



İSTANBUL SABAHATTIN ZAIM
ÜNİVERSİTESİ

BİM 429 VERİ MADENCİLİĞİ

Tepsi Analiz ve Tespit Sistemi

Grup No: 9

Hazırlayanlar:

030120012 - Hamza Gündoğdu
030121061 - Abdullah Melih Dal
030121028 - Murat Demir
030121029 - Eren Keskin
030721090 - Murat Elber Kaya
030119018 - Ahmet Kerem Hızlı

Teslim Tarihi: December 22, 2024

İçindekiler

1 Giriş	2
2 Literatür Özeti	2
3 Çerçevenin Yapısı, Bileşenler ve Mimari Tasarım	2
3.1 YOLOv8Large Modeli	2
3.2 PyFlask	2
3.3 React Native	3
3.4 Python	3
3.5 Sistem Mimarisinin Genel Yapısı	3
4 Yazılımın Kullanılması	3
5 Çerçevenin Algoritmasının Açıklanması	3
6 Çalışma Örnekleri	4
7 Sonuçlar ve Yorumlanması	5
8 Gelecekte Yapılabilecekler	7
9 Kullanılan Kaynaklar	7

1 Giriş

Projemiz; İstanbul Sabahattin Zaim Üniversitesi yemekhanesinden yemek yiyen kişilerin aldıkları yemek tepsilerinde hangi yiyeceklerin bulunduğu, bu yiyecek ve içeceklerin birim fiyatı ve kalorisi gibi önemli bilgilerinin listelendiği ve eğer bu menüyü bir ay boyunca yedikleri takdirde ortaya çıkacak maliyeti ve toplam kaloriyi gösteren bir rapor tablosu sunmaktadır.

2 Literatür Özeti

Projemizin en önemli kısmı olan görüntü işleme [1] için gerekli olan araştırmalar yapıldığında bu alanda en çok kullanılan modelin YOLO (You Only Look Once) [2] olduğu görülmüştür. İncelenen farklı sürümlerdeki YOLO [3] modelleri arasında en uygun modelin YoloV8Large olduğuna karar verilmiştir. Bu modeli diğerlerinden ayıran özellikleri arasında hem en güncel modellerden birisi olması hem de başarı parametrelerinin diğerlerinden daha iyi bir sonuç göstermesi yer almaktadır. Projenin arayüzü olarak geliştirilmesi planan mobil uygulamanın hem Android'i hem de iOS'u desteklemesi adına araştırmalar yapılmış ve bu araştırmalar sonucunda React Native [4] yazılım çerçevesi ile mobil uygulamanın tasarlanmasına karar verilmiştir.

3 Çerçevenin Yapısı, Bileşenler ve Mimari Tasarım

3.1 YOLOv8Large Modeli

YOLOv8Large, görüntü işleme gerçekleştirerek nesneleri algılayan bir yapay zeka modelidir. Kullanım Amacı: Kameradan alınan görüntülerde yakalanmak istenen nesnelerin algılanması.

3.2 PyFlask

Flask, Python tabanlı bir yazılım çerçevesidir. Kullanım Amacı: İnce ayar çekilmiş YOLOv8Large modeliyle mobil uygulama arayüzü arasında köprü görevi görecek API tasarlanması.

3.3 React Native

React Native, mobil uygulama geliřtirmek için kullanılan hem Android hem de iOS iřletim sistemlerini destekleyen bir yazılım çerçevesidir. Kullanım Amacı: Yapay zeka modelinin sahip olması gereken arayüzün mobil uygulama ile sağlanması.

3.4 Python

Görüntü ön iřleme, veri manipölasyonu ve özellik çıkarma gibi temel adımlar Python ile gerçekteřtirilmiřtir.

3.5 Sistem Mimarisinin Genel Yapısı

Mobil Uygulama: Geliřtirilmesinde React Native yazılım çerçevesinden yararlanılmıřtır. Kullanıcı görüntüyü yükledikten sonra PyFlask ile tasarlanmış API'ye istek gönderilir.

Backend (Flask): PyFlask API, görüntüyü alıp iřledikten sonra ince ayar çekilmiş YOLOv8Large modelimize gönderir.

Yapay Zeka Modeli: İnce ayar çekilmiş YOLOv8Large modelimiz aldığı görüntüdeki nesneleri güven deęerini de belirtip etiketler ve mobil uygulamaya bir cevap döner.

Sonuç Görüntülemesi: API'den gelen cevaplar React Native ile yazdığımız mobil uygulamada kullanıcıya gösterilir.

4 Yazılımın Kullanılması

Yazılımın kullanılabilmesi için öncelikle mobil uygulamanın çalıştırılması gerekmektedir. Uygulama açıldıktan sonra kullanıcının önüne iki seçenek gelmektedir; bunlardan birincisi, kamerayı açarak anlık resim çekilmesi, ikincisi telefonda hali hazırda bulunan bir resim dosyasının yüklenmesi. Bunlar gerçekteřtirildikten sonra uygulama çıktıları üreterek kullanıcıya sunacaktır.

5 Çerçevenin Algoritmasının Açıklanması

Algoritma ilk olarak kullanıcıdan görsel bir girdi beklemektedir. Görsel girdiyi aldıktan sonra PyFlask ile tasarlanan API vasıtasıyla ince ayar çekilmiş

YoloV8Large modelimiz çalıştırılır. Resimdeki yiyecekleri algılayan algoritmamız bunu işleyerek kullanıcıya tepside bulunan yiyeceklerin tablosunu ve beraberindeki analiz verilerini ekrana yansıtarak görevini tamamlamış olur.

6 Çalışma Örnekleri

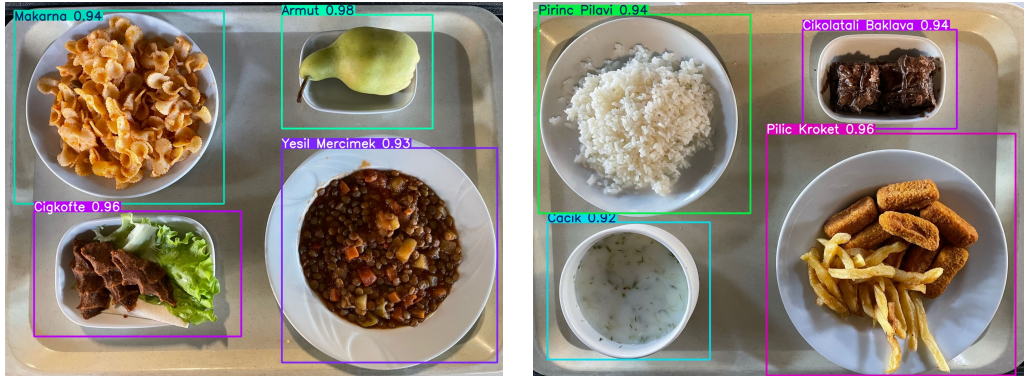


Figure 1: Başarılı Sonuçlar.

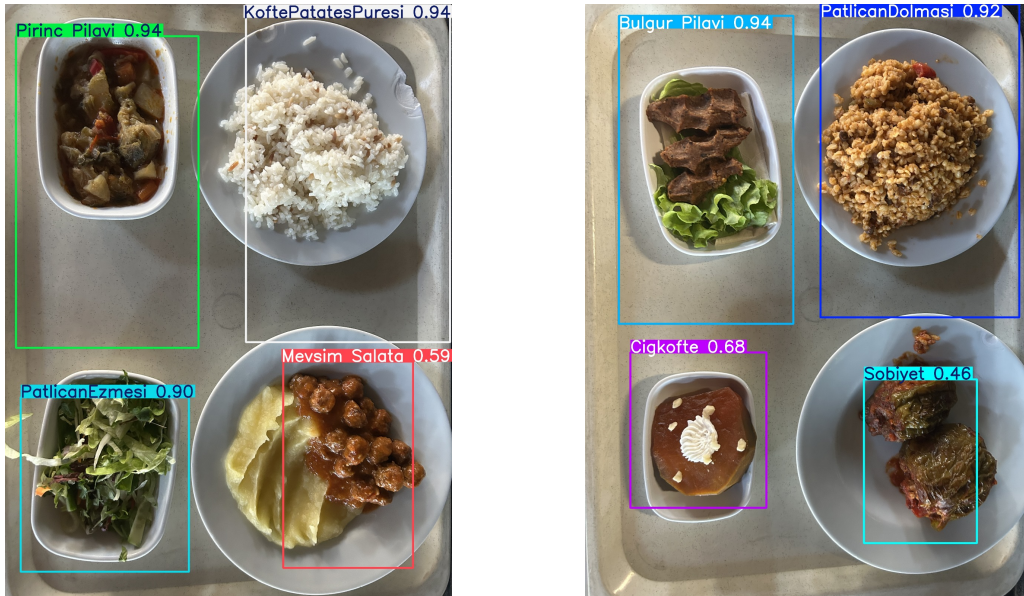


Figure 2: Başarısız Sonuçlar.

7 Sonuçlar ve Yorumlanması

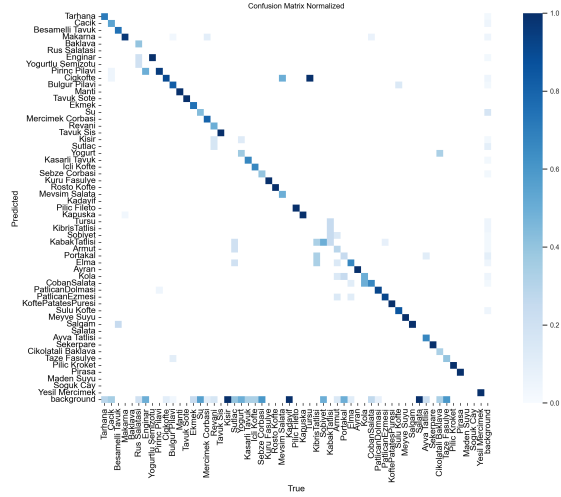


Figure 3: Karışıklık Matrisi.

Modele ait karışıklık matrisi bize hangi sınıfların birbirleriyle karıştırıldığı konusunda bilgi vermektedir. Bizim modelimize bakıldığında sınıfların büyük bir çoğunluğu doğru bir şekilde tahmin ederken, tüm verilerin arka planında tepsinin bulunmasından dolayı bazı sınıfları tepsiyle karıştırmıştır.

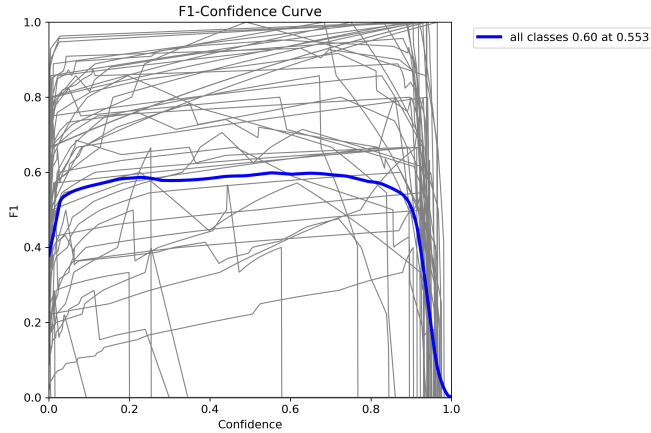


Figure 4: F1 ve Güven Grafiği.

F1 ve Güven değerlerine ait grafik incelendiğinde modelin 0.8 güven değerine kadar benzer doğruluğa sahip olduğunu 0.8'den 1'e kadar giderek doğruluğun azaldığı ve en sonunda sıfıra ulaştığı görülmektedir. Güven artarken doğruluğun gittikçe azalması sınıflar arasındaki dağılımın dengesizliğinden kaynaklanıyor olabilir, bu durumu çözmek için her sınıftan aynı sayıda veri olması gerekmektedir.

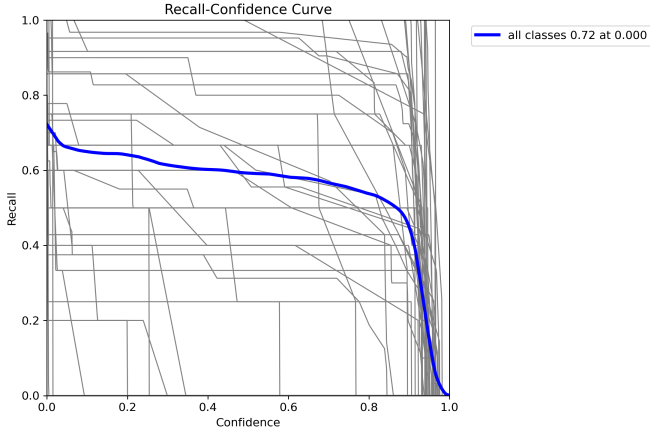


Figure 5: Duyarlılık ve Güven Grafiği.

Duyarlılık ve güven değerlerine ait grafiğe bakıldığında F1 ve güven değerine benzer bir grafik görülmektedir. Modelin kendine güveni arttıkça yaptığı tahminler yanlış olmaktadır. Doğru sınıflandırması gerekenleri yanlış algılamıştır.

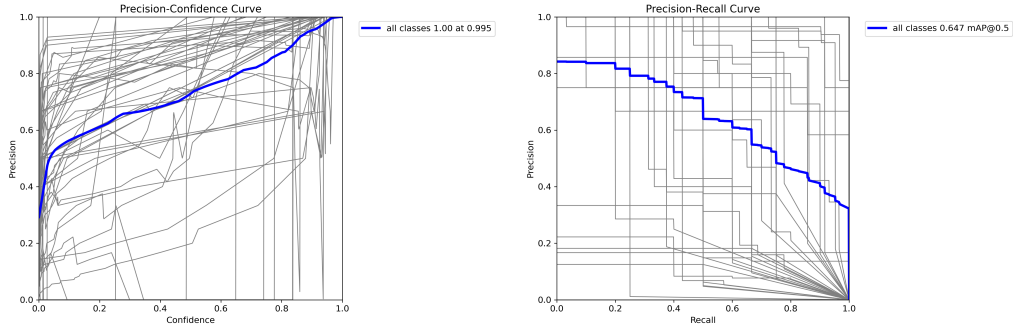


Figure 6: Kesinlik Grafikleri.

Modelin güven değeri ile kesinlik değerinin doğru orantılı olarak artması

diğer bulgularla kıyaslandığında F1 ve duyarlılık grafikleri modelin yüksek güven değerlerinde modelin kötüye gittiğini, kesinlik değerine ait grafik ise iyiye gittiğini göstermektedir.

8 Gelecekte Yapılabilecekler

Projenin ileriye yönelik yapılabilecek çalışmaları arasında diyet konusu örnek verilebilir. Kullanıcılar, her gün aldıkları yemek tepsinin fotoğrafını çekerek uygulamadan faydalanabilirler. Uyması gereken bir yemek programına sahip kişiler, bu uygulama vasıtasıyla aylık almaları gereken kaloriyi girerek programlarına uygun bir şekilde ilerleyip ilerlemediklerini görebilirler. Buna ek olarak da programlarından ne kadar uzakta olduklarını aldıkları toplam kalori ile hedeflenen kalori arasındaki farkı onlara göstererek kolayca takip etmeleri sağlanabilir. Bir diğer geliştirme alanı ise maliyet yönetimi olabilir. Aylık geliri çok yüksek olmayan kullanıcılar, örneğin öğrenciler, bu uygulama vasıtasıyla her gün hangi menüden aldığı takdirde ay sonunu getirebileceğini hesaplamalarına imkan tanınmış olur. Bu geliştirmeler kullanıcıların beslenme ve bütçe planlamalarını daha kolay yapmalarını ve planlarına olan uyumluluklarını görmelerini sağlayacaktır.

9 Kullanılan Kaynaklar

References

- [1] T. S. Huang, W. F. Schreiber, and O. J. Tretiak, “Image processing,” *Proceedings of the IEEE*, vol. 59, no. 11, pp. 1586–1609, 1971.
- [2] T. Diwan, G. Anirudh, and J. V. Tembhurne, “Object detection using yolo: Challenges, architectural successors, datasets and applications,” *multimedia Tools and Applications*, vol. 82, no. 6, pp. 9243–9275, 2023.
- [3] M. Hussain, “Yolo-v1 to yolo-v8, the rise of yolo and its complementary nature toward digital manufacturing and industrial defect detection,” *Machines*, vol. 11, no. 7, p. 677, 2023.
- [4] W. Danielsson, “React native application development,” *Linköpings universitet, Swedia*, vol. 10, no. 4, p. 10, 2016.