

TÜRKİYE CUMHURİYETİ
YILDIZ TEKNİK ÜNİVERSİTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ



PANORAMİK DİŞ RÖNTGEN GÖRÜNTÜLERİNDE
HATALARIN TESPİTİ

19011077 – Ahmet Enis ŞİMŞİR
20011622 – Çağatay Alptekin

BİLGİSAYAR PROJESİ

Danışman
Dr. Öğr. Üyesi Hamza Osman İLHAN

Ocak, 2024

TEŞEKKÜR

Okulumuzun öğretim üyesi olan Hamza Osman İlhan Hocamıza bizi bu projeye dahil ettiği ve her hafta sıkı takip ederek hem koordine olmamızı hem de destek almamızı sağladığı için teşekkürlerimizi sunarız.

Ahmet Enis ŞİMŞİR
Çağatay Alptekin

İÇİNDEKİLER

KISALTMA LİSTESİ	v
ŞEKİL LİSTESİ	vi
TABLO LİSTESİ	vii
ÖZET	viii
ABSTRACT	ix
1 Giriş	1
1.1 Panoramik Diş Röntgeni	1
1.1.1 Panoramik Diş Röntgeni Görüntülerinde Görülebilen Hatalar . .	1
1.1.2 Röntgen Görsele Üzerindeki Hataların Tespiti	2
1.2 Projenin Amacı	2
1.3 Programın Çalışma Şekli	2
1.3.1 YOLOv8	2
1.3.2 TensorFlow	3
1.3.3 Bu Projede Kullanılabilecek Diğer Alternatifler	3
2 Ön İnceleme	5
2.1 Autonomous Dental Treatment Planning on Panoramic X-Ray Using Deep Learning Based Object Detection Algorithm (2023)	5
2.2 Object Detection on Dental X-Ray Images Using Deep Learning Method (2021)	5
2.3 Diffusion-Based Hierarchical Multi-Label Object Detection to Analyze Panoramic Dental X-rays (2023)	6
3 Fizibilite	8
3.1 Teknik Fizibilite	8
3.2 İş Gücü ve Zaman Fizibilitesi	8
3.3 Ekonomik Fizibilite	9
3.4 Yasal Fizibilite	9

4	Sistem Analizi	10
4.1	Veri Seti Analizi	10
5	Sistem Tasarımı	12
5.1	Yazılım Tasarımı	12
5.1.1	Veri Seti Tasarımı	12
5.1.2	Arayüz Tasarımı	12
5.2	Girdi Çıktı Tasarımı	12
6	Uygulama	14
6.1	Uygulamanın Ana Hatları	14
6.2	Örnek Çalıştırma Görüntüleri	14
7	Deneyisel Sonuçlar	18
7.1	İkili Sınıflandırma	18
7.1.1	Derin Öğrenme Ağı Denemeleri	18
7.1.2	Optimizier Denemeleri	18
7.2	Çoklu Sınıflandırma	19
7.2.1	Regülerizasyon Denemesi	19
8	Performans Analizi	21
8.1	İkili Sınıflandırma Analizi	21
8.2	Çoklu Sınıflandırma Analizi	23
9	Sonuç	24
	Referanslar	25
	Özgeçmiş	26

KISALTMA LİSTESİ

CNN	Convolutional Neural Networks
R-CNN	Region-based Convolutional Neural Network
GUI	Graphical User Interface
GPU	Graphics Processing Unit
CPU	Central Processing Unit
TPU	Tensor Processing Unit
YOLO	You Only Look Once
API	Application Programming Interface

ŞEKİL LİSTESİ

Şekil 2.1	Araştırmadaki Ölçüm Sonuçları	7
Şekil 3.1	Gantt Diyagramı	9
Şekil 4.1	Hatalı ve hatasız görüntülerin sayısı	11
Şekil 4.2	Hatalı görüntülerin tipleri ve sayıları	11
Şekil 6.1	Uygulama Ana Ekranı	15
Şekil 6.2	Görsel Yükleme Sonrası	15
Şekil 6.3	Kırpma İşlemi Tamamlandıktan Sonra	16
Şekil 6.4	Sınıflandırma Yapılırken	16
Şekil 6.5	Sonuç Ekranı	16
Şekil 6.6	İkili Sınıflandırma Doğru Yapılmış Ama Hata Tipi Bulunamamış Bir Örnek	17
Şekil 6.7	Hem İkili Hem Çoklu Sınıflandırmanın Doğru Yapıldığı Bir Örnek	17
Şekil 8.1	DenseNet169 İçin İkili Sınıflandırma Karşılaştırma Matrisi	22
Şekil 8.2	DenseNet169 İçin Çoklu Sınıflandırma Karşılaştırma Matrisi . . .	23

TABLO LİSTESİ

Tablo 7.1	Derin Öğrenme Ağı Denemeleri	18
Tablo 7.2	Optimizer Denemeleri	19
Tablo 7.3	Çoklu Sınıflandırma Denemesi	19
Tablo 7.4	Regülerizasyon Denemesi	20
Tablo 8.1	Modelin Analizi	21
Tablo 8.2	Modelin Analizi	23

PANORAMİK DİŞ RÖNTGEN GÖRÜNTÜLERİNDE HATALARIN TESPİTİ

Ahmet Enis ŞİMŞİR
Çağatay Alptekin

Bilgisayar Mühendisliği Bölümü
Bilgisayar Projesi

Danışman: Dr. Öğr. Üyesi Hamza Osman İLHAN

Panoramik diş röntgeni görüntüleri medikal alanda sıkça kullanılan bir araçtır. Bu görüntüler elde edilirken hastanın pozisyonu dolayısıyla çeşitli hatalar oluşabilmektedir. Hastanın başının pozisyonu nedeniyle oluşan bu hata uzman bir doktor veya teknisyen tarafından tespit edilir. Projemiz ise diş röntgeni görüntülerinden oluşan bir veri seti ile eğitilen ve kullanıcının yüklediği röntgen görüntülerinin hatalı olup olmadığını ve eğer hatalıysa hata tipini bildiren bir program tasarlamayı amaçlamaktadır.

Anahtar Kelimeler: Panoramik Diş Röntgen Görüntüleri, YOLOv8, Tensorflow, Nesne Tespiti, İkili Sınıflandırma, Çoklu Sınıflandırma

ABSTRACT

DETECTION OF ERRORS IN PANORAMIC DENTAL X-RAY IMAGES

Ahmet Enis ŞİMŞİR
Çağatay Alptekin

Department of Computer Engineering
Computer Project

Advisor: Assist. Prof. Dr. Hamza Osman İLHAN

Panoramic dental x-ray images are a frequently used tool in the medical field. However, it is not uncommon to make mistakes when obtaining these images. For example, if the patient's head is positioned incorrectly, this can cause an error and the error in the image can only be detected by a specialized doctor or technician. Our project aims to design a program that trains itself with a dataset of dental X-ray images and reports whether the X-ray images uploaded by the user are erroneous or not, and if so, the type of error.

Keywords: Panoramic Dental X-Ray Images, YOLOv8, Tensorflow, Object Detection, Binary Classification, Multi-Classification

Bu bölümde projenin amacından bahsedilmiştir.

1.1 Panoramik Diş Röntgeni

Panoramik Diş Röntgenleri diş doktorlarının sıkça kullanmak için başvurduğu bir görüntüleme şeklidir. Bu röntgen sayesinde tek bir işlemle hastanın çenesinin, dişlerinin, temporomandibular eklemlerinin, burun bölgesinin ve sinüslerinin görüntüsü oluşturulur ve bu görüntü üzerinde doktorlar inceleme yapabilir. Bu röntgen tipi efektif olarak aynı anda farklı bölgelerin incelenmesini sağlasa da hatalı çekim yapılmasına da müsaittir.

1.1.1 Panoramik Diş Röntgeni Görüntülerinde Görülebilen Hatalar

Elimizdeki panoramik diş röntgen görüntülerindeki hatalar hastanın kafasını yanlış yerleştirilmesi veya çekim esnasında hareket ettirmesiyle meydana gelmiştir. Başlıca karşılaşılan hatalar şunlardır:

- BAŞ AŞAĞI
- BAŞ YUKARI
- ÇENTİĞİN ÖNDEN ISIRILMASI
- ÇENTİĞİN ARKADAN ISIRILMASI
- HASTA BAŞINI SAĞA SOLA ÇEVİRMİŞ
- HASTA BAŞINI SAĞA SOLA YATIRMIŞ

1.1.2 Röntgen Görseli Üzerindeki Hataların Tespiti

Panoramik Diş Röntgeni Görselleri üzerindeki hatalar uzman bir diş doktoru veya diş teknisyeni tarafından tespit edilebilir. Dışarıdan bakan ve uzman olmayan bir insanın gördüğü normal ve düzgün sanılan bir görsel aslında hatalı bir görsel olabilir.

1.2 Projenin Amacı

Projenin amacı ise diş hekimlerine ve teknisyenlerine röntgen görüntülerinde hata tespiti yapmalarına yardımcı olacak bir program tasarlamak, insanın gözünden kaçabilecek hataları önlemek ve onların işini hafifletmektir. Günümüzde yapay zeka hayatımızın her alanına girmiştir ve gündelik işlerde bize yardımcı olmak konusunda büyük işler başarmıştır. Dolayısıyla böyle bir sorunun da yapay zeka yardımıyla çözülmesi amaçlanmıştır.

Projede doktorların röntgen görüntülerini yükleyebileceği bir arayüz programı yardımıyla görüntüde hata olup olmadığı ve hata varsa hata tipi bilgisine erişimleri hedeflenmiştir.

1.3 Programın Çalışma Şekli

Programımız günümüzde oldukça gelişmiş olan yapay zeka ve görüntü işleme kütüphaneleri yardımıyla çalışmaktadır. İlk olarak röntgen görüntüsü üzerinde alt çeneyi tespit etmek için **YOLOv8** kullanılmaktadır. Hata tespiti alt çene üzerinden yapıldığı için çeneyi yüksek doğrulukla tespit etmek önemlidir. YOLOv8 ise güncel ve etkili bir model olduğu için tercih edilmiştir. Öncelikle YOLOv8 ile nesne tespiti yapılarak alt çene kemiğini saran bir poligonun koordinatları elde edilir ve bir python kodu ile bu poligon görselden kırılır.

Sonrasında görüntüde hata olup olmadığını tespit etmek için **TensorFlow** ve **DenseNet169** derin öğrenme ağı kullanılarak eğitilmiş modelimiz kullanılır. Bu model röntgen görüntülerinde ikili sınıflandırma yapmak için bizim tarafımızdan eğitilmiştir. Bu şekilde hatalı ve hatasız görseller ayıklanır. En son olarak da hatalı görsellerin hata tipini tespit etmek için yine **DenseNet169** ile eğittiğimiz bir çoklu sınıflandırma modeli kullanılır.

1.3.1 YOLOv8

YOLO, nesne tespiti için kullanılan bir derin öğrenme algoritmasıdır. Görüntüleri veya videoları analiz ederek farklı nesneleri tanımlamak için kullanılır. YOLO, bir görüntüyü

veya videoyu tek bir ađ geidinden geirir ve aynı anda birden ok nesneyi algılamak iin bir "gridding" tekniđi kullanır. Bu teknikle nesneyi kk paralara blp her parayı ayrı ayrı analiz eder ve yksek hızlı bir nesne tespiti bařarısı sađlar.

YOLOv8 ise YOLO'nun en gncel ve efektif srmdr. Bu versiyon, aynı anda daha dođru nesne tespitini, daha yksek bir hızı ve daha gl bir nesne tespiti yeteneđini sunar.

1.3.2 TensorFlow

TensorFlow, Google tarafından geliřtirilen aık kaynaklı bir makine đrenimi ve derin đrenme ktphanesidir. Bu ktphane, byk lekli sayısal hesaplamalar yapmak, yapay zeka ve derin đrenme modelleri oluřturmak ve eđtmek gibi iřlerde kullanılır.

Projemizde ikili ve oklu sınıflandırma iin kullanılmıřtır.

1.3.3 Bu Projede Kullanılabilecek Diđer Alternatifler

Projede hangi ktphanelerin kullanılacađına karar verirken daha nceden yapılmıř projeler incelenmiřtir ve hocamızın tavsiyeleri dikkate alınmıřtır. Nesne tespiti kısmı iin 2 seenek zerine yođunlařılmıřtır. Bunlardan biri nesne tespiti iin TensorFlow kullanılarak geliřtirilen Mask R-CNN, diđer i se kullanım kolaylıđı aısından oka tercih edilen YOLO'dur. Nesne tespiti yaptıktan sonra sınıflandırma iin ise Pytorch ktphanesi ve pek ok derin đrenme ađı denenmiřtir ve bu derin đrenme ađları, raporun Deneyisel Sonular blmnde detaylıca aıklanmıřtır.

1.3.3.1 Mask R-CNN

Mask R-CNN, nesne tespiti, sınıflandırma ve ayrıntılı piksel tabanlı maskeleri tahmin etmek iin kullanılan bir derin đrenme modelidir. "R-CNN" (Region-based Convolutional Neural Network) ailesinin bir yesidir ve nceki versiyonları olan R-CNN ve Faster R-CNN'den evrilmiřtir.

Mask R-CNN, nesne tespiti ve segmentasyonu yapmak iin geliřtirilmiřtir. Modelin avantajı, nesneleri sınıflandırmakla birlikte her pixelin hangi nesneye ait olduđunu bulabilmesidir.

Fakat YOLOv8'in daha yeni olması dolayısıyla Mask R-CNN'ne gre hem daha hızlı hem daha dođru sonu verebildiđine dair bazı bulgular edinilmesi dolayısıyla Mask R-CNN rafa kaldırılarak YOLOv8 ile devam edilme kararı alınmıřtır.

1.3.3.2 Pytorch

PyTorch, derin öğrenme ve yapay zeka modelleri geliřtirmek için kullanılan açık kaynaklı bir Python kütüphanesidir. Kullanım amacı Tensorflow'a benzese de yaklaşım olarak farklılık göstermektedir.

TensorFlow ile ilgili daha çok kaynak bulunduđu için TensorFlow ile devam edilmeye karar verilmiştir.

2 Ön İnceleme

Bu bölümde, projenin yapılacağı alana yakın, daha önce yapılmış olan çalışmalar incelenmiştir.

2.1 Autonomous Dental Treatment Planning on Panoramic X-Ray Using Deep Learning Based Object Detection Algorithm (2023)

[1] Bu çalışma da bizim projemize benzer şekilde panoramik diş röntgen görüntüleri üzerinedir ama bizimkinden farklı olarak görüntüdeki hataları tespit etmek için değil, diş hastalığını tespit edip tedavi yöntemi önermek için geliştirilmiştir.

Projede YOLOv7 kullanılmıştır ve oldukça başarılı olunmuştur. Model eğitmek için 1025 röntgen görüntüsü içeren bir veri seti kullanılmıştır.

%100 hassasiyet, %76 F1 skoru, %96 Geri Çağırma ve %81.9 mAP skoru elde edilmiştir ki bu da projenin başarısını gösterir.

2.2 Object Detection on Dental X-Ray Images Using Deep Learning Method (2021)

[2] Bu çalışma da aynı şekilde diş sorununu tespit edip doğru tedavi yöntemi bulmayı amaçlamaktadır.

Projede Mask R-CNN yöntemi kullanılmıştır.

Veri seti olarak 116 görüntü kullanılmıştır (110 tanesi eğitim, 6 tanesi test için) ki bu öncekine göre çok daha az bir sayı.

Veri seti yeterli büyüklükte olmadığı halde oldukça başarılı bir sonuç alınmıştır.

Güven skoru 0.91 ile 0.96 arasındadır. Ama veri seti küçük olduğu için mevcut hastalık tipi de küçüktür. Bu da göz önünde bulundurulmalıdır.

2.3 Diffusion-Based Hierarchical Multi-Label Object Detection to Analyze Panoramic Dental X-rays (2023)

[3] Bu araştırma da hastalıkların tespiti ve tedavi önerme içindir. Ama bu araştırmada diğer ikisinden farklı olarak birden fazla tespit yöntemi kullanılmıştır ve performansları ölçülmüştür. Bu değişik modellerin hepsi aynı bilgisayar üzerinde yakın koşullarda çalıştırılmıştır.

We evaluate models' performances using a combination of Average Recall (AR) and Average Precision (AP) scores with various Intersection over Union (IoU) thresholds. This included $AP_{[0.5,0.95]}$, AP_{50} , AP_{75} , and separate AP scores for large objects (AP_l), and medium objects (AP_m).

Method	AR	AP	AP_{50}	AP_{75}	AP_m	AP_l
Quadrant						
RetinaNet [16]	0.604	25.1	41.7	28.8	32.9	25.1
Faster R-CNN [20]	0.588	29.5	48.6	33.0	39.9	29.5
DETR [4]	0.659	39.1	60.5	47.6	55.0	39.1
Base (DiffusionDet) [5]	0.677	38.8	60.7	46.1	39.1	39.0
Ours w/o Transfer	0.699	42.7	64.7	52.4	50.5	42.8
Ours w/o Manipulation	0.727	40.0	60.7	48.2	59.3	40.0
Ours w/o Manipulation and Transfer	0.658	38.1	60.1	45.3	45.1	38.1
Ours (Manipulation+Transfer+Multilabel)	0.717	43.2	65.1	51.0	68.3	43.1
Enumeration						
RetinaNet [16]	0.560	25.4	41.5	28.5	55.1	25.2
Faster R-CNN [20]	0.496	25.6	43.7	27.0	53.3	25.2
DETR [4]	0.440	23.1	37.3	26.6	43.4	23.0
Base (DiffusionDet) [5]	0.617	29.9	47.4	34.2	48.6	29.7
Ours w/o Transfer	0.648	32.8	49.4	39.4	60.1	32.9
Ours w/o Manipulation	0.662	30.4	46.5	36.6	58.4	30.5
Ours w/o Manipulation and Transfer	0.557	26.8	42.4	29.5	51.4	26.5
Ours (Manipulation+Transfer+Multilabel)	0.668	30.5	47.6	37.1	51.8	30.4
Diagnosis						
RetinaNet [16]	0.587	32.5	54.2	35.6	41.7	32.5
Faster R-CNN [20]	0.533	33.2	54.3	38.0	24.2	33.3
DETR [4]	0.514	33.4	52.8	41.7	48.3	33.4
Base (DiffusionDet) [5]	0.644	37.0	58.1	42.6	31.8	37.2
Ours w/o Transfer	0.669	39.4	61.3	47.9	49.7	39.5
Ours w/o Manipulation	0.688	36.3	55.5	43.1	45.6	37.4
Ours w/o Manipulation and Transfer	0.648	37.3	59.5	42.8	33.6	36.4
Ours (Manipulation+Transfer+Multilabel)	0.691	37.6	60.2	44.0	36.0	37.7

Diffusion-Based Hierarchical Multi-Label Detection 11

Detection Model	Image Encoder Backbone	Iterations	Learning Rate
Ours	FPN-Swin Transformer	40000	0.000025
DiffusionDet	FPN-Swin Transformer	40000	0.000025
Faster R-CNN	ResNet101	40000	0.02
RetinaNet	ResNet101	40000	0.01
DETR	ResNet50	300(epochs)	0.0001

Table 2: Different detection models are utilized for comparison with our method. The best test metrics for each model are selected for the results. All models are trained with randomly cropped and resized panoramic X-rays with a batch size of 16. All training is done on a single NVIDIA RTX A6000 48 GB GPU.

Dataset	Training	Validation	Testing
Quadrant	590	103	N/A
Quadrant-Enumeration	539	95	N/A
Quadrant-Enumeration-Diagnosis	705	50	250

Table 3: To ensure accurate testing of all models, we only use fully labeled data with quadrant-enumeration-diagnosis for abnormal tooth detection. We do not utilize quadrant or quadrant-enumeration data for testing. Our diagnosis labels have four specific classes: caries, deep caries, periapical lesions, and impacted.

Şekil 2.1 Araştırmadaki Ölçüm Sonuçları

Bu bölümde projenin fizibilite analizlerine yer verilmiştir.

3.1 Teknik Fizibilite

Proje için Windows işletim sisemine sahip 2 bilgisayar kullanılmıştır. Bu bilgisayarların özellikleri farklı olsa da Google Colab üzerinde çalışıldığı için iki bilgisayarda da neredeyse aynı performans sonucu gözlenmiştir.

Google Colab ise Google tarafından sunulan ücretsiz bir bulut tabanlı Jupyter Notebook ortamıdır. Colab, kullanıcıların tarayıcı üzerinden erişebileceği bir hizmettir ve kullanıcıların kendi sunucularını kullanmalarına gerek kalmadan, Google'un altyapısını kullanarak işlemleri gerçekleştirmesini sağlar. Ayrıca, Colab üzerinde bulunan kaynaklar (CPU, GPU veya TPU gibi) kullanıcılara sunulur ve özellikle derin öğrenme gibi yoğun işlemler gerektiren durumlarda daha hızlı sonuç veren GPU veya TPU kullanma imkanı sağlar.

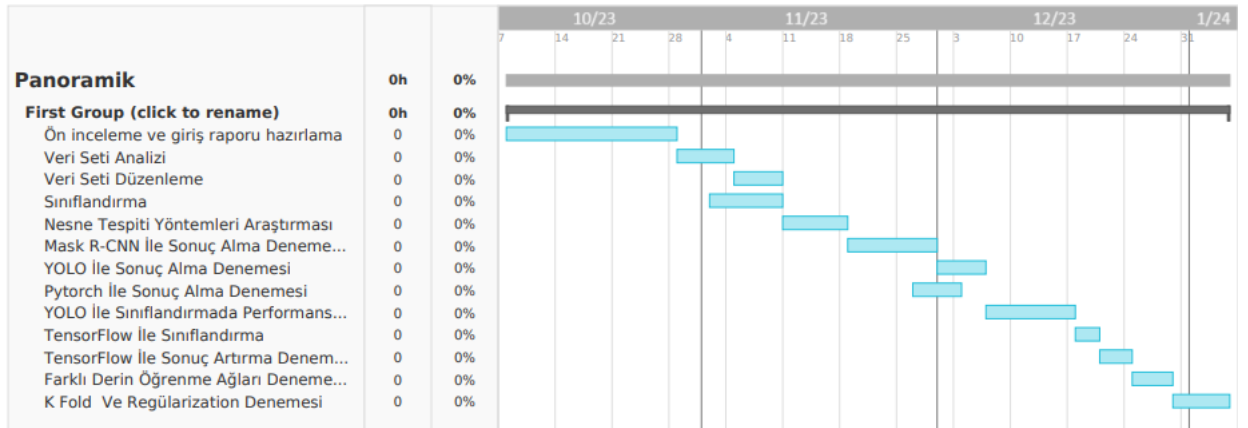
Google Colab sayesinde sanal bir ortam oluşturulmuştur ve bilgisayarımızın GPU'su değil, Colab'ın ücretsiz olarak sunduğu GPU kullanılmıştır. Bu GPU'nun modeli ise Tesla T4'tür. Bu Colab'ın ücretsiz fakat sınırlı kullanıma sunduğu GPU'dur.

Ayrıca Ultralytics şirketine ait olan Pytorch ile entegre olan YOLOv8 deposu kullanılmıştır. Bu kütüphaneler tamamen ücretsiz olduğu gibi şirket çalışanları Discord gibi ortamlarda ücretsiz teknik destek sunmaktadırlar.

3.2 İş Gücü ve Zaman Fizibilitesi

Planlamamızı gösteren Gantt Şeması aşağıdadır.

.



Şekil 3.1 Gantt Diyagramı

3.3 Ekonomik Fizibilite

Google Colab kullanılmaktadır ve Google Colab tamamen ücretsiz fakat sınırlı kullanım sağlayan bir araçtır. Ayrıca GPU ve CPU gereksinimi de buradan sağlanmıştır. Dolayısıyla herhangi bir bilgisayar ve bir internet bağlantısı yeterlidir.

3.4 Yasal Fizibilite

Projede kullanılan kütüphanelerin tamamı halka açıktır. Proje yürütülürken hiçbir etik kural da çiğnenmemektedir. Ayrıca kullanılan veri seti de hocamız tarafından bizlere verilmiştir ve gerekli izinlere sahiptir.

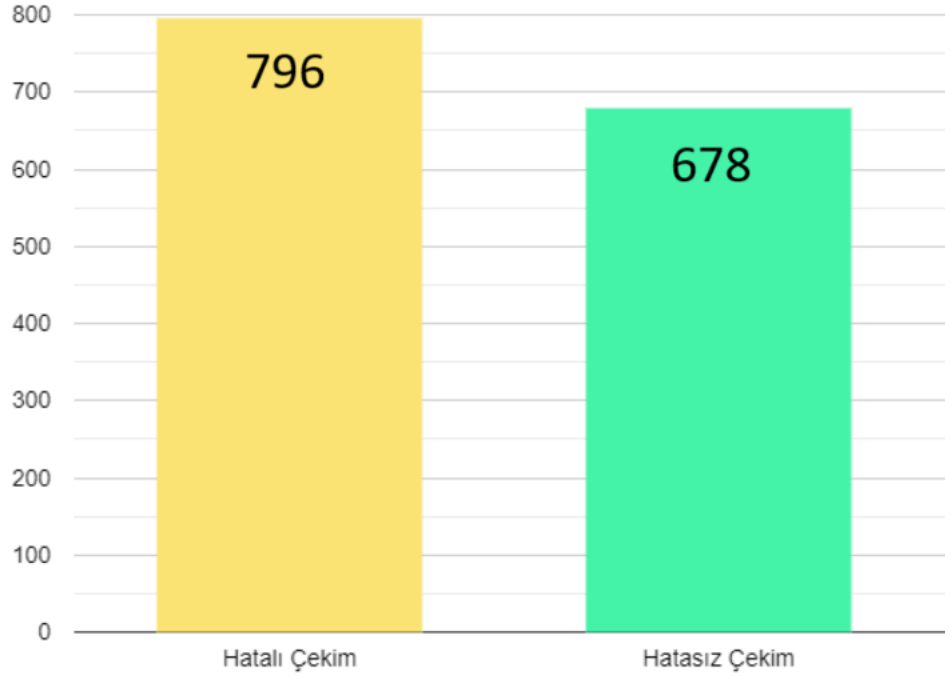
4.1 Veri Seti Analizi

Veri Setimiz 2 adet diř hekiminin sunduđu panoramik röntgen çekimlerinden oluşmaktadır. Her röntgen çekimi için bir adet görsel ve bir adet yazı dosyası bulunmaktadır.

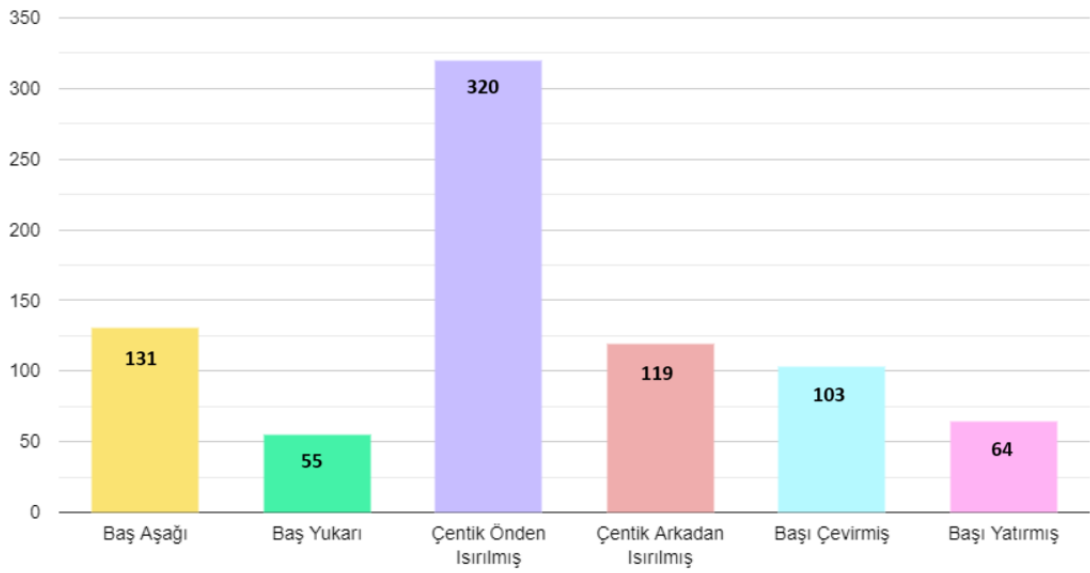
Yazı dosyasında görseldeki alt çene kemiğinin etrafında çizilmiş bir poligonun noktalarının koordinatları vardır. Ama yazı dosyası YOLOv8 için istenen formatta olmadığı için veri setindeki xml ve json dosyalarını güncellemek gerekmiştir. Bunun için de **Roboflow** isimli web sitesi kullanılmıştır.

Bir adet Excel dosyasında ise her röntgen çekimi için hata olup olmadığı (0 hata yok, 1 ise hata var demektir) ve hata varsa hata tipinin ne olduğu tutulmaktadır.

Veri setindeki değerlerin grafikleri aşağıdadır:



Şekil 4.1 Hatalı ve hatasız görüntülerin sayısı



Şekil 4.2 Hatalı görüntülerin tipleri ve sayıları

5

Sistem Tasarımı

5.1 Yazılım Tasarımı

5.1.1 Veri Seti Tasarımı

Veri seti eğitime uygun formatta değildir. Dolayısıyla veri setini güncellememiz gerekmiştir. Bununla ilgili bir Python kodu yazılmıştır ve bu kod yardımıyla "name" etiketi olan yerlere sınıf bilgisi eklenmiştir. Ayrıca "path" kısmına da temsil edilen görselin ismi yazılmıştır.

Daha sonra Roboflow sitesi yardımıyla tam anlamıyla YOLOv8'in okuyabileceği bir veri setine dönüştürülmüştür. Bunun için Roboflow'a görseller ve xml dosyalarımız ayrı ayrı yüklenmiştir ve hazırlanan veri seti export edilmiştir. Export ettikten sonra Roboflow'un sağladığı API Google Colab'ta kullanılarak görüntüler hiç indirme ve yükleme işlemi yapmadan otomatik olarak Google Colab projemize taşınmıştır.

Ayrıca veri setinde görseli bulunup yazı dosyası bulunmayan veya tam tersi olan röntgen görüntüleri silinmiştir.

5.1.2 Arayüz Tasarımı

Programımızın doktorlar ve teknisyenler tarafından kullanımını sağlamak için Python kodlama dili kullanılarak bir adet masaüstü uygulaması geliştirilmiştir.

5.2 Girdi Çıktı Tasarımı

TensorFlow, Pytorch ve YOLOv8 gibi farklı kütüphaneler denenmiştir ve bunların sonuçları kaydedilmiştir. Ayrıca TensorFlow ile sınıflandırma yaparken farklı derin öğrenme ağları kullanılmıştır ve bunların sonuçlarına raporun Deneysel Sonuçlar bölümünde değinilmiştir. En iyi sonucu veren model olduğu için DenseNet169 ağı ile eğittiğimiz model, masaüstü uygulamamızda kullanılmıştır. Kullanıcı bir röntgen görseli yüklediği zaman YOLOv8 ile alt çene kemiği kırplır ve kırplmış olan alt

çene kemiği DenseNet169 modeli ile ikili ve eğer bir hata tespit edilirse çoklu sınıflandırmaya tabi tutularak görselin hatalı olup olmadığına ve hatalıysa hata tipine dair tahmin yapılır. Bu tahmin sonuç ekranında gözüktür.

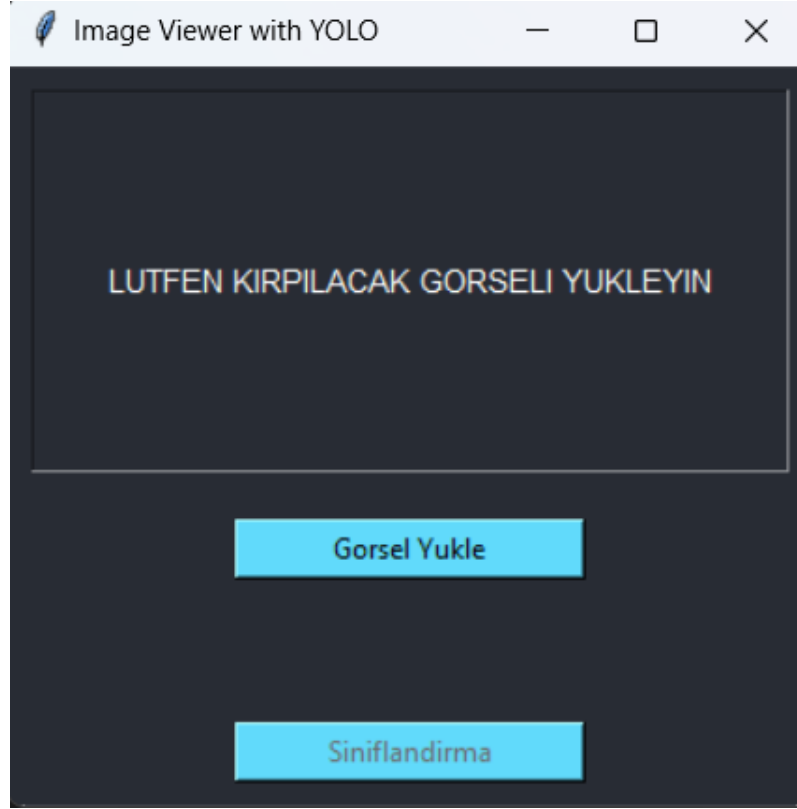
Bu bölümde işin arka planı yerine arayüz uygulaması ile ilgili bilgiler verilecektir.

6.1 Uygulamanın Ana Hatları

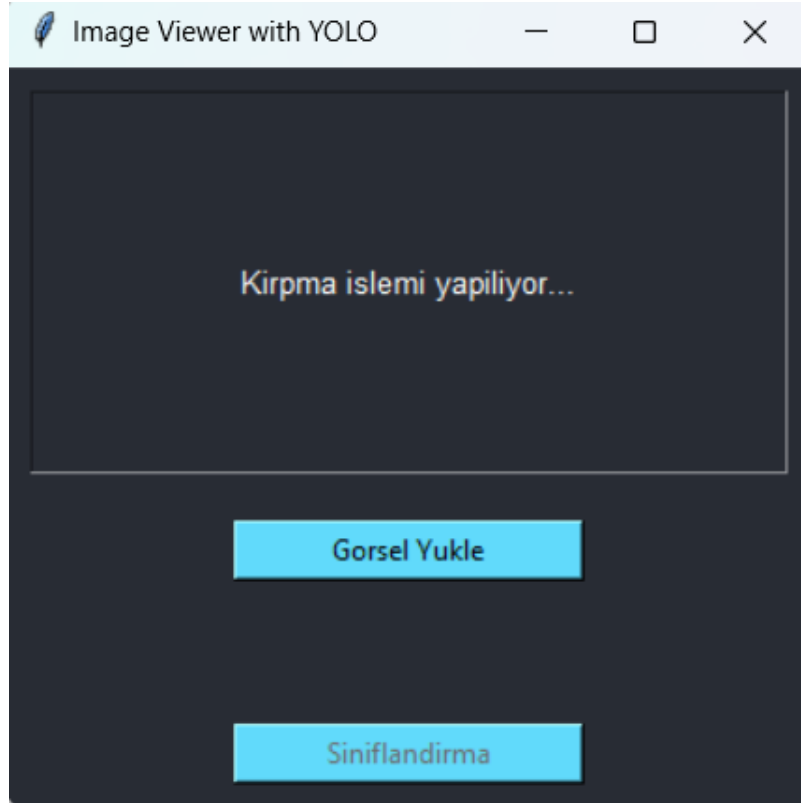
Uygulamanın amacı kullanıcı için rahat bir kullanım sağlamaktır. Kullanıcı öncelikle bir adet röntgen görüntüsü yükler ve uygulama YOLOv8 modelimizi kullanarak görseli kırıp ekrana verir. Daha sonrasında kullanıcı sınıflandırma butonuna tıklayarak sınıflandırmayı başlatır. Bu aşamada uygulama, daha önceden eğittiğimiz CNN modelini kullanarak ikili sınıflandırma yapar ve sonrasında hata varsa yine önceden eğittiğimiz başka bir modelle çoklu sınıflandırma yapar ve sonuçları ekrana yazdırır.

Ayrıca verilen sonucun doğruluğunu anlamak amacıyla resmin gerçek sonucu uygulamanın sonucunun yanında kullanıcıya sunulmuştur.

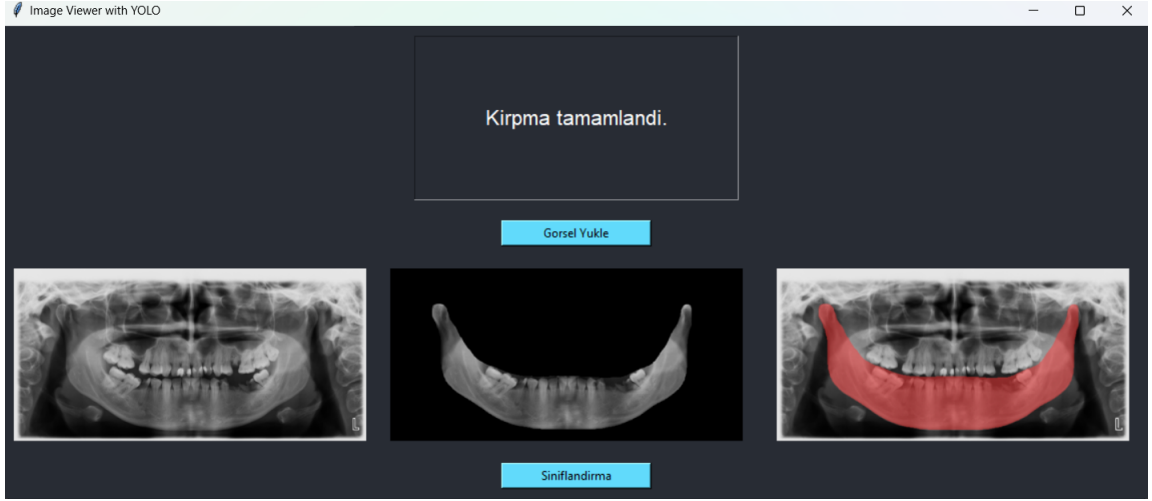
6.2 Örnek Çalıştırma Görüntüleri



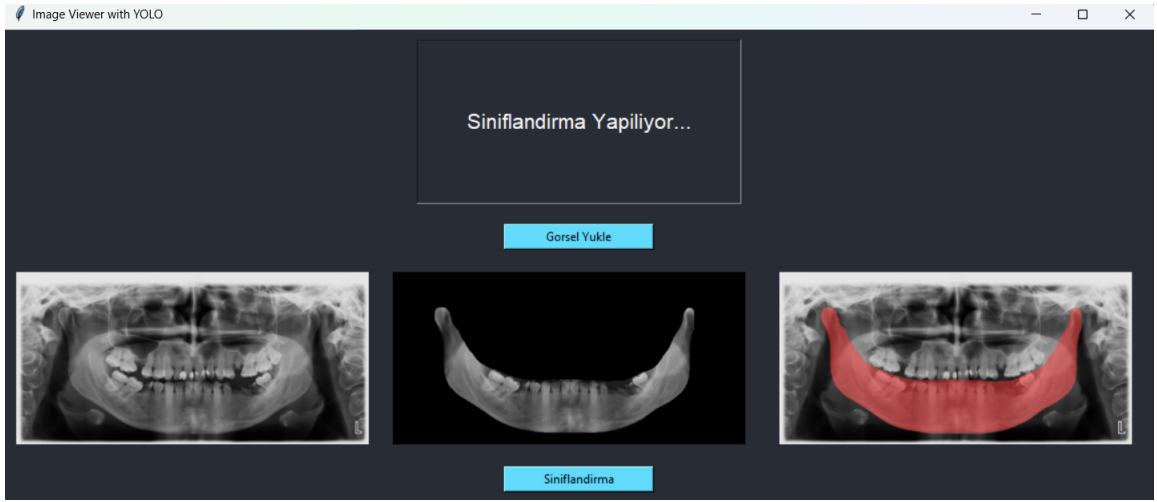
Şekil 6.1 Uygulama Ana Ekranı



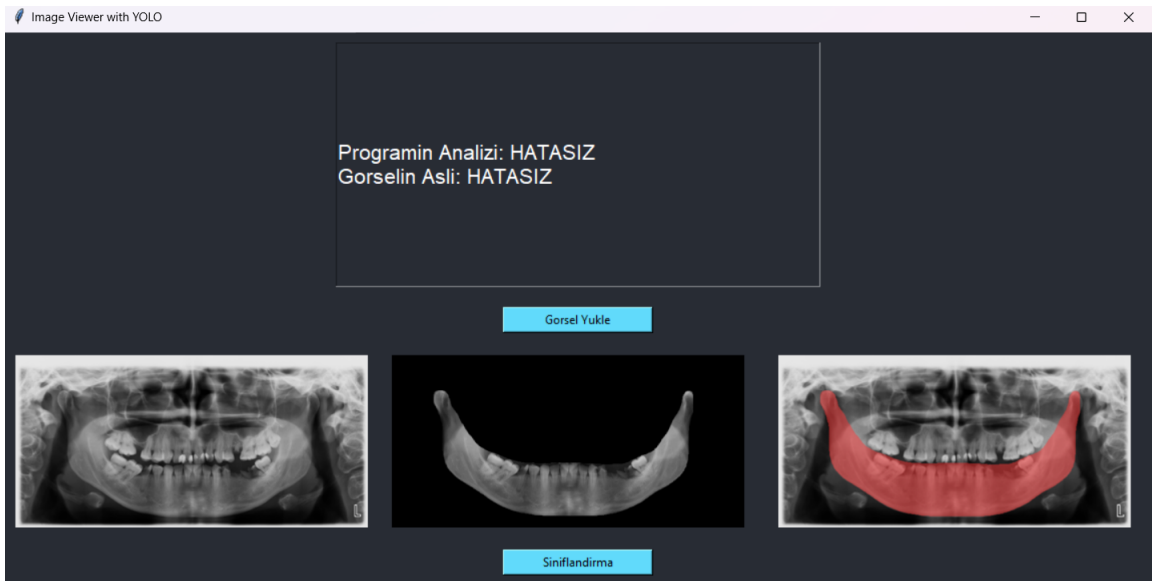
Şekil 6.2 Görsel Yükladikten Sonra



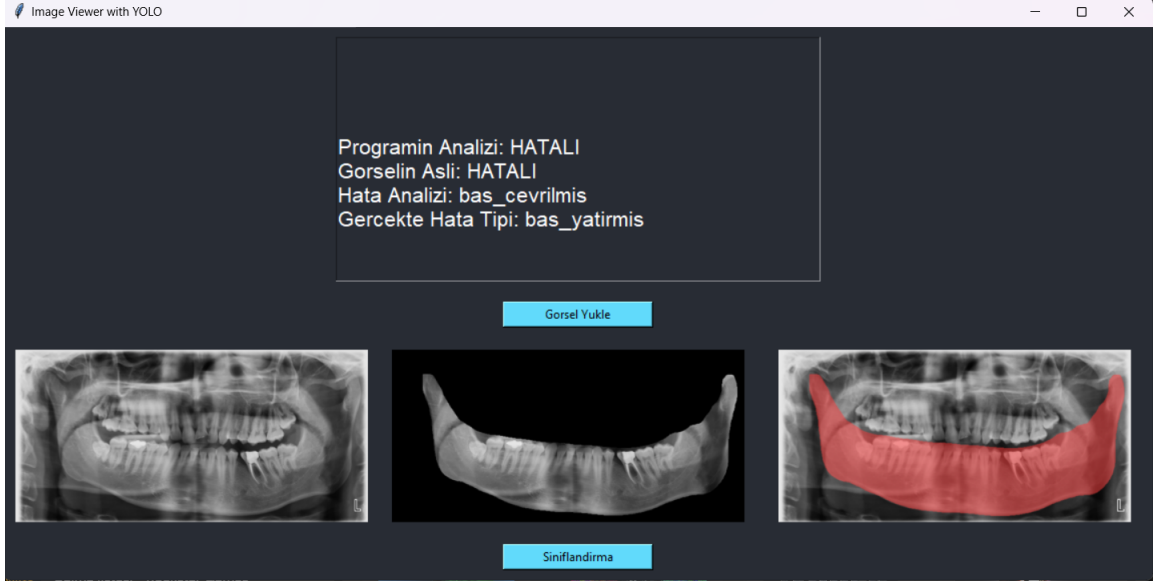
Şekil 6.3 Kırpma İşlemi Tamamlandıktan Sonra



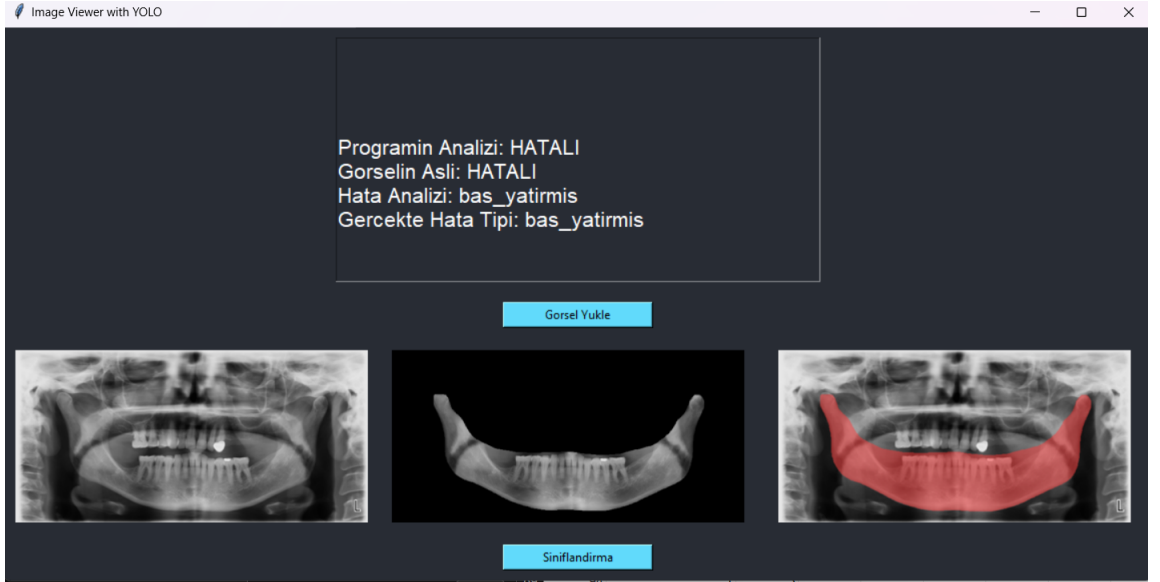
Şekil 6.4 Sınıflandırma Yapılırken



Şekil 6.5 Sonuç Ekranı



Şekil 6.6 İkili Sınıflandırma Doğru Yapılmış Ama Hata Tipi Bulunamamış Bir Örnek



Şekil 6.7 Hem İkili Hem Çoklu Sınıflandırmanın Doğru Yapıldığı Bir Örnek

7

Deneyisel Sonuçlar

Hem en iyi çalışan modeli bulmak, hem de bu modeli en iyi performansla çalıştırmak için pek çok denemeler yapılmıştır. Bunların sonuçları aşağıdadır.

7.1 İkili Sınıflandırma

7.1.1 Derin Öğrenme Ağı Denemeleri

İkili sınıflandırmanın amacı görselde hata olup olmadığını tespit etmektir. Bunu yapabilmek ve en iyi performansı gösteren ağı bulmak için denemeler yapılmıştır. Denemelerin daha gerçekçi sonuç vermesi için **K Fold** yöntemi uygulanmıştır ve veri seti 5'e bölünerek her biri için ayrı ayrı test işlemi uygulanmıştır ve sonuçların ortalaması incelenmiştir.

Denenen modellerin analizleri aşağıdadır.

Tablo 7.1 Derin Öğrenme Ağı Denemeleri

Modeller	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Ortalama
DenseNet169	%67	%62	%59	%65	%62	%63
MobileNetV2	%63	%62	%64	%63	%59	%62
EfficientNetB3	%62	%62	%60	%62	%56	%60
ResNet50	%68	%62	%63	%60	%55	%62

Görüldüğü gibi DenseNet169, MobileNetV2 ve Resnet50'nin sonuçları çok yakın olsa da karşılaştırma matrisleri üzerinde yapılan analizler sonucu ResNet50'nin başarısız olduğu kanaatine varılmıştır ve kalan modeller ile (DenseNet169 ve MobileNetV2) devam etme kararı alınmıştır.

7.1.2 Optimizer Denemeleri

Üstte verilen ağ denemelerinde AdamW optimizer'ı kullanılmıştır. Daha iyi bir optimizer olup olmadığından emin olmak için bazı denemeler yapılmıştır.

Üstte en iyi sonucu veren iki ağ (DenseNet169 ve MobileNetV2) ile farklı optimizeler kullanılarak yapılan deneylerin sonuçları aşağıdadır.

Tablo 7.2 Optimizer Denemeleri

Modeller ve Optimizer'lar	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Ortalama
DenseNet169 + AdamW	%67	%62	%59	%65	%62	%63
DenseNet169 + Adam	%67	%63	%67	%65	%63	%65
DenseNet169 + RMSprop	%67	%66	%61	%69	%62	%65
MobileNetV2 + AdamW	%68	%64	%61	%61	%61	%63
MobileNetV2 + Adam	%60	%63	%63	%60	%58	%61
MobileNetV2 + RMSprop	%66	%65	%61	%62	%62	%63

DenseNet169 modeliyle kullanılan RMSprop optimizere en iyi sonucu verdiği için onunla devam etmeye karar verilmiştir.

Learning rate'i düşürmenin performans üzerinde düşük düzeyde artırıcı bir etkisi olsa da eğitim zamanını uzatmaktadır. Dolayısıyla çok düşürmeyip 0.0001 ile devam edilmeye karar verilmiştir.

7.2 Çoklu Sınıflandırma

Çoklu sınıflandırma denemeleri sadece hatalı görüntüleri içeren bir veri seti ile yapılmıştır.

İkili sınıflandırmada iyi başarı gösteren modeller denenmiştir.

Tablo 7.3 Çoklu Sınıflandırma Denemesi

Model	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Ortalama
DenseNet169	%55	%52	%51	%55	%51	%53
MobileNetV2	%33	%40	%39	%49	%41	%40

Görüldüğü üzere DenseNet169 çoklu sınıflandırmada çok daha başarılı olmuştur.

7.2.1 Regülerizasyon Denemesi

Veri setimiz yeterince örnek görsele sahip değildir. 4.2 grafiğinde de görüldüğü gibi bazı sınıflardan çok az sayıda örnek bulunmaktadır. Bu da ciddi bir **overfitting** sorununa yol açmıştır. Daha fazla veri elde edemeyeceğimiz için bu sorunu çözmek için bir regularization yöntemi olan **L2 Düzenleme** tekniğine ve eldeki verileri değiştirerek kullanmayı amaçlayan data augmentation(veri artımı) yöntemlerine başvurulmuştur. L2 Düzenleme tekniği bir parametreyle çalışmaktadır. L2 Düzenleme

tekniki olmadan, varken ve farklı parametrelerle denemeler yapılarak sonuçları kaydedilmiştir.

Aşağıdaki bütün ölçümler DenseNet169 ağı ve RMSprop optimizier'ı kullanılarak yapılmıştır.

Tablo 7.4 Regülerizasyon Denemesi

Regülerizasyon Tipi	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Ortalama
Regülerizasyonsuz	%48	%42	%50	%41	%49	%46
L2 parametresi=0.01	%50	%48	%46	%45	%46	%47
L2 parametresi=0.1	%55	%52	%51	%55	%51	%53

8 Performans Analizi

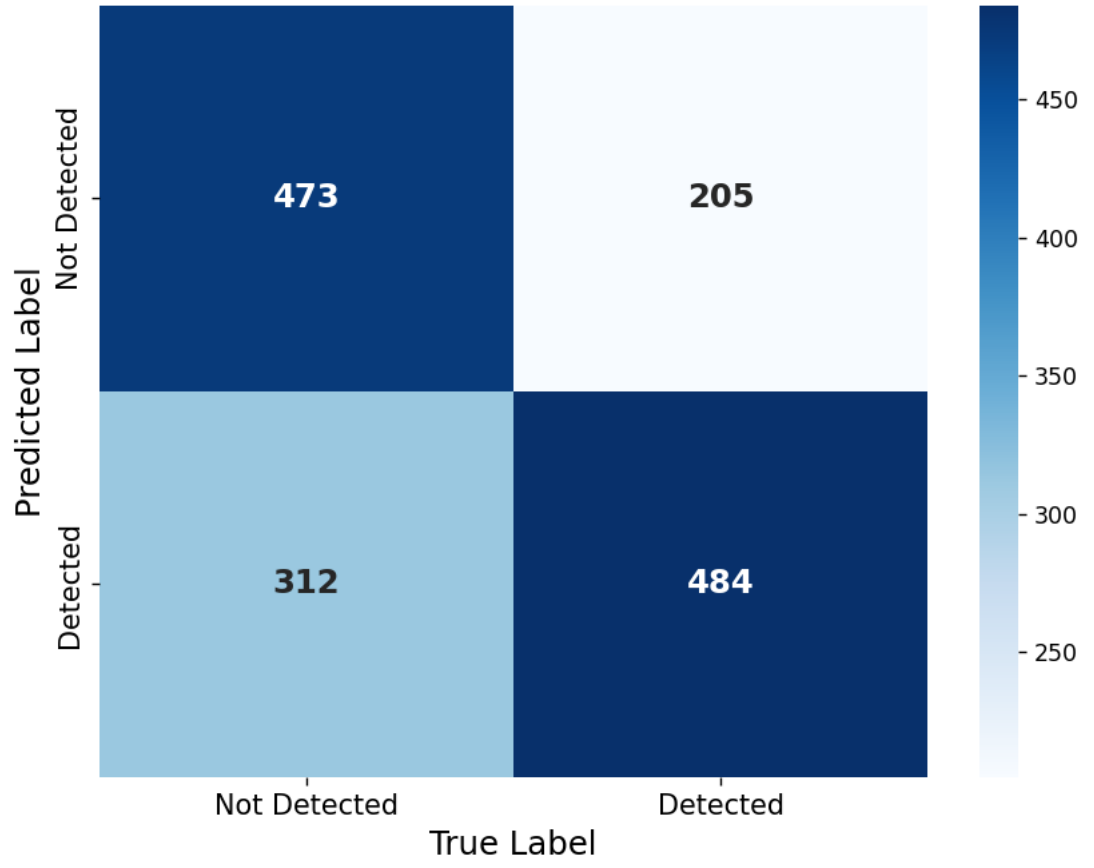
8.1 İkili Sınıflandırma Analizi

Aşağıda en iyi modelimiz olan DenseNet169 ile ilgili bilgiler bulunmaktadır. Bu model aynı zamanda arayüzde de kullanılan modeldir.

Model eğitilirken modelin tamamı hakkında daha gerçekçi bir fikre sahip olma amacıyla **K Fold Çaprazlama** yöntemi kullanılmıştır ve k değeri 5 alınmıştır. Bu şekilde her fold için ayrı bir eğitim ve test süreci yapılmıştır ve yaklaşık 1500 görüntünün sırayla her döngüde farklı 300 tanesi kullanılmıştır. Bu şekilde bütün verilerin bir kez test verisi olarak kullanılması amaçlanmıştır.

Tablo 8.1 Modelin Analizi

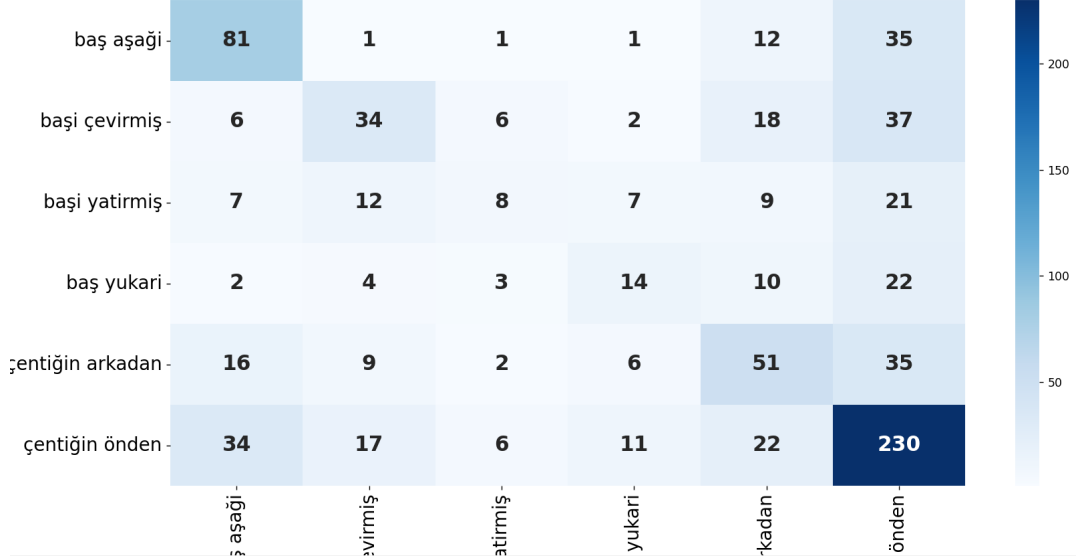
Sınıf	Accuracy	Precision	Recall	F1 Score
Hatasız	%64.93	0.70	0.60	0.65
Hatalı	%64.93	0.61	0.70	0.65



Şekil 8.1 DenseNet169 İçin İkili Sınıflandırma Karşılaştırma Matrisi

8.2 Çoklu Sınıflandırma Analizi

Aşağıda, arayüzde kullanılan çoklu sınıflandırma modeli ile ilgili analizler verilmiştir. Bu analizler üretilirken tamamı hatalı görsellerden oluşan bir veri seti kullanılmıştır.



Şekil 8.2 DenseNet169 İçin Çoklu Sınıflandırma Karşılaştırma Matrisi

Tablo 8.2 Modelin Analizi

Sınıf	Accuracy	Precision	Recall	F1 Score
Baş Aşağı	%85.48	0.62	0.55	0.58
Başı Çevirmiş	%85.86	0.33	0.44	0.38
Başı Yatırmış	%90.66	0.13	0.31	0.18
Baş Yukarı	%91.41	0.25	0.34	0.29
Çentik Arkadan	%82.45	0.43	0.42	0.42
Çentik Önden	%69.7	0.72	0.61	0.66

9 Sonuç

Panoramik Röntgen Görüntülerinde Hata Tespiti projemizdeki amacımız elimizdeki veri setini kullanarak yüklenen bir röntgen görüntüsünün hata içerip içermediğini bulan ve hatalıysa hata tipini tespit eden bir uygulama geliştirmektir. Veri setimizde görsellerle birlikte görseldeki alt çene kemiğinin etrafını saran poligonun koordinatları da bulunmaktaydı.

Projede farklı yöntemler denenmiş olsa da son durumda YOLOv8 modeli ile panoramik röntgen görüntüsü görselinde alt çene kemiğini tespit eden ve eğittiğimiz CNN modelleri ile tespit edilen alt çene kemiğinde ikili sınıflandırma ile hata olup olmadığını belirleyen ve sonra hata var ise çoklu sınıflandırma ile hata tipini tespit eden bir uygulama geliştirilmiştir.

İkili sınıflandırmada DenseNet169 modeli kullanılmıştır ve %65 oranında başarı yakalanmıştır. Çoklu sınıflandırmada ise çoklu sınıflandırma için özel olarak eğittiğimiz bir DenseNet169 modeli kullanılmıştır ve %53 doğruluk oranını yakalamıştır. Ölçümlerimiz esnasında farklı yöntemler, farklı optimizier'lar ve farklı parametreler kullanılmıştır ve aralarında en iyi sonuç verenler seçilmiştir.

YOLOv8 ile röntgen görüntüsünden alt çeneyi kırma işlemi yüksek doğrulukla çalışsa da ikili ve çoklu sınıflandırmada yüksek bir başarı oranı yakalanamamıştır. Bunun sebebi ise veri setinin yeterince büyük olmaması ve röntgen görüntülerinin gözle ayırt edilemeyecek kadar birbirlerine benzemesidir. Bu durum, denediğimiz birçok modelde train kısmının çok yüksek oranlara ulaşırken test kısmının aynı performansı göstermeyip bir süre sonra sabit kalmasıyla anlaşılmıştır. Ve bu sorun, kullandığımız yöntemlerle elimizden geldiğince bastırılmıştır.

- [1] F. Rashidi Ranjbar and A. Zamanifar, “Autonomous dental treatment planning on panoramic x-ray using deep learning based object detection algorithm,” *Multimedia Tools and Applications*, pp. 1–35, 2023.
- [2] D. Suryani, M. Shoumi, and R. Wakhidah, “Object detection on dental x-ray images using deep learning method,” in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1073, 2021, p. 012 058.
- [3] I. E. Hamamci *et al.*, “Diffusion-based hierarchical multi-label object detection to analyze panoramic dental x-rays,” *arXiv preprint arXiv:2303.06500*, 2023.

BİRİNCİ ÜYE

İsim-Soyisim: Ahmet Enis ŞİMŞİR

Doğum Tarihi ve Yeri: 19.07.2000, Manisa

E-mail: enis.simsir@std.yildiz.edu.tr

Telefon: 0551 219 03 65

Staj Tecrübeleri: YTÜ Olasılıksal Robotik Grubu, MentalUp Şirketi Yazılım Departmanı, Jena Uygulamalı Bilimler Üniversitesi Elektrik-Elektronik Bölümü Araştırma Laboratuvarı

İKİNCİ ÜYE

İsim-Soyisim: Çağatay Alptekin

Doğum Tarihi ve Yeri: 24.03.2000, Ankara

E-mail: cagatay.alptekin@std.yildiz.edu.tr

Telefon: 0539 438 01 28

Staj Tecrübeleri: Volosoft Yazılım Departmanı

Proje Sistem Bilgileri

Sistem ve Yazılım: Windows İşletim Sistemi, Python, Ultralytics YOLOv8

Gerekli RAM: 1GB

Gerekli Disk: 256MB