

TÜRKİYE CUMHURİYETİ  
YILDIZ TEKNİK ÜNİVERSİTESİ  
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



ÇOCUK X-RAY PANORAMİK DİŞ GÖRÜNTÜLERİNDEN  
DİŞLERİN BÖLÜTLENMESİ

18011060 – Aydın Kağan Türkoglu  
17011010 – Göktuğ Köksal

BİLGİSAYAR PROJESİ

Danışman  
Doç. Dr. Ayşe Betül OKTAY

Haziran, 2022



## **TEŞEKKÜR**

---

Daha önce makine öğrenmesi, model eğitimi ile obje tespiti ve sınıflandırması gibi konularda tecrübe sahibi olmadığımızı bilmesine rağmen bu proje'yi almamıza izin veren, tecrübeşiz olmamız dolayısı ile projeye en temel düzeyden başlayarak ilerlememizi sağlayan, Tüm süreç boyunca bizimle iletişimde kalarak ihtiyacımız olduğu her zaman bize yardım eden ve yol gösteren Doç. Dr. Ayşe Betül Oktay hocamıza en içten şekilde teşekkür etmeyi kendimize borç biliriz.

Aydın Kağan Türkoğlu  
Göktuğ Köksal

# İÇİNDEKİLER

---

<b>KISALTMA LİSTESİ</b>	<b>v</b>
<b>ŞEKİL LİSTESİ</b>	<b>vi</b>
<b>ÖZET</b>	<b>viii</b>
<b>ABSTRACT</b>	<b>ix</b>
<b>1 Giriş</b>	<b>1</b>
<b>2 Ön İnceleme</b>	<b>2</b>
<b>3 Fizibilite</b>	<b>4</b>
3.1 Teknik Fizibilite . . . . .	4
3.2 İş Gücü ve Zaman Fizibilitesi . . . . .	4
3.3 Yasal Fizibilite . . . . .	5
3.4 Ekonomik Fizibilite . . . . .	5
<b>4 Sistem Analizi</b>	<b>6</b>
<b>5 Sistem Tasarımı</b>	<b>7</b>
5.1 Veri seti Tasarımı . . . . .	7
5.2 Yazılım Tasarımı . . . . .	7
5.2.1 CNN . . . . .	8
5.2.2 R-CNN . . . . .	8
5.2.3 Faster R-CNN . . . . .	9
5.2.4 Mask R-CNN . . . . .	9
5.3 Kodlama . . . . .	10
<b>6 Uygulama</b>	<b>13</b>
<b>7 Deneysel Sonuçlar</b>	<b>17</b>
7.1 Farklı eşik değerleri ile alınan sonuçlar . . . . .	17
7.1.1 Eşik değeri değişiminin dışları kök alanları ile tespit edilmesine olan etkisi . . . . .	17

7.1.2 Eşik değeri değişiminin dişleri kireç tabakası ile tespit edilmesine olan etkisi . . . . .	19
7.2 Farklı iterasyon sayıları ile alınan sonuçlar . . . . .	20
7.2.1 İterasyon sayısındaki değişiminin dişleri kök alanları ile tespit edilmesine olan etkisi . . . . .	20
7.2.2 İterasyon sayısındaki değişiminin dişleri kireç tabakası ile tespit edilmesine olan etkisi . . . . .	21
7.3 Uç durum senaryosu: Yoğun diş telleri bulunan bir çocuğun x-ray görüntüsü ile yapılan deneme sonuçları . . . . .	22
7.3.1 diş tellerinin dişleri kök alanları ile tespit edilmesine olan etkisi	22
7.3.2 diş tellerinin dişleri kireç tabakası ile tespit edilmesine olan etkisi	23
<b>8 Performans Analizi</b>	<b>24</b>
8.1 Veri seti özellikleri . . . . .	24
8.2 Modelin başarı ölçütleri . . . . .	25
8.2.1 Başarı ölçütlerinin yorumlanması . . . . .	27
8.3 Farklı eşik değerleri ile alınan sonuçların değerlendirilmesi . . . . .	28
8.4 Farklı iterasyon sayıları ile alınan sonuçların değerlendirilmesi . . . . .	29
8.5 X-ray görüntüsünde Diş telleri bulunan bir örnek üzerinde modelin çalıştırılması ve oluşan performans değişiminin değerlendirilmesi . . .	29
<b>9 Sonuç</b>	<b>30</b>
<b>Referanslar</b>	<b>31</b>
<b>Özgeçmiş</b>	<b>33</b>

## KISALTMA LİSTESİ

---

CNN	Convolutional Neural Network
MASK R-CNN	MASK Regional Convolutional Neural Network
FAIR	Facebook AI Research Team
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
MS COCO	Microsoft Common Objects in Context JavaScript Object Notation
JSON	JavaScript Object Notation
RoI	Region of Interest
RoIAlign	Region of Interest Alignment
FCN	Fully Convolutional Network
ReLU	Rectified Linear Unit

## ŞEKİL LİSTESİ

---

Şekil 2.1 Palmer Notation[2] . . . . .	3
Şekil 3.1 Proje İş - Zaman çizelgesi . . . . .	4
Şekil 3.2 Ekonomik Fizibilite Tablosu . . . . .	5
Şekil 5.1 LabelMe uygulama görüntüsü . . . . .	7
Şekil 5.2 CNN katmanlarının uygulanması [6] . . . . .	8
Şekil 5.3 Mask R-CNN maskeleme aşamaları [10] . . . . .	9
Şekil 5.4 Gerekli kütüphaneleri dahil eden kod parçası . . . . .	10
Şekil 5.5 Verileri Detectron2 formatına uygun biçimde kaydetme . . . . .	10
Şekil 5.6 Gerekli sınıfların oluşturulduğu kod parçası . . . . .	11
Şekil 5.7 Eğitim için gerekli kütüphanelerin implantasyonu . . . . .	11
Şekil 5.8 Model oluşturmak için kullanılan argümanlar . . . . .	11
Şekil 5.9 COCO Instance Segmentation Baselines with Mask R-CNN[13] .	12
Şekil 5.10 eğitimi başlatan kod parçası . . . . .	12
Şekil 5.11 maskelenmiş panoramik dış örneği (1) . . . . .	12
Şekil 6.1 Mask R-CNN / accuracy tablosu . . . . .	13
Şekil 6.2 Total Loss tablosu . . . . .	13
Şekil 6.3 Mask R-CNN / false positive tablosu . . . . .	14
Şekil 6.4 Mask R-CNN / false negative tablosu . . . . .	14
Şekil 6.5 maskelenmiş panoramik dış örneği (2) . . . . .	14
Şekil 6.6 maskelenmiş panoramik dış örneği (3) . . . . .	15
Şekil 6.7 maskelenmiş panoramik dış örneği (4) . . . . .	15
Şekil 6.8 maskelenmiş panoramik dış örneği (5) . . . . .	16
Şekil 6.9 maskelenmiş panoramik dış örneği (6) . . . . .	16
Şekil 7.1 Eşik değeri = 0.4 . . . . .	17
Şekil 7.2 Eşik değeri = 0.1 . . . . .	18
Şekil 7.3 Eşik değeri = 0.9 . . . . .	18
Şekil 7.4 eşik değeri = 0.4 . . . . .	19
Şekil 7.5 Eşik değeri = 0.1 . . . . .	19
Şekil 7.6 Eşik değeri = 0.9 . . . . .	20
Şekil 7.7 iterasyon sayısı = 2000 . . . . .	20
Şekil 7.8 iterasyon sayısı = 200 . . . . .	21

Şekil 7.9 iterasyon sayısı = 2000 . . . . .	21
Şekil 7.10 iterasyon sayısı = 200 . . . . .	22
Şekil 7.11 Diş telleri bulunan örnek sonuçları 1 . . . . .	22
Şekil 7.12 Diş telleri bulunan örnek sonuçları 2 . . . . .	23
Şekil 8.1 Sınıfların sayısal dağılımı . . . . .	24
Şekil 8.2 dişleri kireç tabakası ile tespit edilmesi için oluşturulmuş modelin başarı metrikleri . . . . .	25
Şekil 8.3 dişleri kök alanları ile tespit edilmesi için oluşturulmuş modelin başarı metrikleri . . . . .	25
Şekil 8.4 dişleri kireç tabakası ile tespit edilmesi için oluşturulmuş modelin farklı sınıflar için ortalama kesinlik değerleri . . . . .	26
Şekil 8.5 dişleri kök alanları ile tespit edilmesi için oluşturulmuş modelin farklı sınıflar için ortalama kesinlik değerleri . . . . .	26
Şekil 8.6 Intersection over Union örnek şekil [14] . . . . .	27

## ÖZET

---

# ÇOCUK X-RAY PANORAMİK DİŞ GÖRÜNTÜLERİNDEN DİŞLERİN BÖLÜTLENMESİ

Aydın Kağan Türkoğlu

Göktuğ Köksal

Bilgisayar Mühendisliği Bölümü

Bilgisayar Projesi

Danışman: Doç. Dr. Ayşe Betül OKTAY

Bu projenin amacı, çocuk X-ray panoramik diş görüntülerinden elde edilen dişlerin bölütlenmesi ve görselleştirilmesidir. Bunu gerçekleştirebilmek için makine öğrenmesi algoritmalarına ve derin öğrenme tekniklerine ihtiyaç duyulmaktadır. Makine öğrenmesi alanında konuya alakalı birçok algoritma mevcuttur. Kullanılan algoritmaların projemize en uygun olanı Mask R-CNN olarak belirlenmiştir.

Bu alanda hazırlanmış başka projelere bakıldığından X-ray panoramik diş görüntülerinden diş bölütlenmesiyle alakalı birkaç proje olduğu gözlenmektedir ancak çocuk X-ray panoramik diş görüntülerinden diş bölütlenmesi yapan herhangi bir projeye rastlanılmamıştır.

Projemiz kapsamında kullandığımız iki adet veriseti bulunmaktadır. Bu verisetleri el ile işaretlenmiştir. Birbirinden farkları ise birinde dişler kök alanları ile birlikte işaretlenmiş öbüründe ise dişlerin sadece kireçlenmiş yerleri işaretlenmiştir.

Projenin sonucunda elde ettiğimiz model ile birlikte kullanıcı, gönderdiği çocuk X-ray panoramik diş görüntüsünü belirlediğimiz dört sınıfta sınıflandırılmış ve maskelenmiş olarak almaktadır. Oluşturulan sınıflar sırası ile "molar", "premolar" , "front" ve "hidden" olarak belirlenmiştir.

**Anahtar Kelimeler:** Makine öğrenmesi, derin öğrenme, CNN, Mask R-CNN, Detectron2, çocuk X-ray diş görüntüleri, MS COCO, obje tespiti, obje sınıflandırması

## **ABSTRACT**

---

# **Teeth Segmentation From Pediatric X-RAY Panoramic Dental Images**

Aydın Kağan Türkoğlu  
Göktuğ Köksal

Department of Computer Engineering  
Computer Project

Advisor: Assoc. Dr. Ayşe Betül Oktay

The purpose of this project is to segment and visualize teeth obtained from pediatric x-ray panoramic dental images. In order to achieve this, machine learning algorithms and deep learning techniques are needed. There are many algorithms about this subject in machine learning. Among the algorithms used, the most suitable one for our project was determined as Mask R-CNN.

When looking at the other projects in this field, it is observed that there are several projects related to tooth segmentation from x-ray panoramic tooth images but there is not a project that performs tooth segmentation from children's x-ray panoramic tooth images.

There are two data sets that we use within the scope of our project. These data sets are marked manually. The differences from each other are that in one, the teeth are marked with their root areas, and in the other, only the calcified parts of the teeth are marked.

Along with the model we obtained as a result of the project, the user receives the child x-ray panoramic tooth image that we have sent, classified and masked in four classes we have determined. The classes created were determined as "molar", "premolar", "front" and "hidden", respectively.

**Keywords:** Machine learning, Deep learning, CNN, Mask R-CNN, Detectron2, pediatric x-ray tooth images, MS COCO, object detection, object classification

# 1

## Giriş

---

Günümüzde tıp alanında bilgisayar teknolojileri oldukça önemli rol oynamaktadır. Yazılım alanındaki gelişmeler sayesinde hastalıkların izlenmesi, sınıflandırılması ve kontrol edilmesi oldukça kolaylaşmıştır. Özellikle son yıllarda kullanılan makine öğrenmesi, yapay zeka ve derin öğrenme teknikleri tıbbi alanda birçok projenin gerçekleşmesine olanak sağlamıştır. Bu projenin de amacı tıp alanına bilgisayar teknolojileri ile katkı sağlamakdır. Gelecekte robotik cerrahi alanında dişlerin bizim modelimiz sayesinde tespit edilmesi başarıya ulaşırsa dışçılere gerek duymadan birçok diş hastalığının tedavi edilmesinde cerrahi robotlar kullanılabilir. Tam anlamı ile bir yapay dişçi üretilেbilir.

Projede ana hedefimiz Mask R-CNN ile çocuk x-ray panoramik diş görüntülerindeki dişlerin bölütlenmesi ve görselleştirilmesidir. Çocuk x-ray panoramik diş görüntülerinde dişler birbirine girintili şekilde bulunduğuundan diş tespiti yapan birçok modelde hatalar oluşmaktadır. Biz de Çocuklar için özelleştirilmiş ve optimize edilmiş Mask R-CNN tabanlı bir model tasarlayarak bu hataları olabildiğince minimize etmeye çalışmaktayız.

Projede çocuk x-ray panoramik diş görüntüleri kullanılarak model eğitimi yapılmaktadır fakat açık kaynaklı görüntü bulma zorluğu sebebiyle veri setimiz 170 görüntü gibi düşük sayılabilen bir sayıdadır. Planlanan geliştirmelerde veri setinin azlığı gözetilerek hareket edilmektedir.

Son durumda elimizde iki farklı eğitim seti bulunmaktadır. Bunlardan birinde dişler kökleri ile birlikte işaretlenmiş öbüründe ise dişler kireçlenmenin olduğu bölgelerden işaretlenmiştir.

Projede hedeflenilen çıktılar X-ray panoramik diş görüntülerinin içinde bulunan dişler için her diş ayrı olarak maskelenerek ve farklı gruplara sınıflandırılacak şekilde görselleştirilmesidir.

## 2 Ön İnceleme

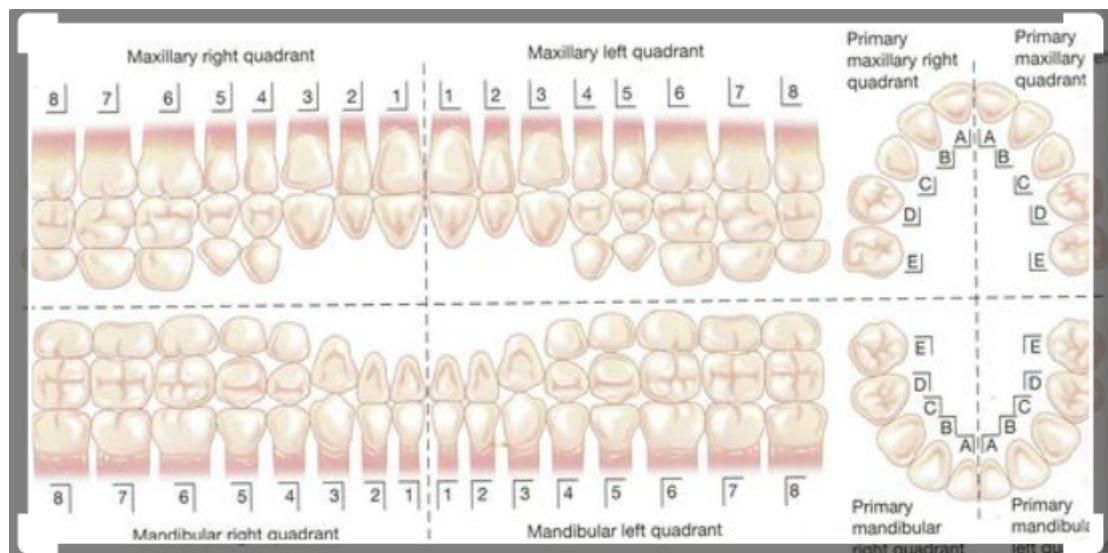
---

Bu projenin genel amacı Mask rcnn ile yetişkin diş X-ray panaromik görüntülerini için yapılan maskeleme işlemlerinin çocukların için optimize edilmiş ve güncellenmiş bir versiyonunu gerçekleştirmektir.

Günümüzde diş segmentasyonu yapan birçok Mask-RCNN modeli tabanlı çalışma bulunmaktadır fakat çocukların için özelleştirilmiş bir model bulunmamaktadır. Mevcut modeller yetişkinler için dişleri doğru anot etebilmektedir. Anotasyonu doğru yapılan dişleri makine öğrenmesi yardımıyla doğru bir şekilde eğitmekte ve doğru maskeleme yapılmaktadır fakat çocuk diş x-ray görüntülerinde 24 süt dişinin yanı sıra diş etinin içinden gelen yeni kalıcı dişlerin varlığı bu modellerin çocukların için doğru sonuç verememesine neden olmaktadır. Dişlerin çocuklarda çok yakın konumlanması ve bunun çocuk X-ray panaromik görüntülerinde iç içe gözükmesi de problemler arasındadır. Bundan dolayı birbirine yakın dişleri anot etmek zorlaşmaktadır. Anotasyonu doğru yapılmadığı için bu dişlerin maskelenmesi de yanlış sonuçlar ortaya çıkarmaktadır. Eğitime verdığımız çocuk diş x-ray panoromik görüntülerinde anotasyon işlemini elle yaptığımız için maskeleme işleminde bir sorun yaşanmamaktadır. Projede veri setinin azlığı sonucunda oluşan düşük doğruluk oranını iyileştirmek için dişleri 4 ana sınıfta sınıflandırılmasına karar verilmiştir. Bu sınıflar Front ( ön diş), Premolar, Molar ve Hidden ( gizli, çıkmamış diş) olarak belirlenmiştir. Bu sınıfların ihtiyaç halinde Diş hekimlerine ön bilgi sağlamaya yeterli olduğu yapılan sözlü geri bildirimler ile onaylanmıştır. Ayrıca projede dişlerin sınıflandırması yapılrken Palmer notasyonundan yardım alınmıştır.

Dişlerin sınıflandırılmasında Dünya çapında genel olarak Palmer notasyonu kullanılmaktadır [1]. Palmer notasyonunda görüntü diş hekiminin karşısından baktığı görüntüdür. Yani hastanın sağ tarafı çizelgedeki sağa ve sol tarafı da sola denk gelmektedir. Palmer notasyonunda bulunan “ $\text{U} \text{ L}$ ” sembollerini dişleri ağızda genel olarak 4 bölgeye ayırır. Bu 4 bölümde bulunan dişler 1'den 8'e kadar numaralandırılır. Örneğin, 1 $\text{U}$  şeklinde bir gösterim hastanın üst bölgede sağdan birinci dişini gösterirken 1 $\text{L}$  bu gösterim hastanın aşağı bölgede soldan üçüncü dişini gösterir.

Çocuklarda bulunan süt dişleri ise rakamlar yerine A'dan E'ye kadar olan harflerle belirtilmektedir.



Şekil 2.1 Palmer Notation[2]

# 3

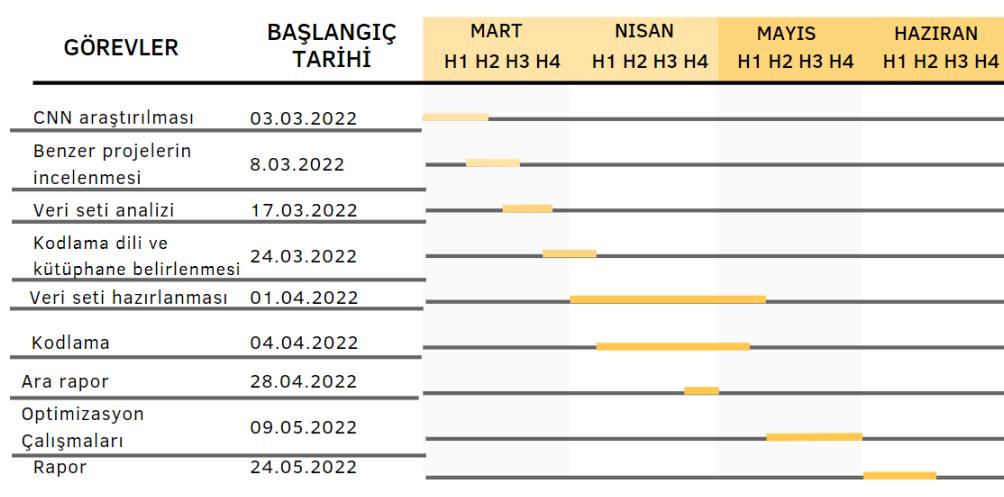
## Fizibilite

### 3.1 Teknik Fizibilite

Model eğitimi yüksek işlem gücü gerektiren bir süreç olduğu için işlem kapasitesi yüksek donanımların tercih edilmesi gereklidir. Daha düşük kapasiteli donanımların kullanılması mümkün olmakla beraber uzun süreler beklenmesi gerekeceği için süreci ciddi oranda yavaşlatmaktadır. Projede hem yüksek işlem kapasitesine sahip bir GPU bulunan bir bilgisayar hem de GPU ve TPU desteği sağlayan online servisler kullanılmıştır. Projede iki kişi kolektif olarak çalışıldığından yazılan kodları hızlı olarak paylaşabilmek için bulut ile dosya arşivleyebilen ve paylaşımabilen online servisler kullanılmıştır.

### 3.2 İş Gücü ve Zaman Fizibilitesi

2 kişi 4 ay sürede gerçekleşmiştir. Şekil 3.1'de görevlerin tamamlanması için harcanan zaman gösterilmiştir.



Şekil 3.1 Proje İş - Zaman çizelgesi

### **3.3 Yasal Fizibilite**

Yapılan proje herhangi bir yasa veya yönetmeliği ihlal etmemektedir. Kullanılan tüm kütüphaneler açık kaynaklıdır. Herhangi bir patent vb. koruyucu hak kullanılmamıştır. Projede kullanılan veri seti Ordu Üniversitesi'nden özel izinle alınmıştır.

### **3.4 Ekonomik Fizibilite**

Geliştirme ortamı olarak kullandığımız Google Colab uygulamasının ücretsiz sürümünü kullandığımızdan dolayı herhangi bir ücret ödenmemektedir.

Derin öğrenme için lazım olan Phyton kütüphaneleri ücretsiz kullanıma açık kütüphanelerdir.

Projede kullandığımız veri seti danışman tarafından sağlanmaktadır ve ücretsizdir.

Kişisel bilgisayarlarımızın donanımı projeyi gerçekleştirmek için yeterli düzeydedir.

Mask RCNN modelimizi Facebook'un geliştirmiş olduğu Detectron2 uygulaması ile eğitmekteyiz. Detectron2 uygulaması da tamamen ücretsizdir. Detaylı maliyetleri Şekil 3.2'de bulabilirsiniz

	<b>Kullanıcı/Adet</b>	<b>Birim Fiyat</b>	<b>Toplam Maliyet</b>
MSI GE72VR 7RF	1	13,500 TL	13,500 TL
Google Colab	2	0 TL	0 TL
I7 10700K/3070 Ti Desktop	1	19,999 TL	19,999 TL
<b>Toplam Maliyet</b>			<b>33,499 TL</b>

**Şekil 3.2 Ekonomik Fizibilite Tablosu**

## 4 Sistem Analizi

---

Uygulamanın genel amacı girdi olarak verilen çocuk panaromik x-ray diş görüntülerindeki dişlerin yüksek doğruluk oranında tespit edilip maskelenmesidir. Projede dişler 4 farklı sınıfta sınıflandırılmıştır. Bu sınıflar Front (ön diş), Premolar, Molar ve Hidden (gizli, çıkmamış diş) olarak belirlenmiştir. Projenin çıktılarında dişlerin doğru şekilde konumlarının tespit edilerek maskelenmesinin yanı sıra dişlerin doğru bir şekilde sınıflandırılmasının yapılması da hedeflenmektedir.

Projemizi gerçeklemede kullandığımız ana algoritma Mask RCNN algoritmasıdır. Mask R-CNN, FAIR ekibi tarafından geliştirilen nesne bölütleme algoritmasıdır. Algoritmanın amacı görüntüdeki belirli nesneleri tespit etmek ve her bir nesnenin üzerine ayrı bir maske oluşturmaktır.

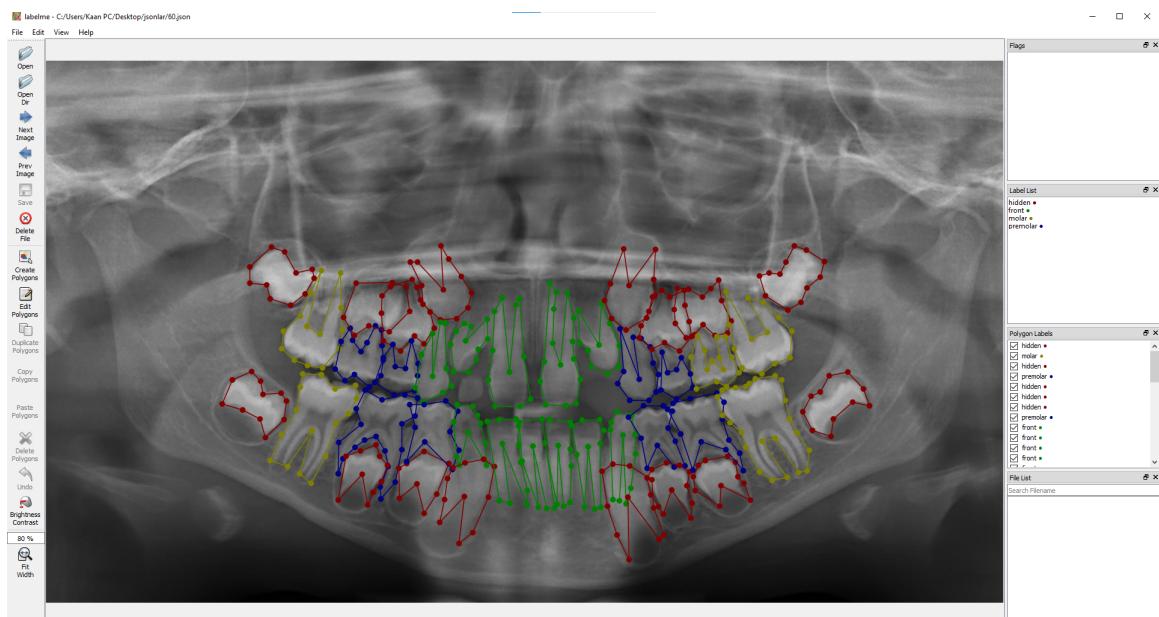
Elimizde hazır olarak bulunan işaretlenmiş resimler ile eğitim yaptığımızda çıkan sonuçlar bekłentimizi karşılamamıştır. Maskelerin doğruluk oranını artttırmak için veri seti içindeki çocuk panaromik x-ray diş görüntülerini kendi elimizle anote etme kararına varılmıştır. Projenin son halinde el ile iki eğitim seti oluşturulmuştur. Anote edilen iki eğitim setinden birinde dişler kökleri ile birlikte işaretlenmiş öbüründe ise dişler kireçlenmenin başladığı bölgeden işaretlenmiştir. Bu işaretleme işlemi yapılrken LabelMe uygulaması kullanılmıştır. LabelMe uygulaması sayesinde el ile anote edilen resimlerin MS COCO formatında JSON dosyalarına dönüştürülmesi sağlanmaktadır. Bu dosyalar resimlerle birlikte eğitime verilerek model oluşturulmaktadır.

Uygulamanın son ürünü olarak elimize geçen panaromik x-ray diş görüntüsündeki her bir diş ayrı ayrı maskelenmiş ve sınıflandırılmıştır.

# 5 Sistem Tasarımı

## 5.1 Veri seti Tasarımı

Daha önce de belirtildiği gibi projeye ilk başlanıldığından hazır işaretli çocuk x-ray panaromik diş görüntüleri üzerinde maskeleme çalışması denenmiştir. Buradan beklenilen sonuç alınamayınca elle işaretleme yöntemine başvurulmuştur. X-ray görüntülerindeki dişleri işaretleme ve sınıflandırma işlemleri için LabelMe uygulaması kullanılmıştır [3]. Şekil 5.1'de el ile işaretlenmiş bir resmin programdaki görüntüsü gösterilmektedir. Veri seti bu şekilde oluşturulmuştur .



Şekil 5.1 LabelMe uygulama görüntüsü

## 5.2 Yazılım Tasarımı

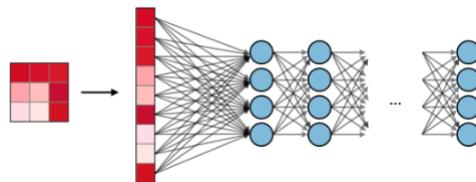
Projemizde Mask R-CNN algoritmasını kullanarak eğitim aşamasını gerçekleştirdiğimiz için projemizi açıklamaya başlamadan önce Mask R-CNN'i açıklamak gereklidir.

Mask R-CNN convolutional neural network tabanlı Faster R-CNN'in, FAIR ekibi tarafından geliştirmeler yapılmış bir versiyonudur [4].

Mask R-CNN ve Mask R-CNN öncesinde yapılmış algoritmaları incelersek:

### 5.2.1 CNN

CNN görüntüyü çeşitli katmanlar halinde inceleyip istediğimiz objenin özelliklerini tespit etme işlemlerinin tekrarlanarak son bilgiye ulaşılması işlemlerinin bütünüdür. CNN Convolution , pooling ve Fully-Connected Layer katmanlarının doğru sıra ve sayıda tekrarlanması ile oluşturulur [5]. Şekil 5.2'de CNN katmanlarından Fully-Connected Layer'ın uygulanması gösterilmiştir.



Şekil 5.2 CNN katmanlarının uygulanması [6]

- Convolution : konvolüsyon aradığımız objenin filtrelenmiş özelliklerini istenen nesnenin her bir parçası için benzerliğinin sorgulandığı ve benzer özellik haritasının çıkarıldığı katmandır. daha sonra ReLU ile negatif değere sahip bölgeler normalize edilir [5].
- Max Pooling: bu katmanda konvolüsyon sonrası elde ettiğimiz haritayı belirlenen boyuttaki alan içindeki en büyük değerin seçilmesi ile daha küçük boyuttaki haritalar oluşturulması sağlanmaktadır [7].
- Fully-Connected Layer : Daha önce elde ettiğimiz özellikler farklı ağırlıklara sahip nöronları oluşturmaktadır. Bu işlem şekil 5.2'nin ilk adımımda gösterilmektedir. Daha sonra Bu nöronlar istediğimiz sonuca yakın bir cevap elde edilene kadar çarpraz oylama ve geri yayılım aşamalarından geçirilir [8].

### 5.2.2 R-CNN

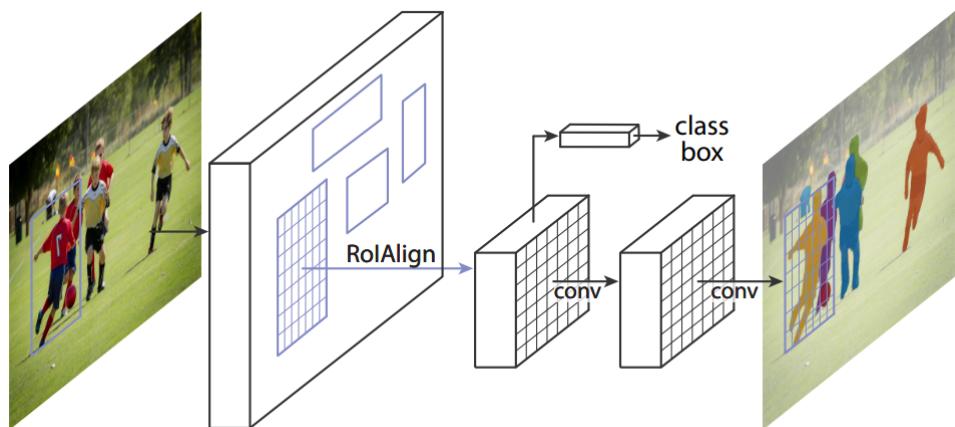
R-CNN CNN'in birden fazla objeyi tespit etmekte oldukça başarısız olması üzerine geliştirilmiş bir algoritmadır. isminden de anlaşılacağı üzere objeler resim içerisinde küçük bölgeler içerisinde ayrı olarak aranmaktadır. ilk önce resim iki bin civarında bölgeye ayrılır. Daha sonra ise her bir bölge için ayrı ayrı CNN işlemleri yapılır. Bu sayede farklı objelerin birbirini etkilemesine izin verilmez ve farklı objelerin tespiti doğru bir şekilde yapılmış olur.[9]

### 5.2.3 Faster R-CNN

Faster R-CNN R-CNN'den farklı olarak CNN işlemlerini ilk aşamada gerçekleştirir. Bölgelere ayırma işlemi daha sonra yapılır. Bu sayede bölge önerileri başlangıç aşamasında ve tek seferde belirlenmiş olur. CNN işlemi bir defa yapıldığı için R-CNN'e göre çok daha hızlı sonuç vermesi sağlanır [9].

### 5.2.4 Mask R-CNN

Mask R-CNN Faster R-CNN'den farklı olarak her bir aday obje için sınıf etiketi ve çevreleyici kutunun yanı sıra obje maskesi eklenir. Mask R-CNN bu nedenle doğal ve sezgiseldir fakat maskeleme işlemi, class ve çerçeveden ayrı olarak daha hassas bir çevrelemeye ihtiyaç duyar ve bağımsız bir adım olarak yapılır.[10] Bu sayede Mask R-CNN obje tespitinin yanı sıra piksel düzeyinde örnekleme yapabilmektedir .



**Şekil 5.3** Mask R-CNN maskeleme aşamaları [10]

Mask R-CNN'in iki temel aşaması bulunur. İlk aşamada CNN kullanılarak görüntü taranır ve obje bulunma olasılığı yüksek bölgeler (RoI) tespit edilir. Bu aşama Faster R-CNN ile benzer şekilde gerçekleştir fakat ikinci aşamada üstte belirtildiği gibi sınıf etiketi ve çevreleyici kutunun yanı sıra her bir RoI için obje maskelemesi yapılır [11].

Maskeleme işlemi aslında küçük çaplı FCN işlemleri gerçekleştirilir. Bu işlemler sadece seçili küçük alanlara uygulandığından işlem yükünü minimum seviyede artırmaktadır. Ayrıca RoI tespiti RoIPool yerine RoIAlign ile yapılmaktadır. Bu sayede her RoI için RoIPool'da oluşan çıkartılmış özelliklerin RoI ile hizalanmasında oluşan kayma sorunu ortadan kalkmış olur. Bu sorunu çözmek için bi-linear interpolation ile birlikte her özelliğin tam değeri bulunur [10].

RoIAlign ve sonrasında konvolüsyon işlemlerinin uygulanması şekil 5.3'de gösterilmiştir

### 5.3 Kodlama

Başlangıç olarak gerekli kütüphaneler dahil edilmektedir.

```
[6] import torch
     assert torch.__version__.startswith("1.8")
     import torchvision
     import cv2

▶ import os
import numpy as np
import json
import random
import matplotlib.pyplot as plt
%matplotlib inline
from detectron2.structures import BoxMode
from detectron2.data import DatasetCatalog, MetadataCatalog
```

Şekil 5.4 Gerekli kütüphaneleri dahil eden kod parçası

Projemizde kendi özel veri setimizi kullandığımız için veri setimizi detectron2'nin kendine özel formatına göre ayarlamamız gerekmektedir. Şekil 5.5'teki fonksiyon, Detectron2'nin kullanıcıları için bunu yerine getiren fonksiyonudur. Kendi resimlerimizin içeriğine göre genişlik ve yükseklik ayarı yapılmalıdır. Görüntüleri segmente ederken polygon formatını kullandığımız için “segmentation” kısmı buna uygun düzenlenmiştir.[12]

```
▶ def get_data_dicts(directory, classes):
    dataset_dicts = []
    for filename in [file for file in os.listdir(directory) if file.endswith('.json')]:
        json_file = os.path.join(directory, filename)
        with open(json_file) as f:
            img_anno = json.load(f)

        record = {}

        filename = os.path.join(directory, img_anno["imagePath"])

        record["file_name"] = filename
        record["height"] = 1127
        record["width"] = 1991

        annos = img_anno["shapes"]
        objs = []
        for anno in annos:
            px = [a[0] for a in anno['points']]
            py = [a[1] for a in anno['points']]
            poly = [(x, y) for x, y in zip(px, py)]
            poly = [p for x in poly for p in x]

            obj = {
                "bbox": [np.min(px), np.min(py), np.max(px), np.max(py)],
                "bbox_mode": BoxMode.XYXY_ABS,
                "category_id": classes.index(anno['label']),
                "segmentation": [poly],
                "iscrowd": 0
            }

            objs.append(obj)
        record["annotations"] = objs
        dataset_dicts.append(record)

    return dataset_dicts
```

Şekil 5.5 Verileri Detectron2 formatına uygun biçimde kaydetme

```

    classes=['molar','front','hidden','premolar']
    data_path='/content/'

    for d in ["train","test"]:
        DatasetCatalog.register(
            "my_ "+d,
            lambda d=d: get_data_dicts(data_path+d,classes)
        )
        MetadataCatalog.get("my_ "+d).set(thing_classes=classes)
microcontroller_metadata = MetadataCatalog.get("my_train")

```

**Şekil 5.6** Gerekli sınıfların oluşturulduğu kod parçası

```

[10] from detectron2 import model_zoo
     from detectron2.engine import DefaultTrainer,DefaultPredictor
     from detectron2.config import get_cfg
     from detectron2.utils.visualizer import ColorMode,Visualizer
     from matplotlib.patches import Rectangle

```

**Şekil 5.7** Eğitim için gerekli kütüphanelerin implantasyonu

```

    cfg=get_cfg() #config dosyasını oluşturur
    cfg.merge_from_file(model_zoo.get_config_file("COCO-InstanceSegmentation/mask_rcnn_R_50_C4_3x.yaml"))
    cfg.DATASETS.TRAIN= ("my_train",) #train verilerini confige kaydetme
    cfg.DATASETS.TEST=()
    cfg.DATALOADER.NUM_WORKERS= 4
    cfg.MODEL.WEIGHTS=model_zoo.get_checkpoint_url("COCO-InstanceSegmentation/mask_rcnn_R_50_C4_3x.yaml")
    cfg.SOLVER.IMS_PER_BATCH=2 #batch size
    cfg.SOLVER.BASE_LR=0.001 #learning rate
    cfg.SOLVER.GAMMA=0.05 #learning rate çarpanı
    cfg.SOLVER.STEPS=[500]
    cfg.TEST.EVAL_PERIOD=200 #modeli değerlendiremek için adım sayısı
    cfg.SOLVER.MAX_ITER=2000 #iterasyon sayısı
    cfg.MODEL.ROI_HEADS.NUM_CLASSES=4 #sınıf sayısı

```

**Şekil 5.8** Model oluşturmak için kullanılan argümanlar

Şekil 5.8' deki NUM-workers kısmında kaç sub-process kullanacağımızın sayısını vermektedir. IMS-PER-BATCH batch size kaç olacağını belirler. BASE-LR learning rate belirlememizi sağlar. GAMMA learning rate çarpanıdır. TEST.EVAL-PERIOD modeli değerlendirirken belirdiğimiz kullanılacak adım sayısıdır. MAX-ITER iterasyon sayısı belirdiğimiz kısımdır. MODEL.WEIGHTS kısmında ise Mask R-CNN algoritması için gerekli olan ağırlıklar çekilmektedir. Projede Mask R-CNN R50-C4 baseline kullanılmıştır. Bunların karşılaştırılması şekil 5.9'de gösterilmiştir.

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	mask AP	model id	download
R50-C4	1x	0.584	0.110	5.2	36.8	32.2	137259246	<a href="#">model   metrics</a>
R50-DC5	1x	0.471	0.076	6.5	38.3	34.2	137260150	<a href="#">model   metrics</a>
R50-FPN	1x	0.261	0.043	3.4	38.6	35.2	137260431	<a href="#">model   metrics</a>
R50-C4	3x	0.575	0.111	5.2	39.8	34.4	137849525	<a href="#">model   metrics</a>
R50-DC5	3x	0.470	0.076	6.5	40.0	35.9	137849551	<a href="#">model   metrics</a>
R50-FPN	3x	0.261	0.043	3.4	41.0	37.2	137849600	<a href="#">model   metrics</a>
R101-C4	3x	0.652	0.145	6.3	42.6	36.7	138363239	<a href="#">model   metrics</a>
R101-DC5	3x	0.545	0.092	7.6	41.9	37.3	138363294	<a href="#">model   metrics</a>
R101-FPN	3x	0.340	0.056	4.6	42.9	38.6	138205316	<a href="#">model   metrics</a>
X101-FPN	3x	0.690	0.103	7.2	44.3	39.5	139653917	<a href="#">model   metrics</a>

Şekil 5.9 COCO Instance Segmentation Baselines with Mask R-CNN[13]

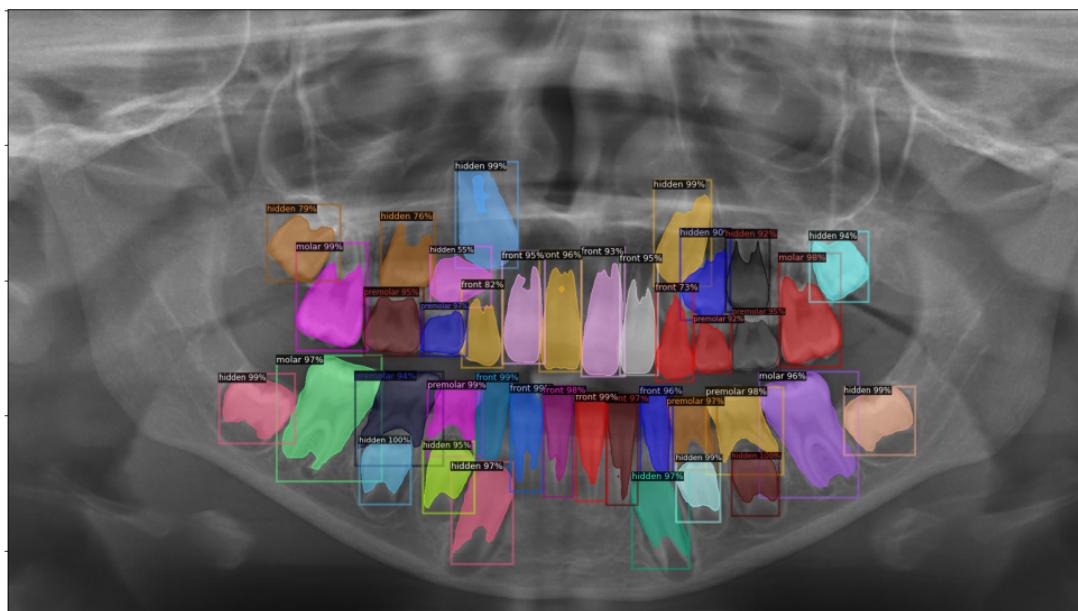
Bu aşamalardan sonra şekil 5.10 da eğitim başlatılmış model oluşturulmuştur.

```
os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)

trainer.train()
```

Şekil 5.10 eğitimi başlatan kod parçası

Oluşturulan örnek bir maskeleme işlemi şekil 5.11'da gösterilmiştir.

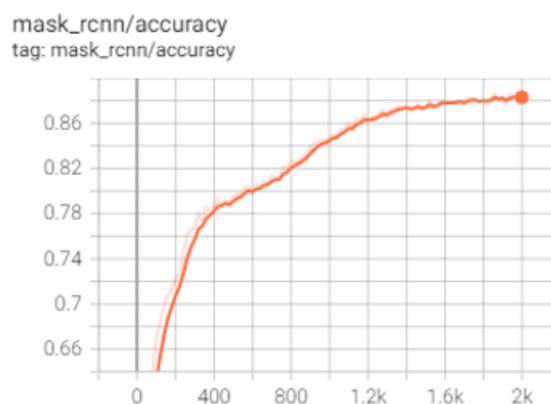


Şekil 5.11 maskelenmiş panoramik dış örneği (1)

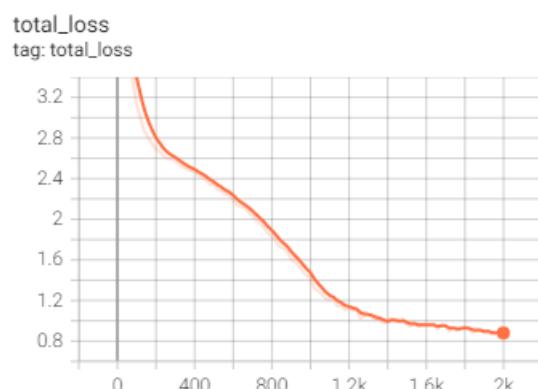
# 6

## Uygulama

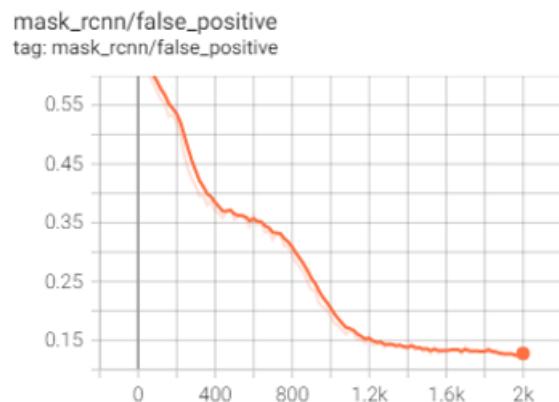
Projede eğitim işlemi iki farklı eğitim veri seti ile gerçekleştirilmiştir. Birinci eğitim seti dişlerin kireç tabakasını bulmaya odaklanmakta iken, ikinci eğitim setinde dişleri kök alanları ile tespit etmek hedeflenmektedir. Yapılan eğitim sonucunda ortaya çıkan model ile ilgili gerekli tablolar aşağıda verilmiştir.



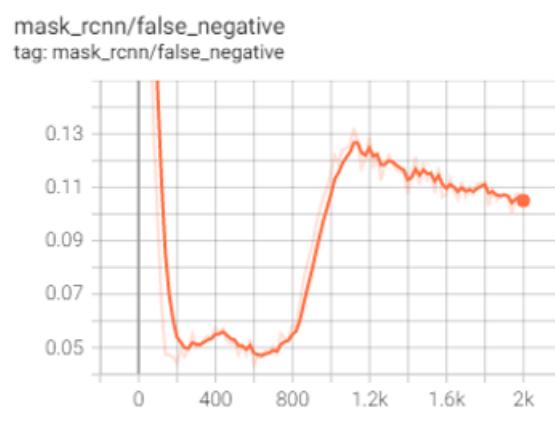
Şekil 6.1 Mask R-CNN / accuracy tablosu



Şekil 6.2 Total Loss tablosu

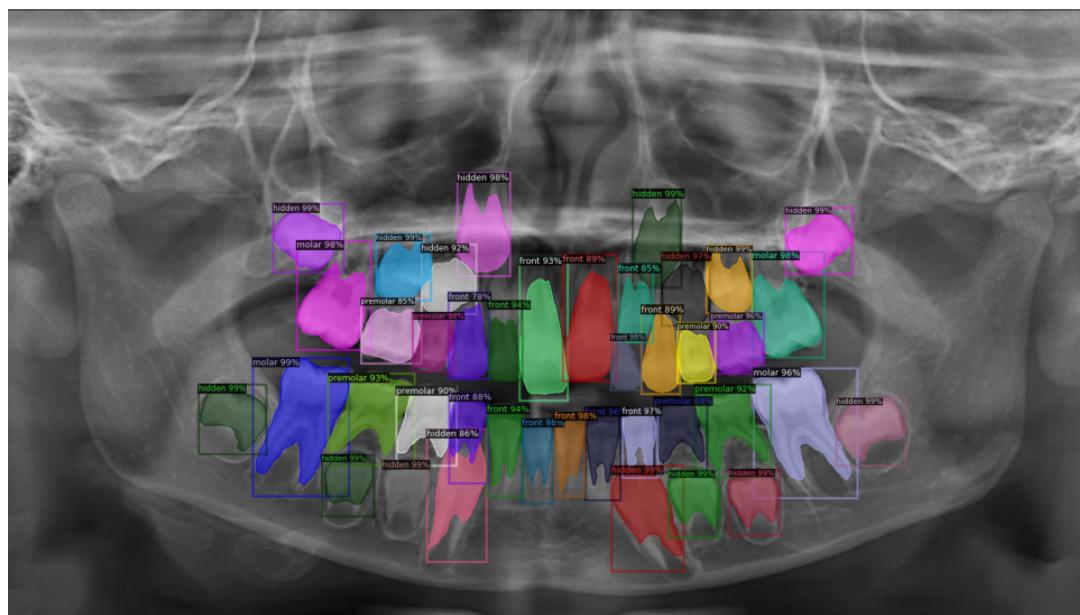


**Şekil 6.3** Mask R-CNN / false positive tablosu

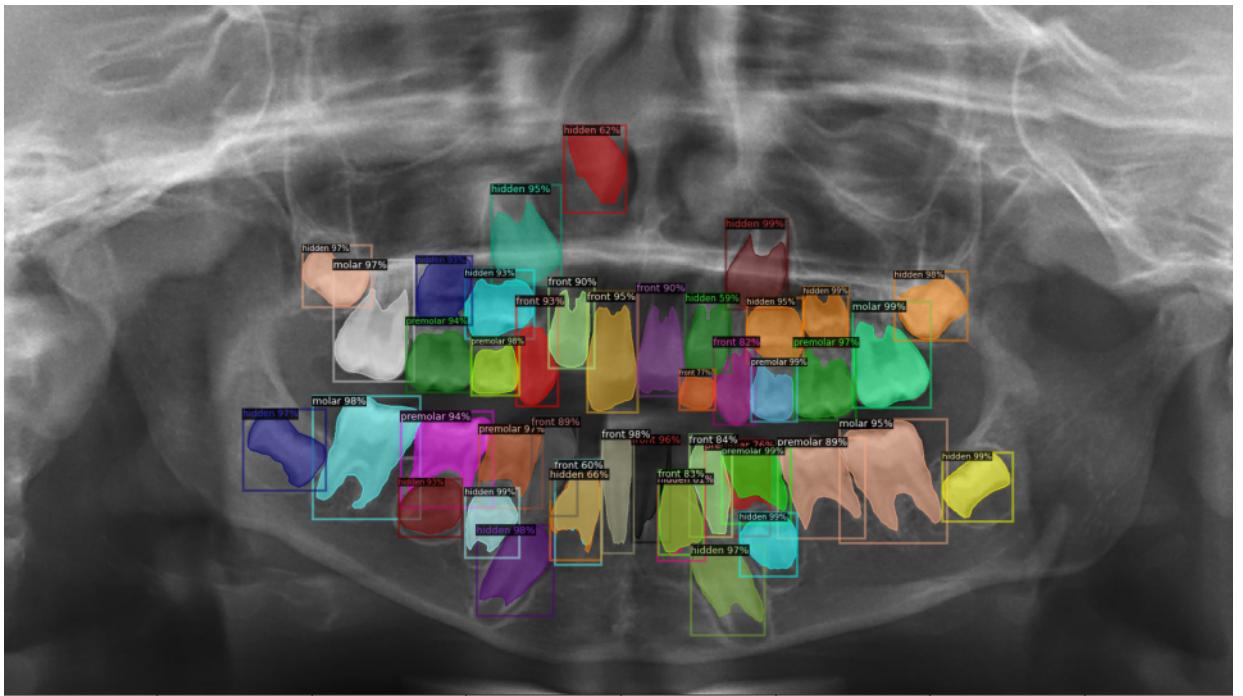


**Şekil 6.4** Mask R-CNN / false negative tablosu

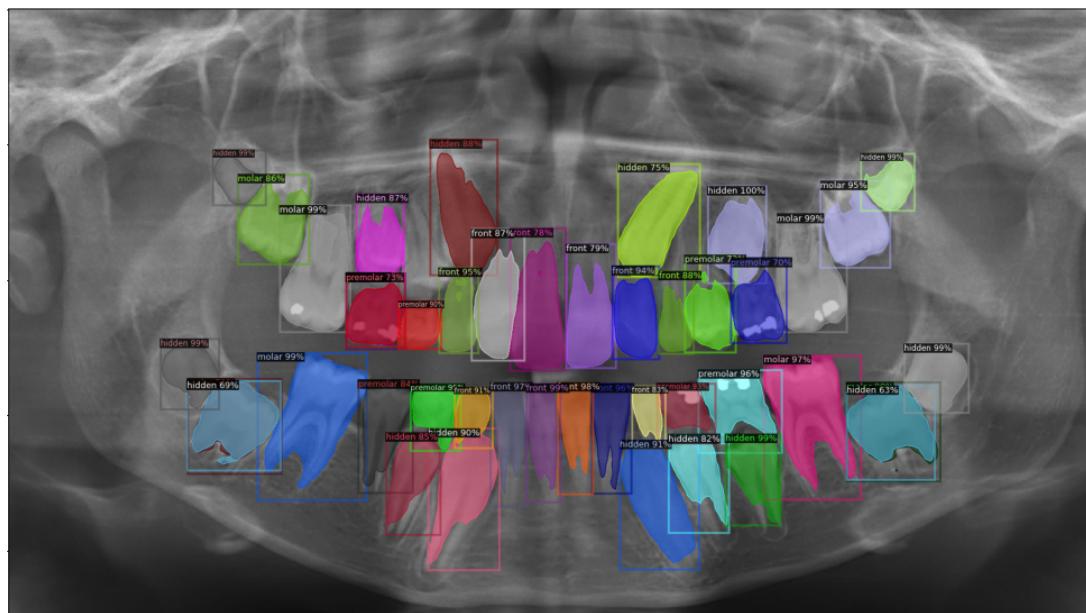
Modelimize gönderdiğimiz test resimlerinin sonuçları aşağıda gösterilmiştir.



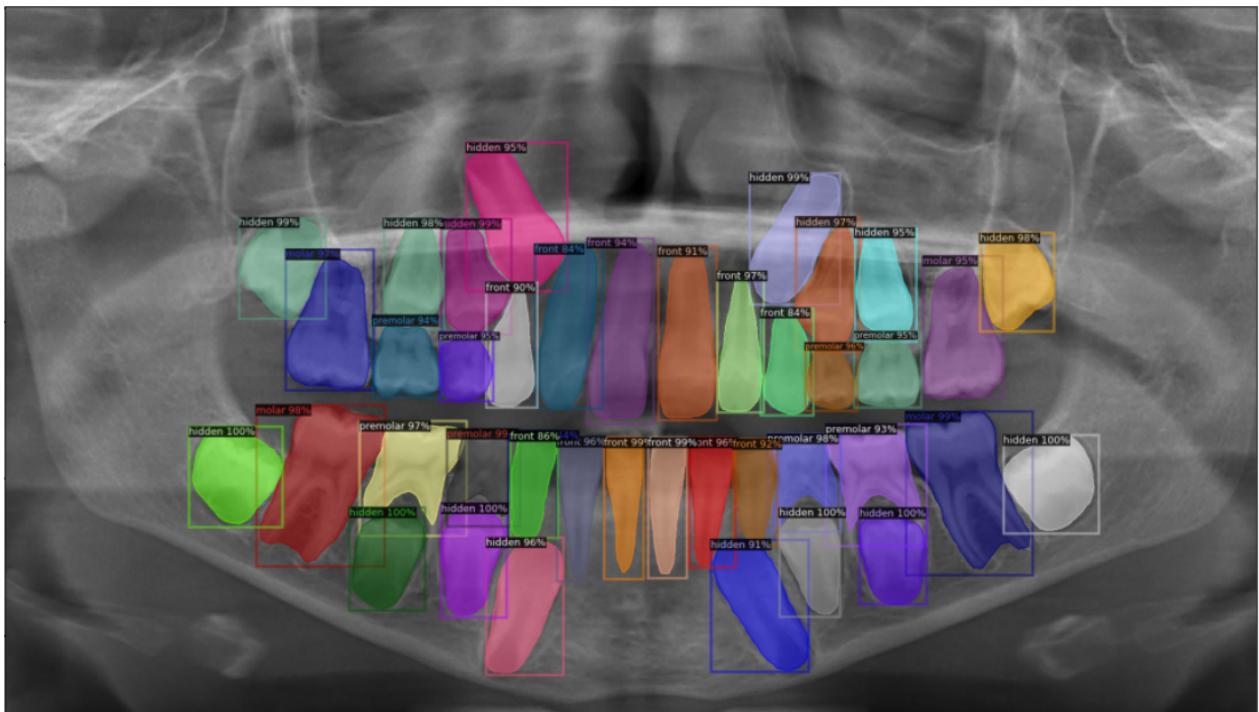
**Şekil 6.5** maskelenmiş panoramik dış örneği (2)



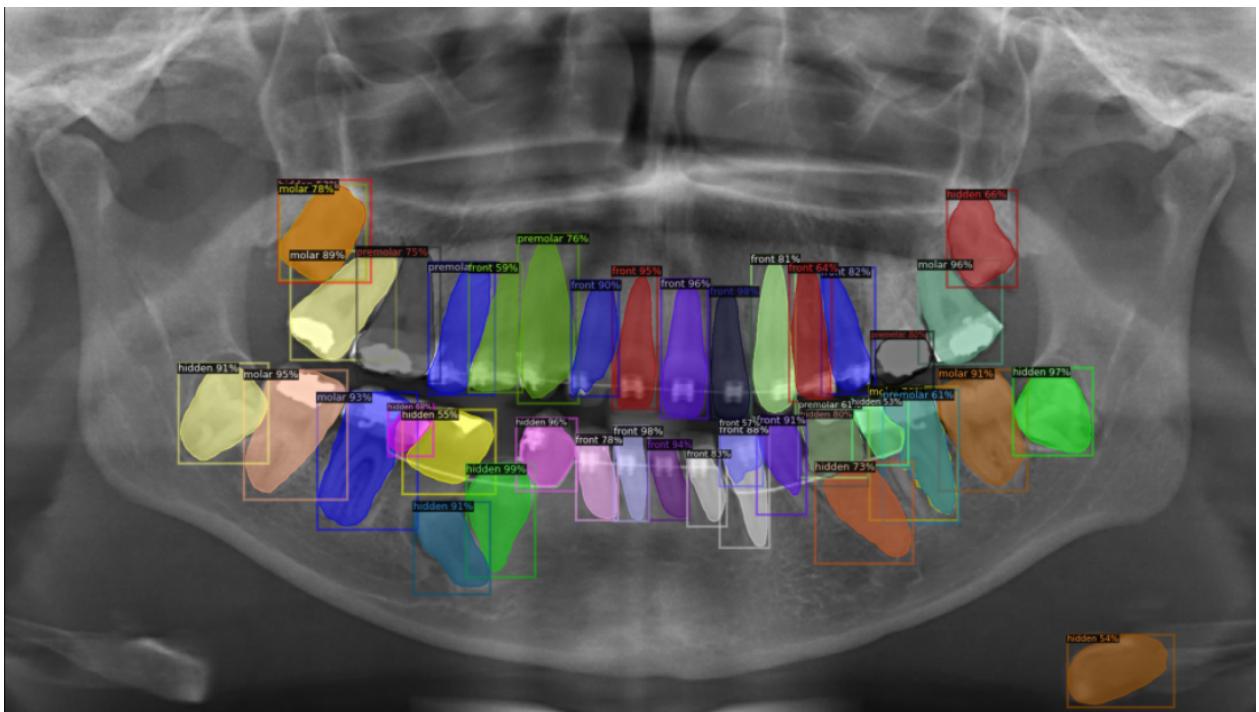
Şekil 6.6 maskelenmiş panoramik diş örneği (3)



Şekil 6.7 maskelenmiş panoramik diş örneği (4)



**Şekil 6.8** maskelenmiş panoramik dış örneği (5)



**Şekil 6.9** maskelenmiş panoramik dış örneği (6)

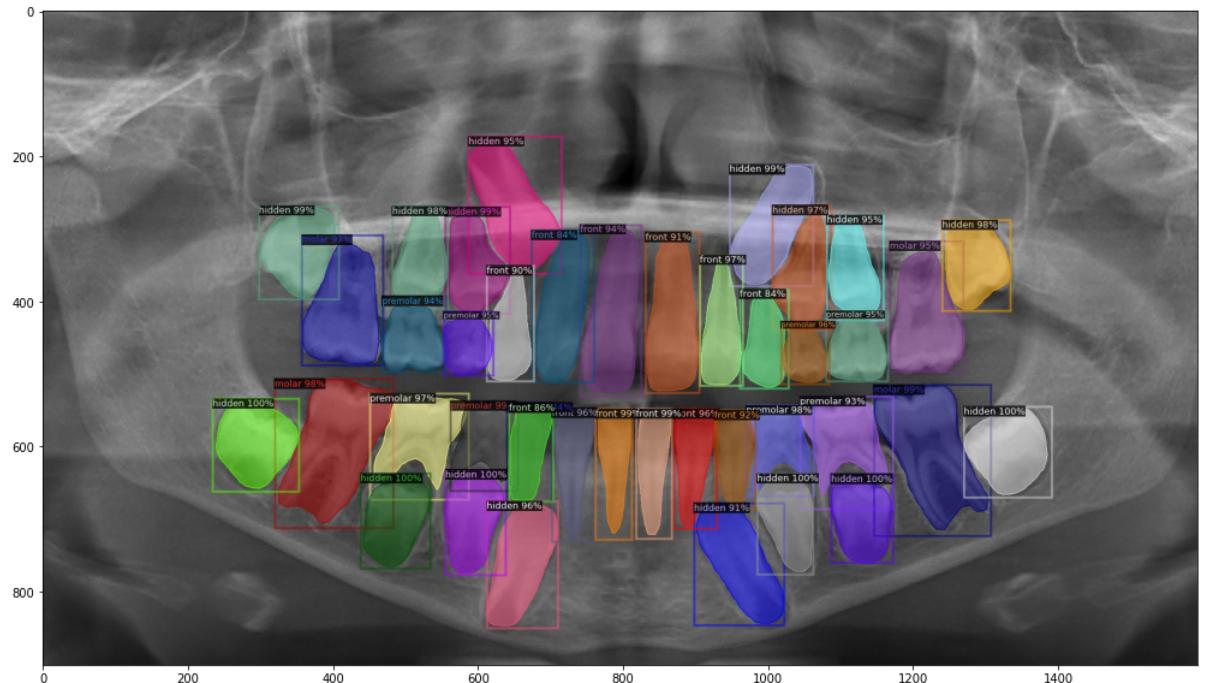
7

Projede modelin başarısını artırma amacıyla train öncesinde ve sonrasında bazı parametreler ile denemeler yapılarak en yüksek doğruluk oranı tespit edilmeye çalışılmaktadır. Eğitim işlemi oldukça uzun süren bir işlem olduğu için 3 farklı parametre değişikliğinin örnekleri gösterilecektir. Burada örneği gösterilmemiş bir çok parametre ile farklı denemeler proje süreci boyunca denenmiştir.

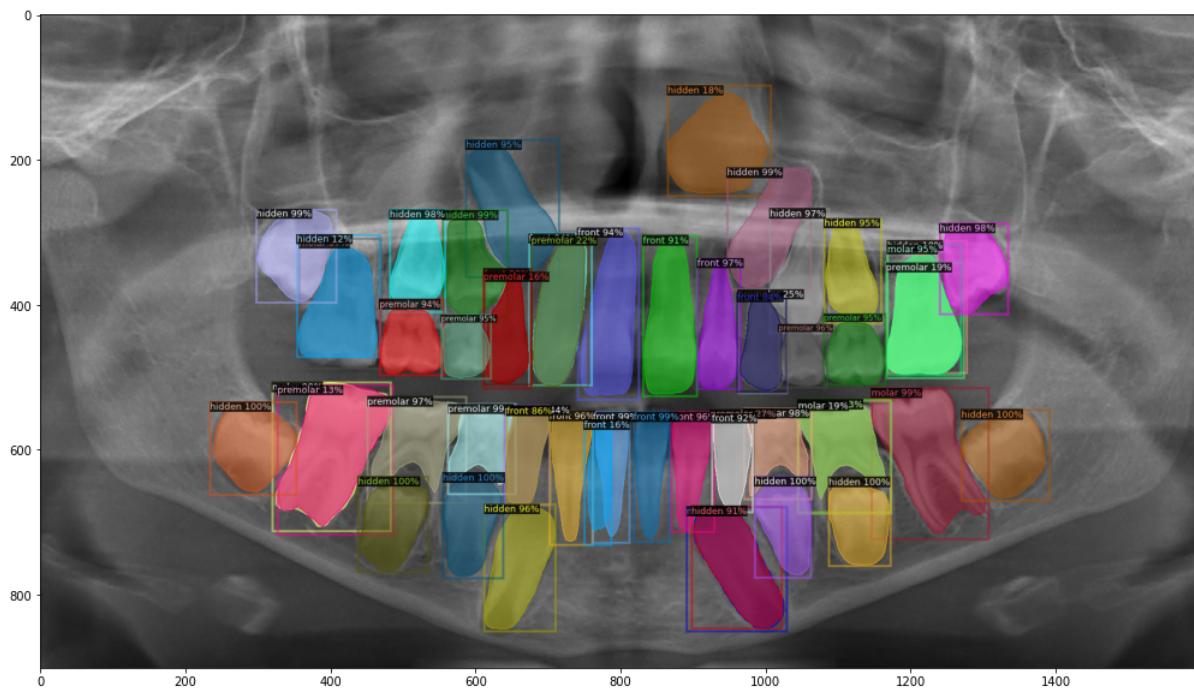
Proje kapsamında 2 farklı eğitim seti ile birbirinden bağımsız sonuçlar alındığı için denemeler, her iki durum için ayrı olarak ele alınmaktadır.

## 7.1 Farklı eşik değerleri ile alınan sonuçlar

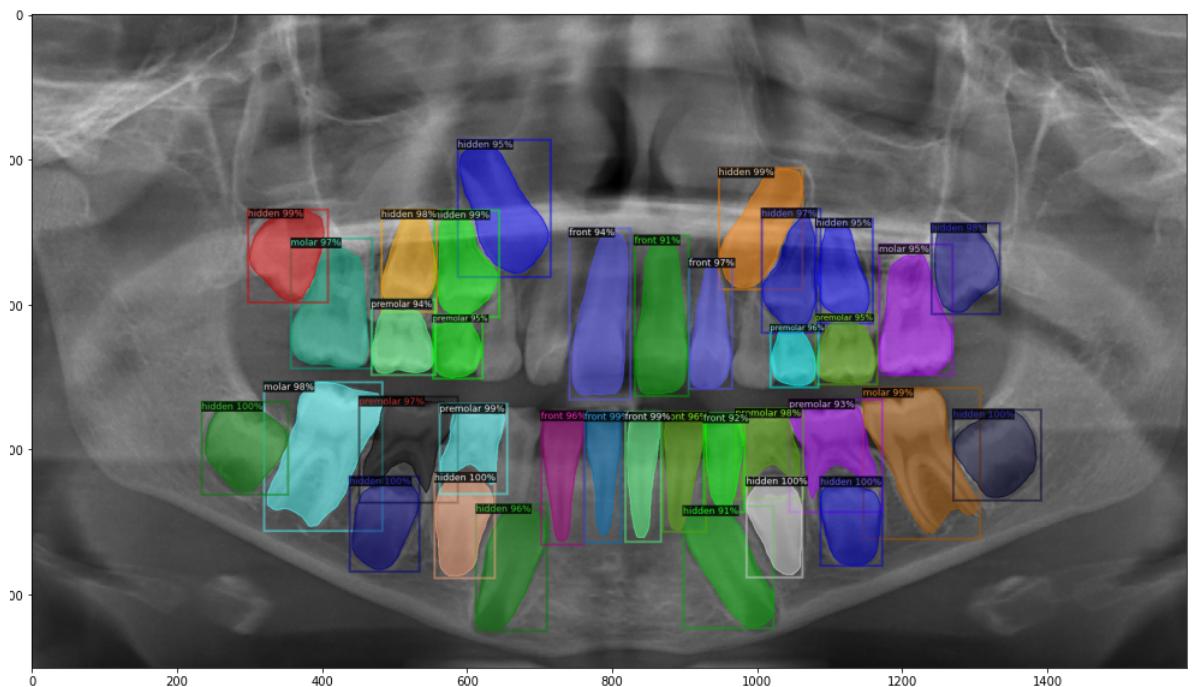
#### 7.1.1 Eşik değeri değişiminin dışları kök alanları ile tespit edilmesine olan etkisi



**Şekil 7.1** Eşik değeri = 0.4

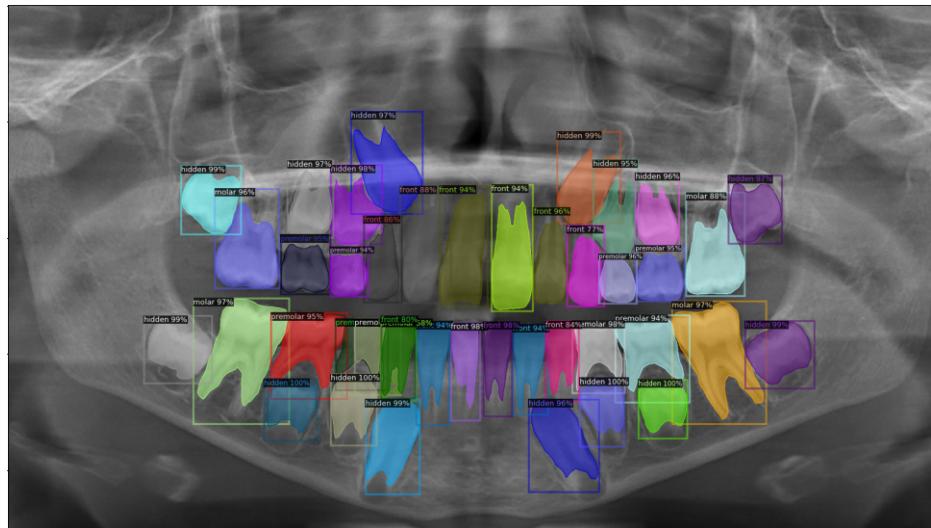


**Şekil 7.2** Eşik değeri = 0.1

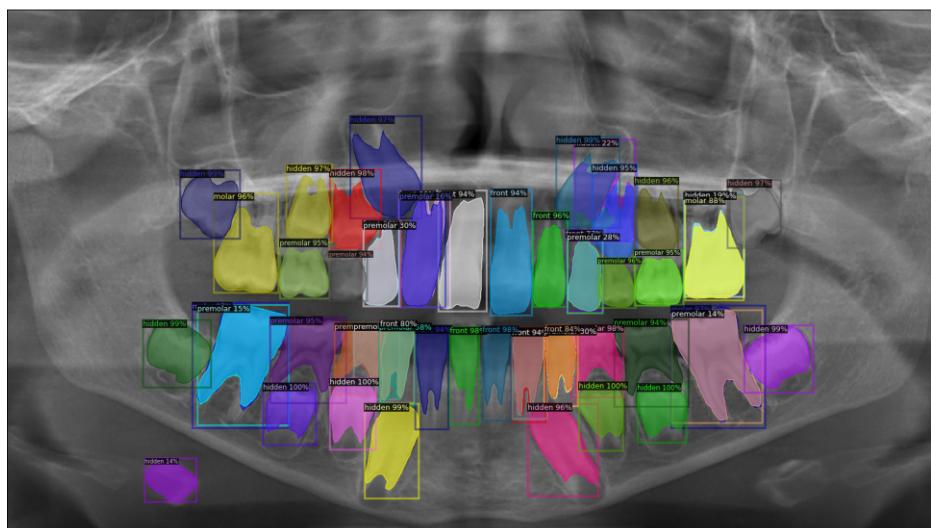


**Şekil 7.3** Eşik değeri = 0.9

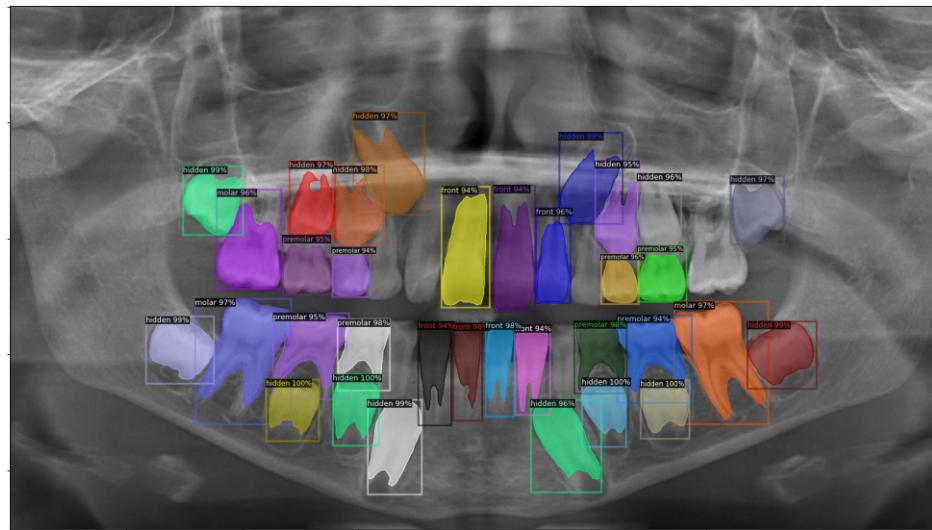
#### 7.1.2 Eşik değeri değişiminin dışları kireç tabakası ile tespit edilmesine olan etkisi



**Şekil 7.4** eşik değeri = 0.4



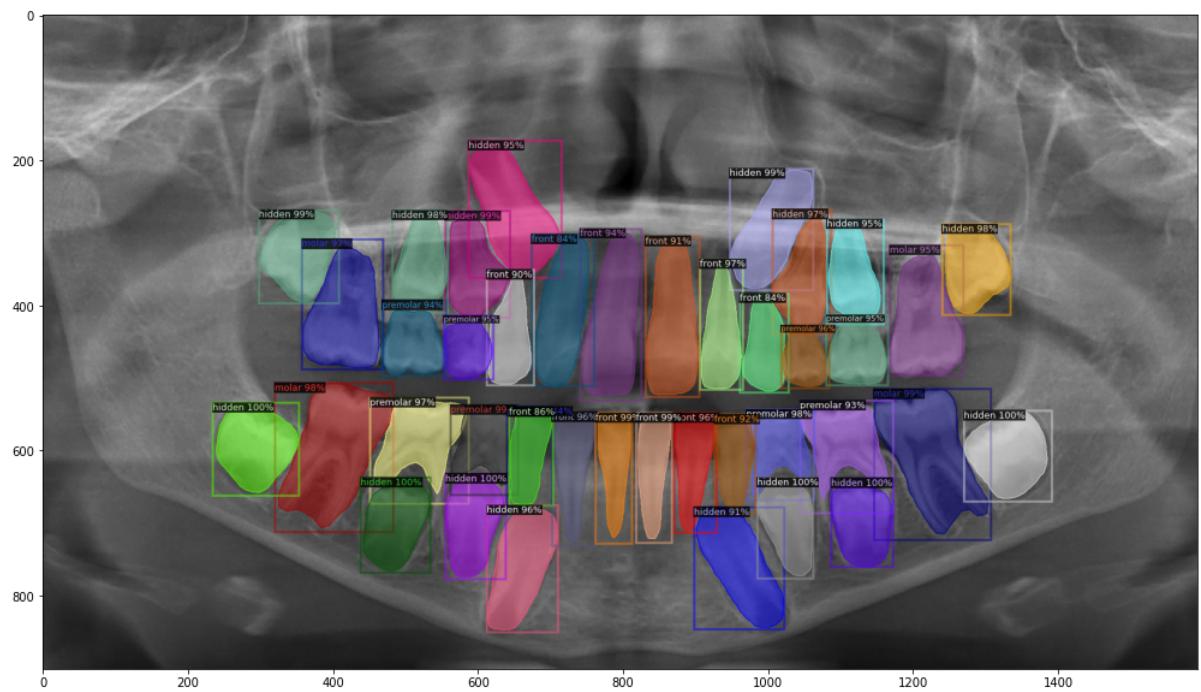
**Şekil 7.5 Eşik değeri = 0.1**



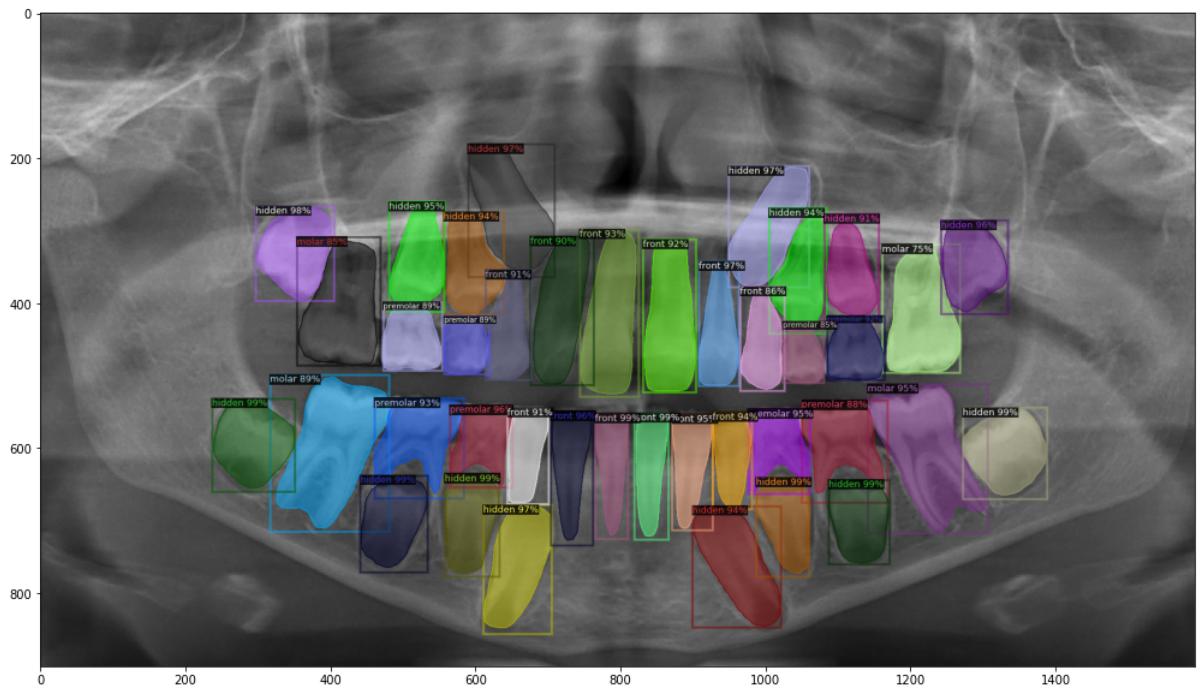
**Şekil 7.6** Eşik değeri = 0.9

## 7.2 Farklı iterasyon sayıları ile alınan sonuçlar

### 7.2.1 İterasyon sayısındaki değişiminin dışları kök alanları ile tespit edilmesine olan etkisi

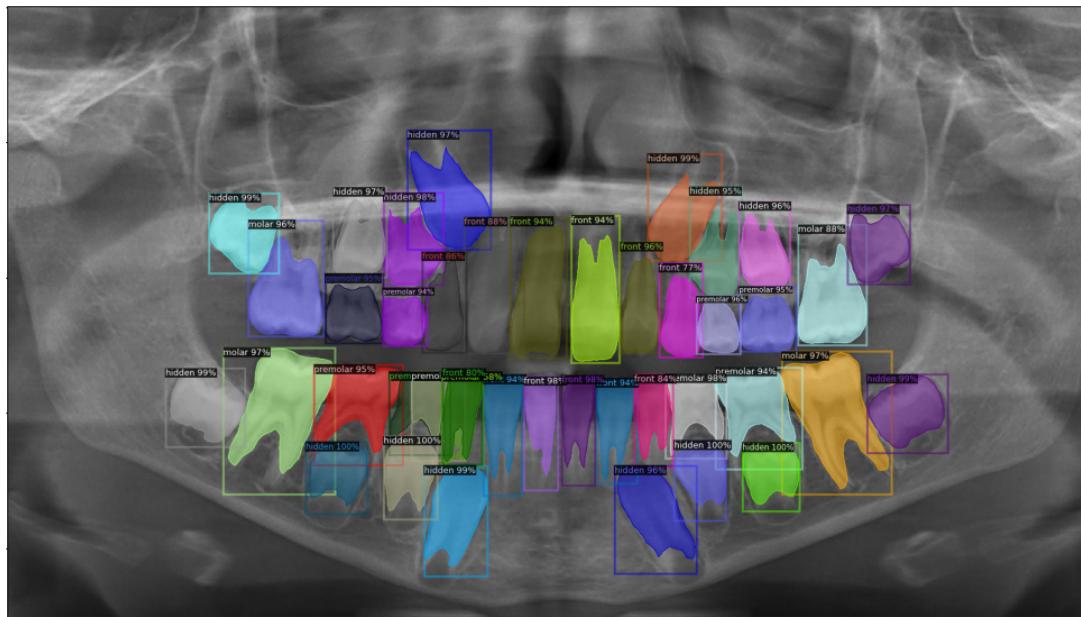


**Şekil 7.7 iterasyon sayısı = 2000**

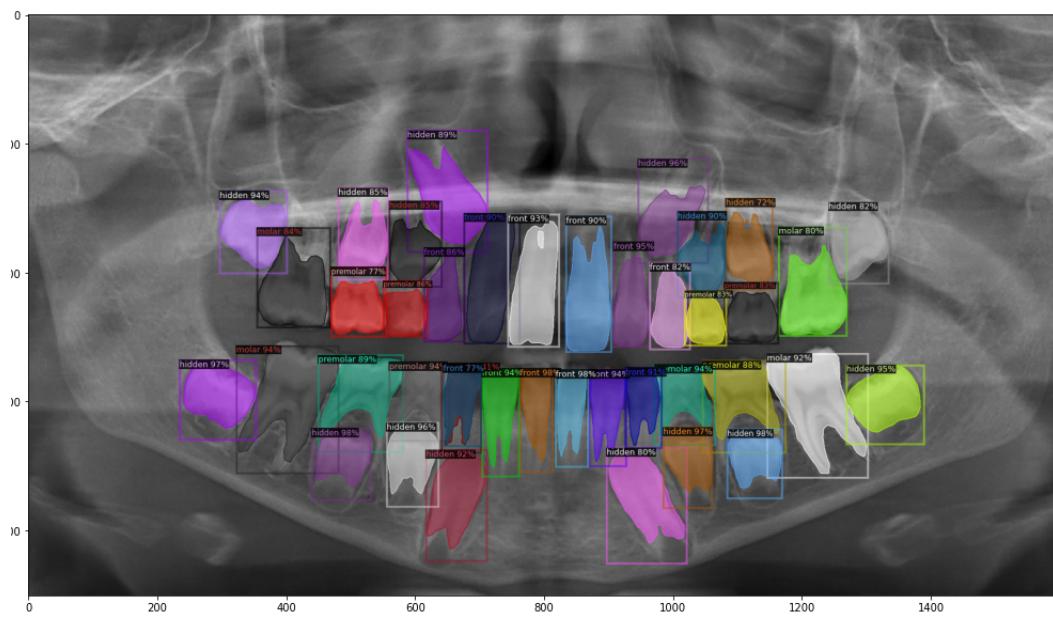


Şekil 7.8 iterasyon sayısı = 200

### 7.2.2 Iterasyon sayısındaki değişiminin dişleri kireç tabakası ile tespit edilmesine olan etkisi



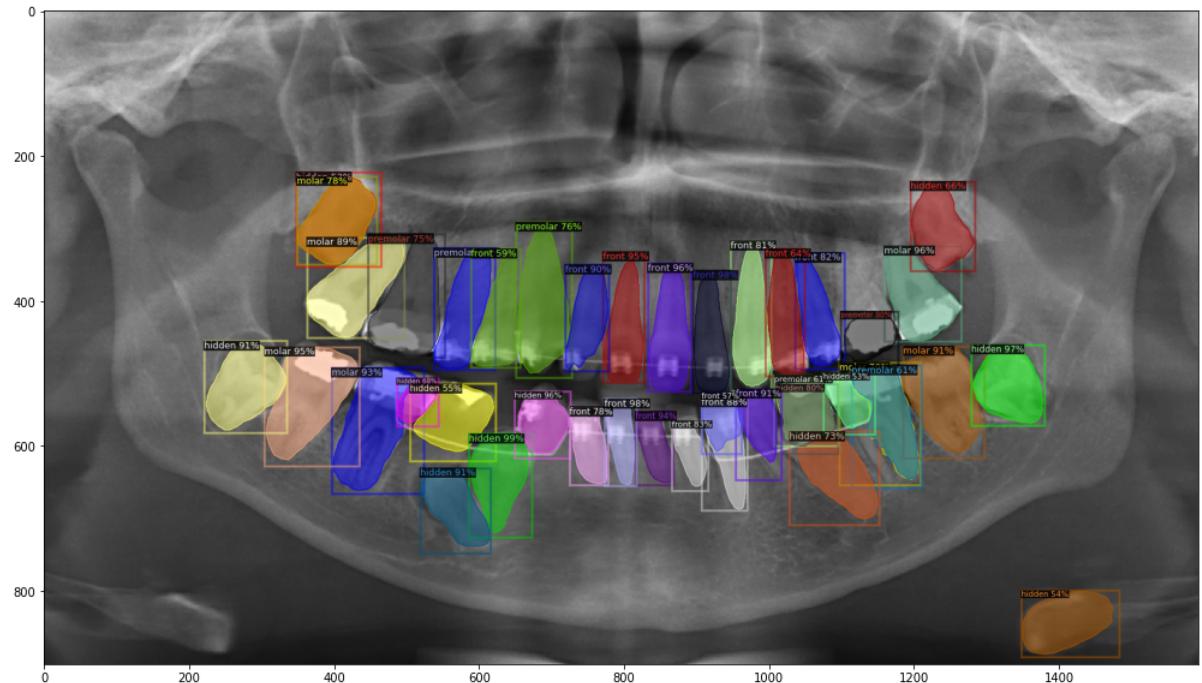
Şekil 7.9 iterasyon sayısı = 2000



**Şekil 7.10 iterasyon sayısı = 200**

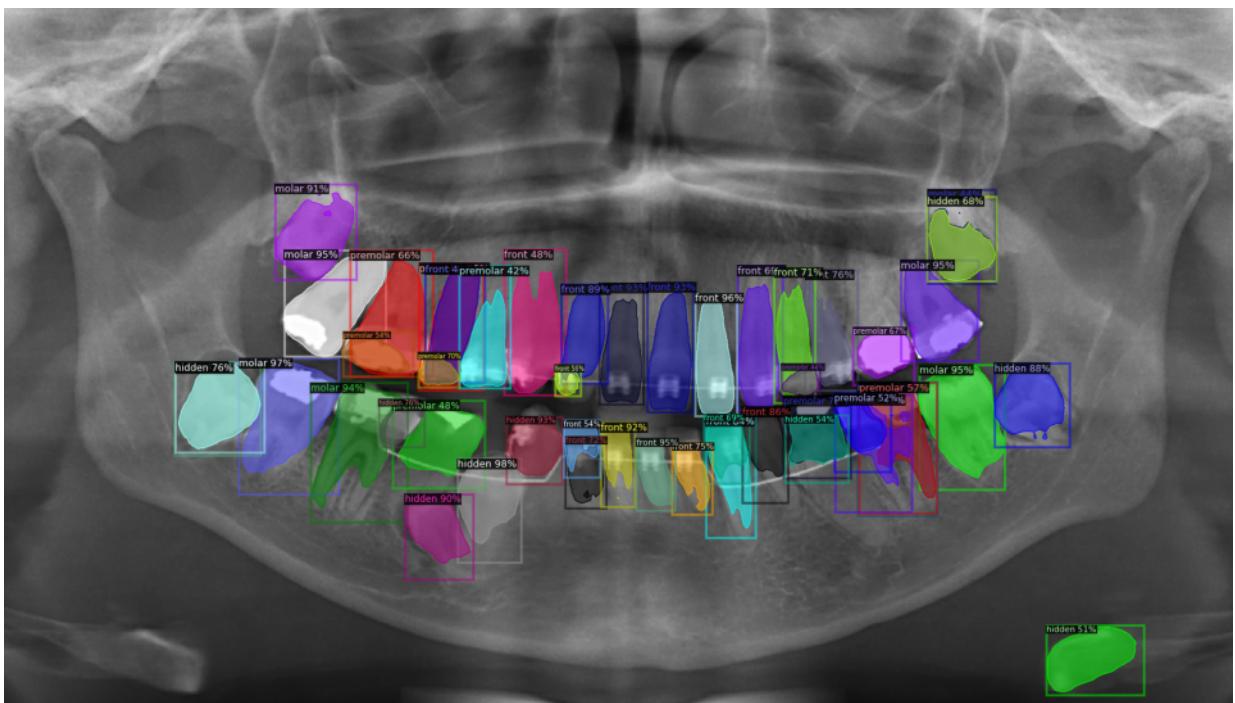
### 7.3 Uç durum senaryosu: Yoğun diş telleri bulunan bir çocuğun x-ray görüntüsü ile yapılan deneme sonuçları

#### 7.3.1 diş tellerinin dişleri kök alanları ile tespit edilmesine olan etkisi



**Şekil 7.11 Diş telleri bulunan örnek sonuçları 1**

### 7.3.2 diş tellerinin dişleri kireç tabakası ile tespit edilmesine olan etkisi



Şekil 7.12 Diş telleri bulunan örnek sonuçları 2

# 8

## Performans Analizi

### 8.1 Veri seti özellikleri

Veri seti toplam 170 adet çocuk x-ray panoramik diş görüntüsü içermektedir. Bu resimlerin 67 tanesi eğitim amaçlı kullanılırken geri kalan 103 tanesi test amaçlı kullanılmaktadır. Resim başına ortalama 37.5 adet diş bulunmaktadır. dişler 4 farklı sınıfa ayrılmıştır. Bu Sınıflar Front (ön), Pre-molar (molar öncesi), Molar ve Hidden (gizli, gömülü) dişlerden oluşmaktadır. Sınıfların veri setindeki dağılımını Şekil 8.1'de gösterilmektedir.

CLASS NAME	TRAIN	TEST	TOTAL
MOLAR	317	590	907
PRE MOLAR	515	725	1240
FRONT	761	1040	1801
HIDDEN	1006	1227	2233
TOTAL	2599	3537	6136

Şekil 8.1 Sınıfların sayısal dağılımı

Tabloda kullanılan İngilizce ifadelerin Türkçe karşılıkları: CLASS NAME : sınıf isimleri, TRAIN : eğitim işlemi, TEST: test işlemi, TOTAL: toplam.

Sınıfların kapsadığı diş türleri ise;

- Molar sınıfı: Büyük azı dişleri.

- Pre molar sınıfı: küçük azı dişleri.
- Front sınıfı: köpek dişleri ve kesici dişleri.
- Hidden sınıfı: gömülü olan dişlerin tamamı

olarak belirlenmiştir.

## 8.2 Modelin başarı ölçütleri

Şekil 8.5 ve 8.2'de Modelin test aşamasındaki başarısını ölçmek için kullanılan değerlendirme kriterlerindeki başarısı gösterilmektedir.

IoU	Area	AP	AR
0.50:0.95	all	0.501	0.600
0.50	all	0.812	0.600
0.75	all	0.512	0.600
0.50:0.95	small	0.000	0.000
0.50:0.95	medium	0.428	0.452
0.50:0.95	large	0.448	0.625

Şekil 8.2 dişleri kireç tabakası ile tespit edilmesi için oluşturulmuş modelin başarı metrikleri

IoU	Area	AP	AR
0.50:0.95	all	0.501	0.533
0.50	all	0.838	0.533
0.75	all	0.536	0,533
0.50:0.95	small	0.000	0.000
0.50:0.95	medium	0.415	0.430
0.50:0.95	large	0.461	0.662

Şekil 8.3 dişleri kök alanları ile tespit edilmesi için oluşturulmuş modelin başarı metrikleri

Category	AP
Front	<b>54.491</b>
Premolar	<b>40.432</b>
Molar	<b>55.118</b>
Hidden	<b>50.377</b>

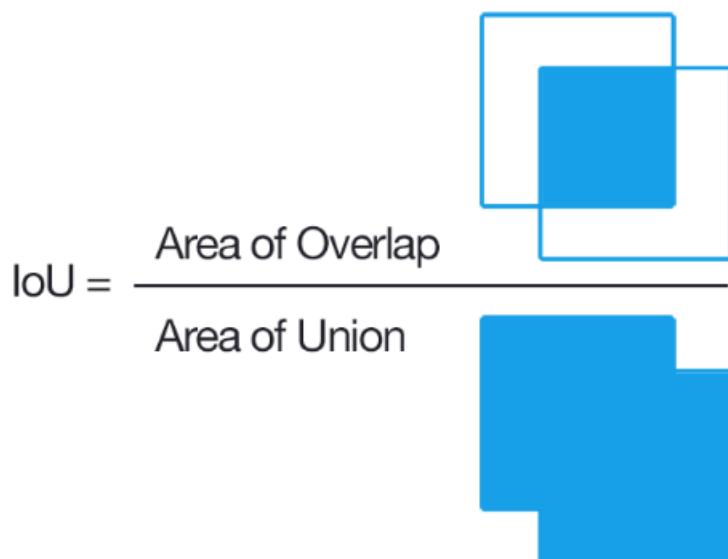
**Şekil 8.4** dişleri kireç tabakası ile tespit edilmesi için oluşturulmuş modelin farklı sınıflar için ortalama kesinlik değerleri

Category	AP
Front	<b>50.454</b>
Premolar	<b>43.835</b>
Molar	<b>60.931</b>
Hidden	<b>45.348</b>

**Şekil 8.5** dişleri kök alanları ile tespit edilmesi için oluşturulmuş modelin farklı sınıflar için ortalama kesinlik değerleri

Tabloların daha iyi anlaşılmasında kullanılan metrikleri açıklamak gerekmektedir.

IoU (Intersection over Union): Modelin tahmin ettiği alanın gerçekte olan alan ile olabildiğince yakın olması istenmektedir. IoU tahmin edilen alan ile gerçek alanın kesişim bölgesinin birleşim bölgesine oranı ile bulunur. Bu değer maskleme işleminin ne kadar doğru yapıldığını tespit etmek için kullanılır. IoU değeri verilen eşik değerinden büyük ise maskleme doğru (true pozitive, gerçek pozitif), küçük ise maskleme yanlış (false pozitive, yanlış pozitif) kabul edilmektedir. IoU kesişim ve birleşim alanları şekil 8.6'te gösterilmektedir [14].



**Şekil 8.6** Intersection over Union örnek şekil [14]

Recall: Duyarlılık değeridir. Modelin doğru olarak tahmin ettiği alanların gerçekte olan tüm doğru alanlara oranı ile bulunur.

Precision: Kesinlik değeridir. Modelin doğru olarak tahmin ettiği alanların ne kadarlık kısmının gerçekten doğru olduğunu bulmamızı sağlar. gerçekten doğru olarak bulunan tahminlerin tüm doğru olduğu varsayılan tahminlere oranı ile bulunmaktadır.

Area: İşaretlenmek istenen nesnenin/alanın piksel alanını tarif etmektedir. 32x32 pikselden küçük alanlar small, 32x32 ile 96x96 piksel arasındaki alanlar medium, 96x96 pikselden büyük alanlar large'dır.

AP: 0.5 ile 0.95 arasındaki farklı eşik değerleri ile elde edilen kesinlik değerlerinin ortalaması ile elde edilmektedir.

AP50: Eşik değerinin 0.5 olduğu durumdaki ortalama kesinlik değeridir

AP75: Eşik değerinin 0.75 olduğu durumdaki ortalama kesinlik değeridir

APs / APm / API / ARs / ARm / ARI: Area değerlerine göre Farklı boyutlardaki dışların ortalama kesinlik ve Duyarlılık değerlerini ifade etmektedir.

### 8.2.1 Başarı ölçütlerinin yorumlanması

İki modelde de benzer oranda artışlar ve azalışlar gözlemlendiğinden ilk önce modeller ortak olarak değerlendirilecektir. Daha sonra iki model arasındaki farklar değerlendirilecektir.

Farklı IoU eşik değerleri ile alınan sonuçlar incelendiğinde eşik değerinin 0.5 (yüzde 50) olduğu durumda 0.8 üstü kesinlik değerine sahip olduğu ancak eşik değeri yükseldikçe başarının düşüğü görülmektedir. Çocuk dişleri düzensiz ve birbirlerinden oldukça farklı şekillerde olduğu için doğru tahmin edilen dişler bile belli bir IoU değerinin altında kalmaktadır.

farklı area değerleri ile ortalama kesinliğin değişimi incelendiğinde small area'nın 0 olduğu görülmektedir. Bunun sebebi küçük kabul edilen boyutların altında hiçbir diş bulunmamasıdır. Medium ve Large için incelersek arada çok küçük farklar gözlemlenmektedir. Bu yüzden dişin büyüklüğünün yapılan tahminlerin doğruluk oranına etkisinin az olduğu tespit edilmiştir.

Area'nın duyarlılığı etkisinin ise çok daha yüksek olduğu gözlemlenmiştir. Bu sonuçlar doğrultusunda dişin boyutu küçüldükçe Model tarafından tespit edilmesinin zorlaştığı tespit edilmiştir.

Sınıf düzeyinde incelersek Molar en rahat doğru tahmin sınıfıdır. Bunu diğerlerinden büyük ve daha farklı bir şekli olduğu için Molar sınıfı tahmin edildiği çoğu zaman doğru çıkmaktadır. Ancak moların Kesinliğinin bir sebebi de modelin Molar olması gereken dişlerin premolar olarak tahmin etmeye daha yatkın olmasıdır. Bu yüzden premolar tahminlerinin çoğu yanlış iken yanlış molar tahminleri daha azdır. Front ve hidden sınıfları ise ortalama başarı oranlarına sahiptirler. front ve hidden hem sayıca en fazla olan sınıf olduğundan hem de bazı garanti yerlerde bulunduklarından başarıları belli bir değer aralığının içinde kalmaktadır

İki modeli kıyaslarsak kireç bulan model front ve hiddenları daha iyi bulabiliyorken kök alanları dahil tüm dişi bulan model molarları daha yüksek doğruluk oranı ile bulabilmektedir. Bunun muhtemel sebebi front ve hidden dişlerde kireç kısmın tüm dişe oranının molar'a kıyasla daha yüksek olması olabilir.

### 8.3 Farklı eşik değerleri ile alınan sonuçların değerlendirilmesi

Verilen eşik değeri modellerin başarısında büyük rol oynamaktadır. düşük eşik değerlerinde şekil 7.5'de görüldüğü gibi gerçekte diş bulunmayan bazı yerlerde diş bulma eğilimi artmakta iken yüksek eşik değerlerinde ise 7.3'de görülebileceği üzere bazı dişleri bulamama sorunu başlamaktadır. yapılan denemelerde en iyi eşik değerinin 0.5 olduğu görülmüştür.

#### **8.4 Farklı iterasyon sayıları ile alınan sonuçların değerlendirilmesi**

Şekil 7.10'de ve şekil 7.9'de görülebileceği üzere iterasyon sayısının düşüklüğü dış bulamama gibi bir soruna neden olmamaktadır ancak modelin bulunan dişler için kendinden emin olma skoru (confidence score) yüzde yirmi oranında düşüş yaşamaktadır. Bu deney sonucunda 200 iterasyonun bile büyük oranda yeterli olduğu ve iterasyon sayısının başarıyı logaritmik olarak arttırdığı gözlemlenmiştir. Maksimum başarı elde edilmek isteniyorsa yüksek iterasyonlar gereklidir ancak belirli bir sayının üstünde iterasyon büyük oranda vakit kaybı olacaktır.

#### **8.5 X-ray görüntüsünde Diş telleri bulunan bir örnek üzerinde modelin çalıştırılması ve oluşan performans değişiminin değerlendirilmesi**

Bu deneyde proje geliştirilirken gözetilmemiş bir istisna durum ele alınmıştır. Diş telleri veri setindeki çoğu resimde bulunmamaktadır ancak nadir de olsa karşılaşılmaktadır. Diş telleri Modeller üzerinde dişlerin bulunamaması gibi bir soruna neden olmamaktadır ancak emin olma skoru (confidence score) ciddi oranda azalmaktadır. Normal koşullarda yüzde doksan ortalama olmasına karşı diş telleri içeren bir örnekte ortalama yüzde altmışa kadar inmektedir. Beklenmedik bir durum ile karşılaşıldığında tahmin edildiği gibi modeli başarısında düşüş yaşanmıştır.

## 9 Sonuç

---

Tamamlanan projede kapsamında Mask R-CNN ile çocuk x-ray panoramik diş görüntülerindeki dişlerin tespit edilmesi, tespit edilen dişlerin maskelenmesi ve sınıflarının belirlenmesi amaçlanmıştır. Çocuk dişleri yetişkinlerden daha karmaşık ve belirli bir düzende olmadığı için yetişkinler için tasarlanan modeller çocuk x-ray panoramik diş görüntülerinde iyi sonuçlar vermemektedir. Bu projede Çocuklar için özelleştirilmiş bir diş tespiti ve sınıflandırılması yapan bir model eksikliği tespit edildiğinden sağlık sektöründe oluşan bu boşluk kapatılmak istenmiştir.

Projede nesne tespiti algoritması olarak Mask R-CNN seçilmiştir. Bu kararın sebebi diğer nesne tespiti ve sınıflandırma algoritmalarına kıyasla daha yeni ve güncel olmasıdır. Piksel düzeyinde maskeleme yapabilmesi, birden fazla sınıf için sınıflandırma konusunda yetenekli oluşu, diğer alternatiflerine göre hızlı ve kolay bir şekilde eğitim yapılmasına elverişli olması gibi bir çok sebep sayılabbilmektedir. Ayrıca Mask R-CNN için model eğitimi yapılabilecek bir kütüphane de belirlenmesi gerekmektedir. Bir çok farklı özellikte kütüphane olmasına rağmen en popüler kütüphaneler Tensorflow ile Detectron2'dir. Projeye Tensorflow ile başlanmasına rağmen Detectron2'nin getirdiği avantajlar yüzünden süreç içinde Detectron2'ye geçilmesine karar verilmiştir. Model eğitimi ise başarıyı artırmak adına hazır anotasyonlar yerine elle işaretlenmiş anotasyonlar ile gerçekleştirılmıştır.

Son aşamada elde edilen model dişleri tespit etmek ve maskeleme konusunda oldukça başarılı sonuçlar göstermekte, sınıflandırma konusunda ise yeterli düzeyde sonuç üretebilmektedir. 70 gibi çok düşük sayıda eğitim yapabileceğimiz resime sahip olduğumuzdan sonuçların büyük ölçüde tatmin edici olduğu söylenelebilir.

Mask R-CNN tabanlı bir proje yapacak kişilere mecbur kalmazlarsa Tensorflow yerine Detectron2 kullanımalarını tavsiye ediyoruz. Detectron2 az sayıda kişiye sahip ekipler için daha elverişli olacaktır. Aynı zamanda internette Google collab ile eğitim için bir çok kaynak bulunduğuundan tüm kodlama gereksinimlerinde Collab yeterli olacaktır.

## Referanslar

---

- [1] Wikipedia. “Dental notation.” (2022), [Online]. Available: [https://en.wikipedia.org/wiki/Dental\\_notation](https://en.wikipedia.org/wiki/Dental_notation) (visited on 04/29/2022).
- [2] StudyBlue. “Dental charting flashcards.” (2022), [Online]. Available: <https://tr.pinterest.com/pin/252412754090743650/> (visited on 05/01/2022).
- [3] A. Vector. “Data annotation tool analysis – how to use labelme.” (2020), [Online]. Available: <https://awkvect.com/data-annotation-tool-analysis-how-to-use-labelme/> (visited on 05/01/2022).
- [4] A. N. Ömeroğlu, N. Kumbasar, E. A. Oral, and I. Y. Ozbek, “Mask r-cnn algoritması ile hangar tespiti hangar detection with mask r-cnn algorithm,” in *2019 27th Signal Processing and Communications Applications Conference (SIU)*, IEEE, 2019, pp. 1–4.
- [5] B. A. meetup. “How convolutional neural networks.” (2018), [Online]. Available: <https://docs.google.com/presentation/d/1R-DnrghbU36j08X4scbrrlx6gFyJHgSL3bD274sutng/edit#slide=id.p1> (visited on 05/01/2022).
- [6] M. LLC. “Evrişimli sinir ağları el kitabı.” (2019), [Online]. Available: <https://stanford.edu/~shervine/l1/tr/teaching/cs-230/cheatsheet-convolutional-neural-networks> (visited on 04/30/2022).
- [7] paperswithcode. “Max pooling.” (2020), [Online]. Available: <https://paperswithcode.com/method/max-pooling> (visited on 05/01/2022).
- [8] ——, “Convolutional neural networks (cnns / convnets).” (2020), [Online]. Available: <https://cs231n.github.io/convolutional-networks/#fc> (visited on 05/01/2022).
- [9] M. LLC. “What-is-the-difference-between-cnn-and-r-cnn.” (2019), [Online]. Available: <https://www.quora.com/What-is-the-difference-between-CNN-and-R-CNN/answer/Burak-Berber> (visited on 04/30/2022).
- [10] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.
- [11] M. E. Sarac. “Cnn , r-cnn , fast r-cnn , mask r-cnn.” (2020), [Online]. Available: <https://merveelifsarac.medium.com/cnn-r-cnn-fast-r-cnn-mask-r-cnn-c90a1a4d76fb> (visited on 05/01/2022).
- [12] H. Mao. “Detectron2.” (2022), [Online]. Available: <https://github.com/facebookresearch/detectron2/tree/main/detectron2/config> (visited on 05/01/2022).

- [13] patrick labatut. “Detectron2 model zoo and baselines.” (2016), [Online]. Available: [https://github.com/facebookresearch/detectron2/blob/main/MODEL\\_ZOO.md](https://github.com/facebookresearch/detectron2/blob/main/MODEL_ZOO.md) (visited on 04/30/2022).
- [14] kHarshit. “Evaluation metrics for object detection and segmentation: Map.” (2019), [Online]. Available: <https://kharshit.github.io/blog/2019/09/20/evaluation-metrics-for-object-detection-and-segmentation> (visited on 06/01/2022).

# Özgeçmiş

---

## BİRİNCİ ÜYE

**İsim-Soyisim:** Aydin Kağan Türkoğlu  
**Doğum Tarihi ve Yeri:** 07.06.2000, Antalya  
**E-mail:** kagan.turkoglu@std.yildiz.com.tr  
**Telefon:** 0535 027 91 51  
**Staj Tecrübeleri:**

## İKİNCİ ÜYE

**İsim-Soyisim:** Göktuğ Köksal  
**Doğum Tarihi ve Yeri:** 17.01.1999, İzmir  
**E-mail:** goktug.koksal@std.yildiz.edu.tr  
**Telefon:** 0539 555 88 40  
**Staj Tecrübeleri:**

## Proje Sistem Bilgileri

**Sistem ve Yazılım:** Windows İşletim Sistemi, Python  
**Gerekli RAM:** 2GB  
**Gerekli Disk:** 256MB