yesil_abel.setMinimumSize(QtCore.QSize(200, 20))
self.yesil_abel.setObjectName("yesil_abel")
self.horizontalLayout.addWidget(self.yesil_abel)
self.mavi_label = QtWidgets.QLabel(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Maximum
, QtWidgets.QSizePolicy.Maximum)
sizePolicy.setHorizontalStretch(0)

```

```

sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.mavi_label.sizePolicy().hasHeightForWidth())
self.mavi_label.setSizePolicy(sizePolicy)
self.mavi_label.setMinimumSize(QtCore.QSize(200, 20))
self.mavi_label.setObjectName("mavi_label")
self.horizontalLayout.addWidget(self.mavi_label)
self.verticalLayout.addLayout(self.horizontalLayout)
self.formLayout.setLayout(0, QtWidgets.QFormLayout.LabelRole, self.verticalLayout)
self.verticalLayout_3 = QtWidgets.QVBoxLayout()
self.verticalLayout_3.setObjectName("verticalLayout_3")
self.comboBox = QtWidgets.QComboBox(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.comboBox.sizePolicy().hasHeightForWidth())
self.comboBox.setSizePolicy(sizePolicy)
self.comboBox.setMaximumSize(QtCore.QSize(250, 50))
self.comboBox.setObjectName("comboBox")
self.verticalLayout_3.addWidget(self.comboBox)
self.pushButton = QtWidgets.QPushButton(self.centralwidget)
sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Minimum)
sizePolicy.setHorizontalStretch(0)
sizePolicy.setVerticalStretch(0)
sizePolicy.setHeightForWidth(self.pushButton.sizePolicy().hasHeightForWidth())
self.pushButton.setSizePolicy(sizePolicy)
self.pushButton.setMinimumSize(QtCore.QSize(50, 20))
self.pushButton.setMaximumSize(QtCore.QSize(250, 50))
self.pushButton.setObjectName("pushButton")
self.verticalLayout_3.addWidget(self.pushButton)
spacerItem = QtWidgets.QSpacerItem(20, 40, QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Expanding)
self.verticalLayout_3.addItem(spacerItem)
self.formLayout.setLayout(0, QtWidgets.QFormLayout.FieldRole, self.verticalLayout_3)
self.horizontalLayout_2 = QtWidgets.QHBoxLayout()

```

```

self.horizontalLayout_2.setContentsMargins(0, 0, 0, 0)
self.horizontalLayout_2.setObjectName("horizontalLayout_2")
self.renk_belirlenmis_foto = QtWidgets.QLabel(self.centralwidget)
    sizePolicy = QtWidgets.QSizePolicy(QtWidgets.QSizePolicy.Minimum,
QtWidgets.QSizePolicy.Minimum)
    sizePolicy.setHorizontalStretch(0)
    sizePolicy.setVerticalStretch(0)

sizePolicy.setHeightForWidth(self.renk_belirlenmis_foto.sizePolicy().hasHeightForWidth())
    self.renk_belirlenmis_foto.setSizePolicy(sizePolicy)
    self.renk_belirlenmis_foto.setMinimumSize(QtCore.QSize(1200, 500))
    self.renk_belirlenmis_foto.setText("")
    self.renk_belirlenmis_foto.setObjectName("renk_belirlenmis_foto")
    self.horizontalLayout_2.addWidget(self.renk_belirlenmis_foto)
    self.formLayout.setLayout(1, QtWidgets.QFormLayout.LabelRole,
self.horizontalLayout_2)
    MainWindow.setCentralWidget(self.centralwidget)
    self.menubar = QtWidgets.QMenuBar(MainWindow)
    self.menubar.setGeometry(QtCore.QRect(0, 0, 1920, 21))
    self.menubar.setObjectName("menubar")
    MainWindow.setMenuBar(self.menubar)
    self.statusbar = QtWidgets.QStatusBar(MainWindow)
    self.statusbar.setObjectName("statusbar")
    MainWindow.setStatusBar(self.statusbar)
    self.retranslateUi(MainWindow)

    QtCore.QMetaObject.connectSlotsByName(MainWindow)
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "MainWindow"))
    self.siyah_label.setText(_translate("MainWindow", "Siyah"))
    self.kirmizi_label.setText(_translate("MainWindow", "Kırmızı"))
    self.sari_label.setText(_translate("MainWindow", "SARI"))
    self.yesil_abel.setText(_translate("MainWindow", "Yeşil"))
    self.mavi_label.setText(_translate("MainWindow", "Mavi"))
    self.pushButton.setText(_translate("MainWindow", "Dis Rengini Belirle"))

```