



Teknoloji Fakültesi

MY FRIDGE APP: MARKET FİŞLERİNDEN OCR VE YAPAY ZEKÂ İLE AKILLI TARİF VE GIDA YÖNETİM SİSTEMİ

BİTİRME PROJESİ

Bilgisayar Mühendisliği Bölümü

DANIŞMAN

Prof. Dr. Öğr. Üyesi Serhat Özkes

İSTANBUL, 2025

MARMARA ÜNİVERSİTESİ

TEKNOLOJİ FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

Marmara Üniversitesi Teknoloji Fakültesi Bilgisayar Mühendisliği

Öğrencileri Özgür Uylaş, Raoof Agh, Taylan Başar Duyğun tarafından

“MY FRIDGE APP: MARKET FİŞLERİNDEN OCR VE YAPAY ZEKÂ İLE AKILLI TARİF VE GIDA YÖNETİM

SİSTEMİ” başlıklı proje çalışması, 19.06.2025 tarihinde

savunulmuş ve jüri üyeleri tarafından başarılı bulunmuştur.

Jüri Üyeleri

Dr. Öğr. Üyesi Serhat Özekes (Danışman) (İMZA).....
Marmara Üniversitesi

Dr. Öğr. Üyesi Timur İNAN (Üye) (İMZA).....
Marmara Üniversitesi

Arş. Gör. Büşra Büyüktanır (Üye) (İMZA).....
Marmara Üniversitesi

ÖNSÖZ

Proje çalışmamız süresince karşılaştığım bütün problemlerde, sabırla yardım ve bilgilerini esirgemeyen, tüm desteğini sonuna kadar yanımda hissettiğim değerli hocalarım, sayın Doç. Dr. Buket Doğan ve sayın Prof. Dr. Serhat Özkes' e en içten teşekkürlerimizi sunarız.

Bu proje çalışması fikrinin gelişmesi ve rafine edilmesi hususunda desteğinden dolayı değerli hocam Prof. Dr. Serhat Özkes' e teşekkür ederiz.

Proje çalışmam sırasında maddi ve manevi desteklerini esirgemeyen okul içerisinde ve okul dışında her zaman yanımda olan değerli çalışma arkadaşlarım ve hocalarım Doç. Dr. Buket Doğan ve Prof. Dr. Serhat Özkes'e sonsuz teşekkürlerimizi sunarız.

Projenin fikren ortaya çıkmasında, veri setinin oluşturulup dijitalleştirilmesinde ve geliştirilmesi için incelenmesinde bize yardım eden tüm aile üyelerimize, komşularımıza, arkadaşlarımıza ve iş arkadaşlarımıza teşekkür ederiz.

İçindekiler

1. GİRİŞ.....	12
1.1 Arka Plan ve Motivasyon.....	12
1.2 Problem Tanımı.....	13
1.3 Proje Amaçları.....	13
1.4 Kapsam ve Sınırlılıklar	13
1.5 Tez Yapısı	14
2.LİTERATÜR TARAMASI	14
2.1 Mobil Uygulamalarda OCR Kullanımı	14
2.2 Doğal Dil Anlama için LLM’ler	15
2.3 API Merkezli Sistem Mimarileri.....	17
2.4 Mevcut Yemek Öneri Sistemleri ve Akıllı Mutfak Uygulamaları	18
2.4.1 KitchenPal Uygulamasının İncelenmesi	19
2.4.2. SuperCook Uygulamasının İncelenmesi	19
2.5 Özet ve Belirlenen Boşluklar	20
3.SİSTEM MİMARİSİ VE TASARIMI	20
3.1 Genel Bakış	20
3.2 Modüler Yapı ve Sorumluluklar	21
3.3 Servisler Arası İletişim	21
3.4 Teknoloji Yığını	22
4.UYGULAMA DETAYLARI	22
4.0 Veri Seti Toplama.....	23
4.1 OCR/YOLOv8	23
4.1.1 Veri Hazırlığı	23
4.1.2 YOLOv8 Eğitimi ve Değerlendirmesi.....	24
4.1.3 OCR İçin Görüntü Ön İşleme.....	26
4.1.4 Yatay Segmentasyon.....	27
4.1.5 Fiş İçeriği Bölgesinin Belirlenmesi	28
4.1.6 Dikey Segmentasyon ve Alan İzolasyonu.....	29
4.1.7 Nihai OCR Aşaması.....	29
4.2 Ürün Ayrıştırma ve Kaydetme	29
4.2.1 Metin Tanıma ve Temizleme.....	30
4.3 API Altyapısı.....	30
4.4 LLM Entegrasyonu.....	31
4.4.1 Streamlit Arayüzü	31
4.5 Android Uygulaması.....	33
4.5.1 UI/UX Design.....	33
4.5.2 API İstekleri ve İşleme.....	36
4.5.3 Tarif ve Ürünlerin Görselleştirilmesi.....	36

5.DENEYSEL KURULUM VE DEĞERLENDİRME.....	37
5.1 API Performansı (Gecikme ve Güvenilirlik)	37
5.2 LLM Çıktı Değerlendirmesi.....	38
5.2.1 Tariflerin Anlamlılığı ve Kullanışlılığı	38
5.2.2 Dil Kalitesi ve Anlaşılabilirlik	38
6.TARTIŞMA.....	38
6.1 Sistem Performansı ve Ölçeklenebilirlik	39
6.2 API Tabanlı Tasarımın Güçlü Yönleri	39
6.3 Büyük Dil Modellerinin Kullanımında Karşılaşılan Zorluklar.....	39
6.3.1 İçerik Uygunluğu ve Malzeme Tutarlılığı.....	39
6.3.2 Çıktıların Kontrol Edilebilirliği ve Tutarlılığı.....	40
6.3.3 API Kota Sınırlamaları ve Erişim Kısıtları	40
6.4 Veri Güvenliği ve Gizlilik	40
7.SONUÇ VE GELECEK ÇALIŞMALAR	41
7.1 Elde Edilen Başarılar	41
7.2 Sınırlılıklar ve Edinilen Dersler	42
7.3 Geleceğe Yönelik Geliştirme Olanakları	43
KAYNAKÇA	45
EKLER	47
Ek A – LLM Prompt Örnekleri.....	47
Örnek Prompt 1: Mod Tabanlı Tarif Önerisi	47
Örnek Prompt 2: Malzeme Tabanlı Tarif Üretimi.....	47
Örnek Prompt 3: Besin Değeri Hesaplama	47
Örnek Prompt 4: Tarif Detayları ve Hazırlık Aşamaları	48
Ek B – YOLOv8 Örnek test çıktıları.....	48

ÖZET

MY FRIDGE APP: MARKET FİŞLERİNDEN OCR VE YAPAY ZEKÂ İLE AKILLI TARİF VE GIDA YÖNETİM SİSTEMİ

Bu çalışma, market fişlerinden veri çıkararak yapay zekâ destekli bir yemek tarifi öneri ve gıda yönetim sistemi geliştirmeyi amaçlamaktadır. Günümüzde gıda israfı hem ekonomik hem de çevresel açıdan ciddi bir sorun haline gelmiştir. İnsanlar, satın aldıkları ürünlerin son kullanma tarihlerini takip edememekte ve mutfaktaki mevcut malzemeleri verimli kullanamamaktadır. Bu sorunlara çözüm üretmek amacıyla, Optik Karakter Tanıma (OCR) teknolojisi ile market fişlerinden ürün adları, fiyatlar ve son kullanma tarihleri okunmuş, bu veriler JSON formatında yapılandırılarak Spring Framework aracılığıyla geliştirilen Uygulama Programlama Arabirimini (API) kullanarak veri tabanına kaydedilmiştir.

Elde edilen veriler, farklı bir API aracılığıyla Büyük Dil Modellerine (LLM) iletilmiş ve bu modelde analiz edilerek kullanıcılara satın aldıkları ürünlere uygun yemek tarifleri önerilmiştir. Ayrıca son kullanma tarihi yaklaşan ürünler için bildirimler oluşturulmuştur. Son dönemlerde öne çıkan 'Topping Economy' adı altında ürünlere kişiselleştirme öğelerinin eklenmesi trendini izleyerek kullanıcıya besin değeri analizi ile kullanıcının vegan, vejetaryen veya normal beslenme tercihlerine göre özel öneriler sunulmuştur.

Sistem, OpenAI, Hugging Face ve Google Gemini API'leri ile entegre edilerek farklı dil modelleri test edilmiş ve en iyi performans gösteren model belirlenmiştir. Bu işlemlerin tamamının entegre çalışması için arka planda farklı API'lar geliştirilmiş olup LLM ve OCR modellerine bağlanmıştır.

Veri tabanı Spring Framework'ün sunduğu H2 veri tabanında oluşturulmuş olup ileri düzenlemelere ihtiyaç duymaktadır. Prototip seviyesinde tüm görevleri yerine getirdiği doğrulanmıştır. Arayüz denemeleri için Web altyapısı oluşturulmuş Postman komutları hazırlanmıştır.

Bu çalışma, mutfakta verimliliği artırmayı, sağlıklı beslenmeyi teşvik etmeyi, gıda israfını azaltmayı amaçlayan yenilikçi ve kişiselleştirilebilen bir sistem ortaya koymaktadır. Gelecekte, sistemin mobil platformlara entegre edilmesi ve öneri algoritmalarının daha kişiselleştirilmiş hale getirilmesi planlanmaktadır.

Anahtar Kelimeler: Optik Karakter Tanıma (OCR), Büyük Dil Modelleri (LLM), Spring Framework, Market Fişi Analizi, Yapay Zekâ Destekli Tarif Önerisi, Gıda İsrafını Önleme, API Entegrasyonu.

Haziran, 2025

Taylan Başar Duyğun- 170420040

Raof Agh- 170421992

Özgür Uylaş- 171421008

ABSTRACT

MY FRIDGE APP: AN INTELLIGENT RECIPE AND FOOD MANAGEMENT SYSTEM USING OCR AND ARTIFICIAL INTELLIGENCE BASED ON SUPERMARKET RECEIPTS

This study aims to develop an AI-supported recipe recommendation and food management system by extracting data from grocery receipts. Today, food waste has become a serious issue both economically and environmentally. People are often unable to track the expiration dates of the products they purchase and fail to efficiently utilize the ingredients available in their kitchens. To address these problems, product names, prices, and expiration dates were extracted from grocery receipts using Optical Character Recognition (OCR) technology. These data were structured in JSON format and stored in a database through an Application Programming Interface (API) developed with the Spring Framework.

The collected data were transmitted to Large Language Models (LLMs) via a separate API, where they were analyzed to suggest recipes tailored to the user's purchased items. Additionally, notifications were generated for products nearing their expiration dates. In line with the recent trend known as the "Topping Economy", where personalization elements are added to products, the system provided nutritional analysis and personalized suggestions based on the user's dietary preferences—vegan, vegetarian, or standard. The system was integrated with APIs from OpenAI, Hugging Face, and Google Gemini, allowing for the testing of different language models to determine the best-performing one. Various APIs were developed in the backend to ensure the seamless integration of the LLM and OCR models.

The database was built using the H2 database offered by the Spring Framework, though it requires further enhancement. It has been verified that the prototype successfully fulfills all intended functions. A web-based infrastructure was established for interface testing, and Postman commands were prepared accordingly. This study presents an innovative and customizable system aimed at enhancing kitchen efficiency, promoting healthy eating, and reducing food waste. In the future, it is planned to integrate the system into mobile platforms and further personalize the recommendation algorithms.

Keywords: Optical Character Recognition (OCR), Large Language Models (LLM), Spring Framework, Grocery Receipt Analysis, AI-Assisted Recipe Recommendation, Food Waste Prevention, API Integration.

June, 2025

Taylan Başar Duyğun - 170420040

Raof Agh – 170421992

Özgür Uylaş - 171421008

KISALTMALAR

OCR : Optical Character Recognition (Optik Karakter Tanıma)

LLM : Large Language Model (Büyük Dil Modeli)

API : Application Programming Interface (Uygulama Programlama Arayüzü)

AI : Artificial Intelligence (Yapay Zeka)

JSON : JavaScript Object Notation (Veri Depolama Formatı)

SQL : Structured Query Language (Yapısal Sorgulama Dili)

DB : Database (Veritabanı)

ML : Machine Learning (Makine Öğrenmesi)

DL : Deep Learning (Derin Öğrenme)

NLP : Natural Language Processing (Doğal Dil İşleme)

UI : User Interface (Kullanıcı Arayüzü)

UX : User Experience (Kullanıcı Deneyimi)

PY : Python (Programlama Dili)

JS : JavaScript (Programlama Dili)

FE : Frontend (Ön Yüz Geliştirme)

BE : Backend (Arka Yüz Geliştirme)

HF : Hugging Face (Açık Kaynak NLP Platformu)

GPT : Generative Pre-trained Transformer (Önceden Eğitilmiş Üretici Dönüştürücü)

CV : Computer Vision (Bilgisayarla Görü)

YOLO : You Only Look Once (Gerçek Zamanlı Nesne Tanıma Algoritması)

FTR : Feature Extraction (Özellik Çıkarımı)

REC : Recommendation System (Öneri Sistemi)

TL : Transfer Learning (Transfer Öğrenmesi)

CLI : Command Line Interface (Komut Satırı Arayüzü)

ŞEKİL LİSTESİ

Şekil 1. Transformer Modelinin Ana Bileşenleri.....	15
Şekil 2. Self Check GPT.....	16
Şekil 3. API'ler ve modellerin iletişimini gösteren şema.....	20
Şekil 4. OCR modelinin Performance Metric'lerini gösteren şekil.....	23
Şekil 5. IoU Distribution (dağılımı) grafiğini gösteren şekil	24
Şekil 6 Etiket ve Tahmin sonuçlarını gösteren şekiller	24
Şekil 7 OCR fiş görüntüsü.....	25
Şekil 8. Ön işleme sonrası fiş görüntüsü	26
Şekil 9. Fiziki fiş görüntüsü.....	26
Şekil 10. Segmente edilmiş fiş görüntüsü	27
Şekil 11. Fiş bölgelerinin başlangıcının belirlendiği şekil	27
Şekil 12. Fiş bölgelerinin sonunun belirlendiği	28
Şekil 13. Ürün adının alındığı segment şekli.....	28
Şekil 14. KDV oranının alındığı segment şekli	28
Şekil 15. Fiyat bilgisinin alındığı segment şekli.....	28
Şekil 16 Besin değeri şekli	30
Şekil 17. Tarif Önerisi Şekli	31
Şekil 18 Günlük Kalori İhtiyacı.....	31
Şekil 19 Alışveriş Listesi şekli	32
Şekil 20 Favorilerim şekli	32
Şekil 22 Kullanıcı Giriş Ekranını gösteren şekil	33
Şekil 21 Kullanıcı kayıt ekranını gösteren şekil	33
Şekil 23. Kullanıcı anasayfasını gösteren şekil	34
Şekil 24. Kullanıcının özelleştirilmiş tariflerin kaydırılabilir listesinin şekli	35
Şekil 25. Adım adım yemek tarifi almayı sağlayan etkileşimli sohbet arayüzü şekli	35
Şekil 26. YOLOv8 gruplandırılmış test çıktılarının şekli	48
Şekil 27. YoloV8 çeşitlendirilmiş fiş ve diğer nesnelerin çıktılarının şekli.....	49

1. GİRİŞ

1.1 Arka Plan ve Motivasyon

Günümüzde, hızlı yaşam tarzının ve plansız tüketim alışkanlıklarının yaygınlaşması, bireylerin satın aldıkları gıda ürünlerini etkin bir şekilde takip edememelerine ve mutfak yönetimini verimli bir biçimde gerçekleştirememelerine neden olmaktadır. Bu durum, küresel ölçekte ciddi boyutlara ulaşan gıda israfının başlıca nedenlerinden biri olarak değerlendirilmektedir. Birleşmiş Milletler Gıda ve Tarım Örgütü (FAO) tarafından yayımlanan verilere göre, dünya genelinde her yıl milyarlarca ton gıda israf edilmekte; bu israf yalnızca ekonomik açıdan büyük kayıplara yol açmakla kalmamakta, aynı zamanda çevresel sürdürülebilirlik üzerinde de olumsuz etkiler yaratmaktadır.

Özellikle son kullanma tarihi geçmiş ürünlerin fark edilmeden tüketilmesi veya doğrudan çöpe atılması, hem insan sağlığı açısından risk teşkil etmekte hem de ciddi bir kaynak israfına neden olmaktadır. Gıda tedarik zincirinin son halkası olan bireysel tüketici düzeyinde ortaya çıkan bu sorun, çoğu zaman bilinç eksikliği, zaman yönetimi zorlukları ve teknolojik destekten yoksun planlama süreçleriyle ilişkilendirilmektedir.

Bu bağlamda, teknolojik çözümler aracılığıyla gıda israfının önlenmesi, günümüzde her zamankinden daha fazla önem arz etmektedir. Özellikle yapay zekâ (AI) tabanlı teknolojiler; kullanıcı alışkanlıklarının analiz edilmesi, veri temelli öneri sistemlerinin geliştirilmesi ve kişiye özel içeriklerin sunulması gibi çeşitli alanlarda etkili biçimde kullanılmaktadır. Bu teknolojilerin mutfak yönetimi süreçlerine entegre edilmesi, yalnızca bireysel verimliliği artırmakla kalmayıp, aynı zamanda toplumsal düzeyde israfın azaltılmasına yönelik sürdürülebilir bir katkı sağlamaktadır.

1.2 Problem Tanımı

Bireyler, günlük yaşamlarında sıklıkla alışveriş yapmalarına rağmen, bu ürünlerin takibini sistemli bir şekilde gerçekleştirememektedir. Özellikle ürünlerin son kullanma tarihleri gözden kaçırılmakta, bu da hem sağlık riski hem de israf anlamına gelmektedir. Ayrıca, evde bulunan malzemelere uygun yemek tariflerine ulaşmak çoğu zaman zahmetli bir süreçtir. Dolayısıyla, gıda israfının önlenmesi adına kullanıcıya özel, güncel ürün verilerine dayalı, akıllı bir yönlendirme sistemi ihtiyacı ortaya çıkmaktadır.

1.3 Proje Amaçları

Bu projenin temel amacı, market fişlerinden elde edilen verileri kullanarak, kullanıcıların mutfaktaki mevcut malzemelerini daha verimli ve sürdürülebilir bir şekilde değerlendirmelerini sağlayacak yapay zekâ destekli bir yemek tarifi öneri ve gıda yönetim sistemi geliştirmektir. Sistem; özellikle son kullanma tarihi yaklaşan ürünlerin tespit edilmesi, bu ürünlerin israf edilmeden değerlendirilmesi ve bireylerin beslenme alışkanlıklarına uygun tariflerin sunulması gibi temel işlevleri yerine getirmek üzere tasarlanmıştır.

Geliştirilen sistem sayesinde, kullanıcıların alışveriş sonrasında satın aldıkları ürünleri daha bilinçli bir şekilde takip etmeleri, plansız tüketim alışkanlıklarının önüne geçmeleri ve sağlıklı

beslenme davranışları geliřtirmeleri hedeflenmektedir. Bu bağlamda sistem, yalnızca bireysel fayda sağlamakla kalmamakta; aynı zamanda günlük yaşam pratiklerine entegre edilebilen yenilikçi bir teknolojik çözüm niteliği taşımaktadır.

Proje, bireysel mutfak yönetimiyle sınırlı kalmayıp, aynı zamanda küresel ölçekte giderek büyüyen gıda israfı sorununa karşı da uygulanabilir, ölçeklenebilir ve sürdürülebilir bir çözüm önerisi sunmayı amaçlamaktadır. Bu doğrultuda sistemin hem bireylerin hem de toplumun genel refahına katkı sağlayacak şekilde teknolojiyi işlevsel bir araç olarak kullanması öngörülmektedir.

1.4 Kapsam ve Sınırlılıklar

Bu çalışma kapsamında geliştirilen sistem, market fişlerinden ürün bilgilerini ve son kullanma tarihlerini otomatik olarak algılayabilen bir Optik Karakter Tanıma (OCR) modülüne sahiptir. OCR modülü aracılığıyla elde edilen veriler, yapılandırılmış bir JSON formatında işlenmekte ve daha sonra sistemin arka planında yer alan veritabanına aktarılmaktadır. Bu yapılandırılmış veriler, Büyük Dil Modelleri (Large Language Models – LLM) aracılığıyla analiz edilerek kullanıcıya kişiselleştirilmiş tarif önerileri sunulmasını mümkün kılmaktadır.

Sistem; kullanıcıların ellerindeki malzemelere uygun yemek tarifleri önerme, son kullanma tarihi yaklaşan ürünler hakkında bilgilendirme ve beslenme tercihlerine göre tariflerin özelleştirilmesi gibi temel işlevleri yerine getirmektedir. Böylece hem kullanıcının günlük yaşamında zaman ve kaynak yönetimini kolaylaştırmakta hem de gıda israfını azaltmaya katkı sağlamaktadır.

Sistemin geliştirilmesinde, farklı yapay zekâ servis sağlayıcılarına ait API'ler entegre edilmiştir. Bu doğrultuda OpenAI, Hugging Face ve Google Gemini gibi platformların LLM çözümleri test edilmiş; uygulamanın gereksinimlerine en uygun çıktıları sunan model tercih edilmiştir. Uygulamanın tüm yapısı Python tabanlı bir yazılım altyapısı üzerinde geliştirilmiş olup, H2 veritabanı ile desteklenmiştir. Ön yüz tarafında, Android ve streamlit, kullanıcı dostu bir arayüz aracılığıyla sistemle etkileşim sağlanmaktadır.

Bu çalışmanın kapsamı, kullanıcının fişi sisteme tanıtmasından başlayarak, ürün bilgilerinin işlenmesi, LLM destekli tarif önerilerinin sunulması ve gıda yönetimi fonksiyonlarının yürütülmesi süreçlerini içermektedir. Bununla birlikte sistemin genel performansı, başta OCR çıktılarının doğruluğu, veritabanı bağlantılarının sürekliliği ve LLM API servislerinin kullanım sınırlamaları olmak üzere çeşitli teknik faktörlere bağlı olarak değişkenlik gösterebilmektedir.

1.5 Tez Yapısı

Bu tez toplam yedi bölümden oluşmaktadır. Birinci bölümde, çalışmanın arka planı, problem tanımı, amacı, kapsamı ve yapısı açıklanmıştır. İkinci bölümde, gıda israfı, yapay zekâ uygulamaları, OCR teknolojisi ve büyük dil modelleri ile ilgili literatür taraması sunulmuştur. Üçüncü bölümde, sistemin mimarisi, kullanılan teknolojiler açıklanmıştır. Dördüncü bölümde, uygulamanın detayları, veri işleme süreçleri, modül entegrasyonları ve fiziksel yapısı hakkında fikir edinirken beşinci bölümde test çıktıları performans metrikleri güvenlik ve gizlilik önemlerine değinilmiş altıncı bölümde, genel değerlendirme yapılmış; elde edilen bulgular

ışığında sonuçlar, sınırlılıklar ve gelecek çalışmalar için öneriler sunulmuştur.yedinci bölümde ise elde edilen başarılar,çıkarılan dersler ve gelecekte yapılması planan alanlar yer almaktadır.

2.LİTERATÜR TARAMASI

2.1 Mobil Uygulamalarda OCR Kullanımı

OCR teknolojisi, özellikle market fişleri, ürün etiketleri ve ambalajlar üzerindeki metinlerin dijital ortama aktarılmasında yaygın olarak kullanılmaktadır. Pettersson ve arkadaşları [1], ürün ambalajları üzerindeki metinleri OCR ile çıkararak, görsel tanıma sistemleriyle entegre çalışan çok modlu bir sınıflandırma modeli geliştirmiştir. Bu yaklaşım sayesinde, özellikle görsel olarak benzer ürünlerin daha doğru sınıflandırılabildiği ortaya konmuştur.

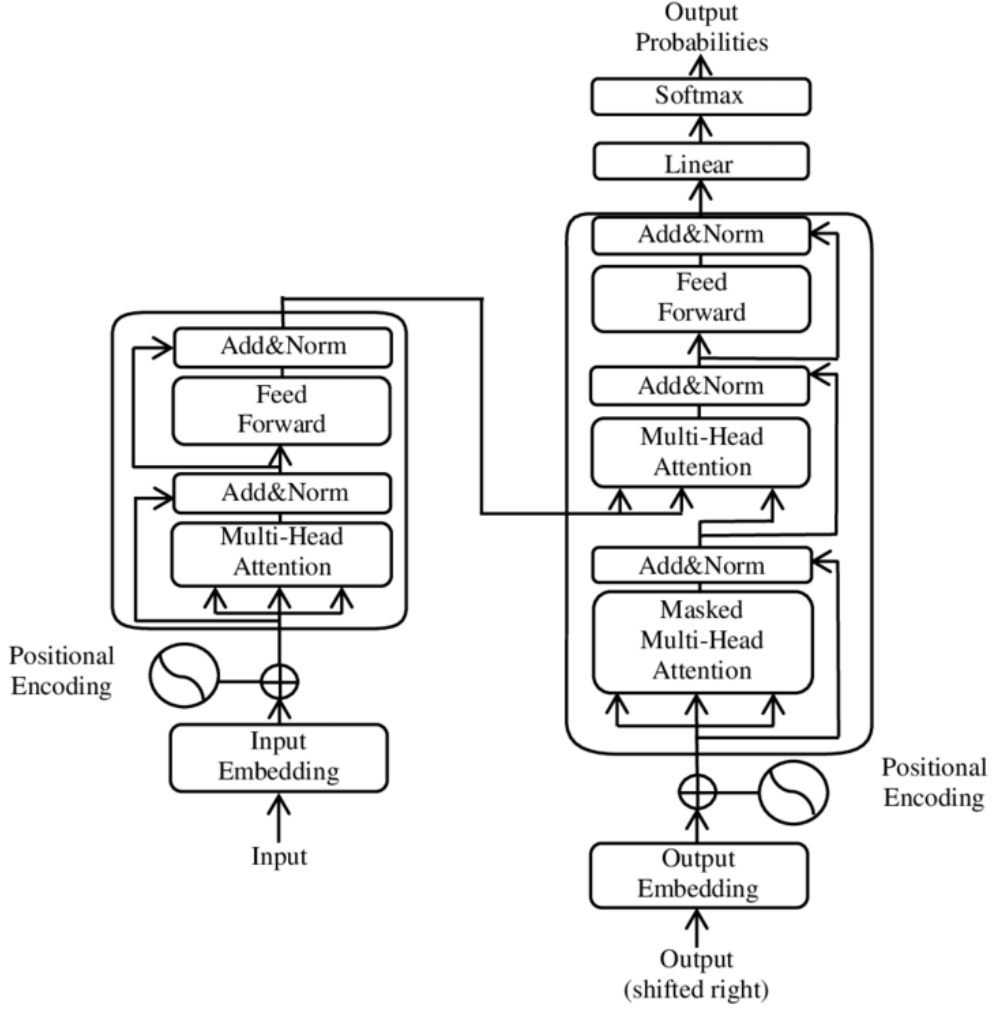
Benzer şekilde Gong ve çalışma arkadaşları [2], gıda ambalajları üzerindeki son kullanma tarihlerini tanımlamak üzere çift yapılı bir derin öğrenme modeli tasarlamış ve düşük ışık koşullarında dahi %95'in üzerinde doğruluk elde etmiştir. Kim ve ark. [3] ise bebek maması ürünleri üzerinde yaptıkları çalışmada, OCR teknolojisiyle ürün etiketlerinden besin içeriklerini ve sağlıkla ilgili iddiaları başarıyla çıkarmış; bu verileri ağ analizine tabi tutarak sektördeki yaygınlıklarını ortaya koymuştur.

OCR'nin bir diğer uygulama alanı, market fişlerinden gıda tüketimi verisi elde etmeye yöneliktir. Kaderabek [4], süpermarket fişlerinden OCR teknolojisi aracılığıyla ürün adı, fiyat ve miktar bilgilerini otomatik olarak çıkaran ve bu bilgileri veritabanına aktaran bir sistem geliştirmiştir. Bu tür otomasyon uygulamaları, manuel veri girişine olan ihtiyacı azaltmakta ve daha büyük örneklemli tüketim analizlerine olanak sağlamaktadır.

2.2 Doğal Dil Anlama için LLM'ler

Büyük Dil Modelleri (Large Language Models - LLM), son yıllarda doğal dilin işlenmesi ve anlaşılması alanında çığır açan gelişmelere öncülük etmiştir. Bu modeller, Transformer mimarisi temel alınarak tasarlanmış ve çok büyük ölçekli veri kümeleri üzerinde önceden eğitilmiştir. Böylece bağlamı kavrayarak anlamlı, tutarlı ve görev odaklı metinler üretebilme yeteneğine ulaşmışlardır [1].

GPT, BERT, PaLM ve LLaMA gibi modeller, hem akademik hem de endüstriyel alanlarda geniş bir kullanım alanı bulmuş ve doğal dil işleme uygulamalarında önemli rol oynamaya başlamıştır [2], [3]. OpenAI tarafından geliştirilen GPT-3 modeli, 175 milyar parametre ile eğitilmiş olup yalnızca birkaç örnekle yeni görevleri öğrenme yeteneğiyle dikkat çekmiştir [4]. Bunu izleyen GPT-4 ise daha güçlü bağlam takibi, çok modlu (multimodal) destek ve görev bazlı ince ayar imkânları ile öne çıkmaktadır [5].



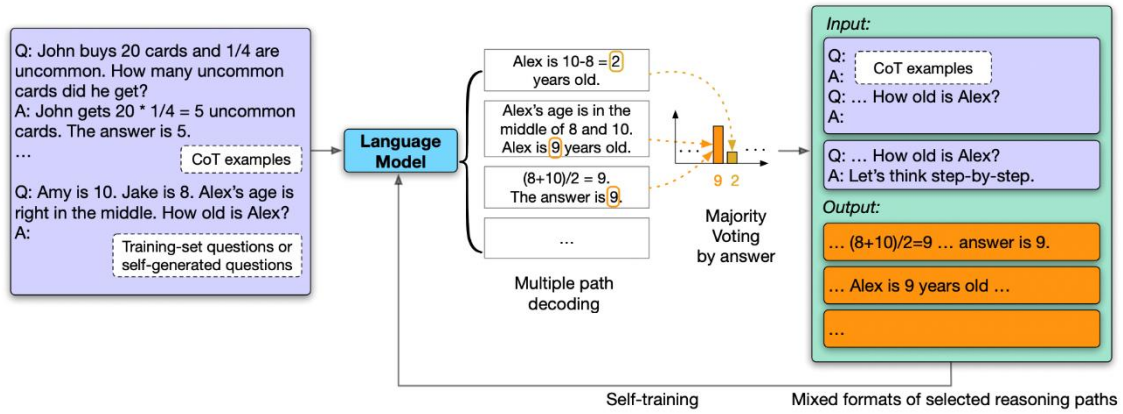
Şekil 1. Transformer Modelinin Ana Bileşenleri

Ouyang ve arkadaşlarının yürüttüğü çalışmada [6], insan geri bildirimiyle pekiştirmeli öğrenme (Reinforcement Learning with Human Feedback - RLHF) yöntemiyle geliştirilen InstructGPT modelinin, kullanıcı komutlarını daha iyi anladığı ve daha tatmin edici yanıtlar sunduğu gösterilmiştir.

LLM'ler yalnızca metin üretimiyle sınırlı kalmamakta; özetleme, soru-cevap, çeviri gibi birçok doğal dil görevinde de yüksek performans sergilemektedir. Brown ve arkadaşlarının çalışması [4], GPT-3'ün bu tür görevlerdeki etkileyici başarısını ortaya koymuştur. Öte yandan, Wei ve ekibi [7], zincirleme düşünme (Chain-of-Thought Prompting) yönteminin, modellerin mantıksal akıl yürütme becerilerini anlamlı biçimde geliştirdiğini göstermiştir.

Uygulama alanları açısından değerlendirildiğinde, sağlık sektörü LLM'lerin etkin biçimde kullanıldığı önemli bir alandır. Singhal ve arkadaşları [8], Med-PaLM 2 modelinin tıbbi bilgileri analiz ederek doğru ve güvenilir öneriler sunabildiğini ve bazı uzmanlık alanlarında hekimlerle karşılaştırılabilir başarı düzeyine ulaştığını belirtmektedir. Eğitimde ise GPT-4 tabanlı sistemlerin, örneğin Khan Academy gibi platformlarda öğrencilere kişiselleştirilmiş destek sunduğu ifade edilmektedir [5].

Ancak LLM'lerin bazı sınırlılıkları da bulunmaktadır. Ji ve arkadaşlarının yürüttüğü kapsamlı bir incelemede [9], bu modellerin zaman zaman yanlış ya da tamamen uydurulmuş içerikler (halüsinasyonlar) üretebildiği tespit edilmiştir. Bu sorunun önüne geçmek için geliştirilen SelfCheckGPT aracı, Manakul ve ekibi tarafından tanıtılmıştır [10].



Şekil 2. Self Check GPT

Öte yandan, Carlini ve arkadaşlarının araştırması [11], modellerin eğitim verilerinden istem dışı olarak kişisel bilgileri tekrar edebildiğini göstermekte; bu durum veri mahremiyeti açısından ciddi bir tehdit oluşturmaktadır.

Etik boyutta ise Bender ve arkadaşları [12], LLM'lerin eğitim verilerinde yer alan cinsiyet, ırk gibi sosyal önyargıları yeniden üretebildiğini ve bu durumun ayrımcılığa neden olabileceğini ortaya koymuştur. Bu tür riskleri azaltmak amacıyla geliştirilen Constitutional AI gibi yaklaşımlar, daha güvenli ve denetlenebilir model çıktıları elde etmeyi hedeflemektedir [13].

Ayrıca, bu modellerin karar alma süreçlerinin çoğu zaman şeffaf olmaması, kullanıcıya sonuçların gerekçelerini açıklamayı zorlaştırmaktadır. Bu durum, özellikle yüksek riskli uygulama alanlarında önemli bir sorun olarak değerlendirilmiştir. Doshi-Velez ve Kim'in çalışmaları [14], bu bağlamda açıklanabilir yapay zekâya duyulan ihtiyacı vurgulamaktadır.

Dilsel çeşitlilik bakımından da sınırlılıklar söz konusudur. Zhao ve arkadaşları [15], LLM'lerin İngilizce dışındaki dillerde zaman zaman tutarsız ve hatalı çıktılar üretebildiğini ifade etmektedir.

2.3 API Merkezli Sistem Mimarileri

Yazılım sistemlerinin mimarisinde son yıllarda gözle görülür bir değişim yaşanmıştır. Geçmişte yaygın olan monolitik, yani bütünleşik yapılar, yerini bağımsız çalışabilen servislerin birleşiminden oluşan mikroservis tabanlı yapılara bırakmaktadır. Bu evrimin ardındaki temel motivasyon, yazılım sistemlerinin daha kolay geliştirilebilmesi, güncellenebilmesi ve sürdürülebilir şekilde yönetilebilmesini sağlamaktır. Bu modern dönüşümde API merkezli mimariler, sistemlerin temel taşı olarak öne çıkmaktadır. Farklı yazılım bileşenlerinin

birbirinden bağımsız bir şekilde haberleşmesini mümkün kılan API tabanlı yaklaşımlar, hem geliştirici verimliliğini artırmakta hem de sistem genelinde esneklik kazandırmaktadır.

Literatürde yapılan incelemelerde, API tabanlı mimarilerin özellikle bulut ortamında çalışan, çok katmanlı ve genişleyebilir sistemler için tercih edildiği görülmektedir. Newman (2015), mikroservis mimarisinin her bileşeninin kendi içinde çalışabilir, bağımsız şekilde dağıtılabilir ve gerektiğinde yeniden başlatılabilir olması gerektiğini vurgulamıştır. Richardson (2018) ise RESTful API'lerin taşınabilirlik, modülerlik ve geliştirici deneyimi açısından sunduğu avantajları ayrıntılı bir şekilde ortaya koymuştur. Bu tür sistemlerde API'ler genellikle HTTP protokolü üzerinden, JSON gibi hafif veri yapıları ile haberleşmekte, böylece istemci ve sunucu tarafında sade, hızlı ve entegre bir yapı sağlanmaktadır.

Bu çalışmada geliştirilen proje de API merkezli bu yaklaşımları temel almıştır. Sistemde yer alan Main API, OCR API, LLM API, DB API ve User API bileşenleri birbirlerinden tamamen bağımsız çalışmakta ve RESTful ilkelerine uygun şekilde yapılandırılmıştır. Her API, yalnızca kendi görevinden sorumlu olacak biçimde geliştirilmiş; bu sayede sistemin geneli etkilenmeden sadece ilgili modülde değişiklik yapılabilmesi mümkün hale getirilmiştir. Ayrıca her bir API'nin farklı portlar üzerinden çalışacak şekilde ayrıştırılması, güvenlik ve yapılandırma açısından büyük kolaylık sağlamıştır. Kullanıcıya özel veri yönetimi için kimlik doğrulama mekanizmaları entegre edilmiş, böylece hem güvenlik hem de kişiselleştirilmiş veri erişimi sağlanmıştır. Sonuç olarak bu mimari, yazılım geliştirme sürecini kolaylaştırmış, sürdürülebilir ve esnek bir yapı kazandırmış ve gelecekte yapılacak geliştirmelere zemin hazırlamıştır.

2.4 Mevcut Yemek Öneri Sistemleri ve Akıllı Mutfak Uygulamaları

Son yıllarda, evde mevcut malzemelere göre yemek tarifi önerisinde bulunan mobil uygulamalar, kullanıcılar arasında büyük ilgi görmeye başlamıştır. Bu tür uygulamalar, yalnızca bireylerin “bugün ne pişirsem?” sorusuna pratik bir yanıt sunmakla kalmamakta; aynı zamanda gıda israfını azaltmak, yemek planlamasını kolaylaştırmak ve daha sağlıklı beslenme alışkanlıklarını teşvik etmek gibi toplumsal faydalar sağlamayı da hedeflemektedir.

Bu alanda dikkat çeken iki önemli örnek uygulama KitchenPal ve SuperCook'tur. Her iki sistem de kullanıcıların evde bulunan malzemeleri girmesiyle çalışan bir yapay zekâ algoritmasına dayanmaktadır. Ancak çözüm yaklaşımları farklılık gösterebilmektedir: KitchenPal, aynı zamanda kullanıcıya market alışveriş listesi oluşturma, ürünleri kategorilere ayırma ve diyet tercihlerine göre filtreleme gibi ek işlevler sunarken; SuperCook, özellikle tarif esnekliği ve gerçek zamanlı içerik uyarlaması ile öne çıkmaktadır.

Bu uygulamalar, akıllı telefonları yalnızca birer iletişim aracı olmaktan çıkararak, kullanıcıların günlük yaşamlarında aktif olarak kullandıkları dijital mutfak asistanlarına dönüştürmeyi başarmıştır. Ayrıca bu tür teknolojiler, kullanıcıyı daha planlı ve israfı önleyici bir mutfak pratiğine teşvik ederek sürdürülebilir tüketim alışkanlıklarının benimsenmesine de katkı sağlamaktadır.

2.4.1 KitchenPal Uygulamasının İncelenmesi

Gıda öneri sistemleri arasında öne çıkan uygulamalardan biri olan KitchenPal, kendisini “akıllı bir kiler yöneticisi” olarak konumlandırmakta ve kullanıcıya kapsamlı bir mutfak yönetim deneyimi sunmaktadır. Uygulama, yemek tarifi önerme, alışveriş listesi oluşturma ve mevcut stok takibi gibi birden fazla işlevi tek bir platformda birleştirmektedir. Kullanıcılar, mutfaklarında bulunan ürünleri barkod tarama yoluyla ya da manuel giriş yaparak sisteme ekleyebilmekte; bu bilgiler doğrultusunda uygulama, mevcut malzemelerle hazırlanabilecek yemek tariflerini önermektedir.

KitchenPal’ın en dikkat çekici yönlerinden biri, yalnızca bir tarif öneri sistemi olmaktan öte, kişiselleştirilmiş ve düzenli bir mutfak yönetimi sunmasıdır. Uygulama, kullanıcıların diyet tercihlerini (örneğin vejetaryenlik, glütensiz beslenme) dikkate almakta ve ürünlerin son kullanma tarihlerini izleyerek olası gıda israfını önlemeye çalışmaktadır. Böylece, kullanıcılar herhangi bir manuel takibe gerek kalmaksızın mutfaklarını verimli şekilde yönetebilmekte, aynı zamanda sağlıklı ve israfı önleyen bir mutfak alışkanlığı geliştirebilmektedir.

Daha geniş bir teknolojik bağlamda ele alındığında, KitchenPal, akıllı mutfak kavramına geçişin işlevsel bir örneği olarak değerlendirilebilir. Her ne kadar IoT (Nesnelerin İnterneti) teknolojilerini doğrudan kullanmasa da, kullanıcı dostu arayüzü ve basit otomasyon altyapısı sayesinde günlük mutfak işlerine belirli bir düzeyde dijital zekâ kazandırmaktadır. Özellikle kiler envanteri ile yemek planlamasını entegre bir şekilde sunması, uygulamayı sıradan bir yemek tarifi uygulamasının ötesine taşımakta; kullanıcıya kapsamlı bir dijital mutfak asistanı deneyimi sunmaktadır.

2.4.2. SuperCook Uygulamasının İncelenmesi

Gıda öneri uygulamaları arasında işlevsel sadeleşmeye odaklanan bir diğer örnek olan SuperCook, uzun vadeli kiler yönetimi yerine, kullanıcıların mevcut malzemelerine göre hızlı ve pratik tarif önerileri sunmayı amaçlayan bir yapıya sahiptir. Kullanıcılar, evde bulunan malzemeleri manuel olarak yazarak, sesli komut kullanarak ya da belirli kategorilerden seçerek uygulamaya girebilmekte; sistem ise bu bilgiler doğrultusunda, ek malzeme gerektirmeyen veya yalnızca birkaç ek bileşenle hazırlanabilecek tarifleri listelemektedir.

SuperCook’un dikkat çeken en güçlü yönü, basitliği ve kullanıcı dostu yapısıdır. Uygulama, herhangi bir kullanıcı kaydı gerektirmeden çalışmakta; internet üzerindeki çok sayıda tarif kaynağını tarayarak zengin içerikli bir veri tabanı sunmaktadır. Bu sayede kullanıcı, uzun süreli tarif aramalarında yaşanan “karar yorgunluğunu” yaşamadan, yalnızca elindeki malzemelerle yapılabilecek yemekleri hızlıca görebilmektedir. Özellikle acil durumlarda veya market alışverişine çıkılamayan anlarda bu yaklaşım, oldukça pratik bir çözüm olarak öne çıkmaktadır.

SuperCook, kiler takibi veya alışveriş listesi oluşturma gibi gelişmiş özellikleri barındırmasa da, temel işlevi olan “eldeki malzemeleri en verimli şekilde değerlendirme” görevini son derece başarılı bir biçimde yerine getirmektedir. Minimalist arayüzü, erişilebilirliği ve geniş tarif veri seti sayesinde, yemek yapma deneyimi sınırlı olan bireyler için de kullanıcı dostu bir seçenek sunmaktadır. Bu yönüyle SuperCook, kullanıcıların mevcut kaynaklarını değerlendirmeye yönelik sade ama etkili bir yapay zekâ temelli çözüm olarak değerlendirilebilir.

2.5 Özet ve Belirlenen Boşluklar

Günümüzde dünya genelinde mevcut olan yiyecek, malzeme ve tarif çeşitliliğinin artması, kullanıcıların kişisel ihtiyaçlarına uygun yemek tariflerine ulaşmasını zorlaştırmaktadır. Özellikle bireylerin damak tadı, diyet kısıtlamaları ve kültürel mutfak tercihleri göz önüne alındığında, manuel olarak tarif arama süreci hem zaman açısından verimsiz hem de sürdürülebilirlik açısından yetersiz kalmaktadır. Bu noktada yapay zekâ (YZ), sunduğu dinamik ve sürekli öğrenen yapısı sayesinde gıda öneri sistemlerinin önemli bir bileşeni hâline gelmiştir.

YZ tabanlı sistemler, yalnızca küresel çapta yaygın olarak bilinen tarifleri değil, aynı zamanda yerel ve kültürel olarak özgün yemekleri de analiz ederek kullanıcıya sunabilmektedir. Bu durum, farklı coğrafyalarda yaşayan kullanıcılar için kişiselleştirilmiş önerilerde bulunmayı mümkün kılarken, aynı zamanda kültürel mutfak çeşitliliğinin dijital ortamda korunmasına da katkı sağlamaktadır.

YZ'nin sağladığı en önemli katkılardan biri de malzeme erişilebilirliği problemine getirdiği çözümdür. Geleneksel tarif platformları, kullanıcıların ellerinde bulunmayan malzemeleri çoğu zaman dikkate almazken, YZ destekli sistemler, mevcut malzeme envanterine göre tarifleri otomatik olarak uyarlayabilmektedir. Bu özellik, kullanıcıya gerçekçi ve uygulanabilir öneriler sunarak hem mutfakta zaman ve emek tasarrufu sağlar hem de gıda israfının önüne geçilmesine yardımcı olur. Ayrıca bu sistemler, kullanıcıyı mutfakta daha yaratıcı olmaya teşvik ederek yemek hazırlama sürecini daha etkileşimli ve keyifli bir hâle getirmektedir.

Sonuç olarak, gıda öneri sistemlerinde YZ kullanımı, yalnızca bireysel kullanıcı deneyimini iyileştirmekle kalmayıp, aynı zamanda sürdürülebilir tüketim alışkanlıklarının geliştirilmesi ve dijital gıda kültürünün yaygınlaştırılması açısından da önemli bir gereklilik olarak öne çıkmaktadır.

3.SİSTEM MİMARİSİ VE TASARIMI

3.1 Genel Bakış

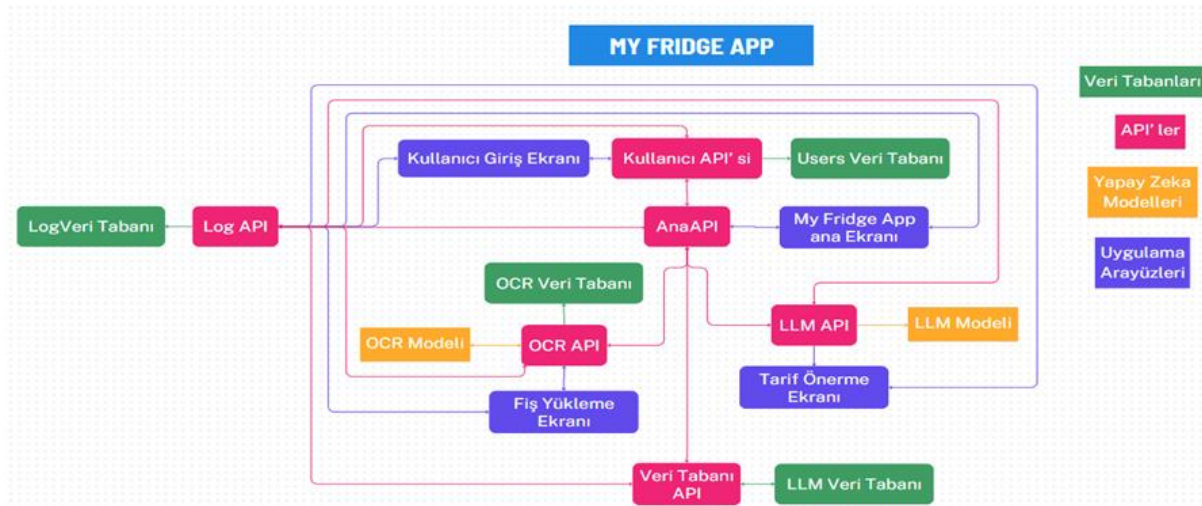
Bu projede geliştirilen yazılım sistemi, kullanıcıların market alışverişleri sonrası elde ettiği fişlerdeki ürün verilerini dijital ortama aktarmayı, ayrıştırmayı ve kullanıcılara ait sanal bir buzdolabında kategori bazlı olarak saklamayı amaçlamaktadır. Günlük yaşamdaki bu sıradan ihtiyacın dijitalleştirilerek daha işlevsel hale getirilmesi hedeflenmiş, bu doğrultuda modern yazılım mimarileri tercih edilmiştir. Proje mikroservis mimarisi prensiplerine uygun olarak yapılandırılmış, sistem beş temel bileşene bölünmüş ve bölünen her kısım bir API ile temsil edilmiştir bunlar: Main API, DB API, User API, OCR API, LLM API. Bu beş servis kendi sorumluluklarını yerine getirecek şekilde birbirinden ayrılmış, bu sayede hem hata izolasyonu sağlanmış hem de sistemin geliştirme, test ve bakım süreçleri kolaylaştırılmıştır.

3.2 Modüler Yapı ve Sorumluluklar

Sistemin modüler yapısı, her bir bileşenin kendi görevinden başka bir alana müdahale etmemesi ilkesine dayanarak tasarlanmıştır. Main API, OCR modelinden gelen JSON veri çıktısını alarak öncelikle veriyi ayrıştırır. Bu aşamada ürün adı, miktar, fiyat, marka ve son kullanma tarihi gibi bilgiler tespit edilir. Ürün ismi içerisindeki anahtar kelimelere bakılarak ürünün kategorisi belirlenir ve kullanıcıya anlamlı bir envanter sunulması sağlanır. Elde edilen bu yapılandırılmış veriler HTTP protokolü üzerinden DB API'ye iletilir. DB API, bu verileri alır, öncelikle veritabanında ilgili kategori ve ürünlerin mevcut olup olmadığını kontrol eder. Mevcut değilse ekleme, mevcutsa güncelleme işlemleri gerçekleştirilir. Ürünler user_fridge adlı tabloya user_id ile birlikte kaydedilir. User API ise, tüm sistem bileşenlerine kullanıcı doğrulaması ve yetkilendirme açısından hizmet sunar. Kullanıcıların sisteme kaydolması, giriş yapması ve kimlik bilgilerinin kontrolü bu yapı aracılığıyla gerçekleştirilir.

3.3 Servisler Arası İletişim

Sistem içerisinde yer alan servisler, RESTful mimari prensipleri doğrultusunda birbirleriyle iletişim kurmaktadır. Servisler arası veri alışverişi HTTP protokolü üzerinden, JSON veri formatı kullanılarak yapılmaktadır. Main API, OCR API sisteminden aldığı veriyi ayrıştırdıktan sonra POST metoduyla DB API'ye gönderir. Kullanıcının buzdolabındaki güncel ürün listesini öğrenmek için ise LLM API Main API aracılığıyla DB API'ye GET isteği gönderilir. Her veri alışverişinde, User API tarafından sağlanan user_id bilgisi kullanılır ve işlem yapan kullanıcının kimliği doğrulanır. Bu iletişim yapısı, sistemin merkezi bir yapıda koordineli ve uyumlu bir şekilde çalışmasını sağlamaktadır.



Şekil 3. API'ler ve modellerin iletişimini gösteren şema

3.4 Teknoloji Yığını

Geliştirilen sistem, Java programlama dili ile Spring Boot framework'ü kullanılarak oluşturulmuştur. API geliştirme sürecinde Spring Web modülü, veritabanı işlemlerinde ise Spring Data JPA kullanılmıştır. Başlangıçta sistem, test ortamı için H2 veritabanı üzerinde

çalıştırılmış, ilerleyen süreçte PostgreSQL gibi kalıcı ve çok kullanıcıli ortamlara uygun bir altyapıya geçilmesi planlanmıştır. Veriler, JSON formatında taşınmış ve API'ler arası iletişimde Spring WebClient kullanılmıştır. Geliştirilen sistem, Postman gibi manuel test araçları ile test edilmiş ve uçtan uca doğrulama senaryoları başarıyla gerçekleştirilmiştir. Her API, farklı bir portta çalışacak şekilde konfigüre edilmiştir: Main API 8080, DB API 8082, OCR 8081 LLM API 8083 User API 8084 portları üzerinden hizmet sunmaktadır. Bu port bazlı ayırım, geliştirme kolaylığı ve güvenlik açısından avantaj sağlamaktadır.

OCR ve LLM (Büyük Dil Modelleri) tabanlı akıllı bir veri işleme sistemi geliştirilmiştir. Python programlama dili, OCR çıktılarının işlenmesi ve LLM tabanlı yanıtların alınması amacıyla kullanılmıştır. Görüntüden metin çıkartma işlemleri için Google Vision API, özelleştirilmiş YOLOv8 modeli ve Tesseract teknolojileri entegre edilmiştir. Bu süreçte görüntü işleme görevleri, OpenCV ve NumPy kütüphaneleri ile desteklenmiştir. OCR sonrasında elde edilen veriler JSON formatında yapılandırılmış ve doğal dil işleme işlemleri için Google Gemini API, OpenAI API, Gemini API ve Hugging Face platformlarına ait modellerden faydalanılmıştır.

Uygulamanın arka uç (backend) yapısı, Java programlama dili ile Spring Boot çatısı kullanılarak geliştirilmiştir. REST API mimarisi aracılığıyla veri akışı ve sistem yönetimi sağlanmıştır. Geliştirme ve test aşamalarında H2 yerel veritabanı tercih edilmiştir. Uygulamanın test süreçleri, Postman ve Terminal CLI araçları ile yürütülmüştür. Ön yüz (frontend) geliştirmesi için Android Studio ile geliştirilmiş; hızlı prototipleme ve test amaçlı olarak Streamlit aracı alternatif olarak değerlendirilmiştir.

4. UYGULAMA DETAYLARI

Geliştirilen yazılım sisteminde, temel işlevlerden biri OCR sistemi aracılığıyla elde edilen ham fiş verisinin ayrıştırılması ve işlenmesidir. Bu veriler genellikle “ürün adıadet:fiyat” biçiminde sisteme ulaşmaktadır. Örneğin “Sütaş Yoğurt2:147” ifadesi sistem tarafından ayrıştırıldığında marka bilgisi "Sütaş", ürün tipi "Yoğurt", miktar "2 adet" ve fiyat "147 TL" olarak yorumlanmaktadır. Bu işlemler Main API üzerinde gerçekleştirilmekte ve ilgili veri parçalara ayrıldıktan sonra belirlenen kategoriye göre sınıflandırılmaktadır. Ürün ismindeki anahtar kelimeler yardımıyla kategori tayini yapılırken, sistem aynı zamanda ürünün miktar türünü (adet, litre, kilogram vb.) belirleyerek veriyi daha anlamlı hale getirmektedir. Ayrıca, fişten elde edilen bilgiler arasında son kullanma tarihi gibi öznitelikler de varsa, bu bilgiler sistem tarafından özel olarak tanınmakta ve ürün kaydına eklenmektedir. Tüm bu ayrıştırılmış ve anlamlandırılmış bilgiler, ilgili kullanıcıya ait user_id ile birlikte DB API'ye iletilmektedir. Böylece her kullanıcının kişisel ürün envanteri sistem içerisinde düzenli ve anlamlı bir yapıda tutulmaktadır.

Sistemdeki API altyapısı farklı portlar üzerinden çalışacak şekilde yapılandırılmıştır. Main API, DB API ve User API olmak üzere üç temel servis arasında net bir görev ayırımı yapılmış ve bu ayırım port bazlı olarak da desteklenmiştir. Main API 8080 portu üzerinden çalışırken, veritabanı işlemlerini üstlenen DB API 8082 portunda konumlandırılmıştır. Kullanıcı doğrulama ve kayıt işlemlerini yöneten User API ise 8084 portu üzerinden hizmet vermektedir.

Bu yapılandırma, hem geliştiricilere hem de sistem yöneticilerine esnek bir çalışma ortamı sunmakta, her API'nin bağımsız test edilebilmesine ve yönetilebilmesine olanak tanımaktadır. WebClient kütüphanesi kullanılarak Main API, DB API'ye doğrudan veri gönderebilmekte ve gelen yanıtları okuyabilmektedir ancak diğer hiçbir API doğrudan diğer API'lerle iletişim kurmamaktadır. API'ler arası tüm veri iletiminde kullanıcıya ait kimlik bilgisi olan user_id mutlaka iletilmekte ve sistem bu sayede kişiselleştirilmiş veri yönetimini sürdürebilmektedir.

4.0 Veri Seti Toplama

Veri seti oluşturma süreci, gerçek market fişlerinin toplanması ve bu fişlerin dijital ortama aktarılması ile başlatılmıştır. Bu kapsamda, farklı market zincirlerinden, çeşitli ürün kategorilerini içerecek şekilde toplamda 2500'ün üzerinde market fişi toplanmıştır. Gerçek dünya kullanım senaryolarını yansıtabilmek adına yalnızca düzgün fişler değil, aynı zamanda yıpranmış, solmuş veya kıvrılmış gibi farklı fiziksel koşullara sahip fişler de veri setine dâhil edilmiştir. Toplanan fişler, farklı ışık koşulları, kamera açıları ve çözünürlük seviyeleri içerecek biçimde mobil cihazlar aracılığıyla görüntülenmiştir. Bu çeşitlilik, geliştirilen OCR modelinin genelleme kabiliyetini artırmak ve gerçek hayatta karşılaşılabilecek çeşitli senaryolara karşı dayanıklı hâle getirmek amacıyla planlanmıştır. Görseller, ön işleme aşamasında gri tonlama, eşikleme (thresholding), segmentasyon gibi temel tekniklerle işlenmiş; ayrıca veri artırma (data augmentation) yöntemleri olarak döndürme (rotation), gürültü ekleme (noise), yer değiştirme (translation) ve yansıtma (flip) gibi işlemler uygulanarak model eğitimi için daha kapsamlı ve dengeli bir veri seti elde edilmiştir.

4.1 OCR/YOLOv8

Bu çalışmada, giriş görüntülerinde fişleri tespit etmek amacıyla nesne algılama omurgası olarak YOLOv8 kullanılmış, ardından metin bilgilerini çıkarmak için Tesseract tabanlı özel bir OCR hattı uygulanmıştır. Tam iş akışı; veri hazırlığı, model eğitimi, algılama değerlendirmesi ve OCR için son işlem adımlarından oluşmaktadır.

4.1.1 Veri Hazırlığı

Başlangıçta veri seti yaklaşık 400–500 fiş görüntüsünden oluşmaktaydı. Modelin dayanıklılığını ve genelleme kabiliyetini artırmak amacıyla kapsamlı veri artırma (data augmentation) teknikleri uygulanmıştır. Bu teknikler arasında rastgele döndürme, gürültü ekleme, kontrast ayarlamaları, kaydırmalar (translation) ve bulanıklaştırma işlemleri yer almıştır. Bu süreç sonucunda veri seti 1.007 görüntüye çıkarılmıştır.

4.1.2 YOLOv8 Eğitimi ve Değerlendirmesi

YOLOv8 modeli, toplam 1.007 adet fiş ve fiş olmayan görüntüden oluşan veri seti üzerinde 50 epoch boyunca eğitilmiştir. Bu görüntüler, orijinal 400–500 görüntüden türetilmiş olup yukarıda belirtilen veri artırma teknikleriyle çeşitlendirilmiştir. Bu sayede modelin farklı senaryolarda daha genel geçer ve sağlam bir şekilde çalışması hedeflenmiştir.

Modelin performansına dair temel değerlendirme metrikleri aşağıda sunulmuştur:

Ortalama IoU (Intersection over Union): 0.95

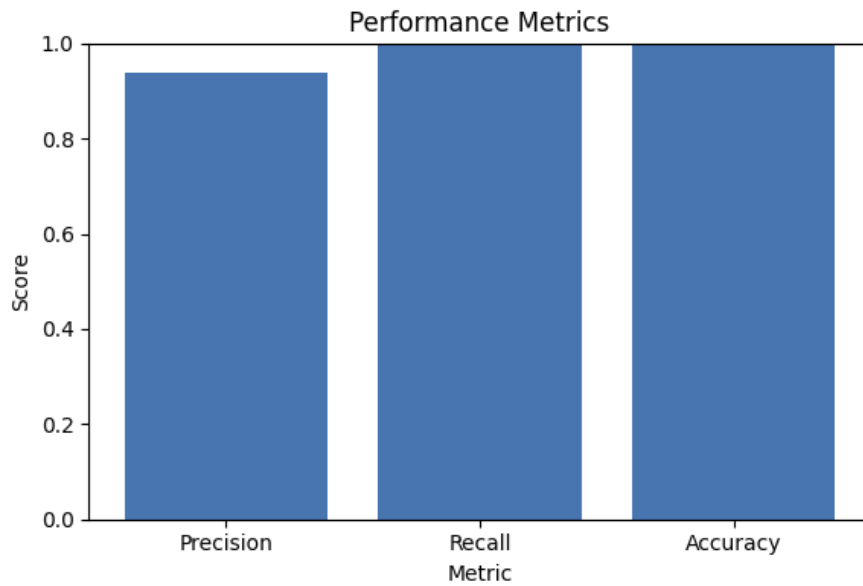
Precision (Kesinlik): 0.94
Recall (Duyarlılık): 1.00
Doğruluk (Accuracy): ~0.99

Metriklerin Yorumlanması:

Yüksek Recall (1.00): Model test setindeki tüm gerçek fişleri başarıyla tespit etmiştir. Başka bir deyişle, hiçbir fiş atlanmamıştır (false negative = 0). Bu, uygulamada fiş varsa YOLOv8'in bunu mutlaka algıladığını gösterir.

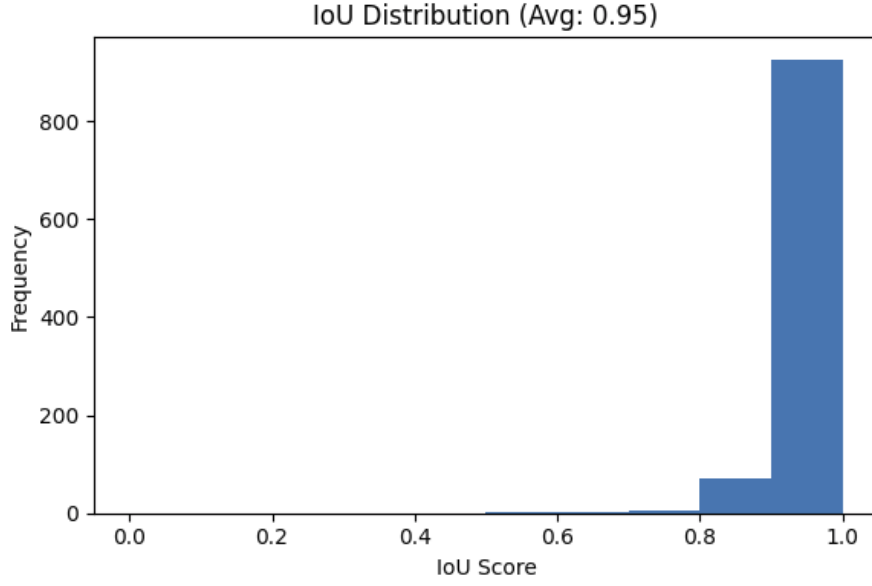
Precision (0.94): Modelin tahmin ettiği fişlerin %94'ü doğrudur. Yani %6 oranında, fiş olmayan görüntülerde yanlış tespitler yapılmıştır (false positive). Bu durum, Algılama Metrikleri grafiğinde de küçük ama gözle görülür sayıda yanlış pozitif olarak kendini göstermektedir.

Doğruluk (~%99): Hem fiş hem de fiş olmayan görüntüler dikkate alındığında, modelin genel sınıflandırma başarımı oldukça yüksektir.



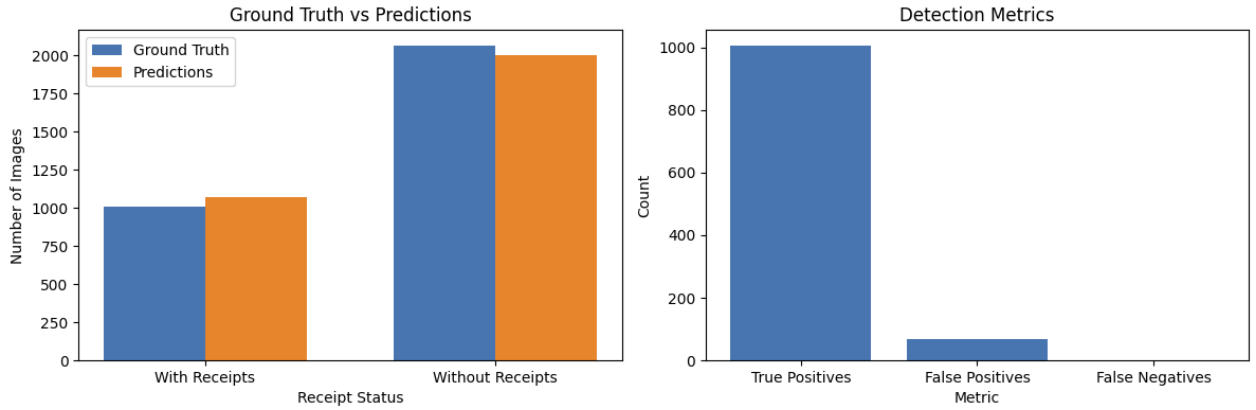
Şekil 4. OCR modelinin Performance Metric'lerini gösteren şekil

IoU Dağılımı: IoU (tahmin edilen kutu ile gerçek kutunun örtüşme oranı) dağılımı, 1.0'a oldukça yakın değerlere doğru yoğunlaşmıştır. Bu durum, fişlerin görüntüler içinde son derece doğru bir şekilde lokalize edildiğini göstermektedir. Histogramda tahminlerin büyük çoğunluğunun IoU değerinin 0.9'un üzerinde olduğu, yalnızca az sayıda tahminin 0.8'in altına düştüğü görülmektedir. Bu, nesne algılama başarımının yüksekliğini pekiştirir.



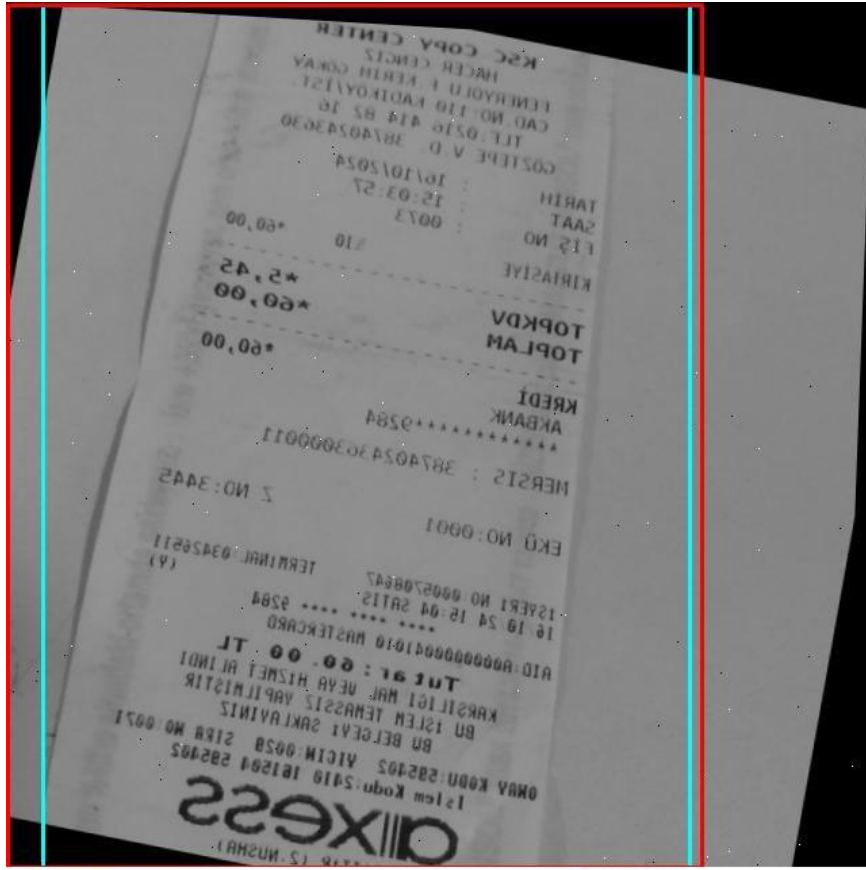
Şekil 5. IoU Distribution (dağılımı) grafiğini gösteren şekil

Gerçek Etiket & Tahminler: Sol üst köşedeki grafikte görüldüğü üzere, modelin tahmin ettiği fişli görüntü sayısı gerçek sayıya göre biraz fazladır. Bu durum, küçük sayıda false positive tespiti desteklemektedir. Ancak iki sayı birbirine oldukça yakındır, bu da modelin fazla tespit eğiliminde olmadığını ve genel dengeyi iyi koruduğunu gösterir.



Şekil 6 Etiket ve Tahmin sonuçlarını gösteren şekiller

Sonuç: Elde edilen sonuçlar, YOLOv8'in fiş tespiti görevinde son derece etkili olduğunu göstermektedir. Model, yüksek lokalizasyon başarımı (IoU), eksiksiz tespit kabiliyeti (Recall 1.00) ve düşük hata oranı (Precision 0.94) ile güçlü bir performans sergilemiştir. Bu özellikleri sayesinde YOLOv8, özellikle hiçbir fişin atlanmamasının kritik olduğu uygulamalarda, otomatik fiş işleme hattının ilk aşaması için ideal bir aday konumundadır.



GT Box GT Polygon Prediction

Şekil 7 OCR fiş görüntüsü

4.1.3 OCR İçin Görüntü Ön İşleme

Bir fiş başarıyla tespit edildikten sonra, OCR performansını artırmak amacıyla çeşitli görüntü ön işleme adımları uygulanmaktadır. İlk adım olarak, görüntü adaptif eşikleme (adaptive thresholding) yöntemiyle binarize edilerek siyah-beyaz hale getirilir. Bu işlem, metin bölgelerinin arka plandan daha belirgin hale gelmesini sağlar.

Ardından, görüntü üzerinde otomatik kırpma (cropping) işlemi gerçekleştirilir. Kırpma işlemi, görüntünün kenarlarında yer alan ve %98'den fazla beyaz piksel içeren bölgelerin tespitiyle yapılır. Bu tür alanlar genellikle bilgi içermeyen kenar boşlukları, arka plan veya gölge alanlarıdır. Bu sayede, OCR algoritmasının çalışacağı bölge daha net ve odaklanmış bir hale getirilir, bu da karakter tanıma doğruluğunu artırır.

Mgz Adı:Seyranfepe Malte / Mgz Kodu:E873
A101 YENİ MAGAZACILIK A.Ş.
Zümrütevler Mh.
Mevlana Halid Cad 93/12B
USKODAR/9480423762

TARİH : 29.10.2024
SAAT : 14:07
FİŞ NO : 0035

BİLGİ FİŞİ

TUR : E-ARSIV FATURA
MÜSTERİ TCKN : 11111111111

Belge No:14873022910424940035
ETTN NO:de771c77-6be1-4fcb-9836-f69bceb53a65

CEREZYA 150G Ç.LEBLE	%01	*25,00
TOPKDV		*0,25
TOPLAM		*25,00
KREDİ KARTI		*25,00

Yapıkredi Peşin (1)
Provizyon No : 256364
Puan:0,00
GARANTİ BANKASI
I:1364343 T:03003716 51704068****6024
29.10.2024 13:51 B:2632 S:8060
Rrn/Ref.No:430352569448 Ref.No:

14873022910424940035
FaturaAdres <https://fatura.a101.com.tr>
Mersis No: 0948042376200016
Tescilli İnternet Adresi: www.a101.com.tr
İRSALİYE YERİNE GEÇER

Şekil 9. Fiziki fiş görüntüsü

Mgz Adı:Seyranfepe Malte / Mgz Kodu:E873
A101 YENİ MAGAZACILIK A.Ş.
Zümrütevler Mh.
Mevlana Halid Cad 93/12B
USKODAR/9480423762

TARİH : 29.10.2024
SAAT : 14:07
FİŞ NO : 0035

BİLGİ FİŞİ

TUR : E-ARSIV FATURA
MÜSTERİ TCKN : 11111111111

Belge No:14873022910424940035
ETTN NO:de771c77-6be1-4fcb-9836-f69bceb53a65

CEREZYA 150G Ç.LEBLE	%01	*25,00
TOPKDV		*0,25
TOPLAM		*25,00
KREDİ KARTI		*25,00

Yapıkredi Peşin (1)
Provizyon No : 256364
Puan:0,00
GARANTİ BANKASI
I:1364343 T:03003716 51704068****6024
29.10.2024 13:51 B:2632 S:8060
Rrn/Ref.No:430352569448 Ref.No:

14873022910424940035
FaturaAdres <https://fatura.a101.com.tr>
Mersis No: 0948042376200016
Tescilli İnternet Adresi: www.a101.com.tr
İRSALİYE YERİNE GEÇER

Şekil 8. Ön işleme sonrası fiş görüntüsü

4.1.4 Yatay Segmentasyon

Temizlenmiş fiş görüntüsü, ardından yatay olarak segmentlere ayrılmaktadır. Bu işlem, piksel satırlarının analiziyle gerçekleştirilir. Bir satırın %99'dan fazlası beyaz piksellerden oluşuyorsa, bu satır ayırıcı (separator) olarak kabul edilir.

Bu ayırıcı satırlar arasındaki bölgeler, anlamlı metin blokları içerdiği varsayımıyla bağımsız segmentler olarak belirlenir ve her biri daha sonra işlenmek üzere ayrı ayrı saklanır. Bu segmentasyon işlemi, ürün adları ve fiyatlar gibi metinlerin konumlarını daha doğru belirlemek ve OCR doğruluğunu artırmak için kritik bir adımdır.

Horizontal Segment 1
Maz Adı:Sesantere Malle / Maz Kodu:E873
Horizontal Segment 2
ATOT YENI MAGAZACILIK A.Ş.
Horizontal Segment 3
Zöarötevier Ab.
Horizontal Segment 4
Nevlana Halid Cad 93/12B
Horizontal Segment 5
OSKODAR/9430423762
Horizontal Segment 6
TARİH : 29.10.2024
Horizontal Segment 7
SART : 14:07
Horizontal Segment 8
FİS NO : 0035
Horizontal Segment 9
BİLGİ FİŞİ
Horizontal Segment 10
TQR : E-ARSIV FATURA KÜSTERİ TCKN : 11111111111
Horizontal Segment 11
Belge No:14873022910424940035 ETTN NO:de771c77-6be1-4fcb-9836-f69bceb53a65
Horizontal Segment 12
CEREZYA 150G C.LEBLE %01 *25,00
Horizontal Segment 13
Horizontal Segment 14
TOPKDV *0,25
Horizontal Segment 15
TOPLAM *25,00 KREDİ KARTI *25,00 Yapıkredi Peşin (1)
Horizontal Segment 16
Provizyon No : 256364

Şekil 10. Segmente edilmiş fiş görüntüsü

4.1.5 Fiş İçeriği Bölgesinin Belirlenmesi

Oluşturulan her segment, Tesseract OCR ile işlenerek ürün listesinin başlangıç ve bitiş noktalarını belirleyen anahtar kelimeler tespit edilmeye çalışılır.

Başlangıç Göstergeleri: “ETTN NO”, “Belge No”

Bitiş Göstergeleri: “TOPKDV”, “TOPLAM”

OCR’nin hata yapma olasılığı göz önünde bulundurularak, bu anahtar terimlerin yaygın yanlış okunma varyantları da referans listesine dahil edilmiştir. Tanıma sürecinde, bir segmentin referans kelimelerle en az %80 oranında uyuşması durumunda, bu segment eşleşmiş olarak kabul edilir. Bu amaçla, bulanık eşleşme (fuzzy matching) yöntemi kullanılmaktadır. Bu yöntem sayesinde, OCR’nin küçük hatalarına rağmen doğru bölgenin belirlenmesi mümkün hale gelir.

Belge No:14873022910424940035
ETTN NO:de771c77-6be1-4fcb-9836-f69bceb53a65

Şekil 11.Fiş bölgelerinin başlangıcının belirlendiği şekil

Şekil 12.Fiş bölgelerinin sonunun belirlendiği

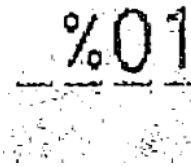
4.1.6 Dikey Segmentasyon ve Alan İzolasyonu

Ürün listesi sınırlarının belirlenmesinin ardından, başlangıç ve bitiş göstergeleri arasında kalan bölge yeniden birleştirilir ve bu bölge üzerinde dikey segmentasyon işlemi uygulanır. Tipik olarak bir fiş satırı üç ayrı satırdan oluşur:

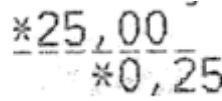
1. Ürün adı

*Şekil 13. Ürün adının alındığı segment şekli*

2. KDV oranı

*Şekil 14 .KDV oranının alındığı segment şekli*

3. Fiyat

*Şekil 15 .Fiyat bilgisinin alındığı segment şekli*

OCR işlemine yalnızca birinci (ürün adı) ve üçüncü (fiyat) satırlar dahil edilir; KDV oranı içeren satır filtrelendir.

Dikey segmentasyon sonucu üçten fazla segment oluşması durumunda, karakter sayısına dayalı bir filtreleme yöntemi uygulanır. Bu yöntem gereği, en fazla karakter içeren segment ürün isimleri, ikinci en fazla karakter içeren segment ise fiyat bilgisi olarak kabul edilir. Bu önlem, segmentasyon hatalarının etkisini azaltmak için geliştirilmiştir.

4.1.7 Nihai OCR Aşaması

Son aşamada, her bir dikey segment, bireysel ürünleri izole etmek amacıyla yatay olarak yeniden bölünür. Her ürün görüntüsünün etrafına 15 piksellik boşluk (padding) eklenerek OCR doğruluğu artırılır. Bu şekilde elde edilen ürün segmentleri, son metin çıkarımı için Tesseract OCR motoruna iletilir.

4.2 Ürün Ayırıştırma ve Kaydetme

Android uygulaması, kullanıcının malzeme eklemesi için iki temel yöntem sunmaktadır: manuel giriş ve fiş görüntüsü yükleme.

Manuel giriş yönteminde kullanıcılar, uygulama arayüzüne doğrudan malzeme isimlerini yazarak veri girişinde bulunur. Bu veriler, daha sonra bir Python tabanlı sunucuya iletilir ve burada Gemini adlı büyük dil modeli (LLM) kullanılarak işlenir. Bu adım, sistem genelinde malzeme adlarının tutarlı biçimde tanımlanmasını sağlar. Aynı zamanda gelecekte benzer isimli girdilerin birleştirilerek tekrarların önlenmesi ve veri kalitesinin artırılması gibi işlevler için de altyapı oluşturur. Gemini tarafından doğrulanan ve normalize edilen veriler, merkezi veritabanıyla iletişim kurmakla sorumlu olan ana API'ye aktarılır.

Buna karşılık, kullanıcı bir fiş fotoğrafı yüklemeyi tercih ettiğinde, daha karmaşık bir ön işleme süreci devreye girer. Yüklenen görsel, içeriğindeki metin bölgelerini tespit etmek için önce YOLO tabanlı nesne algılama modelinden geçirilir. Ardından, OCR işlemi uygulanarak bu bölgelerden metin verisi çıkarılır. Elde edilen ham metinler, tıpkı manuel girişte olduğu gibi Gemini tarafından doğrulanır ve düzeltilir. Son olarak, temizlenmiş ve yapılandırılmış veri, API aracılığıyla veri tabanına kaydedilir.

4.2.1 Metin Tanıma ve Temizleme

Fiş algılanıp ön işlemden geçirildikten sonra, görsel segmentlere ayrılarak Tesseract OCR'a gönderilir ve buradan metin çıkarımı yapılır. Ancak, değişken ışık koşulları, yazı tipi çeşitliliği ve görsel gürültü gibi etkenler, OCR işlemi sırasında hatalara neden olabilir. Bu hataları azaltmak amacıyla, ek bir doğrulama katmanı olarak Gemini-1.5 Flash kullanılmıştır.

Gemini, Tesseract'tan elde edilen her ürün adını anlam açısından değerlendirir ve semantik olarak geçerli olup olmadığını kontrol eder. Bu işlem özellikle düşük baskı kalitesine veya alışılmadık yazı tiplerine sahip fişlerde OCR motorunun üretmiş olabileceği anlamsız ya da hatalı çıktıları filtrelemek açısından kritiktir. Böylece sistem, yalnızca doğru metni değil, aynı zamanda anlamlı ve yorumlanabilir sonuçları da garanti altına alır.

Bu hibrit yaklaşım, kural tabanlı segmentasyon ile modern dil modellerini birleştirerek yalnızca daha yüksek metin tanıma doğruluğu sağlamakla kalmaz, aynı zamanda sonuçların güvenilirliğini ve anlam bütünlüğünü de artırır. Bu yapı sayesinde sistem, veri kalitesinin ciddi şekilde değişkenlik gösterebildiği gerçek dünya kullanım senaryolarında daha dayanıklı ve işlevsel hale gelir.

4.3 API Altyapısı

API altyapısının temel amacı, her bir bileşenin ayrı görev üstlenmesini ve merkezi bir yapı üzerinden koordineli biçimde çalışmasını sağlamaktır. Geliştirilen API türleri şunlardır:

OCR API: OCR çıktısını alır, işler ve merkezi API'ye iletir.

LLM API: Ürün listesini analiz ederek yemek tarifleri önerir.

Veri Tabanı API: Ürün ve tarif verilerinin yönetimini sağlar.

Kullanıcı Giriş API: Kullanıcı doğrulama işlemlerini ve kullanıcı bilgilerini yönetir.

Log API: Tüm işlem kayıtlarını tutar.

Merkezi API: Diğer tüm API'lerin koordinasyonunu sağlar.

Tüm API uç noktaları, Postman aracıyla test edilmiş; asenkron veri aktarımına uygun şekilde tasarlanmıştır. Bu mimari sayesinde sistemde herhangi bir bileşen arızalansa bile diğer bileşenlerin çalışması etkilenmemektedir.

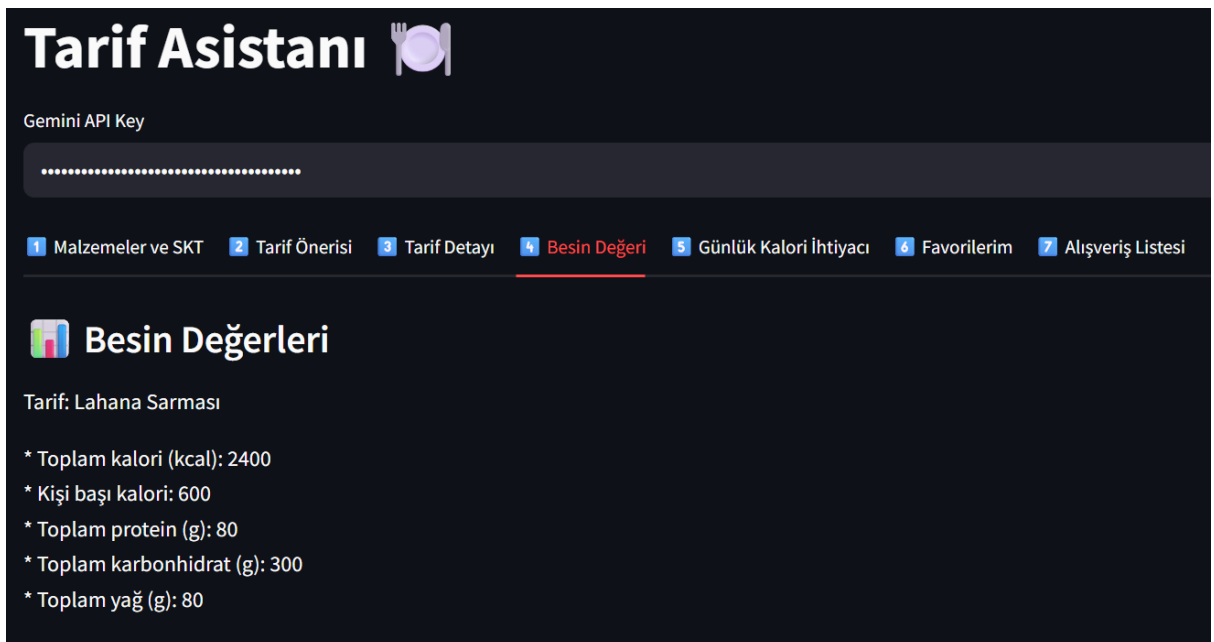
4.4 LLM Entegrasyonu

OCR modülü aracılığıyla elde edilen yapılandırılmış veriler, kullanıcıya özel tarif önerileri oluşturmak amacıyla LLM (Large Language Model) modülüne yönlendirilmiştir. Bu kapsamda, doğal dil işleme süreçlerinde çeşitli büyük dil modeli servisleri değerlendirilmiş; OpenAI GPT-3.5, Hugging Face ve Google Gemini API ile kapsamlı testler gerçekleştirilmiştir. Yapılan karşılaştırmalar sonucunda, kullanım senaryolarına en uygun, anlamlı ve tutarlı yanıtları üreten Google Gemini API tercih edilmiştir. İlk aşamada, OCR modülünden gelen ürün bilgileri, Gemini 1.5 Flash modeli aracılığıyla semantik olarak iyileştirilmiş ve doğal dil formatında daha anlaşılır ve yüksek kaliteli bir yapıya dönüştürülmüştür. Bu iyileştirilmiş veriler, tarif önerileri üretimine uygun bir yapıya getirilmiş ve bir kullanıcı profili sınıfı (UserProfile) içinde yapılandırılmıştır. Söz konusu sınıf, kullanıcının geçmiş verilerini ve mevcut ihtiyaçlarını göz önünde bulundurarak özgün bir prompt (girdi) oluşturmakta ve bu girdi, ilgili LLM API'sine gönderilmektedir.

Ön yüz uygulaması kapsamında Streamlit aracı kullanılarak API anahtarı (API key) üzerinden çalışan işlevsel bir arayüz geliştirilmiştir. Bu arayüz, sistemin OCR API entegrasyonu tamamlanmadan önce bir prototip olarak çalıştırılmış; böylece sistemin temel işleyişi, OCR bağlantısı olmadan bağımsız şekilde test edilerek değerlendirilmiştir. Bu yaklaşım, LLM modülünün doğruluğunu ve kullanıcı deneyimi açısından işlevselliğini ön değerlendirme aşamasında gözlemlemeyi mümkün kılmıştır. Testler sonucunda çalışan model Android studio (mobil uygulama) tarafına bağlanmıştır.

4.4.1 Streamlit Arayüzü

Besin Değeri



Şekil 16 Besin değeri şekli

Tarif Önerisi

Tarif Asistanı

Gemini API Key

.....

1 Malzemeler ve SKT 2 **Tarif Önerisi** 3 Tarif Detayı 4 Besin Değeri 5 Günlük Kalori İhtiyacı 6 Favorilerim 7 Alışveriş Listesi 8 Moduma Göre Tarif

Tarif Önerisi Al

Önerilen Tarifler

Önerilen 3 Tarif:

- Köfte (2-3 Kişi)
- Lahana Sarması (4 Kişi) (Eksik Malzeme: Pirinç, soğan, salça)
- Domates Çorbası (2-3 Kişi) (Eksik Malzeme: Su, yağ, tuz, baharatlar)

Yanında içecek olarak Hazır Çorba tüketilebilir.

Tarif seçiniz:

☐ 1

☒ 2

☐ 3

Şekil 17. Tarif Önerisi Şekli

Günlük Kalori İhtiyacı

Tarif Asistanı

Gemini API Key

.....

1 Malzemeler ve SKT 2 Tarif Önerisi 3 Tarif Detayı 4 Besin Değeri 5 **Günlük Kalori İhtiyacı** 6 Favorilerim 7 Alışveriş Listesi 8 Moduma Göre Tarif

Günlük Kalori İhtiyacı Karşılaştır

Cinsiyet

erkek

Yaş

25

Kilo (kg)

70

Boy (cm)

170

Aktivite düzeyi

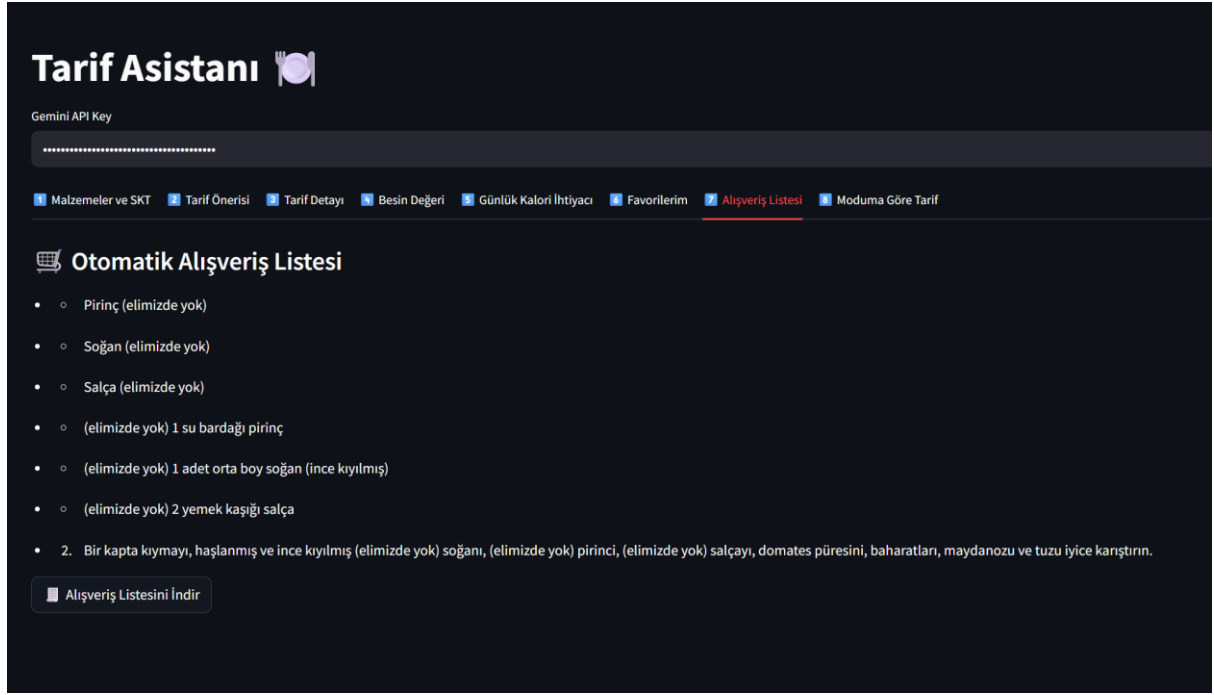
düşük

Kalori Karşılaştır

Tahmini günlük kalori ihtiyacınız: 1971 kcal

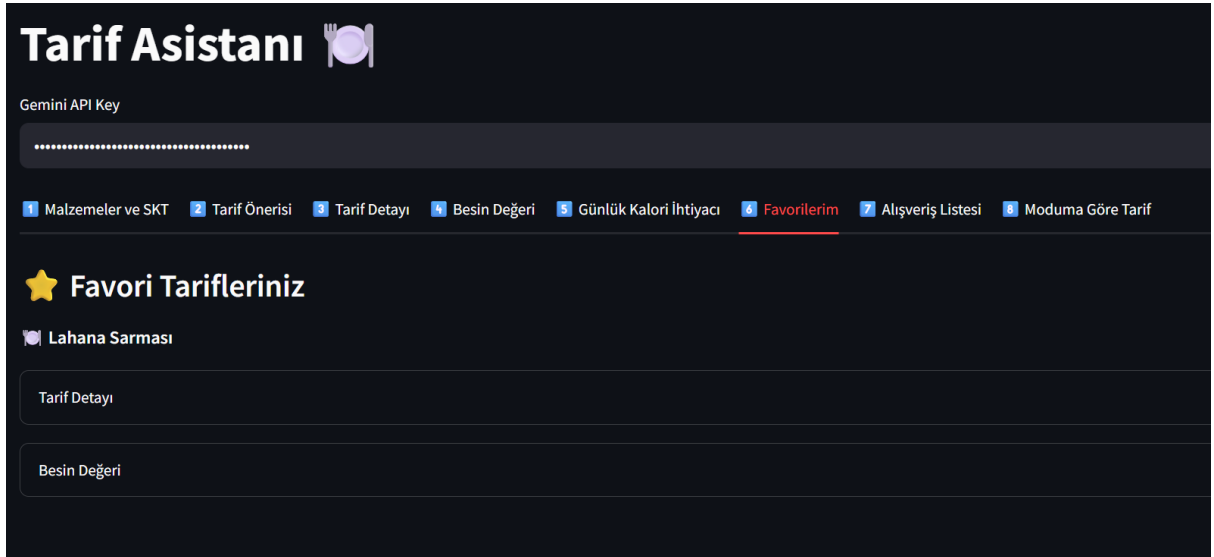
Lahana Sarması yemeği günlük ihtiyacın %30.4'ini karşılar.

Şekil 18 Günlük Kalori İhtiyacı



Şekil 19 Alışveriş Listesi şekli

Favorilerim



Şekil 20 Favorilerim şekli

4.5 Android Uygulaması

My Fridge sisteminin Android bileşeni, son kullanıcılar için birincil arayüz işlevi görür ve akıllı buzdolabı altyapısıyla gerçek zamanlı etkileşim sağlayarak yiyecek yönetimi deneyimini iyileştirir. En güncel Android teknolojileri kullanılarak geliştirilen bu uygulama, modüler yapı, yüksek performans ve kullanıcı dostu arayüz prensiplerini temel alır. Projede, endüstri standardı mimari desenler ve modern kütüphaneler benimsenerek ölçeklenebilirlik ve sürdürülebilirlik sağlanmıştır.

4.5.1 UI/UX Design

My Fridge Android uygulamasının kullanıcı arayüzü, basitlik, anlaşılabilirlik ve hızlı tepki verme odaklı tasarlanmıştır ve Material Design 3 yönergeleriyle uyumlu biçimde geliştirilmiştir. Arayüz, Android'in modern deklaratif UI çerçevesi olan Jetpack Compose ile inşa edilmiştir. Bu sayede hem geliştirme süreci sadeleşmiş hem de tekrar kullanılabilir UI bileşenleri oluşturulmuştur.

Tasarımın Öne Çıkan Noktaları:

Tutarlı Düzenler: Yeniden kullanılabilir composable bileşenler ve tema yapısı ile renkler, tipografi, boşluklar ve şekil dili uygulama genelinde tutarlıdır.

Navigasyon: Navigation Compose kullanılarak ekran geçişleri sorunsuz hale getirilmiş, merkezi bir navigasyon grafiği ile ekran yönetimi kolaylaştırılmıştır.

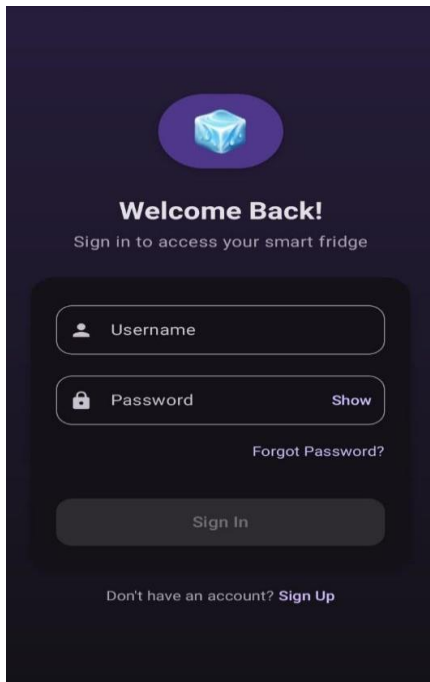
Duyarlı Tasarım: Arayüz farklı ekran boyutlarına ve yönlerine uyum sağlar. Constraint'ler ve modifier'lar aracılığıyla bileşen hizalamaları dengelenmiştir.

Kullanıcı Akışı: Kullanıcının uygulamadaki yolculuğu (kayıt, giriş, buzdolabı envanteri görüntüleme, tarif alma) akıcı ve sezgisel biçimde tasarlanmıştır.

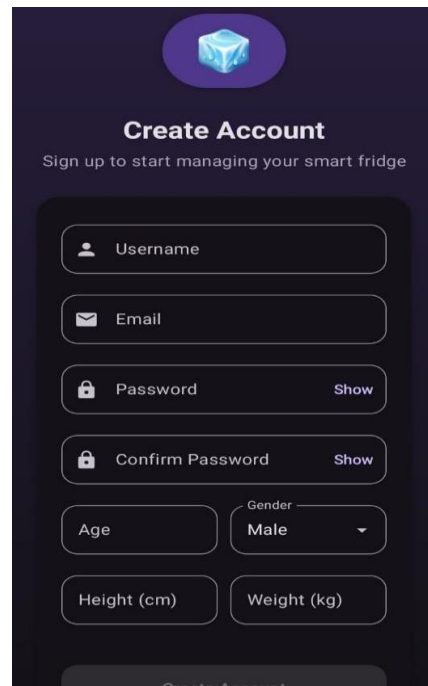
Erişilebilirlik: Yazı tipi ölçeklenebilirliği ve semantik etiketleme ile TalkBack gibi erişilebilirlik araçları desteklenmiştir.

Temel Ekranlar:

Giriş/Kayıt Ekranı: API üzerinden kimlik doğrulama ve güvenli giriş.

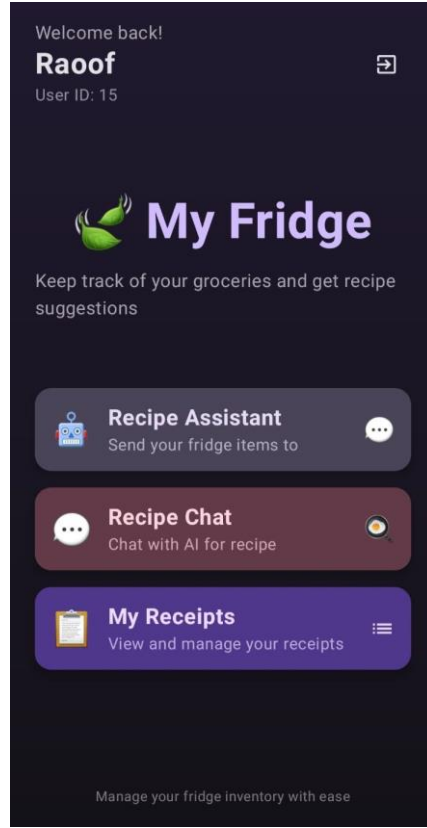


Şekil 22 Kullanıcı Giriş Ekranını gösteren şekil



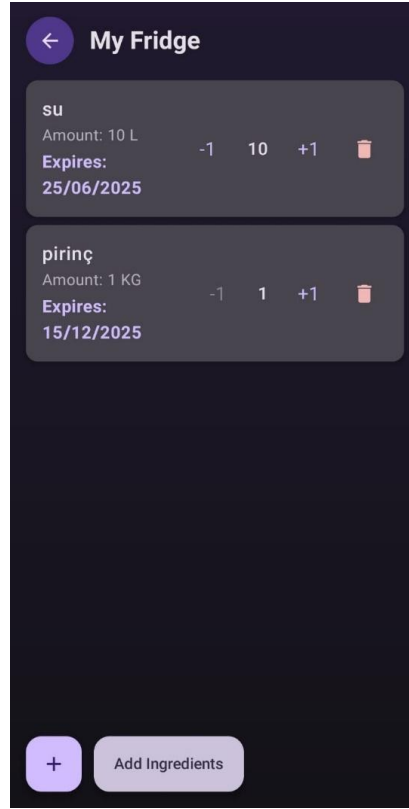
Şekil 21 Kullanıcı kayıt ekranını gösteren şekil

Ana Sayfa: Ürünler, son tarifler ve navigasyon kısayollarını içeren merkezi gösterge paneli.



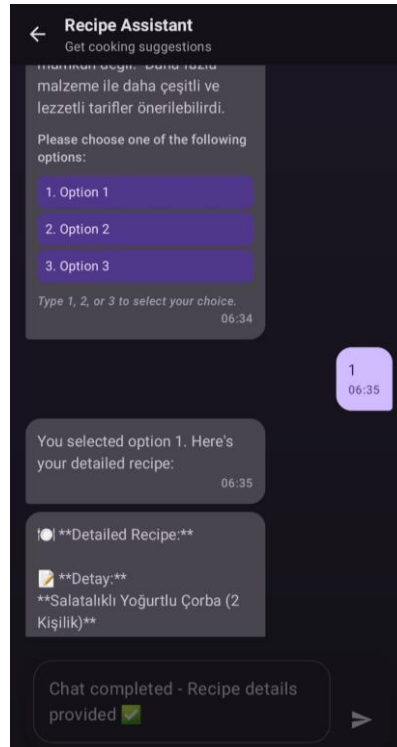
Şekil 23. Kullanıcı anasayfasını gösteren şekil

Fiş Listesi (Tarifler) Ekranı: Kullanıcının envanterine göre özelleştirilmiş tariflerin kaydırılabilir listesi.



Şekil 24. Kullanıcının özelleştirilmiş tariflerin kaydırılabilir listesinin şekli

Tarif Sohbet Ekranı: Adım adım yemek tarifi almayı sağlayan etkileşimli sohbet arayüzü.



Şekil 25. Adım adım yemek tarifi almayı sağlayan etkileşimli sohbet arayüzü şekli

4.5.2 API İstekleri ve İşleme

Arka uç servisleriyle tüm iletişimler, Retrofit 2 kullanılarak yapılan iyi yapılandırılmış API çağrıları üzerinden gerçekleştirilmiştir. Retrofit, HTTP iletişimini soyutlayarak geliştiriciye daha sade ve güvenilir bir arayüz sunar.

Base URL ve Uç Noktalar: Retrofit arayüzleri ve @GET, @POST gibi açıklayıcı anotasyonlar ile tanımlanmıştır.

Veri Serileştirme: Gson ve isteğe bağlı olarak Kotlinx Serialization kullanılarak JSON verileri Kotlin veri sınıflarına dönüştürülür.

Durum Yönetimi: ViewModel'ler, arayüz durumlarını MutableStateFlow ve StateFlow ile UI katmanına reaktif şekilde iletir.

Hata Yönetimi: API yanıtları Result adlı sealed class (Success, Error, Loading) ile sarmalanarak ağ hataları, zaman aşımaları ve geçersiz istekler yönetilir.

Coroutines: Tüm API çağrıları suspend fonksiyonlar ve Kotlin Coroutines ile asenkron biçimde yürütülür, böylece ana iş parçacığı bloklanmaz.

Bağımlılık Enjeksiyonu: Ağ bileşenleri dahil tüm bağımlılıklar, test edilebilir ve ayrıştırılabilir yapılar oluşturmak için Dagger Hilt ile sağlanır.

Örnek İşleyiş Akışı:

1. ViewModel, repository üzerinden getRecipes() API çağrısını tetikler.
2. Repository, Retrofit üzerinden veriyi alır ve sonucu StateFlow aracılığıyla yayar.
3. UI katmanı bu akışı gözlemler ve en güncel tarif verileriyle LazyColumn bileşenini otomatik olarak günceller.

4.5.3 Tarif ve Ürünlerin Görselleştirilmesi

Uygulama, hem kullanıcıya ait buzdolabı ürünlerini hem de bu ürünlere dayalı önerilen tarifleri görsel ve kullanıcı dostu bir biçimde sunar. Bu özellik, kullanıcıların envanter durumunu anlayıp yemek planlaması yapmalarına yardımcı olarak temel kullanıcı deneyimini oluşturur.

Tarif Görselleştirmesi:

Tarifler, LazyColumn içinde kart biçiminde görüntülenir ve her bir kart aşağıdaki bilgileri içerir:

Yüksek kaliteli görsel (Coil ile yüklenir)

Tarif adı, kısa açıklama ve içerik önizlemesi

Ayrıntılı tarif sayfasına ya da sohbet destekli tarif yardımına geçiş için tıklanabilir alan

Tarifler, kullanıcı envanteri, beslenme tercihleri ve ürün tazeliği gibi verilere dayalı olarak dinamik şekilde getirilir.

Ürün Görselleştirmesi:

Buzdolabındaki ürünlerin görsel temsili sunulur:

Ürün adı, kategori, miktar ve son kullanma tarihi

Ürünler gruplandırılır ve örneğin kırmızı renkle işaretlenerek yakında süresi dolacak olanlar vurgulanır

Kaydırılabilir ve filtrelenebilir liste desteği ile detaylı envanter takibi sağlanır

Ek Özellikler:

Canlı Güncellemeler: ViewModel akışları sayesinde ürün listesi ve tarifler gerçek zamanlı olarak yenilenir.

Etkileşimli Öğeler: Kullanıcılar tarifleri favorilere ekleyebilir, ürünleri silebilir ya da yeni veri çekmek için butonlar ve kaydırma hareketlerini kullanabilir.

Bu dinamik ve reaktif görüntüleme sistemi, kullanıcıların yemek planlamasını verimli bir şekilde yapmalarına olanak tanır ve gıda israfını azaltmaya yardımcı olur.

5.DENEYSEL KURULUM VE DEĞERLENDİRME

Sistemin performansını ve güvenilirliğini ölçmek amacıyla yapılan testler sonucunda, sistemin genel olarak kararlı ve hızlı bir şekilde çalıştığı gözlemlenmiştir. Yapılan performans testlerinde, veri iletim süresi ortalama 200 ila 300 milisaniye arasında ölçülmüş, bu da gerçek zamanlı kullanım açısından oldukça yeterli bir süredir. Testler hem doğru hem de hatalı veri giriş senaryoları kullanılarak yapılmış, böylece sistemin farklı durumlar karşısındaki dayanıklılığı gözlemlenmiştir. Özellikle eş zamanlı çoklu istekler karşısında sistemin cevap verme kapasitesi test edilmiş ve kararlılık açısından önemli bir sorunla karşılaşmamıştır. Ayrıca, sistem hata senaryolarında kullanıcıya anlaşılır ve yönlendirici hata mesajları döndürerek, kullanıcı deneyimini artıran bir tasarım sunmuştur.

Kullanıcı testleri sırasında sistemin edge case olarak adlandırılan istisnai durumlara verdiği tepkiler de değerlendirilmiştir. Özellikle boş ürün ismi, eksik tarih bilgisi ya da sistemde tanımlanmamış kategori girdileri gibi durumlar test senaryolarına dahil edilmiştir. Bu gibi durumlarda sistemin hata yönetimini başarılı bir şekilde gerçekleştirdiği görülmüştür. Örneğin, ürün kategorisi belirlenemediğinde sistem ya kullanıcıya bilgi vererek işleme devam etmiş ya da varsayılan bir kategori belirleyerek sistemin sürekliliğini korumuştur. Bu durum, sistemin sadece normal koşullarda değil aynı zamanda beklenmedik girdiler karşısında da tutarlı bir biçimde çalıştığını ortaya koymaktadır.

5.1 API Performansı (Gecikme ve Güvenilirlik)

API sistemleri performans testlerinde de gözlemlendiği üzere 200-300 milisaniye arasında iletişime geçmektedir. Tüm sistemlerin ayağa kalkması 11-15 saniye arasında yer almaktadır. Bu sistemin hızlandırılması adına optimizasyon işlemleri denenmiş ve 20 saniyelik süreçler de 15 saniye bandına indirgenmiştir. Veri yoğunluğu ile de gerekli testler yapılmış ve Postman ile 20 satırlık veri ile test edildiğinde ek gecikmesiz işlem yapabildiği gözlemlenmiştir.

Güvenlik hususunda kullanıcı bilgileri ve şifreleri tamamen ayrı bir veri tabanı ile sistemde tutulmuş ve hashlenmiştir. Kullanıcı bazında hassas olabilecek telefon numarası ve isim gibi bilgilerin alınmaması tercih edilmiştir. Kişinin buzdolabına ait bilgileri ise tamamen ayrı bir sistemde tutulmakta olup direkt android uygulamasından da herhangi farklı bir sistemden de erişimi olmamaktadır. Doğrudan veri tabanına ‘INSERT’ ve ‘DELETE’ gibi işlemler yapılmamaktadır. Bu sayede de sistemi zorlayabilecek bazı durumların önüne geçilmektedir. Ayrıca Android Studio ve Spring Framework’ ün sunduğu temel güvenlik özellikleri de kullanılmıştır.

5.2 LLM Çıktı Değerlendirmesi

Geliştirilen sistemde kullanılan LLM (Large Language Model) modülünün çıktıları, çeşitli niteliksel ölçütler doğrultusunda değerlendirilmiştir. Bu kapsamda; içerik tutarlılığı, kullanıcı tercihlerine uygunluk ve dil kalitesi gibi kriterler esas alınmıştır. Değerlendirme sürecinde farklı market fişi profilleri oluşturulmuş ve bu veriler kullanılarak LLM modellerinden tarif üretmeleri istenmiştir.

5.2.1 Tariflerin Anlamlılığı ve Kullanışlılığı

Üretilen tariflerin büyük çoğunluğu, sistemde tanımlı olan veri tabanındaki malzemelere dayalı olarak anlamlı ve uygulanabilir şekilde oluşturulmuştur. Bununla birlikte, bazı senaryolarda tariflere sistem dışı malzemelerin dâhil edildiği gözlemlenmiştir. Bu durum, kullanılan prompt ifadesinin daha dikkatli ve optimize şekilde tasarlanması ile en aza indirgenebilir. Ayrıca, önerilen tariflerin ortalama hazırlık süresi incelendiğinde, genellikle 20–40 dakika aralığında yer aldığı ve bu sürenin kullanıcı açısından makul olduğu belirlenmiştir.

5.2.2 Dil Kalitesi ve Anlaşılabilirlik

LLM modülü tarafından üretilen tarif açıklamaları, dilsel açıdan genel olarak akıcı, anlaşılır ve kullanıcı dostu bir yapıdadır. Ancak bazı durumlarda, tariflerde geçen teknik terimlerin daha sade ve açık ifadelerle açıklanmasının gerekli olabileceği tespit edilmiştir. Farklı modellerle yapılan karşılaştırmalı değerlendirmelerde, dil kalitesi açısından en başarılı sonuçlar Google Gemini API ile elde edilmiştir. Bu API’nin, hem dil akıcılığı hem de içerik yapısı bakımından kullanıcı beklentilerini karşıladığı gözlemlenmiştir.

6.TARTIŞMA

Yapılan değerlendirmeler sonucunda sistemin mimari yapısının, özellikle mikroservis temelli API tasarımının proje açısından büyük avantajlar sunduğu görülmüştür. Her bir servisin tekil sorumluluklara sahip olması, sistemde yapılacak güncellemelerin veya düzeltmelerin yalnızca ilgili servisi etkilemesini sağlamış, böylece tüm sistemin kararlılığı korunmuştur. Özellikle DB API üzerinde yapılan herhangi bir değişikliğin Main API’yi doğrudan etkilememesi, modüler yapı sayesinde elde edilen önemli bir avantajdır. Bu yapı, ilerleyen süreçlerde sistemin yeni servislerle genişletilmesine veya mevcut servislerin yeniden yapılandırılmasına olanak tanımaktadır. PostgreSQL gibi güçlü bir veritabanı altyapısına geçiş planı, sistemin daha büyük veri hacimlerini yönetebilmesini sağlayacak, böylece kullanıcı sayısının ve veri çeşitliliğinin

artması durumunda da sistemin performansından ödün verilmemiş olacaktır. Ayrıca Docker teknolojisi ile her bir API'nin container ortamında izole şekilde çalıştırılması, dağıtık mimariyle uyumlu ve daha esnek bir yapı kazandıracaktır.

API tabanlı mimari, aynı zamanda geliştirici ekipler açısından paralel çalışabilme imkânı sunmuş, bu da geliştirme sürecini hızlandırmıştır. Her bir API kendi başına geliştirilip test edilebildiğinden, bir modülde yapılan değişiklik diğer bileşenlerde beklenmeyen hatalara yol açmamaktadır. Bu da sistemin sürdürülebilirliğini artırmaktadır. Kullanıcı güvenliği açısından değerlendirildiğinde ise, sisteme giriş yapan her kullanıcının kendisine ait bir kimlik ile işlem yapması ve User API üzerinden yapılan doğrulama işlemlerinin sistemin tamamını koruyan bir güvenlik katmanı oluşturması dikkat çekicidir. Gelecekte HTTPS protokolü ve token-based authentication sistemlerinin entegre edilmesiyle birlikte sistemin veri gizliliği ve bütünlüğü daha da üst düzeye taşınacaktır.

6.1 Sistem Performansı ve Ölçeklenebilirlik

Yapılan testler ve uygulama analizleri, geliştirilen sistemin mevcut kullanıcı senaryolarında yeterli performans gösterdiğini ortaya koymuştur. LLM tarafından üretilen tarifler ise büyük oranda kullanılabilir ve anlaşılabilir içerikler sunmuştur.

Ancak sistem, yüksek yoğunluklu veri girişlerinde (örneğin arka arkaya çok sayıda fiş yüklenmesi gibi) zaman zaman gecikme yaşayabilmektedir. Bu durum, özellikle merkezi API'nin tek sunucu üzerinden çalışıyor olmasından kaynaklanmaktadır. Bu nedenle, gelecekte yük dengeleme (load balancing), mikroservis mimarisi ve bulut tabanlı çoklu sunucu dağıtımı gibi yöntemlerin uygulanması, sistemin ölçeklenebilirliğini önemli ölçüde artıracaktır.

6.2 API Tabanlı Tasarımın Güçlü Yönleri

Sistemde kullanılan katmanlı API mimarisi, modülerlik, esneklik ve bakım kolaylığı açısından ciddi avantajlar sağlamıştır. Her bileşenin kendi API üzerinden bağımsız çalışması sayesinde sistemde bir modülde oluşabilecek hata, diğer modülleri etkilememektedir.

Özellikle OCR API ve LLM API'nin birbirinden ayrılması hem test sürecini kolaylaştırmış hem de gerektiğinde bu bileşenlerin farklı modellerle değiştirilmesini mümkün kılmıştır. Ayrıca, kullanıcı arayüzü ile arka uç bileşenleri arasında iletişim, platformdan bağımsız geliştirmelere olanak tanımaktadır. Bu tasarım yaklaşımı, sistemin sürdürülebilirliği açısından önemli bir temel sunmaktadır.

6.3 Büyük Dil Modellerinin Kullanımında Karşılaşılan Zorluklar

6.3.1 İçerik Uygunluğu ve Malzeme Tutarlılığı

LLM kullanımında karşılaşılan temel zorluklardan biri, modelin kullanıcıya özel, mevcut veriyle uyumlu ve tutarlı içerikler üretebilme yeteneğidir. Bazı durumlarda modelin veritabanında bulunmayan veya kullanıcının elinde olmayan malzemeleri tariflere dâhil ettiği gözlemlenmiştir. Bu durum, sistem çıktılarının güvenilirliğini ve pratik kullanılabilirliğini olumsuz etkileyebilmektedir.

Bu sorunun başlıca nedeni, LLM'e gönderilen prompt yapısının yeterince kısıtlayıcı ve yönlendirici olmaması olarak değerlendirilmiştir. Bu nedenle, ileri düzey prompt mühendisliği tekniklerinin uygulanması ve modelin içerik üretimini sınırlandıracak yapay kuralların oluşturulması gerekliliği ortaya çıkmıştır.

6.3.2 Çıktıların Kontrol Edilebilirliği ve Tutarlılığı

LLM'lerin çıktıların kontrol edilebilirliği de sistemin kalitesi açısından önemli bir zorluk olarak öne çıkmaktadır. Özellikle Hugging Face gibi açık kaynaklı modeller kullanıldığında, bazı çıktılarda dilsel karmaşa, yapısal hatalar veya anlamsal tutarsızlıklar meydana gelebilmektedir. Önerilen tariflerde kullanılan ifadelerin bazen kullanıcı tarafından anlaşılabilirlikten uzak olması, deneyimi olumsuz etkileyen bir unsur hâline gelmiştir. Bu sorunları en aza indirmek amacıyla, sistemde farklı LLM platformları karşılaştırmalı olarak test edilmiş ve dilsel doğruluk, anlamlılık ve tutarlılık açısından en başarılı sonuçları veren Google Gemini API tercih edilmiştir. Bu geçiş hem içerik kalitesini artırmış hem de model çıktıların kullanıcı odaklı olmasını sağlamıştır.

6.3.3 API Kota Sınırlamaları ve Erişim Kısıtları

Çalışma kapsamında karşılaşılan bir diğer önemli zorluk, LLM servis sağlayıcılarının uyguladığı API kota sınırlamaları olmuştur. Özellikle sistemin test aşamasında OpenAI API kullanımında belirlenen günlük ve aylık çağrı kotalarının hızla dolması, modelin sürdürülebilir biçimde test edilmesini zorlaştırmıştır. Bu durum, sistemin sürekliliğini ve güvenilirliğini olumsuz yönde etkilemiş; alternatif modellere yönelme gerekliliğini doğurmuştur. Bu bağlamda, kullanım limitleri açısından daha esnek ve kararlı bir çözüm sunan Google Gemini API'ye geçilmiştir. Bu deneyim, yalnızca teknik performans değil, aynı zamanda API hizmet sağlayıcılarının erişim politikaları ve maliyet planlarının da sistem seçiminde belirleyici rol oynadığını göstermiştir.

6.4 Veri Güvenliği ve Gizlilik

Kullanıcıdan toplanan market fişi verileri, ürün içerikleri ve kullanıcı profilleri gibi bilgiler, hassas veri kategorisine girmektedir. Bu bağlamda, sistemde veri güvenliğine yönelik bazı temel önlemler alınmıştır. Örneğin:

OCR çıktıları ve kullanıcı profilleri, sadece oturum açmış kullanıcılarla ilişkilendirilmekte;

Veriler yerel veri tabanına (H2) ve test ortamında şifrelenmiş dizinlere kaydedilmektedir;

API'ler arası iletişimde HTTPS protokolü ve erişim denetimleri kullanılmaktadır.

Ancak, sistem henüz tam ölçekli kullanıcı veri güvenliği politikalarına (örneğin KVKK veya GDPR) tam uyumlu değildir. Bu durum, uygulamanın akademik bir prototip olmasıyla ilişkilidir. Gelecekte uygulamanın ticarileştirilmesi veya yaygın kullanıma sunulması hâlinde, kullanıcı verilerinin bulut ortamında anonimleştirilmesi, veri şifreleme ve iki faktörlü kimlik doğrulama gibi ek önlemlerin sisteme entegre edilmesi gerekmektedir.

7.SONUÇ VE GELECEK ÇALIŞMALAR

Bu çalışma, proje ekibindeki tüm üyeler açısından önemli bir öğrenme ve gelişim süreci olmuştur. Projenin gerçekleştirilmesi sırasında daha önce deneyim sahibi olunmayan birçok yeni teknoloji ve metodoloji ile doğrudan çalışma imkânı doğmuştur. OCR nesne tespiti, büyük dil modelleri (LLM) ve mobil uygulama geliştirme gibi çok çeşitli teknik alanlar üzerinde çalışılmış; her bir alan, kendine özgü zorluklar ve çözüm gereksinimleri ile süreci şekillendirmiştir.

Bu çok boyutlu teknik keşif süreci, her ne kadar başlangıçta zorlu bir öğrenme eğrisi gerektirse de, proje sonunda edinilen bilgi ve beceriler hem tatmin edici olmuş hem de ileriye dönük önemli bir yatırım niteliği taşımıştır. Ekip üyeleri yalnızca teorik bilgiler edinmekle kalmamış; aynı zamanda bu bilgileri uygulama düzeyinde kullanma fırsatı bularak, gelecekteki akademik çalışmalarında ve profesyonel kariyerlerinde doğrudan fayda sağlayabilecek yetkinlikler geliştirmiştir.

Bu bağlamda, projenin yalnızca teknik bir ürün ortaya koymanın ötesine geçerek, çok disiplinli düşünme, problem çözme ve ekip içi iş birliği gibi becerileri de pekiştirdiği söylenebilir. Gelecek çalışmalarda, özellikle sistemin kullanıcı dostu arayüzünün geliştirilmesi, veri güvenliği optimizasyonu ve farklı dil modelleriyle entegrasyon gibi alanlarda derinlemesine iyileştirmeler yapılması hedeflenmektedir.

7.1 Elde Edilen Başarılar

Veri Seti Oluşturulması ve Optimizasyonu

Proje kapsamında sıfırdan özel bir veri seti inşa edilmiş, sayısallaştırılmış ve YOLOv8 modeline özgü, hedeflenmiş bir ön işleme ve veri artırma stratejisi geliştirilmiştir. Bu strateji, modelin gerçek dünya koşullarında yüksek doğrulukla çalışabilmesini sağladı.

YOLOv8 Nesne Tanıma Modelinin Eğitimi

YOLOv8 mimarisi temel alınarak, gıda ürünlerini ve market fişlerini yüksek doğruluk oranıyla tanıyabilen bir nesne tanıma modeli başarıyla eğitilmiş ve optimize edilmiştir. Bu model, uygulamanın temel yapı taşlarından birini oluşturmaktadır.

OCR Entegrasyonu

Projeye entegre edilen OCR sistemi sayesinde, görsellerde yer alan yazılı içerikler okunabilir hâle getirilmiş; bu da fişlerden ürün bilgilerinin otomatik çıkarımını mümkün kılmıştır. Her ne kadar mutlak doğruluk sağlanamamış olsa da, sistem çoğu senaryoda yeterli doğrulukla çalışmaktadır.

Güvenli ve Modüler RESTful API Geliştirme

Uygulamanın hem mobil arayüzünü hem de OCR modülünü destekleyen, temiz mimari prensiplerine uygun olarak geliştirilmiş RESTful bir API sistemi oluşturulmuştur. API, kimlik doğrulama mekanizmaları ile güvenli hâle getirilmiş ve katmanlar arası görev ayrımı sağlanmıştır.

LLM Tabanlı Tarif Sohbet Modülü

Tespit edilen malzemelere ve kullanıcıdan alınan girdilere dayalı olarak tarif önerileri sunan, düşük halüsinasyon oranına sahip ve etkileşimli bir büyük dil modeli (LLM) uygulamaya entegre edilmiştir. Bu modül, kullanıcılara daha kişiselleştirilmiş ve akıllı öneriler sunma kapasitesine sahiptir.

Tam Ölçekli Android Uygulaması Geliştirme

Modern Android geliştirme bileşenleri olan Jetpack Compose, MVVM mimarisi ve Dagger Hilt kullanılarak, kullanıcıların tarifleri görüntüleyebildiği, buzdolabı içeriklerini yönetebildiği ve sistemle gerçek zamanlı etkileşim kurabildiği, işlevsel olarak tamamlanmış bir mobil uygulama geliştirilmiştir.

7.2 Sınırlılıklar ve Edinilen Dersler

Bu çalışmada geliştirilen sistemin uygulanması sürecinde çeşitli sınırlılıklar ve bu sınırlılıklardan çıkarılan önemli dersler tespit edilmiştir. Öncelikle, toplanan market fişlerinin büyük bir kısmının belirli zincir marketlere ait olması nedeniyle, OCR modelinin eğitildiği veri seti yalnızca sınırlı sayıda fiş formatını temsil etmiş ve bu durum, farklı şablonlara sahip fişlerde doğruluk oranının düşmesine yol açmıştır. Bu bağlamda, OCR modülünün genellenebilirliğini artırmak amacıyla, daha çeşitli kaynaklardan, farklı baskı kalitesi, yazı tipi ve çözünürlük koşullarında fiş verileri toplanarak eğitim setinin zenginleştirilmesi gerektiği anlaşılmıştır. Ayrıca, LLM tabanlı içerik üretiminde bazı durumlarda yanıt süresinin gecikmeli olması, sistemin gerçek zamanlı uygulamalardaki performansını sınırlamıştır. Bu nedenle, benzer sistemlerde API çağrılarının optimize edilmesi, gerektiğinde önbellekleme (caching) ve önyüklemeli içerik sunma gibi stratejilerin entegrasyonu önemli bir gereklilik olarak ortaya çıkmıştır. Bununla birlikte, LLM çıktılarının doğruluğunun büyük ölçüde kullanılan prompt'ların açık, yönlendirici ve bağlama uygun şekilde tasarlanmasına bağlı olduğu gözlemlenmiştir. Sistemde yer almayan malzemelerin tariflere dahil edilmesi gibi durumlar, LLM ile çalışırken gelişmiş prompt mühendisliği uygulamalarının gerekliliğini ortaya koymuştur. Mobil uygulama tarafında ise, Android arayüzünün henüz tamamlanmamış olması nedeniyle kullanıcı etkileşimleri sadece terminal ve web tabanlı prototip ile sınırlı kalmış, bu da gerçek kullanıcı deneyiminin bütüncül biçimde test edilmesini engellemiştir. Bu noktada, mobil uygulamanın erken geliştirme sürecine dahil edilmesi ve kullanıcı testleriyle desteklenmesi gerektiği sonucu çıkarılmıştır. Ayrıca sistemin hâlihazırda KVKK gibi veri koruma regülasyonlarına tam olarak uyumlu olmaması, geniş çaplı kullanıcı yayılımı açısından potansiyel bir risk teşkil etmektedir. Bu nedenle, kullanıcı profili gibi hassas verilerle çalışırken gizlilik ilkelerine bağlı kalınmalı ve sistemin yasal mevzuatlara uyumlu hâle getirilmesi sağlanmalıdır. Tüm bu gözlemlerin yanı sıra, OpenAI API kullanımında karşılaşılan kota sınırlamaları, model seçiminde yalnızca teknik yeterlilik değil, aynı zamanda erişim esnekliği, kota politikaları ve maliyet planlamalarının da dikkate alınması gerektiğini göstermiştir. Son olarak, sistemin OCR, LLM, veritabanı ve arayüz gibi bileşenlerinin entegre biçimde çalışabilmesi, yazılımın sürdürülebilir ve bakım kolaylığı sağlayan modüler bir mimariyle geliştirilmesinin önemini ortaya koymuştur.

OCR Performansında Ortam Uyumsuzlukları

Proje süresince karşılaşılan en temel teknik sorunlardan biri, Tesseract OCR sisteminin farklı çalışma ortamlarında sergilediği tutarsız performans olmuştur. Özellikle Google Colab gibi bulut tabanlı ortamlarda, minimum ön işleme ile yüksek doğruluk oranları elde edilirken; yerel makinelerde aynı performans yakalanamamıştır. Detaylı analiz sonucunda, Colab ortamında kullanılan Tesseract sürümünün veya yapılandırmasının standart versiyonlardan farklı olduğu ve muhtemelen özel olarak eğitilmiş ya da optimize edilmiş olduğu anlaşılmıştır. Bu deneyim, makine öğrenimi projelerinde ortam bağımsızlığı, sürüm kontrolü ve yeniden üretilebilirlik konularının kritik öneme sahip olduğunu açık biçimde ortaya koymuştur.

OCR Doğruluğu ve Türkçe Dil Desteği Yetersizliği

Tesseract OCR'nin mevcut sürümü, Türkçe karakterler (ç, ı, ğ, ö, ş, ü) ve özellikle gürültülü arka planlara sahip fişlerde yeterli performans gösterememektedir. Bu durum, fişlerden ürün ismi ve fiyat gibi temel bilgilerin çıkarılmasında ciddi doğruluk kayıplarına yol açmış ve sistemin gerçek dünyadaki kullanılabilirliğini kısıtlamıştır. Bu, projenin hâlihazırdaki en belirgin teknik sınırlılığı olarak değerlendirilmektedir.

Sistemsel Entegrasyon ve Karmaşıklık

Projedeki farklı bileşenlerin (YOLOv8, OCR, LLM, mobil uygulama vs.) birbiriyle entegre edilmesi beklenenden daha karmaşık ve zaman alıcı olmuştur. Özellikle bileşenler arası veri akışının ve hata yönetiminin sağlıklı şekilde gerçekleştirilmesi için yoğun hata ayıklama süreçleri gerekmiştir. Bu deneyim, ileriye dönük projelerde daha modüler bir yazılım mimarisi tercih edilmesi ve tüm sistem bileşenlerinin erken aşamalarda birlikte test edilmesi gerektiğini açık biçimde göstermiştir.

7.3 Geleceğe Yönelik Geliştirme Olanakları

Bu çalışmada geliştirilen sistemin hem teknik hem de işlevsel açıdan daha ileri seviyelere taşınabilmesi için çeşitli geliştirme alanları belirlenmiştir. Öncelikli hedeflerden biri, mobil ve web tabanlı kullanıcı arayüzünün tamamlanmasıdır. Özellikle Android tabanlı, kullanıcı dostu, sezgisel ve işlevsel bir mobil uygulamanın geliştirilerek yayına alınması planlanmaktadır. Bu sayede sistemin daha geniş kullanıcı kitlelerine ulaşması ve günlük yaşama entegrasyonu mümkün olacaktır. Bir diğer önemli geliştirme alanı, prompt mühendisliği ve LLM çıktılarının doğruluğunun artırılmasıdır. Bu doğrultuda, sistem dışı içeriklerin önerilmesini önlemek amacıyla özel olarak tasarlanmış prompt yapıları ve önceden eğitilmiş tarif üretim modelleri kullanılacaktır. LLM modülünün daha isabetli, bağlama uygun ve kullanıcıya özgü içerikler sunması, bu strateji ile hedeflenmektedir. Sistemin uzun vadeli kullanımında bulut tabanlı mikroservis mimarisine geçilmesi de planlanmaktadır. Bu mimari yapı sayesinde sistem, daha kolay ölçeklenebilir hâle getirilecek; yük dengeleme, modüler geliştirme ve servis bağımsızlığı gibi avantajlar elde edilecektir. Bu yapı, aynı zamanda farklı işlevlerin paralel geliştirilmesine olanak tanıyacaktır.

Veri güvenliği açısından, sistem mimarisine şifreleme, anonimleştirme ve kimlik doğrulama mekanizmaları entegre edilerek, kullanıcı verilerinin güvenli bir şekilde bulut ortamında

saklanması sağlanacaktır. Bu sayede, özellikle kişisel veri barındıran sistem bileşenleri regülasyonlara daha uyumlu hâle getirilecektir.

Ayrıca, makine öğrenmesi tabanlı tercih öğrenimi ile zaman içinde kullanıcıların geçmiş seçimlerinden öğrenen, kendini geliştiren ve giderek daha kişiselleştirilmiş tarif önerileri sunabilen bir modül geliştirilecektir. Bu modül, sistemin etkileşim gücünü ve kullanıcıya sağladığı katma değeri artıracaktır.

Son olarak, sistemin yalnızca bireysel kullanıcılarla sınırlı kalmayıp, toplu yemek hizmeti sunan kurumlar, restoranlar ve sosyal yardım kuruluşları gibi daha geniş ölçekli aktörler tarafından da kullanılabilir hâle getirilmesi hedeflenmektedir. Bu genişleme sayesinde sistem, hem ticari hem de sosyal fayda üreten çok yönlü bir çözüm platformu olarak değerlendirilebilecektir. Bu tez çalışması, yapay zekâ teknolojilerinin gündelik yaşam problemlerine uygulanabilirliğini ortaya koymakta ve özellikle gıda israfını azaltmaya yönelik yenilikçi, uygulanabilir ve sürdürülebilir bir çözüm modeli sunmaktadır. Geliştirilmeye açık yapısı sayesinde sistem, hem bireysel hem de toplumsal düzeyde önemli katkılar sağlama potansiyeline sahiptir.

My Fridge sisteminin sahip olduğu mevcut altyapı, üzerine inşa edilebilecek pek çok yenilikçi işlev için sağlam bir temel sunmaktadır. Bu bölümde, gelecekte hayata geçirilmesi planlanan teknik ve kullanıcı odaklı iyileştirme alanları özetlenmiştir:

OCR Performansının Geliştirilmesi

Tesseract yerine PaddleOCR, EasyOCR gibi modern ve daha uyarlanabilir modellerin kullanılması ya da Türkçe karakterler içeren özel bir veri seti ile özelleştirilmiş bir Tesseract modeli eğitilmesi planlanmaktadır. Bu adım, özellikle gürültülü fişlerdeki ürün ve fiyat bilgilerinin daha güvenilir şekilde çıkarılması açısından kritik önem taşımaktadır.

Çok Dilli ve Bölgesel Kişiselleştirme

OCR sisteminin çok dilli hâle getirilmesiyle uygulamanın küresel ölçekte erişilebilirliği artırılacak ve tarifler sadece içeriklere değil, kültürel ve coğrafi tercihlere göre de kişiselleştirilebilecektir. Örneğin Fransız bir kullanıcıya Fransız mutfağı tarifleri sunulurken, Türk bir kullanıcı için geleneksel Türk yemekleri önerilebilecektir.

Harcama Takibi ve Analitiği

OCR modülü sayesinde toplanan ürün fiyatlarının analiz edilmesiyle bir kişisel harcama yönetimi modülü geliştirilebilir. Bu özellik, kullanıcıların aylık bazda gıda harcamalarını takip etmelerine, en çok satın alınan ürünleri görmelerine ve belirlenen bütçeyi aşmaları durumunda uyarı almalarına imkân tanıyacaktır.

Bulut Tabanlı Dağıtım ve Ölçekleme

Sistemin bazı bileşenlerinin (örneğin OCR çıkarımı, sohbet modülü) bulut altyapısına taşınması, performansı artıracak; ayrıca cihazlardaki donanım gereksinimlerini düşürerek ölçeklenebilirlik ve model güncellemelerinde esneklik sağlayacaktır.

Kullanıcı Geri Bildirimine Dayalı Öğrenme

OCR sonuçları veya tarif önerileri üzerine kullanıcıdan alınacak geri bildirimler sayesinde, sistemin doğruluğu zamanla artırılabilir. Bu geri bildirimler, denetimli ince ayar ya da pekiştirmeli öğrenme yaklaşımları ile modele entegre edilebilir.

Cihaz Üzerinde LLM Kullanımı

Veri gizliliğini artırmak amacıyla, büyük dil modellerinin cihaz üzerinde çalıştırılması ve dışa veri aktarımı gerektirmeden çalışması değerlendirilmektedir.

Alerjen ve Besin Değerine Duyarlı Tarifler

Kullanıcıların alerjen bilgileri ve özel beslenme gereksinimleri dikkate alınarak, tariflerin daha sağlıklı ve güvenli biçimde kişiselleştirilmesi hedeflenmektedir.

Sonuç olarak My Fridge, yalnızca güncel teknolojilerin başarıyla entegre edildiği bir uygulama değil, aynı zamanda daha akıllı, kişiye özel ve küresel ölçekte erişilebilir bir akıllı mutfak asistanı ekosisteminin temellerini oluşturan bir projedir. Ekip olarak bu yolculuğu sürdürme ve projeyi tam ölçekli bir yapay zekâ destekli ev asistanına dönüştürme konusundaki motivasyonumuz güçlü bir şekilde devam etmektedir.

KAYNAKÇA

- [1]Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). *Attention is All You Need*. NeurIPS.
- [2]Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805.
- [3]Chowdhery, A., Narang, S., Devlin, J., et al. (2022). *PaLM: Scaling Language Modeling with Pathways*. arXiv:2204.02311.
- [4]Brown, T. B., Mann, B., Ryder, N., et al. (2020). *Language Models are Few-Shot Learners*. arXiv:2005.14165.
- [5]OpenAI. (2023). *GPT-4 Technical Report*. <https://openai.com/research/gpt-4>
- [6]Ouyang, L., Wu, J., Jiang, X., et al. (2022). *Training Language Models to Follow Instructions with Human Feedback*. arXiv:2203.02155.
- [7]Wei, J., Wang, X., Schuurmans, D., et al. (2022). *Chain-of-Thought Prompting Elicits Reasoning in Large Language Models*. arXiv:2201.11903.
- [8]Singhal, K., Azizi, S., Tu, T., et al. (2023). *Large Language Models Encode Clinical Knowledge*. *Nature*, 620(7972), 172–180.
- [9]Ji, Z., Lee, N., Frieske, R., et al. (2023). *Survey of Hallucination in Natural Language Generation*. ACM Computing Surveys.
- [10]Manakul, P., Fan, A., Liu, Y. (2023). *SelfCheckGPT: Zero-Resource Hallucination Detection for Generative LLMs*. arXiv:2303.08896.

- [11]Carlini, N., Tramer, F., Wallace, E., et al. (2021). *Extracting Training Data from Large Language Models*. USENIX Security Symposium.
- [12]Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). *On the Dangers of Stochastic Parrots: Can Language Models Be Too Big?*. ACM FAccT.
- [13]Bai, Y., Kadavath, S., Kundu, S., et al. (2022). *Constitutional AI: Harmlessness from AI Feedback*. Anthropic.
- [14]Doshi-Velez, F., & Kim, B. (2017). *Towards A Rigorous Science of Interpretable Machine Learning*. arXiv:1702.08608.
- [15]Zhao, W., Zhou, Y., Lin, S., & Zhang, J. (2023). *A Comprehensive Survey of Large Language Models*. arXiv:2307.06435.
- [16]Pettersson, O. et al. (2020). *Multimodal product classification using OCR and visual features*. International Journal of Computer Vision.
- [17]Gong, X., Wang, Y., Liu, Z. (2021). *A Dual Deep Network for Expiration Date Detection under Low-Light Conditions*. IEEE Transactions on Image Processing.
- [18]Kim, H., Lee, J., & Choi, S. (2019). *Nutritional information extraction from infant food labels using OCR and network analysis*. Journal of Food Science and Technology.
- [19]Kaderabek, M. (2020). *Automated Data Extraction from Supermarket Receipts Using OCR*. Procedia Computer Science, 176, 1393–1399.
- [20]Nguyen, T., Tran, Q., & Le, M. (2022). *Personalized Recipe Generation Using GPT-3: Health-Aware Modifications*. AI for Health Conference.
- [21]Zhou, X., Chen, Y., & Wu, J. (2023). *Evaluating ChatGPT for Personalized Meal Planning and Dietary Guidance*. Computational Nutrition Journal, 8(2), 45–61.
- [22]Clark, R., & Thompson, J. (2021). *Smart Waste Management in Commercial Kitchens and Its Implication for Households*. WasteTech Proceedings.
- [23]Mastorakis, G., & Karatsioras, A. (2022). *FoodSaveShare: A Mobile App to Combat Food Waste Using Expiration Notifications*. Sustainable Computing and Informatics Journal.
- [24]Tiwari, A., Sharma, R., & Zhang, L. (2023). *End-to-End Food Waste Minimization Using OCR, LLM and Predictive AI Systems*. Proceedings of the Smart Environments Conference.
- [25]Chen, L., & Zhao, F. (2021). *Cloud Computing in AI-Based Systems: The Role of AWS in Real-Time Data Management*. CloudTech Journal.

EKLER

Ek A – LLM Prompt Örnekleri

Örnek Prompt 1: Mod Tabanlı Tarif Önerisi

prompt = f"Elimdeki malzemeler: {[u['isim'] for u in self.ingredients]}. Kullanıcının ruh hali: {mood}. Bu ruh haline uygun geleneksel Türk mutfağından 3 tarif öner. Sadece tarif adlarını ver. Lütfen tarifler modlara uygun olsun."

Bu prompt, kullanıcının güncel ruh hâlini dikkate alarak, elindeki malzemelerle uyumlu ve psikolojik ihtiyaçlara uygun tarif önerileri sunmayı hedeflemektedir. Doğal dil modeli, bu girdiler doğrultusunda sadece tarif adlarını döndürmekte ve tariflerin ruh hâline uygunluğunu önceliklendirmektedir.

Örnek Prompt 2: Malzeme Tabanlı Tarif Üretimi

Elimdeki malzemeler: Dana Kıyma, Ekmek, Hazır Çorba, Lahana, Domates.

Bu malzemelerle yapılabilecek 3 geleneksel Türk yemeği öner.

Yalnızca yemek isimlerini yaz ve her birinin yanına kısa bir açıklama ekle.

Bu prompt yapısı, kullanıcıların mevcut malzemelerine dayalı olarak kısa açıklamalar içeren üç farklı yemek tarifi adı önermek amacıyla kullanılmaktadır. Özellikle israfı önleyici ve kullanıcıya zaman kazandırıcı bir yapı oluşturmak hedeflenmiştir.

Örnek Prompt 3: Besin Değeri Hesaplama

Lahana Sarması yemeğinin 1 porsiyonunun besin değerlerini yaz:

- Kalori
- Protein
- Karbonhidrat
- Yağ
- Lif

Bu prompt, seçilen bir yemek tarifine ait ortalama besin değerlerini sade ve anlaşılır bir şekilde istemek için tasarlanmıştır. Bu çıktılar, sistemin kalori karşılaştırma modülü ile entegre edilerek, kullanıcının günlük enerji ihtiyacını karşılayıp karşılamadığını analiz etmesine olanak sağlamaktadır.

Örnek Prompt 4: Tarif Detayları ve Hazırlık Aşamaları

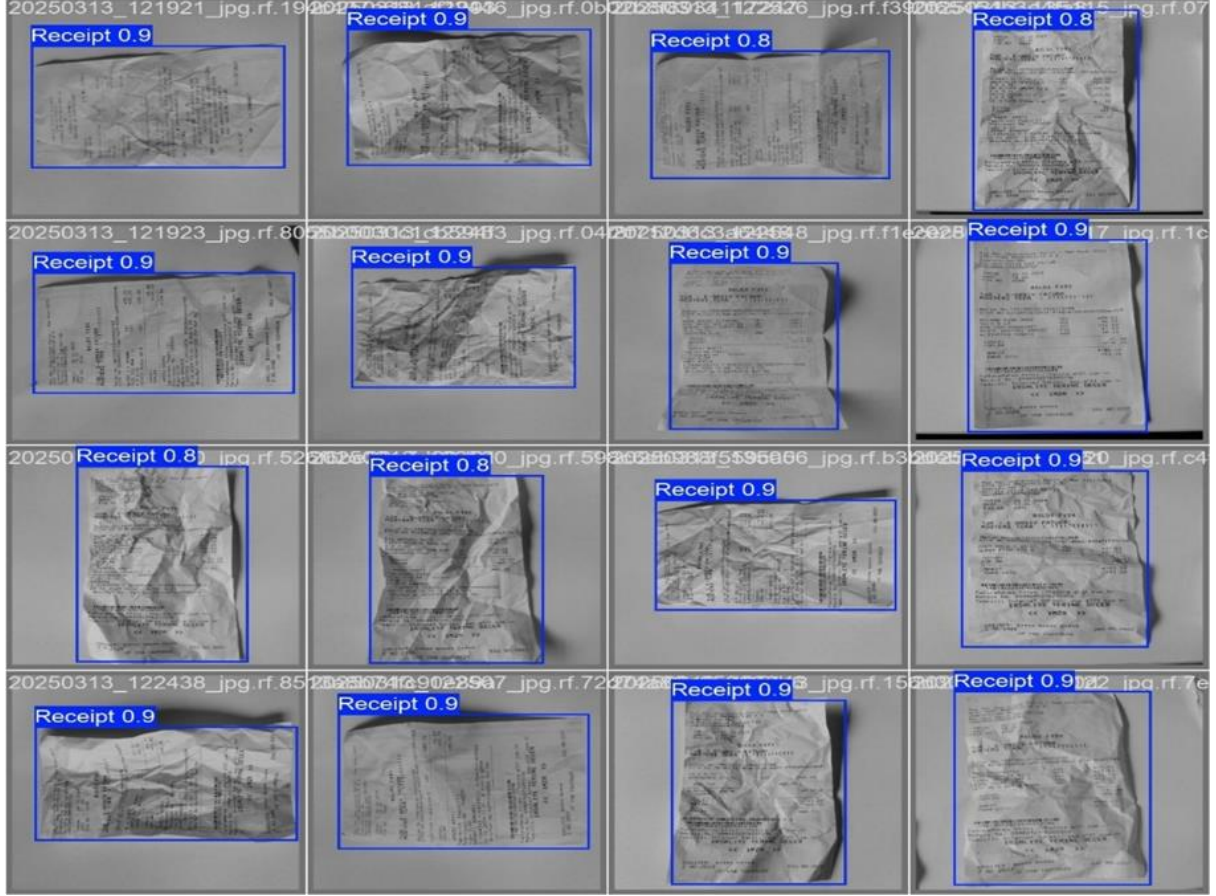
Seçilen yemek: Lahana Sarması.

Gereken malzemeler, hazırlık aşamaları ve pişirme süresi nedir?

Adım adım, sade bir dille açıkla.

Bu yapı, kullanıcının seçmiş olduğu bir yemeğin tüm hazırlanış sürecini adım adım öğrenmesini sağlamaya yöneliktir. LLM'den alınan çıktılar; tarifin içeriğini, hazırlık süresini ve pişirme aşamalarını net, anlaşılır ve sıralı biçimde sunmaktadır.

Ek B – YOLOv8 Örnek test çıktıları



Şekil 26. YOLOv8 gruplandırılmış test çıktılarının şekli



Şekil 27.YoloV8 çeşitlendirilmiş fiş ve diğer nesnelerin çıktıların şekli