MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL UYGULAMA İLE DERİN ÖĞRENME TABANLI NESNE TESPİTİ VE BÜYÜK DİL MODELİ İLE İFADE ÜRETME

NURCİHAN DERE

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği
Anabilim Dalı
Bilgisayar Mühendisliği Programı

DANIŞMAN
Doç. Dr. Kazım YILDIZ

EŞ-DANIŞMAN
Doç. Dr. Önder DEMİR

İSTANBUL, 2025

MARMARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

MOBİL UYGULAMA İLE DERİN ÖĞRENME TABANLI NESNE TESPİTİ VE BÜYÜK DİL MODELİ İLE İFADE ÜRETME

NURCİHAN DERE

(523622607)

YÜKSEK LİSANS TEZİ
Bilgisayar Mühendisliği
Anabilim Dalı
Bilgisayar Mühendisliği Programı

DANIŞMAN
Doç. Dr. Kazım YILDIZ

EŞ-DANIŞMAN
Doç. Dr. Önder DEMİR

İSTANBUL, 2025

ÖNSÖZ

“Mobil Uygulama ile Derin Öğrenme Tabanlı Nesne Tespiti ve Büyük Dil Modeli İle İfade Üretme” adlı bu çalışma Marmara Üniversitesi, Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Anabilim Dalında Yüksek Lisans Tezi olarak hazırlanmıştır.

Tez çalışmamda bana rehberlik eden, değerli bilgi ve deneyimlerini paylaşarak çalışmama katkı sağlayan danışmanım Doç. Dr. Kazım YILDIZ’a, eşdanışmanım Doç. Dr. Önder DEMİR’e teşekkürlerimi sunarım. Bu süreçte sabrı, anlayışı ve desteğiyle her zaman yanımda olan eşime ve aileme minnettarım.

HAZİRAN, 2025 **Nurcihan DERE**

İÇİNDEKİLER

ÖZET

MOBİL UYGULAMA İLE DERİN ÖĞRENME TABANLI NESNE TESPİTİ VE BÜYÜK DİL MODELİ İLE İFADE ÜRETME

Günümüzde yapay zekâ alanındaki gelişmeler, nesne tanıma teknolojilerinde önemli ilerlemeler kaydetmiştir. Bilgisayarla görme, bilgisayar sistemlerine görsel verileri insanların yaptığına benzer şekilde anlama ve yorumlama yeteneği kazandırmayı amaçlayan bir yapay zekâ alandır. Nesne tanıma ise bilgisayarların nesneleri algılayarak tanıma yeteneğini ifade eder. Bu yetenek, güvenlik sistemlerinde çevresel tehditleri tespit etmek, otomobil teknolojilerinde sürücülere kolaylık sağlamak, sağlık alanında etkili teşhisler koymak gibi çeşitli amaçlarla kullanılabilmektedir. Yapay zekâ alanında insan yaşamını daha kolay, güvenli ve verimli hale getirmek için pek çok çalışma yürütülmektedir. Bu çalışmalar nesne tanıma modellerinin hızlarını artırmayı, performanslarını iyileştirmeyi ve geniş kapsamlı uygulamalara entegrasyonunu sağlamayı hedeflemektedir. Özellikle gerçek zamanlı uygulamalarda bu teknolojiler hayatı kolaylaştırarak etkili çözümler sunmaktadır.

Bu çalışma kapsamında gerçek zamanlı olarak YOLO-v11 ile nesne tespiti, mesafe ölçümü ve nesne konumları ile ilgili açıklamaların elde edilebileceği bir mobil uygulama geliştirilmiştir. Bu uygulama, yapay zekâ modellerinin günlük hayatta kullanılmasıyla beraber anlık ihtiyaçların karşılanmasına yönelik pratik bir çözüm sunmaktadır. Araştırmanın temel amaçları arasında nesne konumlarının büyük dil modeli olan GPT-4o'da analiz edilmesi ve LiDAR teknolojisinin bu yapıya entegrasyonu da yer almaktadır. Bu çalışmada derin öğrenme modelinin F1 puanı 0.77, ortalama doğruluk değeri ise 0.806 olarak hesaplanmıştır. İnce ayar işlemi gerçekleştirilen GPT-4o, görsellerdeki nesne konumlarını doğru ve tutarlı bir şekilde belirleyerek doğal dilde açıklamalar üretmiştir. Modelin performansı, ROUGE-1 skoru 0.75, ROUGE-2 skoru 0.61 ve ROUGE-L skoru 0.71 olarak ölçülmüştür.

Bu tez kapsamındaki çalışmalar, derin öğrenme ve doğal dil işleme modellerini birleştirerek mobil uygulamalarda yapay zekanın etkinliğini artırmayı ve gelecekteki araştırmalar için yol gösterici bir kaynak olmayı amaçlamaktadır.

ABSTRACT

DEEP LEARNING BASED OBJECT DETECTION WITH MOBILE APPLICATION AND EXPRESSION GENERATION WITH LARGE LANGUAGE MODEL

Significant progress has been made in object recognition technologies thanks to recent developments in the area of artificial intelligence. Computer vision is an area of artificial intelligence that seeks to enable computer systems to understand and interpret visual data in a manner similar to that of humans. Object recognition refers to the ability of computers to recognize objects by perceiving them. This ability has many applications, including the detection of environmental threats in security systems, the facilitation of driving through technology, and the support of effective diagnoses in healthcare. Many studies are being performed in the field of artificial intelligence to make human life easier, safer and more efficient. These studies aim to increase the speed of object recognition models, improve their performance, and integrate them into wide-ranging applications. Especially in real-time applications, these technologies offer effective solutions by making life easier.

Within the scope of this study, a mobile application was developed with YOLO-v11 in which object detection, distance measurement and explanations about object locations can be obtained in real time. This application offers a practical solution for meeting instant needs with the use of artificial intelligence models in daily life. The main objectives of the research include analyzing object locations in the large language model GPT-4o and integrating LiDAR technology into this structure. In this study, the F1 score of the deep learning model was calculated as 0.77 and the average accuracy value was 0.806. GPT-4o, which was fine-tuned, produced natural language explanations by determining object locations in images accurately and consistently. The performance of the model was measured as ROUGE-1 score 0.75, ROUGE-2 score 0.61 and ROUGE-L score 0.71.

The studies within the scope of this thesis aim to increase the effectiveness of artificial intelligence in mobile applications by combining deep learning and natural language processing models, and to serve as a source of guidance for future research.

SEMBOLLER

AP_i : Her bir sınıf için hesaplanan ortalama hassasiyet

b_j : Bias (önyargı) terimi

c : Işık hızı

d : Nokta ile sensör arasındaki mesafe

d_i : Gizli boyutun büyüklüğü

$\sqrt{d_i}$: Ölçeklendirme faktörü

$f(z_i)$: Aktivasyon fonksiyonu

KB : Kilobayt

m : Metre

n : Giriş değişkenlerinin sayısı

s : Saniye

t : Uçuş süresi

V : Değer

w_i : Ağırlık katsayısı

x_i : Giriş değişkenleri

y : Çıktı değeri

Z_i : Ara katmandaki bir nöronun toplam girdisi

* : Çarpma sembolü

$\sum$: Toplam sembolü

$\cap$: Kesişim sembolü

$\cup$: Birleşim sembolü

KISALTMALAR

AP : Average Precision
AR : Augmented Reality
CNN : Convolutional Neural Networks
DP : Doğru Pozitif
DN : Doğru Negatif
E-ELAN : Extended Efficient Layer Aggregation Network
FP : False Positive
FN : False Negative
GELAN : Generalized Efficient Layer Aggregation Network
GRU : Gated Recurrent Unit
HRC : Human-Robot Collaborative
IR : Infrarred
LLM : Large Language Model
LiDAR : Light Detection and Ranging
mAP : Mean Average Precision
MS COCO : Microsoft Common Objects in Context
NMS : Non-Maximum Suppression
NLP : Natural Language Processing
PCA : Principal Component Analysis
PGI : Programmable Gradient Information
ReLU : Rectified Linear Unit
RL : Reinforcement Learning
RLHF : Reinforcement Learning from Human Feedback
RNN : Recurrent Neural Networks
TP : True Positive

TN **:** True Negative

t-SNE **:** T-distributed Stochastic Neighbor Embedding

YN **:** Yanlış Negatif

YP **:** Yanlış Pozitif

YOLO **:** You Only Look Once

YSA **:** Yapay Sinir Ağları

ŞEKİL LİSTESİ

SAYFA

.

TABLO LİSTESİ

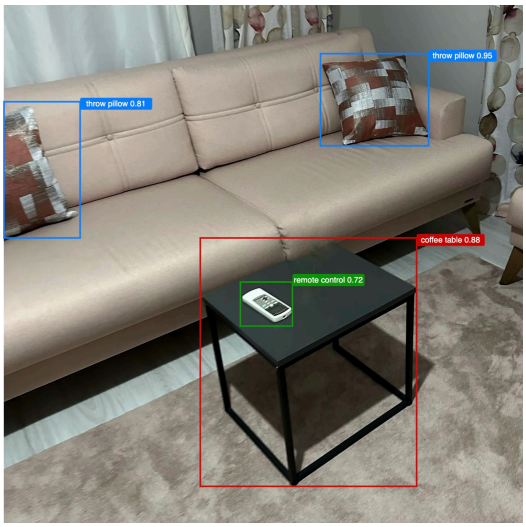
SAYFA

.

1. GİRİŞ

Yapay zekâ günümüzde teknolojinin yönünü belirleyen en önemli alanlardan biri haline gelmiştir. Görüntü işleme, doğal dil işleme, otonom sistemler ve sağlık gibi çeşitli alanlardaki uygulamalarla yapay zekâ, insan yaşamını kolaylaştıran birçok çözüm ve uygulama sunmaktadır. Bilgisayarla görme, bilgisayarların çevresel verileri tanımasını ve anlamlandırmasını sağlayan yapay zekâ alanlarından biridir. Bilgisayarlara derin öğrenme ve makine öğrenmesi gibi yöntemler kullanılarak çevresel bilgileri anlamak ve yorumlamak için insan benzeri yetenekler kazandırılır [1]. Bu alandaki gelişmeler günümüzde otonom araçlar, sağlık sistemleri, nesne tanıma, yüz tanıma teknolojileri, endüstriyel otomasyon ve üretimde kalite kontrolü gibi geniş bir yelpazede uygulama alanı bulmaktadır [2-6].

Görüntü işleme alanındaki önemli gelişmelerden biri nesne tanıma teknolojilerinin ilerlemesidir. Nesne tanıma, bir görüntüdeki farklı nesneleri algılayarak sınıflandırmayı ve konumlarını tespit etmeyi amaçlar. Derin öğrenme yöntemlerin kullanımıyla büyük ilerlemeler kaydedilen bu alanda YOLO (You Only Look Once) yaygın olarak tercih edilen algoritmalardan biridir [7]. YOLO, görüntüleri tek bir işlemde analiz ederek nesnelerin sınıflandırılmasını ve nesnelerin görüntüdeki yerlerinin belirlenmesini sağlar. Şekil 1.1'de gösterildiği gibi tespit edilen nesnelerin etrafına sınırlayıcı kutular (bounding boxes) çizilir ve 0.95 gibi bir güven oranı belirtilir. Ayrıca yüksek doğruluk ve hız sunarak özellikle gerçek zamanlı uygulamalarda etkili sonuçların elde edilmesine imkân tanır.



Şekil 1.1. YOLO ile nesne tespiti.

YOLO, geniş bir uygulama yelpazesinde kullanılmakta olup sürekli geliştirilen yeni yöntemler ile daha verimli hale getirilmektedir. YOLO gibi derin öğrenme tabanlı modeller, CoreML, TensorFlow Lite gibi çerçeveler aracılığıyla mobil cihazlarda çalıştırılabilir. Başta nesne tanıma olmak üzere tıbbi görüntüleme verilerinden hastalık tespiti, güvenlik sistemlerinde tehdit algılama, otonom araçlarda çevresel nesneleri belirleme gibi amaçlarla kullanılabilmektedir [8-10]. YOLO'nun hızlı işlem yeteneği özellikle mobil platformlarda gerçek zamanlı nesne tespit uygulamalarının geliştirilmesini mümkün kılmaktadır.

LiDAR (Light Detection and Ranging) teknolojisi, uzaktan algılama uygulamalarında mesafe ölçümü ve çevresel haritalama amacıyla yaygın olarak kullanılan gelişmiş bir teknolojidir. LiDAR, lazer ışını kullanarak nesnelerden geri yansıyan ışığı ölçer ve bu veriler aracılığıyla çevrenin üç boyutlu (3D) modelini oluşturur [11]. LiDAR teknolojisi özellikle otonom araçlar, robotik sistemler ve artırılmış gerçeklik (AR) uygulamalarında yaygın olarak kullanılmaktadır [11-13]. LiDAR teknolojisi ile mesafe ölçümü nesnelerin uzaklıklarının hassas bir şekilde elde edilmesini sağlar. Nesne tanıma uygulamalarıyla entegre edildiğinde LiDAR verileri, nesnelerin konumlarını doğru bir şekilde ifade ederken, nesne tanıma algoritmaları bu verileri analiz ederek etkileşimli ve güvenli sistemler oluşturulabilir.

Doğal Dil İşleme (NLP), bilgisayar sistemlerine insan dilini anlama, yorumlama ve etkileşim kurma yeteneği kazandırmayı amaçlayan bir Yapay Zekâ (AI) alanıdır. NLP, insanların kullandığı doğal dili, yazılı ya da sözlü şekilde bilgisayarların işleyebileceği hale getiren teknikler ve yöntemler bütünü olarak tanımlanabilir. Metin ve konuşma verilerini analiz ederek dilin karmaşık yapısını çözümler ve bilgisayar sistemlerine dili anlama ve üretme yeteneği kazandırır. Bu teknoloji, veri madenciliği, metin analizi, konuşma tanıma, dil çevirisi ve duygu analizi gibi birçok farklı alanda aktif olarak kullanılır [14-18]. Doğal dil işleme, makine öğrenmesi ve yapay zekâ ile geliştirilerek özellikle büyük veri kümelerindeki kalıpları ve anlamları keşfetmede önemli bir rol oynar [19]. Müşteri hizmetleri için geliştirilen sohbet robotu, sağlık sektöründe kullanılan klinik notların analizi, büyük metinlerin özetlenmesi gibi birçok uygulama ve kullanım alanları mevcuttur [20-22]. Metin özetleme yöntemleri, büyük miktardaki veriden en önemli kısımları ayıklamak için Derin Öğrenme ve Makine Öğrenmesi tekniklerini kullanır. Doğal dil işlemenin bu alandaki etkisi, büyük metin verilerini hızlıca tarayarak önemli bilgileri çıkarabilme yeteneğinde kendini gösterir. Bu süreçte, anahtar kelimeler veya belirli ifadeler tespit edilerek metinlerdeki önemli bilgiler öne çıkarılır [23]. Özetleme aşamasında, metindeki vurgulanan kelimelerin tespiti ve konu modelleme yöntemleri ile sınıflandırmak, belge kümeleme ve özetleme süreçlerinde önemli rol oynar [24].

Doğal Dil İşleme teknolojilerinin gelişimiyle birlikte dil modelleri de önemli bir ilerleme kaydetmiştir. Bu alandaki dikkat çekici gelişmelerden biri de büyük dil modellerinin ortaya çıkışıdır. Büyük dil modeli (LLM)'ler, metin üretme, metin sınıflandırma, soru-cevap sistemleri, çeviri, özetleme, duygu analizi gibi çeşitli dil tabanlı görevlerde kullanılabilir [25-27]. Ayrıca konuşma ve sesli yanıt sistemlerinde de kullanılabilen LLM'ler, yazılım geliştirme süreçlerinde kod üretme ve hata ayıklama gibi görevlerde de yardımcı olmaktadır [28]. Bu çok yönlü modeller, yapay zekâ tabanlı çözümlerle birleştirildiğinde daha akıllı ve etkili sistemlerin geliştirilmesine katkıda bulunur [29, 30].

1.1.1. Problemin tanımı

Mobil cihazlar üzerinde nesne tanıma ve mesafe ölçümü uygulamaları kullanıcıların günlük hayatlarını kolaylaştırmak ve çeşitli sektörlerde verimliliği artırmak için büyük bir potansiyele sahiptir. Günlük hayatta nesne tanıma ve nesnelerin uzaklıklarını ölçme, aynı zamanda bu nesnelerin diğer nesnelerle olan konum ilişkilerini belirleme kullanıcıların hayatını önemli ölçüde kolaylaştıracaktır. Bu nedenle bu ihtiyaçları karşılamak için gerçek zamanlı algılama ve mesafe ölçme işlemlerini etkin bir şekilde gerçekleştirecek sistemler geliştirilmelidir. Nesne konumlarının doğru bir şekilde tespit edilmesi ve diğer nesnelerle olan ilişkilerinin belirlenmesi için yapay zekâ tabanlı yaklaşımların uygulanması önemlidir. Bu noktada LLM teknolojileri ile nesne tanıma ve mesafe ölçme sistemleri entegre edilerek etkili çözümler sunulmalıdır.

1.1.2. Amaç ve hedefler

Bu çalışma ile mobil cihazlar üzerinde gerçek zamanlı nesne tespiti yapan yüksek başarı oranına sahip bir derin öğrenme modeli geliştirilmesi ve bu model ile LLM'in birlikte kullanıldığı mobil uygulamanın tasarlanması ve geliştirilmesi ve amaçlanmıştır. Bu mobil uygulama kullanıcıların günlük hayatlarını kolaylaştırmayı hedeflemektedir. Çalışma kapsamında geliştirilen sistem, nesnelerin doğru bir şekilde tespit edilmesini, nesnelerin uzaklıklarının belirlenmesini ve bu nesnelerin birbirleriyle olan konum ilişkilerinin elde edilmesini içerir. Uygulamanın doğruluğunu ve etkinliğini artırmak amacıyla LLM kullanılarak nesne konumlarının analiz edilmesi sağlanmaktadır. Yüksek doğruluk ve hız gerektiren gerçek zamanlı algılama işlemlerinin etkin bir şekilde gerçekleştirilmesi amaçlanmaktadır. Bu çalışma ile mobil cihazlarda nesne tanıma ve mesafe ölçümü konusunda yeni bir yaklaşım sunulması ve bu alanda yapılacak gelecek çalışmalar için kaynak oluşturulması beklenmektedir.

1.1.3. Ana katkılar

Bu çalışma kapsamında mobil cihazlar üzerinde gerçek zamanlı nesne tanıma ve mesafe ölçümü yapan bir uygulama geliştirilmiştir. Uygulamada kullanılan derin öğrenme modeli, Python diliyle geliştirilmiş olup modelin eğitilmesinde kullanılan veri setinin hazırlanması, bu çalışmanın yenilikçi yönlerinden biri olarak sunulmuştur. Geliştirilen derin öğrenme modeli, yüksek başarı oranı ile nesne tespitini gerçekleştirmiştir. Ayrıca uygulamada LiDAR teknolojisi kullanılarak nesnelerin uzaklık bilgisi elde edilmiştir. Büyük dil modeli ile desteklenen uygulama, nesnelerin konum ilişkilerini ve uzaklıklarını doğru bir şekilde tespit etmiştir. Derin öğrenme ve doğal dil işleme modelinin bir arada kullanımıyla nesne tanıma ve açıklama süreçleri daha hızlı ve verimli gerçekleştirilmiştir. Geliştirilen mobil uygulama düşük işlem gücü gereksinimleriyle etkili bir şekilde çalışarak kullanıcılara hızlı ve doğru sonuçlar sunmaktadır. Kullanılan teknolojiler ve yöntemler yapay zekanın günlük hayatta etkili bir şekilde kullanılabilmesine katkı sağlamaktadır.

1.1.4. Tez organizasyonu

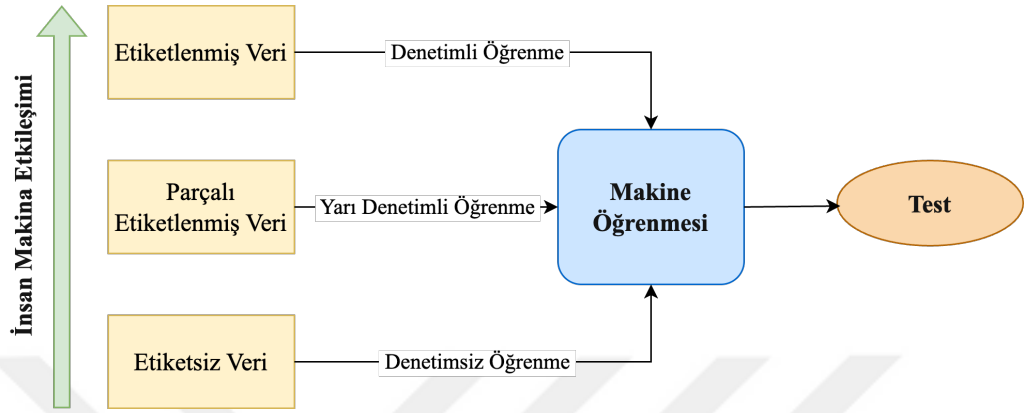
Tez dört bölümden oluşmaktadır. “Giriş” bölümünde, derin öğrenme yöntemleri ve büyük dil modelleri ele alınmış, literatürdeki ilgili çalışmalar incelenmiştir. “Materyal ve Yöntem” başlıklı ikinci bölümde, çalışmadaki veri seti, geliştirilen derin öğrenme modeli, LiDAR, mobil uygulama ve LLM tabanlı yöntem detaylı olarak açıklanmıştır. Üçüncü bölüm olan “Bulgular ve Tartışma” bölümünde, deney ortamı, çalışmanın performansı, elde edilen sonuçlar ve bulgular tartışılmıştır. Son bölüm olan “Sonuçlar” kısmında ise çalışma özetlenmiş, elde edilen bulgular ve literatüre sağlanan katkılar sunulmuş, gelecekte gerçekleştirilebilecek çalışmalar hakkında önerilerde bulunulmuştur.

1.2. Makine Öğrenmesi

Makine öğrenmesi, bilgisayar sistemlerinin veri kullanarak kendilerini geliştirmesini sağlayan bir yapay zekâ alanıdır. Makine öğrenimi algoritması, girdi verilerini kullanarak belirli bir görevi başarmak için kendini otomatik olarak uyarlayan ve deneyim yoluyla öğrenen bir hesaplama sürecidir [31]. Makine öğrenmesinde eğitim ve test olmak üzere iki aşama bulunur. Eğitim aşamasında algoritmalar veriler üzerinde çalıştırılır ve bu süreçte modelin oluşturulmasını sağlayan matematiksel yapı veya fonksiyon geliştirilir. Test aşamasında ise bu model daha önce eğitim için kullanılmayan veriler üzerinden değerlendirilir ve performansı ölçülür. Oluşan matematiksel yapı veya fonksiyon, 'model' olarak adlandırılır.

Şekil 1.2’de makine öğrenimi algoritmalarının eğitim verisi türlerine göre kategorileri

gösterilmektedir. Bunlar, denetimli öğrenme, denetimsiz öğrenme, yarı denetimli öğrenme ve pekiştirmeli öğrenme gibi farklı yaklaşımlar altında sınıflandırılır.



Şekil 1.2. Eğitim verisine göre makine öğrenimi algoritmalarının kategorileri [31].

1.2.1. Denetimli öğrenme (Supervised learning)

Denetimli öğrenme, makine öğrenmesi alanında en yaygın kullanılan yaklaşımlardan biridir. Bu yöntem, bir modelin belirli bir görevi öğrenebilmesi amacıyla önceden etiketlenmiş veri setleri kullanılarak eğitilmesi üzerine kuruludur. Bu yöntemde, her bir girdi verisinin doğru etiketi bilinir ve model, bu etiketli verilerden yola çıkarak beklenen çıktıyı üretmek için gerekli kuralları öğrenir. Denetimli öğrenmeye örnek olarak nesne tanıma verilebilir. Bu yaklaşımda nesne tanıma, etiketlenmiş görüntü verileri kullanılarak gerçekleştirilir. Örneğin, bir modelin "çiçek", "kuş", "araba" gibi farklı nesneleri tanıması isteniyorsa, bu nesneleri içeren ve her biri doğru etiketlerle (örneğin çiçek, kuş, araba) işaretlenmiş binlerce görüntü kullanılarak model eğitilir. Eğitim aşamasında, model, her bir görüntüyü analiz eder ve doğru etiketleriyle eşleştirebilmek için öğrenir. Günümüzde denetimli öğrenmenin uygulama alanlarından biri de sağlık alanıdır. Tıbbi görüntülerde belirli anormalliklerin otomatik olarak tespit edilmesi amacıyla da kullanılmakta ve daha iyi klinik kararlar almaya yardımcı olabilmektedir [32, 33].

1.2.2. Yarı denetimli öğrenme (Semi-supervised learning)

Bu yöntemde, genellikle küçük bir kısmı etiketlenmiş ve büyük bir kısmı etiketlenmemiş olan bir veri seti kullanılır. Model, etiketli verilerden öğrenerek etiketsiz veriler hakkında da tahmin yapmayı veya onları anlamlandırmayı öğrenir. Bu yaklaşım, genellikle etiketlenmiş verilerin sınırlı, etiketlenmemiş verilerin ise büyük miktarda olduğu

durumlarda kullanılır. Ayrıca etiketlemenin maliyetli veya zaman alıcı olduğu senaryolarda da avantajlıdır. Etiketlenmiş veri, modelin temel yapıyı öğrenmesini sağlarken etiketlenmemiş veri bu yapıyı daha geniş bir veri kümesinde genelleştirmesine yardımcı olur [34]. Yarı denetimli öğrenme, görüntü tanıma, doğal dil işleme ve tıbbi teşhis gibi alanlarda etkin bir şekilde kullanılmaktadır [35, 36].

1.2.3. Denetimsiz öğrenme (Unsupervised learning)

Denetimsiz öğrenmede modelin etiketlenmemiş verilerden anlamlı desenler veya yapılar öğrenmesi amaçlanır. Bu yaklaşım, özellikle büyük miktarda etiketlenmemiş verinin bulunduğu durumlarda avantajlıdır. Eğitim verileri üzerinde herhangi bir etiket veya doğru cevap bulunmaz, model veri setindeki benzerlikler, farklılıklar ve gizli yapılar gibi özellikleri keşfetmeye çalışır. Bu yöntemlerden biri olan kümeleme (Clustering), verileri benzer özelliklere sahip gruplara ayırarak anlamlı alt kümeler oluşturur. K-Means Kümeleme ve Hiyerarşik Kümeleme gibi algoritmalar, bu tür gruplama işlemlerini gerçekleştirmek için yaygın olarak kullanılır [37, 38]. Bir diğer denetimsiz öğrenme tekniği olan boyut indirgeme (Dimensionality Reduction), yüksek boyutlu verileri daha basit ve anlamlı bir şekilde temsil etmek için veri özelliklerinin sayısını azaltmayı amaçlar [39, 40]. Bu yöntem verilerin görselleştirilmesi ve fazla özelliklerin neden olduğu gürültünün azaltılması için kullanılır. Temel Bileşen Analizi (Principal Component Analysis - PCA**)** ve T-distributed Stochastic Neighbor Embedding (t-SNE) gibi algoritmalar bu amaçla yaygın olarak kullanılmaktadır [41].

1.2.4. Pekiştirmeli öğrenme (Reinforcement learning)

Pekiştirmeli öğrenme (RL) istenen davranışlar için ödüllendirmeye ve istenmeyen davranışlar için cezalandırmaya dayanır. Ajan (agent)**,** makine öğrenimi bağlamında belirli bir ortamda (environment) gözlem yapabilen, bu gözlemlerden öğrenen, eylemler gerçekleştiren ve bu eylemler sonucunda geri bildirim alan bir varlıktır. Pekiştirmeli öğrenmede ajanlar, belirli bir durumla karşılaştıklarında en yüksek ödülü ve en düşük cezayı elde etmek için en uygun eylemi seçmeye çalışırlar. Bu süreç olumlu ve olumsuz sonuçların bir geri bildirim işlevi gördüğü bir öğrenme yöntemi sunar. Pekiştirmeli öğrenmede etiketli veri kullanılmaz. Ajan gelecekte daha büyük ödüller alabilmek için deneme yanılma yoluyla öğrenir [42]. Bu ajanlar robotlar, yazılımlar, algoritmalar veya yapay zekâ sistemleri gibi çeşitli formlarda olabilir. Pekiştirmeli öğrenme, oyunlardan robotik ve finansal uygulamalara kadar geniş bir yelpazede kullanılarak çevresel etkileşimler ve ödül-ceza mekanizmaları aracılığıyla sürekli öğrenmeyi ve strateji geliştirmeyi sağlar [43, 44].

1.3. Derin Öğrenme

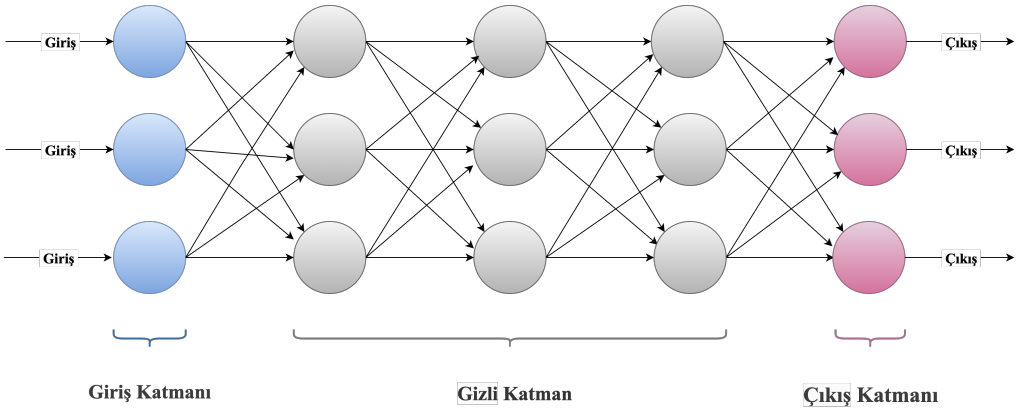
Derin Öğrenme (Deep Learning), yapay sinir ağları ve derin sinir ağları gibi çok katmanlı yapıları kullanan bir makine öğrenme yöntemidir. Derin Öğrenme yöntemleri karmaşık desenleri, gizli ilişkileri ve gösterimleri otomatik olarak çıkarmak için büyük ölçekli veri kümeleri üzerinde eğitilebilir. Derin öğrenme teknikleri, görüntü tanıma, konuşma tanıma, doğal dil işleme gibi çeşitli alanlarda başarıyla kullanılır. Derin öğrenme yöntemleri ayrıca bilgisayarla görme alanında, büyük miktardaki görsel veriler üzerinde çalışarak nesne tanıma, görüntü sınıflandırma ve segmentasyon gibi görevlerde oldukça yüksek başarı oranları elde eder. Günümüzde bilgisayarla görme uygulamaları, görüntü analizi için geleneksel istatistiksel yöntemlerden giderek uzaklaşarak daha doğru görüntü analizi sağlayan derin öğrenme yöntemlerini kullanmaktadır [45]. Bu yöntemler, görsel verilerden anlamlı özellikler çıkararak insan benzeri algılama ve yorumlama kabiliyetlerini güçlendirmektedir. Örneğin derin öğrenme tabanlı yöntemler kullanılarak yangın tespiti gibi önemli alanlarda çalışmalar gerçekleştirilmiş ve etkili sonuçlar elde edilmiştir [46-48].

1.3.1. Yapay sinir ağları

Yapay Sinir Ağları (YSA), insan beynindeki biyolojik sinir ağlarını ve insan beyninin çalışma şeklini taklit ederek verileri işleyen, öğrenen, karmaşık problemleri çözebilen bir yapıdır. YSA'lar, katmanlar halinde düzenlenmiş birbirine bağlı yapay nöronlardan, düğümlerden veya birimlerden oluşur. YSA'lar aşağıdaki gibi üç temel katmandan oluşur.

- Giriş Katmanı (Input Layer): Verilerin ağa ilk aktarıldığı katman.
- Gizli Katmanlar (Hidden Layers): Karmaşık hesaplamaların yapıldığı ara katmanlar.
- Çıktı Katmanı (Output Layer): Sonuçların alındığı katman.

Bilgi, verilerin ağa beslendiği giriş katmanından başlayarak ağ üzerinden akar ve giriş verilerine dayalı bir tahmin veya karar çıktı katmanında üretilir. Derin öğrenmedeki "derin" kelimesi, bir yapay sinir ağında bulunan katmanların sayısını ifade eder. Geleneksel yapay sinir ağlarında genellikle birkaç katman bulunurken, derin öğrenme modellerinde çok sayıda gizli katman bulunur.



Şekil 1.3. Derin sinir ağları [49].

Şekil 1.3'te derin yapay sinir ağının katmanları gösterilmektedir. Giriş katmanı, modelin işleyebilmesi için ham verileri alır ve her nöron belirli bir özelliği temsil eder. Giriş katmanındaki tüm nöronlar, gizli katmandaki nöronlarla bağlantılıdır. Gizli katman, giriş katmanından gelen bilgileri işler, örüntüleri ve ilişkileri öğrenir. Yukarıdaki şekilde gösterildiği gibi birden fazla gizli katman olabilir. Çıkış katmanında ise modelin sonucu üretilir. Çıkış katmanındaki nöron sayısını modelin tahmin ettiği farklı değerlerin veya sınıfların sayısına bağlı olarak belirlenir. Örneğin, çoklu sınıflandırma problemlerinde çıkış katmanındaki nöron sayısı, sınıf sayısına eşit olur.

Yapay sinir ağlarının çalışma prensibi, ileri yayılım ve geri yayılım süreçlerine dayanır. İlk olarak, ağın giriş katmanına verilen veriler, ağırlıklar ve eşik (bias) değerleriyle işlenerek ara katmanlar boyunca aktarılır. Eşitlik 1.1'de gösterildiği gibi bu süreçte girdilere uygulanan ağırlıklar ve eşit değerleri toplanarak net girdi hesaplanır. Bu fonksiyondan elde edilen sonuç bir aktivasyon fonksiyonundan geçirilerek çıktı hesaplanır, eşitlik 1.2'de gösterilmiştir [50].

$$z_i = \sum_{i=1}^{n} (w_{ij}\ x_i + b_j) \qquad (1.1)[50]$$

$$y = f(z_i) = f\left(\sum_{i=1}^{n} \left(w_{ij}\, x_i + b_j\right)\right) \qquad (1.2)[50]$$

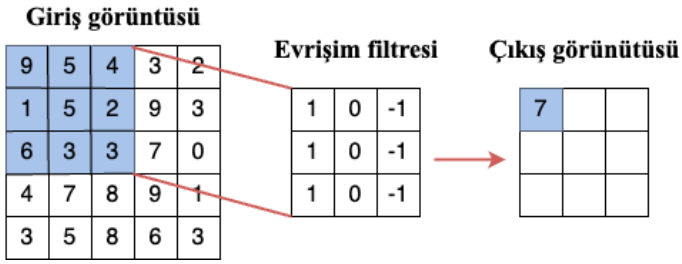
Eşitlik 1.1'de x_i giriş katmanındaki i'inci girdi verisini, w_{ij} giriş katmanındaki i'inci girdinin j'inci nörona olan ağırlığını, b_j j'inci nöronun eşik (bias) değerini ve z_i i'inci nöronun toplam net girdi değeri olarak gösterilmektedir. Eşitlik 1.2'de f aktivasyon fonksiyonu olarak gösterilmektedir.

Aktivasyon fonksiyonları (örneğin, sigmoid, ReLU) ağın doğrusal olmayan problemleri çözmesine olanak tanır. İleri yayılım sonucunda ağın ürettiği çıktı, beklenen değerle karşılaştırılarak hata hesaplanır. Hata, kayıp fonksiyonu (loss function) yardımıyla ölçülür ve bu hata değeri geri yayılım sürecinde ağ boyunca geriye doğru yayılarak ağırlıkların güncellenmesi sağlanır. Bu güncelleme işlemi, genellikle gradyan inişi (gradient descent) veya Adam gibi optimizasyon algoritmaları kullanılarak gerçekleştirilir. Böylece ağ, her bir adımda daha doğru sonuçlar üretecek şekilde kendini optimize eder.

YSA'ların hesaplamalı özellikleri arasında verileri organize etme, öğrenme, uyarlama veya genelleştirme yeteneği ile oluşturulmuş modeller bulunur. Öğrenme aşamasında, hedeflenen giriş-çıkış ilişkisi için çeşitli algoritmalar kullanılır ve birbirine bağlı nöronların ağırlıkları yinelemeli olarak belirlenir [51].

1.3.2. Evrişimsel sinir ağları

Evrişimsel Sinir Ağları (Convolutional Neural Networks - CNN), özellikle görsel veri ve video gibi yüksek boyutlu verilerle çalışan bir yapay sinir ağı türüdür. Bu algoritmada girdi olarak alınan görüntüden anlamlı özellikler çıkarılır ve görüntüdeki nesneler sınıflandırılır. Piksel değerleri, bir görüntüyü oluşturan en küçük birimlerin (piksel) içeriği hakkında bilgi veren 0 ile 255 arasında sayısal değerlerdir. CNN'de girdi olarak görüntülerin piksel değerleri verilir. Evrişim katmanıda Şekil 1.4'de gösterildiği gibi 3x3, 5x5 gibi küçük filtreler tüm görsel üzerinde gezdirilerek öznitelik çıkarılır ve yeni bir görüntü elde edilir [52].



Şekil 1.4. Evrişim katmanı [52].

Elde edilen öznitelik haritası, aktivasyon fonksiyonları ile doğrusal olmayan bir dönüşüme tabi tutulur. Bu dönüşüm, modelin öğrenme kapasitesini artırarak daha karmaşık desenleri anlamasını sağlar. Daha sonra havuzlama (pooling) katmanında öznitelik haritasının boyutu küçülür ve hesaplama maliyeti düşürülür. Bunun yanında önemli öznitelikler korunur. Derin öğrenme mimarilerinin performansı havuzlama

katmanında bu işlemler yapılmadığında önemli ölçüde düşer [53]. Havuzlama, görüntüdeki kritik bilgilerin korunmasını sağlarken, gürültü ve gereksiz detayların etkisini azaltır. Havuzlama ve aktivasyon katmanlarının bir arada kullanımı modelin karmaşık görevlerde yüksek doğruluk oranlarına ulaşmasını destekler. CNN mimarileri, özellikle görüntü sınıflandırma, nesne algılama ve segmentasyon gibi görsel problemler için yüksek doğruluk ve etkinlik sağlayan güçlü bir derin öğrenme yaklaşımıdır.

1.3.3. Tekrarlayan sinir ağları

Tekrarlayan Sinir Ağları (Recurrent Neural Networks - RNN), ardışık veriler üzerinde işlem yapmak için geliştirilmiş bir yapay sinir ağı türüdür. RNN'ler, önceki zaman adımlarındaki bilgiyi saklayarak sonraki adıma aktarır ve zaman bağlantılı veriler üzerinde çalışır. Geçmiş bilgileri hatırlama ve bu bilgileri yeni veriler ile birleştirerek gelecek tahminde bulunma yetenekleri ile ardışık veriler arasındaki bağımlılıkları öğrenerek karmaşık desenleri ve ilişkileri modelleyebilir. Bu özellikleri sayesinde doğal dil işleme, konuşma tanıma, ses işleme, duygu analizi, sensör verilerini anlama gibi alanlarda yaygın olarak kullanılır.

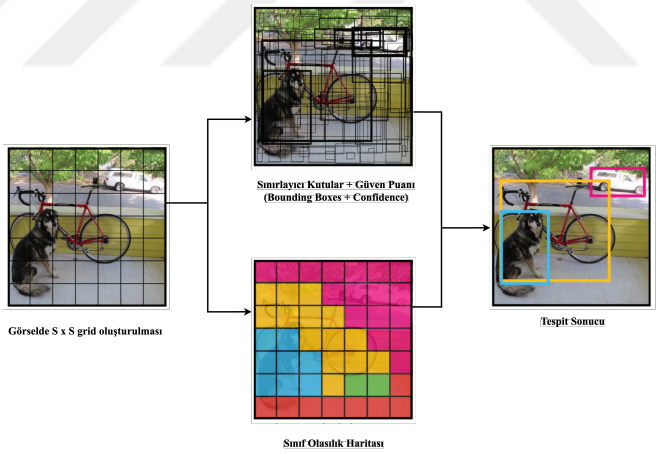
Standart RNN'ler uzun ardışık veriler üzerinde çalışırken gradyan kaybolması (vanishing gradient) sorunu ile karşılaşabilir. Bu sorun ağı güncellemek amacıyla kullanılan gradyanların, çıktı katmanlarından önceki katmanlara geri yayılımı nedeniyle çok küçük hale gelmesi veya kaybolması durumudur. Bu sorun, özellikle uzun vadeli bağımlılıkları öğrenmeye çalışırken daha belirgin hale gelir ve ağın öğrenme kapasitesini önemli ölçüde sınırlandırabilir. Bu problemleri çözmek için başta Gated Recurrent Unit (GRU) ve Long Short-Term Memory (LSTM) olmak üzere gelişmiş RNN mimarileri geliştirilmiştir. Bu mimariler unutma, hatırlama ve güncelleme gibi kapılar kullanarak uzun süreli bağımlılıkları daha etkili bir şekilde öğrenir. LSTM, mevcut örneğin önceki örneklere bağımlılığının önemli olduğu zamansal olarak kalıcı veriler için popüler bir ağ seçimi oluşturur [54]. LSTM' in öne çıkan özelliği bilgileri uzun zaman aralıklarında depolama ve geri alma yeteneğidir. Bunu bilgileri seçici olarak hatırlama veya unutma yeteneğine sahip bir bellek hücresinin kullanımıyla başarır. LSTM'ler, bu kapıları ve bellek hücrelerini kullanarak dizideki önceki zaman adımlarından bilgileri etkili bir şekilde öğrenebilir ve depolayabilir. Böylece uzun vadeli bağımlılıkları yakalayabilir ve tüm dizinin bağlamına göre doğru tahminlerde bulunabilir.

1.3.4. YOLO ile gerçek zamanlı nesne tespiti

2016 yılında Joseph Redmon ve çalışma arkadaşları YOLO'yu nesne tanıma için yeni bir algoritma olarak sunmuşlardır [7]. Geleneksel nesne algılama yöntemleri genellikle

görüntüyü belirli bölgelere ayırarak işlem yaparken YOLO bu süreci tek bir aşamada gerçekleştirmektedir. YOLO mimarisi giriş görüntüsünü bir ızgara yapısına böler ve her bir ızgarada mevcut olan nesnelerin sınıfını ve konumunu tahmin eder. Bu yaklaşım hem işlem hızı hem de zaman verimliliği açısından üstünlük sağlar ve özellikle gerçek zamanlı uygulamalar için uygun çözümler sunar. YOLO temel mimari olarak Evrişimli Sinir Ağı mimarisine dayanır. Bir görüntüyü analiz etmek ve görüntüde algılanan nesnelerin sınırlayıcı kutularını ve olasılıklarını tahmin etmek için derin bir sinir ağı kullanır.

YOLO algoritması tek bir hareketle görüntüdeki birden fazla nesneyi koordinatlarıyla birlikte algılar. Görüntüleri saniyede 45 kare hızında gerçek zamanlı olarak işler. YOLO, gerçek zamanlı performansı ve yüksek doğruluğu sayesinde en hızlı nesne algılama algoritmalarından biridir [55]. YOLO nesne tespit aşamaları Şekil 1.5'te gösterilmektedir. Diğer nesne tanıma algoritmaları tüm görüntüyü işlerken YOLO algoritması görüntüyü bölgelere ayırır. Görüntüde algıladığı nesneleri sınırlayıcı kutular (bounding boxing) adı verilen kutulara çizer. Güven puanı (confidence score), YOLO modelinde her bir bounding box içinde bir nesne içerme olasılığını ifade eder. Bu değer, 0 ile 1 arasında bir olasılık olarak belirtilir ve güven puanı olarak ifade edilir. Yüksek güven puanı, o kutunun gerçekten o nesneyi içerdiğine dair modelin yüksek bir güvene sahip olduğunu gösterir.



Şekil 1.5. YOLO nesne tespit aşamaları[7].

Bir görüntü üzerinde binlerce işlem gerçekleştiren Tekrarlayan Evrişimli Sinir Ağı (R-CNN) gibi yapılardan farklı olarak, YOLO tek bir ağ değerlendirmesiyle tahminlerde bulunur. Bu özellik, YOLO'yu R-CNN'e kıyasla 1000 kat daha hızlı ve etkili kılar [56]. Ayrıca YOLO hem yüksek doğruluk oranını hem de hızlı işlem süresini bir arada sunar.

YOLO mimarisi nesne algılama problemleri için zamanla geliştirilerek birçok versiyona

ulaşmıştır. YOLOv1, görüntüyü NxN ızgaraya bölerek her hücrede nesne sınıfı ve konumunu tek bir evrişimli sinir ağı ile tahmin eder [7]. YOLOv2 ve YOLOv3 ile birlikte farklı boyutlardaki nesneleri daha doğru bir şekilde algılayabilmek için Anchor Box kullanılmıştır [57, 58]. YOLOv4 ve YOLOv5 doğruluk ve hız arasında dengeli bir performans sunmaktadır [59, 60]. YOLOv6, endüstriyel uygulamalar için optimize edilmiş bir mimari sunarak daha yüksek hız ve doğruluk oranı ile gerçek zamanlı nesne tespiti sağlar [61]. YOLOv7'de Genişletilmiş Verimli Katman Toplama Ağı (E-ELAN**),** ağ derinliğini ve genişliğini artırmadan daha iyi gradyan akışı sağlayarak modelin öğrenme yeteneğini güçlendirir [62]. YOLOv8, çapa içermeyen başlık, modern omurga ve bağlantı yapılarıyla, doğruluk ve hız arasında optimize edilmiş bir denge sunar [63]. YOLOv9, programlanabilir gradyan bilgisi (PGI) ve Genelleştirilmiş Verimli Katman Toplama Ağı (GELAN) adlı hafif bir mimariyi entegre ederek daha etkili bir parametre kullanımı sunar [64]. YOLOv10, Non-Maximum Suppression (NMS) gerektirmeyen bir eğitim yaklaşımını benimsemektedir. Modelin daha hızlı, doğru ve verimli çalışmasına olanak tanır. Önceki sürümlere kıyasla son işlem (post-processing) süresinde belirgin bir iyileşme sağlamaktadır [65]. Son işlem süresi modelin tahminlerinin gerçek dünya uygulamalarına uygun hale getirilmesi için gereken süreyi ifade eder.

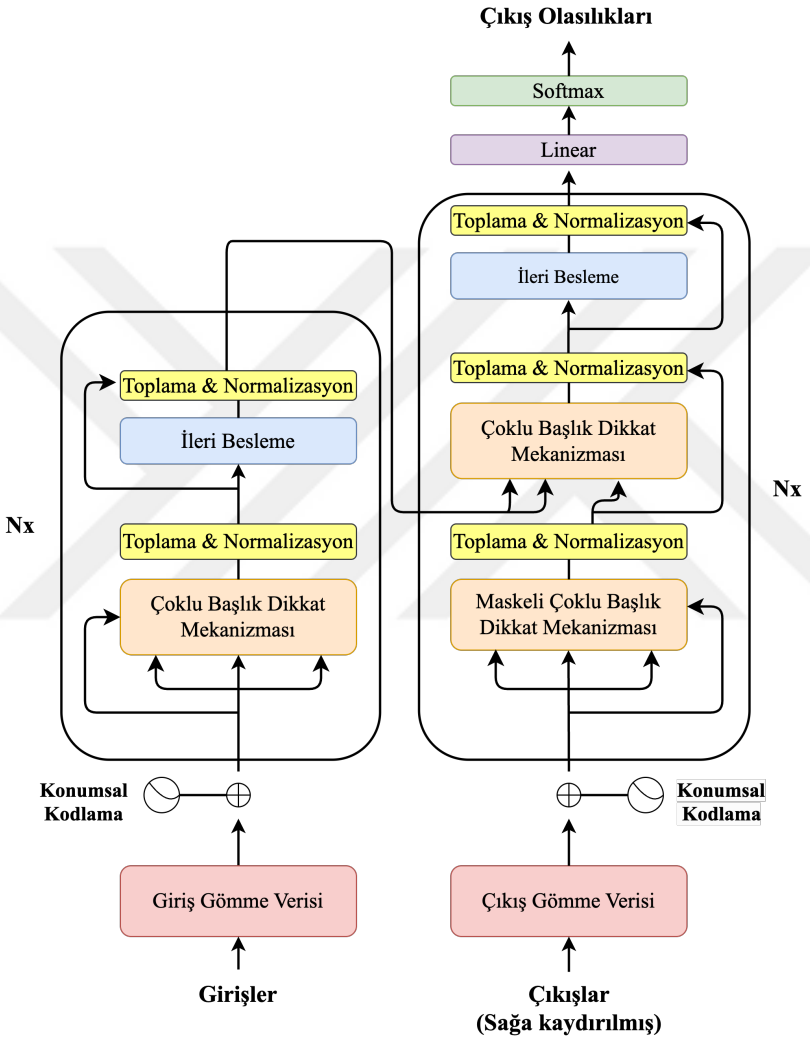
Genel olarak YOLO, ilk versiyonlarından YOLOv8, YOLO-NAS ve Transformers'lı YOLO gibi daha gelişmiş mimarilere doğru evrimleşerek robotik, sürücüsüz arabalar ve video izleme gibi çeşitli uygulamalar için merkezi bir gerçek zamanlı nesne algılama sistemi haline gelmiştir [66, 67]. Algoritmanın sürekli gelişimi ve farklı alanlara uyarlanması, bilgisayarlı görüş ve nesne algılama alanındaki önemini vurgulamaktadır.

1.3.5. GPT

GPT (Generative Pre-trained Transformer), OpenAI tarafından geliştirilmiş bir dil modelidir. Doğal dil işlemede kullanılan bu yapay zekâ modeli, çok büyük veri kümeleri üzerinde eğitilmiştir. Dilin yapısını ve anlamını öğrenerek verilen bir metnin veya sorunun bağlamına uygun olarak metinler üretebilir. İlk GPT modeli 2018 yılında tanıtılmıştır [68]. Dil anlama, metin üretme, çeviri ve özetleme gibi görevlerde gösterdiği başarı ve daha önceki modellere oranla geniş veri kümesi ve model kapasitesiyle önemli bir performans artışı sergilemiştir. Model, Transformer mimari üzerine inşa edilmiştir.

Derin öğrenme tabanlı bir mimari olan Transformer, özellikle dilin uzun mesafeli bağımlılıklarını öğrenme konusunda çok daha verimli ve etkili bir yaklaşım sunmaktadır. Transformer mimarisi Şekil 1.6'da gösterilmektedir ve tamamen dikkat (attention) mekanizmalarına dayalı bir yapıdır. Dikkat mekanizması modelin metindeki belirli kelimeler veya kelime gruplarına odaklanarak bağlamın daha iyi anlaşılmasını sağlar.

Örneğin bir cümlede bulunan kelimenin içerdiği anlamı belirlemek için kelimenin önceki veya kendinden sonra yer alan diğer kelimelerle olan ilişkileri değerlendirilir. Bu yapı sayesinde geleneksel tekrarlayıcı yapıları ortadan kaldırarak dil modellemesinde daha hızlı ve verimli eğitim süreci sunar. Transformer, iki ana bileşen olarak kodlayıcı(encoder) ve kod çözücüyü(decoder) içerir.



Şekil 1.6. Transformer model mimarisi [69].

Kodlayıcı, girdi verisini işler ve her bir kelimenin bağlamını anlamak için dikkat mekanizmalarını kullanarak bilgiyi temsil eder. Kod çözücü ise, kodlayıcı tarafından işlenen bu bilgileri kullanarak verilen girdiyle ilişkili çıktıları üretir. Transformer'da dikkat mekanizmasının temel yapı taşlarından biri çoklu başlık dikkat mekanizması

(Multi-Head Attention)'dır. Bu yapı sayesinde model, metindeki bağlamları aynı anda öğrenebilir ve eğitim sürecini önemli ölçüde hızlandırır [69].

Self-attention, transformer mimarisinin temel bileşenlerinden biri olarak, her kelimenin tüm cümleye olan bağlamını aynı anda öğrenebilmesini mümkün kılar. Bu mekanizma, her öğenin diğer öğelere olan dikkat seviyesini belirleyerek, dizideki her kelimenin ya da öğenin bağlamını anlamasını sağlar. Self-attention fonksiyonu eşitlik 1.3'de gösterilmiştir ve bir sorguyu ve bir anahtar-değer çiftleri kümesini bir çıktıya eşlemek olarak tanımlanabilir; burada sorgu, anahtar, değer ve çıktının hepsi vektördür. Her bir kelime için sorgu, anahtar ve değer vektörü hesaplanır. Sorgu ve anahtar vektörleri arasındaki ilişkiler, nokta çarpımı (dot-product) ile hesaplanarak her öğe için dikkat ağırlıkları oluşturulur. Bu dikkat ağırlıkları her öğenin değer vektörleriyle çarpılarak öğenin bağlamını anlamasını sağlar [69].

$$\text{Attention}(\text{Q}, \text{K}, \text{V}) = \text{softmax}\left(\frac{Q\,K^T}{\sqrt{dk}}\right) V \qquad (1.3)\ [69]$$

Bu eşitlikte Q sorgu vektörü, K anahtar vektörü, V değer vektörünü temsil eder. QK^T her sorgu vektörü ile her anahtar vektörünün çarpımını, d_k key vektörünün boyutunu ifade eder.

GPT serisinin ikinci versiyonu olan GPT-2, önceki mimariyi koruyarak 1.5 milyar parametre ile büyük bir model kapasitesine ulaşmıştır. 40 GB'lık metin verisi üzerinde eğitilmiştir. Bu veriler internetin farklı kaynakları, kitaplar, haberler ve diğer çeşitli kaynaklardan toplanan metinlerdir [70]. Bir diğer versiyon olan GPT-3, 175 milyar parametreye sahip, o zamana kadarki en büyük dil modeli olarak tanıtılmıştır [26]. GPT-3 ve GPT-3.5, doğal dil işleme görevlerinde sıfır denemede öğrenme (zero-shot), tek denemede öğrenme (one-shot) ve birkaç denemede öğrenme (few-shot) yeteneklerine sahip modellerdir. Sıfır denemede öğrenme, modelin eğitim aşamasında görmediği, örnek olarak verilmeyen bir görevi doğru bir şekilde yerine getirebilme yeteneğini ifade eder. Tek denemede öğrenme, yalnızca bir örnekle görevi öğrenmeyi ifade eder. Bu durum modelin sınırlı verilerle genel bilgiye dayanarak doğru sonuçlar üretmesini sağlar. Birkaç denemede öğrenme ise birkaç örnekle modelin daha doğru ve özel sonuçlar elde etmesine olanak tanır. Bu öğrenme yöntemleri, GPT-3 ve GPT-3.5 için esnekliği ve çok çeşitli görevleri yüksek doğrulukla çözme yeteneklerini ortaya koyar. İlk olarak GPT-2 ve özellikle GPT-3 modellerinde daha kapsamlı bir şekilde uygulanan insan geri bildirimiyle güçlendirilmiş öğrenme (Reinforcement Learning from Human Feedback - RLHF) modelin kullanıcı taleplerine daha iyi yanıt vermesini ve daha güvenli, tutarlı çıktılar

üretmesini sağlamaktadır.

2023 yılında tanıtılan GPT-4 versiyonunda ticari ve güvenlik gerekçeleriyle modelin parametre sayısı ve eğitim verisi miktarı açıklanmamıştır. Bununla birlikte GPT-4'ün önceki versiyonu olan GPT-3'e kıyasla daha büyük bir model olduğu ve karmaşık görevlerde daha üstün performans sergilediği vurgulanmıştır [27]. GPT-4, yalnızca metin girdilerini değil aynı zamanda görsel girdileri de işleyebilmektedir. Bu durum dil modelini daha geniş bir uygulama alanına uygun hale getirmektedir. GPT-4 teknik raporuna göre model, insanlara yönelik olarak tasarlanmış çeşitli sınavları içeren bir değerlendirme seti üzerinde test edilmiştir. Modelin görsel ve metinsel verileri işleme kapasitesi, görsel girdiler gerektiren sorularda fotoğraf ve metin girdileri ile modele sunularak değerlendirilmiştir. Değerlendirmelerin sonucunda GPT-4 sınavların çoğunda insan seviyesinde performans sergilemiştir. Modelin sınavlardaki başarısı büyük ölçüde ön eğitim sürecine dayanmaktadır ve RLHF süreci bu sınavlardaki çoktan seçmeli sorular üzerinde belirgin bir etki göstermemiştir [27].

GPT-4, doğal dil işleme alanında önemli ilerlemeler sağlamış olmasına rağmen zaman zaman yanlış bilgiler üretebilmekte ve mantık hataları yapabilmektedir. Bu durum modelin eğitiminde kullanılan veri setlerindeki sınırlamalar ve dil modellerinin doğası gereği oluşan yanıltıcı örüntüler nedeniyle oluşabilmektedir. Özellikle yüksek risk içeren uygulamalarda model çıktılarının insan denetimine tabi tutulması ve ek doğrulama süreçleri ile kontrol edilmedir [27]. TruthfulQA, dil modellerinin önceden örnek gösterilmeden verdiği cevaplarla doğruluk seviyesini ve güvenilirliğini ölçmek amacıyla oluşturulmuş soru-cevap setidir [71]. GPT-4 TruthfulQA gibi değerlendirme setlerinde önceki GPT versiyonlarına kıyasla yanlış veya yanıltıcı bilgi verme oranını azaltmıştır. Model daha doğru ve güvenilir cevaplar verebilmektedir [27].

GPT-4o, “omni” çok yönlü özellikleri sayesinde metin, ses ve görüntü gibi farklı veri türlerini işleyebilen gelişmiş bir yapay zeka modelidir. Modelin çoklu dil desteği farklı dillerdeki soruları anlamasına ve bu sorulara uygun yanıtlar üretmesine imkân tanır.

Bunun yanı sıra GPT-4o belirli alanlara yönelik sorulara yanıt verme konusunda da geliştirilmiştir. Örneğin, Japonya Ulusal Diş Hekimliği Sınavı’nda (JNDE) yapılan bir çalışmada, GPT-4o’nun %73.8 doğruluk oranı ile GPT-4’e kıyasla daha yüksek başarı gösterdiği tespit edilmiştir. Temel bilgi gerektiren sorularda önemli bir iyileşme sağlamıştır. Bu sonuçlar, GPT-4o’nun eğitim alanında ve alan bazlı bilgi değerlendirmelerinde güçlü bir yapay zekâ modeli olarak kullanılabileceğini göstermektedir [72]. Ayrıca görüntü işleme yetenekleri sayesinde verilen görselleri analiz etme ve bu görsellerle ilgili açıklamalar üretme kapasitesine sahiptir. Örneğin bir görüntü

verildiğinde model, bu görüntüdeki nesneleri, sahneleri veya detayları tanımlayabilir. Bu veriler hakkında anlamlı metin tabanlı açıklamalar üretebilir. Bu çok yönlü işleme yeteneği GPT-4o'yu çeşitli uygulama alanlarında etkili bir araç haline getirmektedir [73]. Sadece doğal dil işleme görevlerinde değil aynı zamanda görüntü tanıma, sesli asistan uygulamaları, çeviri, içerik üretme gibi farklı alanlarda da etkili bir şekilde kullanılabilen çok yönlü bir yapay zekâ modelidir.

OpenAI'ın en son duyurduğu GPT sürümü GPT-4.1 olmuştur. Nisan 2025'te tanıtılan bu modelde önceki sürümlere kıyasla daha uzun bağlamları işleyebilme kapasitesi ve daha düşük gecikme süresi gibi önemli iyileştirmeler yer almaktadır. Ayrıca talimatları daha doğru takip etme ve kodlama görevlerinde daha yüksek başarı elde etme gibi özellikleri ile daha verimli bir araç haline geldiği ifade edilmektedir [74].

1.4. Literatür Çalışmaları

Evrişimsel sinir ağları kullanılarak nesne tespiti ve sınıflandırma, doğal dil işleme, metin oluşturma gibi konularda birçok çalışma vardır. Bu bölümde yapılan çalışmalar incelenmiştir. YOLO algoritması ile son yıllarda çeşitli alanlarda uygulamalar gerçekleştirilmiş ve önemli gelişmlere elde edilmiştir.

Liao ve arkadaşları çalışmalarında YOLOv11s mimarisi üzerine inşa ettikleri YOLO-MECD modelini gerçekleştirmişlerdir. Farklı turunçgil türlerini (pomelo, kumkuat vb.) içeren 1200 görüntüyle eğitim gerçekleştirmişlerdir. EMA (Efficient Multi-scale Attention) dikkat mekanizması kullanarak modelin öznitelik çıkarımını güçlendirmişlerdir. Deneyler sonucunda model turunçgil tespitinde %84,4 kesinlik, %73,3 duyarlılık ve %81,6 ortalama doğruluk (mAP) elde etmiştir. Model, farklı arka plan gürültüsü, yaprak örtüsü ve ışık koşullarında da tutarlı sonuçlar vererek gerçek dünya uygulamaları için sağlam bir temel oluşturmuştur [75].

He ve arkadaşlarının çalışmalarında önerdiği YOLOv11-seg tabanlı akıllı heyelan tanıma yöntemi, optimizasyonlu özellik çıkarım ve segmentasyon modülleriyle karmaşık arazi koşullarında yüksek doğruluk sağlamıştır. Bijie-Landslide veri seti üzerinde uygulanan kapsamlı veri arttırma yöntemleri ile model, sınır tespiti için 0,8781 F1 skoru ve piksel düzeyinde segmentasyon için 0,8114 F1 skoru elde edilmiştir. Sonuçlar YOLOv11-seg'in gerçek zamanlı heyelan izleme süreçlerinde güvenilir bir çözüm sunduğu ve jeolojik afet önleme ve risk değerlendirme çalışmalarında kullanılabileceğini göstermektedir [76].

Wang ve arkadaşları gerçek zamanlı nesne tespiti için yeni bir yaklaşım olan Mamba YOLO modeli tanıtmışlardır. Mevcut YOLO modelleri güçlü performans sergilemektedir fakar özellikle Transformer tabanlı yaklaşımlarda kullanılan self-attention mekanizması

yüksek hesaplama maliyetine sahiptir. Bu problemi çözmek amacıyla bu çalışmada self-attention yerine State Space Model (SSM) yapısı kullanılmıştır ve modelin hesaplama karmaşıklığı lineer hale getirilmiştir. Gerçekleştirilen testler sonucunda Mamba YOLO, diğer güncel yöntemlerle karşılaştırıldığında daha düşük gecikme süresi ve daha yüksek ortalama doğruluk (mAP) değerlerine ulaşmıştır [77].

Gallagher ve Oughton çalışmalarında YOLO algoritmasının multispektral nesne tespiti ve görüntüleme bağlamındaki uygulamalarını ve karşılan zorlukları incelemektedir. 2020-2024 yılları arasında yayımlanan 400 makale değerlendirilmektedir. YOLO'nun multispektral ortamlara uyarlanmasında en sık kullanılan versiyonun YOLOv5 olduğu ifade edilmektedir. Çalışmalarda kullanılan en yaygın sensör kombinasyonu ise RGB ve uzun dalga kızılötesi (LWIR) sensörlerinin konfigürasyonu olmuştur. Çalışma, multispektral YOLO uygulamalarının çoğunlukla yer tabanlı sistemlerde kullanıldığı ve insansız hava araçlarıyla yapılan uygulamaların 2020'den bu yana iki katına çıktığı gözlemlenmiştir [78].

Aly ve arkadaşları, meme kanseri taraması için kullanılan dijital mamografi görüntülerinde Tam Alan Dijital Mamogramlarda (Full-field digital mammograms (FFDMs)), YOLO tabanlı meme kitlelerinin tespiti ve sınıflandırılmasına odaklanarak algoritmanın tıbbi görüntüleme uygulamalarındaki çok yönlülüğünü ortaya koymuştur [79]. Ayrıca Gai ve çalışma arkadaşları, su ürünleri yetiştiriciliği için su altı görüntülerinde kiraz meyvelerini, Hu ve çalışma arkadaşları ise yenmemiş yem tanelerini tespit etmede YOLO'nun uygulamalarını sunarak algoritmanın bu konularda da uygulanabilir olduğunu sergilemişlerdir [80, 81].

Sürücü dikkat dağınıklığı ve hatalarının çevreye ve sürüş güvenliğine etkilerini ele alan çalışmada YOLOv7 ve YOLOv8 modelleri kullanılmıştır. Kavşaktaki sürücülerin trafik ışıklarında beklerken görüntüleri kaydedilmiştir ve görüntüler üzerinden sürücü dikkat dağınıklıkları ve cep telefonu kullanımları tespit edilmiştir. Çalışmanın sonucunda YOLOv7 modelinin %91,17, YOLOv8 modelinin %92,03 doğruluk oranı ile etkin bir yöntem olduğu sonucuna varılmıştır [82].

Ayrıca Optik Karakter Tanıma (OCR) teknolojisi ile kameredan okunan metinler yazıya dönüştürülmüştür, sonrasında metinden konuşmaya teknolojisi ile seslenlendirilmiştir. Bu yöntemle görme engelli bireylerin basılı metinleri Braille alfabesine dönüştürmeden anlayabilmeleri sağlanmıştır [83].

Chen ve Zhu 2023 yılındaki çalışmalarında, gerçek zamanlı 3 boyutlu nesne algılayan bir mobil uygulama gerçekleştirmişlerdir. 2 boyutlu nesne tespiti sonuçlarını, ARKit'in sağladığı nokta bulutu verileri ile nesnelerin 3 boyutlu konumları hesaplanmıştır. SiriKit

kullanılarak uygulamaya sesli komutlar entegre edilmiştir. ARKit, yatay ve dikey düzlemleri tespit ederek engellerin veya yüzeylerin 3 boyutlu olarak kutuları tahmin etmiştir. Çalışmanın sonucunda düşük maliyetli ve gerçek zamanlı çalışan bir sistem elde edilmiş ve görme engelli ya da otizm spektrum bozukluğu olan bireyler için kullanışlı olmuştur. Hareketli nesnelerin ve kalabalık ortam sahnelerinin algılanmasında zorluklar yaşanmıştır [84].

Sensor füzyonu, birden çok sensörden alınan verilerin birleştirilerek, bir ortamın veya bir sistemin daha doğru ve kapsamlı bir anlayışını sağlama süreci olarak ifade edilebilir. 2023 yılında yapılan bir tez çalışmasında sensor füzyonu gerçekleştirilerek iPhone'daki yerleşik kamera ve LiDAR sensörü kullanılarak 0 ile 15 m arasında sınırları belirlenen bir alanda nesne tanıma işlemi gerçekleştirilmiştir. LiDAR ile belirlenen belirli bir santimetre genişliğindeki alanda YOLOv5 ile nesnelerin tanıma işlemi gerçekleştirilmiştir. Swift Metal çerçevesi ile video işleme gerçekleştirilmiştir [85].

Alsamurai, 2023 yılında yaptığı çalışmada YOLOv8 algoritması kullanılarak görseller üzerinde orman yangınlarında ateş, duman, insan ve hayvanların tespitini gerçeleştirmiştir. Çalışmadaki model küçük boyuttaki yangınların, dumanın tespitinde ve insan ile hayvan görsellerini birbirinden ayırma konusunda zorlanmıştır. Eğitim için kullanılan veri setinin boyutunu ve çeşitliğini artırmanın modelin doğruluğunu artırmada etkili olduğu sonucu çıkarılmıştır [86].

2023 yılında yapılan yüksek lisans tez çalışmasında R-CNN (Bölgesel Tabanlı CNN) mimarisi kullanılarak YOLOv3 ve SSD mimarisi ile nesne tanıma gerçekleştirilmiştir. Çalışmada gerçek zamanlı olarak alınan görüntülerden nesne tespiti yapacak bir mobil sistem düzenlenmiştir. Kullanılan algoritmaların performansı karşılaştırıldığında YOLO-v3 0,9741 doğruluk oranı ile 0,9394 oranındaki SSD algoritmasına göre başarılı olmuştur [87].

Adli delillerden olay ve kişi tespiti yapabilmek amacıyla gerçekleştirilen tez çalışmasında YOLOv8 modeli ile görsellerden kişi ve yüz tespiti gerçekleştirilmiştir. Görünütüler ve videolar girdi olarak verildiğinde, olayların ve kişilerin tespiti başarılı bir şekilde gerçekleştirilmiştir [88].

Moreira ve arkadaşları çalışmalarında, Google'ın Gemini modeli kullanılarak görme engelli bireyler için bir mobil uygulama geliştirilmiştir. Uygulama, kameradan veya galeriden alınan görüntüleri analiz ederek bağlamsal açıklamalar oluşturmuş ve sesli olarak ifadeler oluşturmuştur. Çalışmada gerçekleştirilen testler sonucunda modelin doğruluğu genellikle yüksek seviyelerde olup yalnızca %14 oranında düşük doğruluk gözlemlenmiştir. Görüntü kalitesi ve ışıklandırma doğruluk üzerinde etkili olduğu

sonucuna varılmıştır. Çalışma, mobil uygulamanın görme engelli bireylerin bağımsızlıklarını artırmalarına yardımcı olduğunu göstermektedir [89].

CoreML, Apple'ın geliştirdiği bir çerçevedir ve makine öğrenimi modellerini Apple platformlarında uygulamalara entegre etmeyi sağlar. Sunil Bhutada ve arkadaşları CoreML kullanarak gerçek zamanlı görüntüler alarak yüz tanıma çalışması yapmışlardır. Çalışmalarının sonucunda yüz tanıma konusunda performans iyileştirmelerine ihtiyaç olduğu belirtilmiştir [90].

Alamsyah ve arkadaşları, bir nesnenin İngilizce adını öğrenmek için kullanılabilecek bir mobil uygulama gerçekleştirmişlerdir. Çalışmada, mobil tabanlı bir evrişimli sinir ağı olan MobileNetV2 modeli kullanılmıştır. Çalışmada kamera ile gösterilen nesnelerin tespitinden sonra ekrana tıklandığında nesnenin ingilizce isimleri gösteren etiketler eklenmiştir, ardından nesnenin ismi seslendirilmiştir. Nesne tanıma modeli için Apple'ın ARKit ve CoreML, metinden sese dönüştürmek için AVKit, artırılmış gerçeklik için ARKit çerçeveleri kullanılmıştır. Artırılmış Gerçeklik (AR) ile Görüntü Tanıma bir arada kullanılmıştır [91].

Kumar ve arkadaşları çalışmalarında pamuk bitkisinin hastalık tespiti için etiketli veriler ile TensorFlow TFLite modeli oluşturmuşlardır ve sonrasında CoreML'e dönüştürülmüşlerdir. Çalışmada hastalık tespiti için tahminleme modelinde CNN algoritması kullanılmıştır. Gerçekleştirilen mobil uygulama ile koza çürüklüğü ve mantar yaprak lekesi tespitinde %90 başarı oranı elde etmişlerdir [9].

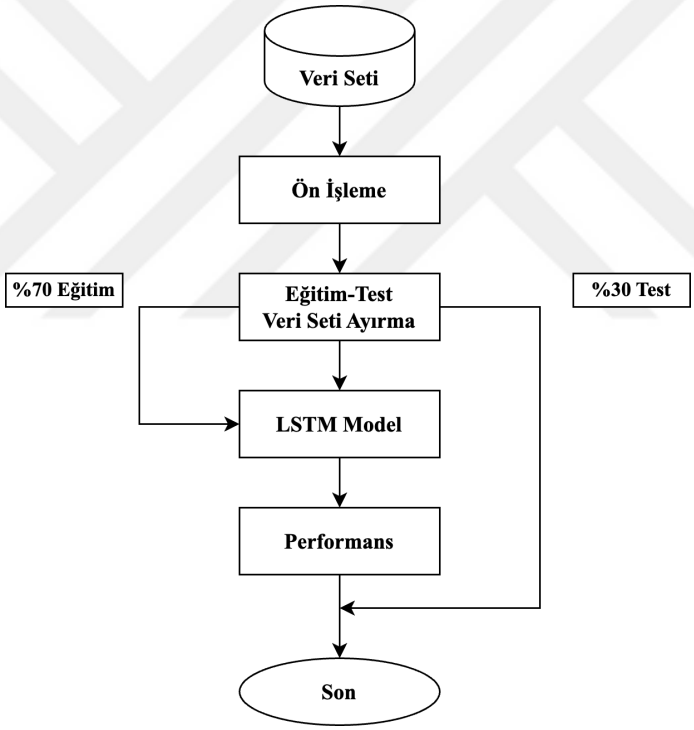
Ciltteki melenom kanserini tespit etmek için iki farklı algoritma kullanılan çalışmada, görüntüleri sınıflandırmak için regresyon algoritması, nesne tespiti için de derin öğrenme yöntemi kullanılmıştır. Melenom varlığının tespitinde hangi algoritmanın daha başarılı olduğunu tespit etmek amaçlanmıştır. Çalışma sonucunda CNN'in hasta bireyler için melonom tespitinde, doğrusal regresyonuna oranla daha iyi bir yöntem olduğu, sağlıklı bireyde sağlıklı olduğunu tespit etmede de regresyon modelinin daha iyi olduğu sonucuna varılmıştır [92].

SSD Mobile Net ile YOLOv5'in görüntülerdeki araç ve insanları tespit etme ve bu nesnelerin sayılarını belirleme görevlerindeki performansları karşılatırılmıştır. YOLOv5s'in gerçek zamanlı olarak nesne tespitinde SSD Mobile Net'e göre daha başarılı ve etkili bir model olduğu sonucuna varılmıştır. Mobile Net-SSD'nin YOLOv5'lere benzer bir hız sunduğu fakat hassasiyet konusunda daha geride kaldığı belirtilmiştir [93].

2021 yılında Karthi ve arkadaşları yaptıkları çalışmada 5 farklı veri seti üzerinden YOLOv5 algoritmasının performansını değerlendirmişlerdir. Kullanılan veri setlerinden biri kütüphanedeki kitapları içeren özel olarak oluşturulmuş bir görüntü setidir. Diğerleri

literatürde mevcut olan büyük veri setleridir. Çalışmanın sonucu YOLOv5 algoritmasının veri setlerinde iyi bir performans sergilediğini ve gerçek zamanlı senaryolarda yaygın olarak kullanılabileceğini göstermiştir. Gelecekteki çalışmalarda daha fazla veri artırma yönteminin incelenmesi ve genişleme oranının artırılması önerilmektedir [94].

Doğal dil işleme ve metin üretme ile ilgili yapılan tez çalışmaları incelenmiştir. 2024 yılında yapılan bir çalışmada LSTM, metindeki uzun süreli bağımılıkları ve anlamları daha iyi kavramak amacıyla kullanılmıştır. LSTM ile bağlam vektörleri çıkarılmış ve kelimeler içerdikleri anlamlara göre gruplandırmışlardır. Bu model için Nietzsche ve Shakespeare'in eserlerini içeren veri setleri ile eğitim gerçekleştirilmiş ve test edilmiştir. Çalışmanın akış diyagaramı Şekil 1.7'de yer almaktadır. Gerçekleştirilen modelde yüksek doğruluk ve düşük kayıp oranları ile metinlerin oluşturulduğu ifade edilmiştir [95].



Şekil 1.7. Cümle üretme akış diyagramı [95].

Mitchell ve arkadaşları 2010 yılındaki çalışmalarında, görsel bir sahnedeki nesnelere referans oluşturma ve bir ifade oluşturma ile ilgili yapıları doğal dil işleme çalışmaları incelenmiştir. Bir görselde nesnelere yapılan referansın, nesnelerin konumlarına dayalı olmasına ek olarak nesnelerin özelliklerine bağlı anlayışın da rol oynadığı ifade edilmiştir. Çalışmaya göre, insanlar tarafından bir nesneye atıfta bulunurken hem

nesnenin konumuna hem de özelliklerine dikkat edilerek ifadeler oluşturulmuştur [96].

Türkçe metin oluşturmada derin öğrenme modellerinin başarılarının değerlendirildiği 2023 yılındaki tez çalışmasında, LSTM, transfer öğrenme ve GPT gibi derin öğrenme modelleri karşılaştırılmıştır. Türkçeye uygun olarak gerçekleştirilen alt kelime tokenizasyonunun diğer yöntemlere oranla daha başarılı olduğu ifade edilmiştir. Ayrıca GPT gibi modellerin yeterli veriyle kullanıldığında anlamlı sonuçlar sunduğu da eklenmiştir. Türk dilinin yapısıyla uyumlu olan Transformer yapısındaki alt kelime yerleştirmeleri, dikkat mekanizması ve konumsal yerleştirmelerin önemi vurgulanmıştır [97].

Anayurt tez çalışmasında derin öğrenme yöntemleri kullanarak video ve fotoğraflardan metin üretme problemine çözüm üretmeye odaklanmıştır. Denetimli ve denetimsiz olmak üzere iki yaklaşım üzerinde çalışmıştır. İlk kısımda videolarda geçen atıfları anlama ve oluşturme üzerinde ilişkisel tümleçler kullanılmıştır. İkinci kısımda ise etiketlenmiş veri eksikliği gibi problemlere çözüm olarak denetimsiz bir yaklaşım denenmiştir. Bu yaklaşım ile ActivityNet Captions veri setinde yapılan denemelerde modelin videolardaki olayları anlayamadığı sonucuna varılmıştır [98].

LLM'ler temelde metin oluşturma, işleme gibi metin bazlı görevler için geliştirilmiştir. LLM tabanlı olan GPT-4 Vision (GPT- 4V)'nin ve görsel işlemler ve görme yeteneği için genişletilmesi yenilikçi ve önemli bir adımdır. Bununla birlikte GPT- 4V'nin metin merkezli mimarisi nedeniyle tıp alanında görsel verileri anlama ve nesneleri tanıma gibi işlemlerde bazı zorlukları da beraberinde getirmektedir. Shunsuke K. ve Wei D. çalışmalında CNN'lerin şu anda tıbbi görüntüleme alanlarında LLM'lere göre daha başarılı olduğunu ifade etmişlerdir. Ek olarak GPT-4V'nin tıbbi tanı koymada etkili olma potansiyelinin hala devam ettiği vurgulanmaktadır. Gelecekteki araştırmalar, karmaşık verilerin daha etkili bir şekilde incelenmesi ve GPT-4V gibi teknolojilerin sağlık alanındaki etkilerini daha doğru bir şekilde değerlendirmek için gelişmiş ölçüm kriterleri geliştirmeyi hedeflemelidir [99].

LLM'ler otomatik kod tamamlama amacıyla yazılım geliştirme süreçlerinde önemli katkılar sağlamıştır. Otomatik kod tamamlama ile bir sonraki değişken ismini, fonksiyon veya diğer kod parçaları tahmin edilerek geliştiricilere daha hızlı ve verimli bir kod yazma deneyimi sunulabilir. LLM'ler bunun gibi çeşitli alanlarda umut verici çözümler sağlar [100].

John Roberts ve arkadaşları, ChatGPT gibi büyük dil modellerinin, insan benzeri yeteneklerini nasıl kazandıkları, bu yeteneklerinin nitel araştırmalar için yapacakları çıkarımlarının hangi durumlarda yanıltıcı ya da aldatıcı olabileceğini keşfetmek için bir

araştırma yapmışlardır. Bir büyük dil modelinin bir soru sorulduğunda bilmediğini kabul edeceği durumlar, kendi eğitim verilerinin ötesindeki bir tarihe ilişkin bir sorunun yöneltilmesi ya da belirli bir yanıtı vermesini engellemek amacıyla modelin önceden sınırlandırılmış olması durumudur. Bunun dışında LLM'ler gerçek ile yalan ayrımı yapma yeteneklerine sahip değillerdir. Bu yüzden kötü niyetli olmaksızın, hayali senaryoları güvenilir bir şekilde gerçekmiş gibi sunabilmektedirler [101].

2024 yılında gerçekleştirilen bir çalışmada, E-ticaret platformlarındaki kullanıcı yorumlarına yönelik duygu analizinde GPT-3.5, LLaMA-2, BERT ve RoBERTa modellerinin karşılaştırmalı performansları değerlendirilmiştir. LLM'ler hem temel hem de ince ayarlı formlarında inceleme puanları için yaklaşık %65 oranında doğru tahminde bulunmuştur. Bu sonuç, Büyük Dil Modellerinin sözcüksel öğeleri nasıl değerlendirdiğini ve bu değerlendirmelerin gerçek insanların değerlendirmeleri ile ne kadar örtüştüğünü vurgulamaktadır [102].

LLM'ler tıpta büyük bir potansiyele sahiptir ancak klinik uygulamalara entegre edilmeden önce sürekli iyileştirme ve etik gözetim gerektirir. Veri sınırlamaları, modelin bağlamı veya durumun karmaşıklığını yeterince iyi anlayamaması ve bu yüzden hatalı veya eksik sonuçlar üretmesi gibi zorluklara dikkat edilmelidir. LLM'ler, bilgi konsolidasyonu, kişiselleştirilmiş bakım ve klinik karar alma süreçlerinde bir dönüşüm gerçekleştirebilir. Ancak insan uzmanlığı ve etik, yüksek kalitede, eşitlikçi bir sağlık hizmetleri için vazgeçilmezdir [103].

LLM'ler süreçleri otomatize etmek amacıyla da kullanılabilmektedir. İnşaat denetim raporlarının oluşturulmasını otomatik olarak gerçekleştirmek için AutoRepo adında bir çerçeve geliştirilmiştir. Bu yapı 3 ana modülden oluşmaktadır. Bu modüller sırasıyla, veri toplama modülü, büyük dil modeli modülü ve rapor oluşturma modülüdür. Geleneksel olarak oluşturulan denetim raporlarına kıyasla önemli kazanımlar elde edilmiştir. Denetim sürecini hızlandırma, kaynakları etkin bir şekilde kullanma ve standartlara uygun, yüksek kalitede raporları otomatik olarak üretme gibi avantajlar sağlamıştır [104].

2024 yılında yapılan çalışmada, Büyük bir dil modeli kullanılarak insan-robot iş birliği (HRC) ile montaj işlemlerini daha etkili hale getirmek için görme tabanlı bir yöntem geliştirilmiştir. Montaj görevlerini metin olarak tanımlamak, robot kontrol komutlarını anlamak ve bu komutları doğru bir şekilde uygulamak için LLM temelli bir akıl yürütme yöntemi geliştirilmiştir. Bu yöntem, robotun insan talimatlarını doğal bir dilde anlayarak görevleri yerine getirmesini sağlanmıştır [105].

2. MATERYAL VE YÖNTEM

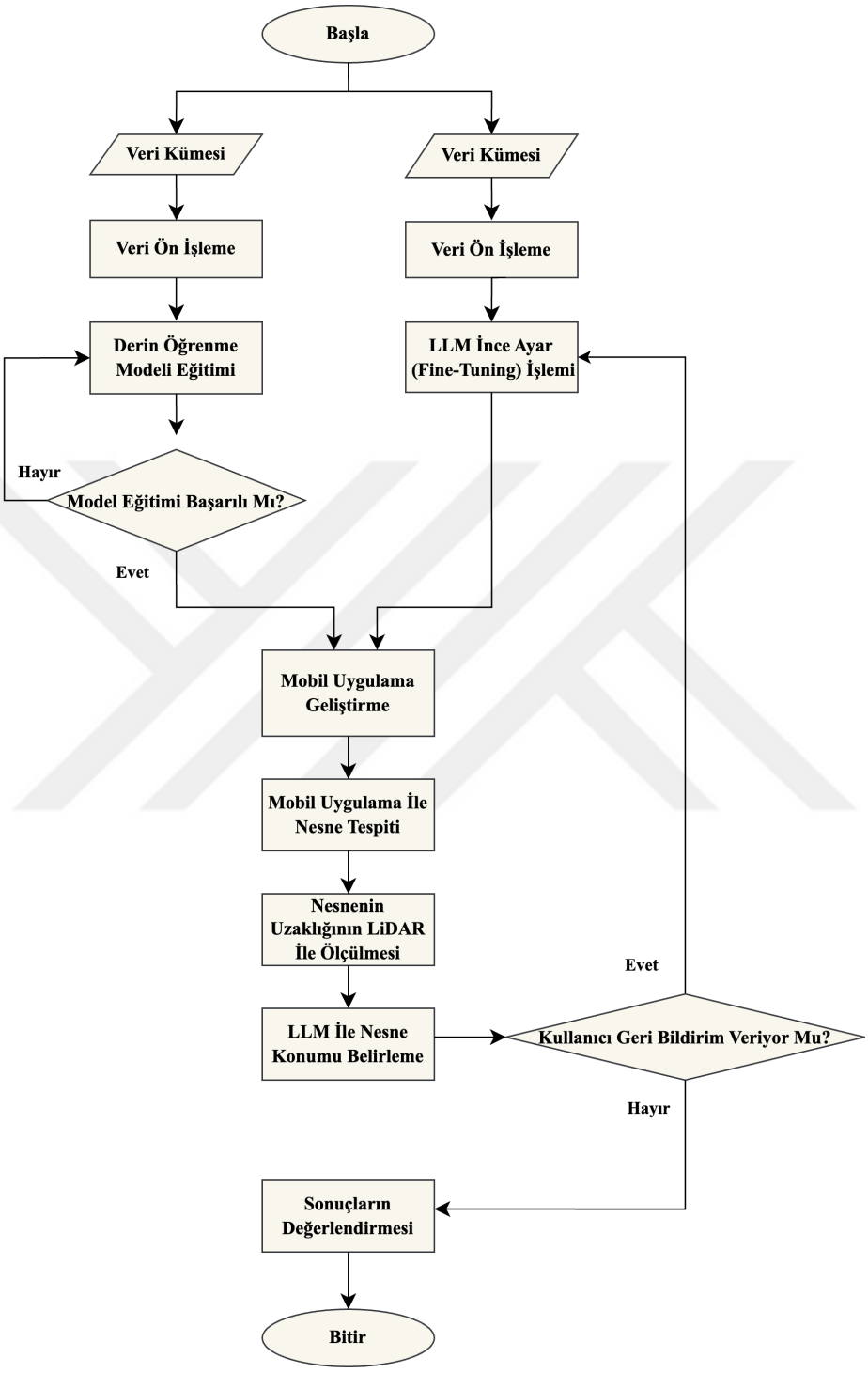
Bu tez kapsamında derin öğrenme mimarisi ile nesne tespiti için YOLO v11 kullanılarak model eğitimi gerçekleştirilmiştir. Model eğitiminde literatürde yaygın olarak kullanılan hazır veri setine ek olarak özgün bir biçimde oluşturulan veri seti eklenmiştir

Elde edilen derin öğrenme modeli CoreML ile mobil kullanımı sağlanmıştır. CoreML, Apple tarafından geliştirilen ve makine öğrenimi modellerini Apple platformlarında kullanmayı kolaylaştıran bir çerçevedir [106]. YOLOv11 modeli, CoreML formatına dönüştürülerek mobil cihazlarda verimli bir şekilde çalışması sağlanmıştır. Tezde uygulanan adımlar ve yöntemler Şekil 2.1'de akış diyagramında gösterilmektedir. Şekil 2.2'de ise çalışmanın sözde kodu yer almaktadır. Derin öğrenme modeli ve LLM tabanlı sistemin bir arada kullanıldığı mobil uygulama geliştirilmiştir.

Kullanıcı mobil uygulama ile kendisine sunulan listeden istediği nesneyi seçerek ilerlemektedir. Nesnenin tespit edileceği ekran açılmaktadır ve bu ekranda YOLOv-11 modeli ile nesne tespiti gerçekleştirilmektedir. Nesne yakalandığı anda anlık olarak fotoğrafı kaydedilmektedir. Ardından mobil uygulama üzerinden LiDAR teknolojisi kullanılarak nesnenin cihazla olan uzaklığı ölçülmektedir. Kaydedilen fotoğraf ve nesnenin konumunu sorgulayan cümle JSON formatında GPT-4o modeline iletilmektedir. GPT-4o, görseldeki nesnenin çevresindeki diğer nesneler ve ortamla ilgili bilgileri analiz ederek bu verileri doğal dilde bir ifade olarak üretmektedir. Son olarak, nesnenin uzaklığını belirten ifade ile GPT-4o'nun ürettiği açıklama birleştirilerek kullanıcıya yazılı ve sesli olarak sunulmaktadır.

Mobil uygulama, görme engelli kullanıcılar için VoiceOver(seslendirme) teknolojisini entegre ederek erişilebilirlik açısından önemli bir özellik sunmaktadır. VoiceOver, kullanıcıların ekrandaki metinleri sesli olarak duyabilmesini sağlayarak uygulamanın özelliklerine kolayca erişmelerini mümkün kılmaktadır [107].

Çalışmanın bu bölümünde kullanılan veri seti, derin öğrenme modeli, LLM ve geliştirilen mobil uygulama hakkında detaylı bilgilere yer verilmektedir.



Şekil 2.1. Tez akış diyagramı.

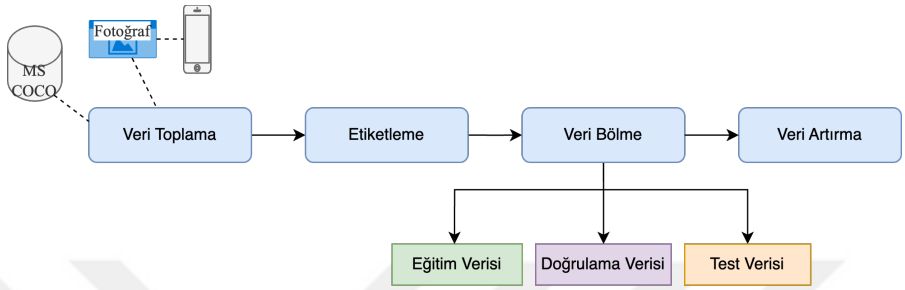
```
START
// Load and preprocess dataset for Deep Learning model
DATASET_DL ← LoadDataset()
PREPROCESSED_DATA_DL ← Preprocess(DATASET_DL)

// Train Deep Learning model
MODEL ← TrainDeepLearningModel(PREPROCESSED_DATA_DL)

// Load and preprocess dataset for LLM
DATASET_LLM ← LoadDataset()
PREPROCESSED_DATA_LLM ← Preprocess(DATASET_LLM)

// Fine-tune the LLM model
LLM_MODEL ← FineTuneLLM(PREPROCESSED_DATA_LLM)

// Check if model training is successful
IF ModelTrainingSuccessful(MODEL) THEN
    // Develop the mobile application
    MobileApp ← DevelopMobileApplication()

    // Set object name and test image
    Object_Name ← SET_ObjectName()
    Test_Image ← SET_TestImage()

    // Perform object detection
    DETECTED_OBJECT ← DetectObject(MobileApp, Test_Image, Object_Name)

    // Measure object distance using LiDAR
    OBJECT_DISTANCE ← MeasureDistanceWithLiDAR(DETECTED_OBJECT,Test_Image)
    // Determine object location using LLM
    OBJECT_LOCATION ← DetermineLocationUsingLLM(DETECTED_OBJECT, Test_Image, OBJECT_DISTANCE, LLM_MODEL)
    // Check if the user provides feedback
    IF UserProvidesFeedback(OBJECT_LOCATION) THEN
        EvaluateResults(OBJECT_LOCATION)
    ELSE
        EvaluateResults(OBJECT_LOCATION)
    ENDIF
ELSE
    // Retry model training if unsuccessful
    REPEAT ModelTraining(PREPROCESSED_DATA_DL) UNTIL ModelTrainingSuccessful(MODEL)
ENDIF

FINISH
```

Şekil 2.2. Sözde kod.

2.1. Veri Seti

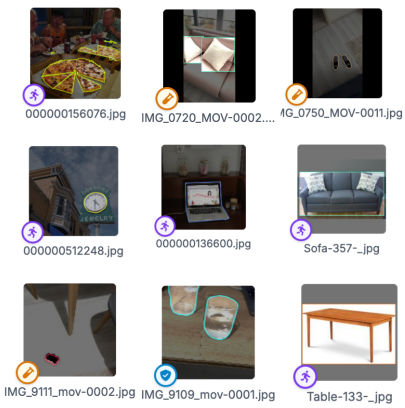
Bu çalışmada, MS COCO 2017 [108] veri seti ihtiyaçlar doğrultusunda özelleştirilmiş ve gereksiz sınıflara ait görseller çıkarılarak uygun bir alt küme oluşturulmuştur. RoboFlow

üzerinden mobilya görselleri içeren bir veri seti eklenmiştir [109]. Buna ek olarak ev içinde kullanılan çeşitli nesnelerin görselleri iPhone 14 Pro ve iPhone 14 Pro Max telefonlar ile fotoğraflanmış, nesneler etiketlenmiş ve veri setine dahil edilmiştir. Şekil 2.3'te veri seti hazırlama aşamaları gösterilmiştir. Test veri setinde modelin performansını artırmak amacıyla veri artırma teknikleri uygulanmıştır, görsellere saat yönünde 90° döndürme (rotate) işlemi uygulanmıştır.



Şekil 2.3. Veri seti hazırlama aşamaları.

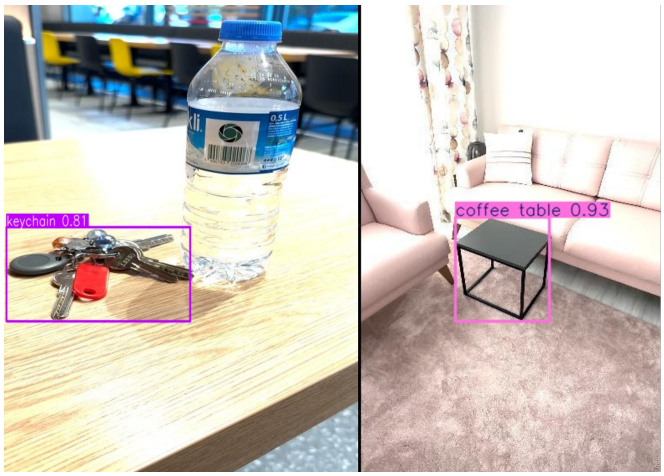
Her biri 640x640 piksel boyutunda, 27 farklı nesneyi içeren toplam 2248 görselden oluşan veri seti elde edilmiştir [110]. Tablo 2.1'de nesne sınıfları ve örnek sayıları gösterilmektedir. Hem hazır verilerden hem de fotoğraf çekilerek üretilen görsellerden oluşan zengin ve özgün bir veri seti hazırlanmıştır. Veri setinde yer alan etiketli görsellerden bir kısmı Şekil 2.4'te gösterilmektedir. Veri seti, %70 eğitim, %20 doğrulama ve %10 test olacak şekilde oluşturulmuştur. Şekil 2.5'te test veri setinde gerçekleşen nesne tespiti örnekleri yer almaktadır.



Şekil 2.4. Etiketli görseller.

Tablo 2.1. Nesne sınıfları ve örnek sayıları

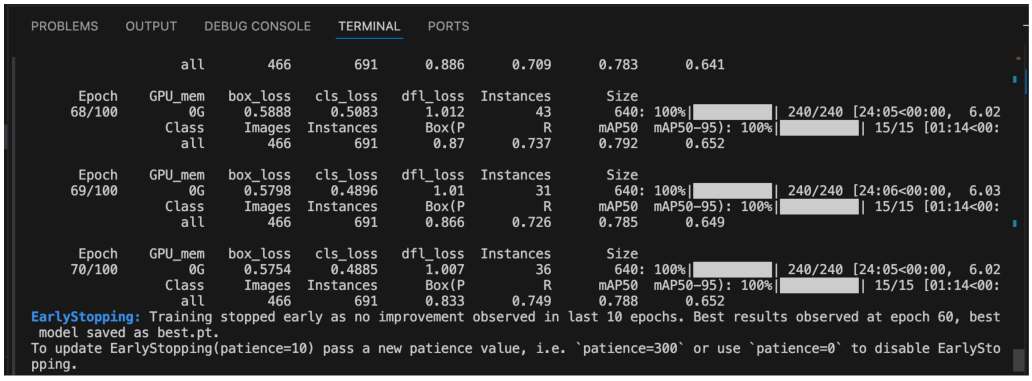
Sınıf Adı	Örnek Sayısı
Saat	232
Sandalye	249
Koltuk	240
Yatak	139
Elbise Askısı	134
Masa	200
Kırlent	198
Sehpa	224
Anahtarlık	144
Ceket	113
Gözlük	381
Bıçak	77
Bilgisayar	169
İlaç	229
Motosiklet	358
Pizza	207
Tabak	52
Buzdolabı	95
Yüzük	88
Terlik	330
Su Bardağı	118
Çorap	136
Domates	105
Cüzdan	95
Kadeh	55
Blender	32
Kolonya Şişesi	331



Şekil 2.5. Test veri setinde nesne tespiti.

2.2. Eğitim Aşaması

Bu çalışmada, nesne tespiti için YOLOv11 derin öğrenme modeli kullanılarak model eğitimi gerçekleştirilmiştir. Veri setinde eğitim için ayrılan görseller üzerinde Python programlama dili ile model eğitilmiştir. Modelin öğrenme süreci, maksimum 100 adım (epoch) olarak belirlenmiş ve erken durdurma (patience) parametresi 10 olarak ayarlanmıştır. 'Patience' parametresi modelin doğrulama setindeki performansının kaç adım boyunca iyileşmemesi durumunda eğitimin durdurulacağını belirler. Bu bağlamda erken durdurma (early stopping), modelin eğitim sürecinde aşırı öğrenme (overfitting) riskini azaltmak için kullanılır [111]. Bu çalışmada belirtilen parametrelerle modelin eğitimi Şekil 2.6'de gösterildiği gibi 70 adımda tamamlanmıştır.



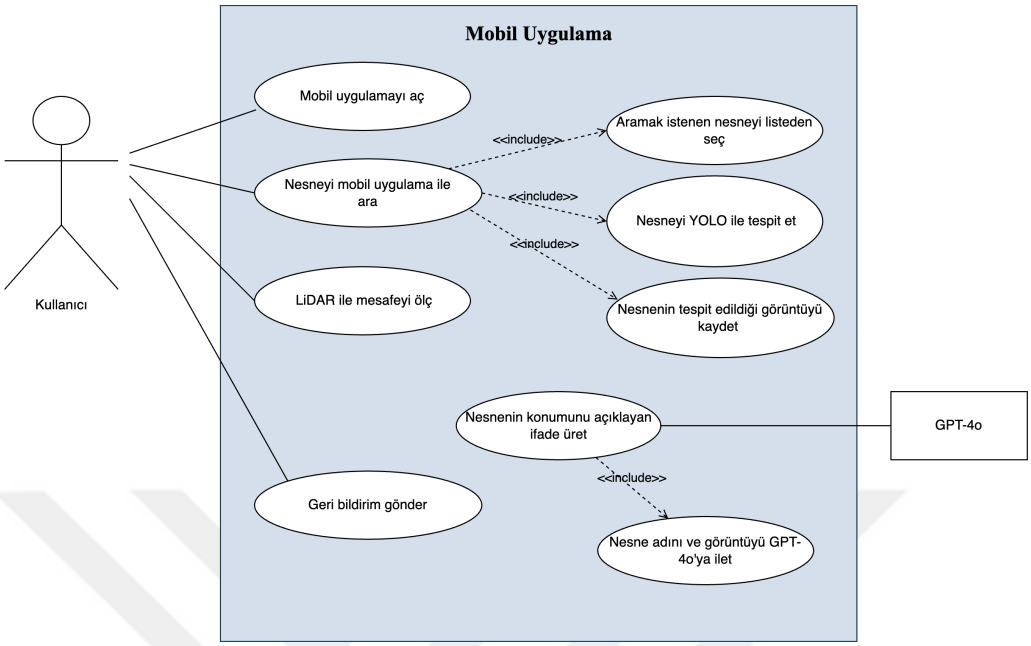
Şekil 2.6. Model eğitimi.

YOLOv11 modelinin eğitimi tamamlandıktan sonra iPhone cihazlarda çalışabilmesi için CoreML formatına dönüştürülmesi gerçekleştirilmiştir. Şekil 2.7'te gösterildiği gibi mlmodel formatına dönüştürmek için coremltools python paket kullanılmıştır.

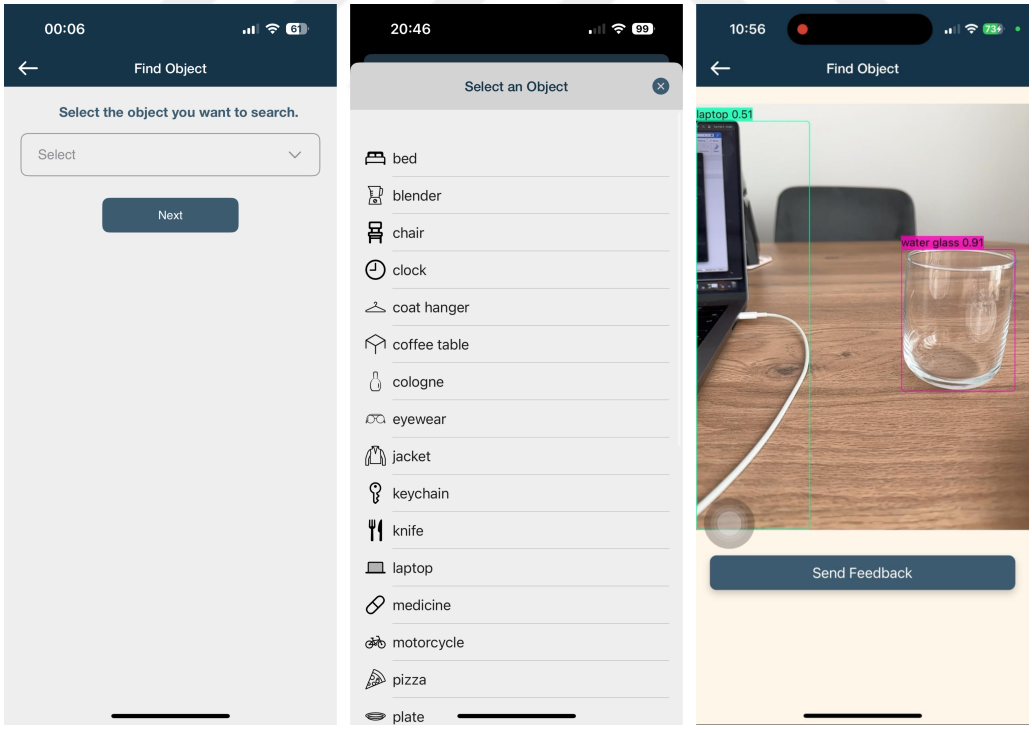
Şekil 2.7. YOLOv11 modelin CoreML formatına dönüştürülmesi.

2.2. Nesne Tespiti ve Mesafe Ölçümü

Nesneleri tespit etme, mesafelerini ölçme ve nesnelerin çevresindeki diğer nesnelere göre konumlarını öğrenme amacıyla iPhone cihazlarında çalışan bir mobil uygulama geliştirilmiştir. Mobil uygulamanın kullanım senaryosu Şekil 2.8'de gösterilmektedir. Uygulama, kullanıcıların aradıkları nesneleri listeden seçerek aramalarına imkân tanımaktadır. Mobil uygulama mobil cihaza entegre edilmiş YOLOv11 derin öğrenme modelini kullanarak belirtilen nesnenin görseldeki konumunu tespit etmektedir. Mobil uygulamanın nesneleri tespit ettiği ekran görüntüsü Şekil 2.9.'da gösterilmektedir. Modelin tespiti başarıyla gerçekleştirmesinin ardından nesnenin fotoğrafı kaydedilmektedir. Ayrıca tespit edilen nesnenin uzaklığı mobil uygulama içerisinde bulunan LiDAR teknolojisi ile ölçülmektedir ve bu mesafe metre (m) cinsinden kullanıcıya bildirilmektedir. LiDAR sensörü iOS işletim sistemli cihazlarda iPhone 12 ve sonrasındaki modellerde kullanılabilmektedir.

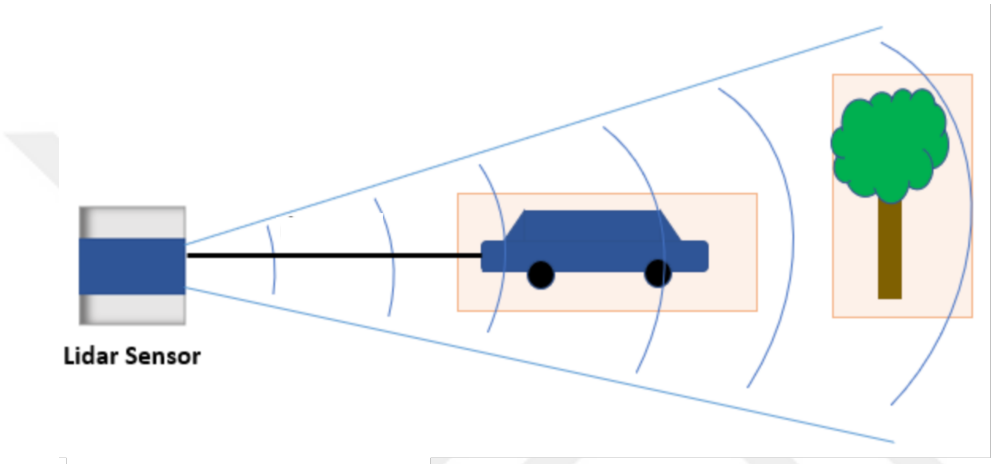


Şekil 2.8. Mobil uygulama kullanım senaryosu.



Şekil 2.9. Mobil uygulama ekran görüntülleri.

LiDAR, lazer darbeleri kullanarak bir hedefe olan mesafeyi ölçen ve çevrenin yüksek çözünürlüklü üç boyutlu modellerini oluşturan bir uzaktan algılama teknolojisidir. LiDAR, lazer ışığını kullanarak çalışan bir sistemdir. Bu sistem, hedefe gönderilen lazer ışığının geri dönmesi için geçen süreyi (uçuş süresi veya time-of-flight) ölçerek mesafeyi hesaplar [112]. Şekil 2.10'de LiDAR ile mesafe ölçümü gösterilmektedir. Bu yöntem, çevresel haritalama, arazi modelleme ve otonom sistemlerde nesne algılama gibi çeşitli alanlarda yaygın olarak kullanılmaktadır.



Şekil 2.10. LiDAR ile mesafe ölçümü [113].

LiDAR sensörü, iPhone cihazlarda lazer ışınları gönderir. Bu ışınlar ortamda bulunan nesnelere çarparak geri yansır. LiDAR sensörü, gönderdiği lazer ışınlarının geri yansımasını algılar. Bu geri dönüş süresi ışığın hedefe gitme ve geri dönme süresidir. Sensör bu süreyi kullanarak hedef nesneye olan mesafeyi hesaplar. Mesafeyi hesaplama formülü eşitlik 2.1'de gösterilmektedir. d: nokta ile sensor arasındaki mesafeyi, c: ışık hızını ve t: uçuş süresini ifade etmektedir. Buradaki bölme işlemi ışığın hem gidiş hem de dönüş süresini kapsaması nedeniyle yapılır.

$$d = \frac{c*t}{2} \qquad (2.1)$$

LiDAR ile nesnenin derinlik bilgisi elde edilebilir ve çevredeki tüm nesneler taranarak bir derinlik haritası oluşturulabilir. LiDAR sensörü mobil cihazlarda ARKit gibi artırılmış gerçeklik platformlarıyla birlikte çalışarak daha doğru ve güvenilir ölçüm sonuçları elde etmeye imkân tanır.

2.3. LLM ile Nesnenin Konumunun Belirlenmesi

Derin öğrenme modeli ile nesne tespiti gerçekleştirildikten sonra kaydedilen görsel ve tespit edilen nesnenin konumunu sorgulayan bir talimat (prompt) GPT-4o'ya iletilmektedir. Bu aşamada LLM'den her bir tespit edilen nesne için çevresindeki diğer nesnelere göre konumunu ifade eden metinsel bir açıklama oluşturulması beklenmektedir. GPT-4o kullanımı sırasında token sayısının artmasını önlemek amacıyla görsellerin boyutları küçültülmüş ve her bir görselin dosya boyutu 100 Kilobayt (KB)'ın altında olacak şekilde ayarlanmıştır. Görseller base64String formatına dönüştürülerek aktarılmıştır.

GPT-4o'ya iletilecek prompt için tez kapsamında bir ön çalışma yapılmıştır. GPT-4o'nun verilen görseldeki nesneleri tanıma ve konumlarını bildirme yeteneğini değerlendirmek amacıyla toplam 100 adet görsel kullanılmıştır.

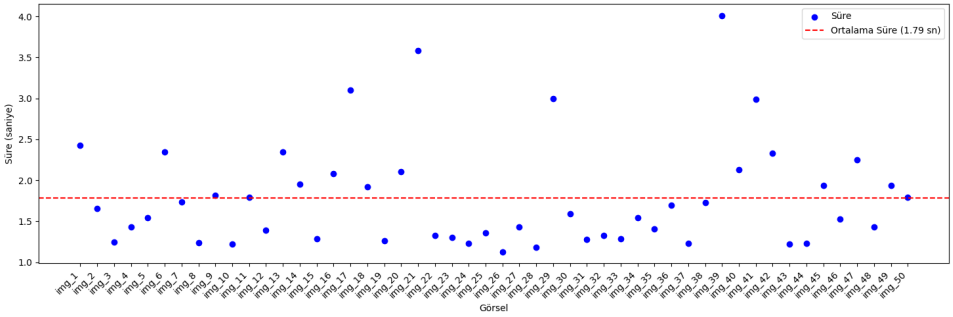
- İlk 50 görselde GPT-4o'ya belirli bir nesnenin konumu, örneğin kitap vb. görseldeki kendi belirleyeceği diğer ana öğeye göre bir cümle ile açıklanması istenmiştir. Kitap nesnesi için kullanılan örnek prompt ve GPT-4o'nun cevabı aşağıda Şekil 2.11'deki gibidir.

```
prompt = "Tell me the position of book relative to the main object
you see? I need only one sentence."
```

```
response = "The book is on the table in front of the couch."
```

Şekil 2.11. Kitap nesnesi için kullanılan prompt ve GPT-4o'nun cevabı.

50 görsel için GPT-4o'nun ortalama cevaplama süresi 1.79 saniye (s) olarak ölçülmüştür. Bu süre, verilen prompt ve görsel verilerin işlenmesi ile ilişkili olup modelin görsel veriyi anlama ve cevap üretme hızını göstermektedir. Görseller ve cevaplama süresine ilişkin dağılım grafiği aşağıdaki Şekil 2.12'de gösterilmektedir.



Şekil 2.12. İlk 50 görsel için GPT-4o'nun cevaplama süreleri.

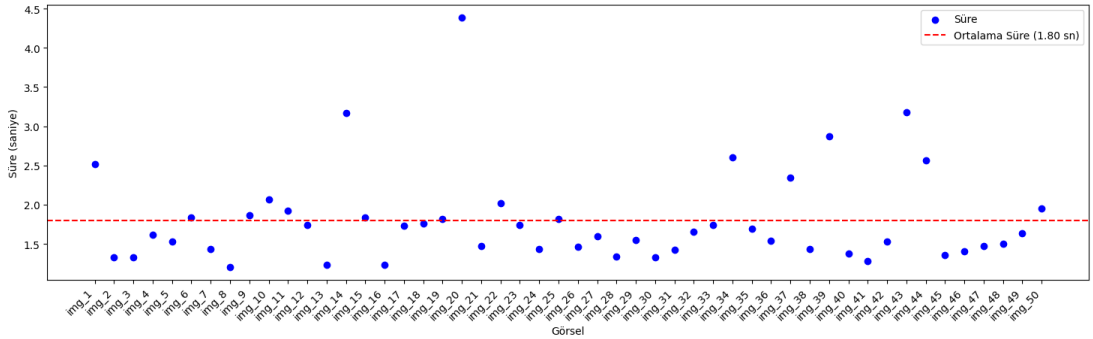
- İkinci 50 görselde ise bir nesnenin konumunun diğer bir nesneye göre ifade edilmesi istenmiştir. Aşağıda Şekil 2.13'de su şişesinin koltuğa göre konumunu ifade etmesini isteyen prompt ve GPT-4o'nun cevabı gösterilmektedir.

```
prompt = "Tell me the position of water bottle according to sofa? I
need only one sentence."
```

```
response = "The water bottle is on the left side of the sofa near the
backrest cushions."
```

Şekil 2.13. Su şişesi ve koltuk için kullanılan prompt ve GPT-4o'nun cevabı.

Bu 50 görsel için GPT-4o'nun ortalama cevaplama süresi 1.80 s olarak hesaplanmıştır. Görseller ve cevaplama süresine ilişkin dağılım grafiği aşağıdaki Şekil 2.14'te gösterilmektedir.



Şekil 2.14. İkinci 50 görsel için GPT-4o'nun cevaplama süreleri.

Bu incelemede GPT-4o modeli, her iki yöntemde de benzer şekilde görsellerdeki nesneleri tanımlama, konumlarını belirleme ve doğal dilde açıklama süreçlerini gerçekleştirmiştir. İlk 50 görselde model, bir nesnenin konumunu kendi belirlediği ana nesneye göre açıklarken, ikinci 50 görselde belirlenen iki nesne arasındaki konumu ifade etmiştir. Her iki durumda da modelin temel işlem aşamaları görseldeki içeriği anlamak ve dilsel bir ifadeye dönüştürmek olduğu için cevaplama süresi büyük ölçüde değişmemiştir. Bu çalışmadaki 100 adet görselin görüntü kalitesi, nesne sayısı gibi faktörleri büyük ölçüde benzerdir. O nedenle farklı yapıdaki test verileri ile daha karmaşık sahneler oluşturularak bu sürenin değişkenliği daha net belirlenebilir. Ancak bu çalışma kapsamında GPT-4o'ya sadece bir nesne belirtilerek konumunu kendi belirleyeceği diğer nesnelere göre ifade etmesi istenmiştir.

2.3.1. GPT-4o ince ayar süreci

LLM'lerin ince ayarı (fine-tuning) önceden eğitilmiş bir dil modelini belirli bir görev ya da veri setine uyarlamak ve geliştirmek için gerçekleştirilen eğitim sürecidir. Bu yaklaşımla birlikte modelin yeteneklerini özelleştirmek ve performansı iyileştirmek amaçlanır [114].

Önceden eğitilmiş modelleri belirli görevler için optimize etmek, dil modelleri ile spesifik uygulamaların ihtiyaçları arasında daha güçlü bir bağ kurar. Modelin yetkinliği, alanın spesifik beklentilerine daha uygun hale gelir [115]. İnce ayar sürecinde modelin çıktısını yönlendirebilmek amacıyla prompt ifadeleri kullanılmaktadır. Prompt yazımı, modelin belirli bir görev veya veri setine uyum yeteneğini doğrudan etkileyen kritik bir faktördür. Bir prompt ifadesindeki küçük değişiklikler dahi olsa modelin performansında önemli farklılıklara yol açabilmektedir [116]. Herhangi bir küçük değişiklik modelin doğru yanıtlar üretme yeteneğini artırabileceği gibi yanıtların yanlış ya da belirsiz olmasına da neden olabilir.

Bu çalışmada, GPT-4o'nun ince ayar süreci nesne konumunu ifade etme görevine özel olarak gerçekleştirilmiştir. Modelin görsel verilerle uyumlu bir şekilde cevaplar üretmesi için 120 adet görsel ve bu görseldeki nesnenin konumunu belirten cümleler ile eğitim veri seti oluşturulmuştur. Bu veri setinde yer alan bir görsel Şekil 2.15'te ve nesne konumunu belirlemeye yönelik prompt jsonl formatında aşağıdaki Şekil 2.16'da gösterilmektedir.

Sistem(system): Modelin bağlamını belirler. Burada modelin bir "yardımcı bot" olduğu ve sorgulanan nesnenin konumunu, görüntüdeki başka bir nesneye göre açıklayacağı tanımlanmıştır.

Kullanıcı (user): Kullanıcının modeli sorguladığı kısım. Örnekte, kullanıcının "Beyaz çiçeklerin konumunu tarif eder misiniz?" şeklindeki isteği yer almaktadır.

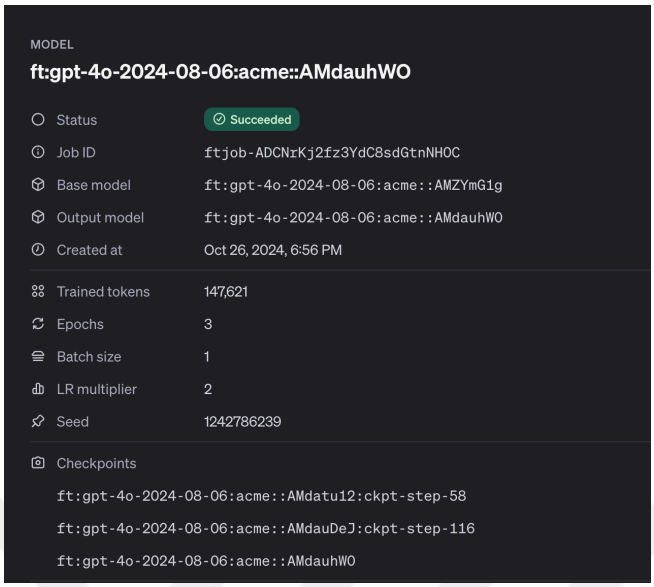
Yardımcı(assistant): Modelin kullanıcının isteğine yanıt verdiği kısım. Örnekte, beyaz çiçeklerin "yemek masasında" bulunduğunu belirttiği bir cevap verilmiştir.

Şekil 2.15. İnce ayar için kullanılan örnek görsel.


```
{
  "messages": [
    {
      "role": "system",
      "content": "When describing the location of an object, you are a helper bot that selects one of the other objects in the
      image and describes the location of the searched object according to it."
    },
    {
      "role": "user",
      "content": "Can you describe the position of the white flowers?",
      "image_url": "resized3/10.jpeg"
    },
    {
      "role": "assistant",
      "content": " The white flowers are on the dining table."
    }
  ]
}
```


Şekil 2.16. İnce ayar için kullanılan örnek JSON.

Şekil 2.17'de GPT-4o modelinin ince ayar süreci tamamlandığındaki detaylar yer almaktadır. Modelin ince ayar süresinde 147621 token işlenmiştir. Model, belirlenen veri seti üzerinde 3 adım boyunca eğitilmiş ve her eğitim adımında (batch size) yalnızca bir veri örneği kullanılmıştır. Bu süreçte öğrenme oranı belirlenen bir çarpan ile artırılarak 2 katına çıkarılmıştır. İnce ayar işlemi sırasında 58. ve 116. adımlarda ara kontrol noktaları kaydedilmiştir. Sonuç olarak ince ayar işlemi başarıyla tamamlanmış ve görev için optimize edilmiş çıktı modeli elde edilmiştir.



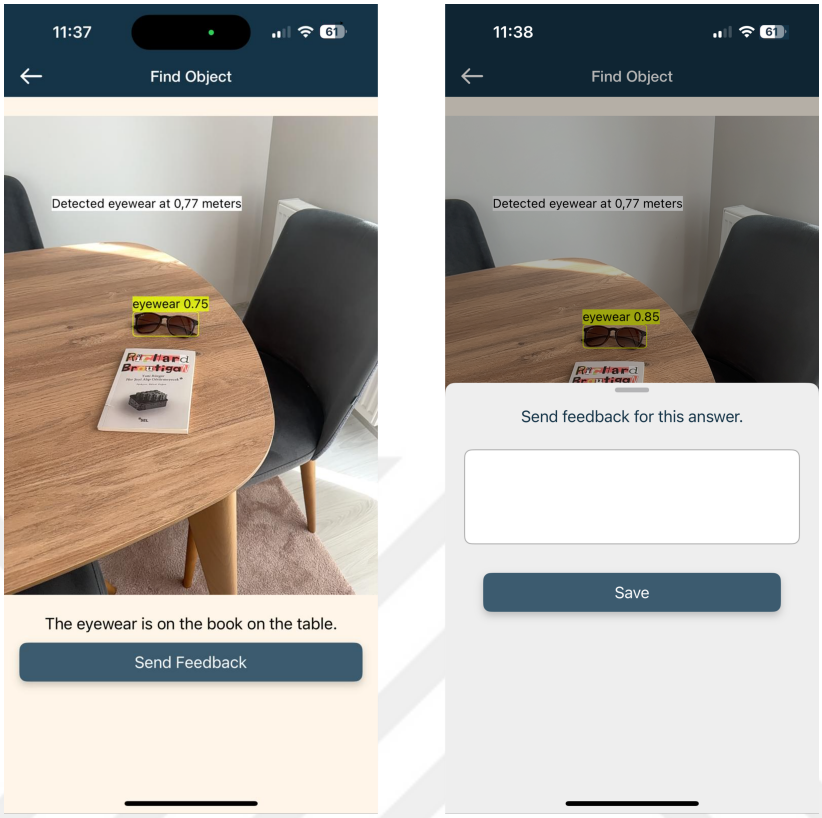
Şekil 2.17. İnce ayar işlemi tamamlanan GPT-4o modelinin eğitim ayrıntıları.

2.3.2. Mobil uygulamada GPT-4o kullanımı

Mobil uygulamada nesne tespiti gerçekleştirildikten sonra tespit anındaki görsel ve aranan nesne ismi GPT-4o'ya iletilmiştir. GPT-4o, bu görsel veriden nesnenin diğer nesneye göre konumunu belirlemiş ve doğal dilde açıklamalar üretmiştir. Örneğin, “Masa, sandalyenin sağında” gibi ifadelerle nesnelerin konum bilgisini mobil uygulama üzerinden yazılı ve sesli olarak kullanıcıya iletilmiştir. Şekil 2.18’deki ekran görüntüsünde kullanıcı, kolonya şişesinin konumunu öğrenmek istemiştir. Bu durumda mobil uygulama üzerinden nesne tespiti gerçekleşmiş ve ardından LiDAR ile uzaklık yaklaşık 2.8 m olarak ölçülmüştür. GPT-4o’nun konum ile ilgili ürettiği cevap ise “Kolonya masanın üzerinde.” şeklinde olmuştur.

Şekil 2.18. Mobil uygulama ile nesne tespiti, uzaklık ölçümü ve GPT cevabı.

Mobil uygulama üzerinden kullanıcı GPT-4o'nun cevabı yanlış ise bir geri bildirim oluşturabilmektedir. Bu geri bildirimler ile insan geri bildirimiyle güçlendirilmiş öğrenme (RLHF) yöntemleri ile modelin doğruluğu ve yanıt tutarlılığı artırılabilir. Kullanıcı geri bildirimleri yanlış veya eksik yanıtları belirlemek ve modelin gelecekte daha doğru tahminler yapmasını sağlamak için kullanılabilir. 2.19'de gözlük nesnesi için uzaklık ölçülmüştür ve GPT-4o'dan bu nesnenin konumunu ifade etmesi istenmiştir. LLM ilgili görseli incelemiştir ve "Gözlük masanın üzerindeki kitabın üzerinde. - The eyewear is on the book on the table." cevabını vermiştir. Aslında "Gözlük masanın üzerinde, kitabın yanında" gibi bir cevap vermesi daha doğru olacaktı. Şekil 2.19'daki ikinci ekran geri bildirim gönderebilmek için oluşturulmuş ekranıdır. Bu durumda kullanıcı mobil uygulama üzerinden geri bildirim gönderebilir ve doğru cevabı ilgili alana yazabilir. Bu geri bildirimler ve görsel Firebase Firestore üzerinde kaydedilmektedir. Bu veriler ile model tekrar eğitilerek doğruluk oranı artırılabilir ve yanıtların tutarlılığı iyileştirilebilir.



Şekil 2.19. Gözlük nesnesi için GPT'nin cevabı ve geri bildirim ekranı.

3. BULGULAR VE TARTIŞMA

Bu bölümde nesne tespiti ve konum verilerinin elde edilmesi için geliştirilen model, mobil uygulama, deneylerin gerçekleştirildiği ortam, deneylerin değerlendirilme ölçütleri, modellerin başarı durumları ve elde edilen sonuçlar değerlendirilmektedir.

3.1. Deney Ortamı

Deneyleri gerçekleştirmek için kullanılan MacBook Pro bilgisayar, 8 çekirdekli CPU, 10 çekirdekli GPU, 16 çekirdekli Neural Engine, 16 GB RAM, 256 GB SSD, M2 Chip'e sahiptir. Kullanılan mobil cihazlar ise iPhone 14 Pro ve iPhone 14 Pro Max'dir. Geliştirme ortamı olarak Visual Studio Code ve Xcode kullanılmıştır. Programlama dili derin öğrenme modeli için Python 3.9.6 tercih edilmiştir. Mobil uygulama geliştirme sürecinde ise iOS platformu için SwiftUI framework'ü ve CoreML teknolojileri kullanılmıştır.

3.2. Performans Metrikleri

Karmaşıklık Matrisi (Confusion Matrix), bir sınıflandırma modelinin başarısını değerlendirmek amacıyla kullanılan ve modelin tahmin ettiği sonuçlar ile gerçek değerlerin arasındaki ilişkiyi gösteren tablodur. Bu matris, sınıflandırma problemlerinde modelin yaptığı doğru ve yanlış tahminleri dört gruba ayırarak görselleştirir. Karmaşıklık matrisi Tablo 3.1'de gösterilmektedir.

- Doğru Pozitif (True Positive, TP): sınıflandırma modelinin pozitif tahmin ettiği ve gerçek değerin de pozitif olduğu durum
- Doğru Negatif (True Negative, TN): sınıflandırma modelin negatif tahmin ettiği ve gerçek değerin de negatif olduğu durum
- Yanlış Negatif (False Negative, FN): sınıflandırma modelin negatif tahmin ettiği ancak gerçek değerin pozitif olduğu durum,
- Yanlış Pozitif (False Positive, FP): sınıflandırma modelin pozitif tahmin ettiği ancak gerçek değerinin negatif olduğu durum [117]

Tablo 3.1. Karmaşıklık matrisi.

	Pozitif Tahmin	Negatif Tahmin
Pozitif Gerçek	Doğru Pozitif (DP)	Yanlış Negatif (YN)
Negatif Gerçek	Yanlış Pozitif (YP)	Doğru Negatif (DN)

Bu çalışmada, derin öğrenme modelinin performansını değerlendirmek için karşılaştırma matrisi kullanılarak elde edilen doğruluk (accuracy), kesinlik (precision), duyarlılık veya geri çağırma (recall), F1 puanı ve ortalama hassasiyet (Mean Average Precision, mAP) değerlendirme metrikleri kullanılmaktadır.

Doğruluk, modelin doğru tahmin ettiği örneklerin toplam örneklere oranıdır. Eşitlik 3.1 de gösterilmektedir. Eşitliklerde DP doğru pozitif, DN doğru negatif, YP yanlış pozitif, YN yanlış negatifi ifade etmektedir.

$$\text{Doğruluk} = \frac{DP+DN}{DP+DN+YP+YN} \quad (3.1)$$

Kesinlik, modelin doğru tahmin ettiği pozitif örneklerin, yaptığı toplam pozitif tahminlere oranıdır. Eşitlik 3.2'de gösterilmektedir.

$$\text{Kesinlik} = \frac{DP}{DP+YP} \quad (3.2)$$

Duyarlılık, modelin tüm gerçek pozitif örneklerden kaç tanesini doğru bir şekilde tahmin ettiğini ölçen bir metriktir. Doğru olarak tahmin edilen pozitif örneklerin, veri setindeki gerçek pozitif örneklere oranıdır. Modelin gerçek pozitifleri ne kadar iyi yakaladığını ifade eder. Eşitlik 3.3 de gösterilmektedir.

$$\text{Duyarlılık} = \frac{DP}{DP+YN} \tag{3.3}$$

F1 puanı, modelin performansını değerlendirmek için kullanılan ve kesinlik (precision) ile geri çağırma (recall) metriklerinin harmonik ortalamasını hesaplayan bir ölçüttür. Dengeli bir performans ölçümü sağlar. Eşitlik 3.4 de gösterilmektedir.

$$\text{F1} = 2 * \frac{Kesinik * Geri\ Çağırma}{Kesinik + Geri\ Çağırma} \tag{3.4}$$

Kesişim Birleşme Oranı (IoU- Intersection over Union), modelin tahmin ettiği sınır kutusu ile gerçek sınır kutusu arasındaki örtüşme oranını ölçen bir metriktir. Eşitlik 3.4'de gösterilmektedir. IoU değeri 0 ile 1 arasında değişir ve IoU ≥ 0.5 olduğunda tahmin genellikle doğru kabul edilir.

$$\text{IoU} = \frac{Gerçek\ Kutu\ \cap Tahmin\ Edilen\ Kutu}{Gerçek\ Kutu\ \cup Tahmin\ Edilen\ Kutu} \tag{3.5}$$

Ortalama hassasiyet (mAP), her sınıfa ait ortalama kesinlik değeri (AP) hesaplanır ve bu değerlerin sınıflar arasında aritmetik ortalaması alınarak genel performans ölçüsü elde edilir. Eşitlik 3.6'da gösterilmektedir.

$$\text{mAP} = \frac{1}{N} \sum_{i=1}^{N} (AP_i) \tag{3.6}$$

Nesne tespiti gerçekleştirdikten sonra GPT-4o'nın görsellerden nesne konumuna ait cevaplarının başarısı ölçülmesi için ROUGE (Recall-Oriented Understudy for Gisting Evaluation) değerleri hesaplanmıştır. ROUGE doğal dil işleme alanında yaygın kullanılan bir değerlendirme metriğidir. Modelin ürettiği metnin doğruluğunu ve tutarlılığını ölçmek için insanlar tarafında oluşturulan referans metinle olan benzerliği değerlendirir. Ölçümler, değerlendirilen metin ile referans metin arasında n-gram, kelime dizileri ve kelime çiftleri gibi kaç tane eşleşen birimin bulunduğunu yakalar [118]. ROUGE-1 metriği, tek kelimelerin (unigrams) örtüşme oranını ölçer. Her kelimenin modelin ürettiği metin ile referans metninde kaç kez geçtiğini ve bu kelimelerin ne kadar örtüştüğünü ifade eder. ROUGE-2 metriği, iki kelimeden oluşan grupların (bigrams) örtüşme oranını ölçer.

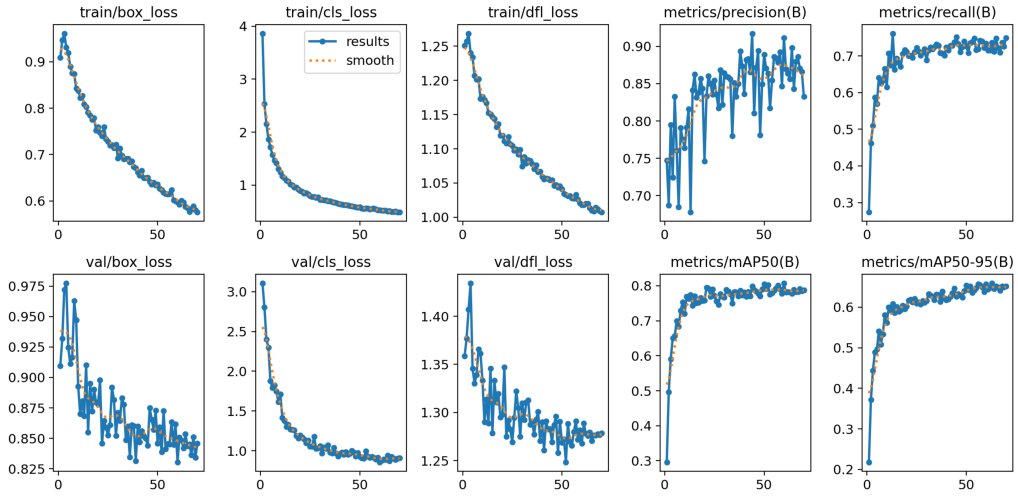
3.3. Deney Sonuçları

Bu bölümde derin öğrenme modeli ve LLM'in performansları değerlendirilmektedir.

3.3.1. Nesne tespiti deney sonuçları

2. bölümde anlatılan YOLOv11 derin öğrenme modeli doğruluk, kesinlik (precision), duyarlılık ve F1 puanı, mAP gibi metrikler kullanılarak değerlendirilmiştir. Şekil 3.1'de modelinin sonuç grafikleri yer almaktadır.

Eğitim ve doğrulama için kutu kaybı (box loss) düzenli bir şekilde azalmıştır. Bu durum eğitim süreci boyunca modelin nesne konumlarını giderek daha doğru tahmin ettiğini ve konumlandırma hatalarının azaldığını göstermektedir. Eğitim ve doğrulama aşamalarında sınıflandırma kaybının (cls_loss) şekildeki gibi belirgin bir şekilde azalması da modelin nesneleri doğru sınıflandırmayı başarılı bir şekilde öğrendiğini göstermektedir.



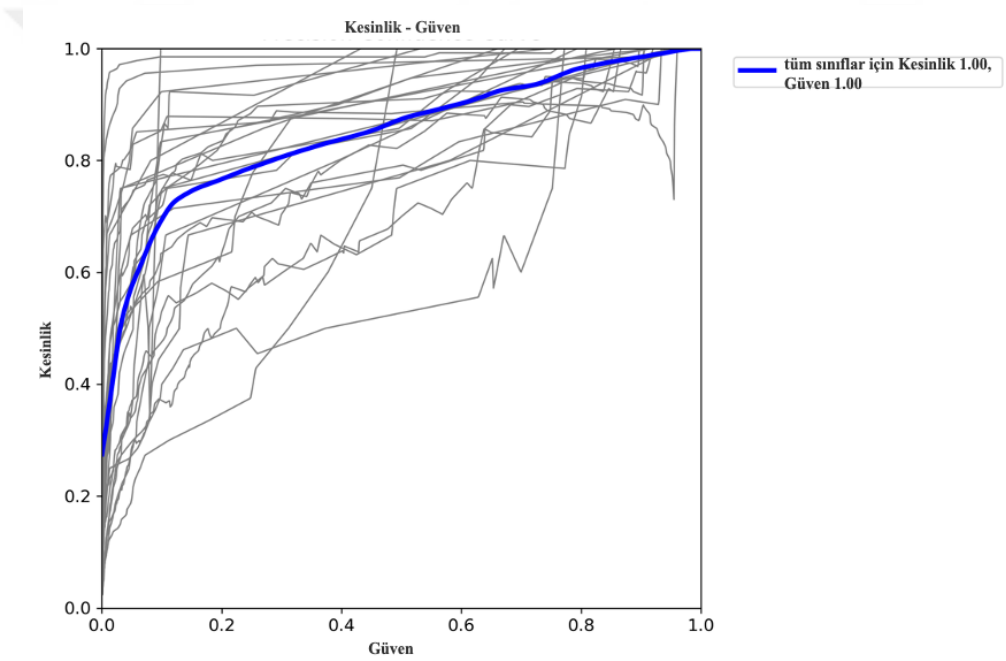
Şekil 3.1. YOLOv11 eğitim sonuçları.

Kesinlik grafiği genellikle artan bir eğilim göstermektedir ancak dalgalanmalar modelin belirli sınıflarda düşük güvenilirlikle tahminler yaptığını göstermektedir. Duyarlılık (recall) grafiğinde ise düzenli bir artış gözlemlenmekte ve belli bir noktadan sonra stabil hale gelmektedir. Bu durum modelin giderek daha fazla doğru tahmin yaptığını göstermektedir.

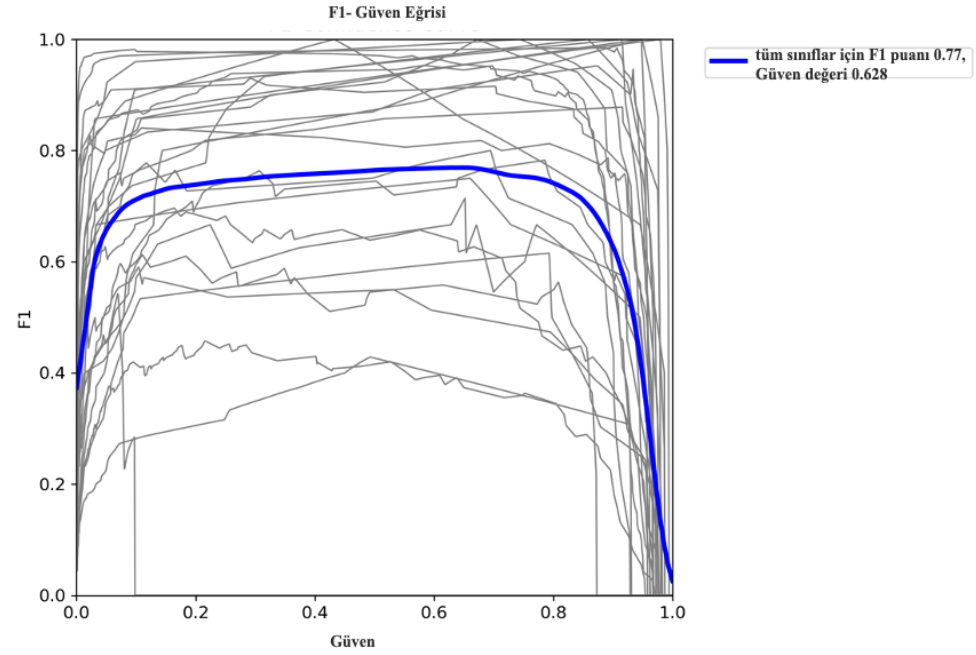
Şekil 3.2'deki Kesinlik-Güven (Precision-Confidence) grafiği incelendiğinde, güven seviyesi arttıkça kesinlik de yükselmektedir. Bu durum modelin yüksek güven değerlerinde daha doğru tahminler yaptığını gösterir.

Derin öğrenme modelinin F1 puanı 0,77 olarak tespit edilmiştir ve sonuç grafiği Şekil 3.3'de gösterilmiştir.

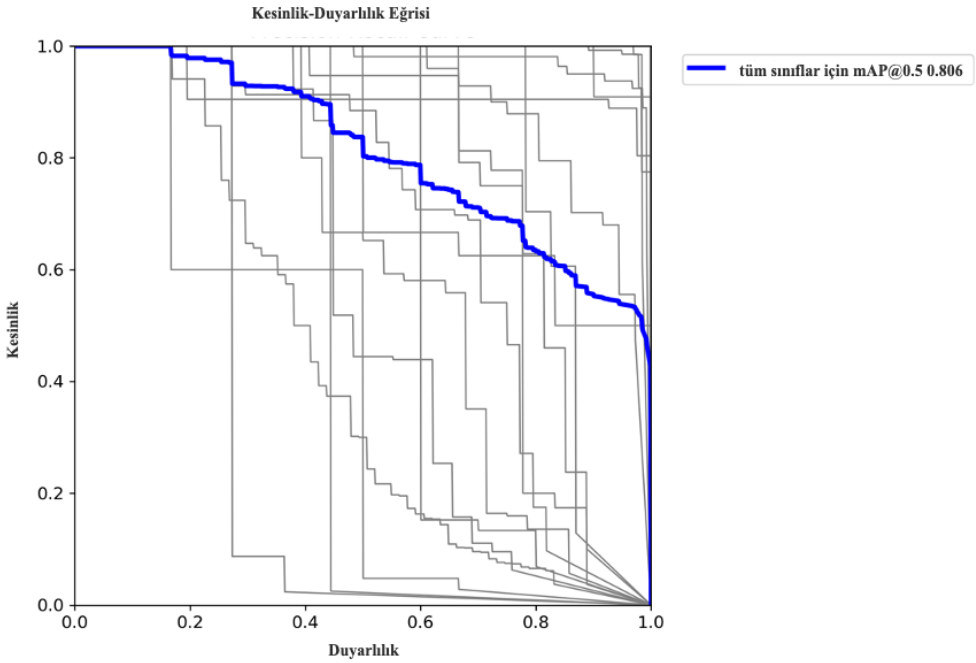
mAP@0.5, IoU eşik değeri 0.5 olarak alındığında hesaplanan ortalama hassasiyet değeridir. Şekil 3.4'te Kesinlik-Duyarlılık eğrisinde gösterildiği gibi mAP@0.5 değerini 0.806 olarak hesaplanmıştır. Modelin ortalama hassasiyeti mAP %80.6' dir.



Şekil 3.2. Kesinlik-güven grafiği.



Şekil 3.3. F1-güven eğrisi.

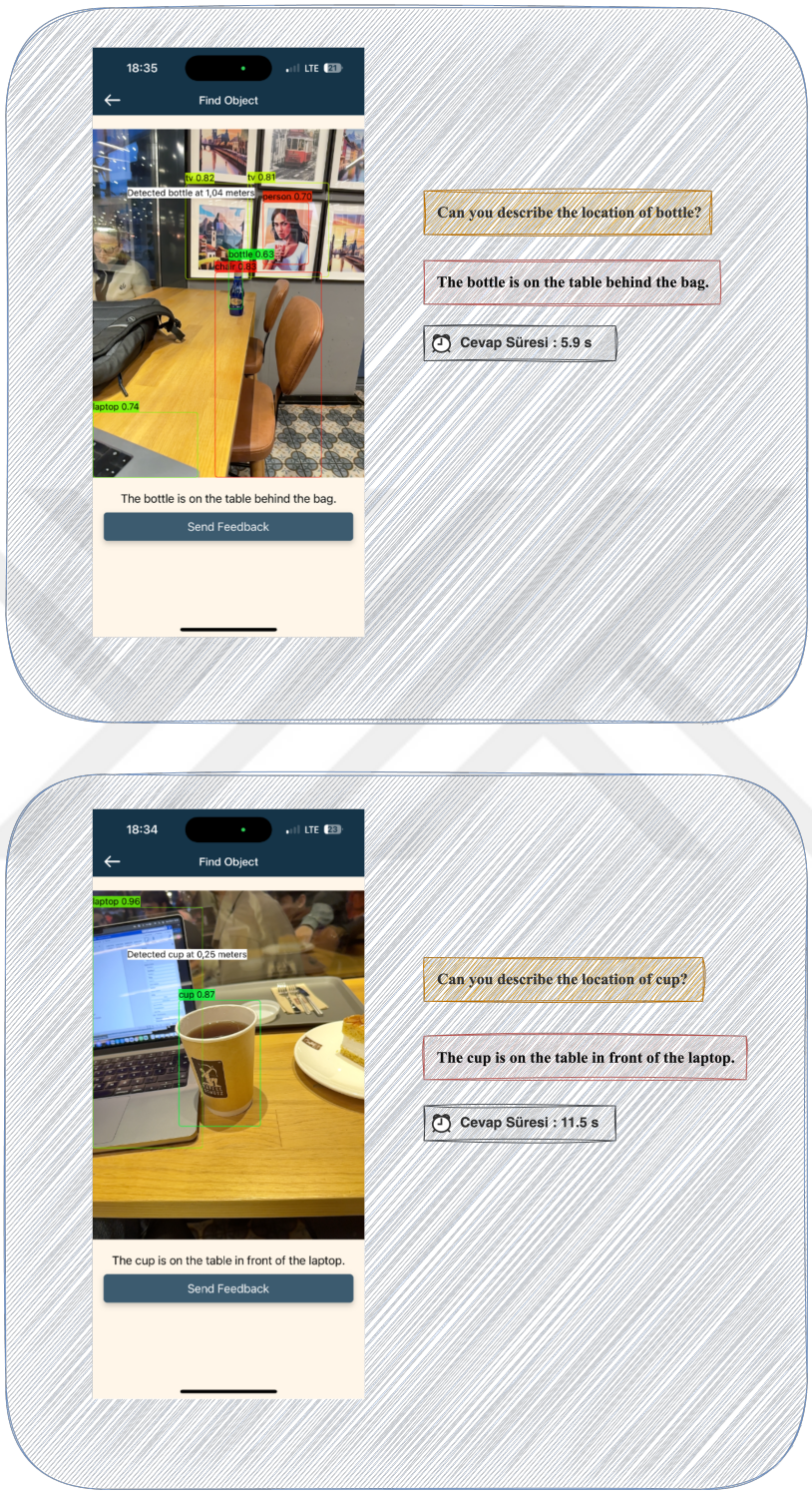


Şekil 3.4. Kesinlik-duyarlılık eğrisi.

3.3.2. LLM ile konumlarının analizi deney sonuçları

GPT-4o'nun nesne konumlarını içeren doğal dilde ifade üretme performansını değerlendirmek amacıyla mobil uygulama üzerinden tespit edilen nesne görselleri ve internetten temin edilen ev görsellerinden oluşan bir veri seti kullanılmıştır [119]. GPT-4o'ya 60 adet test verisi olarak görseller ve bu görsellerdeki konumu belirlenmesi istenen nesnelerin isimleri iletilerek modelin bu nesnelerin konumlarını doğru bir şekilde ifade etmesi beklenmiştir. Şekil 3.5'te GPT-4o'ya iletilen 2 adet görsel, prompt ve bu görseller için modelin cevapları, cevaplama süresi gösterilmektedir. Şekil 3.6'de ise mobil uygulama üzerinden tespit edilen nesneler için GPT-4o'ya iletilen görsel, prompt, GPT-4o'nun cevapları, cevaplama süresi gösterilmektedir.

Şekil 3.5. Test görselleri için GPT-4o cevapları.



Şekil 3.6. Mobil uygulamada GPT-4o cevapları.

Test veri seti kullanılarak gerçekleştirilen deneyler sonucunda, GPT-4o modelinin ortalama yanıt süresi 5.15 s olarak ölçülmüştür. Model, iletilen görsellerde yer alan nesnelerin konumlarını açık ve anlaşılır bir şekilde ifade etmektedir. İnce ayar sürecinde kullanılan yönlendirmelere benzer şekilde nesne konumlarını "sağda", "solda" veya "duvarda asılı" gibi ifadelerle açıklamaktadır. Ayrıca bir nesnenin iki farklı eşya arasında konumlandığı durumlarda ilgili nesneleri belirterek "arasında" ifadesiyle konumu tanımladığı gözlemlenmiştir. Modelin verdiği yanıtlar incelendiğinde, "Kitap masanın üzerinde." gibi kısa ve doğrudan ifadeler hem de "Gardırop, soldaki çalışma masası ile sağdaki beyaz şifonyerin arasında, duvarın önünde yer alıyor." şeklinde nesnenin hangi kısımda olduğu veya çevresindeki diğer nesnelerle ilişkisini ifade eden cümleler kurmuştur.

Tablo 3.2'de gösterildiği gibi 60 adet görsel ile testler gerçekleştirilmiştir. Bu testte sadece 2 görsel için GPT-4o tarafından yanlış cevaplar üretilmiştir. Modelin yaptığı hatalar sırasıyla, konumu sorulan eşya için kendi belirlediği nesne ya da eşyaya göre olan konumunu önünde olduğunu söylemek yerine arkasında olduğunu söylemek, yanında olduğunu söylemek yerine üstünde olduğunu söylemek şeklinde olmuştur. Bu örneklerin dışında genel olarak alt-üst ön-arka, sağda, solda, yanında gibi yönleri diğer görsellerde doğru şekilde ifade etmiştir.

Tablo 3.2. GPT-4o test sonuçları.

Soru Sayısı	60
Doğru Yanıt Sayısı	58
Yanlış Yanıt Sayısı	2
Ortalama Yanıt Süresi (s)	5.15

Modelin test görselleri için ürettiği yanıtlar, metin değerlendirme metriği olan ROUGE ile analiz edilmiştir. Değerlendirme sürecinde referans metin olarak tarafımca oluşturulan ve doğruluğu kabul edilen yanıtlar kullanılmıştır. Tablo 3.3'te gösterilen sonuçlara göre modelin ROUGE-1 skoru 0.75, ROUGE-2 skoru 0.61 ve ROUGE-L skoru 0.71 olarak hesaplanmıştır.

Tablo 3.3. ROUGE değerleri.

	ROUGE-1	ROUGE-2	ROUGE-L
GPT-4o'nun Yanıtları	0.75	0.61	0.71

SONUÇLAR

Ev içinde kullanılan nesneleri tanıma, uzaklıklarını ölçme ve konumlarını doğal dille ifade etme fikriyle başlayan bu tez çalışmasında, nesne tespiti için YOLOv11 derin öğrenme modeli kullanılarak model eğitimi gerçekleştirilmiştir. CoreML formatına dönüştürme uyumluluğu ve mobil cihazlarda önceki YOLO versiyonlarına oranla daha verimli çalışması gibi avantajları nedeniyle YOLOv11 tercih edilmiştir. Derin öğrenme modeli için veri seti özgün olarak oluşturulmuştur. Model, CoreML kullanılarak mobil uygulamada kullanılabilir hale getirilmiştir. Ayrıca nesne tespitinin ardından nesnenin konum bilgisinin doğal dilde ifade edilmesi için GPT-4o'ya ince ayar işlemi gerçekleştirilmiştir. İnce ayar işlemi ile GPT-4o istenen ifadeleri daha kısa ve doğru bir biçimde üretebilecek şekilde optimize edilmiştir. Bu aşamada modelin nesneler arasındaki mekânsal ilişkileri daha iyi anlaması ve kullanıcıya en anlamlı şekilde sunması hedeflenmiştir. Tez kapsamında nesne tespiti, mesafe ölçümü ve konum bilgisinin doğal dilde sunulmasını sağlayan bir mobil uygulama geliştirilmiştir. Mobil uygulamada kullanıcı geri bildirimlerini alabilecek bir yapı eklenmiştir. Derin öğrenme modelinin performansı doğruluk, kesinlik, hassasiyet, F1 skoru ve mAP gibi metrikler kullanılarak değerlendirilmiş, değerlendirme sonuçları analiz edilmiştir. LLM tabanlı sistemin performansı ise nesne konumu ifadelerinin doğruluğu, tutarlılığı ve hızı açısından değerlendirilmiştir.

Derin öğrenme modelinin F1 puanı 0.77 olarak hesaplanmıştır. Bu değer, modelin hem kesinlik hem de hassasiyet açısından dengeli bir performans sergilediğini göstermektedir. 0.77'lik F1 skoru ile, modelin doğru nesne tespiti yapma yeteneğinin yüksek olduğunu ancak hala iyileştirme yapılabilecek alanların bulunduğu sonucuna varılmaktadır. Derin öğrenme modelinin mAP değeri 0.806 olarak elde edilmiştir. Bu sonuç modelin nesne tespiti görevinde yüksek doğruluk elde ettiği ve genel performansının başarılı olduğunu göstermektedir.

İnce ayar işlemi gerçekleştirilen GPT-4o, iletilen görsellerdeki nesnelerin konumunu açıklayan cevaplar üretmiştir. Bunu yaparken nesnelere ve konumlara dair doğru ve anlamlı ilişkiler kurarak doğal dilde ifadeler oluşturmuştur. GPT-4o, test edilen

görsellerden %96.67’ sinde nesne konumlarını doğru ve tutarlı bir şekilde belirleyerek kullanıcıya anlaşılır açıklamalar sunmuştur. Test görselleri için üretilen cevaplarda ROUGE-1 skoru 0.75, ROUGE-2 skoru 0.61 ve ROUGE-L skoru 0.71 olarak hesaplanmıştır.

Mobil uygulama ile nesne tespiti, uzaklık hesaplaması ve nesne konumunun doğal dilde ifade edilmesi başarıyla gerçekleştirilmiştir. Derin öğrenme modeli ve büyük dil modeli mobil uygulama içerisinde entegre edilerek birlikte çalışan bir sistem oluşturulmuştur. Gelecek çalışmalarda, modelin doğruluğunu artırmak ve farklı senaryolara uyum sağlayabilmesini sağlamak amacıyla ek veri setleriyle eğitimi gerçekleştirilebilir. Ayrıca GPT-4o'nun yanıtlarına yönelik geri bildirimler doğrultusunda ince ayar işlemi yeniden uygulanabilir ve modelin çıktılarının doğruluk ve tutarlılığı artırılabilir.

KAYNAKLAR

1. Szeliski, R., "*Computer vision: algorithms and applications"*, 2022: Springer Nature.
2. Dong, X. and M.L. Cappuccio, "*Applications of computer vision in autonomous vehicles: Methods, challenges and future directions",* arXiv preprint arXiv:2311.09093, 2023.
3. Shamshirband, S., M. Fathi, A. Dehzangi, A.T. Chronopoulos, and H. Alinejad-Rokny, "*A review on deep learning approaches in healthcare systems: Taxonomies, challenges, and open issues",* Journal of Biomedical Informatics, 2021, **113**: p. 103627.
4. Jindal, V., S. Narayan Singh, and S. Suvra Khan, "*Facial Recognition with Computer Vision"*, in *Machine Intelligence and Data Science Applications: Proceedings of MIDAS 2021*, 2022, Springer, p. 313-330.
5. Pandey, S.K. and A.K. Bhandari, "*YOLOv7 for brain tumour detection using morphological transfer learning model",* Neural Computing and Applications, 2024, **36**(32): p. 20321-20340.
6. Kim, J.-H., N. Kim, and C.S. Won, "*High-speed drone detection based on yolo-v8"*, in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023: IEEE.
7. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi, "*You only look once: Unified, real-time object detection"*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.
8. Abed, A.A., A. Al-Ibadi, and I.A. Abed, "*Real-time Multiple Face Mask And Fever Detection Using YOLOv3 And TensorFlow Lite Platforms",* Bulletin of Electrical Engineering and Informatics, 2023, **12**(2): p. 922-929.
9. Kumar, S., R. Ratan, and J. Desai, "*Cotton disease detection using tensorflow machine learning technique",* Advances in Multimedia, 2022, **2022**.
10. Mahi, A.B.S., F.S. Eshita, and T. Helaly, "*An Automated System for Wrong-Way Vehicle Detection using YOLO and DeepSORT"*, in *2023 5th International Conference on Sustainable Technologies for Industry 5.0 (STI)*, 2023: IEEE.
11. Aung, N.H.H., P. Sangwongngam, R. Jintamethasawat, S. Shah, and L. Wuttisittikulkij, "*A review of lidar-based 3d object detection via deep learning approaches towards robust connected and autonomous vehicles",* IEEE Transactions on Intelligent Vehicles, 2024.
12. Yang, T., Y. Li, C. Zhao, D. Yao, G. Chen, L. Sun, T. Krajnik, and Z. Yan, "*3D ToF LiDAR in mobile robotics: A review",* arXiv preprint arXiv:2202.11025, 2022.
13. Xu, X., L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo, "*A review of multi-sensor fusion slam systems based on 3D LIDAR",* Remote Sensing, 2022, **14**(12): p. 2835.
14. Khurana, D., A. Koli, K. Khatter, and S. Singh, "*Natural language processing: state of the art, current trends and challenges",* Multimedia tools and applications, 2023, **82**(3): p. 3713-3744.
15. Al-Hassan, A. and H. Al-Dossari, "*Detection of hate speech in social networks: a survey on multilingual corpus"*, in *6th international conference on computer science and information technology*, 2019: ACM.
16. Raffel, C., N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W.

Li, and P.J. Liu, "*Exploring the limits of transfer learning with a unified text-to-text transformer",* Journal of machine learning research, 2020, **21**(140): p. 1-67.
17. Zaheer, M., G. Guruganesh, K.A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, and L. Yang, "*Big bird: Transformers for longer sequences",* Advances in neural information processing systems, 2020, **33**: p. 17283-17297.
18. Chowdhary, K. and K. Chowdhary, "*Natural language processing",* Fundamentals of artificial intelligence, 2020: p. 603-649.
19. Thakur, A., L. Ahuja, R. Vashisth, and R. Simon, "*NLP & AI speech recognition: an analytical review"*, in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, 2023: IEEE.
20. Agarwal, A., S. Maiya, and S. Aggarwal, "*Evaluating Empathetic Chatbots in Customer Service Settings. arXiv"*, 2021.
21. Eguia, H., C.L. Sánchez-Bocanegra, F. Vinciarelli, F. Alvarez-Lopez, and F. Saigí-Rubió, "*Clinical decision support and natural language processing in medicine: Systematic literature review",* Journal of Medical Internet Research, 2024, **26**: p. e55315.
22. Işıkdemir, Y.E., "*Nlp Transformers: Analysis Of LLMs And Traditional Approaches For Enhanced Text Summarization ",* Eskişehir Osmangazi Üniversitesi Mühendislik ve Mimarlık Fakültesi Dergisi, 2024, **32**(1): p. 1140-1151.
23. Zhang, H., P.S. Yu, and J. Zhang, "*A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models",* arXiv preprint arXiv:2406.11289, 2024.
24. Turan, S.C., K. Yildiz, and B. Büyüktanir, "*Comparison of LDA, NMF and BERTopic Topic Modeling Techniques on Amazon Product Review Dataset: A Case Study"*, in *International Conference on Computing, Intelligence and Data Analytics*, 2023: Springer.
25. Zhang, Y., "*Dialogpt: Large-Scale generative pre-training for conversational response generation",* arXiv preprint arXiv:1911.00536, 2019.
26. Brown, T., B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, and A. Askell, "*Language models are few-shot learners",* Advances in neural information processing systems, 2020, **33**: p. 1877-1901.
27. Achiam, J., S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F.L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, and S. Anadkat, "*Gpt-4 technical report",* arXiv preprint arXiv:2303.08774, 2023.
28. Chen, M., J. Tworek, H. Jun, Q. Yuan, H.P.D.O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, and G. Brockman, "*Evaluating large language models trained on code",* arXiv preprint arXiv:2107.03374, 2021.
29. Kök, İ., O. Demirci, and S. Özdemir, "*When IoT Meet LLMs: Applications and Challenges"*, in *2024 IEEE International Conference on Big Data (BigData)*, 2024: IEEE.
30. Bommasani, R., D.A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M.S. Bernstein, J. Bohg, A. Bosselut, and E. Brunskill, "*On the opportunities and risks of foundation models",* arXiv preprint arXiv:2108.07258, 2021.
31. El Naqa, I. and M.J. Murphy, "*What is machine learning?"*, 2015: Springer.
32. Roy, S., T. Meena, and S.-J. Lim, "*Demystifying supervised learning in*

healthcare 4.0: A new reality of transforming diagnostic medicine", Diagnostics, 2022, **12**(10): p. 2549.
33. Gumbs, A.A., V. Grasso, N. Bourdel, R. Croner, G. Spolverato, I. Frigerio, A. Illanes, M. Abu Hilal, A. Park, and E. Elyan, "*The advances in computer vision that are enabling more autonomous actions in surgery: a systematic review of the literature",* Sensors, 2022, **22**(13): p. 4918.
34. Chapelle, O., B. Scholkopf, and A. Zien, "*Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]",* IEEE Transactions on Neural Networks, 2009, **20**(3): p. 542-542.
35. Miyato, T., A.M. Dai, and I. Goodfellow, "*Adversarial training methods for semi-supervised text classification",* arXiv preprint arXiv:1605.07725, 2016.
36. Cheplygina, V., M. De Bruijne, and J.P. Pluim, "*Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis",* Medical image analysis, 2019, **54**: p. 280-296.
37. MacQueen, J., "*Some methods for classification and analysis of multivariate observations"*, in *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability/University of California Press*, 1967.
38. Murtagh, F. and P. Contreras, "*Algorithms for hierarchical clustering: an overview, II",* Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2017, **7**(6): p. e1219.
39. Xie, T., R. Liu, and Z. Wei, "*Improvement of the fast clustering algorithm improved by-means in the big data",* Applied Mathematics and Nonlinear Sciences, 2020, **5**(1): p. 1-10.
40. Ikotun, A.M., A.E. Ezugwu, L. Abualigah, B. Abuhaija, and J. Heming, "*K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data",* Information Sciences, 2023, **622**: p. 178-210.
41. Jolliffe, I.T., "*Principal component analysis for special types of data"*, 2002: Springer.
42. Mnih, V., K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, and G. Ostrovski, "*Human-level control through deep reinforcement learning",* nature, 2015, **518**(7540): p. 529-533.
43. Ribeiro, C., "*Reinforcement learning agents",* Artificial intelligence review, 2002, **17**: p. 223-250.
44. Moody, J. and M. Saffell, "*Learning to trade via direct reinforcement",* IEEE transactions on neural Networks, 2001, **12**(4): p. 875-889.
45. Matsuzaka, Y. and R. Yashiro, "*AI-based computer vision techniques and expert systems",* AI, 2023, **4**(1): p. 289-302.
46. Uppal, S., S. Raheja, and N.R. Das, "*Fire detection alarm system using deep learning"*, in *2023 13th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 2023: IEEE.
47. Zhao, E., Y. Liu, J. Zhang, and Y. Tian, "*Forest fire smoke recognition based on anchor box adaptive generation method",* Electronics, 2021, **10**(5): p. 566.
48. Ahn, Y., H. Choi, and B.S. Kim, "*Development of early fire detection model for buildings using computer vision-based CCTV",* Journal of Building Engineering, 2023, **65**: p. 105647.
49. Wu, F., Y. Chen, and D. Han, "*Development countermeasures of college english education based on deep learning and artificial intelligence",* Mobile Information Systems, 2022, **2022**(1): p. 8389800.

50. Yavuz, S. and M. Deveci, "*İstatiksel normalizasyon tekniklerinin yapay sinir ağin performansina etkisi",* Erciyes Üniversitesi İktisadi ve İdari Bilimler Fakültesi Dergisi, 2012(40): p. 167-187.
51. Nwadiugwu, M.C., "*Neural networks, artificial intelligence and the computational brain",* arXiv preprint arXiv:2101.08635, 2020.
52. Türkoğlu, M., K. Hanbay, I.S. Sivrikaya, and D. Hanbay, "*Derin Evrişimsel Sinir Ağı Kullanılarak Kayısı Hastalıklarının Sınıflandırılması",* Bitlis Eren Üniversitesi Fen Bilimleri Dergisi, 2020, **9**(1): p. 334-345.
53. Akhtar, N. and U. Ragavendran, "*Interpretation of intelligence in CNN-pooling processes: a methodological survey",* Neural computing and applications, 2020, **32**(3): p. 879-898.
54. Ting, A., J. Law, A. Lele, Y. Fang, and A. Raychowdhury, "*A Comparison of CNNs and LSTMs for EEG Signal Classification"*, in *2022 Opportunity Research Scholars Symposium (ORSS)*, 2022: IEEE.
55. Fang, W., L. Wang, and P. Ren, "*Tinier-YOLO: A real-time object detection method for constrained environments",* Ieee Access, 2019, **8**: p. 1935-1944.
56. Sikhin, V., S.S. Subramanian, and R. Sreelekshmi, "*Examination on Fire Detection Methods using Computer Vision"*, in *2022 International Conference on Automation, Computing and Renewable Systems (ICACRS)*, 2022: IEEE.
57. Redmon, J. and A. Farhadi, "*YOLO9000: better, faster, stronger"*, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.
58. Redmon, J., "*Yolov3: An incremental improvement",* arXiv preprint arXiv:1804.02767, 2018.
59. Bochkovskiy, A., C.-Y. Wang, and H.-Y.M. Liao, "*Yolov4: Optimal speed and accuracy of object detection",* arXiv preprint arXiv:2004.10934, 2020.
60. Jocher, G., A. Stoken, J. Borovec, L. Changyu, A. Hogan, L. Diaconu, F. Ingham, J. Poznanski, J. Fang, and L. Yu, "*ultralytics/yolov5: v3. 1-bug fixes and performance improvements",* Zenodo, 2020.
61. Li, C., L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, and W. Nie, "*YOLOv6: A single-stage object detection framework for industrial applications",* arXiv preprint arXiv:2209.02976, 2022.
62. Wang, C.-Y., A. Bochkovskiy, and H.-Y.M. Liao, "*YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors"*, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023.
63. Jocher, G., A. Chaurasia, and J. Qiu, "*Ultralytics YOLOv8"*, https://github.com/ultralytics/ultralytics, (18.05.2025).
64. Wang, C.-Y., I.-H. Yeh, and H.-Y. Mark Liao, "*Yolov9: Learning what you want to learn using programmable gradient information"*, in *European Conference on Computer Vision*, 2025: Springer.
65. Wang, A., H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "*Yolov10: Real-time end-to-end object detection",* arXiv preprint arXiv:2405.14458, 2024.
66. Terven, J., D.-M. Córdova-Esparza, and J.-A. Romero-González, "*A comprehensive review of yolo architectures in computer vision: From yolov1 to yolov8 and yolo-nas",* Machine Learning and Knowledge Extraction, 2023, **5**(4): p. 1680-1716.
67. Jiang, P., D. Ergu, F. Liu, Y. Cai, and B. Ma, "*A Review of Yolo algorithm developments",* Procedia computer science, 2022, **199**: p. 1066-1073.

68. Radford, A., "*Improving language understanding by generative pre-training",* 2018.
69. Vaswani, A., "*Attention is all you need",* Advances in Neural Information Processing Systems, 2017.
70. Radford, A., J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "*Language models are unsupervised multitask learners",* OpenAI blog, 2019, **1**(8): p. 9.
71. Lin, S., J. Hilton, and O. Evans, "*Truthfulqa: Measuring how models mimic human falsehoods",* arXiv preprint arXiv:2109.07958, 2021.
72. Morishita, M., H. Fukuda, S. Yamaguchi, K. Muraoka, T. Nakamura, M. Hayashi, I. Yoshioka, K. Ono, and S. Awano, "*An exploratory assessment of GPT-4o and GPT-4 performance on the Japanese National Dental Examination",* The Saudi Dental Journal, 2024, **36**(12): p. 1577-1581.
73. OpenAI, "*Hello GPT-4o"*, 2024, https://openai.com/index/hello-gpt-4o/, (26.04.2025).
74. OpenAI, "*Introducing GPT-4.1 in the API"*, https://openai.com/index/gpt-4-1/, (18.05.2025).
75. Liao, Y., L. Li, H. Xiao, F. Xu, B. Shan, and H. Yin, "*YOLO-MECD: Citrus Detection Algorithm Based on YOLOv11",* Agronomy, 2025, **15**(3): p. 687.
76. He, L., Y. Zhou, L. Liu, Y. Zhang, and J. Ma, "*Application of the YOLOv11-seg algorithm for AI-based landslide detection and recognition",* Scientific Reports, 2025, **15**(1): p. 12421.
77. Wang, Z., C. Li, H. Xu, X. Zhu, and H. Li, "*Mamba YOLO: A Simple Baseline for Object Detection with State Space Model"*, in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.
78. Gallagher, J.E. and E.J. Oughton, "*Surveying You Only Look Once (YOLO) Multispectral Object Detection Advancements, Applications And Challenges",* IEEE Access, 2025.
79. Aly, G.H., M. Marey, S.A. El-Sayed, and M.F. Tolba, "*YOLO based breast masses detection and classification in full-field digital mammograms",* Computer methods and programs in biomedicine, 2021, **200**: p. 105823.
80. Gai, R., N. Chen, and H. Yuan, "*A detection algorithm for cherry fruits based on the improved YOLO-v4 model",* Neural Computing and Applications, 2023, **35**(19): p. 13895-13906.
81. Hu, X., Y. Liu, Z. Zhao, J. Liu, X. Yang, C. Sun, S. Chen, B. Li, and C. Zhou, "*Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network",* Computers and electronics in agriculture, 2021, **185**: p. 106135.
82. Alemdar, K.D., M. Kayacı Çodur, M.Y. Codur, and F. Uysal, "*Environmental Effects of Driver Distraction at Traffic Lights: Mobile Phone Use",* Sustainability, 2023, **15**(20): p. 15056.
83. Kamble, S., V. Ghaytadak, A. Nadgouda, A. Chavan, and W. DB, "*Real Time Object Detection Using Yolo Technology",* International Research Journal of Engineering and Technology (IRJET).
84. Chen, J. and Z. Zhu, "*Real-time 3D object detection, recognition and presentation using a mobile device for assistive navigation",* SN Computer Science, 2023, **4**(5): p. 543.
85. Yaman, M.C. and Ş. Erel, "*Real-Time Multi-Object Recognition Using the Fusion of LIDAR and Camera Data",* Bozok Journal of Engineering and

Architecture, 2023, **2**(2): p. 1-19.
86. Alsamurai, M.Q.F., "*Development and implementation of YOLOV8-based model for human and animal detection during forest fires"*, 2023, Altınbaş Üniversitesi/Lisansüstü Eğitim Enstitüsü.
87. Jalil, A.J. and İ. Karaca, "*Makine öğrenimi algoritmalarını kullanarak gerçek zamanlı robot ile nesne algılama ve tanıma"*, in *Lisansüstü Eğitim Enstitüsü / Bilgisayar Mühendisliği Ana Bilim Dalı*, 2023, Kütahya Dumlupınar Üniversitesi.
88. Karakuş, S., M. Kaya, and S.A. Tuncer, "*Real-Time Detection and Identification of Suspects in Forensic Imagery Using Advanced YOLOv8 Object Recognition Models",* Traitement du Signal, 2023, **40**(5).
89. Moreira, F.W.R., G. Hermes, and J.M.M. de Lima, "*Development of a Cross Platform Mobile Application Using Gemini to Assist Visually Impaired Individuals"*, in *2024 9th International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*, 2024: IEEE.
90. Pudari, R., S. Bhutada, and S.P. Mudavath, "*Real Time Face Recognition Using Convoluted Neural Networks",* arXiv preprint arXiv:2010.04517, 2020.
91. Alamsyah, D.P., Y. Ramdhani, A.T. Syam, and A. Setiadi, "*Augmented Reality English Education Based iOS with MobileNetV2 Image Recognition Model"*, in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022: IEEE.
92. Sujaini, H., E.Y. Ramadhan, and H. Novriando, "*Comparing the performance of linear regression versus deep learning on detecting melanoma skin cancer using apple core ML",* Bulletin of Electrical Engineering and Informatics, 2021, **10**(6): p. 3110-3120.
93. Phadtare, M., V. Choudhari, R. Pedram, and S. Vartak, "*Comparison between yolo and ssd mobile net for object detection in a surveillance drone",* Int. J. Sci. Res. Eng. Man, 2021, **5**: p. 1-5.
94. Karthi, M., V. Muthulakshmi, R. Priscilla, P. Praveen, and K. Vanisri, "*Evolution of yolo-v5 algorithm for object detection: automated detection of library books and performace validation of dataset"*, in *2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 2021: IEEE.
95. Hussein, M.A.H., "*Text generation with recurrent neural networks"*, in *Computer Engineering and Computer Science and Control*, 2024, Çankırı Karatekin University.
96. Mitchell, M., K. van Deemter, and E. Reiter, "*Natural reference to objects in a visual domain"*, in *Proceedings of the 6th international natural language generation conference*, 2010.
97. Güzel, M., "*Performance comparison of deep learning models in Turkish text generation"*, in *Bilgisayar Mühendisliği Bilimleri-Bilgisayar ve Kontrol*, 2023, İzmir Bakırçay Üniversitesi.
98. Anayurt Özyeğin, H., "*Text Generation and Comprehension for Objects in Images and Videos"*, 2021, Middle East Technical University.
99. Koga, S. and W. Du, "*From text to image: challenges in integrating vision into ChatGPT for medical image interpretation",* Neural Regeneration Research, 2025, **20**(2): p. 487-488.
100. Husein, R.A., H. Aburajouh, and C. Catal, "*Large Language Models for Code*

Completion: A Systematic Literature Review", Computer Standards & Interfaces, 2024: p. 103917.
101. Roberts, J., M. Baker, and J. Andrew, "*Artificial intelligence and qualitative research: The promise and perils of large language model (LLM) 'assistance'",* Critical Perspectives on Accounting, 2024, **99**: p. 102722.
102. Roumeliotis, K.I., N.D. Tselikas, and D.K. Nasiopoulos, "*LLMs in e-commerce: a comparative analysis of GPT and LLaMA models in product review evaluation",* Natural Language Processing Journal, 2024, **6**: p. 100056.
103. Yuan, M., P. Bao, J. Yuan, Y. Shen, Z. Chen, Y. Xie, J. Zhao, Y. Chen, L. Zhang, and L. Shen, "*Large language models illuminate a progressive pathway to artificial healthcare assistant: A review",* arXiv preprint arXiv:2311.01918, 2023.
104. Pu, H., X. Yang, J. Li, and R. Guo, "*AutoRepo: A general framework for multimodal LLM-based automated construction reporting",* Expert Systems with Applications, 2024, **255**: p. 124601.
105. Liu, S., J. Zhang, L. Wang, and R.X. Gao, "*Vision AI-based human-robot collaborative assembly driven by autonomous robots",* CIRP Annals, 2024.
106. Apple Inc, "*Core ML"*, Documentation, https://developer.apple.com/documentation/coreml/, (20.04.2025).
107. Apple Inc., "*Chapter 1. Introducing VoiceOver"*, https://www.apple.com/voiceover/info/guide/_1121.html, (18.05.2025).
108. Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "*Microsoft coco: Common objects in context"*, in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 2014: Springer.
109. "*Furniture Computer Vision Project"*, 2022, https://universe.roboflow.com/objectdetection-uzld5/furniture-ngpea-h6zxi/, (02.04.2025).
110. "*HomeObjects"*, 2025, https://app.roboflow.com/objectdetection-uzld5/homeobjects/4, (18.05.2025).
111. Prechelt, L., "*Early stopping-but when?"*, in *Neural Networks: Tricks of the trade*, 2002, Springer, p. 55-69.
112. Wehr, A. and U. Lohr, "*Airborne laser scanning—an introduction and overview",* ISPRS Journal of photogrammetry and remote sensing, 1999, **54**(2-3): p. 68-82.
113. MathWorks, The MathWorks, Inc., "*Introduction to Lidar"*, https://www.mathworks.com/help/lidar/ug/lidar-processing-overview.html, (03.05.2025).
114. Han, X., Z. Zhang, N. Ding, Y. Gu, X. Liu, Y. Huo, J. Qiu, Y. Yao, A. Zhang, and L. Zhang, "*Pre-trained models: Past, present and future",* AI Open, 2021, **2**: p. 225-250.
115. Tinn, R., H. Cheng, Y. Gu, N. Usuyama, X. Liu, T. Naumann, J. Gao, and H. Poon, "*Fine-tuning large neural language models for biomedical natural language processing",* Patterns, 2023, **4**(4).
116. Salinas, A. and F. Morstatter, "*The butterfly effect of altering prompts: How small changes and jailbreaks affect large language model performance. arXiv",* arXiv preprint arXiv:2401.03729, 2024.
117. Goutte, C. and E. Gaussier, "*A probabilistic interpretation of precision, recall*

and F-score, with implication for evaluation", in *European conference on information retrieval*, 2005: Springer.
118. Lin, C.-Y., "*Rouge: A package for automatic evaluation of summaries"*, in *Text summarization branches out*, 2004.
119. Tautkute, I., A. Możejko, W. Stokowiec, T. Trzciński, Ł. Brocki, and K. Marasek, "*What looks good with my sofa: Multimodal search engine for interior design"*, in *2017 Federated Conference on Computer Science and Information Systems (FedCSIS)*, 2017: IEEE.