

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BENZERLİĞE DAYALI DERİN SİYAM AĞLARI İLE  
ÇOK NESNELİ TAKİPTE KİŞİLERİ YENİDEN  
TANIMLAMA**

**YÜKSEK LİSANS TEZİ**

**Harun SULJAGIC**

**Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ  
MÜHENDİSLİĞİ**  
**Tez Danışmanı : Doç. Dr. Numan ÇELEBİ**

**Haziran 2022**

T.C.  
SAKARYA ÜNİVERSİTESİ  
FEN BİLİMLERİ ENSTİTÜSÜ

**BENZERLİĞE DAYALI DERİN SİYAM AĞLARI İLE  
ÇOK NESNELİ TAKİPTE KİŞİLERİ YENİDEN  
TANIMLAMA**

**YÜKSEK LİSANS TEZİ**

**Harun SULJAGIC**

**Enstitü Anabilim Dalı : BİLİŞİM SİSTEMLERİ  
MÜHENDİSLİĞİ**

**Bu tez 01.06.2022 tarihinde aşağıdaki jüri tarafından oybirliği / oyçokluğu ile kabul edilmiştir.**

## **BEYAN**

Tez içindeki tüm verilerin akademik kurallar çerçevesinde tarafımdan elde edildiğini, görsel ve yazılı tüm bilgi ve sonuçların akademik ve etik kurallara uygun şekilde sunulduğunu, kullanılan verilerde herhangi bir tahrifat yapılmadığını, başkalarının eserlerinden yararlanılması durumunda bilimsel normlara uygun olarak atıfta bulunulduğunu, tezde yer alan verilerin bu üniversite veya başka bir üniversitede herhangi bir tez çalışmasında kullanılmadığını beyan ederim.

Harun SULJAGIC

01.06.2022

## TEŐEKKÜR

Yüksek lisans eğitimim boyunca değerli bilgi ve deneyimlerinden yararlandığım, her konuda bilgi ve desteğini almaktan çekinmediğim, araştırmanın planlanmasından yazılmasına kadar tüm aşamalarında yardımlarını esirgemeyen, teşvik eden, aynı titizlikte beni yönlendiren değerli danışman hocam Doç. Dr. Numan Çelebi'ye teşekkürlerimi sunarım.

Teknik danışmanlığı konusunda anlayış ve yardımlarını esirgemeyen, bilgi ve deneyimlerinden yararlandığım sayın hocam Dr. Öğr. Üyesi Ertuğrul Bayraktar'a teşekkür ederim.

Ayrıca yüksek lisans eğitimim boyunca maddi açıdan beni destekleyen Yurtdışı Türkler ve Akraba Topluluklar Başkanlığı'na, Türkiye Burslarına ve Kromel Makina Yönetim Kurulu Başkanı Sn. İsmet Çalışkan'a teşekkür ederim.

# İÇİNDEKİLER

TEŞEKKÜR.....	i
İÇİNDEKİLER .....	ii
ŞEKİLLER LİSTESİ .....	v
TABLOLAR LİSTESİ .....	vii
ÖZET.....	viii
SUMMARY .....	ix
BÖLÜM 1.	
GİRİŞ .....	1
BÖLÜM 2.	
TEMEL TANIM, KAVRAM VE YÖNTEMLER.....	5
2.1. Görüntü İşleme .....	8
2.2. Nesne Tanıma ve Tespit .....	9
2.3. Veri İlişkilendirme .....	10
2.4. Nesnelerin Yeniden Tanımlaması.....	10
2.5. Çoklu Nesne Takibi.....	11
BÖLÜM 3.	
TEMEL YÖNTEMLER VE LİTERATÜR ARAŞTIRMASI .....	13
3.1. Video Ön İşleme .....	13
3.2. Nesne Algılama ve Tespiti .....	14
3.3. Kalman Filtre .....	16
3.4. Siyam Ağları .....	16
3.5. Benzerliğe Dayalı Yeniden Tanımlama ve İzleme .....	20

3.6. Benzerliğe Dayalı Veri İlişkilendirme.....	22
BÖLÜM 4.	
BULGULAR VE DEĞERLENDİRME.....	25
4.1. Kullanılan Yazılımlar ve Kütüphaneler.....	25
4.1.1. Programlama dili .....	25
4.1.2 Makine öğrenimi için platform.....	26
4.1.3 Programlama fonksiyonları kütüphanesi .....	26
4.1.4 Yapay sınır ağı için kütüphane.....	26
4.1.5 CUDA .....	26
4.1.6 Metrikler .....	27
4.2. Veri Seti.....	27
4.3. Deneyler.....	29
4.4. Konfigürasyonlar .....	30
4.5. MOT Yöntemlerle Karşılaştırma .....	32
BÖLÜM 5.	
TARTIŞMA VE SONUÇLAR .....	40
KAYNAKLAR.....	42
ÖZGEÇMİŞ .....	47

## **SİMGELER VE KISALTMALAR LİSTESİ**

AUC	: Area under the Curve
CNN	: Convolutional Neural Network
GPU	: Graphic Processing Unit
MOT	: Multiple Object Tracking
NMS	: Non-Maximum Suppression
Re-ID	: Re-Identification
RGB	: Red Green Blue
SORT	: Simple Online Realtime Tracking
YOLO	: You Only Look Once

## ŞEKİLLER LİSTESİ

Şekil 2.1. Giriş karesi- ham görüntü, yani yayalar.....	7
Şekil 2.2. Çıkış karesi- tespitler ve takip ID'ler.....	7
Şekil 3.1. DenseNet yapısı (Huang ve ark., 2016).....	18
Şekil 3.2. DenseNet açıklaması.....	18
Şekil 3.3. Siyam Ağların eğitim ve doğrulama grafiği .....	19
Şekil 3.4. Siyam Ağların benzerlik hesaplama örnek çıktısı.....	20
Şekil 3.5. Çevrimiçi çok nesneli izleme çerçevemize genel bakış.....	21
Şekil 3.6. Benzerlik dizisi tabanlı tanımlama ilkesi .....	24
Şekil 4.1. Motmetrics için veri setlerin CSV format örneği .....	28
Şekil 4.2. Siyam Agi kullanarak tek seferde öğrenme gösterimi .....	29
Şekil 4.3. SAT'ın farklı konfigürasyonlarda MOTA değer grafiği .....	31
Şekil 4.4. MOT16 veri setinde SAT'ın ile diğer yöntemlerinin MOTA grafiği.....	34
Şekil 4.5. MOT16 veri setinde SAT'ın ile diğer yöntemlerinin IDs grafiği .....	35
Şekil 4.6. MOT17 veri setinde SAT'ın ile diğer yöntemlerinin MOTA grafiği.....	36
Şekil 4.7. MOT17 veri setinde SAT'ın ile diğer yöntemlerinin IDs grafiği .....	36
Şekil 4.8. Ortalama F puanlarına göre hayatta kalma eğrileri (MOT16).....	37
Şekil 4.9. Ortalama F puanlarına göre hayatta kalma eğrileri (MOT17).....	37
Şekil 4.10. Performans metrikleri arasındaki ilişkisi .....	38
Şekil 4.11. MOT16-09 veri setinde 32 karesi .....	39
Şekil 4.12. MOT16-09 veri setinde 49 karesi .....	39



Şekil 4.13. MOT16-09 veri setinde 52 karesi .....	39
---	----

## TABLolar LİSTESİ

Tablo 4.1. MOT17 veri seti üzerinde çerçevemizin ablasyon çalışması.....	31
Tablo 4.2. SAT'ın MOT16 veri setinde çerçevemiz ile diğer yöntemlerinin karşılaştırılması.....	34
Tablo 4.3. SAT'ın MOT17 veri setinde çerçevemiz ile diğer yöntemlerinin karşılaştırılması.....	35

## ÖZET

Anahtar kelimeler: Çoklu Nesne Takibi, Derin Siyam Sinir Ağı, Benzerlik Dizisi, Yeniden Tanımlama

Hareket eden nesnelere tespit ve izleme süreci, bir başlangıç nesne seti olarak zaman içinde birkaç video karesi boyunca hareket ederken nesnelere izlenmesini içerir. Çoklu nesne izlemede (MOT), bir nesne izleme gerçekleştirme süreci, tespit ve ilişkilendirmeleri kapsar. Nesne izlemenin genel amaçları, ilgili tüm nesnelere yerleştirilerek, tanımlayarak ve bunları video karelerinde izlemeye devam ederek kareler arasında tespit etmeyi ilişkilendirmektir.

İyi bir MOT yaklaşımı, tek bir çerçevede birden fazla nesne bulmayı ve bu çerçeveden kimlik bilgilerini çıkarmayı amaçlar. Bir izleyici, karelerde tespit eksik olsa bile nesnelere takip ederek sahnelerdeki nesnelere her biri için sürekli bir kimliğe sahip olmalıdır. Nesnelere tıkanıp veya geçici olarak çerçevenin dışına çıktığı için MOT sorunları zordur. Bu çalışmada, Derin Siyam Ağı kullanılarak benzerlik temelli bir kişiyi yeniden tanımlama çerçevesi önerilmiştir. Çerçeve, tespit yaptıktan sonra kişinin yeniden tanımlanması için bir benzerlik dizisi kullanarak ve video dizisinin her karesi için nesne benzerliğini göz önüne alıyor.

MOT16 ve MOT17 kıyaslamalarına dayalı deneysel sonuçlar, önerilen çerçevenin çeşitli izleme ölçümlerinde diğer metotlarla karşılaştırıldığında daha iyi performans gösterdiği gözlemlenmiştir.

# **SIMILARITY-BASED PERSON RE-IDENTIFICATION FOR MULTI-OBJECT TRACKING USING DEEP SIAMESE NETWORK**

## **SUMMARY**

Keywords: Multiple Object Tracking, Deep Siamese Neural Network, Similarity Array, Re-Identification

The process of detecting and tracking moving objects involves tracking objects as they move across several video frames over time taking an initial set of object detection. In multiple objects tracking (MOT), the process to perform an object tracking two common steps include detection and associations. The general aims for object tracking are to associate detection across frames by localizing and identifying all objects of interest and keep tracking them across video frames.

A good MOT approach aims to find multiple objects in an individual frame and extract the identity information from that frame. A tracker should have a continuous ID for each of the objects within the scenes by keeping track of objects even when the detection is missing in the frames. MOT problems are challenging since objects occluded or temporarily go out of frame. In this work, we propose a similarity-based person re-identification framework using the Deep Siamese Network. Our framework is using a similarity array for person re-identification after detection is executed and examines the object similarity for each frame of the video sequence.

Experimental results based on the MOT16 and MOT17 benchmarks show that our proposed framework outperforms the state-of-the-art performance on several tracking metrics.

## **BÖLÜM 1. GİRİŞ**

Mobil cihazların popülaritesi nedeniyle resim, video vb. multimedya verileri eskisinden çok daha fazla. Sonuç olarak, bu tür multimedya verilerinin otomatik olarak işlenmesi ve anlaşılması oldukça talep edilmektedir. Görüntüleri ve genel olarak gerçek dünyadan yüksek boyutlu verileri elde etme, işleme, analiz etme ve anlama yöntemlerini içeren bir alandır. Gerek akademik önemi gerekse ticari uygulamaları ile her geçen gün daha fazla ilgi görmektedir. Bilgisayarlı görüde izleme önemli bir rol oynamaktadır. Orta düzey bir görev olarak izleme, akademik araştırmalarda eylem tanıma, olay algılama gibi görevlerin temelidir. Bu arada izleme, ticari uygulamalar için görsel gözetim ve sanal gerçeklikte de temel bir rol oynamaktadır.

Makine zekasında, görsel bilgi, insan bilgisayar etkileşimi, video açıklama ve endüstriyel robotlar gibi birçok uygulamada da faydalı bulunmuştur. Bununla birlikte, günümüzde makine zekalarının çoğu dikkatli bir şekilde tasarlanmıştır ve genellikle yalnızca sınırlı senaryolarda çalışabilir ve dünya ile “insan benzeri” bir şekilde etkileşime girebilecek zekâ oluşturmak son derece zordur. Bu nedenle, nesne algılama ve izleme ile ilgili birkaç farklı algoritma test etmektedir ve farklı modelleri ile arka plan çıkarmayı kullanan birçok algoritma uygulanmaktadır.

Video islemesi, güvenlik tehditlerini otomatik bir şekilde yönetmek, yönlendirmek veya belirlemek amacıyla bir sahnedeki faaliyetleri, davranışları ve değişen faaliyetleri izleme eylemidir. Olağandışı faaliyetleri yakalamak veya gözlem altındaki sahneyi insan çabası olmadan yönlendirmek ve planlamak için gelişmiş görme algoritması kullanarak veri toplama ve bu verilerin analizi için gelişmiş bilgisayar teknolojisinin kullanılması, video işleminin ideal hedefidir. Kameraların doğru kullanımı ile hırsızlık, soygun, terörizm veya diğer suç faaliyetleri gibi güvenlik tehditleri kontrol altına alınabilmektedir. Buna ek olarak, karayolu izleme gözetim sistemi yaya ve trafik

davranışlarını, trafik kural ihlallerini, kazaları vb. anlamak için kullanılabilir. Otopark yönetimi, endüstrilerde tehlike izleme, sınır izleme ve plaka tanıma sistemleri video gözetiminin diğer popüler uygulamalarıdır.

Halihazırda alışveriş marketleri, oteller, devlet ve özel sektör ofisleri, otoyollar ve ana yollar 24 saat video gözetimi altındadır. Geniş alan gözetiminde, yüksek binalara ve kulelere monte edilmiş kameralardan gelen video beslemesi veya İnsansız Hava Araçları (İHA) üzerindeki kameralardan kaydedilen video, olağandışı faaliyetlerin tespit edilmesi ve tanımlanması için kullanılmaktadır. Kameralar yüksek irtifalara yerleştirildiğinde geniş bir alan kapsanabilmekte, ancak bu ek fayda, özellikle düşük resim çözünürlüğü açısından görüntü kalitesi maliyetine yol açmaktadır. Yüksek irtifa kameralarının getirdiği diğer zorluklar, çeşitli çevresel faktörler ve hareket değişiklikleri nedeniyle aydınlatma ve aydınlatma değişiklikleridir. Şiddetli rüzgâr nedeniyle kamera hareketleri veya hava aracının kendi hareketi, gözetim sisteminde ciddi sorunlara neden olabilmektedir.

Bilgisayarla görme algoritmaları, bir sahneyi otomatik olarak daha iyi anlamak için kullanılabilir. Şekiller, aydınlatma ve renk dağılımları gibi önemli özellikleri görüntülerden ve video dizilerinden insanın göz ve beyin görsel sistemi tarafından yapıldığı gibi çıkarmaktadır. Başka bir deyişle, bilgisayar görüşü, gözlem altındaki sahnenin gerçek zamanlı yorumunu sağlamakta ve sistemin acil bir yanıt gerektirmesi durumunda uyarılmaktadır. Bir makine izlenen bir sahneyi anlayabildiğinde ve olağandışı tepkiler hakkında uyardığında, tek bir kişi tarafından çok sayıda video sensörü kullanılarak çok geniş bir alan gözlemlenebilmekte ve kontrol edilebilmektedir. Bu nedenle, çok geniş bir alanın sürekli ve odaklanmış olarak izlenmesi, düşük bir maliyetle mümkün hale gelmektedir. Bu yapay zekâ alanı, video gözetiminin ana bileşenleri olan sahne yeniden yapılandırma, nesne algılama, tanıma, izleme ve hareket tahmini gibi çeşitli alt alanları içermektedir.

Görüntü ve video analizi alanları, kısmen derin öğrenme modellerinin başarısı nedeniyle son yıllarda büyük ilgi gördü. En gelişmiş yöntemlerin çoğu, manuel özellik çıkarımına gerek kalmadan iyi sonuçlar ürettiği gösterilen derin ağlar olan evrişimli

sinir ağlarını (CNN'ler) kullanır. Buradaki zorluk, amacın her bir görüntüyü birden fazla sınıftan biriyle etiketlemek olduğu görüntü sınıflandırmasındaydı. CNN'lerin kullanımı görüntü tanıma ile sınırlı kalmamıştır, ancak nesne algılama (belirli sınıfların nesnelerinin yerleştirilmesi), anlamsal bölümlenme (görüntüdeki her pikseli etiketleme) ve görüntü altyazısı (açıklayıcı bir metin üretme) gibi görevlerde başarıyla benimsenmiştir.

Görsel izleme, video gözetimi ve insan bilgisayar ara yüzü gibi alanlardaki sayısız uygulamaları nedeniyle toplumda önemli bir sorundur. Belirli bir görüntü dizisindeki bir nesnenin durumunu (boyut, konum vb.) tahmin etmeyi amaçlamaktadır. İlk çerçevede, ilgilenilen nesnenin açıklaması manuel olarak veya tespit edilerek verilmektedir. İlgili nesneyi izlemek için bir gözlem modeli ve bir dinamik model gereklidir. Dinamik model, mevcut çerçevenin durumuna bağlı olarak bir sonraki çerçevedeki adayları elde etmektir. Gözlem modeli, adaylardan çıkarılan gözlemler ile nesnenin güncel durumu arasındaki benzerliği ölçmektedir.

Nesne izleme, nesnelere hakkında bilgi toplamak için elde edilen bilgilerinin kullanılması sorununu ifade etmektedir. İzlemenin tipik amacı, hız, konum veya özellikler gibi sahne nesnelerinin durumlarını belirlemektir. Genel olarak, izleme görevleri, izlenen hedef sayısına göre iki kategoriye ayrılabilir: Tek Nesne İzleme (SOT) veya Çoklu Nesne İzleme (MOT). MOT, aynı anda birden fazla ilgi nesnesi (önceden tanımlanmış sınıflardan) için bir yörünge oluşturmakla ilgilidir. Trafik izleme için ilgilenilen kişileri veya araçları belirlemek için şehirlerde gözetim sistemleri konuşlandırılmaktadır. Aynı zamanda, kişileri izleme şehirdeki ortak alanları daha güvenli hale getirmektedir. Öte yandan SOT, görünümü önceden bilinen bir ilgi hedefini izlemekle ilgilidir. Kritik fark, MOT'un olay yerindeki tüm mevcut hedefler arasındaki ilişkileri tanımlaması gerektiğidir.

Yapay zekanın ana hedeflerinden biri, bilgisayarların çevrelerindeki dünyayı anlamalarını sağlamaktır. Dünyayı anlamının temel adımlarından biri, görüş alanımızdaki belirli nesnelere bulmak ve bu nesnelere hareket ederken takip etme görevidir. Hem nesne tanıma hem de izleme kendi başlarına alanlardır, ancak her ikisi

de bilgisayarların görsel sahneleri anlamasını sağlamanın zorunlu bir parçasıdır. Nesne tanıma algoritmalarının geleneksel olarak karşılaştığı ana zorluk, doğal sınıfların izin verdiği geniş sınıf içi çeşitliliktir. Otomobiller gibi orta derecede kısıtlı bir sınıfta bile, yüzlerce farklı model, bu modellerde varyasyonlar ve boya renkleri, sonsuz çeşitlilikteki görüş açılarından, aydınlatma koşullarından ekilebilmektedir. Tüm bu varyasyonları göz önünde bulundurduğunuzda, bilgisayarların nesnelere doğru bir şekilde sınıflandırmak için gerekli genellemeleri yapmakta neden bu kadar zorlandıklarını görmek oldukça doğaldır.

Bu tezin odak noktası, çoklu nesne izleme ve takibi problemini çözmek için algoritmaların geliştirilmesi üzerine kurulmuştur. Bu problem, geleneksel çoklu nesne izleme probleminin genelleştirilmesidir. Hem görsel izleme hem de çoklu nesne izleme sorunlarıyla yakından ilgilidir. Bu çalışmanın geri kalanı şu şekilde organize edilmektedir: BÖLÜM 2'de konuyu ile ilgili önceki ve benzer çalışmalar gözden geçirilmiştir. Araştırma çalışması sırasında yapılmış olan literatür taramaları burada tartışılmıştır. Ayrıca nesne algılama ve nesne takibi ile ilgili literatürün ayrıntılı bir incelemesini de sağlamaktadır. Nesnelerin tespiti ve takibi için mevcut ve bazı yeni yöntemler hakkında özet yapılmaktadır. BÖLÜM 3'te önerilen yöntemin tam açıklaması verilmiştir. Bu bölümde, bu proje çalışmasıyla ilgili arka plan kavramları tartışılmıştır. İzleme akış sistemlerinin mimarisi ve blok diyagramı da bu bölümde açıklanmıştır. BÖLÜM 4, uygulamanın detayları ve deney sonuçlarının raporu açıklanmıştır. Bu bölüm, daha sonra bazı mevcut ve deneysel sonuçların metodolojisini ve uygulamasını sunmaktadır.



## **BÖLÜM 2. TEMEL TANIM, KAVRAM VE YÖNTEMLER**

Bilgisayarda görüntü işleme popüler alanlarından biri nesne izleme ve takibidir. Bu teknoloji yalnızca güvenlikle ilgili yazılımlarda değil, aynı zamanda insanlar ve bilgisayarlar arasındaki gelişmiş ara yüzde de önemlidir. Bu teknolojideki zorluklardan biri çoklu nesne takibidir. MOT'taki çoğu çalışma, tespit edilen nesnelerin görünümüne, hareketlerine ve etkileşimlerine dayalı olarak video kareleri arasında ilişkilendirildiği, tespit yoluyla izleme paradigmasını kullanmaktadır. Bu bölümde, MOT metodunda kullanılan geliştiren ana yaklaşımlar tanımlanmıştır. Literatürü inceledikten sonra, video dizisinden nesneyi tespit etmenin ve nesneyi takip etmenin gerçekten zor bir iş olduğu bulunmuştur. Nesne izleme, videoda bulunan veri miktarı nedeniyle zaman alan bir süreç olabilir. Literatür taramasından hem iç hem de dış gözetim sistemlerinde verimli çalışan birçok arka plan çıkarma algoritması çıkışı olduğu bulunmuştur.

Bilgisayar görüntü işleme uygulamalarının vazgeçilmez konulardan olan nesne tespiti ve nesne tanıma uzun yıllardır üzerinde çalışılan konulardır. Nesne tespiti ve nesne tanıma için farklı algoritmalar ve yöntemler geliştirilmiştir. Video işlemedeki en büyük zorluklardan biri sahneyi anlamaktır. Bu konu birçok uygulamada (örneğin otomatik sürüş, güvenlik video işleme, spor analizi ve robot navigasyonunda) önemli bir rol üstlenebildiğinden bilgisayarla görmede kritik bir sorundur ( Y. Zhang ve ark., 2020; Cakir & Cetin, 2021; Yoon ve ark., 2021). Bu sistemlerin çoğunda, kişi ana ilgi kaynağı olabilir. Bu da farklı algoritmalar kullanarak onları tespit ve takip etme sorunlarına yol açmasına neden olmuştur. (Braso & Lear-Taixe, 2020). Genel olarak nesne izleme süreçlerinde, tespit yoluyla izleme, MOT uygulamalarında ana paradigma olarak kullanılmaktadır (Wojke ve ark., 2018; Chen ve ark., 2019; Y. Wu ve ark., 2019; Yoon ve ark., 2021).

Derin öğrenme algoritmalarına dayalı olarak nesne tespiti için en son önerilen MOT yöntemleri, yüksek performans sonuçlarına sahiptir (Ciaparrone ve ark., 2020; Y. Wu ve ark., 2019). Bu paradigmayı kullanan yöntemler, video kareleri arasında nesne tespiti ilişkisine odaklanmaktadır (F. Yang ve ark., 2021). Önerilen sistem, videonun her karesini girdi olarak alır ve görüntüyü işledikten sonra, nesnelere tespit ederek ve her birini sınırlayıcı kutu ve izleme/takip kimliği gibi özellikleriyle işaretler.

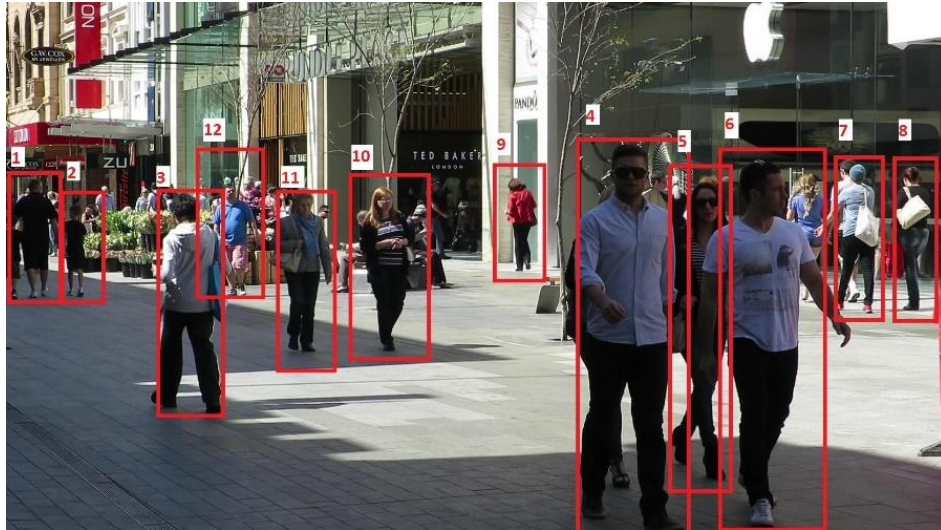
Veri ilişkilendirme yaklaşımları yalnızca mevcut ve önceki karelerden alınan verilerin kullanıldığı çevrimiçi yöntemler ve öğrenmenin gerçek izden önce gerçekleştirildiği çevrimdışı yöntemler olmak üzere iki kategoriye ayrılabilir (Chen ve ark., 2019). Çevrimiçi öğrenme, verilerin sıralı ve gelecekteki adımlar için tahmin etmek için algoritma tarafından gerçek zamanlı olarak kullanıldığı bir yöntemdir. Özellikle videolu gözetleme uygulamalarında araç ve kişi gibi kategorik nesnelere takip etmek için nesne takibi öğrenmenin en çok kullanılan yoludur. Çevrimdışı öğrenme, statik bir girdi verisi seti alır ve eğitim süreci tamamlandıktan sonra hedef fonksiyonun yaklaşımını değiştirmemektedir. Diğer birçok makine öğrenimi uygulamasında iyi bilinen bir yaklaşım olmasına rağmen, bu, 2015 yılına kadar kullanılmayan izleme alanında oldukça yeni bir konudur. Bu yaklaşımın beklentisi, sahnelerde rastgele hedeflerin izlenmesi sorununu ele alırken performansı iyileştirmektir.

Bu çalışmada öncelikle videoda kişi takibine odaklanılmıştır. MOT senaryolarındaki kişi nesnelere genellikle ağır tıkanıklıklarda, özellikle kalabalık sahnelerde meydana gelmektedir. Bunun gibi olaylar kişileri ayırmayı zorlaştırır (Liu ve ark., 2020). Ayrıca, kişilerin izlendiği MOT uygulamalarında kalabalık yerlerde, nesnelere farklı poz varyasyonlarından etkilenebilmektedir (Xu ve ark., 2019). Diğer bir zorluk ise, uzun bir süre boyunca izleri çevrimiçi senaryolarda ile birleştirmektir (Sadeghian ve ark., 2017). Çevrimiçi yöntemlerde, çerçevenin birden fazla nesnesi vardır. Bu durumda mevcut kareye ek olarak nesnelere yalnızca geçmiş karelerden gelen bilgileri kullanabilir. Ayrıca, tespit edilen nesnelere kamera görüntüsünden çıkarsa ve başka bir nesne belirirse, bunların aynı nesnelere olup olmadığını bilmek zordur.

Aynı tür nesne örneklerine ait, yani örnek bir video karelerindeki kişiler için sıralı çerçeveler boyunca çoklu nesne takibine genel bir bakış Şekil 2.1. ve Şekil 2.2.'de gösterilmektedir. Nesne algılamayı kullanan veya bazen statik nesne algılama olarak adlandırılan tipik bir senaryo gösterilmektedir. Bu senaryoda, nesne algılama modeli, resimde yerleştiren sınırlayıcı kutular çıkarır. Nesne algılama görevi, verilen bir görüntüdeki nesnelerin sınıflarını ve konumlarını tahmin etme problemi olarak tanımlanabilir. MOT yöntemi sadece nesnelere lokalize etmekle kalmaz, aynı zamanda onları doğru yörüngelere atayarak zamansal bilgilerin de kullanılmasını gerektirir.



Şekil 2.1. Giriş karesi- ham görüntü, yani yayalar



Şekil 2.2. Çıkış karesi- tespitler ve takip ID'ler

MOT'ta bir nesnenin görünümü, her tespit için ID atama sorunlarını çözen veri ilişkilendirmesi için önemli bir öge olarak kullanılmaktadır. Bununla birlikte, özellikle benzer görünüme sahip nesnelere (örneğin, araçlar veya kişiler) takip etmek için, birden çok nesneyi sınıflandırmak için yalnızca görünüm yeterli değildir (Xu ve ark., 2020). Buna bağlı olarak, MOT uygulamalarındaki ana zorluk, nesnelere yalnızca geçmiş karelerden alınan bilgilerle ilişkilendirildiği çevrimiçi yöntemlerde kareler arasında nesnelere veri ilişkilendirmesidir (Chu ve ark., 2017; M. Yang ve ark., 2017; Y. Wu ve ark., 2019; Y. Zhang ve ark., 2020).

Bahsedilen problemlerin üstesinden gelmek için, benzerlik dizisi ile temsil ve tespit edilen nesnelere arasındaki bilgileri kullanmak için benzerliğe dayalı yeniden tanımlamayı kapsayan bir yaklaşım göz önüne alınmaktadır. Başka bir ifadeyle bu çalışmada, beklenmedik nesne hareketlerini ve belirsiz tespit neden olduğu olasılıkları (örneğin, yanlış negatifler ve pozitifler) dikkate alan, çevrimiçi MOT için nesnelere arasındaki verileri işleyecek yeni bir veri ilişkilendirme yöntemi önermekteyiz. Önerilen MOT çerçevesi sadece bir ilişkilendirme adımından oluşmaktadır. Her karede yeni tespit alındığında, bir tespit kümesi ve bir takip kümesi olmak üzere benzerlik dizisinde iki kümemiz vardır. Ardından, takip işlemi başlatılan bir iz ile ilişkilendirilmesiyle ağımızın girişi oluşturulmaktadır.

Başlıca katkılarımız şunlardır: (1) Her karedeki nesnelere arasındaki benzerliği değerlendirmek için derin bir Siyam ağı kullanımı önerilmiştir; (2) Nesnelere arasındaki ilişkilendirme problemini çözebilecek bir benzerlik dizisinin kullanılması önerilmiştir ve (3) tasarlanmış derin Siyam ağına dayalı önerilen MOT algoritmasının MOT-Challenge veri setleriyle test işlemleri sunulmuştur.

## 2.1. Görüntü İşleme

Günümüzde video ve görüntü işleme, astronomi, tıp, spor, sinema, üretim endüstrisi ve robot navigasyonunda gibi birçok farklı uygulama alanında kullanılmaktadır. Bu uygulamaların çoğu benzer video ve görüntü işleme yöntemlerine dayanmaktadır.

Özellikle bilgisayar işlem gücü ve ağ iletim bant genişliği açısından hızlı teknolojik gelişmeler, birçok başarılı uygulama ile sonuçlanmaktadır.

Bu veri alanlarının yaygın olarak kullanılan uygulamaları, konuşma tanıma, regresyon problemi, örüntü tanıma ve metin tanıma gibi sınıflandırabilmektedir. Bunların dışında bir diğer büyüleyici veri yöntemi video verisidir. Bununla birlikte, video verileri, büyük boyutu açısından araştırma için de ilgi çekicidir. Günümüzde, çoklu sensorlar tarafından ölçülen çok sayıda görüntü ve video bulunmaktadır. Bu görüntüler ve videolar, hedefleri tespit etmek, tanımak ve izlemek için kullanılabilir (Sharma ve ark., 2021). Bu şekilde, savaş alanı gözetimi, yüz tanıma, jeolojik araştırma vb. için anormallikler, ilgili hedefler veya bölgeler bulunabilmektedir.

Yapay zekanın gelişmesiyle, sinir ağları, uzman sistemler ve akıllı algoritmalar içeren çeşitli etkili araçlar ortaya çıkmıştır. Bunlar, özellik çıkarma veya karar verme dahil olmak üzere görüntü ve video analizi için esnek bir şekilde ayarlanabilmektedir. İnternete her gün milyonlarca video verisi yüklenmektedir. Video verileri, zamanla değişen herhangi bir görüntü dizisidir. Videolarda resim bilgisi hem uzamsal hem de zamansal olarak sayısallaştırılmaktadır.

## **2.2. Nesne Tanıma ve Tespit**

Nesne tanıma, otonom araçlar, robotik, güvenlik sistemler gibi birçok alanda yaygın olarak uygulanan bilgisayar görüntü işleme uygulamalarından biridir. Derin öğrenmenin hızlı gelişimi birçok algoritma video analiz ve görüntü anlamayı geliştirmektedir (Mahendru & Dubey, 2021). Görüntü işleme ve makine öğrenimi algoritması, ortamdaki ve çevresindeki nesnelere algılamak için önemli bir araçtır (Kavitha & Nivetha, 2021). Derin öğrenme tekniklerinin ortaya çıkmasıyla, nesne algılamasının doğruluğu büyük ölçüde artmaktadır. Dijital görüntülerde ve videolarda nesnelere veya belirli bir sınıfın (insanlar, çiçekler, hayvanlar gibi) örneklerini tanımaya odaklanmaktadır. Nesne tanımanın amacı, bir bilgisayar kullanarak nesnenin hangi kategoriye ait olduğunu belirlemektir.

### 2.3. Veri İlişkilendirme

Tespit yoluyla izleme çerçevesi iki modelden oluşmaktadır: tüm nesnelere tanımlamak için bir nesne detektörü ve izlemeyi geliştirmek için bir veri ilişkilendirme modeli. Bu görev için iki parçanın eşit derecede kritik olmasına rağmen, çoğu MOT çalışması esas olarak veri ilişkilendirmesine odaklanmaktadır. MOT, birden çok nesnenin izlerini oluşturmak için her karede mevcut izler ve yeni tespitler arasındaki veri ilişkilendirmesinden yararlanmaktadır.

Yeni bir yaklaşım olarak ilgili izleri düzenlemek ve bunları uzun yörüngelerle ilişkilendirmek, arabirimleri gürültülü veya kafa karıştırıcı nesne tespiti sonuçlarından modelleyerek ve azaltarak performansı artırmaktadır (Peng ve ark., 2020). Çok görevli evrişimli sinir ağı, çoklu nesne takibi için ilişkilendirmeyi izlemek ve bir nesnenin görünüm modeline odaklanmak için önerilmektedir (Chen ve ark., 2019). Ayrıca, görünüm modellerini değiştirmek ve bir sonraki karede uzamsal-zamansal dikkat mekanizmasını kullanarak nesne hedeflerini aramak, hedefler arasındaki tıkanmanın nedenini azaltmaktadır (Chu ve ark., 2017). Grafik modellerine dayalı metotlar, veri ilişkilendirmesinin kesinliğini ve çoklu nesne takibi için benzerlik modelinin sağlamlığını iyileştirmektedir (Braso & Lear-Taixe, 2020; Liu ve ark., 2020; Y. Zhang ve ark., 2020). Önerilen metot, tespit yoluyla izleme paradigmasına dayanmaktadır. İlk olarak, video kareleri arasında bir detektör çalıştırılmakta ve ikinci olarak, tespitleri ilişkilendirmek için takip edilen nesnelere birleştirilmektedir. İki tespiti aynı nesneye ait olup olmadığını hesaplamak için bir benzerlik modeli öğrenerek veri ilişkilendirme problemine odaklanma önerilmektedir.

### 2.4. Nesnelere Yeniden Tanımlama

Kişi yeniden tanımlama, videosunda kişileri tanıma ve eşleştirme sorunudur (Q. Wu ve ark., 2021). Re-ID, aynı kişinin farklı kameralarla veya aynı kamerayla farklı zamanlarda çekilmiş görüntülerinin kişilere tekrar ID atanmasıdır. Genellikle yeniden tanımlama kısa bir süre ve küçük bir alanla sınırlıdır. Bu görev, çeşitli bakış açıları, karmaşık ortam, sınırsız pozlar, oklüzyonlar vb. farklı alanlara meydana gelmektedir

(Bergmann ve ark., 2019; Nousi ve ark., 2020) . Kişi görünümü bu koşullara göre değişebilir ve bu da yeniden tanımlamayı zorlaştırmaktadır (Yu ve ark., 2019). Tipik olarak, standart bir yeniden tanımlama sistemi üç ana bileşen içermektedir: özellik üretim yaklaşımlarını geliştirmeye odaklanan özellik temsili öğrenme, çeşitli kayıp fonksiyonu veya örnekleme stratejileri ile eğitim hedeflerini planlamak için derin metrik öğrenme ve iyileştirmek için sıralama optimizasyonu için alınan sıralama listesi. Son yıllarda, video tabanlı yeniden tanımlama uygulamaları büyük gelişmeler göstermektedir.

Video dizileri, kişiyi yeniden tanımlama deneylerinde iyi sonuçlar veren metrik öğrenme algoritmaları için görsel ve zamansal bilgiler sağlamaktadır (Chen ve ark., 2019). Y Li ve ark. ağın öznelik çıkarma yeteneğini re-ID geliştirmek için bir öznelik toplama yöntemi önermektedir (Y. Li ve ark., 2021). P. Nousi ve ark. sınıflandırmaya dayalı yeniden tespit ve izleme için, bu sonuçlarının nesne yeniden tanımlanmasını içeren uzun vadeli bir izleme çerçevesi önermektedir (Nousi ve ark., 2020). Y Lin ve ark. verisinin özellik arasındaki Öklid mesafesine göre denetimsiz yeniden kimlik sorununa odaklanmasını önermektedir (Lin ve ark., 2020). MOT16 veri seti (Milan ve ark., 2016)'dan alınan görüntü örnekleriyle Şekil 3.4.'te önerilen benzerlik dizisinin kişinin yeniden tanımlanması sorununu gösterilmektedir. Benzerlik ve k-karşılıklı yeniden sıralama algoritması kullanan öznelik tanıma modelini tanıtarak yeniden kimlik performansını iyileştirdiğini iddia etmekte ve MOT17'ye kıyasla iki ilkel veri seti üzerinde rekabetçi sonuçlar elde etmektedir (Mansouri ve ark., 2021).

## 2.5. Çoklu Nesne Takibi

Çoklu Nesne Takibi (MOT), akademik ve ticari potansiyeli nedeniyle ilgiyi artırmaktadır. Bu sorunla yüzleşmek için farklı yöntemler önerilmiş olsa da hızlı görünüm değişiklikleri ve nesne tikanıklıkları gibi nedenlerle zor olmaya devam etmektedir. Daha önce de belirttiğimiz gibi, hareketli nesnelerin tespiti ve hareket tabanlı takip, birçok bilgisayar görüntü işleme uygulamasının önemli bileşenleridir. Çoklu nesne takibi iki ana bölümden oluşmaktadır: videoda her karedeki hareketli

nesnelerin tespit edilmesi ve zaman içinde tespit edilen nesneye karşılık gelen takiplerin ilişkilendirilmesi.

Çoğu yaklaşım, bu iki adımdan oluşan tespit yoluyla izleme paradigmasını takip etmektedir (Meinhardt ve ark., 2021). Bazı yaklaşımlar bir izle ilgili son tespiti belleğe alırken, bazı teknikler zamansal bilgileri bir izleme geçmişinde birleştirmektedir (Wang ve ark., 2020; J. Zhang ve ark., 2020). Örneğin, Derin Sıralama (DeepSORT) algoritması, tespit ile önceden iz ile ilişkilendirilmiş herhangi bir takip nesne arasındaki benzerliği hesaplamaktadır (Wojke ve ark., 2018). B. Suhai ve ark. Siyam Sinir Ağlarını MOT metotlarda kullanmanın etkinliğini ve MOT'un zorlukları üzerinde iyi performans sağladığını gösterdi (Shuai ve ark., 2020). Nesnelerin arasındaki benzerliği ölçmenin bir yolu, video kareler arasındaki nesnelerin benzerliğinden kimlik bilgilerini çıkarmaktır. Diğer yol, aynı bilgiyi nesneler arasındaki etkileşimden çıkarmaktır. Çalışmamıza benzer şekilde, K. Yang ve ark., farklı katmanlardan özellikler üreten çeşitli dikkat mekanizmalarını birleştiren bir siyam ağı kullandılar ve yöntemlerini seçilen OTB100 (Y. Wu ve ark., 2013) veri setleri üzerinde test ederek 0,628 AUC puanı elde ettiler. Meimetus ve ark., çalışmasında, DeepSORT'un, nesne başlatmayı ve bir nesnenin önceki karelerdeki takibini göz önünde bulundurarak mantığı değiştirilmiş bir sürümünü kullanmıştır. Çalışmalarında tespit ve takip performansında iyileşme olduğunu iddia etmişlerdir.



## **BÖLÜM 3. TEMEL YÖNTEMLER VE LİTERATÜR ARAŞTIRMASI**

Bir video dizisinde izleme yapmanın birkaç yolu vardır. Sosyal alanlardaki videolar gibi bazen çok kalabalık sahneler için özellikle uygun olan, tespit yoluyla izleme kullanan çoklu nesne takibidir (MOT). Burada tüm nesnelere önce bir nesne detektörü kullanılarak her karede tespit edilir ve ardından tespit edilen nesne konumu ve görünümü kullanılarak kareler arasında ilişkilendirme işlemi gerçekleştirilir.

Geleneksel tespit yoluyla izleme yöntemleri genellikle çok karmaşıktır ve hesaplama açısından ağırdır. Ancak, en son nesne detektörleri tarafından etkinleştirilen daha iyi algılamalarla, basit izleme modelleri kullanılabilir. Son zamanlardaki iki açık kaynak izleyici SORT ve Deep SORT'tur. SORT, Basit Çevrimiçi Gerçek Zamanlı İzleyici anlamına gelir ve izleyici, hareket tahmini ve veri ilişkilendirmesini işlemek için Kalman filtrelemesini ve Macar algoritmasını kullanmaktadır (Bewley ve ark., 2016). SORT, hızlı, basit, yüksek hassasiyet ve doğruluğa sahiptir. Ancak oklüzyonu çok iyi idare edememektedir. Deep SORT, tespit edilen her nesne için bir görsel görünüm tanımlayıcısı kullanarak eşleştirme prosedürünü iyileştiren ve kimlik anahtarlarının sayısını büyük ölçüde azaltan SORT'un bir uzantısıdır (Wojke ve ark., 2018).

### **3.1. Video Ön İşleme**

Otomatik bir görsel gözetim sisteminin performansı, önemli ölçüde, gözlemlenen ortamdaki hareketli nesnelere algılama yeteneğine bağlıdır. Takip etme, hareketi analiz etme veya nesnelere tanımlama gibi sonraki bir eylem, ön plan nesnelere doğru bir şekilde çıkarılmasını gerektirmekte ve hareketli nesne algılamayı sistemin çok önemli bir parçası haline getirmektedir. Bir çerçevedeki bazı bölgelerin ön planda olup

olmadığına karar verebilmek için arka plan yoğunlukları için bir model olmalıdır. Temel olarak nesne algılama yöntemi gri ölçekleme, düzgünleştirme ve görüntü çözünürlüğünü azaltma vb. dahil olmak üzere bir ön işleme adımından başlamaktadır.

Ön işleme aşamasında, hareketli nesnenin ilk adımı algılama işlemi, bir video kamera kullanarak görüntü bilgilerini yakalamaktır. İşlem süresini azaltmak için, tüm işlem için renkli görüntü yerine gri tonlamalı bir görüntü kullanılmaktadır. Gri tonlamalı görüntü, 8 bitten oluşan yalnızca bir renk kanalına sahipken, RGB görüntüsü üç renk kanalına sahiptir. Hareketli nesnelere algılamak için yüksek doğruluk elde etmek amacıyla girdi görüntülerinden gelen görüntü gürültüsünü azaltmak için görüntü yumuşatma gerçekleştirilmektedir. Düzleştirme işlemi,  $m \times m$  pikseli bir medyan filtresi kullanılarak gerçekleştirilmektedir.

Durağan olmayan arka plan, genellikle ilgilenilen nesnenin hareketi dışında sahte bir hareket olarak kabul edilir ve nesnenin algılanamamasına neden olabilmektedir. Bu sorunu çözmek için görüntünün çözünürlüğü düşük çözünürlüklü bir görüntüye indirgenmektedir. Görüntü boyutunu koruyarak görüntünün uzamsal çözünürlüğünü azaltarak düşük çözünürlüklü bir görüntü yapılmaktadır. Düşük çözünürlüklü görüntü, küçük sahte hareketi azaltmak için kullanılabilir. Küçük bir hareket bölgesine sahip olan bu görüntüler, düşük çözünürlüklü görüntülerde kaybolmaktadır.

### 3.2. Nesne Algılama ve Tespiti

Derin öğrenmeye dayalı görsel nesne algılama algoritması temel olarak iki kategoriye ayrılmaktadır (Yanagisawa ve ark., 2018). R-CNN serisi, R-CNN, Fast R-CNN ve Faster R-CNN açısından iki aşamalıdır. Bu algoritmalar bölge önerilerine, ihracat sınırlayıcı kutulara ve sınıflara ihtiyaç duymaktadır. Hepsi bölge tabanlı algoritmalar, Aralarındaki fark CNN özelliğini nasıl kullandıkları ve hesaplamasını nasıl paylaştıklarıdır.

Öte yandan, YOLO ve SSD, sınırlayıcı kutuların tahmininin tek adımda tamamlandığı tek seferlik çözümlerdir. R-CNN tabanlı nesne algılama algoritmaları, görüntü

içindeki görsel nesnelere bulmak için bölgelere dayanmaktadır. Ağlar görüntünün tamamına bakmazlar. Bunun yerine, görüntünün nesneyi içerme olasılığı yüksek olan kısımları taranmaktadır.

Nesne tespiti, dijital görüntülerde veya videolarda çeşitli nesnelere algılanmasını içeren bilgisayar görüntü işleme bir olgudur. YOLO, bir resimdeki çeşitli nesnelere (gerçek zamanlı olarak) algılayan ve tanıyan bir algoritmadır. YOLO'da nesne tespiti bir regresyon problemi olarak yapılmakta ve tespit edilen görüntülerin sınıf olasılıklarını sağlamaktadır.

YOLO algoritması, nesnelere gerçek zamanlı olarak algılamak için evrişimli sinir ağlarını (CNN) kullanmaktadır. Bu algoritma nesnelere algılamak için bir sinir ağı üzerinden yalnızca tek bir ileri yayılım gerektirmektedir. Bu, tüm görüntüdeki tahminin tek bir algoritma çalıştırmasında yapıldığı anlamına gelmektedir. CNN, çeşitli sınıf olasılıklarını ve sınırlayıcı kutuları aynı anda tahmin etmek için kullanılmaktadır.

Redmon ve Farhadi hem nesne algılama hem de sınıflandırma için ortak eğitim gerçekleştirerek bu kadar çok sayıda nesne algılaması elde edebilmektedir (Redmon & Farhadi, 2018). Ortak eğitim kullanarak YOLO9000'i hem ImageNet sınıflandırma veri seti hem de COCO algılama veri seti üzerinde aynı anda eğittiler. Sonuç, etiketli algılama verilerine sahip olmayan nesne sınıfları için algılamaları tahmin edebilen YOLO9000 adlı bir YOLO modelidir.

YOLO ağı, giriş görüntüsünü  $N \times N$  hücrelerinden oluşan bir ızgaraya böler. Doğruluk kutusunun merkezi bir hücrenin içine düşerse, o hücre o nesnenin varlığını tespit etmekten sorumludur. Her ızgara hücresi,  $X$  sınırlayıcı kutu sayısını ve güven puanını, sınıf tahminleriyle birlikte tahmin etmektedir.  $X$  sınırlayıcı kutularının koordinatları, karşılık gelen ızgara hücrelerine göre her sınırlayıcı kutu  $(ax, ay, aw, ah)$  için 4 koordinat tahmin etmektedir. Burada  $ax$ ,  $ay$ , cismin bu ızgaraya göre orta noktasının  $x$  ve  $y$  koordinatlarıdır.  $ah$  değeri, sınırlayıcı kutunun yüksekliğinin ilgili ızgara hücresinin yüksekliğine oranıdır ve  $aw$ , sınırlayıcı kutunun genişliğinin ızgara hücresinin

genişliğine oranıdır. Bundan sonra her kare için bir nesne güven puanı hesaplanmaktadır ve her nesne için sınıf tahmini yapılmaktadır. Bu kutuların çoğunun çok düşük güven puanlarına sahip olacağı için nihai puanı kodun içerisinde tanımlanan değer üzerinde olan kutuları göstermektedir. Ayrıca, Maksimum Olmayan Bastırma (NMS), aynı görüntünün çoklu algılama sorununu gidermeyi amaçlamaktadır. Algoritmamızda bu bilgileri kullanarak nesne tespiti yapılmaktadır ve tespit edilen her nesne için (ax, ay, ah, aw) değerleri kullanarak kesme işlemi uygulayarak, bu resimleri Siyam Ağana iletilmektedir.

### 3.3. Kalman Filtre

Doğrusal bir sistemin durumunu tahmin etmek için bir Kalman filtresi kullanılmaktadır. Kalman filtresi, durum-uzay teknikleri ve özyinelemeli algoritmaların kullanımına dayanan bir tahmin filtresidir. Dinamik bir sistemin durumu tahmin edilmektedir. Bu dinamik sistem, çoğunlukla beyaz gürültü olarak kabul edilen bazı gürültülerden rahatsız olabilmektedir. Tahmini durumu iyileştirmek için Kalman filtresi, durumla ilgili ancak aynı zamanda bozulan ölçümleri kullanmaktadır. Kalman filtreleme iki adımdan oluşmaktadır:

- Tahmin
- Düzeltme

İlk aşamada dinamik model ile durum tahmin edilmektedir. Tahmin adımı, değişkenlerin yeni durumunu tahmin etmek için durum modelini kullanmaktadır. Bu çalışmada hazire olan bir Kalman filtre kütüphanesi kullanılmaktadır.

### 3.4. Siyam Ağları

Kişiyi yeniden tanımlama sorunu için, benzerliklerini öğrenerek çıkarılan görüntü özelliklerini karşılaştırmak gerekir. Bu problem için kullanışlı bir mimari, aynı ağırlıkları paylaşan iki özdeş özellik çıkarıcının bulunduğu Siyam mimarisidir. Siyam ağları, veri çiftlerinin işlenmesi için kullanılmaktadır. Genellikle çıkarılan veri özelliği, çıkarılan iki özellik arasındaki mesafeyi tanımlayan bir ölçü öğrenen bir uzaktan öğrenme algoritmasına iletilmektedir.

Siyam Ağları tabanlı izleyiciler, hassasiyet ve hız dengeleri nedeniyle büyük ilgi gördü (L. ve ark., 2021a). (S. Li ve ark., 2020a)'de, çapraz entropi kaybı işlevini kullanarak sınıflandırma için uzun vadeli izleme için bir indirgeme yöntemi kullanmaktadır. Benzer şekilde, SAT çerçevemizde, eğitim verilerine dayalı olarak nesnenin özelliklerini öğrenmek için denetimli bir eğitim metodolojisi kullanılmaktadır, ardından bilinmeyen nesnelere tahminler yapmaktadır. SAT'ın benzerlik kontrolü kısmındaki fonksiyon denklem 1'de Öklid mesafesi olarak ayarlanmıştır.

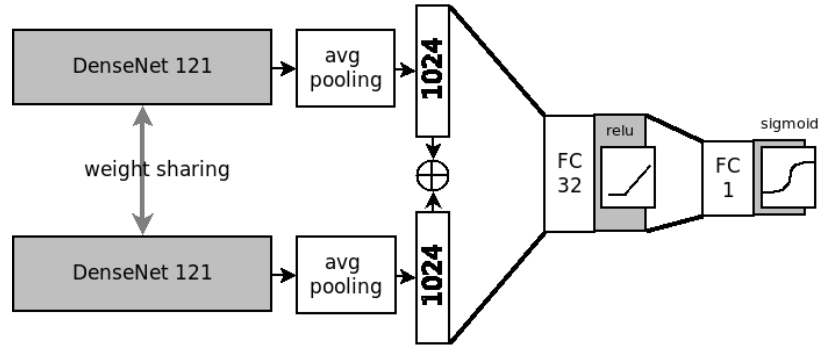
$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.1)$$

Son olarak, SNN çıktılarının kodlanmış temsillerini karşılaştırmak için karşılaştırmalı kayıp kullanma prosedürünü takip eder. Kontrast kaybı ağı çıktısını almaktadır ve içinde verildiği pozitif ve negatif örneklere olan denklem 2'de verilen formül ile mesafesini hesaplamaktadır.

$$L = y \times d^2 + (1 - y) \times \max(\text{margin} - d, 0)^2 \quad (3.2)$$

Burada  $y =$  görüntü çiftleri aynı örneğe aitse 1, aksi takdirde 0'dir,  $d =$  Öklid mesafesidir ve  $\max =$  parametrelerinin maksimumunu alan fonksiyondur. Diğer bir deyişle, temsiller pozitif örnekler için benzer (yakın) ve negatif örnekler için farklı (uzak) ise kayıp düşüktür.

Bu ağ, görüntü çiftleriyle deneyler yapmak için kullanılmaktadır. Görüntülerden öznelikleri çıkarmak için bir dizi evrişimli katman kullanmakta, ardından maksimum havuzlama, ardından bir etkinleştirme işlevinden geçme ve toplu normalleştirme kullanılarak normalleştirilmektedir. Bundan sonra, çıkarılan özellik vektörü çifti bir özellik vektöründe birleştirilmekte ve benzerlik öğrenme bileşeni tarafından işlenmektedir. Bireysel görüntüler oldukça küçüktür. Bu nedenle özellikleri çıkarmak için çok sayıda evrişim katmanına ihtiyaç yoktur. CNN dalları DenseNet'e dayanmaktadır. Genel DenseNet yapısı Şekil 3.1.'de gösterilmektedir. Daha sonra yinelemeli deneyler yoluyla Şekil 3.2.'te gösterildiği gibi özel kompozisyona ulaşılır.



Şekil 3.1. DenseNet yapısı (Huang ve ark., 2016)

Ağda iki giriş vardır. Bu kardeş (siyam) ağların her biri diğeriyle aynıdır. İki alt ağın çıktıları birleştirilmekte ve ardından nihai çıktı benzerlik puanı döndürülmektedir.

```
model.model.summary()
```

Layer (type)	Output Shape	Param #	Connected to
input_2 (InputLayer)	(None, 221, 221, 3)	0	
input_3 (InputLayer)	(None, 221, 221, 3)	0	
sequential_1 (Sequential)	(None, 2048)	20861480	input_2[0][0] input_3[0][0]
concatenate_1 (Concatenate)	(None, 4096)	0	sequential_1[1][0] sequential_1[2][0]
dense_1 (Dense)	(None, 128)	524416	concatenate_1[0][0]
dropout_1 (Dropout)	(None, 128)	0	dense_1[0][0]
dense_2 (Dense)	(None, 32)	4128	dropout_1[0][0]
dropout_2 (Dropout)	(None, 32)	0	dense_2[0][0]
dense_3 (Dense)	(None, 1)	33	dropout_2[0][0]

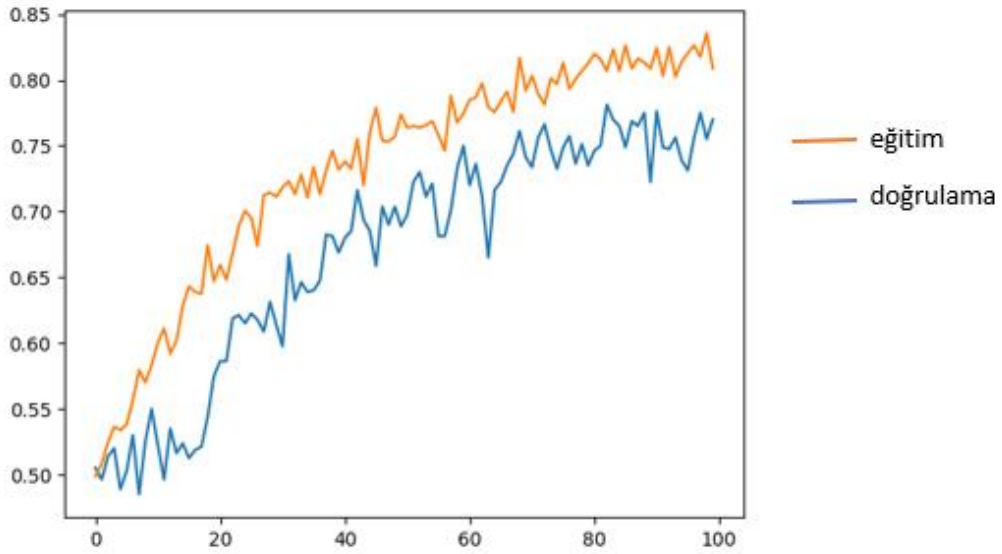
Total params: 21,390,057  
Trainable params: 7,854,953  
Non-trainable params: 13,535,104

Şekil 3.2. DenseNet açıklaması

Siyam ağ için yazılan uygulamamızı kullanarak önerilen ağa iki giriş görüntüsü vermekteyiz. Ağ, bu iki görüntünün aynı sınıfa ait olup olmadığını tahmin edecektir. Ardından, doğrulamayı onaylamak için ağın güven puanı kontrol edilir. Ayrıca, siyam ağları az veri ile eğitilebilir, bu da tek adımda öğrenme ve birkaç adımda öğrenme gibi daha gelişmiş uygulamaları mümkün kılmaktadır.

Siyam ağlarında iki alt ağ aynı mimariye ve aynı parametrelere sahiptir ve birbirlerini yansıtırlar. Bir alt ağdaki ağırlıklar güncellenirse, diğer alt ağlardaki ağırlıklar da güncellenmektedir. Her alt ağın çıktısı, tam bağlantılı (FC) bir katmandır. Tipik olarak bu çıktılar arasındaki Öklid mesafesini hesaplar ve iki girdi görüntüsünün ne kadar benzer olduğunu belirleyebilmek için onları bir Sigmoid aktivasyonu yoluyla besler. Sigmoid aktivasyon fonksiyonu değerleri “1”e yakın daha fazla benzerliği ifade ederken “0”a yakın değerler daha az benzerliği göstermektedir.

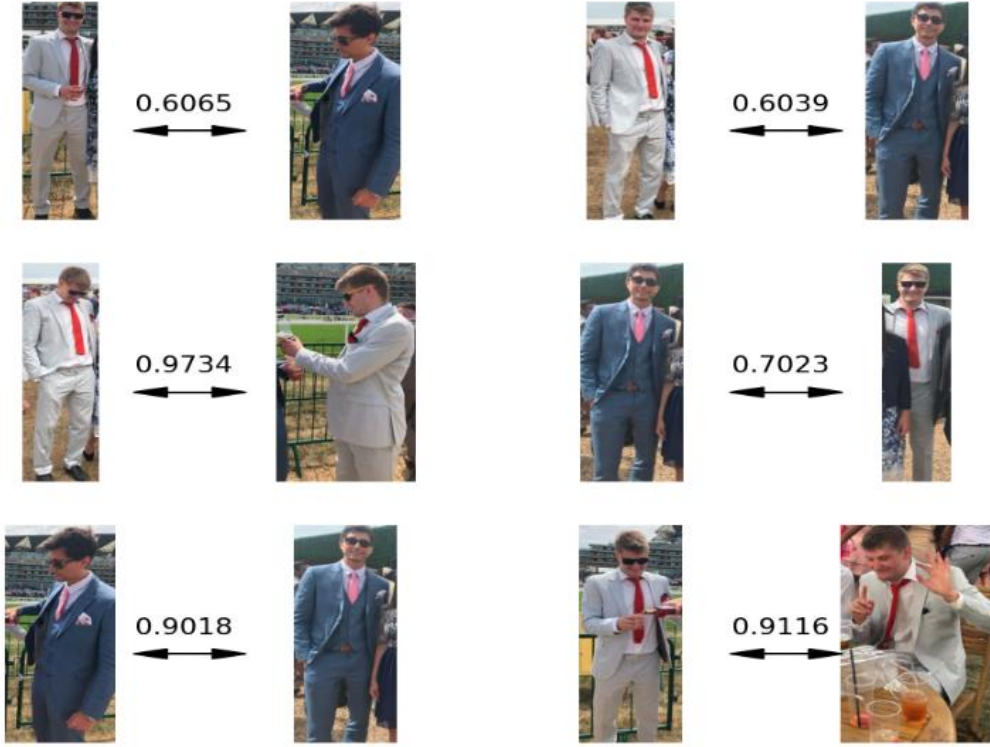
Şekil 3.3.’te, modelimiz doğrulama setimizde %85’e yakın doğruluk elde ediyor, bu da zamanın %85’inde modelin iki giriş görüntüsünün aynı sınıfa ait olup olmadığını doğru bir şekilde belirleyebildiğini göstermektedir.



Şekil 3.3. Siyam Ağların eğitim ve doğrulama grafiği

Önerilen siyam ağını test etmek için küçük bir program yazıldı. Uygulamamız bir çift giriş görüntüsünü kabul etmektedir ve ardından bu iki görüntünün aynı sınıfa ait olup olmadığını belirlemeye çalışmaktadır. Örneğin, modele her biri benzer nesnelere içeren iki görüntü gönderecek olursak, siyam ağı ikisi arasında yüksek benzerlik rapor edecek ve bunların gerçekten de aynı sınıfın parçası olduklarını göstermektedir. Bununla birlikte, biri farklı nesnelere içeren iki görüntü seçersek, iki nesnenin ayrı sınıflara ait

olduğu göz önüne alındığında, ağ düşük benzerlik hesaplamaktadır. Uygulamanın örnek çıktısı Şekil 3.4.'te gösterilmektedir.



Şekil 3.4. Siyam Ağların benzerlik hesaplama örnek çıktısı

### 3.5. Benzerliğe Dayalı Yeniden Tanımlama ve İzleme

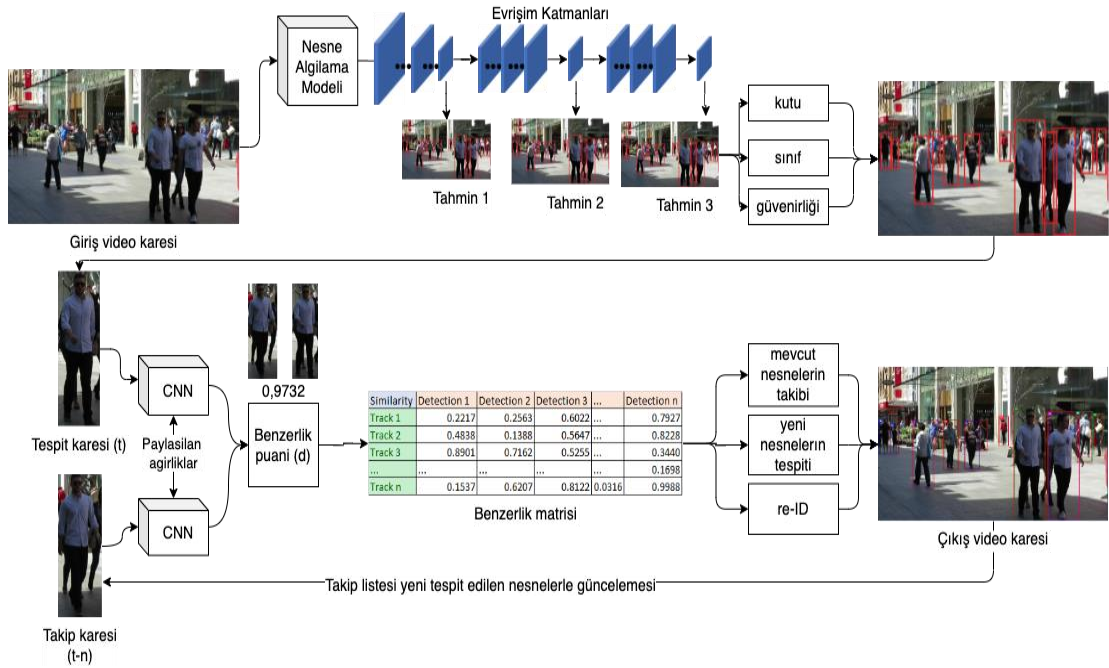
Bu bölümde, Derin Siyam Ağı yöntemini kullanarak MOT için önerilen benzerlik tabanlı kişi yeniden tanımlanması ayrıntılı olarak açıklanmıştır. Çerçevemizde bu bağımlılıkları video karelerinde nasıl birleştirildiği tanıtılmıştır. Nesne detektörleri kavramı içinde tespit ve izleme başlatmanın nasıl gerçekleştirildiğinden başlayarak, nesnenin giriş görüntüleri arasındaki benzerlik hesaplamasını dahil ederek yeniden tanımlamaları detaylandırılmıştır. Önerilen metodumuz Şekil 3.5.'da açıklanmaktadır.

Nesne tespit görevi, bir sınırlayıcı kutuyla bir nesneyi tanımlamaktan ve nesnenin şeklini ve boyutunu değiştirse bile nesneyi sınırlayıcı kutunun içinde tutmaktan oluşmaktadır. Önerilen çerçeve modeli, görüntüdeki bir nesnenin her örneği için sınırlayıcı kutular üretmektedir. Kişi kimliği sorunu için, benzerliklerini öğrenmek için



çıkarılan görüntü özellikleri değerlendirilir. Bu zorluk için pratik bir mimari, iki aynı özellik çıkarıcının aynı ağırlıkları paylaştığı Siyam mimarisidir.

Nesnenin tespit etmesinden sonra, takip yapmak için Siyam CNN'si kullanılır. Siyam CNN yöntemleri, çerçeve çiftleri üzerinde çalışmaktadır. Amaçları, ikinci çerçeveden bir arama alanındaki ilk çerçevedeki hedef nesneyi izlemektir. Takip sistemi, nesnelerin videodaki arandığı  $t$  (mevcut kare) ve  $t-n$  (önceki kare) çiftlerine bakarak çalışır. Aynı nesne örneğini  $t$  karesinde daha önce tespit edildikleri yerin etrafındaki bir alanda arar. Nesne arama alanında tespit edilebilir olduğu sürece, iz yüksek bir  $d$  benzerlik puanı üretmelidir ve  $t$  karesinde bulunan yeni bir sınırlayıcı kutu ile takibine devam edilir. Bu arama alanında nesne bulunmadığında,  $d$  çitisi olarak düşük bir puan vermelidir.



Şekil 3.5. Çevrimiçi çok nesneli izleme çerçevemize genel bakış

Her karedeki tespit,  $t-n$  karesindeki durumlarından bağımsız olarak,  $t$  karesinde görünen tüm nesnelerin tanımlanmasından sorumludur. Yeni bir nesne bulunduğunda, yeni bir takip ID atanır. Her benzerlik kontrolünden sonra, farklı poz değişiklikleri veya oklüzyonlar nedeniyle nesnenin önceki video karesinde kimlik ve görünüm verileriyle güncellenmektedir. Bir nesne video karesinden dışına çıktığı için artık görünmüyorsa, takip süresi dolmuş demektir.

Her tespitten elde edilen nesne görünüm verileri, önceki kareden gelen iz ile benzerlik dizisine kaydedilir. Yeni bir tespiti mevcut izlerle ilişkilendirmek için, bellekteki her bir izden tespitlerin toplanmasına yeni bir ögenin benzerliğini hesaplamak gerekir. İz belleği, tıkanıklıkları ve kaçırılan tespitleri hesaba katmak için birkaç saniye tutulur ve önceki tespitlere yeni bir tespit bağlanırsa, etkin olmayan izler yeniden ilişkilendirebilir. Üç kareden sonra, algılanmayan izler bellekten silinmektedir.

Giriş video kareleri, yeniden tanımlama önerisi sürecini tamamlamak için sınırlayıcı kutu bilgilerinin entegre edildiği nesne tespit sonuçları için kullanılmaktadır. Her karedeki nesnelere sınıflandırmak ve yerleştirmek için YOLOv3 kullanılır. Ardından, nihai nesne izlerini elde etmek için Siyam Sinir Ağı kullanılarak bir veri ilişkilendirme işlemi uygulanmaktadır. Benzerlik dizisi çıktısına göre, mevcut nesnelere izlerine devam edecek, yeni nesnelere tespit edilecek ve yeniden tanımlama tamamlanacaktır. Çıktı olarak, izlenen her nesne, kimliğiyle birlikte bir sınırlayıcı kutuda görüntülenecektir.

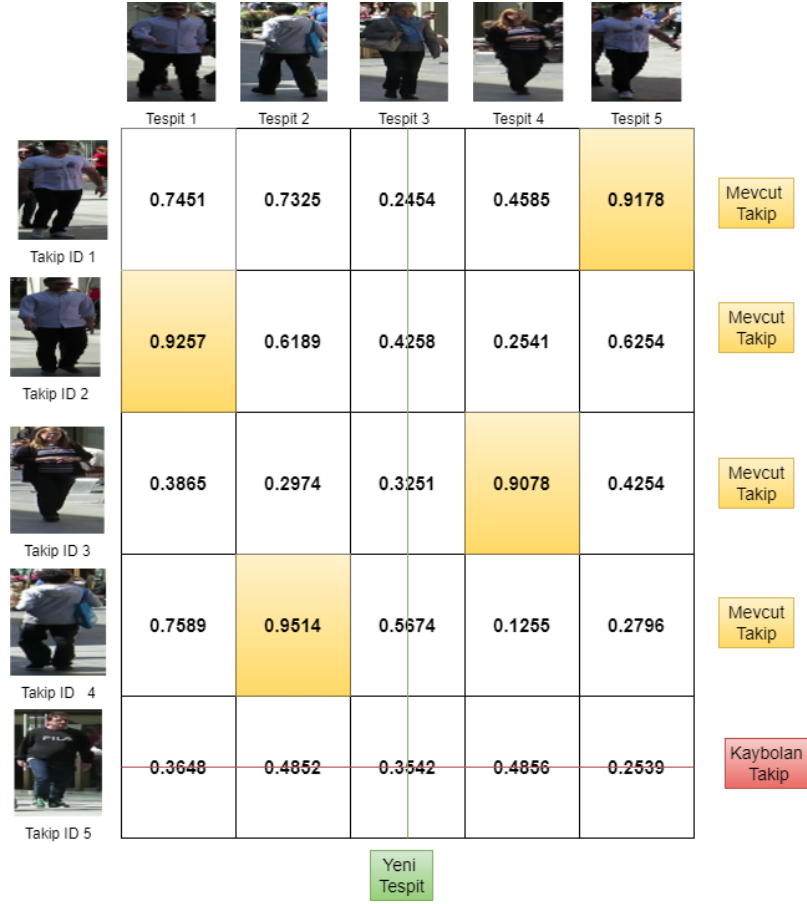
### **3.6. Benzerliğe Dayalı Veri İlişkilendirme**

Derin öğrenme tabanlı çoklu nesne izleme algoritmasının görevi, tespitler arasındaki benzerlik veya mesafe ölçüm modelini optimize etmektir. Siyam Ağının ağ mimarisi, önceki kareden örnek görüntüsünü eşleştirerek hedefin mevcut karesini elde etmektedir. İzleme için gereken ilk eylem, kırpma ve yeniden boyutlandırma operatörü aracılığıyla böyle bir örnek görüntünün oluşturulmasıdır. Bu görüntü, nesnenin algılanabileceği ilk kareden elde edilir ve sınırlayıcı kutuyu içerir. Örnek görüntü elde edildikten sonra, tüm nesnelere benzerlik için listelenip karşılaştırılacağı bir benzerlik dizisi üzerinde gerçekleştirilen izleme ve veri ilişkilendirmesi başlamaktadır. Bu alanın çıkarılması, her kare için bir önceki karenin örnek görüntüsünün alınmasına çok benzer bir şekilde gerçekleştirilmektedir.

Nesnelere ilişkilendirilmesi için, tespit edilen ve izlenen nesnelere benzerliğinin saklanacağı bir benzerlik dizisi önerilmiştir. Tespit yapıldıktan sonra en yüksek

benzerlik puanı kontrol edilerek veri ilişkilendirme ve izleme yapılmaktadır. Önceki kareden tanınan nesne, mevcut karenin bir nesnesi ile ilişkilendirilecek ve izleme aynı ID ile devam edecektir. İzlenen nesne üç kare için kaybolursa, nesne sonlandırılan iz olarak etiketlenecek ve bir diziden temizlenecektir. Benzerlik dizisi hesaplandığında, belirleme mevcut izleri kontrol etmeye dayalı olacaktır. Bundan sonra yeni tespit yeni iz kimliğine atanacaktır. Önerilen benzerlik dizisi, Şekil 3.6.'de gösterilmektedir.

Siyam ağı tabanlı metotlar, hassasiyet ve hız dengeleri nedeniyle büyük ilgi görmekteler (L. ve ark., 2021). Çapraz entropi kaybı işlevi kullanılarak sınıflandırma için uzun vadeli izleme için indirgeme yöntemiyle Siyam ağı önermektedir (S. Li ve ark., 2020). Siyam sinir ağı, eğitim verilerine dayanarak nesnenin özelliklerini öğrenmek için denetimli bir eğitim metodolojisi kullanmakta, ardından bilinmeyen nesnelere tahminler yapmaktadır. Çerçevemizde üçlü kayıp fonksiyonu ile CNN kullanılmaktadır. Sonuç olarak, izlemedeki ilk adım, bir kırpma ve yeniden boyutlandırma operatörü kullanarak böyle bir örnek görüntü oluşturmaktır. Tamamı 221x221 piksellik bir alana ölçeklenen hedef artı bir bağlam marjını içeren sınırlayıcı kutu, ögenin tanımlanabileceği ilk kareden alınmaktadır.



Şekil 3.6. Benzerlik dizisi tabanlı tanımlama ilkesi

Bu örnekte, tespit edilen her nesne ile izlenen nesne arasındaki benzerlik hesaplanır, Takip ID-1, Tespit 5 ile en iyi benzerlik puanına sahiptir ve bu iz, mevcut iz olarak devam edecektir. Aynı mantık, Takip ID-2,3 ve 4 için de kullanılır. Tespit 3'ün eşleşen bir izi yoktur, yeni bir iz olarak işaretlenir ve yeni bir kimlikle atanır. İz, üç kareden fazla herhangi bir tespit ile ilişkili değilse, o iz Takip ID-5 için gösterildiği gibi, kaybolan takip olarak gösterilmektedir.

## **BÖLÜM 4. BULGULAR VE DEĞERLENDİRME**

Bu bölümde önerilen modelin çalıştırılması için kullanılan yazılımlar, kütüphaneler ve sonucu yapılan deneyler ve elde edilen çıktılar açıklanmıştır.

### **4.1. Kullanılan Yazılımlar ve Kütüphaneler**

Bu bölümde proje gerçekleştirilirken kullanılan teknolojilerden, kullanılan kütüphaneler ve programlama dilinden bahsedildi. Projede alınan görüntüleri işlemek, göstermek gibi işlevleri yerine getirmek için kullanılan kütüphane alt başlıklar halinde sırası ile açıklanmıştır.

#### **4.1.1. Programlama dili**

Projede Python dilini kullanmaktaki en büyük neden geliştirilecek olan uygulamanın derin öğrenme işlemini de barındırması ve bu işlemleri gerçekleştirirken kullanılacak kütüphanelerin birçoğunun Python dili ile uyumlu bir şekilde çalışmasıdır. Özellikle Python birçok derin öğrenme kütüphanesinin arka planında kullanılmasından dolayı bu tarz uygulamalarda en çok tercih edilen programlama dili olarak kullanılmaya devam edilmektedir. Bu yüzden projenin bu dil ile oluşturulması daha uygun olmaktadır.

Python, Guido Van Rossum adlı Hollandalı bir programcı tarafından yazılmış bir programlama dilidir. Geliştirilmesine 1990 yılında başlanmıştır. Masaüstü uygulamaları, web uygulamaları, veri analizi ve görselleştirme uygulamaları gibi birçok programı Python ile kolaylıkla geliştirebilmektedir. Bunun yanında Python programlama dili, hemen her tür platformda çalışabilmektedir.

#### 4.1.2. Makine öğrenimi için platform

TensorFlow, büyük ölçekte ve farklı ortamlarda çalışan bir makine öğrenimi sistemidir. TensorFlow, hesaplamayı, paylaşılan durumu ve bu durumu değiştiren işlemleri temsil etmek için veri akışı grafiklerini kullanmaktadır (Abadi ve ark., 2016). Veri akışı grafiğinin düğümlerini bir kümedeki birçok makinede ve bir makinede çok çekirdekli CPU'lar, genel amaçlı GPU'lar ve Tensör İşleme Birimleri (TPU'lar) olarak bilinen özel tasarlanmış ASIC'ler dahil olmak üzere birden çok hesaplama cihazında eşler. Bu mimari, uygulama geliştiricisine esneklik sağlar. Optimizasyonlar ve eğitim algoritmaları ile denemeler yapmasına imkân tanır.

#### 4.1.3. Programlama fonksiyonları kütüphanesi

OpenCV, esas olarak görüntü işleme için kullanılan bir programlama fonksiyonları kütüphanesidir (Mahamkali & Ayyasamy, 2015). Bilgisayar görüntü uygulamaları için standart API sağlar. Görüntü işleme uygulamalarını kullanarak birçok gerçek zamanlı problemi çözebilmektedir.

#### 4.1.4. Yapay sinir ağı için kütüphane

Keras, yapay sinir ağları için bir Python ara yüzü sağlayan açık kaynaklı bir yazılım kütüphanesidir. Kullanıcıyı daha düşük seviyedeki bu kütüphanelerin karmaşıklığından kurtararak, daha basit şekilde yapay sinir ağı modellerinin oluşturulması ve eğitilmesine olanak sağlamaktadır.

#### 4.1.5. CUDA

NVIDIA CUDA Toolkit, yüksek performanslı GPU hızlandırmalı uygulamalar oluşturmak için bir geliştirme ortamı sağlamaktadır. CUDA Toolkit ile GPU hızlandırmalı gömülü sistemler, masaüstü iş istasyonları, kurumsal veri merkezleri, bulut tabanlı platformlar süper bilgisayarlarında uygulamalar geliştirmek ve optimize etmek mümkündür. Hesaplamaları çoklu GPU yapılandırmaları arasında dağıtmak için

yerleşik yetenekleri kullanarak, tek GPU iş istasyonlarından binlerce GPU'lu bulut kurulumlarına kadar ölçeklenen uygulamalar geliştirebilmektedir.

#### 4.1.6. Metrikler

Py-motmetrics kütüphanesi, MOT sonuçları kıyaslamak için bir Python metrik uygulaması sağlamaktadır. Tekli nesne izleyicilerini kıyaslamak oldukça basit olsa da birden çok nesne izleyicinin performansını ölçmek, birden çok eşleşme ortaya çıkabileceğinden dolayı dikkatli bir tasarım gerektirmektedir. Py-motmetrics MOTChallenge kriterleri tarafından gereken metrikler uyumlu hale getirilmektedir.

#### 4.2. Veri Seti

Derin öğrenmenin ortaya çıkmasından bu yana, bilgisayar görüntü algoritmalarının performansını karşılaştırmak çok önemli bir parça olmaktadır. Mevcut ve yeni verileri toplamak ve çoklu nesne izleme yöntemlerinin değerlendirmesi için 2014'ün sonlarında başlatılan tek kameralı MOT için bir kıyaslama verisi olan MOTChallenge veri setleri ortaya çıkmıştır (Dendorfer, Osep ve ark., 2020). Robot navigasyonunda kendi kendini süren arabalara kadar çeşitli uygulamalarla yayalar, izleme topluluğunda açık ara en çok çalışılan nesne olduğundan, kıyaslama birden fazla kişinin takibine odaklanmaktadır. MOT15 adlı veri setinin ilk sürümü, toplam 11.286 kare veya 996 saniyelik video ile eğitim için 11 ve test için 11 diziden oluşmaktadır (Leal-Taixé ve ark., 2015). MOT16 farklı bakış açılarından kaydedilen, kamera hareketi olan/olmayan kalabalık senaryolara sahip, 14 diziden oluşmakta ve çok çeşitli hava ve aydınlatma koşullarını kapsamaktadır. MOT17, MOT16 ile aynı videolardan oluşmakta, ancak iki önemli değişiklik içermektedir: (1) sınırlayıcı kutuların doğruluğunun artırılması, kaçırılan yayaların eklenmesi, ek tıkaçıcıların açıklanması ve (2) çeşitli düzeylerde gürültülü algılamalara karşı sağlamlığı göstermek için üç farklı detektör kullanan izleme yöntemlerinin değerlendirilmesi de dahildir (Milan ve ark., 2016). MOT20 veri kümesi, nesne yoğunluğunun kare başına 246 yayaya ulaşabildiği çok kalabalık sahnelere sahiptir (Dendorfer, Rezatofighi ve ark., 2020). MOTChallenge aynı zamanda çoklu nesne takibi için yöntemlerinin karşılaştırıldığı bir platformdur. Gerçek

verileri, değerlendirme ölçütleri, komut dosyaları ve ayrıca bir dizi önceden hesaplanmış tespit sağlayarak, tüm yöntemler aynı koşullar altında karşılaştırılmakta, böylece izleyicinin performansı diğer faktörlerden izole edilmektedir.

Önerilen algoritmanın performansı, yaygın olarak kullanılan iki MOT kıyaslaması test setleri üzerinde değerlendirilmiştir: MOT16 ve MOT17. MOT16 veri seti, toplamda 7 eğitim seti ve 7 test seti dâhil olmak üzere 14 video dizisi içermektedir. Eğitim seti toplam 110.407 sınırlayıcı kutu ve test setinde 182.326 sınırlayıcı kutudan oluşmaktadır. Bu nedenle, algoritmanın yüksek hızda yüksek bir izleme doğruluğu elde etmesi zordur. Test edilen video dizileri statik veya hareketli bir kamera tarafından çekilmektedir. Bu iki video seti, algoritmanın nesne konumunu tahmin etme gereksinimini artırmaktadır. MOT17'deki videoların senaryoları daha kalabalık ve daha karmaşıktır. Her video dizisi, kameranın farklı yükseklikleri nedeniyle farklı bir bakış açısına sahiptir. Birden çok açıdan videolar nesne tespiti ve özellik çıkarma zorluğu artırmaktadır. Aynı zamanda güneşli bir hava videosu bazı gölgeler içerebilirken, karanlık veya bulutlu havadaki videoların görünürlüğü daha düşük olduğundan nesne tespiti ve izlemeyi zorlaştırmaktadır. Algoritmayı değerlendirmek için her veri setinin gerçek nesne tespit konumları bir dosyada verilmektedir. Dosya formatı, satır başına bir nesne örneği içeren bir CSV metin dosyası olan kesin bilgi dosyasıyla aynı olmalıdır. Şekil 4.1.'de gösterilen veri formatı bu değerleri içermektedir: video karesi, takip edilen nesnenin ID numarası, sınırlayıcı kutunun sol noktası, üst noktası, genişliği, yüksekliği, güven değeri, x, y ve z koordinatları. Dünya koordinatları x, y, z 2D videolar için yok sayılır ve -1 ile doldurulabilir. Benzer şekilde, 3D videolar için sınırlayıcı kutular yok sayılır.

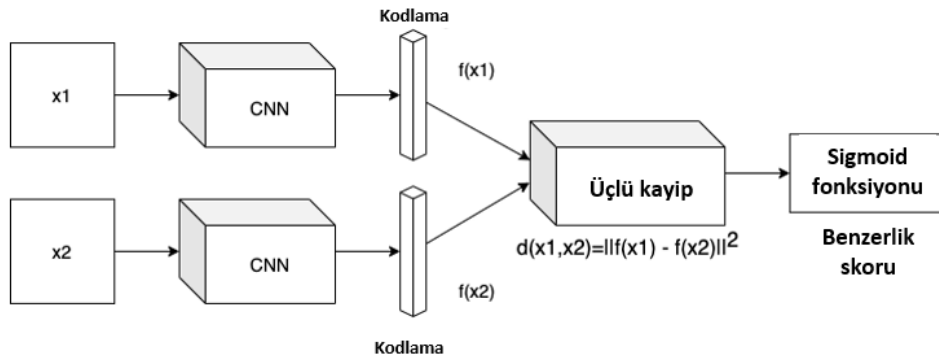
```
1, 3, 794.27, 247.59, 71.245, 174.88, -1, -1, -1, -1
1, 6, 1648.1, 119.61, 66.504, 163.24, -1, -1, -1, -1
1, 8, 875.49, 399.98, 95.303, 233.93, -1, -1, -1, -1
...
```

Şekil 4.1. Motmetrics için veri setlerin CSV format örneği



### 4.3. Deneyler

Ağ eğitimi için pozitif çiftler (aynı sınıfa ait görüntüler) ve negatif çiftler (farklı sınıflara ait görüntüler) kullanıldı. Siyam ağını eğitmek için pozitif ve negatif çift örnekleri rastgele seçilmiştir. Bu çiftler, Siyam ağının benzerliği öğrenebilmesi için eğitim verilerimiz olarak hizmet etmektedir. Temel bir mimari, iki giriş görüntüsünü kabul eder, her bir alt ağ tamamen bağlı bir katmanda biten her bir giriş için aynı alt ağlara sahiptir. Tamamen bağlı katman çıktıları arasındaki Öklid mesafesini hesaplar ve ardından mesafeyi bir sigmoid aktivasyon fonksiyonundan geçirir ve girdiler arasındaki benzerliği belirler. Temel bir Siyam ağ mimarisi uygulaması Şekil 4.2'de gösterilmektedir.



Şekil 4.2. Siyam Ağı kullanarak tek seferde öğrenme gösterimi

Eğitim için, tüm girdi resimleri 221x211 olarak yeniden boyutlandırılır ve evrişim katmanlarına iletilmektedir. Doğruluk ve hız arasında iyi bir denge sağlamak için DenseNet omurga olarak kullanılmıştır. DenseNet, bloklar aracılığıyla katmanlar arasında yoğun bağlantılar kullanan bir tür evrişimli sinir ağıdır. DenseNet'te her katman, önceki tüm katmanlardan ek girdiler alır ve bunlar, birbirleriyle doğrudan eşleşen özellik haritası boyutlarıyla ilgilidir (Huang ve ark., 2016). Bir önceki katmanın çıktısı, ikinci katmanın girişi olarak kullanılmaktadır. Model, öznetelik çıkarıcı olarak DenseNet kullanan oldukça basit bir Siyam Ağıdır.

Çoklu nesne takibinde kişinin yeniden tanımlanması için kullanılan üç farklı veri seti kullanılmıştır. MOT16 veri seti toplam 14 video dizisinden (7 eğitim ve 7 test için) oluşur ve toplam 292.733 algılanan yaya ve 27.966 statik kişi ile statik ve hareketli bir kamera ile kaydedilen videolarından oluşmaktadır (Milan ve ark., 2016). Market-1501 veri seti bir süpermarketin önünde altı farklı kameradan toplanmaktadır ve toplam 32.668 resim üzerinde 1501 farklı kimlik içermektedir (Zheng ve ark., 2015). CUHK03 veri seti, farklı kamera ve açılardan kişilerin kırılmış görüntülerini içermektedir. Veri setinde 1.360 kimlik, manuel olarak kırılan ve otomatik olarak algılanan 13.164 görüntülerden oluşmaktadır (W. Li ve ark., 2014). Tüm deneyler, NVIDIA Quadro K4000 GPU (16 GB VRAM), Intel Core i7-5820K CPU (3.3 GHz) ve 32 GB RAM içeren bir bilgisayarda yapıldı.

#### 4.4. Konfigürasyonlar

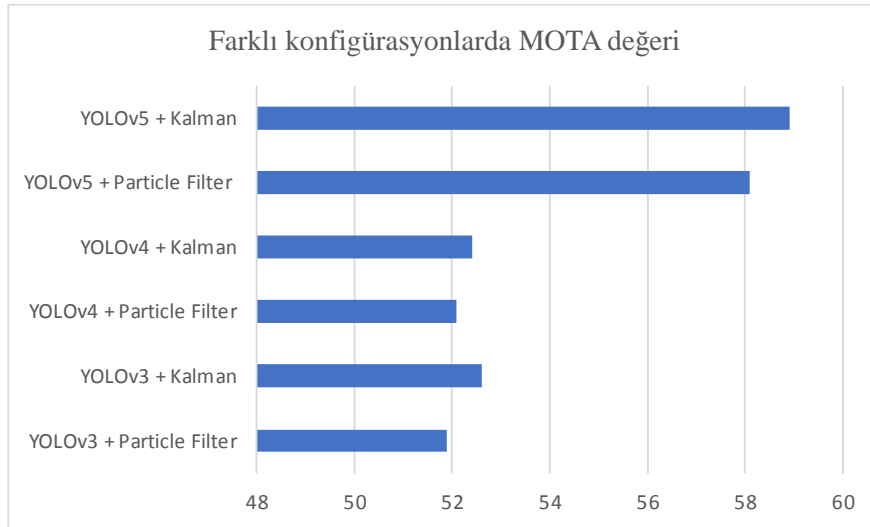
Bu bölümde, önerilen benzerliğe dayalı yöntemimizin farklı konfigürasyonları deneysel olarak gösterilmiştir. Deneyleri için, sağlanan F-RCNN'li MOT17 eğitim setlerini, yarısı eğitim için, diğer yarısı doğrulama seti olarak kullanılan iki parçaya bölündü.

Çalışmamızda yöntemimizin farklı konfigürasyonlarının nicel sonuçlarını raporlanmıştır. Uygulama nicelemesinin izlenmesi için, çok nesneli izleme doğruluğu (MOTA) ve kimlik anahtarı (ID'ler) sayısı aynı anda dikkate alınmaktadır. Tablo 4.1.'den görülebileceği gibi, Partikül filtresi sonucu olan temel yöntem YOLOv3'ün 51.9 MOTA'sı ve 989 ID vardır. Kalman ile sadece Particle filtre modülünü değiştirerek, MOTA'da yüzde 0,7'lik bir iyileştirme ve ID'leri düşürme başarılmıştır. Bu, tahminin kullanılmasıyla sağlanan bir gelişmeyi gösterir. YOLOv3'ü YOLOv4 modülü ile değiştirmek, 52.1 MOTA ve 912'lik ID puanı elde edilmiştir. Sonrasında YOLOv4 detektörü YOLOv5 ile değiştirerek 58,9 MOTA ve 905'lik ID puanı elde edilmiştir. En iyi çözüm olarak Kalman filtresi ile YOLOv5 detektörü kullanmaya karar verilmiştir.

Tablo 4.1. MOT17 veri seti üzerinde çerçevemizin ablasyon çalışması.

Metot	MOTA	FP	FN	IDs
YOLOv3 + Particle Filter	51,9	9749	202987	989
YOLOv3 + Kalman	52,6	9050	202343	961
YOLOv4 + Particle Filter	52,1	10113	229653	912
YOLOv4 + Kalman	52,4	10532	201735	991
YOLOv5 + Particle Filter	58,1	10113	201576	912
YOLOv5 + Kalman	58,9	11125	201343	905

Bu tablo, önerilen metodun performansının birçok yönünü göstermektedir. Farklı detektörlerin ve filtrelerin kombinasyonunu kullanarak belirli yöntemlerle izlemenin genişletilmesinden elde edilen iyileştirmeler kullanılır. Tablo 4.1. MOT17 veri seti üzerinde çerçevemizin ablasyon çalışması. olan MOTA değeri için grafik verisi Şekil 4.3. gösterilmektedir.



Şekil 4.3. SAT'in farklı konfigürasyonlarda MOTA değer grafiği

Çerçevemizde detektör olarak YOLOv5 kullanarak, sınırlayıcı kutu nesne detektörünün doğru algılanan nesnelere daha iyi kapsayabildiğini gözlemekteyiz. Ancak, geçmişten gelen ön bilgileri gerektirdiğinden, daha yüksek FP ve FN'lere yol açmaktadır. Bu uygulama, ID'leri azaltmamızı destekler, ancak MOTA, FP onlar tarafından geliştirilmediği için genel olarak izleme performansı bu harici özelliklerden

gelmemektedir. Ayrıca, önce benzerlik dizisi kullanmanın faydasını göstererek deney çalışması tamamlanmıştır. Bir görünüm CNN içeren önerilen modelimizin verimliliğini göstermek için, modeldeki sınıflandırıcı, bireysel görünüm öznelik vektörlerine dayalı olarak önceki ve mevcut çerçeveden iki tespitin benzerlik puanını tahmin etmektedir. İki farklı filtre türü uygulayarak farklı yapılandırma türlerini kullanarak izleme performansı modelini karşılaştırılmıştır. İkincisi, çerçevemizin bazı niteliksel sonuçlarını göstermektedir. Çözünürlük daha büyük olduğunda veya izleme nesnelerinin sayısı arttığında çalışma süresi hızında da önemli değişiklikler bulunmuştur. Bu, çerçevemizin çeşitli farklı uygulama senaryolarında gerçek zamanlı bilgi işlem kaynaklarını artırabileceğini göstermektedir.

#### 4.5. MOT Yöntemlerle Karşılaştırma

Belirli bir MOT stratejisinin performansını objektif olarak değerlendirmek için metrikler ve veri setleri gereklidir. Optimum sistemi oluşturmak için çeşitli bileşenlerin ve faktörlerin toplam performans üzerindeki etkisini ölçmek gerekir. MOT için değerlendirme metrikleri, çeşitli yaklaşımların nicel karşılaştırmasını değerlendirmek için bir araç sağlar. Bu bölümde, MOT performansını ölçmek için yaygın olarak kullanılan çeşitli değerlendirme ölçütlerini kullanılmıştır.

Önerilen metot, popüler MOT karşılaştırmasında benzer yöntemlerle karşılaştırıldı. Nicel sonuçlar elde etmek için MOT evrim metriklerini kullanılmıştır:

- Çoklu Nesne İzleme Doğruluğu (MOTA), yanlış pozitifler, yanlış negatifler ve uyumsuzluklar dahil olmak üzere sistem tarafından üretilen tüm nesne yapılandırma hatalarını hesaba katar.
- ID F1 Skoru (IDF1), normal yer gerçeği sayısı ve hesaplanan konum üzerinden doğru olarak tanımlanmış tespitlerin oranı.
- Çoğunlukla İzlenen Yörüngeler (MT), bir nesne yaşam döngüsünün en az %80'i boyunca etkin bir şekilde izleniyorsa, çoğunlukla izlenen olarak kabul edilir.

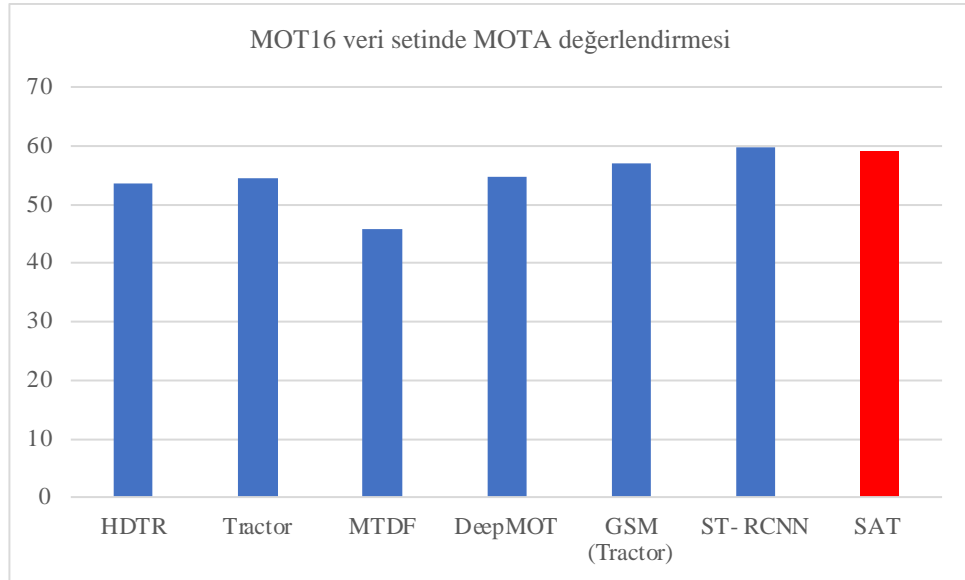
- Çoğunlukla Kayıp Yörüngeler (ML), nesne toplam uzunluğunun yalnızca %20'sinden daha az bir süre boyunca takip edilebiliyorsa, çoğunlukla kayıp olarak kabul edilir.
- Kimlik Anahtarları (IDS), bir parçanın bir yer gerçeği ögesinden diğerine geçiş sayısını sayar
- Yanlış Pozitifler (FP), izleyici tarafından yanlış tanımlanan, ancak temel gerçeklikte mevcut olmayan ögelerin sayısı
- Yanlış Negatifler (FN), tespit edilmeyen temel doğruluk nesnelere ait nesnelere ait sayısı.

Nicel deneyler iki farklı MOT veri seti üzerinde yapılmıştır: MOT16 ve MOT17. Her iki veri seti de hareketli ve statik kameralarla çeşitli açılardan kalabalık kamusal alanları kontrol eden yedi eğitim ve test dizisi içerir. Bu veri setlerinde, yayalar genellikle diğer yayalar veya çevre nesnelere tarafından yoğun şekilde kapatılmaktadır. MOT17, MOT16 veri setine göre tahmin edilen daha kesin bir temel gerçeğe sahiptir. MOT17 ayrıca DPM, Faster-RCNN ve SDP kullanan önceden eğitilmiş üç nesne detektörü ile sonuçları değerlendirilmiştir. En iyi k değerini (izi sonlandıracak değer) bulmak için deneyler de yapılmaktadır. Genel olarak, izleyici biraz daha yüksek bir MOTA elde eder, ancak  $k > 3$  olduğunda hemen hemen aynı IDF1'e ulaşmaktadır. K'nin artmasıyla, iki nesnenin benzerlik puanı daha güvenilir olmakta, fakat bilgisayar hesaplama hızını yavaşlamaktadır.

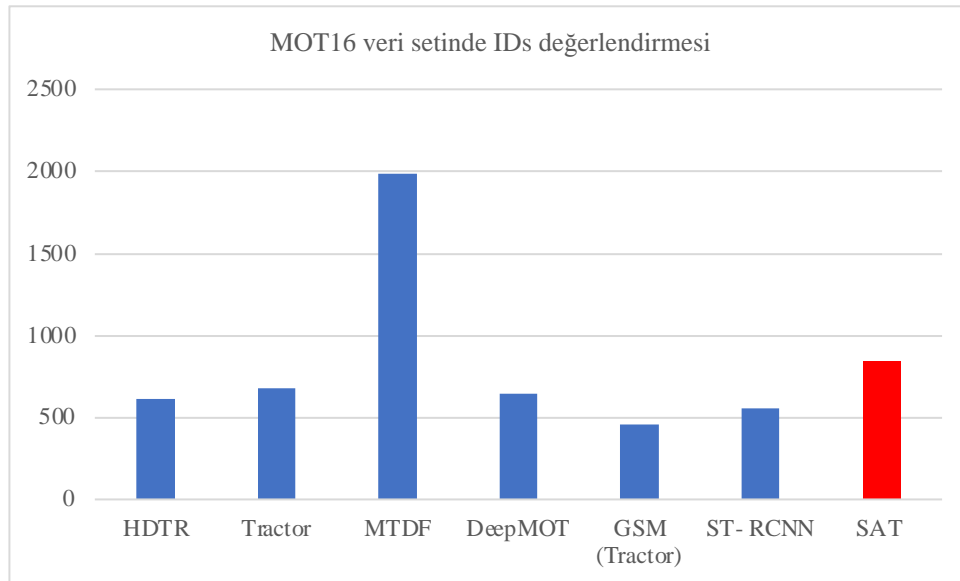
Diğer yöntemlerle karşılaştırıldığında, önerilen metod, benzer izleme kalitesi açısından karşılaştırılabilir performansa ulaşmaktadır. MOT16 sonuçlarından, yöntemimizin iki kişi karşı karşıya geldiğinde re-ID özellikleri ile doğru kimlikleri verebildiğini ve yanlış negatif vaka sayısını azalttığını göstermiştir. MOTA'nın MOT17 veri setindeki sonucu, diğer çerçevelere kıyasla rekabetçi değildir, ancak FP ve FN ölçümlerinin son derece rekabetçi olduğu kanıtlanmaktadır. Tablo 4.2. ve Tablo 4.3.'te gösterilen MOT-Challenge'a sunulan en iyi modelimizin benzer çalışma prensibine sahip olan yöntemlerin sonuçları sunulmuştur. Siamese Track-RCNN, MOTA ve IDF1 metriklerinde sırasıyla %0,7 ve %0,6 oranında performansımızdan daha iyi sonuçlar elde etmektedir (Shuai ve ark., 2020). İzleyicimiz diğer yöntemlerden daha fazla iz oluşturduğu için tüm yöntemler arasında en düşük FN'ye ulaşmaktadır.

Tablo 4.2. SAT'ın MOT16 veri setinde çerçevemiz ile diğer yöntemlerinin karşılaştırılması.

Metot	Sene	MOTA	IDF1	MT	ML	IDs	FP	FN
HDTR	2018	53,6	53,4	21,2	37,0	618	4714	79353
Tractor	2019	54,4	52,5	19,0	36,9	682	3280	79149
MTDF	2019	45,7	40,1	14,1	36,4	1987	120118	84970
DeepMOT	2020	54,8	53,4	19,1	37,0	645	4389	68376
GSM (Tractor)	2020	57,0	58,2	22,0	34,5	457	4332	73573
ST- RCNN	2020	59,8	60,8	22,2	34,5	556	4389	68376
SAT	2021	59,1	60,2	22,5	34,7	842	4032	41253



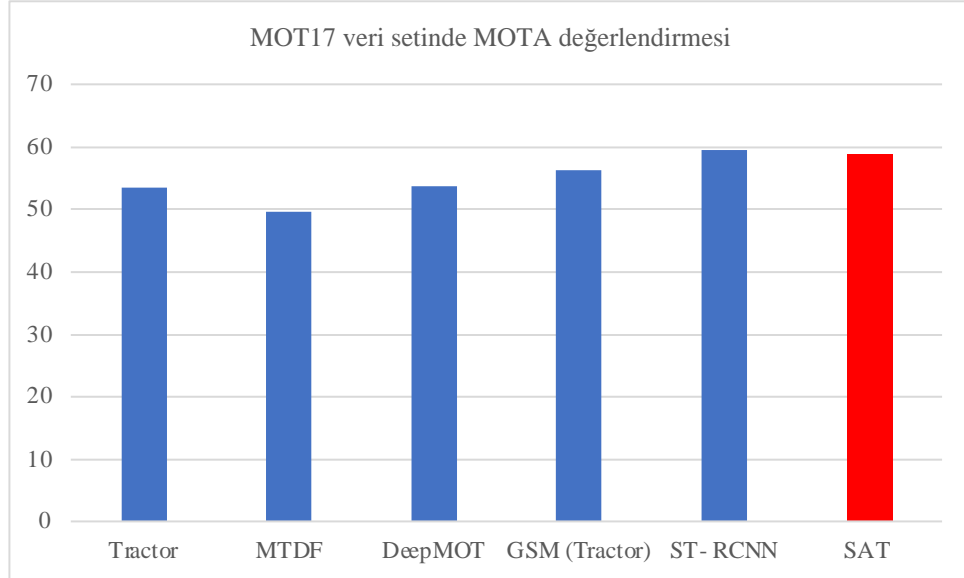
Şekil 4.4. MOT16 veri setinde SAT'ın ile diğer yöntemlerinin MOTA grafiği



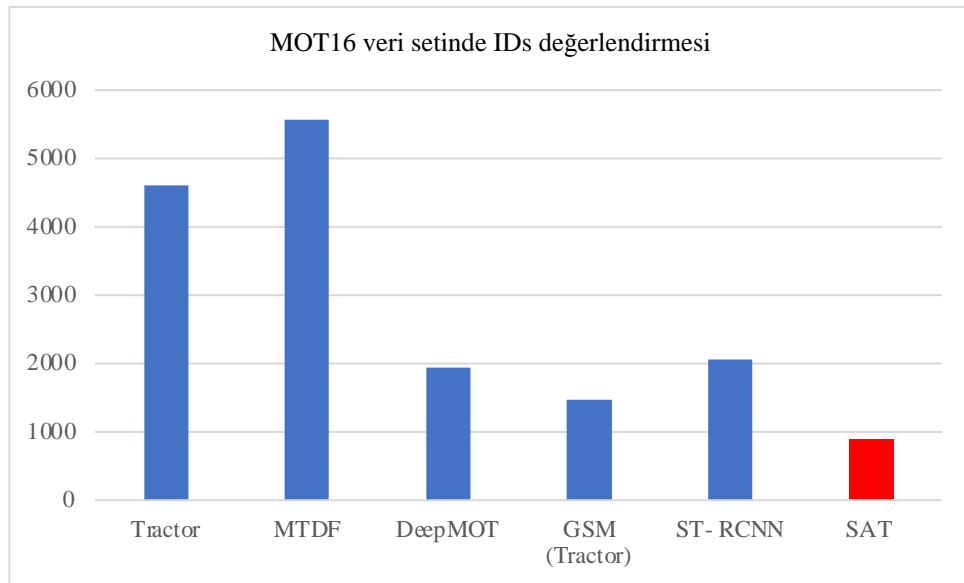
Şekil 4.5. MOT16 veri setinde SAT'ın ile diğer yöntemlerinin IDs grafiği

Tablo 4.3. SAT'ın MOT17 veri setinde çerçevemiz ile diğer yöntemlerinin karşılaştırılması.

Metot	Sene	MOTA	IDF1	MT	ML	IDs	FP	FN
Tractor	2019	53,5	52,3	19,5	36,6	4611	12201	230174
MTDF	2019	49,6	45,2	19,9	33,1	5567	37124	241768
DeepMOT	2020	53,7	53,8	19,4	36,6	1947	11731	247447
GSM (Tractor)	2020	56,4	57,8	22,2	34,5	1485	14379	230174
ST-RCNN	2020	59,6	60,1	23,9	33,9	2068	15532	210519
SAT	2021	58,9	58,1	22,1	33,2	905	11125	201343



Şekil 4.6. MOT17 veri setinde SAT'ın ile diğer yöntemlerinin MOTA grafiği

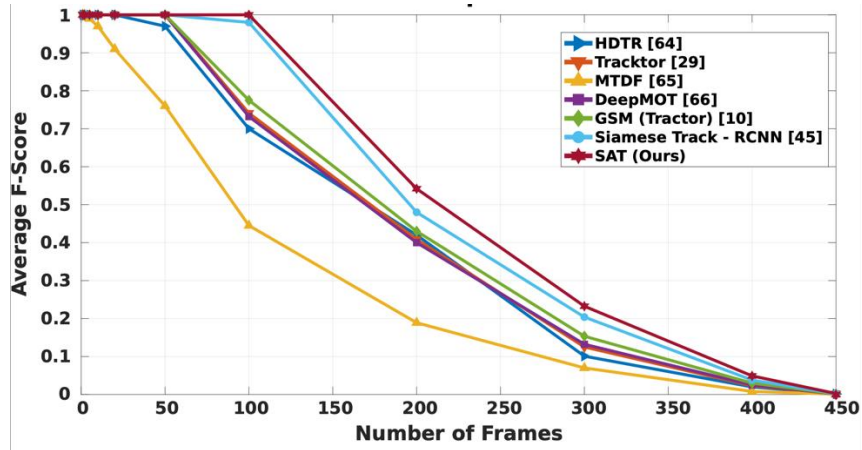


Şekil 4.7. MOT17 veri setinde SAT'ın ile diğer yöntemlerinin IDs grafiği

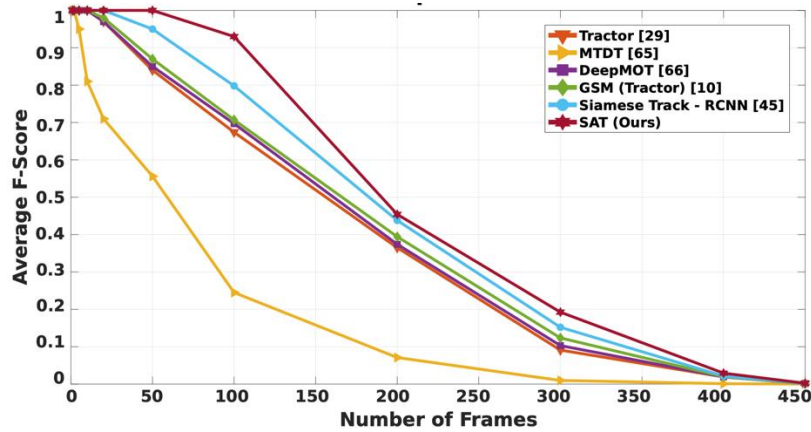
Ayrıca, (Smeulders ve ark., 2013; Valmadre ve ark., 2021)'da bildirilen ilkeleri izleyerek istatistiksel testlerle performans çıktılarını araştırdık. Bu nedenle, MOTA ile ilgili her veri kümesi için ikinci sırada SAT'ın performansı nedeniyle daha zor görüldüğü için test için temel metriğimiz olarak MOTA'yı belirledik. Tüm veri kümeleri için MOTA ve F-skoru arasındaki korelasyon değerleri 0.92 ile 0.97 arasındadır; bu, MOTA ve F-skorunun, algoritmaların tarafından belirtilen aynı özelliklerini değerlendirdiğini belirtmektedir. Test setleri her dizide çeşitli sayıda



örnek görüntü içerdiğinden, hayatta kalma eğrisi sırasında aritmetiği (Kaplan-Meier tahmincisi (Kaplan & Meier, 1958) sonucu) en az sayıda diziyi dikkate alarak ilgili algoritmaların hesaplamasını yapıyoruz. Dikey eksenlerdeki F puanlarının tüm veri kümeleri için ortalama değerleri temsil ettiği MOT16 için Şekil 4.8.'da ve MOT17 için Şekil 4.9.'da gösterildiği gibi görüntülerin (450 kare) sayısı. İlerleme, ortalama F puanlarından yararlanma nedeniyle komşu eğriler tarafından hayatta kalma eğrisi kısıtlamalarını göstermektedir ve ayrıca F puanında kademeli olarak ani düşüşler vardır, bu da çeşitli algoritmalara ait eğriler arasında kesişmelere neden olur. Sonuçlar, SAT algoritmasının yaklaşık 50 kareden sonra diğer tüm yöntemlerden daha iyi performans gösterdiğini ortaya koyuyor ve bu, MOTA ile ilgili olarak biraz daha iyi görünüyor. Bu nedenle, sürekli araştırmaların da bu gerçeği doğruladığı, yaygın olarak kullanılsalar bile performans sonuçlarının değerlendirilmesi oldukça önemlidir (Luiten ve ark., 2021).

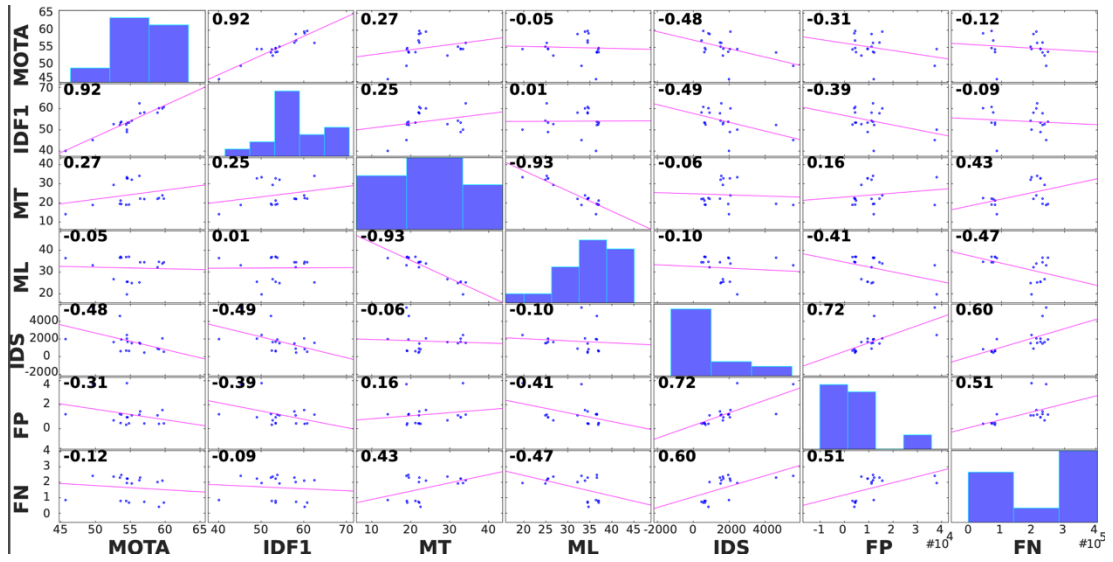


Şekil 4.8. Ortalama F puanlarına göre hayatta kalma eğrileri (MOT16)



Şekil 4.9. Ortalama F puanlarına göre hayatta kalma eğrileri (MOT17)

Ayrıca, birbirleri üzerindeki etkilerini ve etkilerini analiz etmek için Şekil 4.10.'da gösterildiği gibi şimdiye kadar kullanılan değerlendirme metrikleri arasındaki korelasyonu hesapladık. Beklendiği gibi MOTA ve IDF1 yakın bir ilişkiye sahipken, MOTA, IDF1 ve IDS ile neredeyse hiç ilişkisi olmayan MT ve ML, güçlü bir negatif korelasyona sahiptir, yani neredeyse tam tersini yürütürler. Ayrıca IDS'nin hem FP hem de FN ile dikkate değer bir ilişkisi vardır. Özünde sonuçlar, performans ölçütlerinden MOTA, MT veya ML, FP ve FN olmak üzere sadece 4 tanesinin kullanılmasının yeterli olduğunu ortaya koymaktadır.



Şekil 4.10. Performans metrikleri arasındaki ilişki

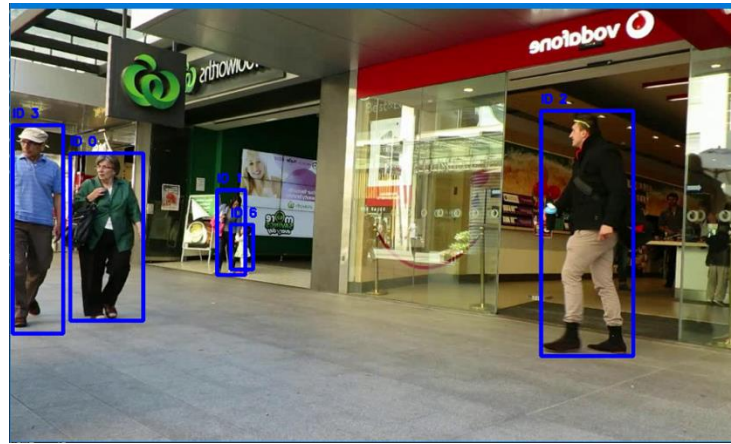
Sol üstteki değerler korelasyon oranını temsil eder (1'e yakın olması daha güçlü bir ilişki anlamına gelirken, 0 hiçbir etkinin olmadığı ve -1'e yakın olması zıt davranış anlamına gelir).

İzleyicimize örnek olarak, izleyicinin tıkanmasının ve performansının kolayca gösterilebildiği MOT16-09 veri setini seçtik. Şekil 4.11. sağ taraftaki nesnelerin algılandığı ve ID-2 ve ID-5 olarak tanımlandığı standart karesidir. Şekil 4.12. karesinde nesne ID-2, nesne ID-5'i kapatır, bu nedenle artık izlenemez. Bu kareden, parçayı ID-5 sonlandırılmış olarak işaretleyene kadar kareleri saymaya başlamaktayız. Birkaç kareden sonra Şekil 4.13. nesne ID-5 detektör tarafından tekrar görünür hale gelir, böylece izini dizimizde tutabiliriz. Oklüzyonlar daha uzun sürerse nesne

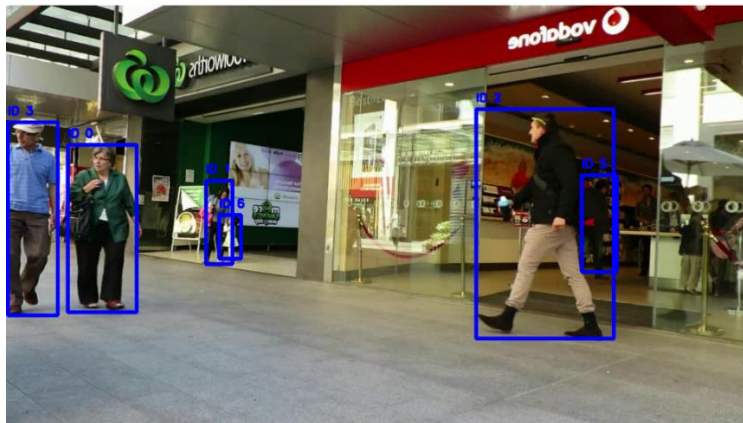
sonlandırılır ve detektörde tekrar görünmesi durumunda yeni bir ID atanır. Bu tür durumlar sahnenin kalabalık olduğu durumlarda oldukça yaygındır, bu nedenle nesnenin konfigürasyonda belirlediğimiz süreden daha uzun süre kaybolduğu durumlarda ID değişecektir.



Şekil 4.11. MOT16-09 veri setinde 32 karesi



Şekil 4.12. MOT16-09 veri setinde 49 karesi



Şekil 4.13. MOT16-09 veri setinde 52 karesi

## **BÖLÜM 5. TARTIŞMA VE SONUÇLAR**

Görüntü tanıma, nesne algılama ve kişinin yeniden tanımlanması alanlarında derin öğrenme yöntemlerinde büyük ilerlemeler göstermekte ve bu da çoklu nesne takibinin geliştirilmesine fayda sağlamaktadır. Bu tezde, genel kıyaslamalarda en üst sıralarda yer alan derin öğrenme tabanlı çok nesneli izleme yöntemlerini özetlemekteyiz. İlk olarak, çoklu nesne takibi için derin öğrenmenin kullanımı düzenlenir ve derin öznitelik aktarımı, sinir ağı yerleştirme ve uçtan uca ağ eğitimi mekanizmaları araştırmaktayız. İkinci olarak, derin ağların izleme çerçevesindeki rollerini araştırıyoruz. Üçüncüsü, bu çok nesneli izleme yöntemleri arasındaki karşılaştırmalar sunmaktayız ve ortak veri kümelerine ve değerlendirmelere göre yorumlamaktayız. Yöntemlerin avantajları ve sınırlamaları vurgulanmıştır.

Deneysel değerlendirmenin analizinden, derin öğrenme paradigmasının izleme sonuçlarını iyileştirmek için çok yer olduğu görülmektedir. Derin öğrenme modellerinin genelleştirilmesini teşvik etmek için olağanüstü olan üretken ağlar tarafından olası bir yol açılabilir. Öte yandan, hareketli platformlar gibi karmaşık ortamlarda reddedilen izleme sonuçlarıyla başa çıkmak için, bu dinamik sahnelerin özelliklerini öğrenmek için bütünleşmiş ağ modellerine ihtiyaç vardır. Ayrıca, değişen koşullara daha fazla uyum sağlamak için, izlenen nesnelere için yüksek mertebeden veya çevrimiçi aktarılan özelliklerin öğrenilmesi beklenmektedir.

Çoklu nesne takibi, yapay görme alanında önemli bir konudur. Bir nesne algılama modeli veya tek nesne izleyici bu sorun için doğrudan kullanılamaz. Sınıf içi varyasyonları olan yüksek yoğunluklu bir sahne gibi MOT ile birçok zorluk ortaya çıkmaktadır. Yine de derin öğrenme tabanlı izleyiciler, yıllar içinde dikkate değer bir performans göstermiştir. Bu tezde, bu alandaki son trendlere, özellikle MOTChallenge'in çevrimiçi izleyicilerine odaklanılmaktadır.

Araştırma, MOT'ta kullanılan teknikleri kapsamaktadır ve bu tür bir sistemin ana yapısını açıklamaktadır. Bir MOT sisteminin, algılama ve izlemenin iki ana görevine odaklanılmaktadır. Tespit için en çok kullanılan tekniğin CNN mimarilerine dayandığını ve izleme için mevcut olan çeşitli tekniklerin etkinliğinin tespit ve tanımlama sonucuna bağlı olduğunu açıkladık. Ayrıca, farklı MOT türleri arasında bir ayırım yapmaktayız.

Videolarda MOT ile bağlantılı yaklaşımlar ve zorluklar bu çalışmada açıklanmıştır. MOT yaklaşımında, videoda her karede birden fazla nesne bulmak ve sürekli video boyunca bu verilerden kimlik bilgilerini çıkarmak amaçlanır. Bu çalışmada ise CNN kullanarak bir görüntü boyunca bir kişiye yaklaşabilen ve onu takip edebilen bir algoritma oluşturmak ve eğitilmiş bir Siyam Ağı kullanan çoklu nesne takibine dayanan bir yaklaşım önerilmiştir. Bu yöntem, belirli bir çerçevedeki nesnelerin benzerliğini analiz ederek gerçekleştirilir. Analiz edilen videolar, sırasıyla 52.6 ve 43.8'lik MOT16 ve MOT17 veri setlerinde ortalama bir MOT'a gösterilmiştir. Önerilen model tıkanmayı ele alır. Nesnelerin benzerliğini hesaplaya dayandığından dolayı dinamik ortamlara dayanıklıdır. Algoritmamızın diğer MOT metotları ve videolarında kabul edilebilir sonuçlar verdiği de kanıtlanmıştır.

Önerilen ağın verimli bir mimarisi için çeşitli konfigürasyon alternatifleri test edilmiştir. Bir deney çalışması olarak bunlardan birkaç ilginç seçenek sunulmuştur. MOT'un son yıllarda önemli ölçüde ilerlemesine rağmen, mevcut MOT araştırmalarında hala çeşitli zorluklar ve araştırılması gereken birçok çözülmemiş sorun vardır.

Önerilen yöntem iyileştirmeler göstermiş olsa da videoda kısa bir süre için aynı sınıftan birçok nesnenin görüldüğü senaryolarda çoklu nesne takibi zor olmaya devam etmektedir. Bu tür senaryolar, halka açık kalabalık sahnelerde yaygındır. GPU'larla daha sağlam bir derin öğrenme algoritması geliştirilerek mevcut sınırlamaların üstesinden gelinebilir.

## KAYNAKLAR

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., ... Zheng, X. (2016). TensorFlow: A system for large-scale machine learning. <http://arxiv.org/abs/1605.08695>
- Bergmann, P., Meinhardt, T., & Leal-Taixé, L. (2019). Tracking without bells and whistles. CoRR, abs/1903.05625. <http://arxiv.org/abs/1903.05625>
- Bewley, A., Ge, Z., Ott, L., Ramos, F., & Upcroft, B. (2016). Simple Online and Realtime Tracking. <https://doi.org/10.1109/ICIP.2016.7533003>
- Braso, G., & Lear-Taixe, L. (2020). Learning a Neural Solver for Multiple Object Tracking. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 6246–6256.
- Cakir, S., & Cetin, A. (2021). Visual object tracking using Fourier domain phase information. Signal, Image and Video Processing.
- Chen, L., Ai, H., Chen, R., & Zhuang, Z. (2019). Aggregate Tracklet Appearance Features for Multi-Object Tracking. IEEE Signal Processing Letters, PP.
- Chu, Q., Ouyang, Q., Li, H., Wang, X., Lui, B., & Yu, N. (2017). Online Multi-object Tracking Using CNN-Based Single Object Tracker with Spatial-Temporal Attention Mechanism. Proceedings of the IEEE International Conference on Computer Vision, 4846–4855.
- Ciaparrone, G., Luque, F., Sanchez, L., & Tabik, S. ve ark. (2020). Deep learning in video multi-object tracking: A survey. Neurocomputing, 381, 61–88.
- Dendorfer, P., Osep, A., Milan, A., Schindler, K., Cremers, D., Reid, I., Roth, S., & Leal-Taixé, L. (2020). MOTChallenge: A Benchmark for Single-camera Multiple Target Tracking. International Journal of Computer Vision, 1–37.
- Dendorfer, P., Rezatofghi, H., Milan, A., Shi, J., Cremers, D., Reid, I., Roth, S., Schindler, K., & Leal-Taixé, L. (2020). MOT20: A benchmark for multi object tracking in crowded scenes. ArXiv:2003.09003[Cs]. <http://arxiv.org/abs/1906.04567>
- Huang, G., Liu, Z., & Weinberger, K. Q. (2016). Densely Connected Convolutional Networks. CoRR, abs/1608.06993.

- Kaplan, E. L., & Meier, P. (1958). Nonparametric estimation from incomplete observations. *Journal of the American Statistical Association*, 53(282), 457–481.
- Kavitha, R., & Nivetha, S. (2021). Pothole and object detection for an autonomous vehicle using YOLO. *Proceedings - 5th International Conference on Intelligent Computing and Control Systems, ICICCS 2021*, 1585–1589. <https://doi.org/10.1109/ICICCS51141.2021.9432186>
- L., Yu., Zhao, Y., & Zheng, X. (2021'a). Towards real-time object tracking with deep Siamese network and layerwise aggregation. *Signal, Image and Video Processing*, 15, 1303–1311. <https://doi.org/10.1007/s11760-021-01861-1>
- L., Yu., Zhao, Y., & Zheng, X. (2021b). Towards real-time object tracking with deep Siamese network and layerwise aggregation. *Signal, Image and Video Processing*, 15, 1303–1311. <https://doi.org/10.1007/s11760-021-01861-1>
- Leal-Taixé, L., Milan, A., Reid, I., Roth, S., & Schindler, K. (2015). MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. *ArXiv:1504.01942 [Cs]*. <http://arxiv.org/abs/1504.01942>
- Li, S., Zhao, Z., Kou, L., Zhou, Z., & Xia, G.-S. (2020'a). Siamese Networks with Distractor-reduction Method for Long-term Visual Object Tracking. *Pattern Recognition*, 112, 107698. <https://doi.org/10.1016/j.patcog.2020.107698>
- Li, S., Zhao, Z., Kou, L., Zhou, Z., & Xia, G.-S. (2020b). Siamese Networks with Distractor-reduction Method for Long-term Visual Object Tracking. *Pattern Recognition*, 112, 107698. <https://doi.org/10.1016/j.patcog.2020.107698>
- Li, W., Zhao, R., Xiao, T., & Wang, X. (2014). DeepReID: Deep Filter Pairing Neural Network for Person Re-identification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 152–159. <https://doi.org/10.1109/CVPR.2014.27>
- Li, Y., Liu, L., Zhu, L., & Zhang, H. (2021). Person re-identification based on multi-scale feature learning. *Knowledge-Based Systems*, 228, 107281.
- Lin, Y., Xie, L., Wu, Y., Yan, C., & Tian, Q. (2020). Unsupervised Person Re-identification via Softened Similarity Learning. *CoRR*, abs/2004.03547. <https://arxiv.org/abs/2004.03547>
- Liu, Q., Chu, Q., Liu, B., & Yu, N. (2020). GSM: Graph Similarity Model for Multi-Object Tracking. *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 530–536.
- Luiten, J., Osep, A., Dendorfer, P., Torr, P., Geiger, A., Leal-Taixé, L., & Leibe, B. (2021). Hota: A higher order metric for evaluating multi-object tracking. *International Journal of Computer Vision*, 129(2), 548–578.
- Mahamkali, N., & Ayyasamy, V. (2015, December). *OpenCV for Computer Vision Applications*.

- Mahendru, M., & Dubey, S. K. (2021). Real Time Object Detection with Audio Feedback using Yolo vs. Yolo\_v3. 2021 11th International Conference on Cloud Computing, Data Science Engineering (Confluence), 734–740. <https://doi.org/10.1109/Confluence51648.2021.9377064>
- Mansouri, N., Ammar, S., & Kessentini, Y. (2021). Re-ranking Person Re-identification using Attributes Learning. *Neural Computing and Applications*, 1–17.
- Meimetus, D., Daramouskas, I., Perikos, I., & Hatzilygeroudis, I. (2021). Real-time multiple object tracking using deep learning methods. *Neural Computing and Applications*, 1–30.
- Meinhardt, T., Kirillov, A., Leal-Taixé, L., & Feichtenhofer, C. (2021). TrackFormer: Multi-Object Tracking with Transformers. *CoRR*, abs/2101.02702.
- Milan, A., Leal-Taixé, L., Reid, I., Roth, S., & Schindler, K. (2016). MOT16: A Benchmark for Multi-Object Tracking. *ArXiv*, abs/1603.00831.
- Nousi, P., Triantafyllidou, D., Tefas, A., & Pitas, I. (2020). Re-identification framework for long term visual object tracking based on object detection and classification. *Signal Processing: Image Communication*, 88, 115969.
- Peng, J., Wang, T., Lin, W., Wang, J., See, J., & Wen, S. (2020). TPM: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, 107, 107480.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. <https://pjreddie.com/yolo/>.
- Sadeghian, A., Alahi, A., & Saverse, S. (2017). Tracking the Untrackable: Learning to Track Multiple Cues with Long-Term Dependencies. *Proceedings of the IEEE International Conference on Computer Vision*, 300–311.
- Sharma, V., Gupta, M., Kumar, A., & Mishra, D. (2021). Video Processing Using Deep Learning Techniques: A Systematic Literature Review. In *IEEE Access* (Vol. 9, pp. 139489–139507). Institute of Electrical and Electronics Engineers Inc. <https://doi.org/10.1109/ACCESS.2021.3118541>
- Shuai, B., Berneshawi, A. G., Modolo, D., & Tighe, J. (2020). Multi-Object Tracking with Siamese Track-RCNN. *CoRR*, abs/2004.07786. <https://arxiv.org/abs/2004.07786>
- Smeulders, A. W. M., Chu, D. M., Cucchiara, R., Calderara, S., Dehghan, A., & Shah, M. (2013). Visual tracking: An experimental survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7), 1442–1468.
- Valmadre, J., Bewley, A., Huang, J., Sun, C., Sminchisescu, C., & Schmid, C. (2021). Local metrics for multi-object tracking. *ArXiv Preprint ArXiv:2104.02631*.



- Wang, Y., Weng, X., & Kitani, K. (2020). Joint Detection and Multi-Object Tracking with Graph Neural Networks. *CoRR*, abs/2006.13164. <https://arxiv.org/abs/2006.13164>
- Wojke, N., Bewley, A., & Paulus, D. (2018). Simple online and realtime tracking with a deep association metric. *Proceedings - International Conference on Image Processing, ICIP*, 3645–3649.
- Wu, Q., Dai, P., Chen, P., & ve ark. (2021). Deep adversarial data augmentation with attribute guided for person re-identification. *Signal, Image and Video Processing*, 15, 655–662. <https://doi.org/10.1007/s11760-019-01523-3>
- Wu, Y., Hu, Y., Wang, K., Li, H., Nie, L., & Cheng, H. (2019). Instance-aware representation learning and association for online multi-person tracking. *Pattern Recognit.*, 94, 25–34.
- Wu, Y., Lim, J., & Yang, M.-H. (2013). Online object tracking: A benchmark. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2411–2418.
- Xu, Y., Cao, Y., & Z., Zhang. (2019). Spatial-temporal relation networks for multi-object tracking. *Proceedings of the IEEE International Conference on Computer Vision*, 3987–3997.
- Xu, Y., Osep, A., Ban, Y., & Horaud, R. (2020). How to train your deep multi-object tracker. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 6786–6795.
- Yanagisawa, H., Yamashita, T., & Watanabe, H. (2018). A study on object detection method from manga images using CNN. *2018 International Workshop on Advanced Image Technology (IWAIT)*, 1–4. <https://doi.org/10.1109/IWAIT.2018.8369633>
- Yang, F., Chang, X., Sakti, S., Wu, Y., & Nakamura, S. (2021). ReMOT: A model-agnostic refinement for multiple object tracking. *Image and Vision Computing*, 106, 104091.
- Yang, K., Song, H., Zhang, K., & Liu, Q. (2020). Hierarchical attentive Siamese network for real-time visual tracking. *Neural Computing and Applications*, 32(18), 14335–14346.
- Yang, M., Wu, Y., & Jia, Y. (2017). A Hybrid Data Association Framework for Robust Online Multi-Object Tracking. *IEEE Transactions on Image Processing*, 26, 5667–5679.
- Yoon, Y., Kim, D., Song, Y., Yoon, K., & Jeon, M. (2021). Online multiple pedestrians tracking using deep temporal appearance matching association. *Information Sciences*, 561, 326–351.

- Yu, T., Li, D., Yang, Y., Timothy, H., & Xiang, T. (2019). Robust Person Re-identification by Modelling Feature Uncertainty. Proceedings of the IEEE International Conference on Computer Vision, 552–561.
- Zhang, J., Zhou, S., Chang, X., Wan, F., Wang, J., Wu, Y., & Huang, D. (2020). Multiple Object Tracking by Flowing and Fusing. CoRR, abs/2001.11180. <https://arxiv.org/abs/2001.11180>
- Zhang, Y., Sheng, Y., Wu, Y., Wang, W., Ke, W., & Xiong, Z. (2020). Multiplex Labeling Graph for Near-Online Tracking in Crowded Scenes. IEEE Internet Things J., 7, 7892–7902.
- Zheng, L., Shen, L., Tian, L., Wang, S., Wang, J., & Tian, Q. (2015). Scalable Person Re-identification: A Benchmark. Computer Vision, IEEE International Conference On.

## ÖZGEÇMİŞ

**Adı Soyadı** : Harun SULJAGIC

### ÖĞRENİM DURUMU

Derece	Eğitim Birimi	Mezuniyet Yılı
Yüksek Lisans	Sakarya Üniversitesi / Fen Bilimleri Enstitüsü / Bilişim Sistemleri Mühendisliği	2022
Lisans	Sakarya Üniversitesi / Bilgisayar ve Bilişim Bilimleri Fakültesi / Bilişim Sistemleri Mühendisliği	2019
Lise	Saraybosna Lisesi	2014

### İŞ DENEYİMİ

Yıl	Yer	Görev
2021-Halen	Klika (Bosna Hersek)	Yazılım Mühendisi
2020-2021	Kromel Makina A.Ş	SAP Proje Mühendisi

### YABANCI DİL

Türkçe, İngilizce

### ESERLER (makale, bildiri, proje vb.)

- Düşük Maliyetli Bir Robot Kolunun Dijital İkizinin Elde Edilmesi- 6. Uluslararası Öğrenci Sempozyumu Bildiriler Kitabı, Mühendislik Bilimleri, 2021
- Similarity Based Person Re-Identification for Multi-Object Tracking Using Deep Siamese Network, Neural Computing and Applications, 2022