

ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE İÇ ORTAMLAR İÇİN YENİ
BİR DURUM ANALİZİ VE KALABALIK YÖNETİMİ YAKLAŞIMI

Mehmet Eren YEŞİLYURT

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2024

Her hakkı saklıdır

ÖZET

Yüksek Lisans Tezi

MAKİNE ÖĞRENMESİ ALGORİTMALARI İLE İÇ ORTAMLAR İÇİN YENİ BİR DURUM ANALİZİ VE KALABALIK YÖNETİMİ YAKLAŞIMI

Mehmet EREN YEŞİLYURT

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Doç. Dr. Mehmet Serdar GÜZEL

Hastane, otobüs terminali ve mağaza gibi iç ortamlarda uygulanan kalabalık yönetimi ve durum analizi teknikleri bireylerin güvenliğini sağlar ve bireylere verilen hizmetin kalitesini artırır. Dünya nüfusundaki hızlı artış ve şehirleşme iç ortamlar için durum analizi ve kalabalık yönetimi uygulamalarının önemini artırmıştır. Bu uygulamalar makine öğrenmesi tabanlı bilgisayarlı görü teknikleri ile hızlı ve verimli bir şekilde gerçekleştirilebilir.

Bu çalışmada kalabalık yoğunluğu analizi ve kuyruk tespiti üzerine odaklanılmıştır. Özel bir hastaneden alınan iki farklı güvenlik kamerasının görüntülerindeki insanlar YOLOv8 nesne tespiti modeli ile tespit edilmiştir. Tespit edilen insanların sınırlayıcı kutularının konumları görüntü sıcaklık haritalarının oluşturulması için kullanılmıştır. İnsanların sahnedeki gerçek konumları kamera kalibrasyonu yapıldıktan sonra bazı varsayımlar altında tahmin edilmiştir. Tahmin edilen gerçek konumlar ile sahnelerin kuş bakışı görüntüleri elde edilmiş ve bu görüntüler üzerinde sıcaklık haritaları oluşturulmuştur.

Kuyruk tespiti için evrimsel sinir ağlarına dayanan bir model geliştirilmiştir. Etiketli kamera görüntüleri üzerinde eğitilen model, sadece sınırlayıcı kutu koordinatlarını kullanarak kişilerin bir kuyruğun parçası olup olmadığını tahmin etmektedir. Çapraz doğrulama sonucunda model 0,90 doğruluğa ve 0,91 F1 skoruna ulaşmıştır. Çalışma sonucunda giriş olarak sadece görüntü koordinatlarını kullanan bir evrimsel sinir ağının kuyruk tespiti konusunda başarılı olduğu gösterilmiştir. Kalibre edilmiş kameralar yardımıyla sahnelerin kuş bakışı sıcaklık haritaları elde edilmiş, kuyruk tespiti ve yoğunluk analizi uygulamalarını gözlemlemek için bir web uygulaması oluşturulmuştur.

Temmuz 2024, 54 sayfa

Anahtar Kelimeler: Makine Öğrenmesi, Kalabalık Yönetimi, Nesne Tespiti, YOLO

ABSTRACT

Master Thesis

A NEW APPROACH TO SITUATION ANALYSIS AND CROWD MANAGEMENT FOR INDOOR ENVIRONMENTS USING MACHINE LEARNING ALGORITHMS

Mehmet EREN YEŞİLYURT

Ankara University
Graduate School of Natural and Applied Sciences
Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Mehmet Serdar GÜZEL

In indoor environments such as hospitals, terminals and stores, application of crowd management and situation analysis techniques ensure the safety of individuals and increase the quality of service provided. The rapid increase in population and urbanization have increased the importance of situation analysis and crowd management applications for indoor environments. These applications can be implemented quickly and efficiently with machine learning-based computer vision techniques.

This study focuses on crowd density analysis and queue detection. People in the footages of two different security cameras taken from a private hospital were detected with the YOLOv8 model. The bounding box positions of the detected people were used to create image heatmaps. The real-world positions of people in the scene were estimated under some assumptions after camera calibration. Bird's-eye views of the scenes were obtained with the estimated real locations and heatmaps were created over these views.

A convolutional neural network model has been developed for queue detection. Trained on labeled camera images, the model predicts whether a given person belongs to a queue using only bounding box coordinates. The cross-validated model achieved an accuracy of 0.90 and an F1 score of 0.91. As a result of the study, it was shown that a convolutional neural network using only image coordinates as input was successful in queue detection. Bird's eye view temperature maps of the scenes were obtained with the help of calibrated cameras, and a web application was created to observe the queue detection and density analysis applications.

July 2024, 54 pages

Keywords: Machine Learning, Crowd Management, Object Detection, YOLO

ÖNSÖZ VE TEŞEKKÜR

Yüksek Lisans çalışmalarında ve tez yazma sürecimde bilgi ve tecrübelerini benimle paylaşan ve bana yol gösteren danışman hocam Sayın Doç. Dr. Mehmet Serdar GÜZEL'e saygı ve teşekkürlerimi sunarım.

Çalışmalarım süresince her zaman benim yanımda olan ve beni destekleyen değerli aileme teşekkürlerimi sunarım.

Mehmet Eren YEŞİLYURT
Kastamonu, Temmuz 2024



İÇİNDEKİLER

TEZ ONAY SAYFASI

ETİK.....	i
ÖZET.....	ii
ABSTRACT	iii
ÖNSÖZ VE TEŞEKKÜR.....	iv
SİMGELER VE KISALTMALAR DİZİNİ	vi
ŞEKİLLER DİZİNİ	vii
ÇİZELGELER DİZİNİ	viii
1. GİRİŞ	1
2. KURAMSAL TEMELLER VE KAYNAK ÖZETLERİ	3
2.1 Kalabalık Yönetimi ve Durum Analizi.....	3
2.1.1 Yoğunluk analizi	3
2.1.2 Kuyruk tespiti ve grup etkinliği tanıma.....	6
2.1.3 Anormal davranış tespiti	8
2.2 Yapay Zekâ	10
2.2.1 Makine öğrenmesi	12
2.2.2 Derin öğrenme	17
2.2.3 Evrimsel sinir ağları.....	20
2.2.4 Derin öğrenme kütüphaneleri.....	21
2.3 Bilgisayarlı Görü	22
2.3.1 İğne deliği kamera modeli	24
2.3.2 Kamera kalibrasyonu	27
2.3.3 Nesne tespiti	28
3. MATERYAL VE YÖNTEM.....	34
3.1 Veri Seti.....	34
3.2 Kuyruk Tespiti	35
3.3 Yoğunluk Analizi.....	39
3.4 İnteraktif Web Uygulaması.....	45
4. TARTIŞMA VE SONUÇ.....	47
KAYNAKLAR	49
ÖZGEÇMİŞ.....	54

SİMGELER VE KISALTMALAR DİZİNİ

α	Alfa
σ	Sigma
Σ	Sigma

Kısaltmalar

ADNKS	Adrese Dayalı Nüfus Kayıt Sistemi
AI	Artificial Intelligence
BCE	Binary Cross Entropy
BOW	Bag of Words
CNN	Convolutional Neural Network
DLT	Direct Linear Transformation
FN	False Negative
FP	False Positive
GSP	Generalized Sequential Patterns
HOG	Histogram of Oriented Gradients
IOU	Intersection Over Union
LLM	Large Language Model
MAE	Mean Average Error
MAP	Mean Average Precision
MCNN	Multi-column Convolutional Neural Network
MLP	Multilayer Perceptron
MSE	Mean Squared Error
NAS	Neural Architecture Search
NMS	Non Maximum Suppression
OCR	Optical Character Recognition
RCNN	Region Based Convolutional Neural Network
RELU	Rectified Linear Unit
RGB	Red, Green, Blue
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
TÜİK	Türkiye İstatistik Kurumu
YOLO	You Only Look Once

ŞEKİLLER DİZİNİ

Şekil 2.1 Kolektif bir etkinlik olarak kuyruğa girme davranışı	7
Şekil 2.2 Yapay zekâ kavramları hiyerarşisi.....	12
Şekil 2.3 İkili bir sınıflandırma problemi için karmaşıklık matrisi.....	15
Şekil 2.4 Tek gizli katmana sahip bir MLP'nin yapısı.....	17
Şekil 2.5 Temel evrimsel sinir ağı yapısı	20
Şekil 2.6 Evrimsel sinir ağlarında maksimum ortaklama.....	21
Şekil 2.7 İğne deliği kamera modeli	25
Şekil 2.8 Intersection Over Union ölçütünün hesaplanması	29
Şekil 2.9 YOLO sisteminin genel yapısı	33
Şekil 3.1 Yetişkin yeşil kapı bölgesinde bekleyen insanlar	35
Şekil 3.2 Triyaj acil kayıt bölgesinden kuyruk tespiti için etiketlenmiş bir görüntü	36
Şekil 3.3 Kuyruk tespiti modelinin yapısı.....	37
Şekil 3.4 Kuyruk modeline ait kayıp ve doğruluk grafikleri	38
Şekil 3.5 Kuyruk modelinin bir görüntü üzerinde çalıştırılmasıyla elde edilen çıktı	39
Şekil 3.6 Triyaj kamerasında kalibrasyon için nokta ikililerinin bulunması	40
Şekil 3.7 Yeşil kapı kamerasında kalibrasyon için nokta ikililerinin bulunması.....	41
Şekil 3.8 Triyaj ve yeşil kapı bölgelerine ait serpilme grafiği örnekleri.....	42
Şekil 3.9 Triyaj bölgesi görüntü sıcaklık haritası	43
Şekil 3.10 Yeşil kapı bölgesi görüntü sıcaklık haritası	44
Şekil 3.11 Triyaj kayıt ve yeşil kapı önü bölgelerine ait kuş bakışı sıcaklık haritaları...	45
Şekil 3.12 Streamlit interaktif web uygulaması arayüzü	46

ÇİZELGELER DİZİNİ

Çizelge 2.1 Makine öğrenmesi performans ölçütleri	16
Çizelge 2.2 Derin öğrenme kütüphaneleri	21
Çizelge 2.3 Bilgisayarlı görü uygulama alanları	23
Çizelge 2.4 NMS algoritmasının adımları	30
Çizelge 2.5 YOLO nesne tespiti yöntemi modelleri	32
Çizelge 3.1 Veri seti özet bilgileri	34
Çizelge 3.2 Kuyruk modeli eğitim sonuçları	38



1. GİRİŞ

Günümüz dünyasında insanlar alışveriş yapmak, sağlık hizmeti almak ve benzeri faaliyetleri gerçekleştirmek için sıklıkla iç ortamlarda toplu halde bulunmak durumunda kalmaktadır. Bu durum hastane, mağaza, havalimanı ve otogar gibi ortamlarda kalabalık yönetimi ve durum analizi yöntemlerinin önemini artırmıştır. Kalabalık yönetimi ve durum analizinin amacı, insan davranışlarının ve insan hareketlerinin verimliliği, hizmet kalitesini ve güvenliği artırmak için takip edilmesi ve analiz edilmesidir. Nüfus artışı ve modernleşme sürecinde giderek artan şehirleşme, iç ve dış ortamlardaki insan yoğunluklarının artışı da beraberinde getirmiştir. Türkiye İstatistik Kurumu (TÜİK) Adrese Dayalı Kayıt Sistemi (ADNKS) verilerine göre 2007 yılında 70.586.256 olan Türkiye nüfusu, 2023 yılında 85.372.377 kişi olmuştur. İl ve ilçe merkezlerinde yaşayanların oranı ise 2007 yılında %70,5 iken, 2023 yılında bu oran %93 olmuştur (TÜİK 2008, 2024). Nüfus artışı ve şehirleşmenin bir sonucu olarak kalabalık yönetimi ve iç mekân planlaması modern dünyada bir gereklilik haline gelmiştir.

Kalabalık yönetiminde geleneksel olarak uygulanan yöntem, insanların davranış ve hareketlerinin doğrudan gözlemleme yoluyla veya güvenlik kameraları görüntülerinin gözlemlenmesi yoluyla takip edilip incelenmesidir. Gözlemleme yapan kişi böylece gözlemlenen bölgede yoğunlukların nerede oluştuğu ve insanların anormal davranışlarda bulunup bulunmadıkları gibi bulgulara ulaşabilir. Ancak kalabalık yönetiminin bu şekilde gerçekleştirilmesi zaman alıcı ve verimsizdir. Bu nedenle kalabalık yönetimi ve durum analizini otomatik olarak gerçekleştiren bilgisayarlı sistemler bu alanda büyük bir potansiyel sunmaktadır. Son yıllarda büyük bir hızla gelişen makine öğrenmesi tabanlı bilgisayarlı görü sistemleri bu potansiyelin temelini oluşturmaktadır.

Programcılar tarafından oluşturulan klasik algoritmalara nazaran veri içerisindeki örüntülerden öğrenen makine öğrenmesi yöntemleri, klasik yöntemlere göre birçok alanda daha başarılıdır. Son yıllarda veri miktarı ve bilgisayar hesaplama kapasitelerindeki artış, makine öğrenmesi ve özellikle de makine öğrenmesinin bir alt alanı olan derin öğrenme yöntemlerinin popülerliğini artırmıştır. Derin öğrenmenin

bilgisayarlı görü uygulamalarına örnek olarak nesne tespiti, yüz tanıma, insan davranışlarının tanınması ve poz tespiti verilebilir (Voulodimos vd. 2018).

Bu tez çalışmasında kalabalık yoğunluğu analizi ve kuyruk tespiti üzerine durulmuştur. Çalışma kapsamında bir hastanenin iç ortamında bulunan iki farklı güvenlik kamerasından alınan görüntüler kullanılmıştır. You Only Look Once (YOLO) nesne tespiti yöntemiyle tespit edilen insanlar, görüntü üzerindeki yoğun bölgeleri görselleştiren sıcaklık haritalarının oluşturulmasında kullanılmıştır. Ayrıca gözlemlenen kişilerin gerçek dünya koordinatları yaklaşık olarak belirlenerek bölgelerin kuş bakışı haritaları oluşturulmuştur. Bunu yapabilmek için öncelikle iki kamera için de Doğrudan Doğrusal Dönüşüm (DLT) yöntemi kullanılarak kamera kalibrasyonu gerçekleştirilmiştir. Elde edilen projeksiyon matrisinin tersi, bazı varsayımlar altında kişilerin gerçek dünyadaki 2 boyutlu konumlarını yaklaşık olarak tespit etmek için kullanılmıştır. Böylece YOLO ile tespit edilen kişilerin sınırlayıcı kutularından alınan noktalar, dünya koordinat sisteminin zeminini oluşturan xy düzlemindeki noktalara çevrilmiştir.

Çalışma kapsamında kuyruk tespiti için CNN tabanlı yeni bir model geliştirilmiştir. Modeli eğitmek ve doğrulamak için Ankara'da bulunan bir hastanenin “Acil Triyaj” bölgesini gözlemleyen bir güvenlik kamerasından alınan görüntülerden oluşan bir veri seti kullanılmıştır. Oluşturulan modelin girişini YOLO ile tespit edilen kişilerin koordinatları oluşturmaktadır. Model, tespit edilen her kişi için “Kuyrukta” veya “Kuyrukta Değil” şeklinde bir sınıflandırma yapmaktadır.

Tez çalışmasının ikinci bölümünde kalabalık yönetimi ve durum analizi genel hatlarıyla betimlenmiştir. Yapay zekâ ve makine öğrenmesi ile ilgili önemli kavramlar açıklanmış, yapay sinir ağları ve özellikle bilgisayarlı görü uygulamalarında sıklıkla kullanılan Evrişimsel Sinir Ağı (CNN) anlatılmıştır. Çalışmada kullanılan kamera kalibrasyonu ve YOLO nesne tespiti yöntemi açıklanmıştır. Üçüncü bölümde uygulanan yöntemler ve kullanılan veri seti anlatılmıştır. Dördüncü bölümde ise elde edilen sonuçlar tartışılmış ve değerlendirilmiştir.

2. KURAMSAL TEMELLER VE KAYNAK ÖZETLERİ

2.1 Kalabalık Yönetimi ve Durum Analizi

Belirli bir zaman aralığında bir bölgede yoğunlaşan insanlar, o bölgede bir kalabalık (crowd) meydana getirir. Kalabalık yönetimi, kalabalıkların verimli bir şekilde hareket etmesine olanak sağlayarak bireylerin konforunu ve güvenliğini artırmak amacıyla gerçekleştirilen proaktif eylemleri ifade eder (Feliciani vd. 2022). Örneğin bir bölgede kuyrukta bekleyen insanların haftanın belirli günleri ve saatlerinde daha çok yoğunlaştıkları tespit edilirse, bu insanların konforunu artırmak ve verimliliği sağlamak amacıyla o gün ve saatlerde daha fazla personel tahsis edilebilir.

İç ortamlarda kalabalık yönetimi ve kalabalık durumu; yoğunluk analizi, kuyruk tespiti ve anormal davranış tespiti başlıkları altında incelenebilir.

2.1.1 Yoğunluk analizi

Yoğunluk analizi, belirli bir bölgedeki insanların sayısını veya yoğunluğunu belirlemeyi amaçlar. Yoğunluk analizi yöntemleri tespit tabanlı yöntemler, regresyon tabanlı yöntemler ve CNN tabanlı yöntemler olmak üzere üç kategori altında incelenir (Sindagi ve Patel 2018, Li vd. 2021).

Yoğunluk analizi ve nesne tespiti konularındaki birçok öncül çalışmanın tespit tabanlı olduğu görülmektedir. Tespit tabanlı yöntemlerde görüntülerdeki insanlar sayım yapmak veya takip etmek amacıyla ayrı ayrı tespit edilir. Bu tür yöntemlerin başarısı kalabalık yoğunluğu arttıkça azalır (Sindagi ve Patel 2018). Geleneksel olarak tespit tabanlı yöntemlerde görüntülerden çıkarılan özellikler makine öğrenmesi algoritmaları tarafından sınıflandırılarak insanların tespiti gerçekleştirilir. Dalal ve Triggs (2005) Yönlü Gradyanlar Histogramı (HOG) yöntemi ile çıkarılan özellikleri Destek Vektör Makinesi (SVM) ile sınıflandırarak insan tespiti gerçekleştirmiştir. Wu ve Nevatia (2007) insanları vücut parçalarının birleşimi olarak modellemiş, “edgelet” adını verdikleri kenar özelliklerini kullanan parça sınıflandırıcıları aracılığıyla insanları tespit etmiş ve takip

etmiştir. Gall vd. (2011) Random Forest algoritmasını Hough dönüşümünü gerçekleştirmek için uyarlamış, “Hough Forest” adını verdikleri bu yöntemle kişilerin tespitini, takibini ve davranışlarının sınıflandırılmasını yapmıştır. Gao vd. (2016) insan akışlarını tespit etmek için derinlik ve renk bilgisini birlikte kullanan bir yöntem geliştirmiştir. Derinlik sensörüne ve RGB (Red, Green Blue) kameraya sahip bir Microsoft Kinect cihazının yukarıdan aşağıya görüntüleme yapacak şekilde kullanıldığı çalışmada, önce derinlik görüntüleri üzerinde hareketli bölgeler tespit edilmiştir. Yukarıdan aşağı görünümde insanların derinlik sensörüne en yakın vücut parçaları kafaları olduğu için, hareketli bölgeler içinde yerel minimumlar tespit edilerek “kafa adayları” bulunmuştur. Daha sonra HOG özelliklerini kullanan bir SVM sınıflandırıcısı ile insanların kafaları tespit edilmiştir.

Regresyon tabanlı yöntemler, görüntülerden çıkarılan özellikler ve görüntülerdeki kişi sayısı arasındaki ilişkiyi öğrenmeyi amaçlar (Hao vd. 2023). Lempitsky ve Zisserman (2010) görüntülerden yoğunluk haritası elde edip insanların sayısının tespitini yoğunluk fonksiyonunun görüntü üzerinde integrali olarak ele almıştır. Chen vd. (2012) hücrelere ayrılmış görüntülerden yerel olarak kenar ve doku gibi özellikler çıkarmıştır. Yerel olarak çıkarılan özellikler bir özellik vektörü oluşturmak için kullanılmış, daha sonra bu vektör ile her hücre için insan sayısını tahmin edebilen çok çıkışlı bir ridge regresyonu modeli eğitilmiştir. Xu ve Qiu (2016) kalabalık yoğunluğunu bulabilmek için çok sayıda görüntü özelliğinden yararlanmış ve yüksek performans sağlamak için Random Forest algoritmasını regresyon yöntemi olarak kullanmıştır.

Son yıllarda derin öğrenme ve CNN tabanlı modeller yoğunluk analizinde sıklıkla kullanılmaya başlanmıştır. Zhang vd. (2016) bir görüntünün yoğunluk haritasını elde etmek için Çok Sütunlu Evrişimsel Sinir Ağlarını (MCNN) geliştirmiştir. Çok sütunlu mimarinin kullanılmasındaki amaç, perspektif etkisinden dolayı görüntünün farklı bölgelerinde farklı ölçek ve yoğunluklarda bulunan insanların özelliklerinin öğrenilmesine modelin adapte olabilmesidir. Üç sütunlu mimarinin her sütununda farklı filtre boyutları kullanılmıştır. Filtre boyutları ve daha yüksek filtre boyutlarına sahip sütunlarda daha az filtrenin kullanılması dışında üç sütunun yapısı da aynıdır. Modelin

çıktısı olan yoğunluk haritasını elde etmek için sütunların sonuçları üst üste konulur ve 1x1 evrişim işlemi gerçekleştirilir.

Zhang vd. (2018) çok sütunlu mimari yerine, tek sütunlu ve sabit filtre boyutuna sahip bir ağ mimarisi tasarlamıştır. Ağın perspektiften kaynaklanan farklı ölçekteki insanlara adapte olabilmesi için, ağın içerisindeki farklı ölçekli özellik haritaları birleştirilerek yoğunluk haritası tahmini yapılmıştır.

Hu vd. (2020) yoğunluk tabanlı kalabalık sayımı modellerinin mimarisini manuel olarak tasarlamak yerine, Sinir Ağı Mimarisi Araması (NAS) yöntemleriyle arayarak bulmuştur.

Wang ve Breckon (2022) tasarladıkları ağın temelinde Inception-v3 mimarisini kullanmış; görüntünün arka planındaki özellikleri görmezden gelip ön plandaki özellikleri yoğunlaştırmak amacıyla ağa attention katmanı eklemiştir.

Liang vd. (2023) zor ve zaman alıcı etiketleme işlemi gerektiren denetimli kalabalık sayımı yöntemlerine alternatif olarak görü-dil modellerini kullanarak denetimsiz olarak kalabalık sayımı gerçekleştiren bir sistem geliştirmiştir.

Liu vd. (2023) kalabalık sayımı problemini parçalara ayrılabilir bir nokta sorgulama süreci olarak modellemiştir. Bir transformer ağının kullanıldığı çalışmada nokta sorgularının sayısı dinamik olarak belirlenmektedir.

Ma vd. (2024) kalabalık sayımı yapmak için yoğunluk haritası kullanmadan kişilerin koordinatlarını doğrudan tahmin eden bir sistem geliştirmiştir. Sistem, birleştirilen özellik haritalarını regresyon ve sınıflandırma uçlarına teslim eder. Regresyon ucu görüntüdeki koordinatları tespit ederken, sınıflandırma ucu koordinatların güven değerlerini belirler.

Ranasinghe vd. (2024) yoğunluk haritasını üretmek ve kalabalık sayımı yapmak için difüzyon modellerini kullanmış, açık veri setlerinde diğer yöntemlere karşı üstünlük sağlamıştır.

Shi vd. (2024) eğitim gerektirmeyen bir nesne sayıcı modeli geliřtirmek için “Segment Anything” segmentasyon modelini kullanmıřtır. Geliřtirilen nesne sayıcı modeli, kendisine verilen nokta, koordinat veya metin komutunda belirtilen nesneyi istenilen görüntü üzerinde sayar.

2.1.2 Kuyruk tespiti ve grup etkinlięi tanıma

Bir hizmet veya ürün almak isteyen insan grupları, hizmet almak için bekledikleri bölgede kuyruklar oluřtururlar. Kuyruklar, kalabalık kavramından farklı olarak insanların düzeni içinde belirli bir amaç doęrultusunda sıraya girmeleriyle oluřur. Kuyruk tespiti verimlilik artışı için önemlidir; hangi zaman dilimlerinde kuyrukta bekleyen insanların daha fazla olduęunun bilinmesi personel tahsisinin daha verimli bir řekilde yapılmasına ön ayak olur. Kuyruk tespitinden elde edilen çıkarımlar sayesinde yapılan iyileřtirmeler, bekleme zamanlarını düşürerek müşteri memnuniyetini artırır. Ayrıca kuyrukta bekleyen insan sayısı canlı olarak takip edilip ortalama bekleme süresi istatistięinden yararlanılarak kuyruktaki insanlara tahmini bekleme süreleri bildirilebilir.

En az iki kiřinin varlığı ile kolektif olarak geręekleřtirilen kuyruk oluřturma davranışı, bireysel bir davranış olmaktan ziyade bir grup etkinlięidir. Kuyruęa girme eylemini bireysel olarak deęerlendirmemenin önemi řekil 2.1’de görülmektedir. Bireysel olarak bakıldığında benzer vücut pozlarına sahip kiřiler, etraftaki kiřilerin bağlamında bakıldığında kuyrukta bekliyor veya sohbet ediyor olabilir.



Şekil 2.1 Kolektif bir etkinlik olarak kuyruğa girme davranışı (Choi vd. 2009)

Grup etkinliği tanıma konusunda geleneksel yöntemler manuel olarak çıkarılan özellikleri makine öğrenmesi algoritmalarıyla değerlendirir; ancak son zamanlarda derin öğrenme tabanlı yöntemler geleneksel yöntemlere göre üstün başarı elde etmiştir (Wu vd. 2021).

Choi vd. (2009) HOG özelliklerini kullanan parça tabanlı bir sınıflandırıcı yardımıyla karşıdan karşıya geçme, grup olarak bekleme, kuyruğa girme, yürüme ve sohbet etme grup etkinliklerini tespit etmiştir.

Tran vd. (2014) bir sahnenin, köşelerin insanları, kenar ağırlıklarının ise insanlar arasındaki ilişkilerin yoğunluğunu temsil ettiği bir çizge ile modellenebileceğini belirtmiştir. İnsanlar arası etkileşimin şiddetini temsil eden kenar ağırlıkları, Sosyal Kuvvet Modeli (SFM) ile modellenerek hesaplanmış, çizge kümelemesi ile bir grubun parçası olarak etkileşimde bulunmayan kişiler elenmiştir. Grup içi etkileşimleri kodlayan Yerel Grup Etkileşimleri özellikleri çıkarılmış, bu özellikler kullanılarak bir Kelime Çantası Modeli (BoW) oluşturularak SVM ile grup etkinliği sınıflandırması yapılmıştır.

Wu vd. (2019) grup etkinliği tanınması gerçekleştirilecek sahneyi Tran vd. (2014) benzer şekilde köşelerin insanları, kenarların insanlar arası ilişkileri temsil ettiği bir çizge ile modellemiştir. Kullandıkları yöntemde çizgenin oluşturulmasından önceki aşamada

videodan rastgele örneklenen karelerden Inception-v3 ağı ile özellik haritaları çıkarıldıktan sonra karelerdeki her insan için özellik vektörleri elde edilmiştir. Bu vektörlerden çizgilerdeki her kenarın ağırlığı, kenarın bağladığı insanların görüntü ve pozisyon özellikleri dikkate alınarak hesaplanmıştır. Çizgelere Çizge Evrişimsel Ağları (GCN) uygulandıktan sonra, genel grup etkinliği ve etraftaki insanların bağlamında bireysel davranışlar sınıflandırılır.

Xu ve Yin (2023) aktörler arasındaki ilişkileri öğrenmek için MLP'lere dayanan bir sistem geliştirmiştir. Üç adet modülden meydana gelen sistemde modüllerden biri uzamsal ilişkileri, diğeri zamansal ilişkileri modellerken bir diğeri modül boyutları arası ilişki özelliklerini işler.

Zhang vd. (2024) grup etkinliği tanıma probleminde sadece aktörler arası ilişkileri değil, aktör-nesne etkileşimlerini de ele almış ve transformer temelli bir ağ kullanmıştır.

2.1.3 Anormal davranış tespiti

Güvenlik kamerası görüntülerinden anormal davranışların otomatik olarak tespiti, insanların can ve mal güvenliğini sağlamak açısından önem taşır. Anormal davranış tespiti sayesinde tehlikeli durumlar ortaya çıkmadan önlenabilir veya ortaya çıkmakta olan tehlikeli durumlar büyümeden etkisiz hale getirilebilir. Video akışından yapılan otomatik tespitler, bir insanın manuel olarak yapabileceği tespitlerden daha hızlı ve verimlidir.

İnsanlarda anormal davranışları yerel ve küresel olmak üzere iki kategoriye ayıran Kuppusamy ve Bharathi (2022), yere düşmek ve başıboş dolaşmak gibi sadece bir veya iki kişinin davranışının insanların geri kalanının davranışlarından farklılık gösterdiği anormal davranışları yerel olarak sınıflandırmıştır. Kavga etmek, bir yeri tahliye etmek gibi çok sayıda insanın katıldığı anormal davranışları ise küresel olarak nitelendirmiştir.

Başıboş dolaşma davranışı (loitering), “bir yerde orada bulunmak için bir neden olmadan bulunmak” olarak tanımlanır (Merriam-Webster, 2014). Başıboş dolaşma davranışı, suç

faaliyetlerinin ön habercisi olabilir. Örneğin bir lise önünde bu davranışı sergileyen kişi öğrencilere uyuşturucu madde satmaya çalışan bir uyuşturucu taciri olabilir; benzer şekilde bir kuyumcu yakınlarında başıboş dolaşan biri hırsızlık yapmayı planlıyor olabilir. Başıboş dolaşma davranışı tespitin bir başka kullanım yeri de akli dengesi yerinde olmayan hastaların güvenliklerini sağlamak amacıyla otomatik olarak tespit edilmesidir.

Gómez vd. (2015) başıboş dolaşmayı tespit etmek için Genelleştirilmiş Ardışık Örüntüler (GSP) algoritmasını kullanmıştır. Eğitim aşamasında kamera görüntülerinden takip edilen kişinin eylemleri videonun her saniyesi için ardışık olarak sıralanarak (yürü, yürü, dur, yürü, dur gibi) etiketlenmiştir. GSP algoritması pozitif örnekleri tarayarak başıboş dolaşma olaylarında sıklıkla rastlanılan mikro-örüntüleri ortaya çıkarır. Eğer test aşamasında gözlemlenen diziler başıboş dolaşma davranışıyla uyuşursa hasta bakıcısına haber verilir.

Nayak vd. (2019) başıboş dolaşan insanları çok sayıda kameradan oluşan bir gözetim ağında tespit etmek ve takip etmek için derin öğrenme tabanlı bir sistem geliştirmiştir. Çalışmada insan tespiti için YOLOv3 algoritması, tespit edilen insanların takibi için DeepSORT algoritması kullanılmıştır. Tespit edilen bir kişinin başıboş dolaşma davranışı sergileyip sergilemediği, zaman ve yer değiştirme kriterlerine göre belirlenmiştir. Eğer bir kişi bulunduğu bölgede o bölgede normal sayılabilecek süreden daha fazla zaman geçirdiyse zaman kriterini sağlamış olur. Eğer kişi bölgede bulunduğu süre boyunca belirlenen bir eşiğin altında kalacak şekilde yer değiştirmişse yer değiştirme kriterini sağlamış olur. Eşik değerlerinin bölgenin özelliğine göre belirlendiği bu iki kriteri de sağlayan kişiler, başıboş dolaşan kişiler olarak işaretlenmiştir.

Nguyen vd. (2014) tarafından bildirildiğine göre, yaşlı insanlar arasında düşmeler yaralanmaların önde gelen nedenlerinden biridir ve yaralanma sonucu ölümlerin %40'ı düşmeler sonucu gerçekleşir (WHO 2014). Bu nedenle Nguyen vd. (2014) insanların düşmesini otomatik olarak tespit edebilecek bir sistem geliştirmiştir. Sistemin ilk aşamasında gri ölçeğe çevrilen video görüntülerinden arka plan çıkarılır. Eşikleme işlemiyle ön plandaki nesnelere ait pikseller tespit edilir. Morfolojik operasyonlar

aracılığıyla boşluklar doldurulur. Tespit edilen şekiller sonraki aşamada kural tabanlı basit bir sınıflandırıcı ile şekillerin en-boy oranı ve açıları dikkate alınarak “düşüyor”, “ayakta” veya “oturuyor” olarak sınıflandırılır.

Gul vd. (2020) acil tıbbi yardım gerektiren eylemleri anormal hasta davranışı olarak nitelendirdikleri çalışmalarında, aralarında bayılma, geriye doğru düşme ve kusma gibi eylemlerin olduğu sekiz anormal davranıştan meydana gelen bir veri seti oluşturmuştur. Hem kişileri tespit etmek, hem de sınıflandırmak için YOLO nesne tespit modelini kullanmışlardır.

2.2 Yapay Zekâ

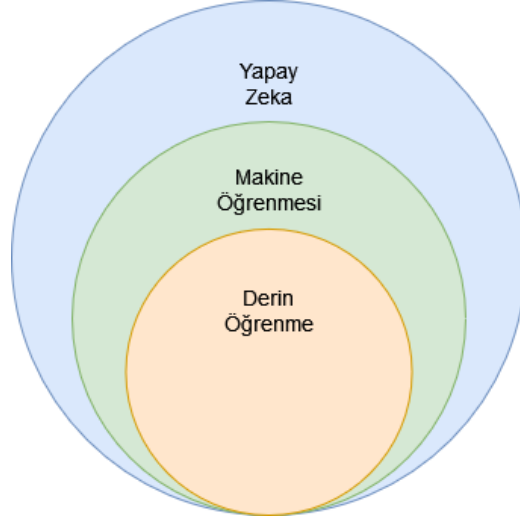
Yapay zekâ (AI), insan zekâsıyla ilişkilendirilen davranışların otomasyonu ile ilgilenen bilgisayar bilimleri dalıdır (Luger 2009). Russel ve Norvig’e (2022) göre, yapay zekâ alanında yapıldığı kabul edilen ilk çalışma insan beynindeki sinir hücrelerinin çalışmasından ilham alarak yapay sinir hücrelerini modelleyen McCulloch ve Pitts (1943) tarafından gerçekleştirilmiştir. 1955 yılında Dartmouth Koleji’nde McCarthy, Minsky, Shannon ve Rochester tarafından organize edilen çalıştay önerisi, “yapay zekâ” teriminin ilk resmi kullanımı olarak tarihe geçmiştir (Russel ve Norvig 2022).

Yapay zekâ alanında erken dönemdeki gelişmeler, bu alanla ilgili felsefi tartışmaları da beraberinde getirmiştir. Alan Turing (1950) “Makineler düşünebilir mi?” sorusunu ortaya atmış ve bu soruyla ilgili bir düşünce deneyi geliştirmiştir. Turing Testi olarak anılan bu deneyde, bir sorgulayıcı farklı odalarda bulunan ve yazılı olarak haberleştiği X ve Y kişilerine çeşitli sorular yöneltir. X ve Y’den biri insan, diğeri makinedir. Eğer deneyin sonunda sorgulayıcı kimin makine, kimin insan olduğunu ayırt edemiyorsa makine testi geçmiş olur. Turing Testini geçebilen bir makine doğal dil işleme, bilgi temsili, otomatik çıkarımlar yapabilme ve öğrenerek yeni durumlara adapte olabilme yeteneklerine sahip olmalıdır (Russel ve Norvig 2022).

Yapay zekâ teknolojilerinin getirebilecekleri üzerine duyulan aşırı iyimserlik ve yapay zekâ teknolojilerinden istenen gerçek dışı beklentiler, bu beklentilerin yeterli düzeyde karşılanamaması üzerine “yapay zekâ kışı” adı verilen duraklama dönemleriyle sonuçlanmıştır. Yapay zekâ alanına duyulan ilginin ve bu alana yapılan yatırımların azaldığı bu dönemlerin ilki 1974-1980 yılları arasında, ikincisi ise 1990’lı yıllarda yaşanmıştır (Reza 2020).

Son dönemlerde büyük veri setleri ve hızla artan hesaplama kapasitesi, yapay zekâ teknolojilerinin hızlı gelişimiyle sonuçlanmıştır. Özellikle son yıllarda Büyük Dil Modeli (LLM) alanında yaşanan gelişmeler insanlarda heyecan ve kaygı yaratmıştır. Başlangıçta sadece metinler ile çalışabilen büyük dil modelleri, teknolojinin kısa sürede ilerlemesiyle görüntü, ses ve videoları da algılayabilecek ve bunlar hakkında sorulan soruları cevaplayabilecek kapasiteye ulaşmıştır. OpenAI tarafından geliştirilen ChatGPT ve Google tarafından geliştirilen Gemini büyük dil modellerine örnek olarak verilebilir. ChatGPT ve benzeri LLM modellerinin hızlı gelişimi, daha önce makinelerin başarılarının sınırlı olduğu sohbet etme ve yaratıcı yazı yazma gibi konulardaki yüksek başarıları ve bu modellerin ve artan popülaritesi, yapay zekânın yarattığı tehlikeler hakkındaki tartışmaları da ön plana çıkarmıştır. Bazı bilim insanları, yapay zekânın insan zekâsının çok üstünde bir seviyeye çıkarak “süper zekâ”ya ulaşacağını ve bu durumun insanları ve dünya üzerindeki diğer yaşam formlarını tehdit eder vaziyete geleceğini öne sürmektedir.

Öğrenebilen bilgisayar sistemlerini inceleyen makine öğrenmesi, yapay zekânın bir alt dalıyken, derin öğrenme de makine öğrenmesinin alt dalıdır. Yapay zekâ, makine öğrenmesi ve derin öğrenme arasındaki ilişki şekil 2.2’de görülebilir.



Şekil 2.2 Yapay zekâ kavramları hiyerarşisi

2.2.1 Makine öğrenmesi

Murphy (2012) makine öğrenmesi kavramını “veriden otomatik olarak örüntüler çıkaran ve bu örüntüleri gelecekle ilgili tahminlerde bulunmak veya belirsizlik altında kararlar vermek için kullanan yöntemler kümesi” olarak tanımlar. Makine öğrenmesi çeşitleri denetimli öğrenme, denetimsiz öğrenme ve pekiştirmeli öğrenme olmak üzere üçe ayrılır (Russel ve Norvig 2022).

Denetimli öğrenmenin hedefi, $D = \{(x_i, y_i)\}$, $1 \leq i \leq N$ şeklindeki giriş-çıkış ikililerinin kümesini kullanarak girişler ve çıkışlar arasındaki ilişkiyi öğrenmektir (Murphy 2012). Burada D eğitim kümesi, x_i özellikler vektörü, y_i bu özelliklere karşılık gelen etiket ve N veri setinin büyüklüğüdür. Eğer etiketler kategorik ise (sınırlı sayıda ayrık değerlerden oluşuyorsa) bu bir sınıflandırma problemidir. Eğer etiketler reel sayı ise bu durumda problem regresyon problemidir. Sınıflandırma problemlerine örnek olarak bir haber makalesinin hangi kategoride olduğunu (gezi, spor, politika vb.) belirlenmesi verilebilir. Bu problemde eğitim kümesinin her elemanı bir makaleden ve bu makalenin bir insan tarafından etiketlenmiş kategorisinden meydana gelir. Regresyon problemine bir örnek ise konut fiyatlarının tahminidir. Bu problemde x_i konutun konumu,

yaşı, oda sayısı vb. özelliklerden meydana gelir. y_i ise ilgili konutun gözlemlenmiş fiyatıdır.

Denetimsiz öğrenmede, etiketlerin olmadığı veri seti içerisindeki örüntüler keşfedilir. Denetimsiz öğrenmenin bir örneği kümeleme problemidir. Kümeleme probleminin amacı birbirine benzer olan veri noktalarının gruplanmasıdır. Kümelemenin uygulama alanlarından biri görüntü sıkıştırma. Çok sayıda renk içeren bir görüntünün birbirine yakın olan piksellerin kümelenmesi ve küme merkezlerinin bu pikseller için yeni değerler olarak kullanılmasıyla orijinal görüntüden az sayıda renk içeren sıkıştırılmış görüntü elde edilir.

Makine öğrenmesinin başka bir tipi de pekiştirmeli öğrenmedir. Pekiştirmeli öğrenmede yapay zekâ ajanı, çevresiyle girdiği etkileşimlerden aldığı geri dönüş sinyalleriyle (ödül veya ceza) kendini geliştirir. Örneğin amacın en yüksek skora ulaşmak olduğu bir atari oyununu oynayan pekiştirmeli öğrenme ajanı, zaman içerisinde içinde bulunduğu bağlama göre skorunu artıracak ve oyunu kaybettirecek hareketlerden kaçındıracak eylemleri yapmayı öğrenir.

Makine öğrenmesi veri setlerini hazırlamak ve eğitime uygun bir hale getirmek için bazı adımlar izlenir. Öncelikle keşifsel veri analizi gerçekleştirilerek veri istatistikleri ve grafiklerle anlamlandırılır ve keşfedilir. Veri setindeki kategorik değerler sayısal hale getirilerek kodlanır. Örneğin özelliklerinden biri “hava durumu” olan ve bu özelliği “güneşli”, “yağmurlu” ve “kapalı” seçeneklerinden birini alan bir veri setinde, bu özellik her biri ikili değerler alan üç sütun ile kodlanabilir. Bu durumda veri setinin her elemanı için bu üç sütundan biri 1, diğer iki sütun 0 değerini alır. Bu şekilde gerçekleştirilen kodlamaya “one-hot encoding” ismi verilir.

Bir veri setinde her özellik çok farklı ölçeklere sahip olabilir. Bu özellikleri birbirine benzer ölçeğe çevirmek için normalizasyon teknikleri uygulanır. Kullanılabilecek en basit normalizasyon tekniği, özelliğin aldığı en düşük ve en büyük değer arasında doğrusal interpolasyon gerçekleştirmektir. Bu işlem tüm verileri 0-1 arasına ölçekler. Yöntemin dezavantajı uç değerlerden fazla etkilenmesidir. En popüler normalizasyon

yöntemlerinden biri standardizasyondur. Bu yöntemde veri noktalarından özelliğin ortalaması çıkarılır ve sonuç standart sapmaya bölünür. Böylece özellik için ortalaması 0 ve standart sapması 1 olan bir dağılım elde edilir.

Makine öğrenmesinde uç değerler, özelliğin aldığı normal değerlerden önemli ölçüde farklılık gösteren değerleri ifade eder. Uç değerler veri toplama sürecindeki hatalardan kaynaklanabilir. Örneğin bir hastane yönetim sisteminde hasta bilgilerini kaydeden çalışan, hasta yaşına “50” yerine yanlışlıkla “500” girebilir. Bu tür uç değerleri tespit edip ayıklamak için özelliğin ortalama değerinden belirlenen standart sapma uzaklıktaki değerler veri setinden çıkarılabilir.

Veri setlerinde karşılaşılan sorunlardan biri de veri seti dengesizliği problemidir. Bu problem, belirli sınıflara ait örneklerin veri setinin çoğunluğunu oluştururken bazı sınıfların ise yeterli derecede temsil edilmemesiyle oluşur. Bu sorunu çözmek için çoğunluk sınıfın örnekleri alt örneklenebilir, ancak bu seçim veri kaybıyla sonuçlanır. Başka bir seçenek az temsil edilen sınıflara ait örneklerin kopyalanarak veya diğer örneklere benzer sentetik örnekler oluşturularak aşırı örneklenmesidir. Aşırı örnekleme aşırı uyum ile sonuçlanabilir.

Makine öğrenmesi yöntemleriyle öğrenilen fonksiyonların başarısı eğitim setinden bağımsız olan ancak eğitim setiyle aynı veri dağılımından gelen ikinci bir veri seti kullanılarak değerlendirilir. Eğer model eğitim setinde yüksek başarı gösterip test setinde başarısız oluyorsa bu duruma aşırı uyum (overfitting) denilir. Aşırı uyum modelin iyi bir şekilde genelleşmediğini, yani daha önceden görmediği veri üzerindeki başarısının düşük olduğunu gösterir. Aşırı uyumun önüne geçmek için düzenleme (regularization) teknikleri kullanılabilir veya model karmaşıklığı azaltılabilir.

Makine öğrenmesi yöntemlerinin test setindeki başarısı çeşitli performans ölçütleriyle değerlendirilir. Regresyon problemlerinde sıklıkla kullanılan değerlendirme ölçütlerinden biri ortalama karesel hatadır (MSE).

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.1)$$

MSE, veri setindeki her örnek için gerçek değer y_i ile modelin tahmin ettiği değer $\hat{y}_i$ arasındaki farkın karesini hesaplayarak bu karelerin ortalama değerini bulur (Denklem 2.1).

Regresyon problemlerinde kullanılan bir başka değerlendirme ölçütü ortalama mutlak hatadır (MAE). Ortalama mutlak hata, tahmin edilen değer ile gerçek değer arasındaki farkların mutlak değerlerinin ortalaması alınarak bulunur (Denklem 2.2). MSE gerçek değerden fazla miktarda sapan tahminleri şiddetle cezalandırdığından, MSE ortalama mutlak hataya göre uç değerlere daha duyarlıdır.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.2)$$

Sınıflandırma problemlerinde modelin tahminleri ile gerçek değerler arasındaki fark karmaşıklık matrisi ile ifade edilir. Karmaşıklık matrisinin yapısı şekil 2.3'teki gibidir.

		Gerçek	
		Pozitif	Negatif
Tahmin	Pozitif	TP	FP
	Negatif	FN	TN

Şekil 2.3 İkili bir sınıflandırma problemi için karmaşıklık matrisi

Karmaşıklık matrisinde TP (True positive), modelin pozitif olarak tahmin ettiği ve gerçekte de pozitif olan örneklerin sayısıdır. FP (False positive) modelin pozitif olarak

tahmin ettiđi ama gerek etiketi negatif olan rneklerin sayıdır. FN (False negative) modelin negatif olarak tahmin ettiđi ama gerekte pozitif olan rnek sayıdır. Son olarak TN (True negative) modelin negatif olarak tahmin ettiđi ve gerekte de negatif olan rneklerin sayıdır.

Karmařık matrisindeki deđerler kullanılarak hesaplanan sınıflandırma performansı ltlerinden bazıları izelge 2.1’de verilmiřtir.

Dođruluk (accuracy) deđerı modelin dođru tahmin yapma olasılıđıdır. Bu deđer veri seti dengesizliđinin fazla olduđu durumlarda yanıltıcı sonular verebilir. rneđin veri rneklerinin %98’inin negatif olduđu bir veri setinde, karřılařtıđı tm rnekleri negatif olarak sınıflandıran bir modelin dođruluk deđerı %98 olacaktır. Byle bir durumda dođruluk deđerine deđil F1 skoru gibi diđer metriklere bakılmalıdır.

izelge 2.1 Makine đrenmesi performans ltleri

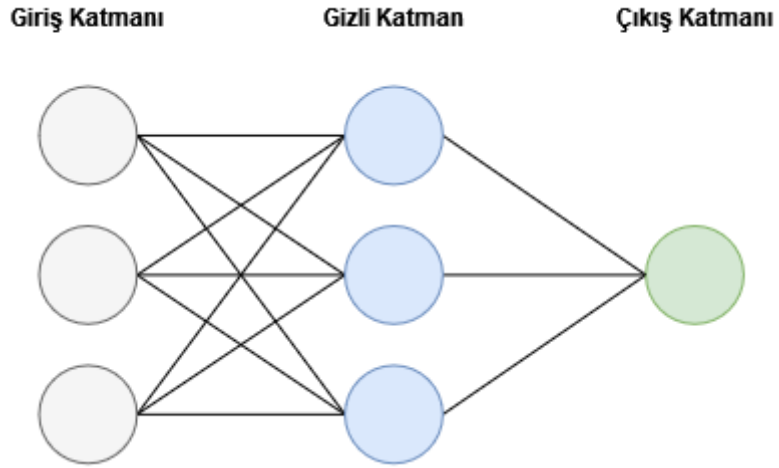
Performans lt	Forml
Accuracy (Dođruluk)	$\frac{TP + TN}{TP + TN + FP + FN}$
Precision (Kesinlik)	$\frac{TP}{TP + FP}$
Recall (Duyarlılık)	$\frac{TP}{TP + FN}$
F1 Score (F1 Skoru)	$\frac{2 * Precision * Recall}{Precision + Recall}$

Kesinlik (precision), modelin pozitif olarak sınıflandırdıđı rneklerin gerekten de pozitif olma olasılıđıdır. Duyarlılık (recall) modelin pozitif bir rneđi bařarıyla bulma olasılıđıdır. F1 Skoru ise kesinlik ve duyarlılık deđerlerinin harmonik ortalamasıdır.

2.2.2 Derin öğrenme

Derin öğrenme, Russel ve Norvig (2022) tarafından “ayarlanabilir, temel hesaplama öğeleri içeren çok sayıda katmanın kullanıldığı makine öğrenmesi türü” olarak tanımlanmıştır. Karmaşık problemleri “kavramlar hiyerarşisi” oluşturarak çözen derin öğrenme yöntemleri ile bilgisayarlar karmaşık bilgi temsillerini daha basit temsiller kullanarak inşa eder (Goodfellow vd. 2016).

Derin öğrenmenin ana konusunu yapay sinir ağları oluşturur. Giriş katmanı, çıkış katmanı ve bir ya da daha fazla gizli katmandan meydana gelen ileri beslemeli yapay sinir ağı yapısına multilayer perceptron (MLP) adı verilir. Şekil 2.4’te tek gizli katman içeren bir MLP’nin yapısı görülmektedir.



Şekil 2.4 Tek gizli katmana sahip bir MLP’nin yapısı

Bir MLP’de art arda gelen iki katman arasındaki hesaplama birimleri (nöronlar) birbirine tam bağlıdır; başka bir deyişle bir nöronun kendisinden sonra gelen katmandaki tüm nöronlarla bağlantısı vardır. Her bağlantının bir ağırlık değeri bulunmaktadır. Bir nöronun çıkış değeri (aktivasyonu), kendisine bağlı olan nöronların çıkış değerleri ve ağırlıkları kullanılarak hesaplanır. Denklem 2.3’te bir MLP’nin $k+1$ ’inci katmanının aktivasyonlarının hesaplanması verilmiştir. Bu denklemde W katmana ait ağırlıklar matrisi ve z_k bir önceki katmanın aktivasyonlarıdır. σ doğrusal olmayan bir aktivasyon fonksiyonudur. Bu fonksiyonun yokluğunda modelin katmanları çökerek tek bir afin

dönüşüme eş değer hale gelir ve model giriş ve çıkışları arasındaki doğrusal olmayan ilişkileri öğrenemez. Bias değeri $\mathbf{b}$, aktivasyon fonksiyonunu kaydıran öğrenilebilir bir parametredir.

$$\mathbf{z}_{k+1} = \sigma(\mathbf{W}\mathbf{z}_k + \mathbf{b}) \quad (2.3)$$

Yapay sinir ağlarında kullanılan aktivasyon fonksiyonlarından biri sigmoid fonksiyonudur. Sigmoid fonksiyonu giriş olarak aldığı reel sayıyı 0 ile 1 arasında bir değere çevirir (Denklem 2.4).

$$\sigma(x) = \frac{1}{1+e^{-x}} \quad (2.4)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.5)$$

$$\text{ReLU}(x) = \max(0, x) \quad (2.6)$$

Bir başka aktivasyon fonksiyonu ise $\tanh$ fonksiyonudur. $\tanh$, giriş olarak aldığı reel sayıyı -1 ile 1 arasında bir değere çevirir (Denklem 2.5).

Son dönemlerde derin öğrenme ağlarında sıklıkla kullanılan Rectified Linear Unit (ReLU) aktivasyon fonksiyonu denklem 2.6'da verilmiştir (Glorot vd. 2011).

Bir MLP'nin ağırlıkları ve bias değerleri, o ağın öğrenilebilir parametrelerini oluşturur. Yapay sinir ağlarında kayıp fonksiyonu, ağın yaptığı tahminler ile gerçek değerler arasındaki farkı belirtir. Yapay sinir ağlarında öğrenmenin hedefi, kayıp fonksiyonunu minimize edecek parametreleri bulmaktır. Problemin türüne göre kayıp fonksiyonları değişkenlik gösterir. Regresyon problemlerinde kullanılabilecek kayıp fonksiyonlarından biri MSE'dir (Denklem 2.1). İkili sınıflandırma problemlerinde kullanılan Binary Cross Entropy (BCE) kayıp fonksiyonu denklem 2.7'deki gibidir.

$$BCE(y, \hat{y}) = -(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})) \quad (2.7)$$

Yapay sinir ağıları geriye yayılım algoritması ile eğitilir. Ağda ileri yayılımın ardından, zincir kuralının bir uygulaması olan geriye yayılım ile gradyan hesaplanır. Gradyan iniş ile kayıp fonksiyonunu en çok minimize eden yönde bir adım atılarak parametreler değiştirilir. Bu adımın büyüklüğünü öğrenme oranı α belirler.

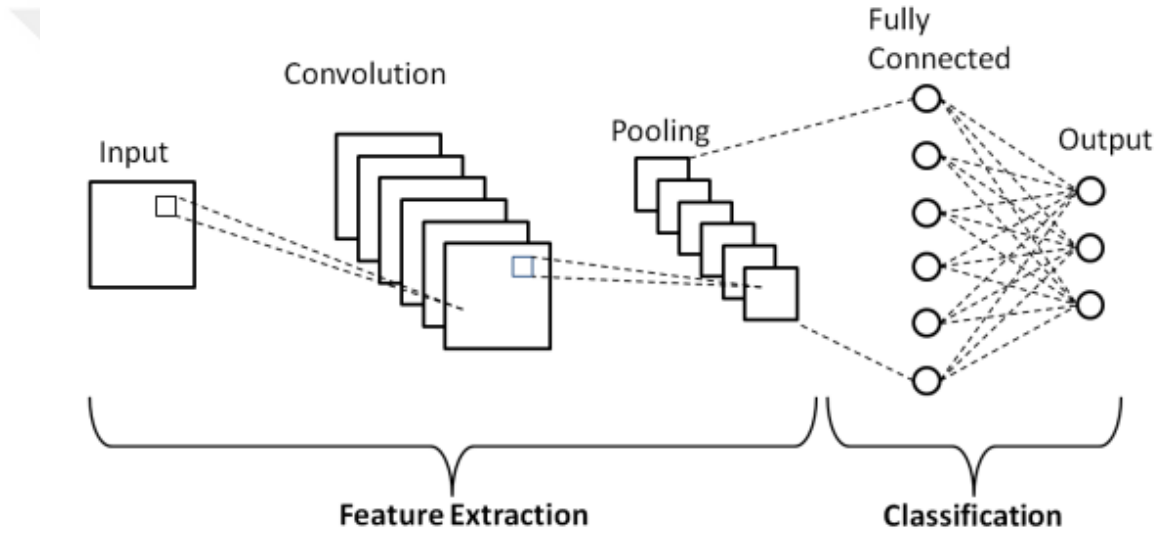
Yapay sinir ağlarında aşırı uyumun (overfitting) önüne geçmek için çeşitli yöntemler bulunmaktadır. Bunlardan biri Srivastava vd. (2014) tarafından önerilen Dropout yöntemidir. Dropout, eğitim sırasında yapay sinir ağının nöronlarının bazılarını bu nöronların bağlantıları ile birlikte ağdan çıkarır. Belirli bir olasılıkla gerçekleştirilen bu işlem, nöronların diğer nöronların hatalarını düzeltmek için değiştikleri karmaşık adaptasyonları engelleyerek aşırı uyumun önüne geçer (Srivastava vd. 2014).

Aşırı uyumu engellemenin bir başka yolu da erken durdurmadır. Yapay sinir ağlarının eğitimi sırasında bir noktadan sonra model eğitim setine aşırı uyum sağlamaya ve modelin genelleşme başarısı düşmeye başlar. Bunun önüne geçmek için modelin başarısı her devirden sonra doğrulama kümesi üzerinde değerlendirilir. Eğer modelin doğrulama kümesindeki başarısı artmıyorsa veya azalıyorsa eğitim erken durdurularak elde edilen model kullanılır.

Büyük ölçekli yapay sinir ağlarının eğitimi için gereken hesaplama kapasitesi oldukça maliyetlidir. Ayrıca büyük modellerin eğitimi zaman alır ve yüksek enerji harcamalarıyla karbondioksit salınımını artırır. Bu nedenlerden dolayı aktarımlı öğrenme (transfer learning) yöntemi popülerleşmiştir. Aktarımlı öğrenme ile önceden eğitilmiş bir model, amaca uygun bir şekilde adapte edilir. Örneğin bir sınıflandırma problemi için eğitilmiş tipik bir CNN modelinin evrişimsel katmanları, farklı bir sınıfı tespit etmek üzere eğitim yapmak için yeniden kullanılabilir. Sadece modelin sonundaki tam bağlı katmanların yeniden eğitildiği bu durum sayesinde orijinal modelin özellik çıkarma konusunda önceden edindiği bilgiler yeni görev için kullanılır.

2.2.3 Evrişimsel sinir ağıları

Evrişimsel sinir ağıları (CNN), görüntüler gibi düzenli ızgara benzeri yapıya sahip verileri işlemek için özelleşmiş yapay sinir ağılarıdır (Goodfellow vd. 2016). Evrişimsel sinir ağıları katmanları, filtre ismi verilen ağırlıklar matrisini girişlerine uygular. Filtrelerin kayan pencere mantığıyla uygulanmasıyla özellik haritaları elde edilir. Bir evrişimsel katmanda uygulanan filtrelerin sayısı kadar özellik haritası elde edilir. Temel CNN yapısı şekil 2.5'te görülebilir. Tipik bir CNN tekrar eden evrişim ve ortaklama işlemlerini izleyen tam bağlı bir katmandan meydana gelir.

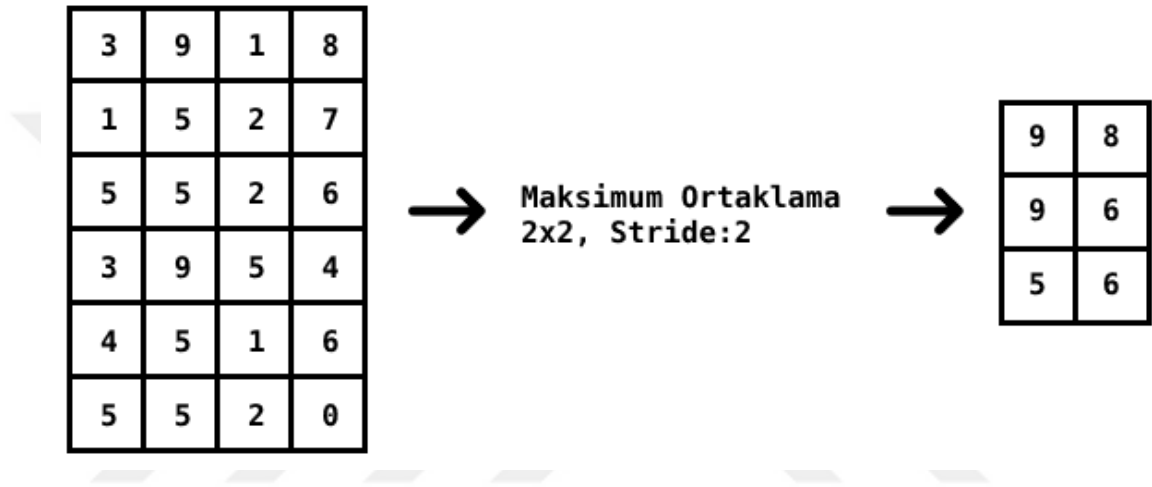


Şekil 2.5 Temel evrişimsel sinir ağı yapısı (Phung ve Rhee, 2019)

Evrişimsel sinir ağıları katmanlarının tasarımında belirlenmesi gereken çeşitli parametreler bulunmaktadır. Örneğin, filtre boyutu iki boyutlu görüntülerde yaygınlıkla 3x3 olarak seçilir. Adım aralığı (stride) pencerenin her seferinde ne kadar kaydırılacağını belirler. Doldurma miktarı (padding) giriş özellik haritalarının kenarlarına eklenecek boşlukların miktarını belirler.

Evrişimsel sinir ağlarında ortaklama (pooling) özellik haritalarının boyutunu düşüren bir örnekleme işlemidir. Ortaklama işlemi kayan pencereler aracılığıyla gerçekleştirilir. Eğer bir kayan penceredeki maksimum değer alınarak ortaklama yapılıyorsa buna maksimum

ortaklama ismi verilir. Eğer penceredeki değerlerin ortalaması alınıyorsa bunun adı ortalama ortaklamadır. Maksimum ortaklamanın bir özellik haritası üzerinde uygulandığı bir örnek şekil 2.6’da görülebilir. Bu örnekte maksimum ortaklama sonucunda orijinal boyutları 6x4 olan özellik haritası, maksimum ortaklama sonucunda 3x2 boyutuna indirilmiştir. Özellik haritası üzerinde kaydırılan 2x2 boyutundaki penceredeki en büyük değerler çıkış özellik haritasının değerlerini oluşturmuştur. Bu örnekte pencerenin adım aralığı 2 olarak belirlenmiştir.



Şekil 2.6 Evrişimsel sinir ağlarında maksimum ortaklama

2.2.4 Derin öğrenme kütüphaneleri

Derin öğrenme kütüphaneleri, sahip oldukları araçlar ve sundukları özellikler vasıtasıyla yapay sinir ağlarının tasarımını ve kullanımını kolaylaştırır. Bu kütüphanelerden bazıları çizelge 2.2’de görülebilir.

Çizelge 2.2 Derin öğrenme kütüphaneleri

Kütüphane	Çıkış Yılı	Geliştirici
Tensorflow	2015	Google
Keras	2015	François Chollet
PyTorch	2016	Meta
JAX	2018	Google

Tensorflow (Tensorflow 2024) Google tarafından geliştirilmiş açık kaynak kodlu derin öğrenme kütüphanesidir. Apache 2.0 lisansına sahip kütüphane, derin öğrenme dışındaki makine öğrenmesi uygulamaları için de kullanılabilir.

Keras (Keras 2024) Google mühendisi François Chollet tarafından geliştirilmiş, “insanlar için derin öğrenme” sloganına sahip derin öğrenme kütüphanesidir. Keras 3. sürümü itibariyle arka ucunda Tensorflow, PyTorch veya JAX’i kullanabilir.

PyTorch (PyTorch 2024) Meta AI tarafından geliştirilen bir derin öğrenme kütüphanesidir. Tensorflow gibi bu kütüphane de derin öğrenme dışında makine öğrenmesi uygulamalarında kullanılabilir. PyTorch ile tensör ismi verilen çok boyutlu diziler üzerinde işlemler yapılır. Tensörler üzerinde yapılan işlemler ekran kartı ile hızlandırılabilir. Pytorch 2, Just-in-time derleyicisi desteği ile programcılara daha yüksek performans sunar.

JAX (JAX 2024) Google tarafından geliştirilmiş düşük seviyeli derin öğrenme ve sayısal hesaplama kütüphanesidir. JAX, yüksek performans gerektiren sayısal hesaplamalar ve makine öğrenmesi görevlerini desteklemek üzere geliştirilmiştir.

2.3 Bilgisayarlı Görü

Bilgisayarlı görü, bilgisayarların görüntüleri anlamlandırarak yararlı bilgiler çıkarmasına olanak sağlayan yapay zekâ dalıdır. Bilgisayarlı görünün uygulama alanlarından bazıları çizelge 2.3’te bulunmaktadır.

Optik karakter tanıma (OCR) sayesinde bilgisayarlar dijital görüntülerdeki el yazılarını veya basılı yazıları metne çevirirler. OCR sayesinde basılı belgelerdeki yazılar veya bir defterdeki el yazıları dijital ortama aktarılabilir. OCR’ın bir başka kullanım alanı da araç plakalarının tanınmasıdır. Üniversite kampüsü veya şirket otoparkı gibi alanların giriş kapılarında bulunan OCR sistemleri, yaklaşan araçların plakalarını otomatik olarak

okuyup veri tabanında kayıtlı plakalarla karşılaştırırlar. Böylece plakası kayıtlı olan araçların geçişi otomatik olarak sağlanır.

Çizelge 2.3 Bilgisayarlı görü uygulama alanları (Szeliski 2022)

Uygulama Alanı	Örnek
Optik karakter tanıma	Araç plakalarının okunması
Makine incelemesi	Hatalı ürünlerin tespiti
Otonom arabalar	Kendi kendini süren araçlar
Tıbbi görüntüleme	Tümör tespiti
Yüz tespiti	Kamera odaklaması için yüz tespiti
Fotogrametri	İHA görüntülerinden 3B modeller oluşturulması
Gözetleme	Boğulmakta olan insanların tespiti

Makine incelemesi ile bir üretim hattındaki ürünler otomatik olarak incelenir. İnceleme sonucunda hangi ürünlerin hatalı olduğu tespit edilerek kalite kontrolü gerçekleştirilir. Üretim hattındaki ürünlerin otomatik sayımı da bilgisayarlı görü ile gerçekleştirilebilir.

Otonom araçlar, bir insan tarafından kontrol edilmeye ihtiyaç duymadan belirlenen hedeflere doğru yol alabilen arabalardır. Otonom araçlar sensörler aracılığıyla algıladıkları çevrelerindeki nesneleri ve bu nesnelerin davranışlarını tahmin ederek planlama yaparlar.

Bilgisayarlı görü ile X-ray ve ultrason gibi tıbbi görüntüler analiz edilebilir. Görüntü bölütleme ile tıbbi görüntüler, doku ve organ gibi kısımlara ayrılarak doktorların bu bölgeleri daha rahat bir şekilde tanımlayabilmeleri sağlanır. Görüntülerdeki anormal dokular otomatik olarak tespit edilebilir.

Yüz tespiti çok sayıda kullanım alanı bulunmaktadır. Biyometrik doğrulama alanında yüz tespiti, yüksek güvenli korunan bölgelere sadece yüzü tanımlı olan kişilerin erişebilmesini sağlar. Benzer şekilde akıllı telefon gibi kişisel cihazlar da yüz tespiti ile kullanıcı doğrulaması gerçekleştirerek sadece tanımlı kullanıcının cihaza erişebilmesini sağlar. Yüz tespiti kameralarda yüzlere otomatik odaklama yapmak için kullanılır. Sosyal medya uygulamaları yüz tespiti ile fotoğraftaki kişileri etiketleyip tanımlayarak kullanıcılara daha zengin bir tecrübe sunar.

Bilgisayarlı görüde fotogrametri yöntemleri ile farklı açılardan çekilmiş görüntülerden 3D bilgiler elde edilir. Fotogrametri ile tarihi eserlerin veya binaların 3D görüntüleri elde edilebilir.

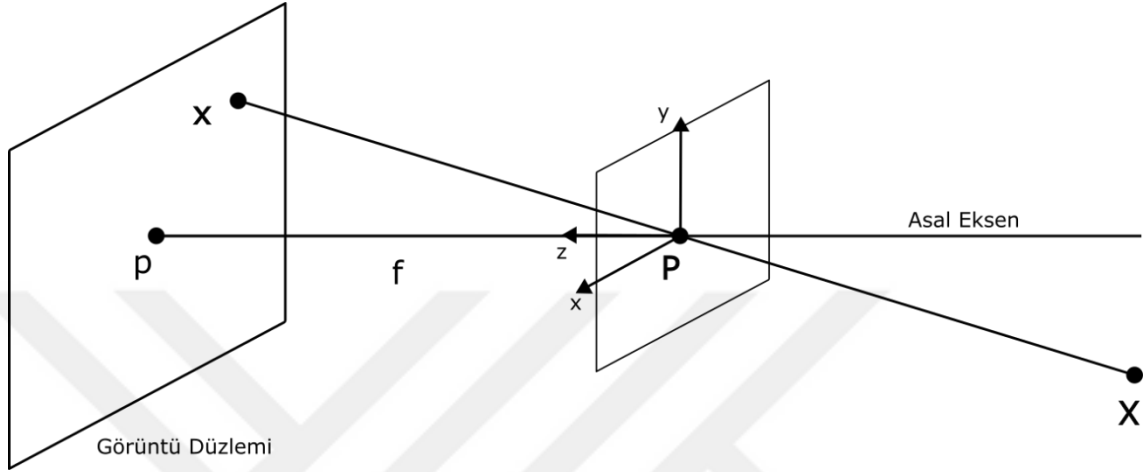
Bilgisayarlı görünün uygulama alanlarından biri de insan hareketlerini ve davranışlarını gözetlemek ve incelemektir. Kameralardan elde edilen görüntülerdeki insanlar tespit edilip sayılabilir. Bu insanların ne yaptıkları davranış tespiti ile otomatik olarak tanınabilir. Elde edilen bilgilerden yararlanılarak insanların güvenliği sağlanır veya insanlara daha verimli bir şekilde hizmet sunulur.

2.3.1 İğne deliği kamera modeli

İğne deliği kamera modeli, 3 boyutlu uzaydaki noktaların 2 boyutlu görüntü düzlemindeki yansımalarının oluşumunu tanımlayan bir kamera modelidir. Lenslerin bulunmadığı bu modelde ışığın girdiği açıklık “iğne deliği” ismi verilen bir noktadan ibarettir. İğne deliği modeli ile görüntü oluşumu tarih boyunca “karanlık oda” düzenekleri ile gözlemlenmiştir. Bu düzenekte ön tarafında küçük bir delik bulunan kapalı bir kutu veya oda kullanılır. Kutuya giren ışık, kutunun arka tarafında görüntüyü oluşturur. Optik alanında yaptığı çalışmalar ile tanınan 10. yüzyılda yaşamış bilim insanı İbn-i Heysem, Kitāb el-Manāzir isimli eserinde açıkladığı karanlık oda deneylerine dayanarak ışığın doğrusal olarak yayıldığını belirtmiştir (Topdemir 1998).

Şekil 2.7’de iğne deliği kamera modelinde görüntü oluşumu görülmektedir. İğne deliği P , kameranın merkezidir ve görüntü düzlemine düşen ışınlar bu noktadan geçer. Görüntü

düzlemi ile kamera merkezi arasındaki uzaklığa odak uzunluğu adı verilir. Diyagramda bu uzunluk f ile gösterilmiştir. Kamera merkezini görüntü düzlemiyle birleştiren eksene asal eksen ismi verilir. 3D uzaydaki X noktasının düzlemdeki iz düşümü olan x 'in koordinatları benzer üçgenlerden yararlanılarak denklem 2.8'deki gibi bulunur.



Şekil 2.7 İğne deliği kamera modeli

$$x = \left(f \frac{X_x}{X_z}, f \frac{X_y}{X_z}, f \frac{X_z}{X_z} \right)^T \quad (2.8)$$

Projeksiyonun bu şekilde gerçekleşmesi, kameradan uzaklaşan nesnelerin görüntülerinin giderek ufalması olarak açıklanabilecek kısalma etkisi ile sonuçlanır. Buna örnek olarak tren raylarının kameradan uzaklaştıkça küçülüp bir noktada kaybolması verilebilir. Perspektif iz düşümde tren raylarının kenarları gibi paralel doğru kümeleri, görüntüde tek bir noktada kavuşurlar. Bu noktalara birleşme noktası adı verilir.

Uzaydaki bir noktanın projeksiyonunu yapabilmek için önce koordinat sistemi dönüşümü yapılarak noktanın kamera merkezine göre koordinatları tespit edilmelidir. Kamera koordinatları bulunan noktanın iz düşümünün koordinatları, içsel matrisin homojen kamera koordinatlarıyla çarpımı sonucu elde edilir. İçsel matris denklem 2.9'da verilmiştir.

$$M_{i\check{c}} = \begin{bmatrix} f_x & 0 & c_x & 0 \\ 0 & f_y & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (2.9)$$

İçsel matriste f_x ve f_y piksel cinsinden odak uzunluklarıdır. Asal eksenin görüntü sensörünü kestiği noktanın koordinatları c_x ve c_y ile belirtilmiştir.

Uzaydaki bir noktayı kamera koordinatlarına çeviren dışsal matris denklem 2.10'da verilmiştir. Dışsal matris, kameranın konumu ve yönünü ifade eden dışsal parametrelerden meydana gelir.

$$M_{dış} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.10)$$

$$x = M_{i\check{c}} M_{dış} X \quad (2.11)$$

Dışsal matris ve içsel matris dönüşümleri uzaydaki noktaya sırayla uygulanarak noktanın önce kamera koordinatlarına, sonra da görüntü koordinatlarına dönüşümü gerçekleştirilir (Denklem 2.11). İçsel matrisin dışsal matrisle çarpılarak birleştirilmesiyle denklem 2.12'de verilen projeksiyon matrisi elde edilir.

$$M_P = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (2.12)$$

$$x = M_P X \quad (2.13)$$

Projeksiyon matrisinin uzaydaki noktaya uygulanmasıyla noktanın iz düşümünün homojen koordinatları elde edilir (Denklem 2.13). Projeksiyon matrisinin değerleri bilinen bir kameranın içsel ve dışsal parametrelerine ulaşmak da mümkündür.

2.3.2 Kamera kalibrasyonu

Kamera kalibrasyonu, bir kameranın içsel ve dışsal parametrelerinin veya buna eş değer olarak projeksiyon matrisinin değerlerinin tespit edilmesidir. Kamera kalibrasyonunu gerçekleştirmek için gerçek ebatları bilinen bir nesne referans olarak kullanılır. Bunun için yaygınlıkla kullanılan nesnelerden biri, uzunlukları bilinen siyah ve beyaz karelerden oluşan dama tahtasıdır. Kalibre edilecek kamera kullanılarak nesnenin görüntüsü çekilir. Dünya koordinatları bilinen nesne üzerindeki noktalar ile görüntüye yansıyan ve dünya koordinatlarına karşılık gelen piksel koordinatları eşleştirilir. Böylece dünya koordinatları ve bunlara karşılık gelen görüntü koordinatları ikililerinden oluşan bir veri seti oluşur.

Elde edilen koordinat ikililerini kullanarak kamera kalibrasyonunu gerçekleştirmek için kullanılan yöntemlerden biri doğrudan doğrusal dönüşümdür (DLT). Projeksiyon matrisinin değerlerini DLT ile bulmak için ilk adım boyutları bilinen bir nesne yardımıyla en az 6 adet dünya-görüntü koordinat ikilisi belirlemektir (Denklem 2.14). Belirlenen örneklerin hepsi dünyada aynı düzlemde yer almamalıdır.

$$x_i = M_P X_i, \quad i \in [1, 2, \dots, N], \quad N \geq 6. \quad (2.14)$$

Projeksiyon matrisinin satırları sırasıyla A , B ve C sütun vektörleriyle ifade edilirse bu ikililer arasındaki ilişki denklem 2.15'te olduğu gibi gösterilir.

$$x_i = \begin{bmatrix} A^T X_i \\ B^T X_i \\ C^T X_i \end{bmatrix} \quad (2.15)$$

Bu durumda belirlenen her nokta ikilisinden iki adet denklem elde edilir (Denklem 2.16). Bu denklemler bir doğrusal denklem sistemi oluşturur.

$$x_{xi} = \frac{A^T X_i}{C^T X_i}, \quad x_{yi} = \frac{B^T X_i}{C^T X_i}, \quad (2.16)$$

Projeksiyon matrisinin 12 elemanı bir p sütun vektörüne yerleştirilirse bu denklem sistemi denklem 2.17'deki gibi ifade edilir.

$$\begin{aligned} \mathbf{a}_{x_{xi}}^T \mathbf{p} &= 0, & \mathbf{a}_{x_{xi}}^T &= [-\mathbf{X}_i^T, \mathbf{0}^T, x_i \mathbf{X}_i^T] \\ \mathbf{a}_{x_{yi}}^T \mathbf{p} &= 0, & \mathbf{a}_{x_{yi}}^T &= [\mathbf{0}^T, -\mathbf{X}_i^T, y_i \mathbf{X}_i^T] \end{aligned} \quad (2.17)$$

Her nokta ikilisinden elde edilen $\mathbf{a}_{x_{xi}}^T$ ve $\mathbf{a}_{x_{yi}}^T$ vektörleri bir M matrisinde birleştirilirse denklem 2.18'deki eşitlik elde edilir.

$$\mathbf{M} \mathbf{p} = \mathbf{0} \quad (2.18)$$

Burada M $2N \times 12$ boyutunda bir matris, p ise projeksiyon matrisinin değerlerini taşıyan 12×1 boyutundaki vektördür.

Denklem 2.18'deki denklem sistemini çözmek için M 'nin sıfır uzayından bir vektör bulunmalıdır. Bunun için tekil değer ayrışımı (SVD) yöntemi kullanılır. SVD ile M matrisi üç matrisin çarpımı haline gelecek şekilde parçalarına ayrılır (Denklem 2.19).

$$\mathbf{M} = \mathbf{U} \mathbf{S} \mathbf{V}^T \quad (2.19)$$

Son olarak p vektörünün değerlerini bulmak için en küçük tekil değere karşılık gelen tekil vektör seçilir. Böylece projeksiyon matrisinin elemanları elde edilmiş ve kamera kalibrasyonu gerçekleştirilmiş olur.

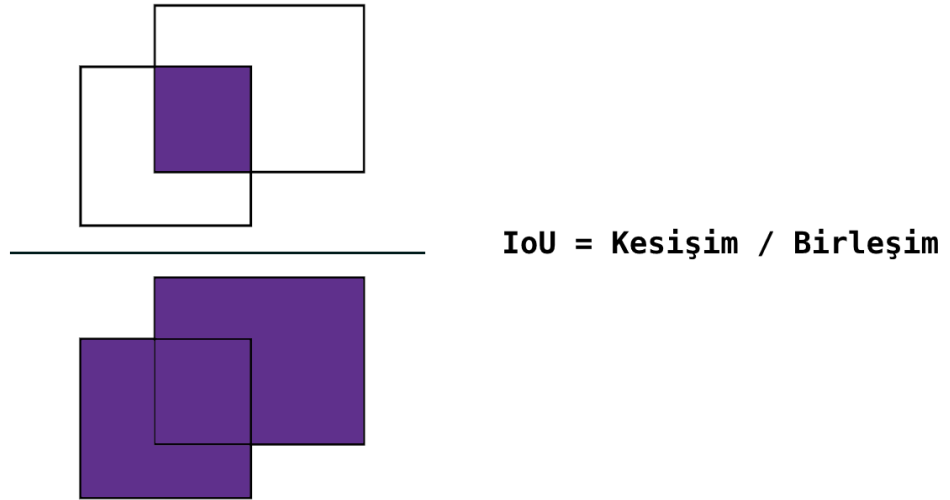
2.3.3 Nesne tespiti

Nesne tespiti bir görüntüdeki nesnelerin sınıflandırılmasını ve konumlarının belirlenmesini amaçlar. Nesne tespiti problemi görüntü sınıflandırma problemine göre daha zordur. Görüntü sınıflandırma probleminde bir görüntüye o görüntünün içeriğine göre kedi, köpek, uçak gibi sınıflar atanır. Nesne tespitinde ise görüntü üzerinde farklı

sınıflara ait nesnelerin örnekleri tespit edilerek sınıflandırılır ve bu nesnelerin sınırlayıcı kutularının konum ve ebatları belirlenerek konumlandırma yapılır.

Nesne tespiti modellerini eğitmek için kullanılan veri setleri, görüntü üzerindeki nesnelerin insanlar tarafından sınıflandırılıp kutu içerisine alınmasıyla oluşturulur. Etiketleme ismi verilen bu işlem için Computer Vision Annotation Tool (CVAT 2024) veya Image Annotation Lab (IALAB 2024) gibi özel yazılımlar kullanılır. Bu yazılımlar etiketleme sürecini hızlandırmak ve kolaylaştırmak için çeşitli özellikler sunarlar. Bu özelliklerden biri önceden eğitilmiş bir model ile otomatik etiketleme seçeneğidir. Otomatik etiketleme ile görüntülerdeki nesneler yazılım tarafından etiketlendikten sonra insanlar görüntülerin üzerinden geçerek yazılımın yaptığı hataları düzeltir ve eksik etiketleri ekler. Böylece etiketleme işlemi önemli bir şekilde hızlanmış olur.

Bir nesnenin model tarafından tahmin edilen sınırlayıcı kutusunun gerçek sınırlayıcı kutuyla ne kadar iyi eşleştiğini değerlendirmek için IoU (Intersection Over Union) ölçütü kullanılır. Bu ölçütü hesaplamak için karşılaştırılmak istenen iki sınırlayıcı kutunun alanlarının kesişimi, bu sınırlayıcı kutuların alanlarının birleşimine bölünür (Şekil 2.8).



Şekil 2.8 Intersection Over Union ölçütünün hesaplanması

IoU ölçütü 0 ile 1 arasında değerler alır. Eğer iki kutu tamamıyla örtüşüyorsa IoU değeri 1, iki kutu birbirinden tamamıyla ayrıysa IoU değeri 0 olur.

Nesne tespiti modellerinin başarısını değerlendirmek için mAP (Mean Average Precision) ölçütü kullanılır. mAP, her sınıfın ortalama kesinliğinin ortalaması alınarak bulunur (Denklem 2.20). Ortalama kesinlik (AP) farklı eşik değerleri için kesinlik ve duyarlılık değerlerinin değişimini gösteren kesinlik-duyarlılık eğrisinin altındaki alanı ifade eder.

$$mAP = \sum_c AP_c \quad (2.20)$$

Bir nesne tespiti modeli görüntüde çok sayıda birbirine benzer ve gereksiz sınırlayıcı kutu tahmin edebilir. Nesne tespitinin kalitesini iyileştirmek ve birbirine benzer kutuları elemek için NMS (Non Maximum Suppression) algoritması kullanılır. NMS algoritmasının adımları çizelge 2.4'te verilmiştir. NMS algoritmasının adımları her sınıf için ayrı ayrı veya sınıf gözetmeksizin uygulanabilir.

Çizelge 2.4 NMS algoritmasının adımları

Non Maximum Suppression (NMS):	
Giriş: Sınırlayıcı kutular ve bu kutulara ait güven değerleri, IoU eşik değeri.	
Çıkış: Filtrelenmiş sınırlayıcı kutular.	
Adım	Açıklama
1	Sınırlayıcı kutuların güven değerlerine göre büyükten küçüğe sıralandığı X dizisini oluştur. Filtrelenmiş kutuları tutacak Y dizisini oluştur.
2	X dizisi boşsa algoritmayı sonlandır. Boş değilse X'ten en büyük güven değerine sahip kutuyu çıkar ve Y dizisine ekle.
3	2. adımda çıkarılan kutunun X'teki kutularla IoU değerini hesapla.
4	IoU değeri eşik değerini aşan kutuları X'ten çıkar.
5.	2. adıma git.

NMS algoritmasında kullanılan IoU eşik değerinin çok düşük seçilmesi, algoritmanın agresif bir şekilde filtreleme yapmasına neden olur. Karşılaştırma yapılan kutuyla biraz da olsa eşleşen kutular elenir. Bu durum görüntüde birbirine yakın olan nesnelerin sınırlayıcı kutularının filtrelenmesine sebep olabilir. IoU eşiği çok yüksek seçilirse de algoritmanın etkisi azalır ve sadece yüksek derecede eşleşen kutular elenir. Bu durumda aynı nesneye ait kopya sınırlayıcı kutular elenmeyebilir. IoU eşiği seçilirken bu değiş-tokuş göz önünde bulundurularak dengeli bir değer seçilmelidir.

Nesne tespiti problemini çözmek için çeşitli yöntemler bulunmaktadır. Bu yöntemlerden biri evrişimsel sinir ağlarına dayanan RCNN (Region Based Convolutional Neural Network) sistemidir (Girshick vd. 2014). Bu sistem harici bir algoritma tarafından üretilen bölge önerilerini kullanır. CNN'e verilmek üzere ölçeklenen bölge önerileri giriş olarak kullanılarak bir evrişimsel sinir ağı ile her bölge için 4096 boyutlu özellik vektörü çıkarılır. Bölgelere ait özellik vektörlerini sınıflandırmak için sınıfa özel SVM modelleri kullanılır.

Girshick (2015) RCNN mimarisini iyileştirerek Fast-RCNN sistemini geliştirmiştir. RCNN gibi bölge önerilerini kullanan bu sistem, RCNN'in aksine tek aşamada eğitilir. Fast-RCNN önce görüntüyü bir evrişimsel sinir ağı ile işleyerek bir özellik haritası elde eder. Görüntünün özellik haritası kullanılarak her bölge önerisi için sabit boyutta küçük özellik haritaları elde edilir. Bu özellik haritaları tam bağlı katmanlar ile özellik vektörlerine dönüştürülür. Özellik vektörlerini kullanan ağ iki adet çıkışa sahiptir. Bu çıkışlardan ilki sınıf olasılıklarını verirken, ikinci çıkış sınırlayıcı kutu ofsetlerini verir.

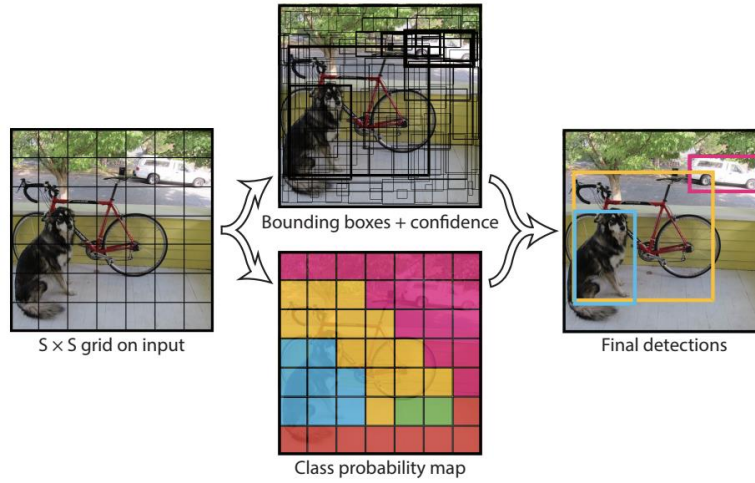
Ren vd. (2015) Fast-RCNN'i daha da iyileştirerek Faster-RCNN'i ortaya çıkarmıştır. Bu sistemde bölge önerileri harici bir algoritma ile üretilmek yerine "bölge önerisi ağı" tarafından üretilir. Hem bölge önerisi ağı, hem de önerileri kullanarak tespit yapan dedektör görüntüden evrişimsel sinir ağı ile çıkarılan özellik haritalarını kullanır. Bu özellik haritalarının ortak olarak kullanımı yapılan hesaplamaları azaltarak modeli hızlandırır.

YOLO (You Only Look Once), ilk iterasyonu Redmon vd. (2015) tarafından gerçekleştirilen ve gerçek zamanlı uygulamalarda kullanılan nesne tespiti modelleri ailesidir. Çizelge 2.5'te yıllar içerisinde ortaya çıkmış YOLO modellerine ilişkin bilgiler verilmiştir.

Çizelge 2.5 YOLO nesne tespiti yöntemi modelleri

Yöntem	Yazar/Geliştirici	Yıl	COCO mAP50-95
YOLO	Redmon vd.	2015	-
YOLOv2	Redmon ve Farhadi	2016	21,6
YOLOv3	Redmon ve Farhadi	2018	33,0
YOLOv4	Bochkovskiy vd.	2020	43,5
YOLOv5	Ultralytics	2020	50,7
YOLOv6	Meituan	2022	52,3
YOLOv7	Wang vd.	2022	52,9
YOLOv8	Ultralytics	2023	53,9

Bölge üretimi ve nesne tespiti olmak üzere iki aşamadan oluşan RCNN yöntemlerinin aksine, YOLO tek aşamalı bir nesne tespiti yöntemidir. Görüntünün üzerinden yalnızca bir kez geçen YOLO, nesne tespitini görüntüdeki sınırlayıcı kutuların ve bu kutulara ait olasılıkların regresyonu olarak ele alır (Redmon vd. 2015).



Şekil 2.9 YOLO sisteminin genel yapısı (Redmon vd. 2015)

YOLO sisteminin genel yapısı şekil 2.9’da görülmektedir. Sistem nesne tespiti gerçekleştirilecek görüntüyü $S \times S$ boyutunda bir ızgaraya ayırır. Izgaranın her hücresi için sınıf olasılıkları tahmin edilir. Bu olasılıklar, hücre içinde bir nesne olma koşulu altındadır. Bir hücre için yapılan koşullu sınıf olasılıkları tahminleri, o hücrede bulunan sınırlayıcı kutuların sayısından bağımsızdır. Her hücre için B adet sınırlayıcı kutu ve bu kutulara ait güven değerleri tahmin edilir. Böylece her hücrede merkezi o hücrede bulunan nesnelerin sınırlayıcı kutuları tahmin edilir. Sınırlayıcı kutu güven değerleri, modelin sınırlayıcı kutunun doğruluğuna olan güvenini temsil eder. Kutu güven değeri, hücrede bir nesnenin bulunma olasılığı ile tahmin edilen kutu ve gerçek kutu arasındaki IoU değerinin çarpımı olarak tanımlanmıştır. Model tahmin yaparken her kutunun güven değeri ve o kutunun bulunduğu hücrenin koşullu sınıf olasılıklarını çarparak o kutu için her sınıfa ait güven değerlerini elde eder.

Redmon vd. (2015) YOLO ağını 24 evrişimsel katman ve bunları izleyen 2 tam bağlı katman olarak tasarlamıştır. Ağı eğitmek için öncelikle ImageNet veri seti ile ön eğitim gerçekleştirilmiştir. Ön eğitim sonucunda ağın ilk 20 evrişimsel katmanı elde edilmiştir. Bu katmanlara 4 adet evrişimsel katman ve 2 adet tam bağlı katman ilave edilerek ağ nesne tespiti için gerekli yapıya kavuşturulmuştur. Ağın son katmanı hem sınırlayıcı kutuları, hem de sınıf olasılıklarını tahmin eder.

3. MATERYAL VE YÖNTEM

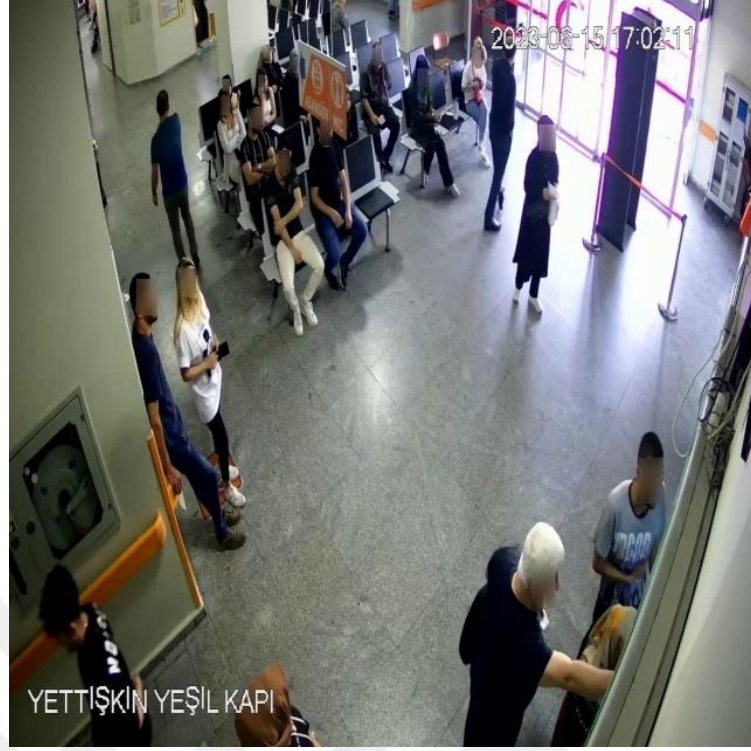
3.1 Veri Seti

Veri seti, Ankara’da bir özel hastane içerisindeki iki farklı alanda bekleyen insanların bulunduğu iki güvenlik kamerasından günün farklı saatlerinde çekilen video kliplerinden çıkarılan karelerden oluşmaktadır. Güvenlik kameralarından biri hastanenin “triya acil kayıt” bölümünün önünde bekleyen insanları, diğeri ise “yetişkin yeşil kapı” önünde bekleyen insanları göstermektedir. Söz konusu video klipleri, 15-06-2023 ve 22-06-2023 tarihleri arasında çekilmiştir ve her saat başına ait ikişer dakikalık bir klip bulunmaktadır. Video kliplerinin dosya tipi mp4 ve toplam boyutu 23,7 GB’tır. Bununla birlikte görüntülerin örneklenmesi ve yeniden ölçeklenmesi ile oluşturulan veri seti diskte sadece 91,1 MB yer tutmaktadır. Triya acil kayıt bölümü güvenlik kamerasından toplanan 168 video klipi 1920x1080 çözünürlüğünde ve 25FPS hızındadır. Yetişkin yeşil kapı bölümü güvenlik kamerası 168 video klipi ise 1280x960 çözünürlüğüne ve 10FPS değerine sahiptir. Veri setine ilişkin özet bilgiler çizelge 3.1’de yer almaktadır.

Çizelge 3.1 Veri seti özet bilgileri

Acil kayıt görüntü Sayısı	672
Yeşil kapı görüntü sayısı	672
Çözünürlük	640x640
Dosya Tipi	JPEG
Yetişkin kapı ortalama kişi sayısı	13
Acil kayıt ortalama kişi sayısı	7
Acil kayıt kuyruktaki insan oranı	%53

Veri setini oluşturmak için her klipten 30 saniye aralıklarla kareler alınmıştır. Böylece güvenlik kamerası başına 672 kare olmak üzere toplamda 1344 görüntü elde edilmiştir. Görüntüler, 640x640 boyutuna ölçeklenmiş ve JPEG formatında kaydedilmiştir.



Şekil 3.1 Yetişkin yeşil kapı bölgesinde bekleyen insanlar¹

Yoğunluk analizi için iki kameranın görüntüleri de etiketsiz olarak kullanılmıştır. Yetişkin yeşil kapıdan alınan bir kare şekil 3.1’de görülmektedir. Kuyruk tespiti için etiketlenmiş triyaj acil kayıt görüntüleri kullanılmıştır. Triyaj acil kayıt görüntülerini etiketlemek için Image Annotation Lab (IALAB 2024) yazılımı kullanılmıştır. Etiketleme iki aşamada gerçekleşmiştir. İlk aşamada yazılımın otomatik nesne tespiti modülü aracılığıyla insanlar tespit edilmiştir. İkinci aşamada ise etiketleme ekibi, tespit edilen insanları bir kuyruğun parçası olup olmamalarına göre “kuyrukta” veya “kuyrukta değil” olarak etiketlemiştir. Şekil 3.2’de bu şekilde etiketlenmiş bir görüntü görülebilir.

3.2 Kuyruk Tespiti

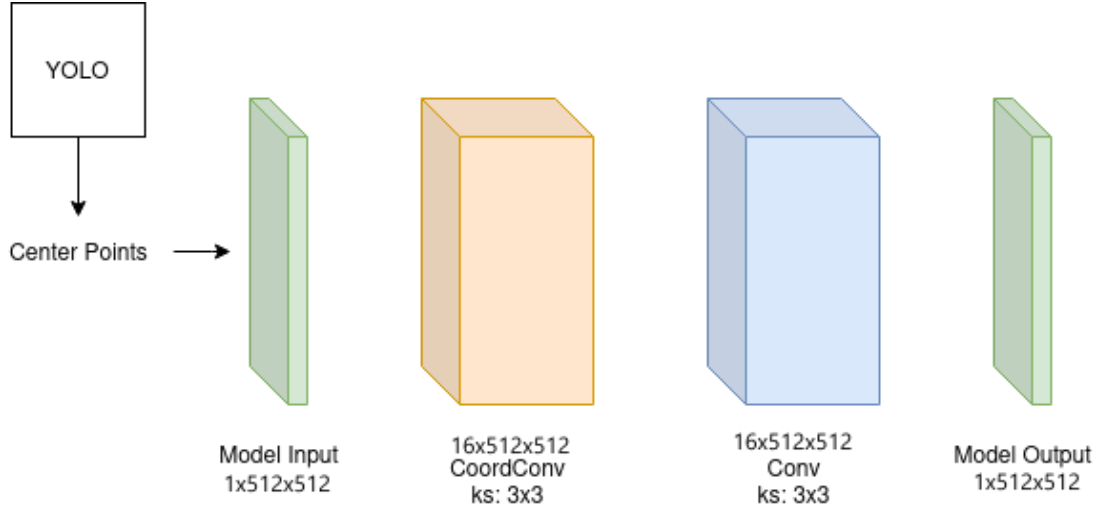
Kuyruk tespiti triyaj acil kayıt bölgesinde gerçekleştirilmiş ve kuyruk modelinin eğitimi için bu bölgeden alınan görüntüler kullanılmıştır.

¹ Kişilerin kimliklerini saklamak amacıyla yüzler bulanıklaştırılmıştır.



Şekil 3.2 Triyaj acil kayıt bölgesinden kuyruk tespiti için etiketlenmiş bir görüntü

Çalışma kapsamında kuyruk tespiti için tasarlanan model, evrimsel sinir ağlarına dayalı bir derin öğrenme modelidir. Modelin girişi, bir görüntüde YOLO ile tespit edilen insanların sınırlayıcı kutularının orta nokta koordinatlarının 512x512 boyutunda bir özellik haritasında işaretlenmesiyle oluşturulur. Modelin çıktısı da aynı şekilde 512x512 boyutunda bir özellik haritasıdır. Giriş özellik haritasında görüntüde tespit edilen her bireyin orta nokta koordinatları 1 yapılırken, modelin tahmin etmeye çalıştığı hedef özellik haritasında kuyrukta olmayan bireylerin koordinatlarındaki değer 0 yapılır. Eğitim ve doğrulama için 5-Kat Çapraz Doğrulamanın (5-Fold Cross Validation) kullanıldığı modelin mimarisi şekil 3.3'te gösterilmiştir. Bu mimariye çapraz doğrulama ile farklı model yapıları denenerek erişilmiştir.



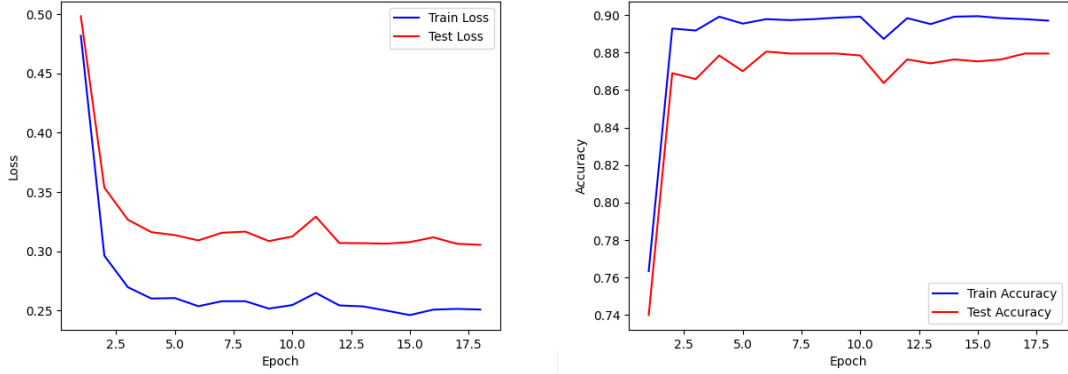
Şekil 3.3 Kuyruk tespiti modelinin yapısı

Model tahmin yapmak için koordinatları baz aldığından, başarıyı yükseltmek için ilk konvolüsyonel katmanda CoordConv (Liu vd. 2018) yapısı kullanılmıştır. CoordConv, evrişimden önce özellik haritalarına koordinat kanalları ekleyerek ağı giriş koordinatlarına erişebilmesini sağlar.

Model Python dilinde PyTorch kütüphanesi ile gerçekleştirilmiştir. Model 3073 eğitilebilir parametreye sahiptir. Evrişimsel katmanların filtre büyüklüğü 3, stride değeri 1 ve padding değeri “aynı” olarak belirlenmiştir. Aktivasyon fonksiyonu ReLU iken, modelin çıktısını elde etmek için her piksel için bir olasılık değeri üreten sigmoid fonksiyonu kullanılır. Böylece her çıkıştaki her piksel için o pikselde bulunan orta noktanın bir kuyruğa ait olma olasılığı elde edilir.

Eğitim sürecinde 10^{-3} öğrenme oranıyla Adam Optimizer kullanılmıştır. Kullanılan kayıp fonksiyonu Binary Cross Entropy'dir. 5-Kat Çapraz Doğrulama'nın her katında o kata ait eğitim ve doğrulama kümeleri kullanılarak yeni bir model eğitilmiştir. Her kat içinde eğitim, ilgili doğrulama kümesindeki kayıp art arda üç devir boyunca iyileşme göstermeyene kadar sürdürülmüş ve sonunda o kat için en düşük kayıp değerine sahip model seçilmiştir. Eğitim sürecinde modele giriş olarak verilen orta noktalar, Image Annotation Lab (IALAB 2024) ile etiketlenmiş sınırlayıcı kutuların orta noktalarıdır. Eğitilmiş model ile tahmin yapılırken ise YOLO ile bulunan orta nokta koordinatları

kullanılır. İlk katın eğitimine ait kayıp ve doğruluk grafikleri şekil 3.4’te verilmiştir. Eğitim sonucunda elde edilen sonuçlar çizelge 3.2’de verilmiştir.

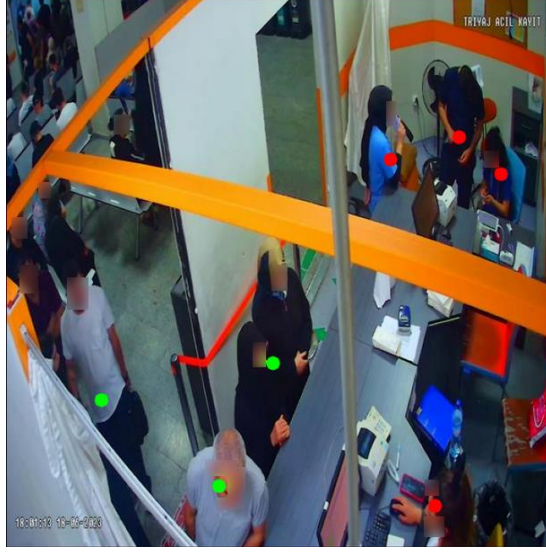


Şekil 3.4 Kuyruk modeline ait kayıp ve doğruluk grafikleri

Çizelge 3.2 Kuyruk modeli eğitim sonuçları

	Doğruluk	Kesinlik	Duyarlılık	F1 Skoru
Kat 1	0,88	0,86	0,93	0,89
Kat 2	0,89	0,87	0,92	0,90
Kat 3	0,90	0,88	0,93	0,91
Kat 4	0,91	0,90	0,95	0,92
Kat 5	0,91	0,87	0,96	0,92
Ortalama	0,90	0,87	0,94	0,91

Şekil 3.5’te eğitilmiş bir kuyruk tespit modelinin bir görüntü üzerinde çalıştırılmasıyla elde edilen çıktı görülmektedir. YOLO görüntüde altı kişi tespit etmiştir. Çıktıda görülen yeşil ve kırmızı noktalar, tespit edilen kişilerin sınırlayıcı kutularının orta noktalarına karşılık gelmektedir. Yeşil noktalar modelin “kuyrukta” olarak sınıflandırdığı kişiler, kırmızı noktalar ise modelin “kuyrukta değil” olarak sınıflandırdıklarıdır.

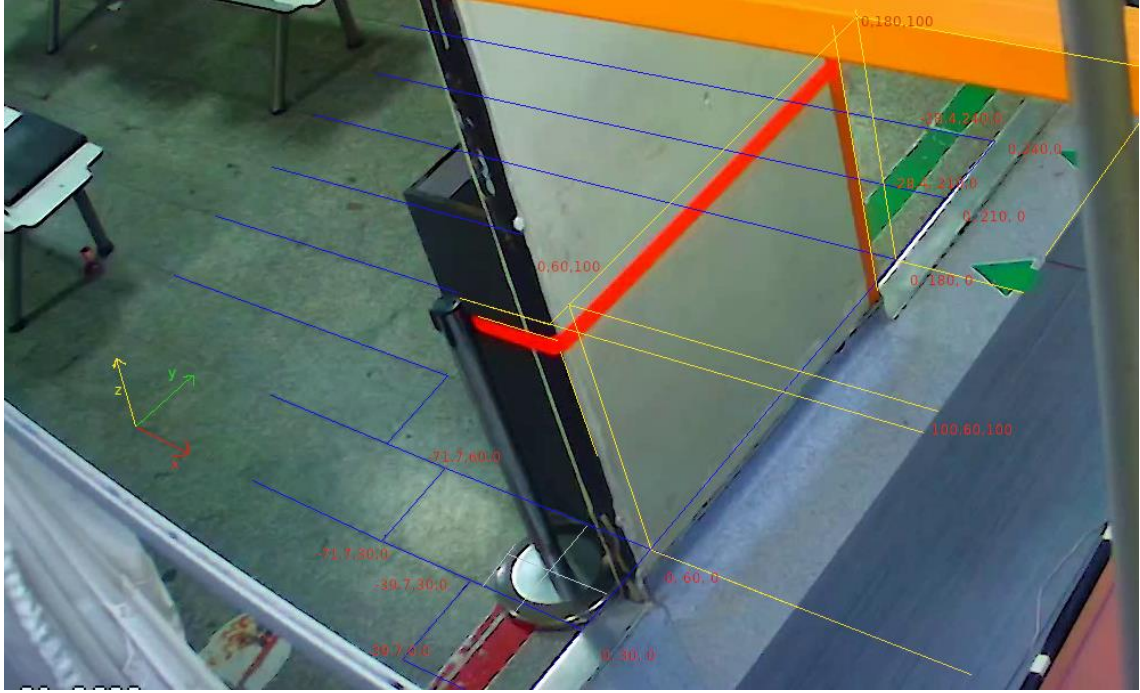


Şekil 3.5 Kuyruk modelinin bir görüntü üzerinde çalıştırılmasıyla elde edilen çıktı

3.3 Yoğunluk Analizi

Yoğunluk analizi her iki kameranın görüntüleri için de gerçekleştirilmiştir. Bölgelere ait serpilme grafiği ve gerçek dünya koordinatlarını kullanan kuş bakışı görünüm yoğunluk haritalarını çıkarabilmek için öncelikle her iki kamera için de kamera kalibrasyonu yapılarak projeksiyon matrislerinin değerleri bulunmuştur. Kamera kalibrasyonu için DLT yöntemi kullanılmıştır. DLT ile kalibrasyon için gerekli olan eşleştirilmiş görüntü düzlemindeki nokta-gerçek dünyadaki nokta ikilileri görüntüdeki görsel ipuçlarından yararlanılarak manuel olarak elde edilmiştir. Şekil 3.6’da triyaj acil kayıt kamerasının kalibrasyonu için bir görüntü editörü yazılımı olan GIMP ile işaretlenen noktalar görülmektedir. Nokta ikililerini belirlemek için öncelikle videolardan bölgede kimsenin olmadığı ve arka planın açıkça görüldüğü bir görüntü karesi belirlenmiş ve sonraki adımlarda bu görüntü kullanılmıştır. Gerçek dünya koordinat sisteminin eksenleri $z=0$ zeminde olacak şekilde belirlenmiştir. Sonra dünya koordinat sisteminin merkezi yerdeki karolardan birinin köşe noktası olarak seçilmiştir. Merkez noktası haricindeki diğer noktaları belirlemek için görüntüdeki paralel çizgiler ve nesneler referans alınmıştır. Örneğin, şekil 3.6’da kırmızı şeridi taşıyan metal direğin standart boyutları, internette yapılan kısa bir araştırma sonucu taban çapı 14 inç ve yüksekliği 1m olarak bulunmuştur. Bu nesnenin görüntü üzerinde çizilen piksel cinsinden taban uzunluğunu yine piksel cinsinden karo uzunluğuna bölerek y ekseninde iki karo arasındaki mesafenin yaklaşık

30cm olduğu belirlenmiştir. Diğer noktaları işaretlemek için aynı yöntemle birinin gerçek uzunluğu belirlenmiş olan iki doğru parçası oranlanarak ve görüntü üzerinde çizilen paralel çizgilerden yararlanılarak işaretleme yapılmıştır. Belirlenen gerçek dünya koordinatlarına karşılık gelen görüntü koordinatlarını bulmak için GIMP ile işaretlenen pikselin üzerine gelindiğinde okunan piksel koordinatı kullanılmıştır.



Şekil 3.6 Triyaj kamerasında kalibrasyon için nokta ikililerinin bulunması

Aynı yöntem kullanılarak yetişkin yeşil kapı kamerası için de nokta ikilileri bulunmuştur (Şekil 3.7). Dünya koordinat sisteminin eksenleri belirlendikten sonra koordinat sisteminin merkezi olan (0,0,0) noktası yerdeki karolardan birinin köşesine konulmuştur. Yerdeki karoların ebatları, videolar gözlemlenerek insanların ayak uzunluklarının karo uzunluğuna oranı baz alınarak tespit edilmiştir. Yetişkin bir erkeğin ayak uzunluğunun 25cm olduğu varsayımına ve bir erkeğin ayağının karo uzunluğunun yarısına denk geldiği gözlemine dayanarak karo boyutları 50x50 olarak tespit edilmiştir. Bu boyutlar belirlendikten sonra 100 cm aralıklarla $z=0$ düzleminde noktalar işaretlenmiştir. DLT yönteminde tüm noktalar aynı düzlemde olamayacağından yukarıda bulunan iki nokta daha belirlenmiştir. Bu noktalar, Türkiye'deki bir kadının boyunun ortalama olarak 160cm'ye yakın olduğu varsayımına dayanarak işaretlenmiştir.



Şekil 3.7 Yeşil kapı kamerasında kalibrasyon için nokta ikililerinin bulunması

Nokta ikilileri hazırlandıktan sonra DLT ile iki kameranın da projeksiyon matrisleri elde edilmiştir. Bir sonraki aşama dünyadaki bir noktayı görüntüdeki bir noktaya gönderen bu matrisleri kullanarak bunun tam tersini yapan, yani görüntüdeki piksel koordinatlarından dünya koordinatlarına erişen bir dönüşüm tasarlamak olmuştur. x görüntü koordinatı, P projeksiyon matrisi ve X dünya koordinatları ise denklem 3.1 bunlar arasındaki ilişkiyi verir.

$$x = PX \quad (3.1)$$

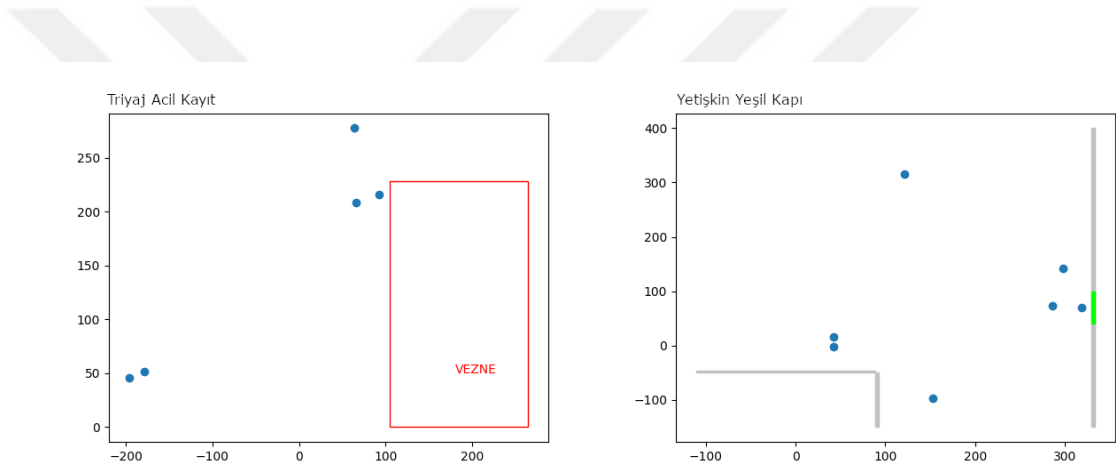
Eğer X dünya koordinatında $z=0$ ise bu denklem denklem 3.2'deki gibi yeniden yazılabilir.

$$x = \hat{P}\hat{X} \quad (3.2)$$

Bu denklemde $\hat{P}$, P 'nin üçüncü sütununun kaldırılmasıyla elde edilen 3×3 matristir. X 'in z koordinatının kaldırılmasıyla da $\hat{X}$ elde edilir. Bu durumda dünya koordinatları görüntü koordinatları cinsinden denklem 3.3'teki gibi bulunabilir.

$$\hat{X} = \hat{P}^{-1}x \quad (3.3)$$

Kişilerin YOLO ile tespit edilen görüntü koordinatlarını denklem 3.3 ile dünya koordinatlarına çevirebilmek için bu kişilerin dünya koordinat sistemindeki z değerleri hakkında varsayımda bulunmak gerekir. Bu çalışmada kişilerin sınırlayıcı kutularının üst-orta noktalarının zeminden 170cm yükseklikte olduğu varsayılmıştır. Bu varsayımla birlikte $z=0$ olacak şekilde gerekli dönüşümler yapıldıktan sonra kişilerin dünya koordinat düzlemindeki yerleri yaklaşık olarak bulunabilir. Şekil 3.8’de kişilerin dünya koordinatlarının bulunup işaretlenmesi ile oluşturulan triyaj acil kayıt ve yeşil kapı bölgelerine ait serpilme grafiklerinin örnekleri görülebilir.

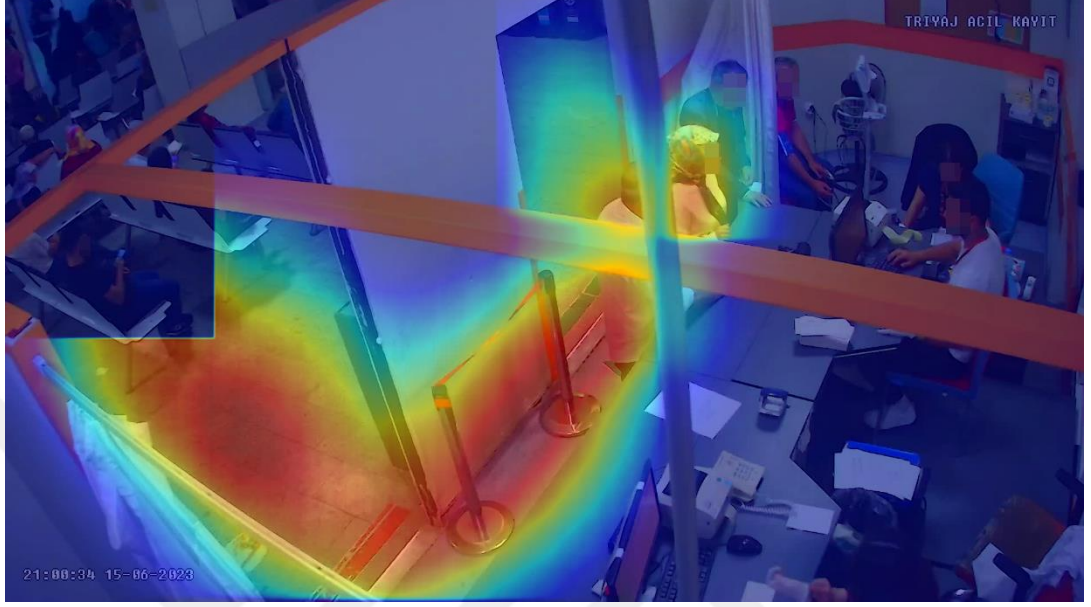


Şekil 3.8 Triyaj ve yeşil kapı bölgelerine ait serpilme grafiği örnekleri

Bu çalışmada yoğunluk analizi kapsamında yapılan bir başka uygulama da hem görüntü üzerinde hem de kuş bakışı görünümde sıcaklık haritalarının elde edilmesidir. İnsanların tespit edildiği bölgelerin işaretlenmesi ve takip edilmesiyle oluşturulan sıcaklık haritaları, belirli bir zaman aralığında gözlemlenen bölgede hangi bölgelerin daha yoğun olduğunu görselleştirir.

Görüntü üzerinde sıcaklık haritalarını elde etmek için, görüntü ile aynı boyutta iki boyutlu bir histogram oluşturulmuştur. İşlenen videoların her karesi için, o karede tespit edilen insanların sınırlayıcı kutularının içine karşılık gelen sayılar artırılmıştır. Tüm videolar işlendikten sonra histogram içindeki değerler 0 ile 1 arasına gelecek şekilde normalleştirilmiş. Normalleştirilen değerler Gauss filtresi ile yumuşatıldıktan sonra

OpenCV kütüphanesi ile sıcaklık haritasına dönüştürülmüş, sonra ilgili kameradan alınan bir görüntü üzerine yarı saydam olarak çizilmiştir.



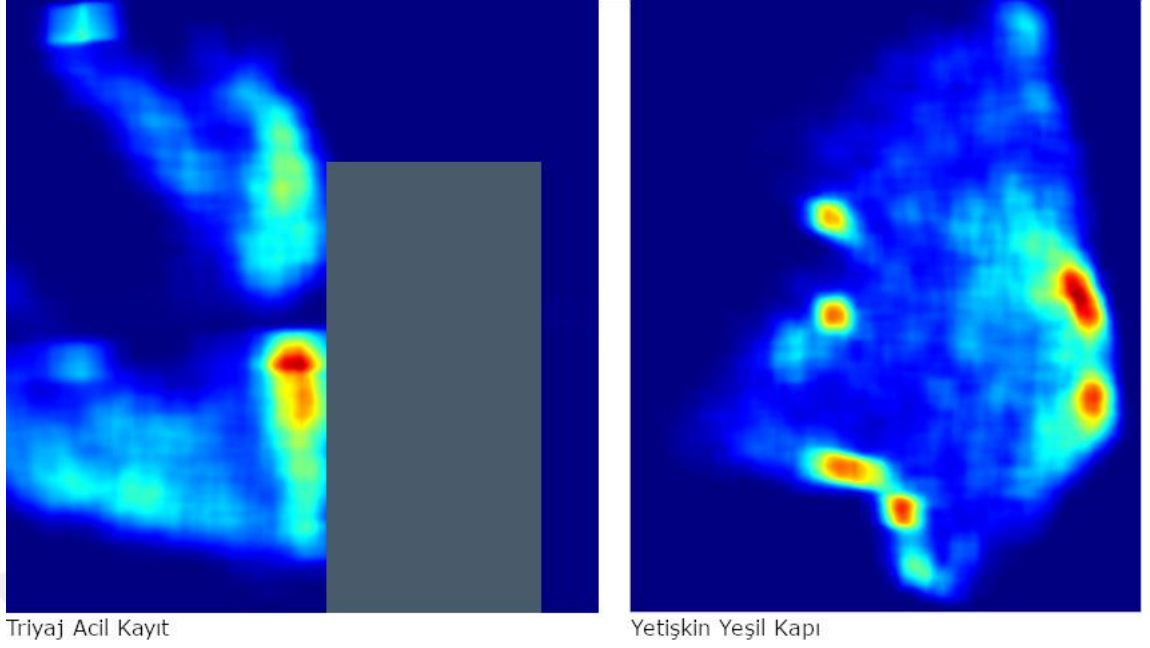
Şekil 3.9 Triyaj bölgesi görüntü sıcaklık haritası

Veri setindeki tüm videoların işlenmesiyle oluşturulan görüntü sıcaklık haritaları şekil 3.9 ve şekil 3.10'da verilmiştir. YOLO ile tespit yapılan bölgeler, vezne arkası, oturaklar gibi istenmeyen bölgelerde gerçekleşen tespitlerin maskelenmesi yöntemiyle sınırlandırılmıştır. Ancak yine de istenen bölge ile arzu edilmeyen bölgeler arasındaki sınırlarda tespitler gerçekleşebilir. Bu tespitlerin sıcaklık haritalarını etkilememesi için histogramda istenmeyen bölgelere ait sayıların değeri sıfırlanarak bu bölgeler devre dışı bırakılmıştır.



Şekil 3.10 Yeşil kapı bölgesi görüntü sıcaklık haritası

Aynı yöntemle videoların her karesi için insanların dünya koordinatları bulunduğundan sonra kuş bakışı görünüm sıcaklık haritaları elde edilmiştir. İnsanların dünya koordinatları, oluşturulan histogram üzerinde 30x30 büyüklüğünde bir kare olarak işaretlenmiştir. Veri setindeki tüm videoların işlenmesiyle triyaj acil kayıt ve yetişkin yeşil kapı bölgeleri için elde edilen kuş bakışı sıcaklık haritaları Şekil 3.11’de verilmiştir. Şekilde triyaj acil sıcaklık kayıt haritasında görülen gri dikdörtgen, masanın yerini belirtmek amacıyla sonradan manuel olarak eklenmiştir.



Trijaj Acil Kayıt

Yetişkin Yeşil Kapı

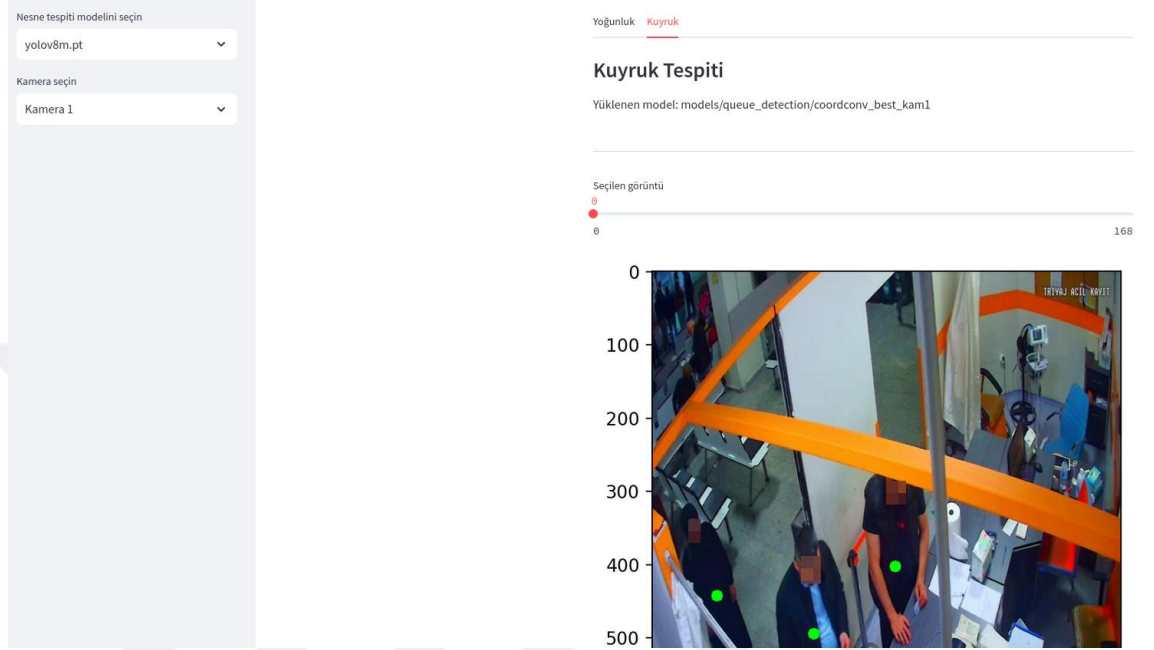
Şekil 3.11 Triyaj kayıt ve yeşil kapı önü bölgelerine ait kuş bakışı sıcaklık haritaları

3.4 İnteraktif Web Uygulaması

Çalışma kapsamında kuyruk tespiti ve bazı yoğunluk analizi operasyonlarının test edilerek gözlemlenebilmesi için bir web uygulaması geliştirilmiştir. Uygulama, Python dilinde Streamlit (2024) kütüphanesi kullanılarak yazılmıştır. Açık kaynak kodlu bir kütüphane olan Streamlit, sadece Python dilini kullanarak interaktif web uygulamaları geliştirmeye olanak sağlar. Streamlit kütüphanesinin kullanıcılardan veri girişi alabilmek için buton, checkbox, slider ve radyo butonu gibi çok sayıda giriş aracı sunar. Veri, medya ve yerleşim düzeni elemanları ile kullanıcılara zengin bir tecrübe sunmak için gerekli elemanları sağlar. Apache 2.0 yazılım lisansı ile sunulan Streamlit, veri görselleştirme veya interaktif makine öğrenmesi projelerinin hızlı prototiplerini geliştirmek için idealdir.

Şekil 3.12’de çalışma kapsamında Streamlit ile geliştirilmiş interaktif web uygulamasının arayüzü görülmektedir. Sol tarafta nesne tespitlerini gerçekleştirmek için kullanılacak YOLO modelleri seçilebilir. Seçenekler arasında görülmesi istenen modeller indirilerek uygulama dizininin modeller klasörüne yerleştirilmelidir. Burada büyük bir modeli seçmenin avantajı nesne tespitlerinin daha başarılı olmasıdır, ancak küçük modeller daha hızlı çalışır.

Sol taraftaki menüden uygulamada kullanılacak kamera (triyaj acil kayıt veya yeşil kapı önü) de seçilir. Uygulama ilk açıldığında veya seçilen model veya kamera değiştiğinde tüm veri seti üzerinde nesne tespiti gerçekleştirilir.



Şekil 3.12 Streamlit interaktif web uygulaması arayüzü

Uygulamanın “yoğunluk” ve “kuyruk” olmak üzere iki kısmı vardır. Yoğunluk kısmında test setinden seçilen görüntü için serpilme haritaları görüntülenebilir. Ayrıca görüntüde tespit edilen kişi sayısı da burada belirtilir. Kuyruk kısmında ise seçilen görüntü için kuyruk tahminleri yapılarak gösterilir.

4. TARTIŞMA VE SONUÇ

Bu tez çalışması kapsamında Ankara'da bulunan bir özel hastanenin güvenlik kameralarından elde edilen görüntüler üzerinde kuyruk tespiti ve yoğunluk haritası çalışmaları yapılmıştır. Görüntülerdeki insanları tespit etmek için YOLOv8 modeli kullanılmış, tespit edilen insanların konumları kuyruk tespiti yapmak ve yoğunluk haritalarını oluşturmak için kullanılmıştır.

Kuyruk tespiti ile görüntülerde belirli bir düzen ve organizasyon içinde bekleyen insanları tespit etmek amaçlanmıştır. Bu maksatla etiketli kamera görüntüleri üzerinde bir evrişimsel sinir ağı modeli eğitilmiştir. Çapraz doğrulama sonucunda model ortalama 0,91 F1 skoru ve 0,90 doğruluğa ulaşarak başarılı bir performans sergilemiştir. Elde edilen ortalama duyarlılık 0,94, ortalama kesinlik ise 0,87 olmuştur. Duyarlılığın yüksek olması, modelin artan FP sayısı ve düşen kesinlik bedeliyle de olsa pozitif örnekleri büyük bir başarıyla bulduğunu göstermektedir. Model sınırlayıcı kutuların orta nokta koordinatlarını kullanmaktadır. Bu durum sadece koordinat tabanlı dönüşümlere dayanan bir evrişimsel sinir ağı ile kuyruk tespitinin başarıyla yapılabileceğini göstermektedir.

Tez çalışmasının diğer odak noktası sıcaklık haritaları aracılığıyla gerçekleştirilen yoğunluk analizi olmuştur. Belirlenen zaman zarfında hangi bölgelerin daha yoğun olduğunu gösteren sıcaklık haritaları, hem görüntü üzerinde hem de sahnelerin kuş bakışı görünümüleri üzerinde gösterilmiştir. Kuş bakışı görünümüleri elde edebilmek üzere kişilerin sahnedeki dünya koordinatlarını bulmak için kameralar DLT yöntemi ile kalibre edilmiştir. Görüntülerde kamera kalibrasyonu için standart olarak kullanılan dama tahtası veya benzeri kalibrasyon nesneleri bulunmadığı için, sahnedeki nesnelerin ve insanların tahmini boyutları üzerinden fikir yürütülerek kalibrasyon için gereken nokta ikilileri belirlenmiştir. Bu nokta ikililerini belirlemek için bir görüntü editörü yazılımı ve bu yazılımın içindeki çizim ve ölçüm araçlarından yararlanılmıştır. Kamera kalibrasyonunun bu çalışma için yeterli düzeydeki başarısı, sahneye ve kalibrasyon araçlarına erişim olmadığı ve tahminlerin yaklaşık olarak doğru olmasının yeterli olduğu durumlarda kamera kalibrasyonunun bu şekilde gerçekleştirilmesinin makul bir yaklaşım olduğunu göstermektedir.

Kamera kalibrasyonu sonucunda elde edilen projeksiyon matrisleri kullanılarak, temel matris operasyonları ile kişilerin sahnedeki 2D koordinatları tahmin edilmiştir. Bu işlemleri gerçekleştirebilmek için kişilerin sınırlayıcı kutularının üst-orta noktaların yer düzleminde 170cm yukarıda olduğu varsayımı yapılmıştır. Bu işlemin ardından kişilerin sınırlayıcı kutularının görüntü üzerindeki konumları ve kuş bakışı görünüm üzerindeki dünya konumları 2 boyutlu histogramlar aracılığıyla sıcaklık haritalarına dönüştürülmüştür.

Bu çalışmada gerçekleştirilen yoğunluk analizi ve kuyruk tespiti modüllerinin girişlerini YOLOv8 nesne tespiti modeli ile elde edilen sınırlayıcı kutular oluşturmaktadır. Dolayısıyla nesne tespiti modelinin kişileri tespit edememesi veya hatalı tespit etmesi bu modüller için sorun teşkil etmektedir. Bu durumu iyileştirmek için gelecekteki çalışmalarda daha yeni ve başarılı nesne tespiti modelleri kullanılıp, bu modellerin ağırlıkları sahneler için fine-tune edilebilir.

KAYNAKLAR

- Chen, K. Loy, C. Gong, S. ve Xiang, T. 2012. Feature Mining for Localised Crowd Counting. British Machine Vision Conference (BMVC2012), 3-7 Eylül, Proceedings British Machine Vision Conference, 21.1-21.11, Surrey.
- Choi, W. Shahid, K. ve Savarese, S. 2009. What are they doing? : Collective activity classification using spatio-temporal relationship among people. IEEE 12th International Conference on Computer Vision (ICCV), 29 Eylül-2 Ekim, 1282-1289, Kyoto, Japonya.
- CVAT. 2024. Web sitesi: <https://www.cvat.ai/> Erişim Tarihi: 2 Haziran 2024.
- Dalal, N. ve Triggs, B. 2005. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), 20-26 Haziran, Volume 1, 886-893, California, ABD.
- Feliciani, C. Shimura, K. ve Nishinari, K. 2022. Introduction to Crowd Management: Managing Crowds in the Digital Era: Theory and Practice. Springer Nature, 303, İsviçre.
- Gall, J. Yao, A. Nima, R. Gool, L. ve Lempitsky, V. 2011. Hough Forests for Object Detection, Tracking, and Action Recognition. IEEE transactions on pattern analysis and machine intelligence, 33(11); 2188-2202.
- Gao, C. Liu, J. Feng, Q. ve Lv, J. 2016. People-flow counting in complex environments by combining depth and color information. Multimed Tools Appl, 75; 9315–9331.
- Girshick, R. 2015. Fast R-CNN. Proceedings of the IEEE international conference on computer vision (ICCV), 13-16 Aralık, 1440-1448, Santiago, Şili.
- Girshick, R. Donahue, J. Darrell, T. ve Malik J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 23-28 Haziran, 580-587, Columbus, ABD.
- Glorot, X. Bordes, A. ve Bengio, Y. 2011. Deep Sparse Rectifier Neural Networks. Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 315-323.
- Goodfellow, I. Bengio, Y. ve Courville, A. 2016. Deep Learning. The MIT Press, 781.
- Gómez A, H. Tomás, R. Tapia, S. Caballero, A. Ratté, S. Eras, A. ve González, P. 2015. Identification of Loitering Human Behaviour in Video Surveillance Environments. Lecture Notes in Computer Science, 516–525.
- Gul, M. Yousaf, M. Nawaz, S. Rehman, Z. ve Kim, H. 2020. Patient Monitoring by Abnormal Human Activity Recognition Based on CNN Architecture. Electronics, 9(12); 1993.

- Hao, Y. Du, H. Mao, M. Liu, Y. ve Fan, J. 2023. A Survey on Regression-Based Crowd Counting Techniques. *Information Technology and Control*, 52(3); 693-712.
- Hu, Y. Jiang, X. Liu, X. Zhang, B. Han, J. Cao, X. ve Doermann, D. 2020. NAS-Count: Counting-by-Density with Neural Architecture Search. *European Conference on Computer Vision (ECCV2020)*, 23-28 Ağustos, 747-766, Glasgow, Birleşik Krallık.
- IALAB. 2024. Web sitesi: <https://www.4smartmachines.com/ialab> Erişim Tarihi: 2 Haziran 2024.
- JAX. 2024. Web sitesi: <https://jax.readthedocs.io> Erişim Tarihi: 2 Haziran 2024.
- Keras. 2024. Web sitesi: <https://keras.io/> Erişim Tarihi: 2 Haziran 2024.
- Kuppusamy, P. Bharathi, V. 2022. Human abnormal behavior detection using CNNs in crowded and uncrowded surveillance – A survey. *Measurement: Sensors*, 24.
- Lempitsky, V. ve Zisserman, A. 2010. Learning To count objects in images, 23rd International Conference on Neural Information Processing Systems (NIPS10), 6-9 Aralık, Proceedings Volume 1, 1324-1332, Vancouver, Canada.
- Liang, D. Xie, J. Zou, Z. Ye, X. Xu, W. ve Bai, X. 2023. CrowdCLIP: Unsupervised Crowd Counting via Vision-Language Model. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 17-24 Haziran, 2893-2903, Vancouver, Canada.
- Li, B. Huang, H. Zhang, A. Liu, P. ve Liu, C. 2021. Approaches on Crowd Counting and Density Estimation: A Review. *Pattern Analysis and Applications*, 24(3); 853-874.
- Liu, C. Lu, H., Cao, Z. and Liu, T. 2023. Point-query quadtree for crowd counting, localization, and more. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 1-6 Ekim, 1676-1685, Paris, Fransa.
- Liu, R. Lehman, J. Molino, P. Such, F. Frank, E. Sergeev, A. ve Yosinski, J. 2018. An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution. *NeurIPS 2018*.
- Luger, G. 2009. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving* Sixth Edition. Pearson Education, 779.
- Ma, H. Zhang, L. ve Wei, X. 2024. FGNet: Fine-Grained Extraction Network for Congested Crowd Counting. *Multimedia Modeling*, 43-46.
- Merriam-Webster. 2014. *Merriam-Webster's Advanced Learner's English Dictionary*. Encyclopaedia Britannica Inc, 2032, India.
- Murphy, K. 2012. *Machine Learning A Probabilistic Perspective*. The MIT Press, 1104, İngiltere.

- Nayak, R. Behera, M. Girish, V. Pati, U. ve Das, S. 2019. Deep Learning Based Loitering Detection System Using Multi-Camera Video Surveillance Network. IEEE International Symposium on Smart Electronic Systems (iSES), 16-18 Aralık, 215-220, Rourkela, India.
- Nguyen, V. Le, M. Do, A. Duong, H. Thai, T. ve Tran, D. 2014. An efficient camera-based surveillance for fall detection of elderly people. 9th IEEE Conference on Industrial Electronics and Applications, 9-11 Haziran, 994-997, Hangzhou, China.
- Phung, V. ve Rhee, E. 2019. A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets. Applied Sciences, 9(21); 4500.
- PyTorch. 2024. Web sitesi: <https://pytorch.org/> Erişim Tarihi: 5 Haziran 2024.
- Ranasinghe, Y. Nair, N. G. Bandara, W. G. C. ve Patel, V. M. 2024. Crowddiff: Multi-hypothesis crowd density estimation using diffusion models. arXiv. <https://doi.org/10.48550/arXiv.2303.12790>
- Redmon, J. Divvala, S. Girshick, R. ve Farhadi, A. 2016. You only look once: Unified, real-time object detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27-30 Haziran, 779-788, Las Vegas, ABD.
- Ren, S. He, K. Girshick, R. ve Sun, J. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. Advances in Neural Information Processing Systems, 28; 91-99.
- Reza, F. 2020. Machine Learning and Other Artificial Intelligence Applications. In: Neuroimaging Clinics of North America. Mukherji, S. ve Forghani, R. (eds), Elsevier Health Sciences, 393-517, ABD.
- Russel, S. ve Norvig, P. 2022. Artificial Intelligence A Modern Approach Fourth Edition Global Edition. Pearson, 1166, Slovakiya.
- Shi, Z. Sun, Y. ve Zhang, M. 2024. Training-free Object Counting with Prompts. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 322-330, Waikoloa, ABD.
- Sindagi, V. ve Patel, V. 2018. A survey of recent advances in CNN-based single image crowd counting and density estimation. Pattern Recognition Letters, Video Surveillance-oriented Biometrics, 107; 3-16.
- Srivastava, N. Hinton, G. Krizhevsky, A. Sutskever, I. ve Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research (JMLR), 15(56); 1929-1958.
- Streamlit. 2024. Web sitesi: <https://streamlit.io/> Erişim Tarihi: 23 Mayıs 2024.
- Szeliski, Richard. 2022. Computer Vision: Algorithms and Applications 2nd ed. Springer International Publishing, 947.

- Tensorflow. 2024. Web sitesi: <https://www.tensorflow.org/> Erişim Tarihi: 5 Haziran 2024.
- Topdemir, H. 1998. Işığın yayılımının niteliği konusunda üç önemli adım: İbnü'l-Heysem, Kemalüddin El-Farisi, Takiyüddin B. Maruf. Ankara Üniversitesi Dil ve Tarih-Coğrafya Fakültesi Dergisi, 38(1-2); 381-403.
- Tran, K. Gala, A. Kakadiaris, I. Shah, S. 2014. Activity analysis in crowded environments using social cues for group discovery and human interaction modeling. Pattern Recognition Letters, 44; 49-57.
- Turing, A. 1950. Computing Machinery and Intelligence. Mind, LIX(236); 433–460.
- TÜİK. 2024. Web Sitesi: <https://data.tuik.gov.tr/Bulten/Index?p=Adrese-Dayali-Nufus-Kayit-Sistemi-Sonuclari-2023-49684>. Erişim Tarihi: 02.06.2024.
- Türkiye İstatistik Kurumu. 2008. Adrese Dayalı Nüfus Kayıt Sistemi Sonuçları. Türkiye İstatistik Kurumu Matbaası, 34, Ankara.
- Voulodimos, A. Doulamis, N. Doulamis, A. ve Protopapadakis E. 2018. Deep Learning for Computer Vision: A Brief Review. Computational Intelligence and Neuroscience.
- Wang, Q. ve Breckon, T. 2022. Crowd Counting via Segmentation Guided Attention Networks and Curriculum Loss. IEEE Transactions on Intelligent Transportation Systems, 23(9); 15233-15243.
- Wu, B. ve Nevatia, R. 2007. Detection and Tracking of Multiple, Partially Occluded Humans by Bayesian Combination of Edgelet based Part Detectors. Int J Comput Vis, 75; 247-266.
- Wu, J. Wang, L. Wang, L. Guo, J. ve Wu, G. 2019. Learning Actor Relation Graphs for Group Activity Recognition. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 16-20 Haziran, 9956-9966, Long Beach, ABD.
- Wu, LF. Wang, Q. Jian, M. Qiao, Y. Zhao, B. 2021. A Comprehensive Review of Group Activity Recognition in Videos. International Journal of Automation and Computing, 18; 334–350.
- Xu, B. ve Qiu, G. 2016. Crowd density estimation based on rich features and random projection forest. IEEE Winter Conference on Applications of Computer Vision (WACV), 7-10 Mart, 1-8, New York, ABD.
- Xu, G. and Yin, J. 2023. Mlp-air: An efficient mlp-based method for actor interaction relation learning in group activity recognition. arXiv. <https://doi.org/10.48550/arXiv.2304.08803>
- Zhang, L. Shi, M. ve Chen, Q. 2018. Crowd Counting via Scale-Adaptive Convolutional Neural Network. IEEE Winter Conference on Applications of Computer Vision (WACV), 12 Mart-15 Mart, 1113-1121, Lake Tahoe, ABD.

- Zhang, Y. Liu, W. Xu, D. Zhou, Z. Wang, Z. 2024. Bi-Causal: Group Activity Recognition via Bidirectional Causality. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 17-21 Haziran, 1450-1459, Seattle, ABD.
- Zhang, Y. Zhou, D. Chen, S. Gao, S. ve Ma, Y. 2016. Single-Image Crowd Counting via Multi-Column Convolutional Neural Network. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 26 Haziran-1 Temmuz, 589-597, Las Vegas, ABD.

