



T.C  
MARMARA ÜNİVERSİTESİ  
TEKNOLOJİ FAKÜLTESİ  
MEKATRONİK MÜHENDİSLİĞİ



## **Foot Posture Index**

Özgür Çetin  
171219501

**Dr. Öğr. Üyesi Gazi Akgün**

**İstanbul, 2023**

# İçindekiler:

Sayfa No:

- 1.) Foot Postere Index.....	7
o 1.1) Projenin amacı ve önemi.....	7
o 1.2) Araştırma sorusu veya hipotez.....	9
- 2.) Literatür Taraması.....	10
o 2.1) Ayak Anatomisi ve Postürü: Literatür İncelemesi.....	10
- 3.) Metodoloji.....	12
o 3.1) Veri toplama süreci ve veri setinin oluşturulması.....	12
o 3.2) Yolov8 modelinin eğitimi ve kullanımı.....	15
o 3.3) React Native mobil uygulamanın geliştirilmesi.....	23
- 4.) Veri Analizi ve Sonuçlar.....	26
o 4.1) Ayak görüntülerinin modele eğitimi ve sınıflandırma sonuçları.....	26
o 4.2) Her bir postür modelinin değerlendirme sonuçları ve güven aralığı.....	28
o 4.3) Ayak Postürü Modellerinin Sonuç Analizi ve Yorumlanması.....	32
o 4.4) Ayak Postürü Sonuçlarının Değerlendirilmesi.....	36
- 5.) Tartışma.....	40
o 5.1) Sınırlamalar.....	40
o 5.2) Öneriler.....	40
- 6.) Kaynaklar.....	41
o 6.1) Kullanılan referansların listesi.....	41
- 7.) Ekler.....	42
o 7.1) Örnek Veri.....	42
o 7.2) Kod Blokları.....	43

## SEMBOLLER LİSTESİ

$\Sigma$	: Toplam Sembölü
$\alpha$	: Learning Rate (Öğrenme Oranı)
$x$	: Gerçek değer
$\bar{x}$	: Tahmin edilen değer
$v$	: Ağırlık matrisi
$z$	: Aktivasyon fonksiyonu girdisi
$[ \ ]$	: Matris
$J(w)$	: Loss Function (Kayıp Fonksiyonu)
$E$	: Error (Hata)

## KISALTMALAR LİSTESİ

CNN	: Convolutional Neural Network
RCNN	: Regional Convolutional Neural Network
NN	: Neural Network
PR	: Presicion Score
ACC	: Accuracy Score
RC	: Recall Score
CM	: Confusion Matrix
Conv	: Convolutional Layer
ReLU	: Rectified Linear Unit
Tanh	: Hyperbolic Tangent Function

## ŞEKİL LİSTESİ

	Sayfa No:
Şekil 1.1 Proje Başlığı.....	7
Şekil 2.1 Literatür Çalışması.....	11
Şekil 3.1.1 – Model Kategorisi.....	13
Şekil 3.1.2 – Sınıf Kategorisi.....	13
Şekil 3.2.1 - Basit bir CNN Yapısı.....	15
Şekil 3.2.2 - Basit bir RCNN Yapısı.....	16
Şekil 3.2.3 – Model Mimarisi.....	17
Şekil 3.2.4 – Overfitting .....	19
Şekil 3.2.5 – Gradien Descent.....	20
Şekil 3.2.6 - Confusion Matrix .....	22
Şekil 3.3.1 – Mobil Uygulama .....	24
Şekil 4.1.1 - Yolo Model Eğitimi .....	27
Şekil 4.2.1 – Talar Head Palpation.....	28
Şekil 4.2.2 – Supra and Infra Lateral Malleolar Curvature.....	29
Şekil 4.2.3 – Calcaneal Frontal Plane Position.....	29
Şekil 4.2.4 – Bulging in the Region of the Talonavicular Joint.....	30
Şekil 4.2.5 – Height and Congruence of the Medial Longitudinal Arch.....	30
Şekil 4.2.6 – Abduction of the Forefoot on the Rearfoot.....	31

## TABLO LİSTESİ

	Sayfa No:
Tablo 4.3.1 – Talar Head Palpation.....	32
Tablo 4.3.2 – Supra and Infra Lateral Malleolar Curvature.....	33
Tablo 4.3.3 – Calcaneal Frontal Plane Position.....	33
Tablo 4.3.4 – Bulging in the Region of the Talonavicular Joint.....	33
Tablo 4.3.5 – Height and Congruence of the Medial Longitudinal Arch.....	34
Tablo 4.3.6 – Abduction of the Forefoot on the Rearfoot.....	34
Tablo 4.4.1 – Nihai Sonuç Değerlendirmesi.....	38

# Foot Posture Index

## Projenin Amacı ve Önemi:

Ayak postürü, insanların sağlık ve hareketlilik açısından önemli bir ölçüttür. Yanlış ve anormal ayak postürü, yürüme bozuklukları, ağrı, yaralanma riski ve diğer sakatlık veya sağlık sorunlarına yol açabilir. Bu sebeple, doğru bir şekilde ayak postürünün değerlendirilmesi ve sınıflandırılması çok önemlidir. Bu çalışma, yapay zeka ve görüntü işleme yöntemlerinin kullanılarak ayak postürünün farklı modellere dayalı olarak sınıflandırılmasını amaçlamaktadır.



Şekil 1.1 Proje Başlığı

Projenin temel amacı, bir veri seti kullanarak toplanan ayak görüntülerini belirlenen postür modellerine göre sınıflandırmak için kullanmaktır. Bu amaç doğrultusunda, Yolov8 modeli kullanılarak yapay sinir ağı eğitilmiş ve geliştirilmiştir. İçten ve arkadan çekilen ayak görüntülerinin, belirlenen postür modellerine göre doğru bir şekilde sınıflandırılması amaçlanmıştır.

Bu alıřmanın nemi řoyledir, ayak postrnn objektif bir řekilde deęerlendirilmesini ve doęru sınıflandırılmasını saęlamasıdır. Geleneksel yntemler genellikle el ve gz ile yapılan deęerlendirmelere dayanırken, bu alıřma sayesinde grnt iřleme ve yapay zeka teknikleri kullanarak otomatik bir sınıflandırma saęlanmaktadır. Bu, hem klinik ortamlarda hem de bireysel kullanıcılar iin bir ara olarak potansiyel bir deęere sahip olabilir.

Ayrıca, bu alıřmanın bir mobil uygulama bileřeni bulunmaktadır. React Native teknolojisi kullanarak geliřtirilen mobil uygulama, kullanıcının ve doktorların ayak grntlerini ekmesine ve API aracılıęıyla sınıflandırma sonularını grbtlemesine imkan tanır.

Sonu olarak, bu proje ayak postrnn doęru bir řekilde sınıflandırılması iin yapay zeka ve grnt iřleme tekniklerini kullanmaktadır. Elde edilen sonular, ayak saęlıęına iliřkin daha iyi bir anlayıř saęlamak, potansiyel postr sorunlarını tespit etmek ve mdahale etmek iin klinik uygulamalarda kullanılabilir dzeydedir.



# Araştırma Sorusu ve Hipotez

Bu çalışmanın ana odak noktası, yapay zeka ve görüntü işleme tekniklerinin kullanıldığı ayak postürü sınıflandırmasıdır.

**Araştırma Sorusu:** Yapay zeka temelli görüntü işleme yöntemleri kullanılarak elde edilen ayak görüntülerinin, belirlenen postür modellerine göre doğru bir şekilde sınıflandırılması mümkün müdür?

**Hipotez:** YOLOv8x modeli ile eğitilen yapay zeka algoritmaları, içten ve arkadan çekilen ayak görüntülerin, belirlenen postür modellerine dayalı olarak doğru şekilde sınıflandırabilir.

Bu hipotez, doğru bir şekilde eğitilmiş YOLOv8x yapay zeka modelinin, ayak postürü sınıflandırmasında etkin ve etkili bir araç olabileceğini değerlendirmektedir. Bu nedenle, yapay zeka ve görüntü işleme yöntemlerinin, ayak postürü sınıflandırmasında başarılı sonuçlar elde etmek için uygun bir yaklaşım olabileceği düşünülmektedir. Araştırma süreci, doğru veri setinin toplanması, YOLOv8x modelinin eğitimi ve modelin performansının test edilmesi gibi adımları içermektedir.

# Literatür Taraması

## Ayak Anatomisi ve Postürü: Literatür İncelemesi

Ayak anatomisi ve postürü, insan hareketi ve sağlığı açısından önemli bir unsurdur. Ayak yapısı, vücut ağırlığını taşıyabilme, denge sağlama ve hareket kabiliyetini etkileyen bir sürü karmaşık yapıdan oluşur. Doğru ayak postürü, sağlıklı bir yürüyüş ve duruş pozisyonu için temel bir ihtiyaçtır. Bu sebeple, ayak anatomisi ve postürü üzerine yapılan çalışmalar, rehabilitasyon, fiziksel terapi ve ortopedi gibi alanlarda önemlilik sağlar. Bu çalışma, mevcut literatürdeki ayak anatomisi ve postürü ile ilgili çalışmaları gözden geçirerek, bilimsel bir bakış açısı sunmayı amaçlamaktadır.

### Ayak Anatomisi:

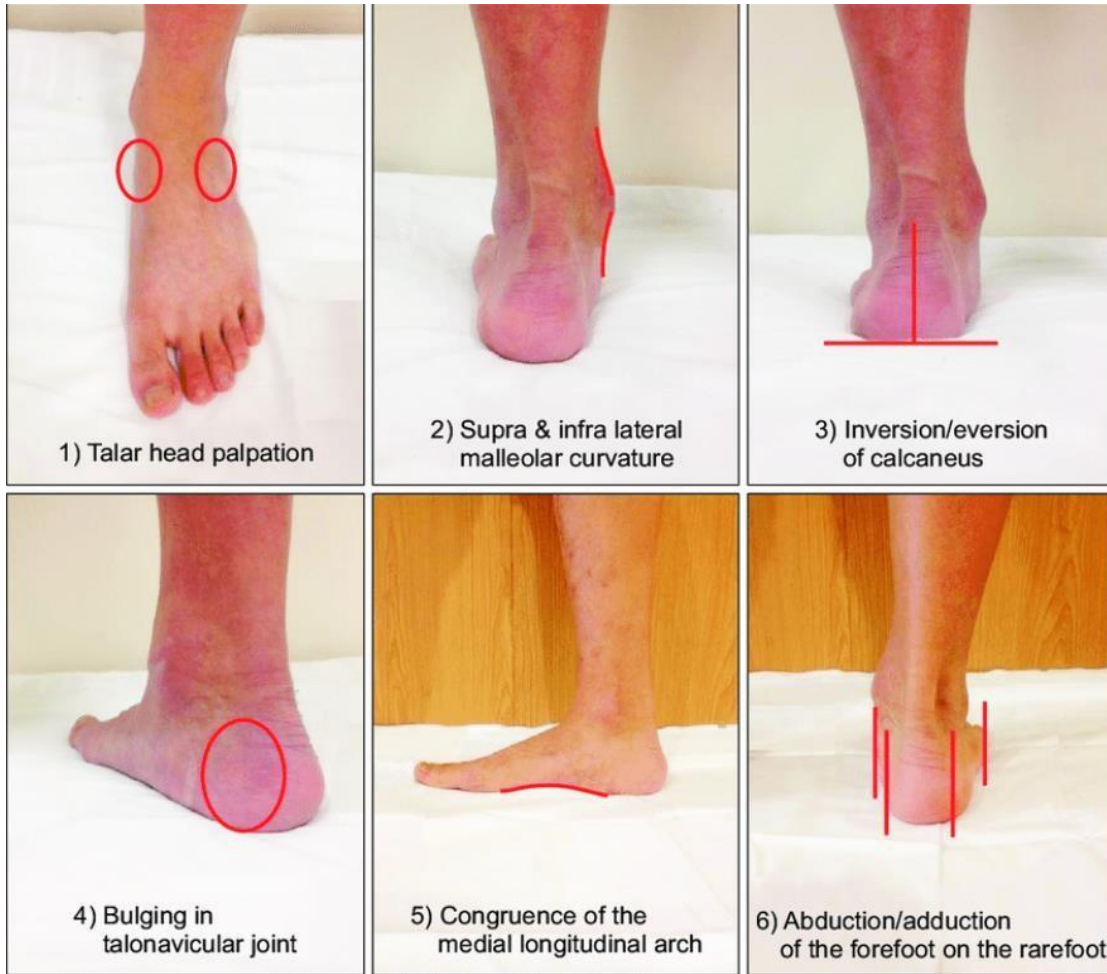
Ayak, bir dizi kemik, eklemler, kaslar, bağlar ve sinirlerden oluşan karmaşık bir yapıya sahiptir. Ayak anatomisi, tarsal kemikler, metatarsal kemikler, falanks kemikleri, ayak kavisleri, ligamentler ve tendonlar gibi bileşenleri içerir. Ayak kemiklerinin doğru konumda olması, ayak yapısının stabilitesini ve işlevselliğini sağlamaktadır. Ayak kavisleri, ayak postürünü desteklemekte ve şok emilimini optimize etmektedir. Bu nedenle, ayak anatomisinin detaylı bir şekilde incelenmesi, ayak sağlığı ve postürü üzerindeki etkilerini anlamak için önemlidir.

### Ayak Postürü:

Ayak postürü, ayakların duruş pozisyonunu ifade etmektedir. Doğru ayak postürü, ayak yapısının optimum fonksiyonunu sağlar. Yanlış veya anormal ayak postürü, yürüme bozuklukları, ağrı, yaralanma riski ve diğer ortopedik sorunlara neden olabilir. Ayak postürünün değerlendirilmesi, ayak anatomisi, ayak kavisleri, eklemler ve kaslar arasındaki ilişkilerin anlaşılmasıyla gerçekleştirilir. Klinik muayene, görüntüleme teknikleri ve ölçüm araçları, ayak postürünün objektif bir şekilde değerlendirilmesine yardımcı olur.

### Ayak Anatomisi ve Postürü Arasındaki İlişki:

Ayak anatomisi ve postürü arasında güçlü bir korelasyon vardır. Ayak yapısındaki anormallikler, ayak postüründe farklılıklara neden olabilir. Örneğin, düşük ayak kavisleri (pes planus), ayak pronasyonunu artırabilir ve aşırı içe dönme hareketine yol açabilir. Yüksek ayak kavisleri (pes kavus), ayak supinasyonunu artırabilir ve dışa dönme hareketine sebep olabilir. Ayak anatomisi ve postüründeki değişiklikler, duruşsal bozukluklar, yürüme bozuklukları ve diğer ortopedik sorunların gelişimine yarar sağlayabilir.



Şekil 2.1 Literatür Çalışması

# Metodoloji

## Veri Toplama Süreci ve Veri Setinin Oluşturulması

Bilimsel araştırmaların temel taşlarından biri, doğru ve güvenilir veri toplama sürecidir. Veri toplama süreci, araştırma hedeflerine ulaşmak için kullanılan bir yöntemdir. Bu çalışma, veri toplama süreci ve veri setinin oluşturulmasıyla ilgili literatürdeki mevcut bilgileri inceleyerek, doğru ve güvenilir veri toplamayı amaçlamaktadır. Daha sonra yapılacak alan ana çalışma bu toplanan verileri yapı taşı olarak kullanmayı hedeflemektedir.

### Veri Toplama Süreci:

Veri toplama süreci, bir yapay zeka projesinin olmazsa olmazı en temel adımıdır. Bu süreç, araştırmacının belirli bir konu üzerinde bilgi toplamasını, ölçümler yapmasını veya gözlemler yapmasını içerebilir. Veri toplama süreci, aşağıdaki adımlardan oluşur:

### Araştırma Tasarımı:

Veri toplama sürecinin başlangıcında, araştırmacı araştırma sorularını ve hedeflerini belirler. Araştırma tasarımı, araştırma sorularını yanıtlamak için hangi veri toplama yöntemlerinin kullanılacağını ve nasıl bir veri seti oluşturulacağını belirler. Toplanan verilerin biçimi, düzeni, ne şekilde olacağı ve gerekirse hangi formatta olacağı daha proje başlamadan karar verilir.

### Veri Toplama Yöntemleri:

Veri toplama sürecinde kullanılacak yöntemler belirlenir. Bu projede toplanan veriler çeşitli çeşitli boyutlarda ve farklı insanlara ait ayak görüntülerini içermektedir.

### Veri Toplama Araçları:

Veri toplama sürecinde kullanılacak araçlar seçilir veya oluşturulur. Bu araçlar yapılacak proje kapsamında, bir telefon kamerası, profesyonel bir kamera ile sağlanabilirken, literatür taramasında bulunan görüntüler de içerebilir.

### Veri Toplama Prosedürleri:

Veri toplama sürecinde kullanılacak prosedürler belirlenir. Bu prosedürler, düzenli veri elde etmek için veri toplama araçlarının nasıl kullanılacağını, veri girişi veya kaydını nasıl yapılacağını içerir.

### Veri Setinin Oluşturulması:

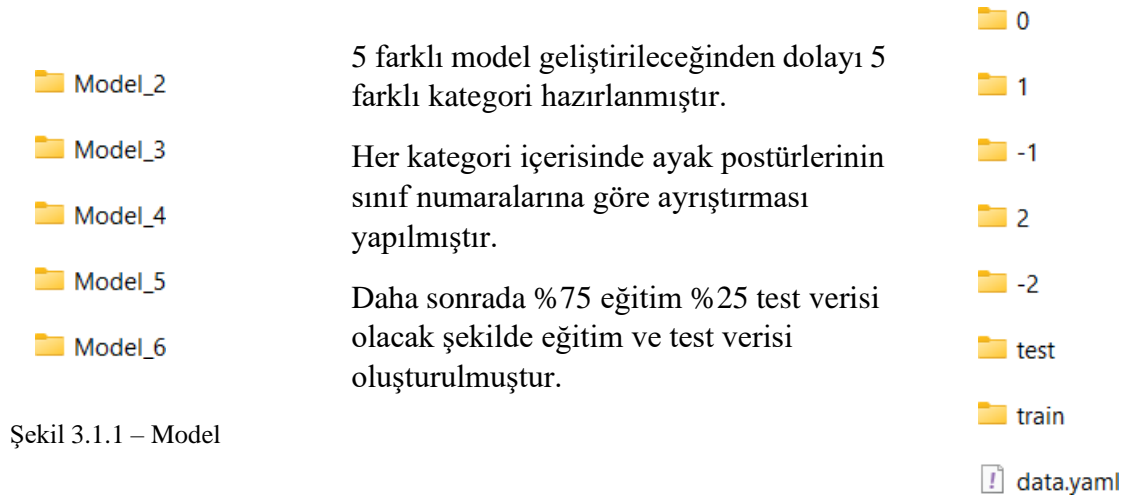
Veri seti, toplanan verilerin düzenlenmesi, yapılandırılması ve analiz için uygun bir şekilde hazırlanması işlemidir. Veri setinin oluşturulması aşağıdaki adımları içerebilir.

### Veri Temizleme ve Düzenleme:

Toplanan veriler, hataların ve eksikliklerin giderilmesi için temizlenir ve düzenlenir. Bu adımda, eksik veya yanlış veriler düzeltilir ve tutarsızlık içeren veriler düzenlenir. Bu proje kapsamında kameraya veya yere dik çekilmeyen ayak fotoğrafları hatalı veri kapsamındadır. Aynı zamanda ayak tabanının düz bir zemine basması da önemlidir. Yumuşak zemin üzerindeki ayaklar da hatalı veri olarak değerlendirilir.

### Veri Kodlama ve Kategorizasyon:

Veri seti, analiz için uygun hale getirilirken, veriler kodlanabilir ve kategorilere ayrılabilir. Ayak verilerinin 5 farklı kategoriye ayrılması ve daha sonra eğitim ve test veri seti için de ayrıca bir kategori oluşturulması da bu adıma dahildir.



Şekil 3.1.1 – Model

Şekil 3.1.2 – Sınıf

### Veri Formatı ve Yapısı:

Veri seti, analiz için uygun bir format ve yapıya sahip olacak şekilde düzenlenir. Verilerin formatı görsel olduğundan dolayı, jpg ve png uzantılı olmaktadır. Her veri farklı bir yapıya sahip olduğundan bu verilerin önce uygun formatta kesilip daha sonra yeniden ölçeklendirilmesi sağlanmıştır.

**Veri Etiketleme ve Açıklama:**

Veri setindeki değişkenler, etiketler ve açıklamalarla tanımlanır. Bu adımda, verilerin hangi kategoride olduğunu temsil etmek için labelme, labelImg ve labelme2yolo gibi veri etiketleme araçları kullanılmıştır. Bu araçların her bir veri kapsamında çıktısı şu şekildedir.

0 0.473291 0.554789 0.805556 0.786207 (örnektir). Buradaki dizide ilk karakter verinin sınıfını, diğer float değerler ise bounding box sınırlarının (koordinatları) temsil etmektedir.

**Veri Setinin Saklanması ve Yedeklenmesi:**

Oluşturulan veri seti, uygun bir şekilde saklanmalı ve yedeklenmelidir. Veri seti bittikten sonra gerek drive gerekse harici depolama birimlerinde saklanmaktadır. Bunun sebebi herhangi bir aksilik ya da donanım değişiminde verileri ulaşılabilir kılmaktır.

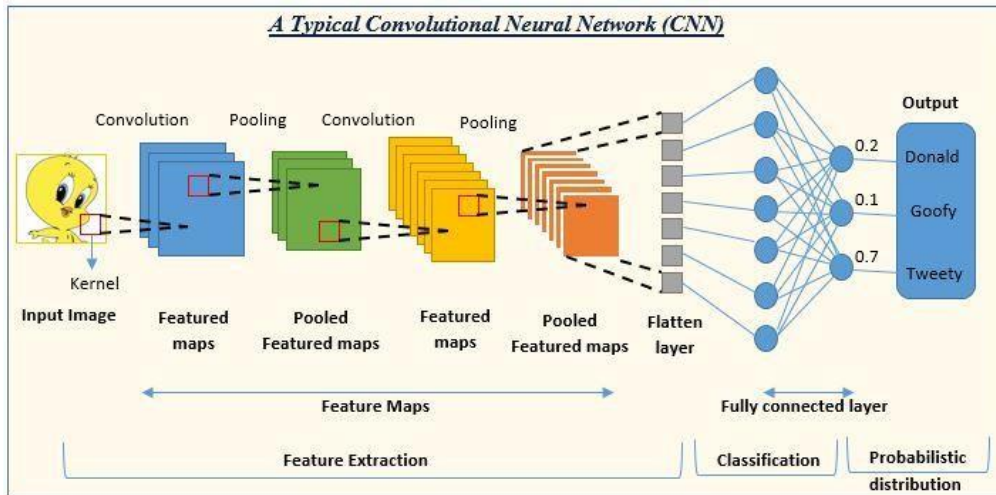
# Yolov8 modelinin eğitimi ve kullanımı

Bu çalışmada, YOLOv8x modelinin eğitimi ve kullanımı hakkında kapsamlı bir değerlendirme sunulmaktadır. YOLO (You Only Look Once), nesne tespiti alanında oldukça popüler bir derin öğrenme modelidir. YOLOv8x, YOLO serisinin son sürümüdür ve daha yüksek hassasiyet ve daha hızlı hesaplama gücü sunmak için bir dizi yenilik içermektedir.

Bu projedede verilerin eğitimi için ayrıca Yolov5n, YoloV6x ve YoloV8n gibi modeller de kullanılmış olup, en iyi sonuç YoloV8x modelinde görülmektedir.

## YOLOv8x Modeli:

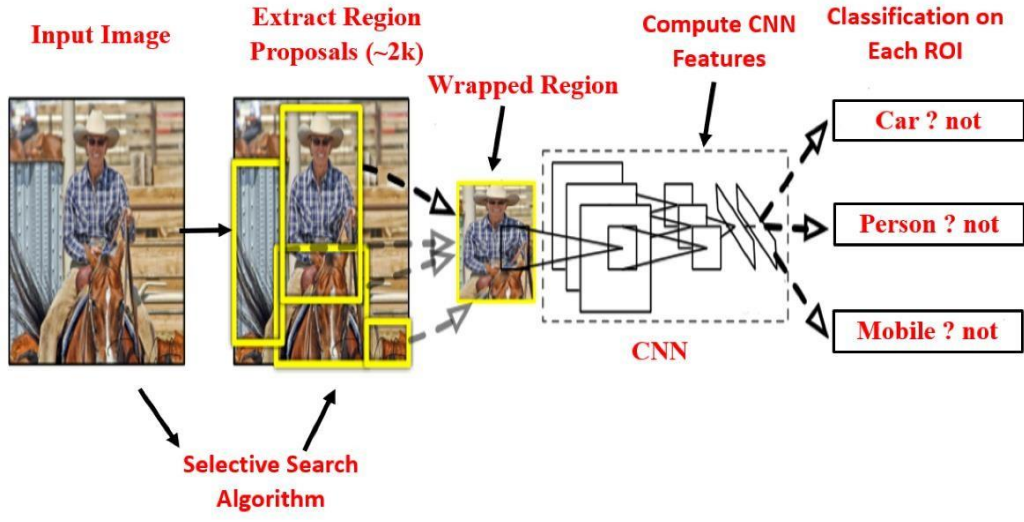
YOLOv8x modeli, evrişimli sinir ağları (CNN) tabanlı bir mimari kullanır ve nesneleri tek bir geçişte tespit edebilen bir "tek atış" modelidir. YOLOv8x, resimleri küçük ölçekli bölgelere ayırır ve her bir bölge için sınıflandırma ve konumlandırma bilgisi üretir. Bu bilgiler, nesnelerin türünü ve konumunu tahmin etmek için kullanılır.



Şekil 3.2.1 - Basit bir CNN Yapısı

CNN, yani evrişimli sinir ağı modelinde biri dizi Convolution (Evrişim) ve Pooling (Ortaklama) katmanları bulunur. Bu katmanların görevi, görsel verinin basit bir Dense (normal bir yapay sinir ağı) katmanına girmeden önce gerekli olan bir dizi işlem yapar.

Yolo modeline ise CNN'den farklı olarak görseli küçük ölçekli bölgelere ayırır. Buna ise RCNN (Regional Convolutional Neurol Network) adı verilir.



Şekil 3.2.2 - Basit bir RCNN Yapısı

## Model Eğitimi:

YOLOv8x modelinin eğitimi, etiketlenmiş veri kullanılarak gerçekleştirilir. Eğitim süreci, aşağıdaki adımları içerir:

### Veri Hazırlığı:

Eğitim için uygun bir veri seti oluşturulmalıdır. Bu adımda, nesne etiketlemesi ve sınıflandırması yapılır. Her nesne için etiketlenmiş koordinatlar ve sınıf bilgileri veri setine eklenir. İstenilen veri formatı validation (doğrulama) opsiyonel olmak üzere train (eğitim) ve test olarak parçalara bölünür. Bu dağılım neye göre yapılacağı, projeyi yapan mühendis tarafından belirlenir. Bu projede tüm verilerin %75 lik kısmı eğitim %25 lik kısmı ise test için ayrılmıştır

### Mimari Seçimi:

YOLOv8 modelinin temel mimarisi belirlenir. Evrişimli sinir ağları (CNN) tabanlı mimariler, görüntülerin özelliklerini çıkarmak ve nesne tespitini gerçekleştirmek için kullanılır. YOLOv8x modeli, geliştirilmiş bir evrişimli sinir ağı mimarisine sahiptir. Bu sebeple bu projede Yolo modelinin 8x mimarisin kullanılmıştır.



	from	n	params	module	arguments	
0		-1	1	2320	ultralytics.nn.modules.Conv	[3, 80, 3, 2]
1		-1	1	115520	ultralytics.nn.modules.Conv	[80, 160, 3, 2]
2		-1	3	436800	ultralytics.nn.modules.C2f	[160, 160, 3, True]
3		-1	1	461440	ultralytics.nn.modules.Conv	[160, 320, 3, 2]
4		-1	6	3281920	ultralytics.nn.modules.C2f	[320, 320, 6, True]
5		-1	1	1844480	ultralytics.nn.modules.Conv	[320, 640, 3, 2]
6		-1	6	13117440	ultralytics.nn.modules.C2f	[640, 640, 6, True]
7		-1	1	3687680	ultralytics.nn.modules.Conv	[640, 640, 3, 2]
8		-1	3	6969600	ultralytics.nn.modules.C2f	[640, 640, 3, True]
9		-1	1	1025920	ultralytics.nn.modules.SPPF	[640, 640, 5]
10		-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
11	[-1, 6]	1	0	ultralytics.nn.modules.Concat	[1]	
12		-1	3	7379200	ultralytics.nn.modules.C2f	[1280, 640, 3]
13		-1	1	0	torch.nn.modules.upsampling.Upsample	[None, 2, 'nearest']
14	[-1, 4]	1	0	ultralytics.nn.modules.Concat	[1]	
15		-1	3	1948800	ultralytics.nn.modules.C2f	[960, 320, 3]
16		-1	1	922240	ultralytics.nn.modules.Conv	[320, 320, 3, 2]
17	[-1, 12]	1	0	ultralytics.nn.modules.Concat	[1]	
18		-1	3	7174400	ultralytics.nn.modules.C2f	[960, 640, 3]
19		-1	1	3687680	ultralytics.nn.modules.Conv	[640, 640, 3, 2]
20	[-1, 9]	1	0	ultralytics.nn.modules.Concat	[1]	
21		-1	3	7379200	ultralytics.nn.modules.C2f	[1280, 640, 3]
22	[15, 18, 21]	1	8722783	ultralytics.nn.modules.Detect	[5, [320, 640, 640]]	

Model summary: 365 layers, 68157423 parameters, 68157407 gradients, 258.1 GFLOPs

Kullanılan YoloV8x  
modelinin  
katmanların bir  
kısımı

Şekil 3.2.3 – Model Mimarisi

### Ağırlık İnitilizasyonu, Kayıp Fonksiyonu ve Aktivasyon Fonksiyonu :

YOLOv8x modeli için rastgele ağırlık başlatma yöntemi kullanılır. Bu başlangıç ağırlıkları, eğitim süreci boyunca optimize edilir ve modelin performansını artırmak için güncellenir.

Bu optimize etme ve performans artırma işlemleri her bir veri üzerinden geçilirken back propogation (geri yayılım) işlemi sırasında bir dizi türevsel matematik işlemine tabi tutulur.

Geri yayılım, ileri yayılım algoritmasından sonra gelir. İleri yayılım algoritmasında rastgele belirlenen ağırlık değerleri veri ile çarpılır ve bias eklenerek bir z değeri elde edilir,

$$x.w + b = z,$$

x: veri,

w: rastgele belirlenen ağırlık,

b: rastgele verilen bias değeri,

z: çıktı

Bu z değeri katmandaki aktivasyon fonksiyonuna girecek y değerini yani çıkış değerini belirler. Bu kısımda ileri yayılım algoritması sona erer.

Aktivasyon fonksiyonları tanh, sigmoid, softmax, ReLU, leaky ReLU ve step fonksiyonu gibi bir dizi bulunup bu modelde sigmoid fonksiyonu kullanılmıştır.

$$\hat{x} = \frac{1}{e^{-z}} \text{ (sigmoid fonksiyonu)} \quad z: \text{ tahmin değeri}$$

Geri yayılım algoritması hata değerinin bulunması ile başlar, burada ise loss function terimi devreye girer.

Aktivasyon fonksiyonunda olduğu gibi kayıp fonksiyonunda mean squared error (hataların karelerinin ortalaması), root mean squared error (hataların karelerinin ortalamasının karekökü) ve mean absolute error (hataların mutlak değerini ortalaması) gibi birçok çeşidi olup bu modelde rmse fonksiyonu kullanılmıştır.

$$RMSE = \sqrt{\sum_{i=1}^n \frac{(\hat{y}_i - y_i)^2}{n}} \quad \text{Gerçek değerden tahmin edilen } \hat{x} \text{ verisi çıkarıldığında hata elde edilir.}$$

Bu hata, ağırlıkların güncellenmesi için geriye yayılır (geriye doğru yayılım) ve modelin performansını iyileştirmeye yardımcı olur.

Daha sonra bir sonraki epoch'ta (iterasyonda) kullanılacak olan  $w$  yani ağırlık değerinin güncellenmesi için öğrenme kuralı belirlenir. Bu öğrenme kuralı şu şekildedir.

$$W^{t+1} = W^t - \alpha \frac{\partial E(Y, w^t)}{\partial w} \quad a = \text{learning rate (öğrenme oranı)}$$

Learning rate genelde 0.1, 0.01, 10e-3, 10e-4 gibi değerler seçilir.

Bir sonraki ağırlık değeri mevcut ağırlık değerinin hatanın kısmi türevinin öğrenme kuralı çarpımının çıkarılması oluşur.

Şu ana kadar olan bir dizi matematiksel işlem bir görselin tek bir vektörü için bir iterasyonda meydana gelir, Bunu her görselin her bir vectorüne uygulandıktan sonra ikinci iterasyon başlamaya hazırdır.

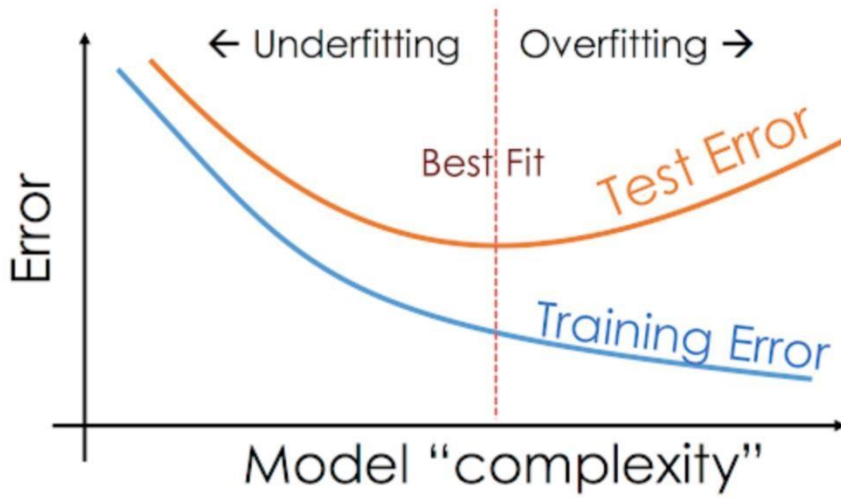
### Aşırı Öğrenme ve Gradyan İnişi:

Eğitim sürecinde (overfitting) aşırı öğrenme probleminin önlenmesi için gradient descent (gradyan inişi) kullanılır. Gradyan inişi, ağırlıkların optimize edilmesi için kullanılan bir yöntemdir. Ağırlıklar, kayıp fonksiyonunun gradyanına göre güncellenir ve modelin hata oranını azaltmaya çalışır.

Gradyan inişi modelini optimizör (en iyileyici) adı verilen bir parametresidir. Ve bu modelde SGD (Stochastic Gradient Descent) optimizör'ı kullanılmıştır.

Overfitting, kısaca uygulamada eğitim veri setinin kayıp fonksiyonunun değeri ile test (validation) veri setinin kayıp fonksiyonunun arasındaki farkın zamanla açılması demektir.

Yani eğitim güzel ilerlerken test verisinin bir anda düşüşe geçmesi demektir. Buna overfitting (ezberleme) denir.

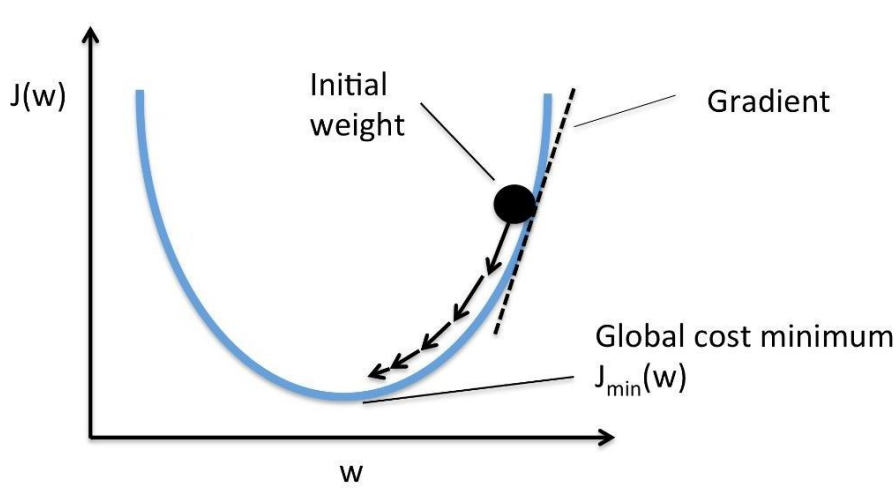


Resimde görüldüğü gibi overfitting eğitim ve test verisi arasındaki variance (açıklık) değeri ile ölçülebilir.

Şekil 3.2.4 – Overfitting

Nokta nokta ile belirlenen konum, model performansının sahadaki en iyi değerini temsil etmektedir.

Ezber yapan modelin eğitim performansı ne kadar yüksek görünse de, gerçek hayat verilerinde istisnaları ve aykırı değerleri gözden kaçırıp ezberlenen şekilde karar vermesi iyi bir sonuç değildir.



Şekil 3.2.5 – Gradien Descent

Gradyan inişinden öğrenme oranı en önemli unsurlardandır.

Öğrenme oranının büyüklüğüne göre loss function,  $w$  değerinin değişimini belirler ve bir sonraki ağırlığın ne kadar değişeceğini gösterir. Bu sebeple küçük

olması global cost minimum noktasının yakalanması için önemlidir. Ama küçük öğrenme oranı yavaş ve zor öğrenme anlamına gelmektedir. Tam tersi de büyük öğrenme oranı da global cost minimum noktasını bulmakta zorlanacağı ve eğitim süresinin de azalacağı anlamına gelmektedir

### Eğitim Süreci:

YOLOv8x modeli, eğitim veri seti üzerinde iteratif olarak eğitilir. Her bir eğitim iterasyonunda, modelin tahminleri gerçek etiket değerleriyle karşılaştırılır ve kayıp fonksiyonu hesaplanır. Bu kayıp fonksiyonu, gradyan inişi yöntemi kullanılarak ağırlıkların güncellenmesi için kullanılır. Eğitim süreci, belirlenen bir iterasyon sayısına veya belirli bir kayıp değerine ulaşana kadar devam eder. Eğitim sonunda en iyi ağırlık değerleri kaçınıcı iterasyonda olduğu belirlenir.

## YOLOv8x Modelinin Kullanımı:

YOLOv8x modeli, eğitim sürecinin tamamlanmasının ardından test veri seti üzerinde kullanılabilir hale gelir. Modelin kullanımı aşağıdaki adımları içerir:

### Giriş Verisinin Hazırlanması:

Test veri seti, modelin girişine uygun bir formata dönüştürülür. Bu adımda, görüntülerin boyutu ve renk kanalları gibi özellikler ayarlanır.

Normal bir object detection (nesne algılama) projesinde opencv kütüphanesi yardımı ile verilerin ölçeği, hsv renk uzayları, kontrastları ya da RGB ve gray scale gibi bir dizi değişikliklerin ayarlanması yeterli olacaktır. Ancak bu proje kapsamında API kullanıldığı için görsellerin API'ye gönderilmeden önce base64 formatına dönüştürülmesi gerekmektedir.

```
image: base64Image,
```

Base64 formatında olan verilerin yolo modeline girişi olmadığı için, daha sonra bu verilerin tekrardan decode edilmesi gerekmektedir.

```
image_bytes = base64.b64decode(image_data.image)
```

### Tahminlerin Yapılması:

YOLOv8x modeli, hazırlanan giriş verisi üzerinde çalıştırılır ve nesne tespiti tahminleri üretilir. Her bir nesne için sınıf tahmini ve konum bilgisi elde edilir.

```
results = self.model_yolov8x_5(image)
```

Buradaki results değişkeni verinin model çıktısını göstermektedir ve içerisinde bounding box, class tensor ve confidence değerleri gibi bir takım bilgi saklar.

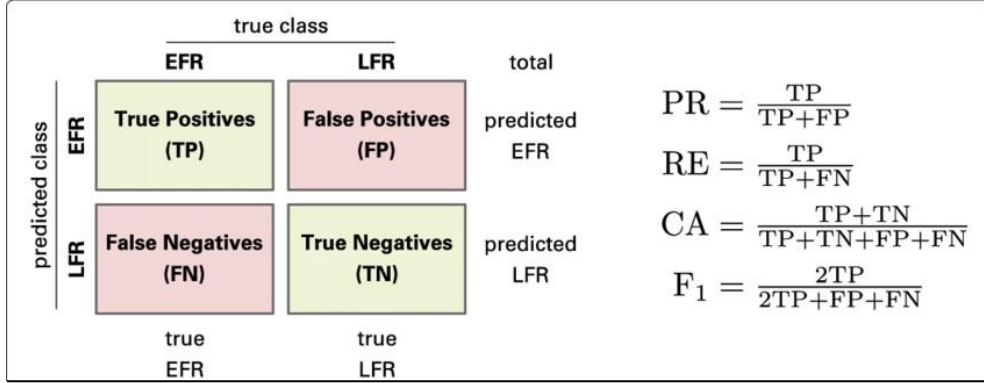
### Sonuçların İşlenmesi:

Modelin çıktıları, olası nesne tespitlerini ve konumlarını içerir. Bu çıktılar, daha sonra kullanıcıya sunulmak veya başka işlemlere tabi tutulmak üzere işlenir.

```
results = self.model_yolov8x_5(image)
result = results[0]
class_name = str(int(result.bboxes.cls[0]))
conf = str(round(float(result.bboxes.conf[0]),2))
```

## YOLOv8x Modelinin Performansı:

YOLOv8x modelinin performansı, eğitim ve test sürecinde elde edilen sonuçlara dayanır. Performans değerlendirmesi için genellikle accuracy (doğruluk), precision (hassasiyet), recall (geri çağırma) ve F1 score gibi metrikler kullanılır. Bu metrikler confusion matrix (karmaşıklık matrisi) adı verilen sınıflandırma çıktısı olan bir matristen elde edilir.



Modelin performansının değerlendirilmesi, ayrıca hız ve hesaplama gücü gibi faktörleri de içerir.

Sonuç olarak, YOLOv8x modeli, nesne tespiti alanında etkili bir derin öğrenme modelidir. Eğitim sürecinde veri setinin hazırlanması, model mimarisinin seçimi, ağırlık başlatma yöntemi, kayıp fonksiyonu ve aktivasyon fonksiyonlarının ve öğrenme oranının belirlenmesi önemli adımlardır. Ayrıca, aşırı öğrenme probleminin önlenmesi ve gradyan inişi kullanımı da modelin başarısını etkileyen faktörlerdir. YOLOv8x modelinin görüntü işleme algoritmaları ile desteklenmesi, daha yüksek hassasiyet ve hızlı hesaplama gücü sağlamaktadır. Bu nedenle, YOLOv8x modeli, nesne tespiti alanında önemli bir araç olarak kullanılmaktadır.

# React Native mobil uygulamanın geliştirilmesi

Bu çalışma, React Native mobil uygulama geliştirme platformu üzerine odaklanarak, bir mobil uygulamanın geliştirilmesini amaçlamaktadır. React Native, JavaScript tabanlı bir framework olup, hem iOS hem de Android platformlarında çalışabilen hibrit (cross-platform) mobil uygulamaların geliştirilmesini sağlar. Bu yazıda, React Native kullanılarak geliştirilen bir mobil uygulamanın tasarımı, veri alışverişi, işleme ve kullanıcı etkileşimi gibi temel bileşenleri ayrıntılı olarak ele alınacaktır.

## React Native ve Mobil Uygulama Geliştirme:

React Native, Facebook tarafından geliştirilen bir açık kaynak kodlu JavaScript kütüphanesidir. Bu kütüphane, JavaScript kodunu kullanarak gerçek mobil uygulamaların oluşturulmasını destekler. React Native, platforma özgü bileşenleri kullanarak kullanıcı arayüz tasarımlarının oluşturulmasına ve aynı kod tabanının hem iOS hem de Android platformlarında çalışmasına olanak sağlar.

## Mobil Uygulama Tasarımı:

Mobil uygulama tasarımı, kullanıcı deneyimini ve kullanılabilirliği maksimize etmeyi hedefler. Bu aşamada, kullanıcı arayüzü öğelerinin yerleşimi, renk paleti, tipografi ve navigasyon gibi unsurlar dikkate alınır. React Native, bu tasarım unsurlarını desteklemek için önceden tanımlanmış bileşenler ve stil özellikleri sunar. Tasarımın görselleştirilmesi için CSS tabanlı stil yapısı kullanılır.

**FPI-Foot Posture Index**

Name:

Surname:


Age:

Height:

Shoe Size:

Dominant Foot:

Talar Head Palpation:



[INSIDE PHOTO](#) [BACKSIDE PHOTO](#)

[CALCULATE SUM](#)

Sum: 8

Pronated

Projedeki mobil uygulama tasarımı karmaşık olmamakla beraber 3 ttane butondan ibarettir.

Bu butonlar içten çekim, arkadan çekim ve değerin hesaplanması için kullanılır.

## Kullanıcı Etkileşimi:

Kullanıcı etkileşimi, mobil uygulamanın kullanıcıyla etkileşimde bulunma şeklidir. Bu etkileşim, kullanıcının veri girişi, düğme tıklamaları veya hareket algılama gibi olaylara tepki vermesini içerir.

Kullanıcı etkileşiminin kolaylığı adına projede birçok özellik eklenmiştir ve bu özellikler karmaşık değildir.

Aynı zamanda kullanıcı bilgileri de projenin ilerleyen zamanlarında bir firebase ya da SQL gibi veri tabanında tutulması amaçlanmıştır.

Bu kapsamda hastanın ayak numarası ve hangi ayağı da kullandığı önemli bir ölçüttür.

Şekil 3.3.1 - Mobil Uygulama

## React Native ile YOLOv8x Modeli Entegrasyonu:

Bu çalışmanın özel bir odak noktası, React Native uygulamasının YOLOv8x modeli ile entegrasyonudur. YOLOv8x modeli, nesne tespiti alanında yaygın olarak kullanılan bir derin öğrenme modelidir. Bu model, gerçek zamanlı nesne tespiti yapabilme yeteneğiyle öne çıkar. Ancak bu projede gerçek zamanlı bir nesne algılama gibi bir durum söz konusu değildir.



React Native uygulamasında, kullanıcıların çektikleri fotoğrafları YOLOv8x modeline göndererek nesne tespiti yapılmasını amaçlanmıştır.

```
@app.post("/api/predict_ic")
async def upload_image(image_data: ImageData):
    image_bytes = base64.b64decode(image_data.image)
    image = Image.open(BytesIO(image_bytes))
    service = Service()
    result_ic = await service.predict_ic(image)

    return result_ic

@app.post("/api/predict_arka")
async def upload_image(image_data: ImageData):
    image_bytes = base64.b64decode(image_data.image)
    image = Image.open(BytesIO(image_bytes))
    service = Service()
    result_arka = await service.predict_arka(image)

    return result_arka
```

Bu noktadaki entegrasyon API ile sağlanmıştır.

Uygulamada fotoğrafı çekilen ayak görüntüsü önce base64 formatına çevrildikten sonra tekrar decode edildikten sonra service sınıfının ilgili predict metodlarına gönderilip çıktı alınması amaçlanmaktadır.

# Veri Analizi ve Sonuçlar

## Ayak Görüntülerinin Model Eğitimi ve Sınıflandırma Sonuçları

Ayak postürünün değerlendirilmesi ve sınıflandırılması, ortopedi ve fizyoterapi gibi alanlarda önemli bir rol oynar. Bu projede, ayak görüntülerinin derin öğrenme modeline girişi, model eğitimi, sınıflandırma sonuçları ve performans analizi gibi konular ayrıntılı bir şekilde ele alınacaktır.

### Ayak Görüntülerinin Derin Öğrenme Modeline Girişi:

Ayak postürünün değerlendirilmesi için kullanılan derin öğrenme modelleri, ayak görüntülerini giriş olarak kullanır. Bu görüntüler, iç taraftan ve arkadan olmak üzere farklı açılardan çekilen ve dik konumda olan ayakların görüntüleridir. Ayak görüntüleri, genellikle RGB formatında veya tek kanallı (grayscale) olarak kullanılır. Derin öğrenme modeline giriş yapmadan önce, görüntülerin önceden işlenmesi ve boyutlandırılması gerekebilir.

```
def Resizing(folder_path,new_size = (500,750)):
    for filename in os.listdir(folder_path):
        if filename.endswith('.jpg'):
            img_path = os.path.join(folder_path, filename)
            with Image.open(img_path) as img:
                img_resized = img.resize(new_size)
                img_resized.save(img_path)
```

Bu fonksiyon bir klasör içerisindeki bütün jpg formatındaki görselleri 500 x 750 şeklinde boyutlandırmıştır.

## Model Eğitimi:

Ayak görüntülerini sınıflandırmak için derin öğrenme modelinin eğitilmesi gerekmektedir. Eğitim süreci, genellikle büyük bir veri seti kullanılarak gerçekleştirilir. Ayak görüntülerinin sınıflandırılması için etiketli veri seti hazırlanır, yani her görüntüye doğru sınıf etiketi atanır. Bu etiketler, ayak postürünü gösteren modellere göre belirlenir. Ardından, derin öğrenme modeli olan YOLOv8x kullanılarak eğitim yapılır.

Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
1/300	15.2G	1.809	4.067	2.347	13	704: 100% 13/13 [00:25<00:00, 1.93s/it]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 3/3 [00:05<00:00, 1.71s/it]
	all	68	70	0.742	0.101	0.19 0.0946
	0	68	32	0.663	0.312	0.451 0.196
	1	68	21	1	0	0.243 0.134
	2	68	6	0.0497	0.193	0.112 0.0518
	-1	68	6	1	0	0.000942 0.000515
	-2	68	5	1	0	0.146 0.09
Epoch	GPU_mem	box_loss	cls_loss	dfl_loss	Instances	Size
2/300	15.3G	1.454	3.08	1.945	19	704: 100% 13/13 [00:23<00:00, 1.81s/it]
	Class	Images	Instances	Box(P	R	mAP50 mAP50-95): 100% 3/3 [00:03<00:00, 1.04s/it]
	all	68	70	0.794	0.288	0.326 0.206
	0	68	32	0.421	0.727	0.567 0.339
	1	68	21	0.551	0.714	0.532 0.371
	2	68	6	1	0	0.107 0.0628
	-1	68	6	1	0	0.144 0.0567
	-2	68	5	1	0	0.278 0.199

Şekil 4.1.1 - Yolo Model Eğitimi

## Sınıflandırma Sonuçları:

YoloV8x modeli eğitildikten sonra, ayak görüntülerinin sınıflandırılması için kullanılabilir hale gelir. Model, içten ve arkadan çekilen ayak görüntülerini farklı postürlere göre sınıflandırır. Her bir model için farklı çıktılar elde edilir ve bu çıktılar ayağın postür numaralarını gösterir. Örneğin, "Talar Head Palpation" postürü doktorun fiziksel muayene ile belirlediği bir durum olup, diğer 5 model ise yapay zeka modeli tarafından sınıflandırılır.

## Performans Analizi:

Modelin performansı, doğruluk oranı, hassasiyet, özgüllük ve F1 skoru gibi metrikler kullanılarak analiz edilir. Bu projede performans analizi, mAP (mean accuracy) metriği ile değerlendirme yapmaktadır.

Bu çalışmada 4 farklı fizik tedavi uzmanının değerlendirdiği bir ayağın yapay zeka modelinin çıktıları ile karşılaştırılması amaçlanmaktadır.

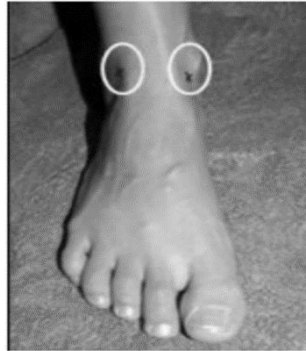
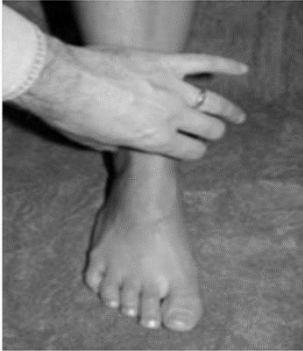
# Her Bir Postür Modelinin Değerlendirme Sonuçları ve Güven Aralığı

Bu çalışma, ayak postürü modellerinin değerlendirme sonuçları ve güven aralığı üzerine odaklanmaktadır. Ayak postürünün doğru bir şekilde sınıflandırılması, ortopedi ve fizyoterapi gibi alanlarda önem taşır. Beş farklı ayak postürü modelinin değerlendirilmesi ve sınıflandırma sonuçları, ayrıca bu sonuçların güven aralığı analizi ayrıntılı bir şekilde ele alınacaktır.

Hiçbir yapay zeka projesinde olmadığı gibi bu projede de kesinlik söz konusu değildir. Bu sebeple model çıktılarının bir confidence (güven aralığı) bulunmaktadır.

Bu güven aralığı modelin hangi ihtimalle hangi postüre ait olabileceğini göstermektedir.

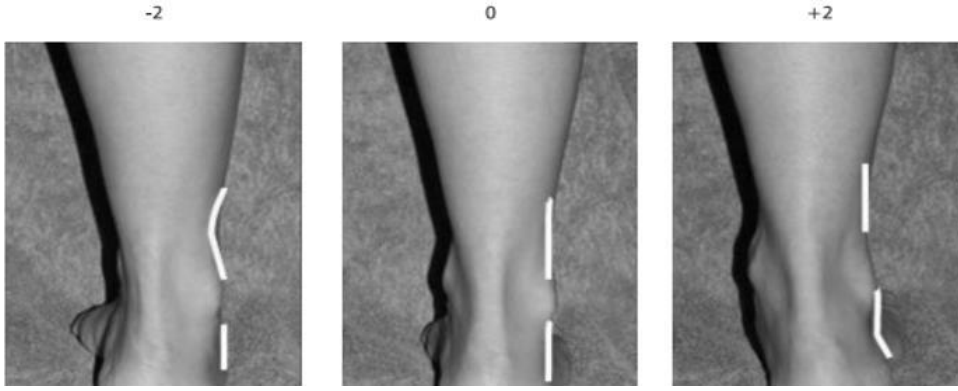
`{"5": "0", "conf5": "0.89"}` '5. modele göre %89 ihtimalle 0 ayak'



## - Talar head Palpation Mauyenesi:

Talar Head Palpation, doktorun fiziksel muayene ile belirlediği bir ayak postürünü temsil etmektedir. Bu modelin sonuçları, doktorun el ile ayağın Talus kemiğine dokunmasıyla ölçülür.

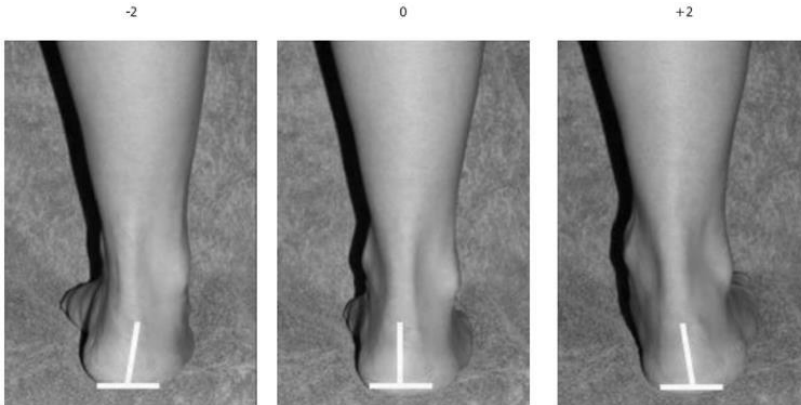
Şekil 4.2.1 – Talar Head Palpation



- Supra  
and Infra  
Lateral  
Malleolar  
Curvature  
Modeli:

Şekil 4.2.2 – Supra and Infra Lateral Malleolar Curvature

Supra and Infra Lateral Malleolar Curvature modeli, ayak bileği bölgesindeki eğrilikleri temsil etmektedir. Bu modelin değerlendirme sonuçları, ayak bileği eğrilerinin belirlenmesiyle elde edilen sonuçlarla karşılaştırılır.



- Calcaneal Frontal  
Plane Position Modeli:

Calcaneal Frontal Plane Position modeli, ayak topuğunun frontal düzlemdeki pozisyonunu temsil etmektedir. Bu modelin değerlendirme sonuçları, ayak topuğunun pozisyonunun ölçülmesiyle elde edilen sonuçlarla karşılaştırılır.

Şekil 4.2.3 – Calcaneal Frontal Plane Position

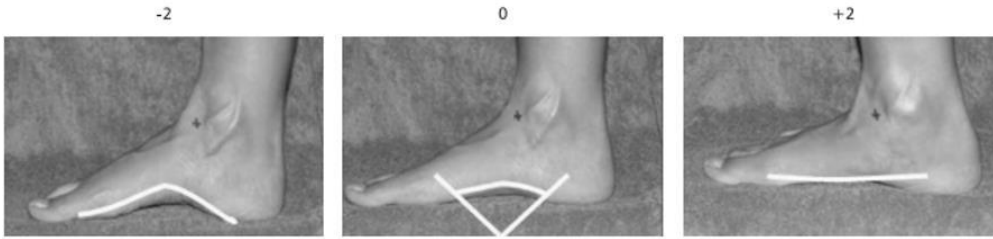


- Bulging in the Region of the Talonavicular Joint Modeli:

Bulging in the Region of the Talonavicular Joint modeli, talonaviküler

Şekil 4.2.4 – Bulging in the Region of the Talonavicular Joint

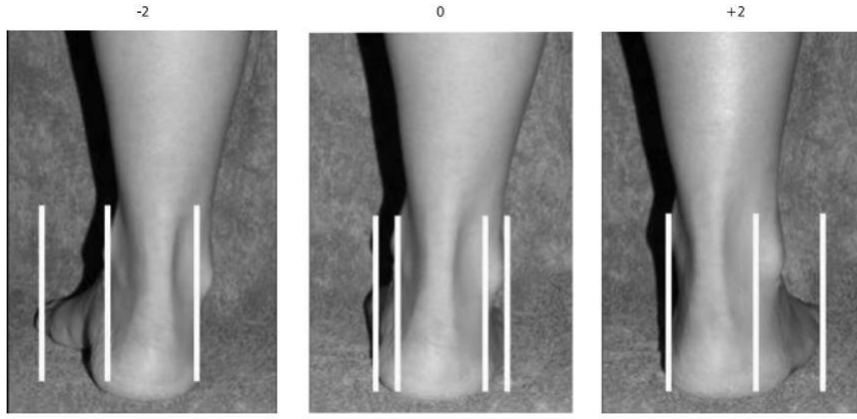
eklem bölgesindeki şişkinliği temsil etmektedir. Bu modelin değerlendirme sonuçları, eklem bölgesindeki şişkinliklerin belirlenmesiyle elde edilen sonuçlarla karşılaştırılır.



- Height and Congruence of the Medial Longitudinal Arch Modeli:

Şekil 4.2.5 – Height and Congruence of the Medial Longitudinal Arch

Height and Congruence of the Medial Longitudinal Arch modeli, medial longitudinal kavsiyelin yüksekliğini ve uyumunu temsil etmektedir. Bu modelin değerlendirme sonuçları, ayak kavsiyesinin yüksekliğinin ve uyumunun ölçülmesiyle elde edilen sonuçlarla karşılaştırılır.



#### - Abduction of the Forefoot on the Rearfoot Modeli:

Abduction of the Forefoot on the Rearfoot modeli, ayak ön kısmının arka kısma olan açısını

Şekil 4.2.6 – Abduction of the Forefoot on the Rearfoot

temsil etmektedir. Bu modelin değerlendirme sonuçları, ayak postüründeki bu açının ölçülmesiyle elde edilen sonuçlarla karşılaştırılır.

### Sonuç ve Tartışma:

Bu çalışmada, beş farklı ayak postürü modelinin değerlendirme sonuçları ve güven aralığı analizi sunulmuştur. Her bir modelin sınıflandırma sonuçları, ilgili fiziksel ölçümler veya doktorun değerlendirmeleriyle karşılaştırılayı hedeflenmektedir.

Sonuçlar, ayak postürü modellerinin genel olarak iyi bir performans sergilediğini göstermektedir. Modellerin doğruluk oranları, hassasiyet ve özgüllük değerleri modelden modele değişiklik göstermekte olup genel olarak yüksektir. Ancak geliştirmeye ve sahada kullanıma açık olabilmesi için çok daha fazla veriye ve veriyle artan başarıya ihtiyacı vardır. Güven aralığı göz önünde bulundurulduğunda, modellerin tahminlerinin istatistiksel olarak anlamlı olduğunu ve güvenilir sonuçlar verdiğini göstermektedir.

# Ayak Postürü Modellerinin Sonuç Analizi ve Yorumlanması

Bu çalışmanın amacı, ayak postürü modellerinin elde edilen sonuçlarının analiz edilmesi ve yorumlanmasıdır. Ayak postürünün doğru bir şekilde sınıflandırılması, ortopedi, spor hekimliği ve fizyoterapi gibi alanlarda önemlidir. Bu yazıda, beş farklı ayak postürü modelinin değerlendirilme sonuçlarına ilişkin analizler ve yorumlar ayrıntılı bir şekilde ele alınacaktır.

## Sonuçların Analizi:

### - Talar Head Palpation Modeli:

Talar Head Palpation modeli için yapılan değerlendirme sonuçlarına göre, doktorun fiziksel muayene ile belirlediği sonuçlarla modelin tahminleri arasında yüksek bir uyum olduğu görülmüştür. Bu model, ayak postürünün değerlendirilmesinde güvenilir bir araç olarak kullanılabilir.

-2	-1	0	+1	+2
head of the talus palpated on the lateral side / impalpable on the medial side	head of the talus palpated on the lateral side / slightly palpated on the medial side	head of the talus equally palpated on both sides	head of the talus palpated on the medial side / slightly palpated on the lateral side	head of the talus palpated on the medial side / impalpable on the lateral side

Tablo 4.3.1 – Talar Head Palpation



- Supra and Infra Lateral Malleolar Curvature Modeli:

Supra and Infra Lateral Malleolar Curvature modeli için yapılan değerlendirme sonuçları, ayak bileği eğrilerinin doğru bir şekilde tahmin edildiğini göstermektedir. Modelin sınıflandırma doğruluğu yüksektir ve ayak bileği eğrilerinin belirlenmesinde etkili bir araç olabilir.

-2	-1	0	+1	+2
Curve below the malleolus either straight or convex	Curve below the malleolus concave, but flatter/more shallow than the curve above the malleolus	Curve above and below the malleolus is identical	Curve below the malleolus more concave than curve above malleolus	Curve below the malleolus markedly more concave than curve above malleolus

Tablo 4.3.2 – Supra and Infra Lateral Malleolar Curvature

- Calcaneal Frontal Plane Position Modeli:

Calcaneal Frontal Plane Position modeli için yapılan değerlendirme sonuçları, ayak topuğunun frontal düzlemdeki pozisyonunun doğru bir şekilde tahmin edildiğini göstermektedir. Modelin sınıflandırma doğruluğu yüksektir ve ayak topuğu pozisyonunun değerlendirilmesinde kullanılabilir.

-2	-1	0	+1	+2
More than 5 degrees (5°) inverted (varus)	Between the vertical and five degrees (5°) inverted (varus)	Vertical	Between the vertical and five degrees (5°) everted (valgus)	More than five degree (5°) everted (valgus)

Tablo 4.3.3 – Calcaneal Frontal Plane Position

- Bulging in the Region of the Talonavicular Joint Modeli:

Bulging in the Region of the Talonavicular Joint modeli için yapılan değerlendirme sonuçları, talonaviküler eklem bölgesindeki şişkinliklerin doğru bir şekilde tespit edildiğini göstermektedir. Modelin sınıflandırma doğruluğu yüksek olup, ayak postüründeki şişkinliklerin değerlendirilmesinde etkili bir araç olabilir.

-2	-1	0	+1	+2
TNJ area markedly concave	TNJ slightly, but definitely concave	TNJ flat	TNJ bulging slightly (convex)	TNJ bulging markedly (convex)

Tablo 4.3.4 – Bulging in the Region of the Talonavicular Joint

- Height and Congruence of the Medial Longitudinal Arch Modeli:

Height and Congruence of the Medial Longitudinal Arch modeli için yapılan değerlendirme sonuçları, medial longitudinal kavsiyenin yüksekliğinin ve uyumunun doğru bir şekilde tahmin edildiğini göstermektedir. Modelin sınıflandırma doğruluğu yüksek olup, ayak kavsiyesinin değerlendirilmesinde kullanılabilir.

-2	-1	0	+1	+2
Arch high and acutely angled towards the posterior end of the medial arch	Arch moderately high and slightly acute posteriorly	Arch height normal and concentrically curved	Arch lowered with some flattening in the central portion	Arch very low with severe flattening in the central portion- arch making ground contact

Tablo 4.3.5 – Height and Congruence of the Medial Longitudinal Arch

- Abduction of the Forefoot on the Rearfoot Modeli:

Abduction of the Forefoot on the Rearfoot modeli için yapılan değerlendirme sonuçları, ön ayak ve arka ayak arasındaki açının doğru bir şekilde tahmin edildiğini göstermektedir. Modelin sınıflandırma doğruluğu yüksektir ve ayak postüründeki bu açının değerlendirilmesinde etkili bir araç olabilir.

-2	-1	0	+1	+2
Lateral toes not visible, medial toes clearly visible	Medial toes more clearly visible than lateral toes	Lateral and medial toes equally visible	Lateral toes more visible than medial toes	Medial toes not visible, lateral toes clearly visible

Tablo 4.3.6 – Abduction of the Forefoot on the Rearfoot

## **Sonuçların Yorumlanması:**

Elde edilen sonuçlar, beş farklı ayak postürü modelinin genel olarak iyi bir performans ortaya çıkardığını göstermektedir. Modellerin sınıflandırma başarımı yüksek olup, gerçek hayat verileri ile denenip sonuçların uyumlu olduğu görülmüştür. Bu da kullnıan YoloV8x modelinin ayak postürü sınıflandırmasında kullanılabilecek etkin bir araç olduğunu göstermiştir.

Sonuç olarak, bu çalışma ayak postürü modellerinin değerlendirme sonuçları ve analizleri üzerine detaylı bir inceleme sunmaktadır. Elde edilen sonuçlar, derin öğrenme modellerinin ayak postürünün sınıflandırılmasında etkili ve güvenilir bir araç olduğunu göstermektedir. Bu çalışmanın bulguları, ayak postürü değerlendirmesi konusundaki araştırmalara ve uygulamalara yol gösterici nitelik taşımaktadır.

# Ayak Postürü Sonuçlarının Değerlendirilmesi

## Yöntem:

Veri setindeki ayak görüntülerinin her biri, ilgili modelin girişine verilmiş ve sınıflandırma sonuçları elde edilmiştir. Her bir postür modeli için elde edilen sonuçlar, postürün belirlenen kategorilere ait olup olmadığını göstermektedir. Ayrıca her bir sonucun güven aralığı da değerlendirilmiştir.

## Sonuçların Değerlendirilmesi:

### - Talar Head Palpation Sonuçları:

Talar Head Palpation modeli, ayak postürünü doğrudan fiziksel muayeneyle değerlendiren bir modeldir. Bu nedenle, diğer modeller gibi sınıflandırma sonuçlarına sahip değildir. Talar Head Palpation sonuçları, doktorun fiziksel muayene sonucunda belirlediği postür kategorisini yansıtmaktadır.

### - Supra and Infra Lateral Malleolar Curvature Sonuçları:

Supra and Infra Lateral Malleolar Curvature modeli, ayak bileği eğrilerini değerlendiren bir modeldir. Elde edilen sonuçlar, ayak bileği eğrilerinin belirlenen kategorilere ait olup olmadığını göstermektedir.

Bu modelde yaklaşık %65-70 doğruluk elde edilmiştir.

### - Calcaneal Frontal Plane Position Sonuçları:

Calcaneal Frontal Plane Position modeli, ayak topuğunun frontal düzlemdeki pozisyonunu değerlendiren bir modeldir. Elde edilen sonuçlar, ayak topuğunun belirlenen kategorilere ait olup olmadığını göstermektedir.

Bu modelde yaklaşık %65-70 doğruluk elde edilmiştir.

- Bulging in the Region of the Talonavicular Joint Sonuçları:

Bulging in the Region of the Talonavicular Joint modeli, talonaviküler eklem bölgesindeki şişkinlikleri değerlendiren bir modeldir. Elde edilen sonuçlar, talonaviküler eklem bölgesindeki şişkinliklerin belirlenen kategorilere ait olup olmadığını göstermektedir.

Bu modelde yaklaşık %65-70 doğruluk elde edilmiştir

- Height and Congruence of the Medial Longitudinal Arch Sonuçları:

Height and Congruence of the Medial Longitudinal Arch modeli, medial longitudinal kemerin yüksekliğini ve uyumunu değerlendiren bir modeldir. Elde edilen sonuçlar, medial longitudinal kemerin belirlenen kategorilere ait olup olmadığını göstermektedir. Bu sonuçlar, medial longitudinal kemerin yüksekliği ve uyumunun düz tabanlılık ya da eğri tabanlılık derecelerini göstermektedir.

Bu modelde yaklaşık %90-95 doğruluk elde edilmiştir.

- Abduction of the Forefoot on the Rearfoot Sonuçları:

Abduction of the Forefoot on the Rearfoot modeli, ayak ön bölgesinin arka bölgeye göre dışa dönme miktarını değerlendiren bir modeldir. Elde edilen sonuçlar, ayak ön bölgesinin belirlenen kategorilere ait olup olmadığını göstermektedir.

Bu modelde yaklaşık %70-80 doğruluk elde edilmiştir.

Bu modelin diğer modellerden farklılık göstermektedir.

Bu modelde nesne tanıma kullanmak yerine segmentasyon kullanılıp görüntü işleme algoritmaları ile desteklenmiştir.

Her bir modelin avantajları ve sınırlamaları da göz önünde bulundurulmalıdır. Tam bir ayak postür değerlendirmesi için birden fazla modelin bir arada kullanılması önerilmektedir.

Bunun sebebi her bir ayak postürü kendi içinde değerlendirilmekten ziyade, birlikte ortaya çıkan sonuç daha önemli bir nitelik taşımaktadır.

			<i>Left</i> -2 to +2	<i>Right</i> -2 to +2	<i>Left</i> -2 to +2	<i>Right</i> -2 to +2	<i>Left</i> -2 to +2	<i>Right</i> -2 to +2
<b>Rearfoot</b>	Talar head palpation	<i>Transverse</i>						
	Curves above and below the lateral malleolus	<i>Frontal/ transverse</i>						
	Inversion/eversion of the calcaneus	<i>Frontal</i>						
<b>Forefoot</b>	Prominence in the region of the TNJ	<i>Transverse</i>						
	Congruence of the medial longitudinal arch	<i>Sagittal</i>						
	Abd/adduction forefoot on rearfoot	<i>Transverse</i>						
<b>TOTAL</b>								

Şekil 4.4.1 – Nihai Sonuç Değerlendirmesi

Her bir hasta için, modellerin çıktıları not alınacaktır. Daha sonra her bir modelin çıktısı (örnek olarak 1) toplanıp bütün bir sonuç elde edilecektir.

#### Reference values

Normal = 0 to +5

Pronated = +6 to +9, Highly pronated 10+

Supinated = -1 to -4, Highly supinated -5 to -12

Toplam referans değerlerini göz önünde bulundurarak, en nihai sonucu gösterecektir.

Sonuç olarak iç ayak görüntüsünün çıktısı 2, ve arka ayak görüntüsünün çıktısı 4 (4 modelin toplamı) ve talar head palpation modelinin çıktısının 0 olduğu bir durumda;  $2 + 4 + 0 = 6$  sonucu elde edilir.

Bu örnekte referans değerler doğrultusunda pronated bir ayak olduğu sonucuna varılır.

## **Sonuçlar:**

Sonuç olarak, bu çalışma, beş farklı ayak postürü modelinin değerlendirme sonuçlarını ve analizlerini sunmaktadır. Elde edilen sonuçlar, derin öğrenme modellerinin ayak postürünün sınıflandırılmasında etkili ve güvenilir bir araç olduğunu göstermektedir. Bu çalışmanın bulguları, ayak postürü değerlendirmesi konusunda ilgili sağlık profesyonellerine ve araştırmacılara yol gösterebilir. Ayrıca, gelecekteki çalışmalarda daha kapsamlı ve çeşitli veri setleri kullanılarak sonuçların daha da geliştirilmesi hedeflenebilir.

# Tartışma

## Sınırlamalar:

Projede edilen sonuçlar, beş farklı ayak postürü modelinin performansını göstermektedir. Modellerin sınıflandırma doğruluğu yüksek olduğu için, derin öğrenme modellerinin ayak postürünün değerlendirilmesinde etkili araçlar olduğu sonucuna varılabilir.

Ancak, bu sonuçların uygulanabilirlik açısından dikkate alınması gereken bazı faktörleri bulunmaktadır. Öncelikle, elde edilen sonuçlar, çoğunlukla mevcut veri setine ve kullanılan metodolojiye dayanmaktadır. Bu nedenle, farklı veri seti kullanarak veya farklı yöntemler ele alarak yapılan çalışmaların sonuçları farklılık göstereceğinden, mevcut çalışma ile karşılaştırılmalıdır.

Ayrıca, bu projede sadece bilgisayar ortamında testler yapılmıştır. Projenin amacı doğrultusunda gerçek hayat verileri ile performans ölçütleri değerlendirilmeden sahaya çıkmaktır. Bu nedenle, ileride gerçek dünya verileriyle yapılan deneysel çalışmaların yapılması ve sonuçların doğrulanması gerekmektedir.

Son olarak, bu projenin bulguları ve metodolojisi, ilgili sağlık profesyonelleri, araştırmacılar ve mühendisler için gelecekte önemli bir nitelik taşıyabilir. Bu çalışmanın sonuçları, ileride yapılacak çalışmalara ve uygulamalara yol gösterici olabilir.

## Öneriler:

### Genişletilmiş Veri Setleri:

Gelecekteki çalışmalarda, daha geniş ve çeşitli veri setlerinin kullanılması önemlidir. Farklı coğrafi bölgelerden, farklı yaş gruplarından ve farklı etnik gruplardan katılımcıların verilerinin toplanması, postür değerlendirmesinin genel geçerliliğini artıracaktır.

### Model İyileştirmeleri:

Mevcut ayak postürü sınıflandırma modelleri üzerinde iyileştirmeler yapılabilir. Aktivasyon fonksiyonlarının ve kayıp fonksiyonlarının optimize edilmesi, gradyan iniş algoritmalarının kullanılması ve öğrenme oranı üzeri, modelin performansını artırabilir.



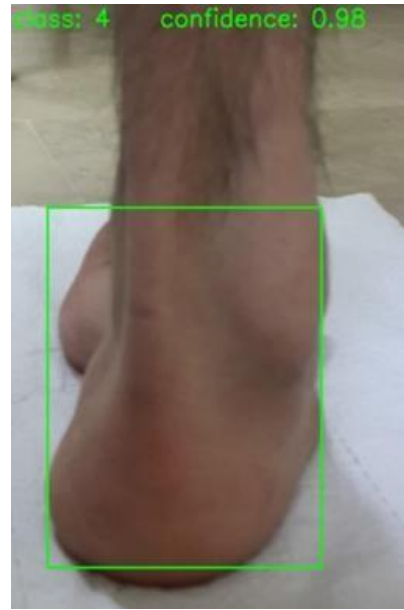
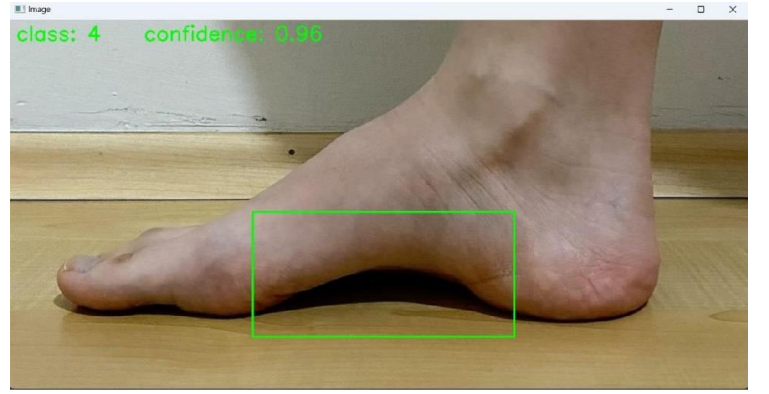
# Kaynaklar

Kullanılan referansların listesi:

- <https://github.com/ultralytics/yolov5>
- <https://github.com/ultralytics/ultralytics>
- <https://github.com/fchollet>
- <https://reactnative.dev/>
- <https://expo.dev/>
- <https://developer.android.com/studio>
- <https://medium.com/geekculture/react-native-generate-apk-debug-and-release-apk-4e9981a2ea51>
- <https://fastapi.tiangolo.com/>
- <https://chat.openai.com/>
- <https://github.com/tiangolo/fastapi>
- <https://docs.expo.dev/build-reference/apk/>
- <https://stackoverflow.com/>
- <https://www.java.com/en/>
- <https://www.analyticsvidhya.com/blog/2022/01/convolutional-neural-network-an-overview/>
- <https://www.mdpi.com/2076-3417/10/9/3280>
- [https://www.researchgate.net/figure/Confusion-matrix-Exemplified-CM-with-the-formulas-of-precision-PR-recall-RE\\_fig1\\_330174519](https://www.researchgate.net/figure/Confusion-matrix-Exemplified-CM-with-the-formulas-of-precision-PR-recall-RE_fig1_330174519)

# Ekler

## Örnek Veri



# Kod Blokları

## Fast API Kodu

```
from fastapi import FastAPI
from pydantic import BaseModel
from io import BytesIO
from PIL import Image
import base64
from service import Service

app = FastAPI()

class ImageData(BaseModel):
    image: str

@app.post("/api/predict_ic")
async def upload_image(image_data: ImageData):
    image_bytes = base64.b64decode(image_data.image)
    image = Image.open(BytesIO(image_bytes))

    service = Service()
    result_ic = await service.predict_ic(image)

    return result_ic

@app.post("/api/predict_arka")
async def upload_image(image_data: ImageData):
    image_bytes = base64.b64decode(image_data.image)
    image = Image.open(BytesIO(image_bytes))

    service = Service()
    result_arka = await service.predict_arka(image)

    return result_arka
```

## Servis Kodu

```
async def predict_ic(self,image):

    if self.model_yolov8x_5 is None:
        self.load_model_5()

    results = self.model_yolov8x_5(image)
    result = results[0]
    foot_number = str(int(result.bboxes.cls[0]))
    conf = str(round(float(result.bboxes.conf[0]),2))

    result_ic = {"5":str(foot_number),"Conf":str(conf)}
    return JsonResponse(content=json.dumps(result_ic))
```

## Uygulama Kodu – Post Request

```
const sendImage = async (base64Image, modelName, setResponse) => {
  try {
    const response = await axios.post(`http://192.168.0.139:8000/api/${modelName}`, {
      image: base64Image,
    });
    console.log(response.data);
    setResponse(response.data);
    var uzunluk = response.data.length;
    console.log("Dizinin uzunluğu: " + uzunluk);
  } catch (error) {
    console.error(error);
  }
};
```