Sosyal Mesafe Algılayıcı

Onur Kantar, Ömer Resul Ertan

Bilişim Sistemleri Mühendisliği / Teknoloji Fakültesi

Kocaeli Üniversitesi

[onurkantr@gmail.com](mailto:onurkantr@gmail.com), [omerresulertan1034@gmail.com](mailto:omerresulertan1034@gmail.com)

Özet

*Devam eden COVID-19 corona virüsü salgını, ölümcül yayılımıyla küresel bir felakete neden oldu. Etkili tedavilerin yokluğu ve virüse karşı aşılama eksikliğinden dolayı, popülasyon savunmasızlığı artmaktadır. Mevcut durumda aşı olmadığından, sosyal mesafenin pandemik virüsün yayılmasına karşı yeterli bir önlem olduğu düşünülmektedir. İnsanlar arasında fiziksel temastan kaçınarak virüs yayılma riskleri en aza indirilebilir. Bu nedenle, bu çalışmanın amacı, tepegöz perspektifi kullanarak sosyal mesafe izleme için derin öğrenme platformu sağlamaktır. Görüntüdeki insanları tanımlamak için YOLOv3 nesne tanıma yöntemi kullanılır. Algılama modeli, algılanan sınırlayıcı kutu bilgilerini kullanarak insanları tanımlar. Öklid mesafesi kullanılarak, tespit edilen sınırlayıcı kutu ağırlık merkezinin insanların ikili mesafeleri belirlenir. İnsanlar arasındaki sosyal mesafe ihlallerini tahmin etmek için piksele olan yaklaşık fiziksel mesafeyi kullandık ve bir eşik belirledik. Mesafe değerinin asgari sosyal mesafe eşiğini aşıp aşmadığını değerlendirmek için bir ihlal eşiği belirlenir. Ek olarak, sosyal mesafe eşiğini ihlal eden / geçen kişi de izlenecek şekilde video dizilerindeki bireyleri tespit etmek için bir izleme algoritması kullanılır. Ayrıca video girişi olarak 5 farklı yöntem sunulmaktadır. Bulgular, geliştirilen çerçevenin çok yakın yürüyen ve sosyal mesafeleri ihlal eden bireyleri başarılı bir şekilde ayırt ettiğini göstermektedir*

*Anahtar kelimeler: derin öğrenme, sosyal mesafe, COVID-19, OpenCV, havadan görünüm, kişi algılama, YOLOv3*

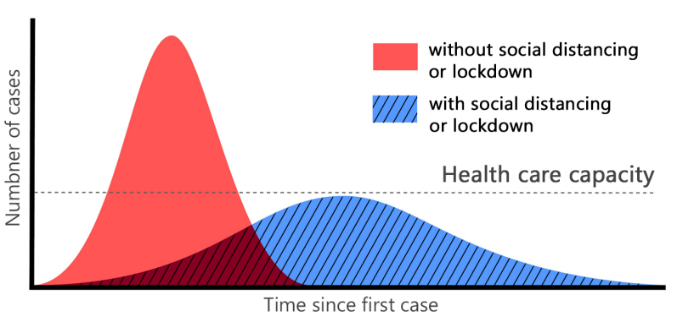
Abstract

*The ongoing COVID-19 corona virus outbreak caused a global disaster with its deadly spread. Due to the lack of effective treatments and lack of vaccination against the virus, population vulnerability is increasing. Social distancing is considered to be an adequate measure against the spread of the pandemic virus, as there is currently no vaccine. Virus spread risks can be minimized by avoiding physical contact between people. Therefore, the purpose of this study is to provide a deep learning platform for social distance monitoring using an overhead perspective. YOLOv3 object recognition method is used to identify people in the image. The perception model identifies people using perceived bounding box information. By using the Euclidean distance, the binary distances of the people of the detected bounding box center of gravity are determined. We used the approximate physical distance to the pixel and set a threshold to predict violations of social distance between people. An infringement threshold is set to assess whether the distance value exceeds the minimum social distance threshold. In addition, a tracking algorithm is used to detect individuals in video sequences so that the person who violates / crosses the social distance threshold is also tracked. In addition, 5 different methods are offered as video input. Findings show that the developed framework successfully distinguishes individuals who walk too close and who violate social distances.*

*Keywords: deep learning, social distance, COVID-19, OpenCV, aerial view, person perception, YOLOv3*

# Giriş

COVID-19, başlangıçta Çin'in Wuhan kentinde Aralık 2020'nin sonlarında bildirilen koronavirüsün neden olduğu hastalıklar ailesine aittir. 11 Mart'ta 118.000 aktif vaka ve 4000 ölümle 114 ülkeye yayıldı, DSÖ bunu bir pandemi olarak ilan etti [1], [2]. 4 Mayıs 2020'de dünya çapında 3.519.901'den fazla vaka ve 247.630 ölüm bildirildi. Bazı sağlık kuruluşları, tıp uzmanları ve bilim adamları bu ölümcül virüs için uygun ilaçlar ve aşılar geliştirmeye çalışıyor, ancak bugüne kadar hiçbir başarı bildirilmedi. Bu durum, küresel topluluğu bu bulaşıcı virüsün yayılmasını durdurmak için alternatif yollar aramaya zorluyor. Sosyal mesafenin mevcut senaryoda en iyi yayılma durdurucu olduğu iddia ediliyor ve etkilenen tüm ülkeler sosyal mesafeyi uygulamak için çalışıyor. Bu araştırma, koronavirüs pandemisini minimum ekonomik çaba kaybıyla desteklemeyi ve azaltmayı amaçlamaktadır.

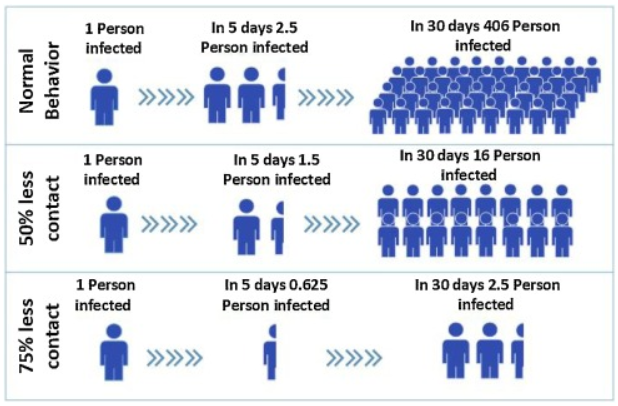


Şekil 1: Sosyal mesafenin mevcut sağlık hizmeti kapasitesiyle eşleşmesinin gösterimi.

Sosyal mesafe kavramı, COVID-19'un bulaşmasını en aza indirmeyi veya kesmeyi amaçlayan çeşitli yollar ve çabalar doğrultusunda ortaya çıkan en iyi uygulamadır. Enfekte kişiler ile sağlıklı kişiler arasındaki fiziksel teması azaltmayı amaçlamaktadır. DSÖ normlarına [3] göre, insanların sosyal mesafeyi takip edebilmek için aralarında en az 2 metre mesafeyi korumaları öngörülmüştür.

Yakın zamanda yapılan bir çalışma, sosyal mesafenin önemli bir sınırlama önlemi olduğunu ve SARSCoV-2'yi önlemek için gerekli olduğunu göstermektedir, çünkü hafif semptomları olan veya hiç semptomu olmayan kişiler tesadüfen korona enfeksiyonu taşıyabilir ve başkalarına bulaştırabilir [4]. Şekil 1, uygun sosyal mesafenin bulaşıcı fiziksel teması azaltmanın en iyi yolu olduğunu ve dolayısıyla enfeksiyon oranını düşürdüğünü göstermektedir [5], [6]. Bu azalan zirve, kesinlikle mevcut sağlık hizmetleri altyapısı ile eşleşebilir ve koronavirüs pandemisine karşı savaşan hastalara daha iyi olanaklar sunmaya yardımcı olabilir.

Sosyal mesafe, özellikle COVID-19'dan ciddi hastalık riski yüksek olan kişiler için çok önemlidir. Enfekte bir kişiden sağlıklı bir kişiye virüs bulaşma riskini azaltarak, virüsün yayılması ve hastalık şiddeti önemli ölçüde azaltılabilir Şekil 2.



Şekil 2: Sosyal mesafenin öneminin gösterimi.

Buradan esinlenilen bu çalışmada, popüler nesne algılama ve izleme yöntemleri kullanılarak sosyal mesafe takip sistemi oluşturulmuştur. Rapor yapısının geri kalanı aşağıdaki şekilde düzenlenmiştir: Bölüm 2, problemin tanımından ve neye çözüm getirildiğinden bahsedilmiştir. Ardından Bölüm 3’de bu projeye benzer çalışmalar özet olarak anlatılmıştır. Bölüm 4’te ise diğer projelerden farkının neler olduğu açıklanmıştır.

# Problem

Dünya genelinde virüsün yayılması ve her geçen gün artması insanların can sağlığını riske atmakta. Bunun çözümü için devletler adımlar atsa da bir sonuca ulaşmakta pek de başarılı olamadılar. Virüsün genel olarak korunma yöntemleri arasında maske takmak, hijyeni sağlamak ve sosyal mesafeyi korumak olsa da zaman zaman insanların bunlara dikkat etmedikleri görülmektedir. Bu yöntemlerin toplumlar tarafından sağlanmaması salgının riskinin artmasına neden olur.

Devletler insan sağlığını korumak için birçok çalışmada bulundu. Günlerce sokağa çıkma yasakları uygulandı. Toplu taşımalarda, kuyruklarda ve diğer kalabalık yerlerde insanların sosyal mesafeye uyması için uyarılar içeren yapıştırmalar yerleştirildi. Bazı kolluk kuvvetleri ile toplumlar uyarıldı ve uyarılara uymayanlara cezalar kesildi. Fakat bu ve buna benzer yöntemler toplumların düzelmesini sağlamamasıyla birlikte virüsü de engellemeyi sağlamamıştır. Bunun nedeni ise virüsün kontrol edilemeyecek kadar çok hızlı yayılması.

Son zamanlarda yayılan aşı haberleri virüsün sona ereceğini söylemekte. Fakat aşının tüm insanlığa yapılıp sağlıklı toplumlar oluşturulana kadar salgının devam edeceği bilinmekte. Aşı gelişmeleri tamamlanana kadar maske takmak, hijyeni sağlamak ve sosyal mesafeyi korumak şu anda yapılması gereken en önemli virüsten korunma yöntemleri. Bu yöntemler arasında toplumların uymakta en çok zorlandıkları ise sosyal mesafeyi korumak olduğu biliniyor. Bunun nedeni ise insanın tek başına yaşayamamasıdır. İnsan tek başına olduğunda ihtiyaçlarını karşılayamaz, hayat şartlarıyla baş edemez. Ayrıca insan mutlaka eşi, dostu, arkadaşları ile yaşamak, acı tatlı olayları paylaşmak ister. Başkalarına ihtiyaç duymadan yaşayan insanlar ise ruh ve beden sağlığını kaybettiği görülmektedir. Böyle bir yaşam tarzı olan insanın medeni toplumlarla da bunu görmezden gelmesi beklenemez. İnsan içgüdüsel olarak diğer insanlara yakın olmak ve iletişim kurmak ister. Durum böyle olunca da insanlığın bir süreliğine birbirinden uzak durmasını sağlamak için daha etkili kontrollerin yapılması gerekir. Bunu sağlayamayan toplumların salgın nedeniyle krizin eşiğinde olduğu görülmektedir.

# Benzer Çalışmalar

Bu kısımda literatürdü bu projeye benzer olan çalışmalar özet olarak anlatılmaktadır.

COVID-19 için sosyal mesafe. COVID-19, Aralık 2019'dan beri dünya çapında ciddi akut solunum sendromlarına neden olmuştur [7]. Yakın zamanda yapılan çalışmalar, sosyal mesafenin COVID-19'un yayılmasını yavaşlatmak için etkili bir önlem olduğunu göstermiştir [8]. Sosyal mesafe, olası temastan kaçınmak için her bireyden en az 2 metre (6 fit) uzakta tutulması olarak tanımlanır. Daha ileri analizler ayrıca sosyal mesafenin önemli ekonomik faydaları olduğunu göstermektedir [9]. COVID-19 kısa vadede tamamen ortadan kaldırılamayabilir, ancak sosyal mesafe önlemlerinin izlenmesine ve analiz edilmesine yardımcı olabilecek otomatik bir sistem toplumumuza büyük ölçüde fayda sağlayabilir.

Gelişen teknolojiler, sosyal mesafenin uygulanmasına yardımcı olabilir. Yakın zamanda yapılan bir çalışma [10], kablosuz iletişim, ağ oluşturma ve yapay zeka (AI) gibi yeni teknolojilerin sosyal mesafeyi nasıl etkinleştirebileceğini ve hatta güçlendirebileceğini belirledi. Çalışma, sosyal mesafe için olası temel kavramları, ölçümleri, modelleri ve pratik senaryoları tartıştı.

Ohio Eyalet Üniversitesi’nde 5 arkadaş tarafından COVID-19 için Vizyon Tabanlı Sosyal Mesafe ve Kritik Yoğunluk Algılama Sistemi geliştirildi [11]. Bu çalışmada bireyler arasındaki mesafeleri tespit edebilen ve onları uyarabilen aktif bir gözetim sistemi, ölümcül hastalığın yayılmasını yavaşlatabileceği düşünülmüştür. Bu çerçevede, sosyal mesafeyi ölçmek için monoküler bir kamera ve derin öğrenmeye dayalı gerçek zamanlı nesne dedektörleri kullanmayı öneriyorlar. Ayrıca sosyal mesafe tedbirini ihlal eden kişiyi hedef almadan müdahaleci olmayan görsel-işitsel uyarı sinyali verileceği söylenmiştir.

YOLO v3 ve Deepsort teknikleriyle kişi algılama ve izleme ile COVID-19 sosyal mesafeyi izleme çalışması gerçekleştirildi [12]. Mevcut aşı olmadığından, pandemiye karşı savaşmak için tek uygun yaklaşım sosyal mesafe olduğu düşünülmektedir. Gözetleme videosu kullanarak sosyal mesafeyi izleme görevini otomatikleştirmek için derin öğrenmeye dayalı bir çerçeve önermektedir. Önerilen çerçeve, insanları arka plandan ayırmak için YOLO v3 nesne algılama modelini, sınırlayıcı kutular ve atanmış kimlikler yardımıyla tanımlanan kişileri izlemek için Deepsort yaklaşımını kullanır.

Diğer bir çalışmada [13] ise kablosuz ve ağ iletişimi, yapay zeka (AI) gibi yeni teknolojilerin sosyal mesafe uygulamalarını nasıl etkinleştirebileceği, teşvik edebileceği ve hatta uygulayabileceği hakkında kapsamlı bir anket sağlamayı amaçlamaktadır. Bu amaçla, önce temel kavramlar, ölçümler, modeller dahil olmak üzere kapsamlı bir sosyal mesafe arka planı sağlıyor ve çeşitli pratik sosyal mesafe senaryoları öneriyorlar. Daha sonra, mesafeyi korumak, genel olarak sosyal mesafeyi teşvik etmek ve güçlendirmek için özellikle etkili olan ve pratikte yaygın olarak benimsenebilecek kablosuz teknolojilerin etkinleştirilmesini tartışıyorlar. Bundan sonra, makine öğrenimi, bilgisayarla görme, termal, ultrason vb. gibi ortaya çıkan ve ilgili diğer teknolojiler tanıtıyorlar. Bu teknolojiler, sosyal mesafedeki sorunların üstesinden gelmek için birçok yeni çözüm ve yön açacağını söylüyorlar.

Sosyal mesafenin korunmasını sağlayan diğer bir uygulamada [14] ise video çekimlerinden sosyal uzaklığı tespit etmek için SD-Measure adlı yeni bir çerçeve önermektedir. Önerilen çerçeve, bir video karesindeki insanları tespit etmek için Mask R-CNN derin sinir ağından yararlanıyor. İnsanlar arasındaki etkileşim sırasında sosyal mesafenin uygulanıp uygulanmadığını tutarlı bir şekilde belirlemek için centroid izleme algoritması kullanılır. İnsanların kameradan ve kendi aralarındaki mesafesini tahmin etmek için otantik algoritmalar yardımıyla, sosyal mesafe kurallarına uyulup uyulmadığını belirliyorlar. Çerçeve, Özel Video Görüntü Veri Kümesi (CVFD) ve Özel Kişisel Görüntü Veri Kümesi (CPID) üzerinde test ettiklerinde düşük bir yanlış alarm oranıyla yüksek bir doğruluk değeri elde etti ve burada sosyal mesafe kurallarının uygulanıp uygulanmadığını belirlemede etkinliğini gösterdiler.

# Diğer Çalışmalardan Farkı

Covid-19 döneminde sosyal mesafenin sağlanması için birçok çalışma gerçekleştirilse de birçoğunda hala eksik bulunan özellikler bulunmakta. Bu aşamada bizim yaptığımız bu çalışmada projelerin eksiklerini tamamlanması yönünde de başarı sağlamıştır. Genel olarak diğer projeler incelendiğinde sistemi yoran gereksiz işlemlerin gerçekleştirildiği görülmektedir. Bizim çalışmamızda oldukça basit bir yapı oluşturulmuştur. Bu sayede projenin performansında artış sağlanmıştır. Bu projenin diğer projelerden farklı olan diğer bir yönü ise görüntü girişi için birçok seçenek sunulmasıdır. Youtube, Youtube-Live, video dosyası, ekran görüntüsü ve kamera kullanılarak başarılı bir şekilde sosyal mesafe kontrolü gerçekleştirilebilir. Ayrıca işlenen görüntünün kaydedilebiliyor olması sağlanmıştır. Bu sayede tercihe göre proje çalıştırdıktan sonra işlenen görüntü .avi uzantılı bir dosya olarak kaydedilebilir.

# Metodolojiler

Bu kısımda projede kullanılan yöntemlerden kısaca bahsedilmiştir.

## OpenCV

OpenCV (Açık Kaynak Bilgisayar Görme Kitaplığı), esas olarak gerçek zamanlı bilgisayar görüşünü hedefleyen bir programlama işlevleri kütüphanesi [16]. Başlangıçta Intel tarafından geliştirilmiş, daha sonra Willow Garage ve daha sonra Itseez (daha sonra Intel [17] tarafından satın alınmıştır) tarafından desteklenmiştir. Kitaplık, çapraz platformdur ve açık kaynak Apache 2 Lisansı altında kullanım için ücretsizdir. 2011'den itibaren OpenCV, gerçek zamanlı işlemler için GPU hızlandırma özelliğine sahiptir [18].

OpenCV, yüzleri ve nesneleri algılama ve tanımlama, videolarda insani eylemleri sınıflandırma, kamera hareketlerini ve hareketli nesneleri izleme, nesneleri 3 boyutlu modellerine ayıklama, stereo kameralardan 3D nokta bulutları üretme, görüntüleri yüksek çözünürlükte birleştirme gibi alanlarda başarılı bir şekilde kullanılabilir.

OpenCV, modüler bir yapıya sahiptir, yani paket birkaç paylaşılan veya statik kitaplık içerir. Aşağıdaki modüller mevcuttur:

### [Core functionality](https://docs.opencv.org/master/d0/de1/group__core.html) (core)

Yoğun çok boyutlu Mat dizisi ve diğer tüm modüller tarafından kullanılan temel fonksiyonlar dahil olmak üzere temel veri yapılarını tanımlayan kompakt bir modül.

### Image Processing (imgproc)

Doğrusal ve doğrusal olmayan görüntü filtreleme, geometrik görüntü dönüşümleri (yeniden boyutlandırma, afin ve perspektif çarpıtma, genel tablo tabanlı yeniden eşleme), renk alanı dönüştürme, histogramlar vb. içeren bir görüntü işleme modülü.

### Video Analysis (video)

Hareket tahmini, arka plan çıkarma ve nesne izleme algoritmalarını içeren bir video analiz modülü.

### Camera Calibration and 3D Reconstruction (calib3d)

Temel çoklu-görünüm geometri algoritmaları, tek ve stereo kamera kalibrasyonu, nesne poz tahmini, stereo yazışma algoritmaları ve 3B rekonstrüksiyon unsurları.

### 2D Features Framework (features2d)

Göze çarpan özellik dedektörleri, tanımlayıcılar ve tanımlayıcı eşleştiriciler.

### Object Detection (objdetect)

Önceden tanımlanmış sınıfların nesnelerinin ve örneklerinin tespiti (örneğin, yüzler, gözler, kupalar, insanlar, arabalar vb.).

### High-level GUI (highgui)

Basit UI yetenekleri için kullanımı kolay bir arayüz.

### Video I/O (videoio)

Video yakalama ve video codec bileşenleri için kullanımı kolay bir arayüz

FLANN ve Google test sarmalayıcıları, Python bağlamaları ve diğerleri gibi diğer bazı yardımcı modüller.

## YOLO

YOLO, konvolüsyonel sinir ağlarını (CNN) kullanarak nesne tespiti yapan bir algoritmadır. Açılımı “You Only Look Once”, yani “Sadece Bir Kez Bak”. Bu adın seçilmesinin nedeni algoritmanın nesne tespitini tek seferde yapabilecek kadar hızlı olmasıdır. YOLO algoritması çalışmaya başladığında görüntülerdeki veya videolardaki nesneleri ve bu nesnelerin koordinatlarını aynı anda tespit eder.

Video ve resim işleme arasında tek fark resimlerin tek bir kareden (frame), videoların ise birçok kareden oluşmasıdır. Resimlerde algoritma tek bir kare için çalışırken, videolarda tüm kareler için tekrar tekrar çalışır. YOLO algoritması, öncelikle görüntüyü bölgelere ayırır. Daha sonra her bir bölgedeki nesneleri çevreleyen kutuları (bounding box) çizer ve her bir bölgede nesne bulunma olasılığı ile ilgili bir hesabı yapar. Ayrıca her bir bounding box için bir güven skoru hesaplar. Bu skor bize o nesnenin yüzde kaç olasılıkla tahmin edilen nesne olduğunu söyle. Örneğin, bulunan bir araba için güven skoru 0,3 ise bunun anlamı o nesnenin araba olma olasığının oldukça düşük olduğudur. Diğer bir deyişle, YOLO yaptığı tahminin güvenilmez olduğunu bize söyler.

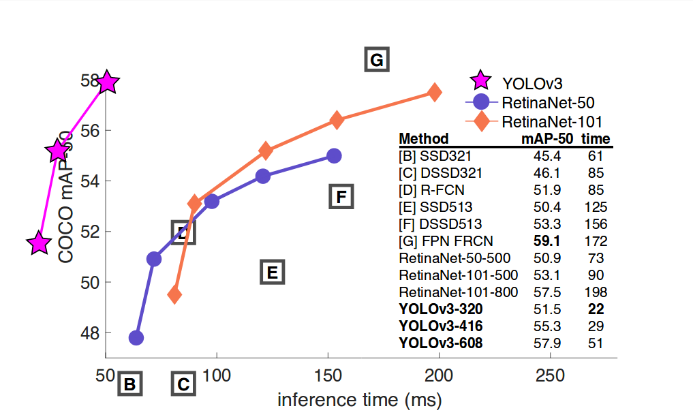
Bounding box’ların içindeki nesnelere non-maximum suppression denen bir teknik uygulanır. Bu teknik güven skoru düşük olan nesneleri değerlendirmeden çıkarır ve aynı bölgede güven skoru daha yüksek bir bounding box‘ın varlığını kontrol eder.



Şekil 3: Örnek YOLO projesi çıktısı.

Her bir bölgede nesne olup olmadığı araştırılır. Eğer bir nesne bulunursa o nesnenin orta noktası, yüksekliği ve genişliği bulunur daha sonra bounding box çizilir. Bunun yapılabilmesi için bir takım alt işlemlerin yapılması gerekir. Her bir bölge için bir tahmin vektörü oluşturulur, bu vektörlerin içinde güven skoru yer alır. Eğer güven skoru 0 ise orada nesne yok, 1 ise orada nesne var demektir. Aynı içerisindeki aynı nesne için birden fazla bounding box çizdirilebilir. İşte bu sorundan kurtulmak için de non-maximum suppression tekniği kullanılır. Bu teknik ile yapılan şey basitçe, en yüksek güven skoru olan bounding box’ın kalması diğerlerinin ise görüntüden atılmasıdır. Tüm işlemlerden sonra Şekil 3’teki çıktıya erişilir.

Şekil 4’te YOLOv3 ve diğer algoritmaların COCO veri setinde 0.5 IoU (mAP-50) ile karşılaştırması görülmektedir. Grafikten de anlaşılacağı üzere YOLO rakiplerine karşı süre ve doğruluk açısından çok iyi durumda. Ayrıca istenilirse doğruluk ve hız arasında rahatlıkla takas yapılabilir.



Şekil 4: YOLOv3’ün diğer alternatifleriyle kıyaslanmasının gösterimi.

Bu projede de kullanılan YOLO’nun bazı formülleri şu şekildedir:

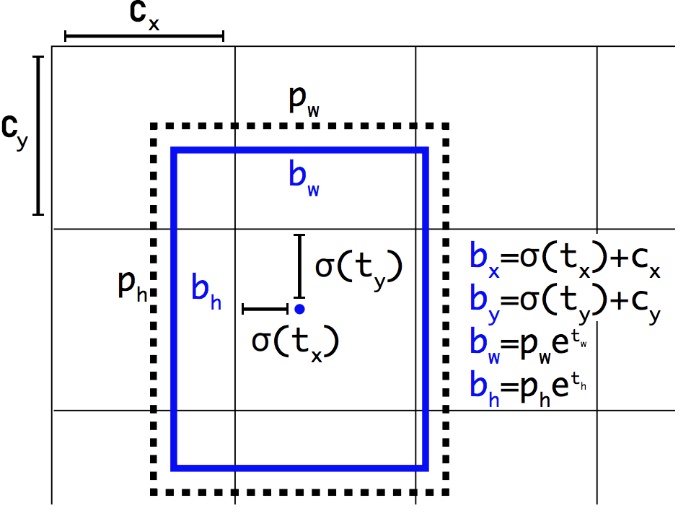
(1)

Denklemde (1), Pr(p), mevcut kişinin tespit edilen sınırlayıcı kutuda olup olmadığını gösterir. Pr(p) değeri evet için 1 ve hayır için 0'dır. IoU(pred, actual), sınırlayıcı kutuların tahmin hassasiyetini değerlendirmek için bir parametredir. IoU, tahmin edilen sınırlar ile gerçek sınırlar arasındaki örtüşme oranını değerlendirerek algılama performansını doğrular.

(2)

BoxT ile temsil edilen eğitim veri setinde manuel olarak etiketlenmiş kesin referans kutusu (gerçek) ve tahmin edilen sınırlayıcı kutu BoxP olarak görüntülenir. area, kesişme alanını gösterir. Giriş çerçevesinde tespit edilen her kişi için kabul edilebilir bir alan tahmin edilir ve kararlaştırılır. Güvenilirlik değeri, optimum sınırlayıcı kutuyu elde etmek için tahminden sonra uygulanır. Tahmin edilen her sınırlayıcı kutu için h, w, x, y tahmin edilir, burada sınırlayıcı kutu koordinatları x, y ile tanımlanır ve genişlik, yükseklik; w, h ile belirlenir. Model, Şekil 5 ve Denklem 3'te görüldüğü gibi sınırlayıcı kutu değerlerini üretir.

(3)



Şekil 5: Sınırlayıcı kutunun algılanan koordinatları.

Yüksek güven değerlerini işleyen ve düşük güven değerlerini atan bir eşik değeri tanımlanır. Maksimal olmayan bastırma kullanılarak, tespit edilen sınırlayıcı kutu için nihai konum parametreleri türetilir. Son olarak, tespit edilen sınırlayıcı kutu için kayıp fonksiyonu hesaplanır. Verilen kayıp işlevi, üç işlevin toplamıdır, yani regresyon, sınıflandırma ve güven. Her ızgara hücresinde, nesne tespit edilirse, sınıflandırma kaybı, koşullu sınıf olasılıklarının karesi olarak hesaplanır.

(4)

Denklemde (4), grid hücresinde eğer kişi tespit edilirse =1, aksi takdirde 0'a eşittir. grid hücresindeki sınıfı için koşullu sınıf olasılıkları olarak temsil edilir. Yerelleştirme kaybı, tahmin edilen sınırlayıcı kutu boyutları ve konumlarındaki hataları tahmin eder. Algılanan nesneyi içeren sınırlayıcı kutu, yani bir kişi eklenir.

(5)

Izgara hücresi 'deki 'inci sınırlayıcı kutu nesne algılama için kullanılırsa 1’e eşittir, aksi takdirde 0’a eşittir. Model, basit yükseklik ve genişliği tahmin etmek yerine, sınırlayıcı kutu genişliğinin ve yüksekliğinin karekökünü tahmin eder. Denklemde (5) ölçek parametreleri , sınırlayıcı kutu koordinatlarının tahminleri için kullanılır ve 5'e eşittir. Tahmin edilen pozisyonlar, tespit edilen sınırlayıcı kutunun hücresinde ile temsil edilirken, 'inci hücredeki sınırlayıcı kutunun gerçek pozisyonları kullanılarak tanımlanır. Eşitlikte (5) , koordinat değerine sahip tahmin edilen sınırlayıcı kutunun kayıp fonksiyonunu ölçer. 'inci sınırlayıcı kutu 'de tespit edilen kişinin olasılığını temsil etmek için kullanılır. Denklemdeki sabittir (5). Her ızgara hücresi için tahmin edici olarak 'yi kullanarak her sınırlayıcı kutu üzerindeki toplamı hesaplar.

Son olarak Denklemde verilen güven kaybı hesaplanır (6).

(6)

Güven puanı olarak tanımlanır, ızgara hücresi ve 'deki ’inci sınırlayıcı kutu için ve hücresinde ’inci sınırlayıcı kutu nesne tespitinden sorumluysa 1'e eşittir; aksi takdirde 0'a eşittir. Nesne tespit edilmezse güven kaybı sağlanır.

(7)

Denklemde (7) , 'ün tamamlayıcısı olarak tanımlanır. ve hücresindeki sınırlayıcı kutunun güven puanı, arka plan tespit edilirken kaybı ağırlıklandırmak için kullanılır. Algılanan çoğu durumda olduğu gibi, sınırlayıcı kutular sınıf dengesizliği sorununa neden olan nesneler içermez; bu nedenle model, nesneleri tespit etmek yerine arka planı tespit etmek için daha sık eğitilir. Bunu çözmek için, kayıp, ağırlık olarak faktör kadar azaltılır (varsayılan: 0,5).

## COCO

COCO, büyük ölçekli bir nesne algılama, bölümleme ve altyazı oluşturma veri kümesidir. COCO'nun birkaç özelliği vardır:

* Nesne segmentasyonu
* Bağlam içinde tanıma
* Superpiksel malzeme segmentasyonu
* 330.000 görüntü (200.000’den fazla etiketli)
* 1,5 milyon nesne örneği
* 80 nesne kategorisi
* 91 malzeme kategorisi
* Resim başına 5 başlık
* Kilit noktalara sahip 250.000 kişi

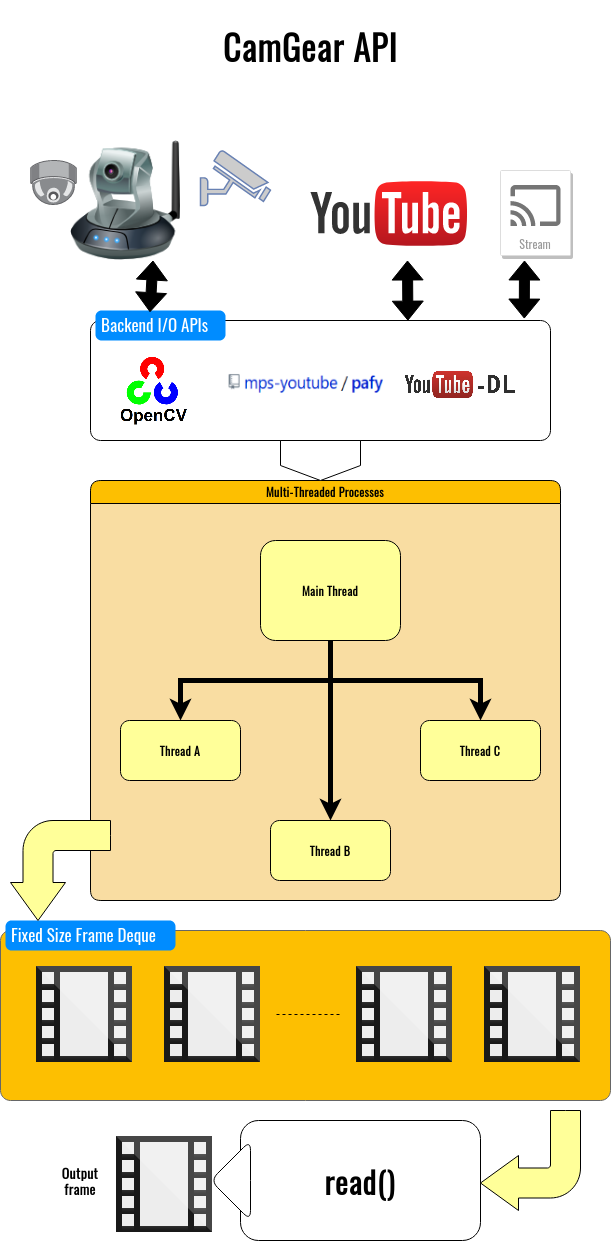
## VidGear

VidGear, python da gerçek zamanlı uygulamalar oluşturmak için hepsi bir arada eksiksiz bir Video İşleme çözümü sağlayan Yüksek Performanslı bir Çerçevedir. VidGear gerçek zamanlı olarak çeşitli cihazlardan video karelerini okuyabilir, yazabilir, işleyebilir, gönderebilir ve alabilir. VidGear, programcıların ve yazılım geliştiricilerin karmaşık Video İşleme görevlerini mevcut veya yeni uygulamalarında yalnızca birkaç satır kodla kolayca entegre etmelerine ve gerçekleştirmelerine olanak tanır.

VidGear, her biri bazı benzersiz işlevlere sahip birden fazla API’ler oluşturmuştur. Bu API'lerin her biri, özel olarak cihaza özgü farklı video akışlarını, ağ akışlarını ve medya kodlayıcılarını işlemek / kontrol etmek için tasarlanmıştır. Bu API'ler, sağlam hata işleme ve benzersiz performans sağlarken dahili parametrelerinden ve yöntemlerinden esnek bir şekilde yararlanmak için bünyesinde bulundurduğu son teknoloji kütüphanelerin üzerinde kullanımı kolay, oldukça genişletilebilir, çok iş parçacıklı ve asyncIO katmanı sağlar.

### CamGear

CamGear, neredeyse tüm IP/USB kameraları, multimedya video dosyası formatını (4k'ye kadar test edilmiş), çeşitli ağ akış protokollerini ve ayrıca YouTube desteğini içeren çok çeşitli cihazlardan / akışlardan ultra hızlı kareler yakalayabilir. CamGear API’sinin yapısı Şekil 6’da gösterilmiştir.



Şekil 6: CamGear yapısının gösterimi.

### VideoGear

VideoGear, kodu fazla değiştirmeden farklı video kaynaklarına geçmek gerektiğinde idealdir. Ayrıca, çeşitli video akışları için (gerçek zamanlı veya değil) minimum çabayla ve çok daha az kod satırı kullanarak kolay stabilizasyon sağlar.

### PiGear

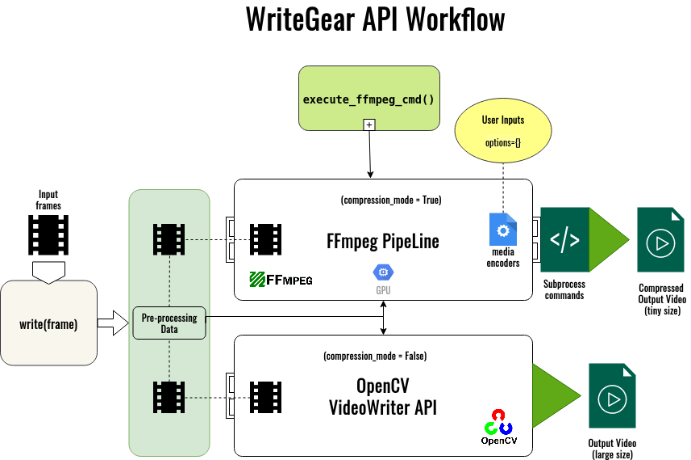
PiGear, CamGear'a benzer, ancak çeşitli Raspberry Pi Kamera Modüllerini (OmniVision OV5647 Kamera Modülü ve Sony IMX219 Kamera Modülü gibi) desteklemek için yapılmıştır.

### ScreenGear

ScreenGear, ultra hızlı Ekran Kaydı için özel olarak tasarlanmıştır. Bu, monitördeki kareleri bilgisayar ekranında bir alan tanımlayarak veya tam ekran olarak gerçek zamanlı yakalayabilme anlamına gelir.

### WriteGear

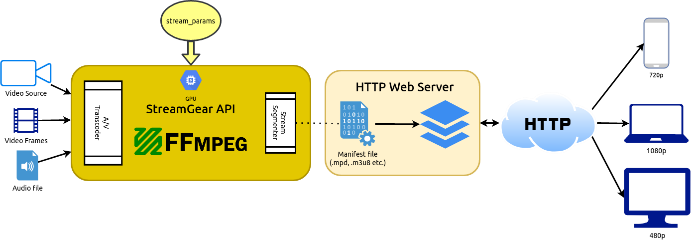
WriteGear, multimedya dosyalarıyla hayal edebileceğiniz neredeyse her şeyi yapma özgürlüğü sağlayan çeşitli güçlü Yazar Araçlarını kullanır. WriteGear’ın yapısı Şekil 7’de gösterilmektedir



Şekil 7: WriteGear yapısının gösterimi.

### StreamGear

StreamGear, yalnızca birkaç satır python kodunda Ultra Düşük Gecikme, Yüksek Kalite, Dinamik ve Uyarlanabilir Akış Biçimleri (MPEG-DASH) oluşturmak için kod dönüştürme iş akışını otomatikleştirir. StreamGear’ın yapısı Şekil 8’de gösterilmiştir



Şekil 8: StreamGear yapısının gösterimi.

### NetGear

NetGear, özel olarak video karelerini ağ üzerinden gerçek zamanlı olarak birbirine bağlanan sistemler arasında eşzamanlı ve eşzamansız olarak aktarmak için tasarlanmıştır.

### WebGear

WebGear, canlı video karelerini bir ağ üzerindeki herhangi bir web tarayıcısına aktaran güçlü bir Video Akış Sunucusu olarak hareket edebilir. Buna ek olarak, WebGear API hem CamGear hem de PiGear API'lerine dahili erişim sağlar. Böylece stabilizasyonun etkin olduğu gerçek video akışı gibi herhangi bir cihazdan / kaynaktan gelen akış kareleri için özel güç sağlayan VideoGear etrafında dahili bir ortam sağlar.

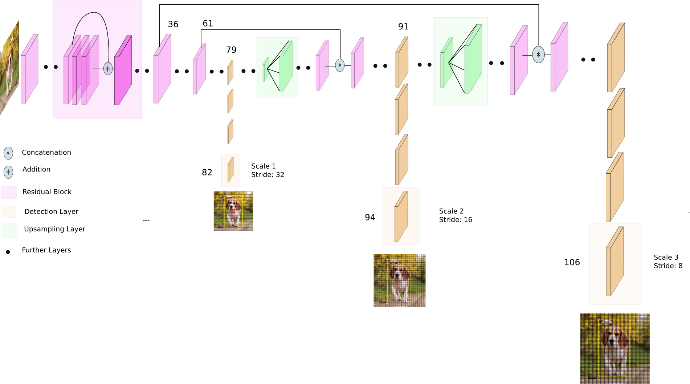
### NetGear\_Async

NetGear\_Async, yaklaşık 1/3 bellek tüketiminde NetGear API'ye kıyasla iki kat performans oluşturabilir. Temel olarak, bu API, sistemde önemli bir yüke neden olmadan binlerce çerçeveyi birkaç saniye içinde aktarabilir.

# Önerilen Model

Bu kısımda projenin önerilen modeli anlatılmıştır. Araştırmacılar, sosyal mesafeyi uzaktan izleme için önden veya yandan bir perspektif kullanırlar. Bu çalışmada, tepegöz perspektifi kullanan derin öğrenmeye dayalı bir sosyal mesafe izleme çerçevesi tanıtılmıştır. Genel nesne tespiti için en iyi performansı nedeniyle, bu çalışmada YOLOv3 kullanılmıştır. Model, sınırlayıcı kutuları ve sınıf olasılıklarını tahmin etmek için tek aşamalı ağ mimarisini kullandı. Model başlangıçta COCO veri seti üzerinde eğitilmiştir ve sonucunda algılama işlemi gerçekleştirilmiştir. Ayrıca video girişi olarak 5 farklı yöntem sunulmuştur ve işlenen video .avi uzantılı olarak kaydedilebilir.

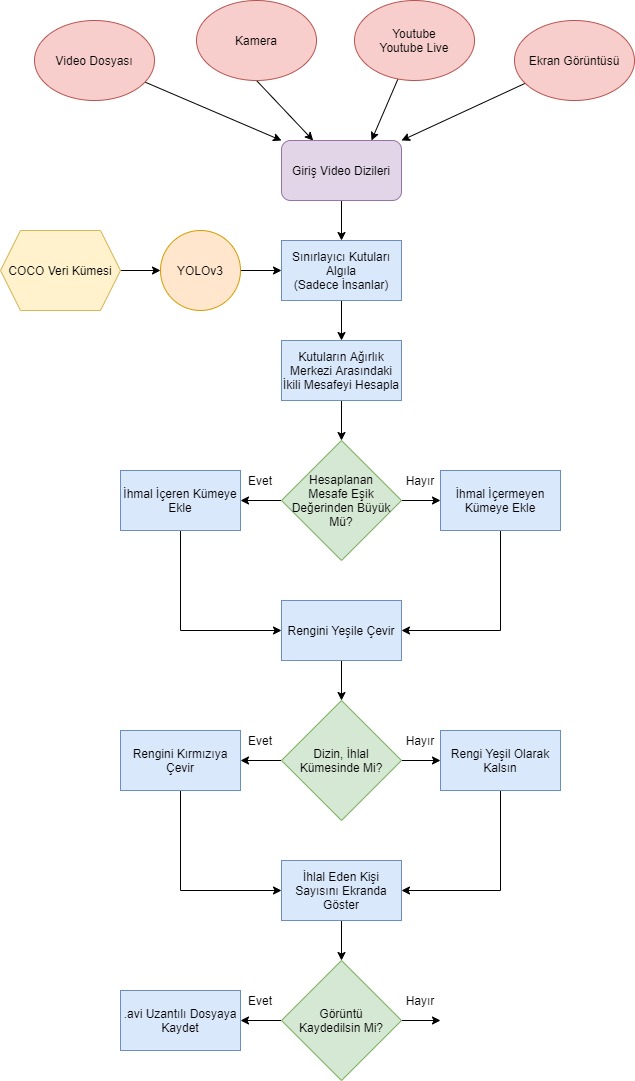
Algılamanın ardından, sınırlayıcı kutu bilgisi, esas olarak ağırlık merkezi bilgisi, her bir sınırlayıcı kutu ağırlık merkez mesafesini hesaplamak için kullanılır. Öklid mesafesini kullandık ve tespit edilen her sınırlayıcı insan kutusu arasındaki mesafeyi hesapladık. Merkez mesafesinin hesaplanmasının ardından, herhangi iki sınırlayıcı kutu ağırlık merkezi arasındaki mesafenin yapılandırılan piksel sayısından daha az olup olmadığını kontrol etmek için önceden tanımlanmış bir eşik kullanılır. İki kişi birbirine yakınsa ve mesafe değeri asgari sosyal mesafe eşiğini ihlal ediyorsa. Sınırlayıcı kutu bilgisi, bir ihlal kümesinde saklanır ve sınırlayıcı kutunun rengi güncellenir / kırmızıya dönüştürülür. İzleme için bir centroid izleme algoritması benimsenmiştir, böylece sosyal mesafe eşiğini ihlal eden / aşan kişilerin izlenmesine yardımcı olur. Çıktıda, model, tespit edilen insan sınırlayıcı kutuları ve ağırlık merkezleriyle birlikte toplam sosyal mesafe ihlali sayısı hakkındaki bilgileri görüntüler.



Şekil 9: YOLOv3 ağ mimarisi.

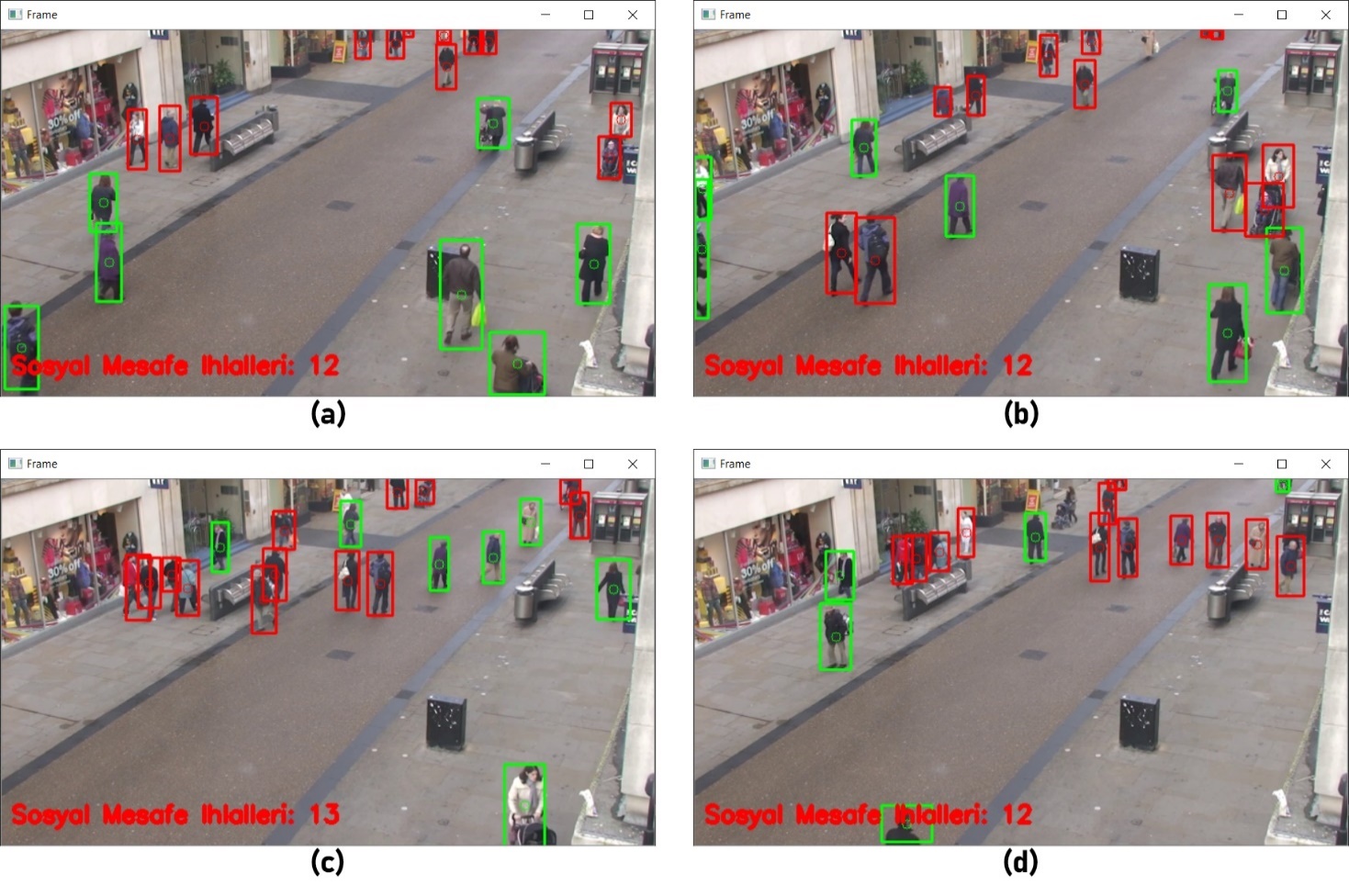
Bu çalışmada YOLOv3, özellikle küçük ölçekli nesnelerde öngörü doğruluğunu artırdığından insan algılaması için kullanılmıştır. Ana avantajı, çok ölçekli nesne tespiti için ayarlanmış ağ yapısına sahip olmasıdır. Modelin genel mimarisi Şekil 9'da sunulmuştur; Bloklar olarak da adlandırılan evrişimli katmanlar kullanılarak özellik öğrenmenin gerçekleştirildiği görülebilir. Bloklar birçok evrişimli katmandan oluşur ve bağlantıları atlar. Modelin benzersiz özelliği, Şekil 9'da gösterildiği gibi, algılama işlemini üç ayrı ölçekte gerçekleştirmesidir. Belirli bir adımda evrişimli katmanlar, özellik haritasını alt örneklemek ve değişmez boyutlu özellikleri aktarmak için uygulanır [15]. Şekil 9'da gösterildiği gibi üç özellik haritası, nesne tespiti için kullanılır.

Şekil 9'da gösterilen mimari, tespit edilen nesnelerin sınırlayıcı kutusunu ve sınıf olasılığını tahmin etmek için tüm girdi görüntüsü tek aşamalı bir ağ kullanır. Özellik çıkarımı için mimari, evrişim katmanlarını kullanır ve sınıf tahmini için tamamen bağlantılı katmanlar kullanılır. Hayvan tanımlama sırasında, Şekil 9'da görüldüğü gibi, giriş çerçevesi, grid hücreleri olarak da adlandırılan bir SxS bölgesine bölünür. Bu hücreler, sınırlayıcı kutu tahmini ve sınıf olasılıkları ile ilgilidir.



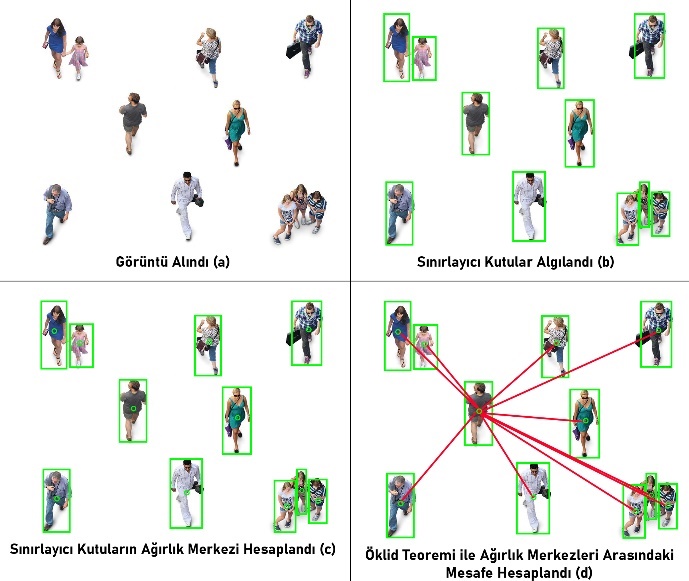
Şekil 10: Önerilen modelin akış şeması.

Video karelerinde insanları tespit ettikten sonra, bir sonraki adımda, Şekil 12 (b)'de gösterildiği gibi, yeşil kutular olarak gösterilen her bir tespit edilen kişi sınırlayıcı kutularının ağırlık merkezi mesafe hesaplaması için kullanılır. Tespit edilen sınırlayıcı kutu koordinatları (x, y), sınırlayıcı kutunun ağırlık merkezini hesaplamak için kullanılır. Şekil 12 (c), bir dizi sınırlayıcı kutu koordinatını kabul etmeyi ve ağırlık merkezini hesaplamayı gösterir. Hesaplamadan sonra, centroid, tespit edilen her sınırlayıcı kutuya benzersiz bir kimlik atanır. Bir sonraki adımda, Öklid mesafesini kullanarak tespit edilen her centroid arasındaki mesafeyi ölçer. Video akışındaki sonraki her kare için, ilk olarak Şekil 12 (c)'de gösterilen sınırlayıcı kutu ağırlık merkezlerini hesaplar ve sonra tespit edilen sınırlayıcı kutu ağırlık merkezlerinin her çifti arasındaki mesafeyi (kırmızı çizgilerle vurgulanan) hesaplar, Şekil 12 (d). Her centroidin bilgisi bir liste şeklinde saklanır. Mesafe değerlerine bağlı olarak, herhangi iki kişinin N pikselden daha az uzaklıkta olup olmadığını kontrol etmek için bir eşik tanımlanır.



Şekil 11: Deney sonuçlarının gösterimi.

Mesafe, minimum sosyal mesafe setini ihlal ediyorsa veya iki kişi çok yakınsa, bilgi ihlal setine eklenir. Sınırlayıcı kutu rengi yeşil olarak başlatılır. Bilgiler ihlal setinde kontrol edilir; mevcut dizin ihlal seti varsa, renk kırmızı olarak güncellenir. Ayrıca, video sekansında tespit edilen kişileri izlemek için centroid izleme algoritması kullanılır. İzleme algoritması ayrıca sosyal mesafe eşiğini ihlal eden kişilerin izlenmesine de yardımcı olur. Çıktıda, model toplam sosyal mesafe ihlali sayısı hakkında bilgi görüntüler.



Şekil 12: Önerilen modelin gösterimi.

Şekil 10’da da görüldüğü üzere modelin akış şeması, uygulanan adımları açıklamaktadır.

# Deney Sonuçları

Bu çalışmada gerçekleştirilen çeşitli deneylerin ayrıntılı açıklamaları bu bölümde sunulmaktadır. Sosyal mesafe izleme için, YouTube üzerinde yayınlanan şehir kameraları kullanılır ve mümkün oldukça tepeden görünümden yakalanan video dizilerini içerir. Sahne boyunca kişilerin hareketliliğinde herhangi bir kısıtlama yoktur. Sahnedeki insanlar özgürce hareket eder; görsel görünümleri radyal uzaklık ve kamera pozisyonundan etkilenir. Örnek çerçevelerden, insanın görsel görünümünün aynı olmadığı ve veri setinde insanların boylarının, pozlarının, ölçeklerinin farklılaştığı gözlemlenebilir. Uygulama için OpenCV kullandık. Deneysel sonuçlar önceden eğitilmiş modelin test sonuçları ile gerçekleştirilir.

Şekil 11'de önceden eğitilmiş bir model kullanılarak sosyal mesafe çerçevesinin test sonuçları görselleştirilmiştir. Test sonuçları, farklı video dizileri kullanılarak değerlendirilir. Video görüntülerindeki insanlar sahnelerde özgürce hareket ediyor; Örnek çerçevelerden kişinin görsel görünümünün önden veya yandan görünümle aynı olmadığı görülebilir (Şekil 11). Kişinin boyutu da Şekil 11'de gösterildiği gibi farklı yerlerde değişiklik göstermektedir. Model sadece insan (kişi) sınıfını dikkate aldığından; bu nedenle, önceden eğitilmiş bir model tarafından yalnızca insan gibi görünüme sahip bir nesne tespit edilir. Önceden eğitilmiş model, Şekil 11'de yeşil dikdörtgenlerle gösterildiği gibi iyi sonuçlar verir ve çeşitli boyutlardaki kişi sınırlayıcı kutuları algılar. Şekil 11'deki örnek çerçevelerden, insanlar sosyal mesafe eşiğini korurken yeşil dikdörtgenlerle işaretlenmiştir. Model aynı zamanda, gösterildiği gibi birden çok insan için test edilmiştir. Örnek görüntülerde, kişi tespitinden sonra, olay yerinde bulunan kişinin sosyal mesafeyi ihlal edip etmediğini kontrol etmek için tespit edilen her sınırlayıcı kutu arasındaki mesafenin ölçüldüğü görülmektedir. Şekil 11 (c)'de, sahnedeki 13 kişi, sosyal mesafe eşiğini ihlal ettikleri veya aştıkları için kırmızı sınırlayıcı kutularla işaretlenmiştir. Örnek çerçevelerden, bir kişinin birkaç sahne konumunda etkili bir şekilde algılandığı görülebilir. Ancak bazı durumlarda kişinin görünümü değişiyor; bu nedenle model yanlış tespitler verir. Yanlış tespitin nedeni, önceden eğitilmiş model uygulandığında ve bir bireyin üstten görünümden görünüşü değiştikçe model için yanıltıcı olabilir.

# Sonuç

Bu çalışmada derin öğrenmeye dayalı, bir sosyal uzaktan izleme çerçevesi genel bir bakış açısı kullanılarak sunulmuştur. Önceden eğitilmiş YOLOv3 modeli, insan tespiti için kullanılır. Bu çalışma, genel bakış açısı sosyal mesafe izleme için kullanılan, birden fazla giriş sağlanan ilk çalışmadır. Algılama modeli, ağırlık merkezi koordinat bilgilerini içeren sınırlayıcı kutu bilgisi verir. Öklid mesafesini kullanarak, tespit edilen sınırlayıcı kutular arasındaki ikili ağırlık merkezi mesafeleri ölçülür. İnsanlar arasındaki sosyal mesafe ihlallerini kontrol etmek için piksele olan yaklaşık bir fiziksel mesafe kullanılır. Mesafe değerinin ayarlanan minimum sosyal mesafeyi ihlal edip etmediğini kontrol etmek için bir ihlal eşiği kullanılır. Ayrıca, olay yerindeki insanları izlemek için bir centroid izleme algoritması kullanılır. Deneysel sonuçlar, çerçevenin çok yakın yürüyen insanları etkili bir şekilde tanımladığını ve sosyal mesafeyi ihlal ettiklerini gösterdi. Sosyal mesafe eşiğini ihlal eden kişi veya kişilerin izlenmesine yardımcı olmak için farklı algılama ve izleme algoritmaları kullanılabilir.

# Kaynaklar

[1] W. H. Organization, “WHO corona-viruses (COVID-19)”, https://www.who.int/emergencies/diseases/novel-corona-virus-2019, 2020.

[2] WHO, “Who director-generals opening remarks at the media briefing on covid-19-11 march 2020”, https://www.who.int/dg/speeches/detail/, 2020.

[3] L. Hensley, “Social distancing is out, physical distancing is inheres how to do it”, Global News, Canada, 2020.

[4] ECDPC, “Considerations relating to social distancing measures in response to COVID-19 second update”, https://www.ecdc.europa.eu/en/publications-data/considerations, 2020.

[5] M. W. Fong, H. Gao, J. Y. Wong, J. Xiao, E. Y. Shiu, S. Ryu, and B. J. Cowling, “Nonpharmaceutical measures for pandemic influenza in nonhealthcare settingssocial distancing measures”, 2020.

[6] F. Ahmed, N. Zviedrite, and A. Uzicanin, “Effectiveness of workplace social distancing measures in reducing influenza transmission: a systematic review”, BMC public health, vol. 18, no. 1, p. 518, 2018.

[7] D. S. Hui, E. I. Azhar, T. A. Madani, F. Ntoumi, R. Kock, O. Dar, G. Ippolito, T. D. Mchugh, Z. A. Memish, C. Drosten et al., “The continuing 2019-ncov epidemic threat of novel coronaviruses to global healththe latest 2019 novel coronavirus outbreak in wuhan, china”, International Journal of Infectious Diseases, vol. 91, pp. 264–266, 2020.

[8] C. Courtemanche, J. Garuccio, A. Le, J. Pinkston, and A. Yelowitz, “Strong social distancing measures in the united states reduced the covid-19 growth rate: Study evaluates the impact of social distancing measures on the growth rate of confirmed covid-19 cases across the united states”, Health Affairs, pp. 10–1377, 2020.

[9] M. Greenstone and V. Nigam, “Does social distancing matter?”, University of Chicago, Becker Friedman Institute for Economics Working Paper, no. 2020-26, 2020.

[10] C. T. Nguyen, Y. M. Saputra, N. Van Huynh, N.-T. Nguyen, T. V. Khoa, B. M. Tuan, D. N. Nguyen, D. T. Hoang, T. X. Vu, E. Dutkiewicz et al., “Enabling and emerging technologies for social distancing: A comprehensive survey”, arXiv, arXiv:2005.02816, 2020.

[11] Y. [Dongfang](https://arxiv.org/search/eess?searchtype=author&query=Yang%2C+D), Y. [Ekim](https://arxiv.org/search/eess?searchtype=author&query=Yurtsever%2C+E), R. [Vishnu](https://arxiv.org/search/eess?searchtype=author&query=Renganathan%2C+V), R. [Keith A.](https://arxiv.org/search/eess?searchtype=author&query=Redmill%2C+K+A), Ö. [Ümit,](https://arxiv.org/search/eess?searchtype=author&query=%C3%96zg%C3%BCner%2C+%C3%9C) “A Vision-based Social Distancing and Critical Density Detection System for COVID-19”, arXiv, arXiv:2007.03578, 2020.

[12] P. Narinder S., S. Sanjay K., A. Sonali, “Monitoring COVID-19 social distancing with person detection and tracking via fine-tuned YOLO v3 and Deepsort techniques”, arXiv, arXiv:2005.01385, 2020

[13] N. Cong T., S. Yuris M., H. Nguyen V., N. Ngoc-Tan, K. Tran V., T. Bui M., N. Diep N., H. Dinh T., V. Thang X., D. Eryk, C. Symeon, O. Bjorn, arXiv, arXiv:2005.02816,2020

[14] S. Gupta, R. Kapil, G. Kanahasabai, S. S. Joshi and A. S. Joshi, "SD-Measure: A Social Distancing Detector", 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), India, doi: 10.1109/CICN49253.2020.9242628, pp. 306-311, 2020.

[15] A. Imran, A. Misbah, R. Joel J.P.C., J. Gwanggil, D. Sadia, "A deep learning-based social distance monitoring framework for COVID-19", Sustainable Cities and Society, 2210-6707, 102571, 2020.

[16] P. Kari Antero, B. Anatoly, K. Kirill, E. Victor, “Realtime Computer Vision with OpenCV: Mobile Computer-Vision Technology Will Soon Become as Ubiquitous as Touch Interfaces”, Association for Computing Machinery, 10, 40-56, 2012.

[17] OpenCV Library, “Intel acquires Itseez”, <https://opencv.org/intel-acquires-itseez.html>, 2020

[18]"CUDA", opencv.org, 2020.