İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ELEKTRİK-ELEKTRONİK FAKÜLTESİ

DERİN ÖĞRENME İLE UZUN-SÜRELİ İNSAN İZLEME

BİTİRME ÇALIŞMASI

Selahaddin HONİ

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik ve Haberleşme Mühendisliği Bölümü

OCAK 2022

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ ELEKTRİK-ELEKTRONİK FAKÜLTESİ

DERİN ÖĞRENME İLE UZUN-SÜRELİ İNSAN İZLEME

BİTİRME ÇALIŞMASI

Selahaddin HONİ
(040160046)

Elektronik ve Haberleşme Mühendisliği Anabilim Dalı

Elektronik ve Haberleşme Mühendisliği Bölümü

Tez Danışmanı: Prof. Dr. Bilge GÜNSEL

OCAK 2022

İTÜ, Elektrik-Elektronik Fakültesi'nün 040160046 numaralı Lisans öğrencisi Selahaddin HONİ, ilgili yönetmeliklerin belirlediği gerekli tüm şartları yerine getirdikten sonra hazırladığı "DERİN ÖĞRENME İLE UZUN-SÜRELİ İNSAN İZLEME" başlıklı tezini aşağıdaki imzaları olan jüri önünde başarı ile sunmuştur.

Tez Danışmanı :	**Prof. Dr. Bilge GÜNSEL** İstanbul Teknik Üniversitesi	.....
Jüri Üyeleri :		.....
		.....

Teslim Tarihi : **24 Ocak 2022**
Savunma Tarihi : **1 Şubat 2022**

Aileme,

ÖNSÖZ

Öncelikle lisans öğrenim hayatımı anlamlandıran, başta bu bitirme çalışması olmak üzere her konuda destek olup yol gösteren değerli danışmanım Prof. Dr. Bilge Günsel'e teşekkürlerimi sunarım. Yardımları ve çalışmaya olan katkılarından dolayı Dr. Filiz Gürkan'a ve daima arkamda olan aileme teşekkürü borç bilirim.

Ayrıca, yürütücülüğünü danışman hocamın üstlendiği İTÜ Çoğulortam İşaret İşleme ve Örüntü Tanıma laboratuvarına sağladığı imkanlardan ötürü teşekkür ederim.

Ocak 2022 Selahaddin HONİ

İÇİNDEKİLER

Sayfa

ÇİZELGE LİSTESİ

Sayfa

ŞEKİL LİSTESİ

Sayfa

DERİN ÖĞRENME İLE UZUN-SÜRELİ İNSAN İZLEME

ÖZET

Nesne izleme, başlangıçta belirlenen hedef nesnenin konumunun video boyunca kestirimini hedefler. Sezim-ile-izleme nesne izlemede sıklıkla kullanılan bir yaklaşımdır. Bu yaklaşıma göre öncelikle izleyicide bulunan nesne sezici olası nesne konumlarını belirler; ardından, veri ilişkilendirme ile gerçek hedef konumu kestirilir. Derin öğrenme kullanan nesne izleyicilerde nesne sezme ve veri ilişkilendirme konvolüsyonel nöral ağlar ile çıkarılan öznitelik haritaları kullanılarak gerçeklenir. Ancak, nesne sezici ağlar genel amaçlı nesne sınıflandırıcı mimarisine sahip olduklarından çıkarılan özniteliklerin izlenen hedefi ayırt ediciliği yetersizdir. Bu nedenle literatürde, video boyunca hedef nesnenin görünüm değişikliklerini doğrulukla modelleyebilen, 'yeniden-tanılayıcı' (ReID) olarak adlandırılan ek bir öznitelik çıkarıcı ağ kullanımı yaygındır.

Bitirme çalışması kapsamında, videoda uzun-süreli insan izleme amacıyla, derin öğrenmeye dayalı bir nesne izleyici önerilmiştir. Veri ilişkilendirme için 'yeniden-tanılayıcı' (ReID) özniteliklerinin kullanıldığı, sezim-ile-izleme konsepti altında yeni bir çıkarım ağ mimarisi gerçeklenmiştir. Önerilen izleyicide, izleyici girişine gelen her video çerçevesi öncelikle nesne seziciden geçirilerek olası hedef nesne konumları belirlenir. Her bir aday konum içerisindeki görüntü parçasının ve verilen hedef görüntüsünün ReID öznitelikleri çıkarıldıktan sonra kosinüs benzerlik kriterine göre eşleme yapılarak hedefin son konumu kestirilir. Bu izleyici tasarımı, veri ilişkilendirmede hedefe özgün ayırt edilebilir ReID özniteliklerini kullanma önerisiyle, olası yanlış eşleştirme sorunlarının önüne geçmeyi sağlamış ve izleme performansını artırmıştır.

Bitirme çalışmasında önerilen izleyici, Mask R-CNN derin nesne sezici ve sezicinin önerdiği aday konumlarla beslenen ResNet-50 tabanlı Piramit ReID öznitelik çıkarıcısıyla bütünleşmiş bir karar verici içermektedir. Mask R-CNN iki aşamalı bir nesne sezici olup öncelikle girdi olarak verilen bir video çerçevesini barındırdığı InceptionV2 omurga mimarisinden geçirerek öznitelik haritası çıkarır. Hesaplanan öznitelik haritası Bölge Önerim Ağına (RPN) gönderilir ve olası nesne konumları, sınıftan bağımsız olarak olabilirlik skorlarıyla birlikte belirlenir. InceptionV2 çıkışındaki öznitelik haritasından önerilen olası nesne konumlarına karşı düşen öznitelikler sezici üst katmanında RoI hizalama işleminden geçirilir. Ardından Mask R-CNN tepe katmanında bulunan sınıflandırıcı ve regresörden geçirilerek sezilen nesneleri çevreleyen kutular, nesne sınıfları ve sınıflılık skorları hesaplanır. ReID ağında sezilen her bir aday nesneye ayrıntılı bir tanılama ve doğrulama işlemi

uygulanır. Bu amaçla, aday nesne konumları ResNet-50 omurgasından geçirilir ve hedef insanın tanınmasına imkân verecek öznitelik haritaları elde edilir. Adaylar ile hedef nesne arasında doğru bir benzerlik eşleme yapılabilmesi amacıyla, ReID öznitelik haritası piramitsel bir hiyerarşide farklı ölçeklerde çıkarılır. Tümleştirilmiş karar kuralı kapsamında, hedefin aynı ağ ile çıkarılmış ReID öznitelikleriyle aday öznitelikler kosinüs benzerliği kullanılarak eşlenir ve en yüksek benzerlikli konum izlenen insanın son konumu olarak kestirilir. İlerleyen video çerçevelerinde de benzer döngü tekrarlanmaktadır.

Önerilen nesne izleyici mimarisi, izleme amaçlı yeniden eğitim gerektirmediğinden, farklı nesne seziciler ve ReID ağlarının kullanımına olanak tanır. Kullanılan nesne sezicilerin ve ReID ağlarının genel amaçlı eğitilmiş olması yeterlidir. Bitirme çalışmasında COCO veri tabanında 80 farklı nesneyi sezmek için eğitilmiş Mask R-CNN derin nesne sezicisi kullanılmıştır. ReID ağı ResNet-50 omurga ağıdır, 'karşıt-entropi' ve 'triplet' kayıp fonksiyonları birlikte kullanılarak dinamik eğitim ile insan tanılama amacıyla eğitilmiştir. İzleyicinin eğitilmiş ReID modelinin basit bir transfer öğrenme ile farklı nesneleri izleyecek şekilde özelleştirilmesi olanaklıdır.

Bitirme çalışması kapsamında geliştirilen izleyici yazılımı Tensorflow ve PyTorch derin ağ kütüphaneleri yardımıyla eğitilmiş modellerin OpenCV-Python ortamına entegresiyle geliştirilmiştir. İzleyici başarımının raporlanmasında kamera açısı, poz, ölçek, bulanıklık ve ışıklılık değişimleri zorluklarıyla güncel çalışmalarda çokça tercih edilen VOT-LT (Visual Object Tracking - Long Term) ve LASOT (Large-scale Single Object Tracking) veri setleri kullanılmıştır. Uzun süreli insan izleme performansı, hedefin video çerçevesindeki gerçek-referans konumuyla en az 0.5 örtüşme oranı ölçüsüyle, VOT-LT ve LASOT veri tabanlarında sırasıyla, %79 ve %61 olarak raporlanmıştır. Ayrıca, önerilen izleyicinin güncel nesne izleyiciler ile karşılaştırması, VOT-LT veri setinin seçili videolarında raporlanmış ve literatürdeki diğer yöntemlerle rekabet edebilecek seviyede olduğu gösterilmiştir.

LONG-TERM PERSON TRACKING VIA DEEP LEARNING

SUMMARY

Object tracking aims to estimate the location of the target object through video sequence where the target is initialized at the first frame. Tracking-by-detection is a commonly used approach for object-tracking. It refers to detect all candidate target locations via a detector-network; where the final tracked object location is determined with data association. Deep learning based object trackers perform object detection and data association using feature maps extracted by convolutional neural networks. However, since the object detectors are trained for several object classes, the feature maps extracted by them are not as discriminative as desired for a specific object class. In order to alleviate this problem, it is common to use an additional feature extraction network called 're-identification' (ReID) that enables to accurately model the appearance changes of the target object throughout the video.

In this graduation project, a novel inference architecture that employs ReID features for data association is proposed for long-term person tracking. The developed tracker receives each video frame and feeds it into the internal object detector that outputs candidate target locations. For each image patch corresponding to a candidate location as well as the initial target patch, the ReID network extracts the discriminative feature vectors. Similarity matching between the target and candidate ReID features is achieved by cosine similarity metric where the candidate having maximum similarity is estimated as the tracked target object. It is demonstrated that the proposed tracker significantly improves the tracking accuracy.

The designed tracker architecture does not require a re-training for tracking purpose; thus, it allows the use of any object detectors and ReID networks. The developed inference architecture includes Mask R-CNN object detector trained on COCO dataset to recognize 80 different objects, and a decision maker integrated with Pyramidal ReID to extract the features of the candidate target locations. Pyramidal ReID basically is a ResNet-50 backbone dynamically trained on Market-1501 dataset to distinguish the person objects where dynamic training mentions to use both cross-entropy and triplet loss functions together for optimization. However, the ReID feature extractor of the tracker can adapt to other objects by a simple transfer-learning, if it is desired.

In the context of the graduation project, the tracker software is developed by integrating the deep network models, trained on Tensorflow and PyTorch libraries, into the OpenCV-Python environment. Tracking perforamance has been evaluated on VOT-LT (Visual Object Tracking - Long Term) and LASOT (Large-scale Single Object Tracking) datasets, which are two commonly used challenging benchmarkig data sets.

Long-term person tracking performance is reported on the video sequences, having camera angle, exposure, scale, blur and luminance changes, as 79% and 61% in the VOT-LT and LASOT databases, respectively, with at least 0.5 intersection over union (IoU). Moreover, it is demonstrated that the proposed tracker provides comparable performance compared to the state-of-the-art trackers.

1. GİRİŞ

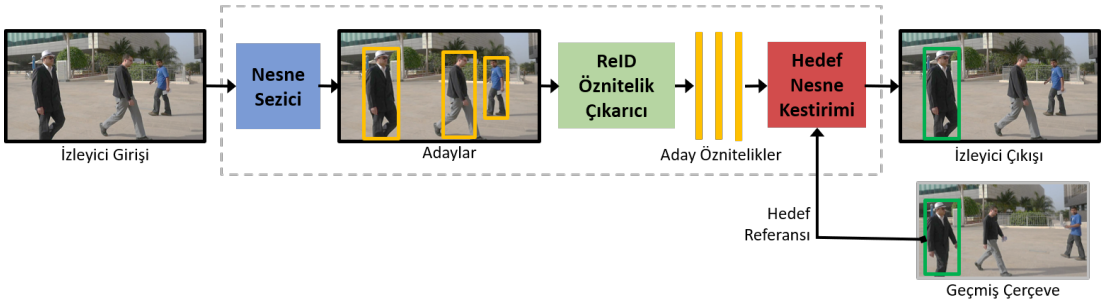
Nesne izleme, otonom araçların kontrolü, güvenlik gözetleme sistemleri, sinema-eğlence sektörü içerik üretimi gibi birçok alanda gerek duyulan önemli bir video analizi adımıdır. Nesne izlemeyi zorlaştıran problemlerin başında hedefin fiziksel görünümünün harekete ve ortam koşullarına bağlı değişmesi gelmektedir. Kamera açısı, poz, ölçek değişimleri, bulanıklık ve ışıklılık varyantları farklı görünümlere sebep olduğu gibi diğer nesneler hedefin görüntüsünün bir kısmını veya bütününü kapatabilir. Bir diğer zorluk, hedefin izleme sahnesinden tamamen ayrılıp bir süre sonra yeniden giriş yapmasıdır. Başarılı bir nesne izleyicinin bu çıkışları sezip girişlerde hedefe yeniden adapte olması beklenmektedir ki güncel izleyici tasarımları çoğunlukla bu sorunun üzerine yoğunlaşmaktadır.

Derin öğrenme tabanlı ağların karmaşık problemlerin çözümünde klasik makine öğrenmesi yaklaşımlarına göre daha iyi performans sergilediği bilinmektedir. Son dönemde katlanarak artan büyük verinin eğitime katılmasıyla, günümüz derin ağları nesne sezimi başarımında insan hatasının altına inebilmiştir. Gradyan temelli özniteliklerin sezme başarısı göz önünde bulundurulduğunda, bu bitirme çalışmasında önerilen izleyici dahil, birçok izleyici mimarisinin derin nesne sezicileri baz alan sezim-ile-izleme (tracking-by-detection) konseptini benimsediğini söylemek mümkündür. Bu konsepte göre izleyicide öncelikle hedefin olası konumları sezilir ardından veri ilişkilendirme ile gerçek hedef konumu kestirilir. Literatürde bu amaçla geliştirilen nesne izleyicilerde bir diğer önemli yaklaşım farkı, tüm izleyici alt sistemlerinin izleme amacıyla uçtan-uca yeniden eğitimi ya da kendi amacı için eğitilmiş alt sistemlerin yeniden eğitim gerektirmeyen bir çıkarım ağ mimarisi ile kullanılarak tümleştirilmesidir. İzleme amaçlı uçtan-uca yeniden eğitim bazı özel amaçlı gerçeklemelerde, amaca uygun eğitim verisinin toplanabilmesi koşulu altında, yüksek izleme başarımı sunabilmektedir. Ancak, genel amaçlı nesne sezme ve nesne yeniden-tanılama mimarilerinin çıkarım aşamasında tasarlanan mimaride, yeniden eğitim yapılmaksızın, kullanımı çok daha pratik çözümler sunabilmektedir. Bu

bitirme çalışmasında sezim-ile-izleme konsepti altında yeni bir çıkarım ağ mimarisi ile izleyici gerçekleme yaklaşımı benimsenmiştir. Nesne sezme amacıyla Mask R-CNN [4] derin nesne sezici kullanılmıştır, nesne yeniden-tanılayıcı olarak ResNet-50 [5] tabanlı piramit mimari [3] ile bütünleşmiş bir karar vericiyle izleme gerçeklenmiştir. Literatürde, sezicilerin ara ve çıkış katmanlarını nesne izleme için özelleştiren TDIOT [6], basitçe Siam ağların (Siamese) karşılaştırıcı yeteneğinden faydalanarak hedefin benzer özniteliklerine sahip konumunu bir sonraki çerçevede arayan SiamRPN++ [7] ve LTMU [8] gibi çalışmalar bulunmaktadır.

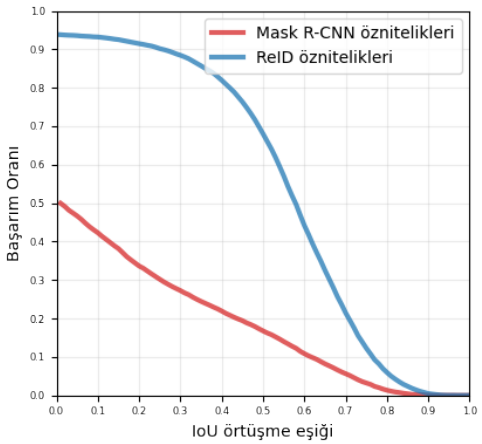
Nesne sezicilerin gelişimi, her dönemin yüksek performans çizgisini belirleyen bölge tabanlı konvolüsyonel sinir ağları (R-CNN) ailesinin genişlemesiyle gerçekleşmiştir. Her yeni nesil, ata mimarilerinden kalıtım almış ve çeşitli güncellemelerle kendini geliştirmiştir. R-CNN [9], ilk olarak Seçici Arama [10] (Selective Search) algoritması ile nesne konumları için olası sınırlayıcı kutu adayları belirlemiş; ardından, her bir adayın öznitelik haritasını birbirinden bağımsız çıkarmıştır. Hesaplanan bu öznitelik haritaları, arkaplan ve diğer tüm nesne sınıfları için ayrı eğitilmiş ikili SVM ile sınıflandırılmıştır. Seçici Aramanın önerdiği aday bölgelerin konumsal hataları, sınıfa özel eğitilen sınırlayıcı kutu regresyon modeliyle düzeltilmeye çalışılmıştır. Fast R-CNN [1], her aday bölgenin öznitelik haritasını ayrı ayrı hesaplamak yerine giriş görüntüsünün tamamının öznitelik haritasını bir seferde hesaplayıp Seçici Aramanın önerilerini bu harita üzerinden kesecek RoI örnekleme (pooling) yöntemini önermiştir. Paylaşılan hesaplama sayesinde ciddi hız kazanımı elde edilmiştir. Faster R-CNN [11], deterministik Seçici Arama algoritmasını öğrenilebilir parametreleri vasıtasıyla performansı artıracak Bölge Önerim Ağı (RPN) ile değiştirmiştir. Daha hızlı öneri sunan RPN, mimariyi hızlandırırken aynı zamanda modelin uçtan-uca eğitilebilmesini mümkün kılmıştır. Mask R-CNN [4] ise ata mimariye eklediği maske dallanması ile piksel düzeyinde bölütleme (segmentasyon) yeteneğini aileye kazandırmıştır. Segmentasyonun ince ayar konumsal doğruluk gerektirmesi öneri bölgelerinin öznitelik haritasından kesildiği RoI pooling işleminin geliştirilerek RoI hizalama (align) yöntemine evrilmesini sağlamıştır. Artırılan hassasiyet doğal olarak sezim performansını da artırmıştır. Bu nedenle önerilen izleyicide nesne sezici olarak Mask R-CNN seçilmiştir.

Sezim-ile-izleme metodolojisine göre, herhangi bir ilave yapının bulunmadığı basit bir izleyici tasarımında, sezicinin nesne sınıf skorlarına dayanan bir hedef eşleştirme kuralı tanımlanabilir; ancak, bu yöntemin izlenen nesne sınıfıyla aynı sınıfta başka örneklerin bulunduğu videolarda başarısız olacağını öngörmek zor değildir. Hedef eşleştirme için sezicinin skorları yerine özniteliklerinin kullanıldığı durumda, özniteliklerin hedefe benzerliğine göre bir karar verilebilir. Buradaki sorun ise sezici özniteliklerinin sınıf-içi nesne kimliklerini ayırt edebilecek kadar güçlü olmamasıdır. Örneğin, nesne sezici eğitimi sırasında insan ile arabayı tanımayı öğrenirken insanları kendi içinde dış görünüşlerine göre birbirinden ayırt edemez. Bitirme kapsamında önerilen izleyicide, nesne sezici Mask R-CNN'i takiben 'yeniden-tanılayıcı' (ReID) öznitelik çıkarıcısı yerleştirilmiştir. ReID modeli kişilerin benzerliklerini öğrenmesi için eğitildiğinden bu öznitelikler kişilerin kimlik bazında tanınmasını sağlamaktadır. Gerçeklenen nesne izleyicinin temel alt bloklarını gösteren şema Şekil 1.1'de görülmektedir. İzleyici girişine gelen her video çerçevesi öncelikle nesne seziciden geçirilerek olası hedef nesne konumları belirlenir. Her bir aday konum içerisindeki görüntü parçası ve verilen hedef görüntüsü ReID öznitelik çıkarıcıdan geçirildikten sonra kosinüs benzerlik kriterine göre eşleme yapılarak hedefin son konumu kestirilir. En yüksek benzerlik skorlu bu aday, hedef nesne kestirimi bloğundan izleyicinin son kararı olarak çıkışa aktarılır.



Şekil 1.1 : İzleyici blok şeması

Önerilen izleyicide, nesne sezici ve ReID ağlarının eğitimi birbirinden bağımsız gerçeklenmiş olup izleme amaçlı uçtan-uca bir eğitim gerekmemektedir; bu nedenle, farklı sezme ve ReID ağları ile birlikte kullanılabilme esnekliği sağlamaktadır. Kişileri ayırt etmek için eğitilmiş ReID modelinin farklı nesneler için basit bir transfer öğrenme ile özelleştirilmesi olasıdır.



Şekil 1.2 : Mask R-CNN ile önerilen izleyici performansı
VOT-LT group1, group2 ve group3 video performanslarının ortalaması.

Bitirme çalışması kapsamında izleyici performansı, güncel çalışmalarda sıklıkla kullanılan VOT-LT [12] (Visual Object Tracking - Long Term) ve LASOT [13] (Large-scale Single Object Tracking) veri setleri üzerinde raporlanmıştır. Şekil 1.2, VOT-LT veri seti group1, group2 ve group3 videolarında Mask R-CNN ile ReID özniteliklerinin ortalama izleme performansını karşılaştırmaktadır. İzleyicinin öneri kutusuyla, gerçek-referansın örtüşme alanları kesişiminin birleşimine oranı (Intersection over Union, IoU) yatay eksen olmak üzere düşey eksen başarım oranı IoU koşulunu sağlayan çerçeve sayısı oranıdır. Örneğin, örtüşme eşiği 0.5 seçilirse başarım oranı Mask R-CNN öznitelikleri ile %17 olurken ReID öznitelikleri eklenmiş izleyicide %68'e çıkabilmektedir.

İzleyicinin güncel en iyi durumla (SOTA) karşılaştırması, VOT-LT veri setinde hedef nesnenin 'insan' olduğu 21 video[1] üzerinden, Çizelge 1.1'de verilmiştir. Sıralama, izleyicilerin ortalama örtüşme skorlarının F ölçüsüyle yapılmıştır.

Bitirme çalışmasında önerilen izleyicinin gerçeklemesinde, COCO [15] veri setiyle ön-eğitimli Mask R-CNN'in Tensorflow [16] ve Market-1501 [17] veri setiyle ön-eğitimli Piramit ReID'nin PyTorch [18] model dosyaları OpenCV-Python [19] ortamından çağrılarak DNN modülüyle program akışına entegre edilmiştir. Hazırlanan izleyici yazılımı matris işlemleri için NumPy [20], makine-öğrenmesi fonksiyonları

[1]ballet, bicycle, group1, group2, group3, kitesurfing, longboard, person2, person4, person5, person7, person14, person17, person19, person20, rollerman, skiing, sup, tightrope, wingsuit, yamaha

Çizelge 1.1 : Önerilen izleyicinin performansının güncel (SOTA) izleyicilere göre durumu.

Modeller	F-Skor
1. LT_DSE	0.723
2. LTMU [8]	0.710
3. CLGS	0.693
4. mbdet [14]	0.623
5. Önerilen	**0.605**
6. TDIOT [6]	0.601
7. Siamfcos	0.593
8. CooSiam	0.579
9. ASINT	0.559

için Scikit-learn [21] ve istatistiksel veri yapıları için Pandas [22] kütüphaneleriyle desteklenmiştir.

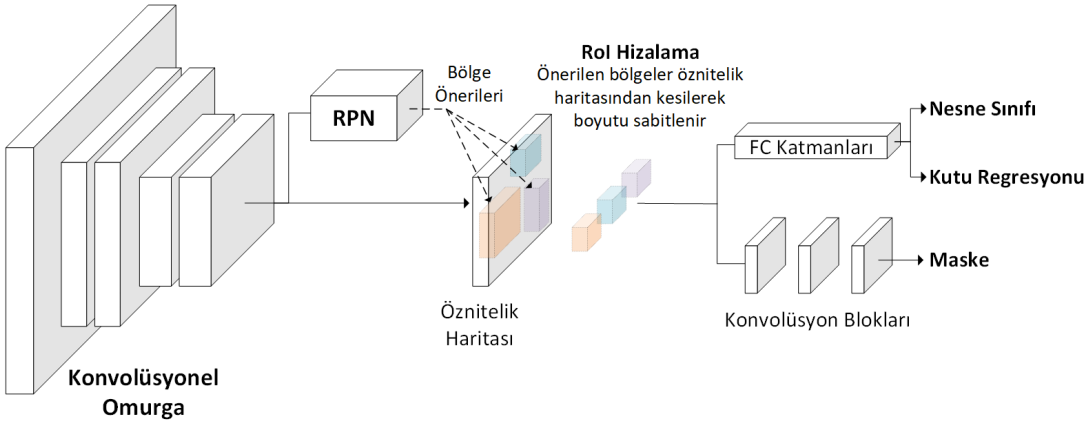
Bu çalışma sonucunda, ReID özniteliklerinin en yüksek benzerlik skoruyla hedef eşleştirmenin potansiyel başarımı görülmüştür. Bunun yanında, hedefin sahneden bir süreliğine ayrıldığı çerçevelerde, yanlış adayların benzerlik skorlarının düştüğü gözlemlense de izleyicinin var olan performansı düşürmeden 'hedef çıktı' kesin yargısına varmasını sağlayacak optimum karar belirlenememiştir. Ayrıca, Mask R-CNN'in hedefle örtüşen bir öneri veremediği çerçevelerde izleme kopuklukları meydana gelmektedir. Harici bir öneri ağıyla izleme sürekliliğinin devam ettirilmesi ve hedef giriş-çıkışlarının sezilmesi üzerine çalışmalar devam etmektedir.

2. NESNE SEZİCİ

Derin öğrenme tabanlı görsel nesne sezimi, sayısal görüntülerdeki nesnelerin ilgili çerçevedeki ön eğitimli sınıfının tanınması ve konum bilgisinin algılanması olarak basitçe tanımlanabilir. Tanımdan da anlaşılacağı üzere nesne sezimi sınıflandırma ve konumlandırma şeklinde iki ana göreve ayrılmaktadır; bunun yanında, güncel derin öğrenme mimarileri bu çoklu görevi birleşik öğrenme yeteneğine sahiptir. Nesne sezicinin bitirme çalışmasındaki görevi Şekil 1.1'de izleyici giriş görüntüsü üzerindeki nesneleri yani aday kutuları önermektir.

Nesne sezici ağ mimarileri belirtilen görevleri tek aşamada veya iki aşamada gerçekleştirmelerine göre ikiye ayrılmaktadır. Genel olarak tek aşamalı mimariler daha hafif hesaplama yükü sayesinde iki aşamalılara göre daha kısa çıkarım süresine sahipken iki aşamalı mimariler ise daha yüksek performansıyla öne çıkmaktadır. Önerilen izleyicide performans isterleri göz önünde bulundurularak iki aşamalı Mask R-CNN mimarisi kullanılmıştır.

2.1 Mask R-CNN Mimarisi

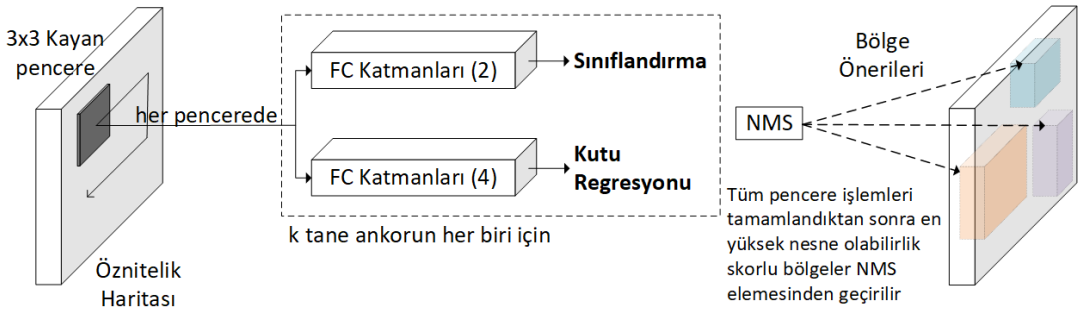


Şekil 2.1 : Mask R-CNN mimarisi

Mask R-CNN mimarisi Şekil 2.1'de görselleştirilmiştir. Öznitelik çıkarıcı konvolüsyonel omurgada, bu çalışmada InceptionV2 [23], hesaplanan öznitelik haritası Bölge Önerim Ağına (RPN) olası nesne bölgelerinin sınıftan bağımsız

yalnızca nesne olabilirlik skoruna göre önerilmesi için gönderilir. Aynı zamanda bu öznitelik haritası paralel olarak nesnenin gerçek sınıfının belirlenmesi için baş yapısına taşınır. RPN'den beslenen bölge önerilerine öznitelik haritasında karşı düşen alanlar RoI hizalama işlemiyle kesilerek sabit bir uzamsal boyuta indirgenir. Bu aşamadan sonrası Mask R-CNN'in baş yapısı (head) adını alır. Nesnenin sınıflandırılması ve bu sınıfa özel iyileştirilmiş sınırlayıcı kutu kestirimi için iki farklı koldan tam bağlı (fully-connected veya FC) katmanlara ilerletilen kesilmiş öznitelik haritalarının boyutu 7×7'dir. Sınıflandırma görevi için ayrılan FC katmanları çıkış boyutu eğitimde kullanılan COCO veri seti [15] sınıf sayısı 80'e eşittir ve eğitim sırasında gerçek-referanstan uzaklık maliyetinin belirlenmesinde Cross-Entropi kayıp fonksiyonu kullanılır. Konumlandırmanın kesinleştirilmesinde ise FC katmanları çıkış boyutu çizilen kutunun koordinat bilgisini açıklayacak biçimde 4'tür ve eğitim sırasında maliyet hesaplamasında Smooth L1 fonksiyonuna bağlanır. Bölütleme (segmentasyon) dalına ulaşan öznitelik haritalarının uzamsal boyutu 14×14 olup bu haritalar konvolüsyon bloklarından ilerletilerek çıkışta 28×28 boyutlu maskelere dönüşür. Bu maskelerin ilgili nesneye ait pikselleri seçebilmesi beklenir. Eğitim sırasında, her pikselin gerçek-referans maskeye ait olup olmadığına göre ikili Cross-Entropi optimizasyonuna gidilmektedir. Önerilen maske, sınırlayıcı kutu kestiriminde olduğu gibi sınıfa özeldir. Şekil 2.4.b nesne sezim kutuları ve piksel düzeyinde segmentasyon çıktı örneklerini göstermektedir.

RPN mimarisi. Bir görüntüdeki nesnelerin nerelerde konumlanabileceği problemine çözüm sunan RPN, konvolüsyonel öznitelik haritasını kullanarak olası nesne bölgelerini bu bölgelere ait nesne olabilirlik (objectness) skorlarıyla birlikte dönmektedir.



Şekil 2.2 : RPN mimarisi

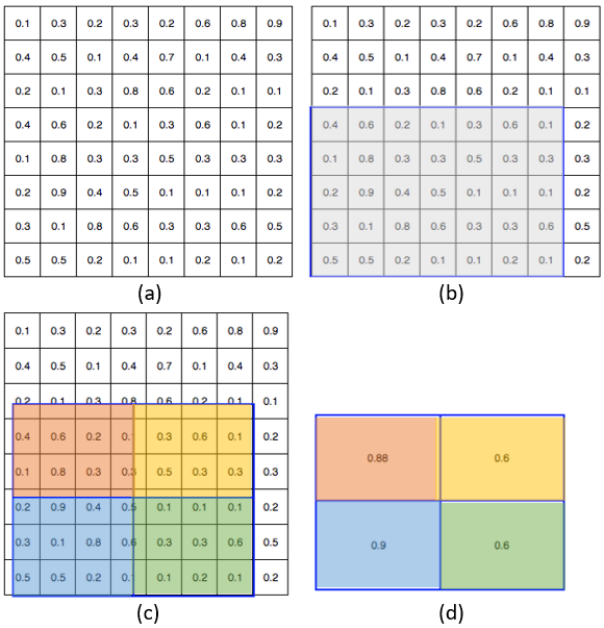
Şekil 2.2'de incelenen RPN mimarisine göre, bölge önerilerinin üretilebilmesi için öncelikle InceptionV2 omurgasının çıkış öznitelik haritası üzerinde 3×3 uzamsal

pencere gezdirilmektedir. Kayan pencere konumlarını baz alan k tane farklı boyut ve çerçeve oranlarında referans ankor kutuları oluşturulmaktadır. Ankorların öznitelik haritalarını filtreleyen bu konvolüsyon işlemini ReLu aktivasyon fonksiyonu takip etmekte ve her biri için 512 kanallı öznitelik haritası çıkarılmaktadır. Bu haritalar 1×1 konvolüsyonundan geçirilerek iki farklı koldan sınıflandırma ve kutu regresyonu FC katmanlarına ulaşır. Sınıflandırıcının görevi ilgili ankor alanında herhangi bir nesnenin olup olmadığı yönünde 2 çıkışlı bir olasılık dağılımı kararı vermektir. Eğitim sırasında bu maliyet ikili Cross-Entropi ile hesaplanmaktadır. Kutu regresyonunun amacı ise sınıflandırıcının nesne olabilir kararı üzerine ilgili nesneyi çevreleyen kutunun 4 konum bilgisini tahmin etmektir.

Bu detektörün eğitimini yapan araştırmacılar k'yı 3 farklı boyut ve 3 farklı çerçeve (0.5,1,2) kombinasyonu olmak üzere 9 seçmiştir. Adım kaydırma değeri $stride = 1$ durumunda, öznitelik haritasındaki ankor sayısı $W \times H \times k$ formülüyle hesaplanacaktır. Örneğin, 40×60 boyutlu bir harita yaklaşık 20,000 ankor kutusu içerecektir; ancak, oluşturulan ankor kutularının çoğunluğu harita boyutlarının dışına taşmaktadır. Bu kutular eğitim sırasında elendiğinde geriye yaklaşık 6,000 ankor kutusu kalacaktır.

Tüm pencerelerde sınıflandırma ve regresyon operasyonları tamamlandıktan sonra en yüksek nesne olabilirlik skoruna sahip bölgeler **Non-Maximum Supression** işlemine tabi tutulur. Bu işleme göre, RPN önerilerinden en yüksek olabilirlik skorlu bölge alınır ve diğer önerilerin her biriyle kesişim alanı birleşim alanına oranlanır. (Intersection over Union veya IoU) Örtüşme skoru belirli bir değerin üzerinde olan öneriler bastırılırken diğerleri korunur. Daha sonra kalan önerilerden en yüksek olasılıklı bölge ile süreç tekrarlanır. **NMS** sonunda elenmeyen bölge önerileri nesnenin COCO sınıfının tahmini ve kesinleşmiş konumunun kestirimi için Mask R-CNN baş yapısına aktarılır. Şekil 2.4.a RPN bölge önerilerinin örnek çıktısını göstermektedir.

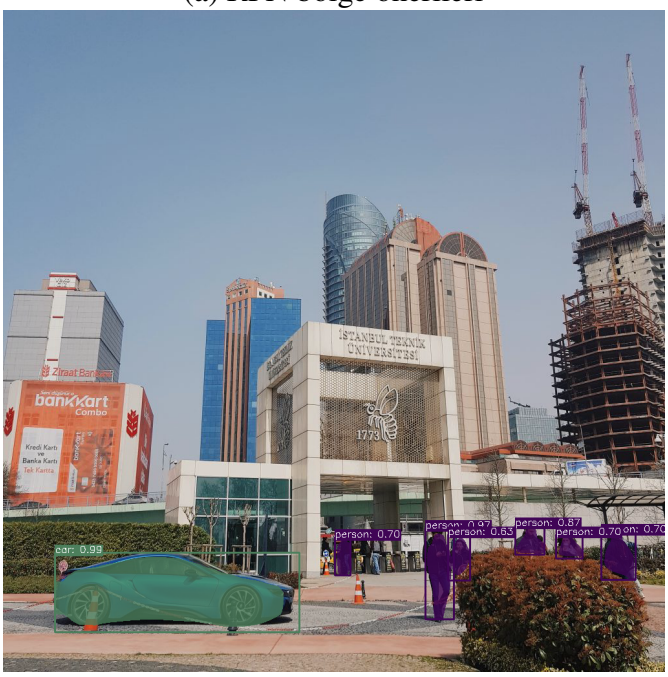
RoI hizalama. Sınıflandırma ve konumlandırma görevlerinin gerçeklemesi, giriş boyutunun değişken olamayacağı tam bağlı katmanlardan ilerletilmeyi gerektirir. Benzer şekilde segmentasyon görevinde de eğitim sırasında maskenin gerçek-referansla karşılaştırılabilmesi için belirlenen boyutlara sadık kalınmalıdır. Bu sebeple, RPN'in önerdiği bölgelerin boyutları sabitlenmelidir.



Şekil 2.3 : RoI hizalama

RoI hizalama işlemi basit bir illüstrasyon ile Şekil 2.3'te görselleştirilmiştir. (a) şıkkında bir öznitelik haritası bulunmaktadır. (b) ise RPN tarafından önerilen $5x7$ boyutlu alanın bu öznitelik haritası üzerine düşümüdür. Bu problemde amaç bölge boyutunu $2x2$'ye indirmektir. (c) RoI hizalama, öneri bölgesini eş uzaklıklı noktalara ayırdıktan sonra bu noktaları referans alarak bölgeyi olabildiğince eş $2x2$ parçalara böler. Bölge üzerinde seçilen noktaların her biri öznitelik haritasındaki en yakın 4 komşu köşeye uzaklığının bilineer interpolasyonu ile temsil edilmektedir. Noktaların haritadaki bu karşılık değerleri standart MaxPool ile (d) şıkkındaki hassas kesimle boyutu indirgenmiş öznitelik haritasına dönüşür.

(a) RPN bölge önerileri

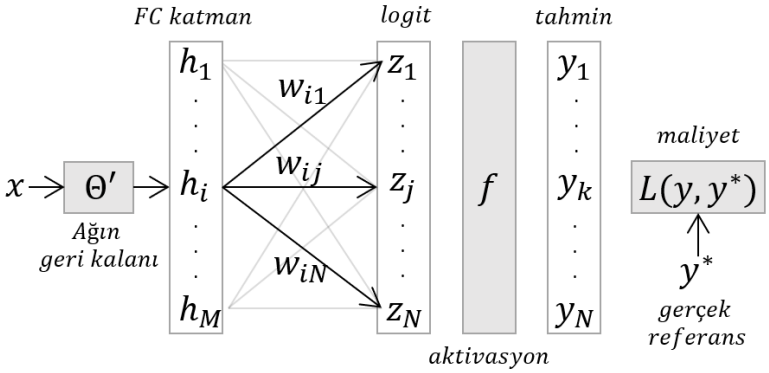


(b) Nesne sezim ve segmentasyon çıktısı

Şekil 2.4 : Mask R-CNN ağının çıktıları[1]

[1]Orijinal görüntü İTÜ mezunu öğrencilerden Mareva Zenelaj'a aittir.

2.2 Mask R-CNN Ağının Eğitimi



Şekil 2.5 : Derin ağ modeli illüstrasyonu

Mask R-CNN ağının eğitimi açıklanırken Şekil 2.5 tam bağlı (FC) katmanlar ile tahmin çıkışı veren derin ağ modeli illüstrasyonu ve notasyonundan faydalanılacaktır. Giriş örneği x'in ağ mimarisinin tüm parametreleri θ üzerinden ilerletilmesiyle önerilen tahmin y, son katmana ulaşan z logitlerinin f fonksiyonuyla aktivasyonunda elde edilen görüntüdür. z ise bir önceki tam bağlı katmanın çıkışı h'nin ağırlık w ve bayas b parametreleriyle $\underline{w} \cdot \underline{h} + \underline{b}$ lineer ilişkisinden ilerletilmesinin sonucudur. (Şekilde basitliğin sağlanması adına b eklenmemiştir.) Çıkarım aşamasını basitçe açıklayan bu süreç Denklem 2.1 ile ifade edilebilir.

$$\underline{y} = f(\underline{z} = \underline{w}\underline{h} + \underline{b}) \tag{2.1}$$

Ağın eğitimi sırasında x için çıkarılan tahmin y'nin beklenen gerçek-referans y^*'ye uzaklığı kayıp fonksiyonu $L(y, y^*)$ ile tanımlanır. Stokastik Gradyan İnişi (SGD), L'nin hesapladığı maliyeti enküçükleyecek θ'yı bulmak için her parametreden kendi hatasını çıkaran iteratif bir çözüm sağlar. Genel maliyetin her parametreye dağılımı zincir kuralıyla sağlanır. Örneğin, Şekil 2.5'te son katmanın j'inci düğümüyle bir önceki katmanın i'inci düğümünü bağlayan ağırlık $w_{i,j}$'ye Denklem 2.2 zinciriyle ulaşılabilir.

$$\frac{\partial L}{\partial w_{i,j}} = \frac{\partial L}{\partial y}\frac{\partial y}{\partial z_j}\frac{\partial z_j}{\partial w_{i,j}} \tag{2.2}$$

Hesaplanan gradyan değeri $w_{i,j}$'nin t iterasyonu için olması gerekenden uzaklığıdır. Gradyan inişi, Denklem 2.3'te bir sonraki iterasyon $t+1$'de bu farkı azaltacak güncellemeyi öğrenme katsayısı α'ya göre gerçekleştirir.

$$w_{i,j}^{t+1} \leftarrow w_{i,j}^{t} - \alpha\frac{\partial L}{\partial w_{i,j}^{t}} \tag{2.3}$$

Mask R-CNN mimarisinde 2'si RPN 3'ü baş yapısında olmak üzere toplam 5 maliyet terimi bulunmakta ve bu çoklu terimler birleşik eğitim sırasında SGD ile enküçüklenmektedir.

RPN Eğitimi. Konvolüsyonel omurganın hesapladığı öznitelik haritasını giriş alan RPN, harita üzerinde oluşturduğu ankorlarda nesne olup olmadığını sınıflandırmakta ve bir nesne olabilirlik skoru tahmin etmektedir. Eğer nesne var sınıflandırması yapmışsa bu ankoru temel alan bir sınırlayıcı kutu regresyonu gerçekleştirmektedir. Öyleyse, istenen çıktı nesnenin gerçekten var olduğu ankorların nesne olabilirlik skorunun 1 ve regresyonla önerilen sınırlayıcı kutunun gerçek referansla örtüşmesinin maksimum olmasıdır.

RPN genel kayıp fonksiyonunda, Denklem 2.4, ankorların gerçek referans kutularıyla örtüşmesi ölçülerek herhangi bir gerçek referansla 0.7'nin üzerinde IoU skoru bulunanlar 'nesne var', 0.3'ün altındakiler ise 'arkaplan' etiketlenmektedir. Bu koşulların dışındaki ankorlar ise eğitimin dışında tutulmaktadır.

$$L_{RPN}(p_i,t_i) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{2.4}$$

Eğitim kümesindeki her bir ankor i ile temsil edilmek üzere p_i^* o ankorun gerçek referans sınıfı ('nesne var':1, 'arkaplan':0), p_i sınıflandırıcının tahmin ettiği nesne olabilirlik skoru, t_i^* gerçek referans kutusunun 4 parametresi ve t_i sınırlayıcı kutu kestirimidir. Normalizasyon terimleri N_{cls} ve N_{reg} sırasıyla eğitim kümesindeki (mini-batch) toplam ankor sayısı ve 'nesne var' etiketlenmiş ankor sayısıdır. λ regularizasyon parametresi 10 seçilmiştir.

Sınıflandırma görevinde RPN genel kaybı Denklem 2.4'te ifade edilen L_{cls} terimi ikili Cross-Entropi (BCE) kayıp fonksiyonuyla Denklem 2.5'teki gibi tanımlanmıştır.

$$L_{cls} = L_{BCE}(p_i,p_i^*) = -[p_i^* \cdot log(p_i) + (1-p_i^*)\cdot log(1-p_i)] \tag{2.5}$$

L_{cls}, tahmin p_i olasılıklarını gerçek-referans p_i^*'ye yaklaştıracak cezalandırmayı sağlamak için [0,1] arasında değer alan olasılıkların görüntü kümesini logaritmayla $(-\infty,0]$ aralığına taşır. p_i^*'nin 0 olduğu durumda, birinci terim elenir; p_i'nin 0'dan uzaklığı kadar $log(1-p_i)$ terimi $-\infty$'a, ceza puanı ise ∞'a yaklaşır; tahmin p_i gerçek referansa eşitse ikinci terim $log(1)$ ve beraberinde ceza puanı sıfırlanır. p_i^*'nin 1 olduğu durum da bunun tam tersidir.

Derin ağ modeli Şekil 2.5 nesne olabilirliği yönündeki ikili sınıflandırma problemine uyarlanırsa logit ve tahmin düğüm vektörlerinin boyutu $N = 2$ olmalıdır. Aktivasyon fonksiyonu f ise z logit skorlarını p ayrık olasılık dağılımına dönüştürecek Denklem 2.6 ile belirtilen Softmax seçilmelidir. Sınıflandırıcının j sınıfı için önerdiği olasılık j logitinin üstel karşılığının tüm logitlerin üstel toplamına oranıdır.

$$p_j = Softmax(z_j) = \frac{exp(z_j)}{\sum_{logit=1}^{N} exp(z_{logit})} \tag{2.6}$$

İkili sınıflandırıcı için Softmax aktivasyonunun gradyan hesaplaması Denklem 2.7 ile yapılmaktadır. Elde edilen sonucun genellenmesi mümkündür. Logit ile tahmin düğümlerinin sırası karşılıklı olmadığında süreksizlik oluşmaktadır.

$$\begin{aligned} \frac{\partial p_1}{\partial z_1} &= \frac{\exp(z_1)}{\exp(z_1)+\exp(z_2)} - \left(\frac{\exp(z_1)}{\exp(z_1)+\exp(z_2)}\right)^2 = p_1(1-p_1) \\ \frac{\partial p_2}{\partial z_1} &= \frac{-\exp(z_2)\exp(z_1)}{(\exp(z_1)+\exp(z_2))^2} = -p_2 \cdot p_1 \end{aligned} \tag{2.7}$$

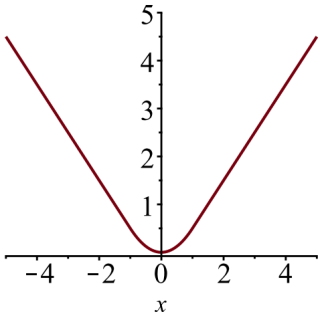
L_{cls}'nin hesapladığı maliyetin $w_{i,j}$ ağırlığına geri yayılım zinciri ise Denklem 2.8'de verilmiştir.

$$\begin{aligned} \frac{\partial L}{\partial w_{i,j}} &= \frac{\partial L}{\partial p_1}\frac{\partial p_1}{\partial z_1}\frac{\partial z_1}{\partial w_{i,j}} + \frac{\partial L}{\partial p_2}\frac{\partial p_2}{\partial z_1}\frac{\partial z_1}{\partial w_{i,j}} \\ &= \frac{-p_1^*}{p_1}[p_1(1-p_1)]h_i + \frac{-p_2^*}{p_2}(-p_2 \cdot p_1)h_i \\ &= h_i(p_2^* \cdot p_1 - p_1^* + p_1^* \cdot p_1) \\ &= h_i(p_1(p_1^* + p_2^*) - p_1^*) \\ &= h_i(p_1 - p_1^*) \end{aligned} \tag{2.8}$$

Belirlenen gradyan değeri Denklem 2.3'teki gibi $w_{i,j}$ parametresinin güncellenmesinde kullanılmaktadır. Güncelleme süreci, hesaplanan gradyan değerinin sıfıra yakınsaması gibi belirlenen durma kriteri sağlanana dek devam edecektir.

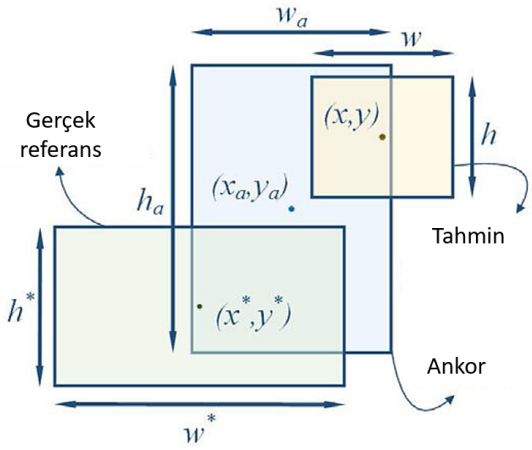
Kutu regresyonu görevi için RPN genel kaybı Denklem 2.4'te ifade edilen L_{reg} terimi, L_1^{smooth} kayıp fonksiyonu Denklem 2.9'da tanımlanmıştır ve ankorun gerçek referans sınıfı p_i^* 1 olduğunda aktif olmaktadır.

$$L_{reg} = L_1^{smooth}(t_i, t_i^*) = \begin{cases} 0.5(t_i - t_i^*)^2, & eger\,|t_i - t_i^*| < 1 \\ |t_i - t_i^*| - 0.5, & diger \end{cases} \tag{2.9}$$



Şekil 2.6 : Smooth L_1 fonksiyonu [1]

Smooth L_1 fonksiyonu nesne seziminde sınırlayıcı kutu regreyonu görevlerinde sıkça kullanılmaktadır ve istisnalara karşı gürbüz olduğu iddia edilmektedir. Şekil 2.6 fonksiyon karakteristiğini göstermektedir.



Şekil 2.7 : Tahmin, ankor ve gerçek referans sınırlayıcı kutuları [2]

RPN'den önerilen bir ankora ait öznitelik vektörünü giriş alan sınırlayıcı kutu regresyonu, ankoru temel alan merkez piksel koordinatlarını, genişliğini ve yüksekliğini açıklayan $t_i = (t_x, t_y, t_w, t_h)$ tahmin kutusunu oluşturur. t_x ve t_y tahmin edilen merkezin koordinatları x ve y'nin ankorun merkezi x_a ve y_a'dan sırasıyla çıkarılması ve ankorun genişliği w_a ve yüksekliği h_a ile normalize edilmesidir. t_w ve t_h ise tahmin ile ankorun genişlik ve yüksekliklerinin logaritmik oranıdır. Regresyon, gerçek referans $t_i^* = (t_x^*, t_y^*, t_w^*, t_h^*)$ parametrelerinin ankorla aynı ilişkisinin geometrik farkını minimize edecek dönüşümü öğrenmektedir. Öyleyse, regresyon hedefi Denklem 2.10 ile ifade edilebilir. Açıklaması yapılan tahmin, ankor ve gerçek

referans kutuları Şekil 2.7 ile görselleştirilmiştir.

$$\begin{aligned} t_x &= (x - x_a)/w_a \quad , \; t_x^* = (x^* - x_a)/w_a \\ t_y &= (y - y_a)/h_a \quad , \; t_y^* = (y^* - y_a)/h_a \\ t_w &= log(w/w_a) \quad , \; t_w^* = log(w^*/w_a) \\ t_h &= log(h/h_a) \quad , \; t_h^* = log(h^*/h_a) \end{aligned} \tag{2.10}$$

Derin ağ modeli Şekil 2.5 regresyona uyarlanırsa logit ve tahmin düğüm vektörlerinin boyutu sınırlayıcı kutunun 4 koordinat bilgisini açıklayacak biçimde $N = 4$ olmalıdır. Son katmanın aktivasyonu ise regresyon odaklı bir fonksiyon belirlenmelidir. ($f(z) = \alpha z$) Bu durumda Denklem 2.2'de olduğu gibi, FC katmanı h vektörünün i'nci düğümünü z logitlerinin j'inci düğümüne bağlayan ağırlık $w_{i,j}$'ye giden gradyan zinciri yazılmak istensin. Öncelikle L_1^{smooth} kayıp fonksiyonunun gradyanı Denklem 2.11 belirlenmelidir.

$$\frac{\partial L_{reg}}{\partial t_i} = \frac{\partial L_1^{smooth}(t_i, t_i^*)}{\partial t_i} = \begin{cases} t_i - t_i^*, & |t_i - t_i^*| \leq 1 \\ 1, & t_i - t_i^* > 0 \\ -1, & t_i - t_i^* < 0 \end{cases} \tag{2.11}$$

Lineer aktivasyon $f(z) = \alpha z$'nin türevi α olacağından $|t_i - t_i^*| \leq 1$ için $w_{i,j}$ ağırlığına Denklem 2.12 ile basitçe ulaşılabilir.

$$\frac{\partial L_{reg}}{\partial w_{i,j}} = h_i(t_i - t_i^*)\alpha \tag{2.12}$$

Mask R-CNN Baş Yapısı Eğitimi. Baş yapısı genel kayıp fonksiyonu Denklem 2.14'te, COCO veri seti [15] için $K = 80$ olmak üzere $K + 1$ kategorili sınıflandırma yapılmaktadır. İlave sınıf, 0 (sıfır) etiketli 'arkaplan' olup RPN'den gelen bölge tahmin kutularının gerçek referansla örtüşmesinin 0.5 IoU skorunun altında kalanlarını içermektedir. Sınıf etiketlerinin u ile temsilinde 1[u$\geq$ 1] göstergesi tanımlanmıştır. Böylece arkaplan sınıfı için regresör ve maske yitim fonksiyonları pasif kalmaktadır.

$$\begin{aligned} u &\in \{0, 1, ..., K\} \\ 1[u \geq 1] &= \begin{cases} 1 & u \geq 1 \\ 0 & diger \end{cases} \end{aligned} \tag{2.13}$$

$$L(p, u, t^u, t^*) = L_{cls}(p, u) + 1[u \geq 1] L_{reg}(t^u, t^*) + 1[u \geq 1] L_{mask}(y^u, y^*) \tag{2.14}$$

Genel maliyet denkleminde, p sınıfların olasılık dağılımı, u gerçek-referans sınıf örnek uzayı; t^u sınıfa özel sınırlayıcı kutu koordinatları, t^* gerçek-referans kutu koordinatları; y^u sınıfa özel maske ve y^* gerçek referans maskedir.

Sınıflandırma görevi için Denklem 2.14'te ifade edilen L_{cls} terimi yerine Cross-Entropi (CE) kayıp fonksiyonu tanımlanmıştır.

$$L_{cls} = L_{CE}(p_k, p_k^*) = -\sum_{k=1}^{N} p_k^* \cdot log(p_k) \tag{2.15}$$

Bu aşamadaki sınıflandırıcı FC katmanlarının, derin ağ modeli Şekil 2.5 karşılığında $N = K + 1 = 81$ olmalıdır. Geri yayılımı da RPN sınıflandırıcısına oldukça benzerlik göstermektedir.

Kutu regresyonu görevi yine RPN'de açıklanan regresyon tanımıyla örtüşmektedir. Tek fark, baş yapısındaki regresörün öğrendiği dönüşümün her sınıf için özel olmasıdır.

Segmentasyon görevi için Mask R-CNN $K \cdot m \cdot m$ boyutlu maskeler çıktısı vermektedir. Buna göre, Denklem 2.16'da m^2 boyutlu gerçek-referans maske y^* ile tahmin edilen maske y^u'nin yatay ve düşey piksellerinde i ve j indisleri ile gezilsin. y'nin her pikselinin gerçek maskeyle karşılaştırılması ikili sınıflandırma problemi olup yine BCE (Denklem 2.5) kayıp fonksiyonuna evrilmektedir.

$$L_{mask}(y^u, y^*) = -\frac{1}{m^2} \sum_{1 \le i,j \le m} y_{ij}^* \, log(y_{ij}^u) + (1 - y_{ij}^*) \, log(y_{ij}^u) \tag{2.16}$$

3. KİŞİYİ YENİDEN TANIMA (ReID) DESTEKLİ NESNE İZLEME

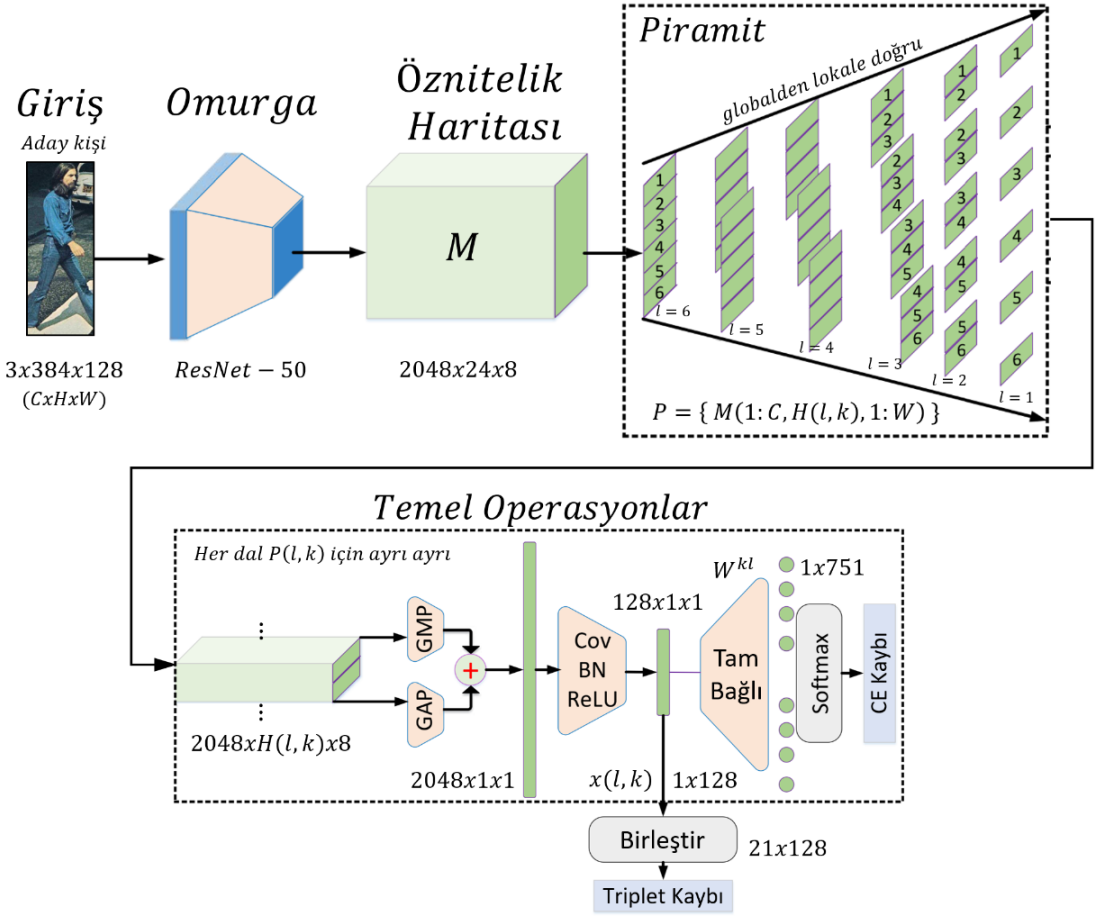
ReID ağının bitirme çalışmasındaki görevi Şekil 1.1'deki nesne sezici Mask R-CNN'in önerdiği aday kişilerin kimlik bazında tanınmasını sağlayacak özniteliklerini çıkarmaktır. Bu kullanımla birlikte izlenmek istenen hedef kişinin kimliği ilgili çerçevedeki diğer kişiler arasından ayırt edilebilecek ve takip eden çerçeveler boyunca nesne izleme devam edecektir.

Daha genel tanımla ReID ağları, bir insana ait sayısal görüntü için hesaplanan özniteliklerin aynı kişinin kamera açısı, ışıklılık ve poz değişimi gibi farklı fiziksel koşullar altındaki öznitelikleriyle benzerlik göstermesini ve aynı zamanda farklı kişilerin özniteliklerinden de olabildiğince uzak tutarak o kişiye özgün kimliksel öznitelik haritası yaratmayı amaçlar. Eğitim verisinde farklı kişilere ait görüntü kümeleri kullanılır ve her kişi bir sınıf olarak düşünüldüğünde kişiyi yeniden tanıma görevi sınıflandırma problemine dönüşür. ReID ağları da geleneksel sınıflandırıcı modelleri gibi giriş görüntüsünün gerçek referans sınıfına en yüksek olasılığı atamak üzere eğitilir. Ancak, eğitilen model çıkarım aşamasında sınıflandırma ağırlıklarını kullanmaz ve sınıf olasılık dağılımı yerine öznitelik haritası döner.

ReID ağlarının başarımı giriş kutularının düzgün kesilmiş olmasına oldukça bağımlıdır. Önerilen izleyicide ReID ağı direkt olarak Mask R-CNN çıkışıyla tümleştirildiğinden nesne sezimindeki hataların ReID ağına ekleneceği öngörülmüştür. Dolayısıyla, sezim kusurlarına olabildiğince gürbüz bir yaklaşım sergileyen 'Pyramidal Person Re-IDentification via Multi-Loss Dynamic Training' [3] ReID ağının bu bitirme çalışmasında kullanılması uygun bulunmuştur. Özetçe, kişilerin hesaplanan öznitelikleri lokal parçalara bölünür; daha sonra, parçalar piramide benzer biçimde globale doğru birleştirilerek farklı boyutlu ölçekler oluşturulur. Bu ölçekler, özellikle karşılaştırılan kişilerin görüntülerinde sezim kusurları olduğunda efektif rol alır. Bir ölçekteki kıyaslama başarısızsa bir başka ölçekte aramaya devam edilir ve bu şekilde tüm ölçekler incelenmiş olur. Seçilen ReID ağının bir başka güçlü yanı, dinamik eğitim

yapılarak iki farklı kayıp fonksiyonunun dönüşümlü kullanımından faydalanılması ve başarımın artırılmasıdır.

3.1 ReID Özniteliklerinin Çıkarılması



Şekil 3.1 : ReID ağının eğitim mimarisi [3]

ReID ağının eğitim mimarisini gösteren Şekil 3.1, nesne sezicinin belirlediği aday kişilerden bir görüntü almaktadır. Bu RGB görüntünün kanal sayısı C, uzamsal yükseklik H ve uzamsal genişlik W bileşen boyutları sırasıyla $3 \times 384 \times 128$'dir. Aday görüntü I, ResNet-50 [5] konvolüsyonel omurgasından ilerletilerek $2048 \times 24 \times 8$ boyutlu öznitelik haritası M çıkartılır. ResNet-50 mimarisinin ara katman çıkış boyutları Çizelge 3.1'de ayrıntılı verilmiştir.

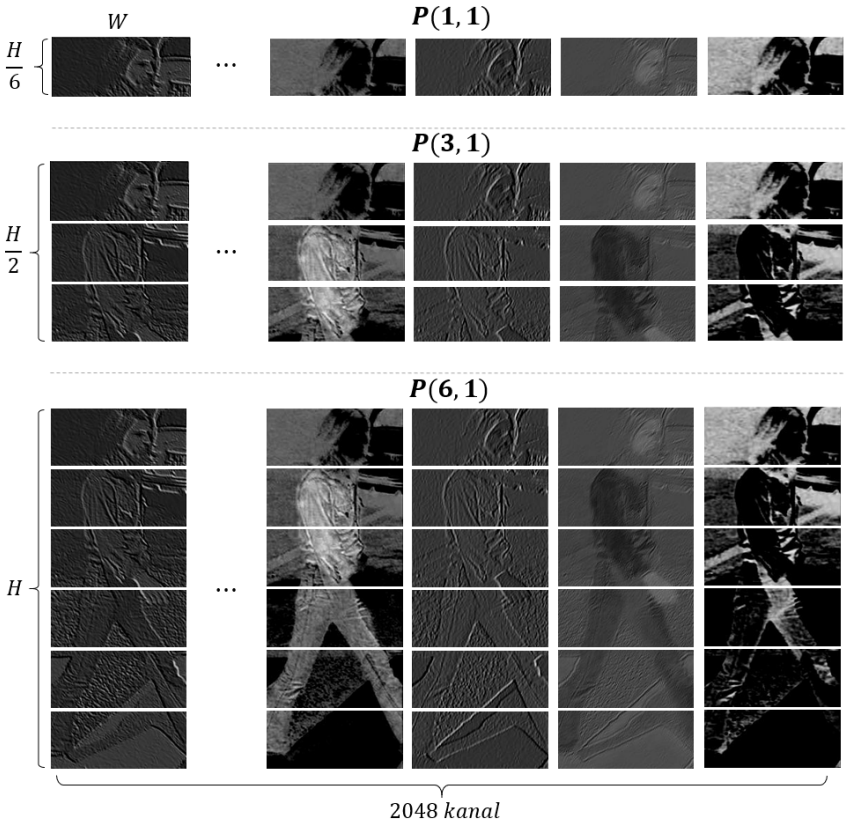
Öznitelik haritası M, uzamsal yükseklik H ekseninde n temel parçaya ayrılır. n, H'yi tam bölebilen bir sayı seçildiğinde her parçanın boyutu $C \times H/n \times W$'dir. Önerilen piramit modelinin taban seviyesi bu n temel parçadan oluşur. Bir üst seviye önceki seviyedeki her komşu iki parçanın birleştirilmesiyle üretilen tümleşik parçaları içerir. Tepe seviyesinde yalnızca bir parça kalana dek üst seviyeler tasarlanır. Piramidin farklı

seviyelerinde farklı boyutlara sahip bu parçalar M'nin kombinasyonel alt-haritalarıdır ve dal terimiyle adlandırılır. Referans bildiride seviyeler l (level) ve dallar k notasyonu ile gösterilmiştir. Öyleyse, taban seviyesinde $l = 1$ ve tepe seviyesinde $l = n$'dir, her seviyede $n - l + 1$ dal bulunabilir ve toplam dal sayısı $\sum_{l=1}^{n} l$ formülüyle hesaplanabilir. $n = 6$ durumunda, seviye sayısı ve tabandaki en küçük lokal dalların sayısı 6'dır; tüm seviyelerdeki toplam dal sayısı ise 21'dir. Piramidin tüm dallarını içeren P kümesi için l'inci seviyedeki k'ıncı dalın öznitelik haritası M ile ilişkisi $P(l,k)$ Denklem 3.1 ile tanımlanır:

$$\begin{aligned} P = \quad \{ \quad & M(1:C, ilk:son, 1:W): \\ & ilk \quad = (k-1)\cdot H/n + 1, \\ & son \quad = (k-1)\cdot H/n + l\cdot H/n\}, \end{aligned} \tag{3.1}$$

burada $1:C$ ifadesi M'nin C boyutunun 1'inci indisinden C'inci indisine kadar tüm bileşenleri içerdiği anlamına gelir. Benzer şekilde H ve W boyutlarının $P(l,k)$ dalına dahil edilen bileşenleri indis aralıklarıyla belirlenmiştir. Taban seviyesinden yukarı doğru daha büyük uzamsal ölçeklere erişilmektedir. Yani, tepe seviyesi $l = 6$ dalı $P(6,1)$, ResNet öznitelik haritası M'nin kendisidir ve en büyük global uzamsal ölçektir. Şekil 3.2, artan seviyeyle ulaşılan bazı piramit dallarından örnek göstermektedir.

Temel operasyonlar, P piramidinin her dalı için bağımsız gerçeklenir. Giriş alınan bir dalın farklı kanallarının istatistiksel özelliklerini yakalamak için öncelikle maske boyutları o piramit dalının uzamsal alanı $H \times W$'ye eşit olan GAP (global average pooling) ve GMP (global maximum pooling) işlemleri ayrı ayrı uygulanır. İki ayrı istatistiksel değişkenler kanal uzunluğu korunacak biçimde toplanarak vektör elde edilir. Ardından, bu vektör sırasıyla 2D-konvolüsyon, batch normalizasyonu ve ReLu aktivasyonundan ilerletilerek boyutu sabitlenmiş (128-dim) öznitelik vektörüne dönüşür. Bu vektör, temel operasyonların uygulandığı l'inci seviye k'ıncı piramit dalı $P(l,k)$'yı açıklayan son öznitelik vektörü $x(l,k)$'dır.



Şekil 3.2 : ResNet çıkışından alınan öznitelik haritası M'nin bazı piramit dalları. Burada, $P(1,1)$ en lokal ölçekli dallardan biriyken $P(6,1)$ ise en global ölçekli M'nin kendisidir.

Çizelge 3.1 : Piramit ReID'nin ResNet-50 omurga mimarisi

Katman Adı	Çıkış Boyutu	ResNet-50
giriş (I)	3x384x128	-
conv1	64x192x64	7x7, 64, stride 2
conv2_x	64x96x32 256x96x32	3x3 max pool, stride 2 $\begin{bmatrix} 1x1, 64 \\ 3x3, 64 \\ 1x1, 256 \end{bmatrix}$ x 3
conv3_x	512x48x16	$\begin{bmatrix} 1x1, 128 \\ 3x3, 128 \\ 1x1, 512 \end{bmatrix}$ x 4
conv4_x	1024x24x8	$\begin{bmatrix} 1x1, 256 \\ 3x3, 256 \\ 1x1, 1024 \end{bmatrix}$ x 6
conv5_x	2048x24x8	$\begin{bmatrix} 1x1, 512 \\ 3x3, 512 \\ 1x1, 2048 \end{bmatrix}$ x 3
çıkış (M)	2048x24x8	-

3.2 ReID Ağının Dinamik Eğitimi

Bu çalışmada 'Pyramidal Person Re-IDentification via Multi-Loss Dynamic Training' [3] referansında önerilen ReID ağı kullanılmıştır. Önerilen ReID ağ eğitiminde dinamik eğitim yapılarak yüksek tanılama başarımına ulaşılması hedeflenmektedir. Dinamik eğitimde birden fazla kayıp fonksiyonu birlikte dönüşümlü olarak kullanılır. Eğitimde kayıp fonksiyonları arasında geçişin modellenmesi önem taşımaktadır. Bitirme çalışması ReID ağı eğitiminde referans bildiride önerildiği gibi sınıflandırma problemlerinde sıklıkla kullanılan Cross-Entropi kayıp fonksiyonunun yanı sıra Triplet kayıp fonksiyonu birlikte kullanılmaktadır. Denklem 3.2, t. iterasyondaki toplam eğitim hatası L'yi göstermektedir ve her iki kayıp fonksiyonunun ağırlıklı toplamı olarak modellenmiştir. Dinamik ağırlıklandırmanın temel amacı L'yi global minimuma en hızlı ulaştıracak optimizasyonu sağlamaktır. Bu doğrultuda, her iterasyonda Cross-Entropi ve Triplet kayıplarının seyri gözlemlenerek L'yi daha çok azaltacak kayıp fonksiyonuna odaklanılır. Focal kayıp [24] fonksiyonu $FL(.)$ bu odaklanmayı veya ağırlıklandırmayı sağlayan gözlemcidir. τ üst indisi gözlenen kayıp fonksiyonunu gösterir ve sırasıyla, ce ve tp kısaltmaları ile gösterilen Cross-Entropi ve Triplet kayıp fonksiyonlarından birisine karşı düşmektedir.

$$L = \sum_{\tau \in \{ce, tp\}} FL(p_t^\tau, \gamma) L_t^\tau \tag{3.2}$$

Focal kayıp fonksiyonu $FL(.)$ dinamik eğitimde t iterasyonunda hangi kayıp fonksiyonunun ne ağırlık ile toplam kayıp üzerinde etkili olacağını γ hiper-parametresi ve p_t^τ kayıp düşüm olasılığına bağlı olarak Denklem 3.3'e göre belirler. γ için 2 değerinin amaca uygun olduğu saptanmıştır. [3] Kayıp düşüm olasılığı p_t^τ ise τ kayıp fonksiyonunun t iterasyonunda ne kadar azalabileceğinin ölçüsüdür. Bu olasılık ne kadar büyükse τ kayıp optimizasyonunun lokal minimumda takılma olasılığı da o kadar fazladır; küçük olasılık ise global minimuma yakınsayacak daha büyük bir adımın habercisidir. Dolayısıyla, $FL(.)$ gözlemcisi toplam eğitim hatasını küçük p_t^τ olasılıklı τ kayıp fonksiyonuna odaklayarak ağırlıklandırır.

$$FL(p_t^\tau, \gamma) = -(1 - p_t^\tau)^\gamma log(p_t^\tau). \tag{3.3}$$

Kayıp düşüm olasılığı p_t^τ, τ kaybının önceki iterasyon değerlerine bağlı değişim oranıyla Denklem 3.4'e göre belirlenir. Burada payın $min\{.\}$ ile normalizasyonu τ kaybının ilerleyen iterasyonla birlikte azalmadığı durumlarda p_t^τ'yi 1'e sınırlamaktadır ve Denklem 3.3 bu olasılığa sahip τ kayıp fonksiyonunu, ağırlığını sıfırlayarak, toplam eğitim hatası L'den hariç tutmaktadır.

$$p_t^\tau = \frac{min\{k_t^\tau, k_{t-1}^\tau\}}{k_{t-1}^\tau} \tag{3.4}$$

Denklem 3.4'te görülen k_t^τ ifadesi, kayıp fonksiyonunun art arda iterasyonlardaki kayıpların unutma parametresi α'ya bağlı üstel azalan etkisi göz önüne alınarak Denklem 3.5 ile hesaplanır. Başlangıç iterasyonunda $k_{-1}^\tau = L_0^\tau$'dir. α hiper-parametresi ise 0.25 seçilmiştir. [3]

$$k_t^\tau = \alpha L_t^\tau + (1-\alpha)k_{t-1}^\tau \tag{3.5}$$

ReID ağının dinamik eğitiminde öğrenmenin devam edebilmesi için k_t^τ'nin ilerleyen iterasyonla azalması beklenmektedir. Bir diğer önemli unsur, Triplet örneklemesinin yeteri kadar etkili yapılabilmesidir. Denklem 3.6, eğer Triplet ile Cross-Entropi ağırlıkları oranı Triplet margini δ'dan küçükse,

$$\frac{FL(p_t^{tp}, \gamma)}{FL(p_t^{ce}, \gamma)} < \delta \tag{3.6}$$

örneklemenin Triplet öğrenmesi için elverişli olmadığı kararı verilir ve Denklem 3.2 yerine yalnızca Denklem 3.7 optimize edilir.

Cross-Entropi Kayıp Fonksiyonu. Her piramit dalı $P(l,k)$, temel operasyonlardan birbirinden bağımsız ilerletilerek tam bağlı katmanlar W^{kl} üzerinden $Softmax$ fonksiyonuna bağlanır ve çıkışından eğitim veri setindeki farklı kimlik (sınıf) sayısı boyutlu ayrık olasılık dağılımı elde edilir. Bu dağılımı takip eden Cross-Entropi kaybı ise her eğitim örneğinin gerçek-referans sınıfı olasılığını enbüyükleyecek cezalandırmayı üretir. Denklem 3.7, Cross-Entropi fonksiyonunda,

$$\begin{aligned} L^{ce} &= \frac{1}{N_{ce}} \sum_i \sum_{k,l} Softmax\left((W_c^{kl})^T x_i(l,k) \right) \\ &= \frac{1}{N_{ce}} \sum_i \sum_{k,l} -log \frac{(W_c^{kl})^T x_i(l,k)}{\sum_j (W_j^{kl})^T x_i(l,k)} \end{aligned} \tag{3.7}$$

i alt indisi batch içerisindeki her bir örneği, $x_i(l,k)$ o örneğin l'inci seviye k'ıncı piramit dalı öznitelik vektörünü, c yine aynı örneğin sınıfını ve W_c^{kl} piramidin (l,k) dalına ait

ağırlık matrisininin c'inci satırını temsil etmektedir. N_{ce} normalizasyon terimi, batch genişliği ile bir örneğin toplam piramit dalı sayısının çarpımıdır.

Geri yayılım sırasında başta tam bağlı katman düğümleri, ardından temel operasyonlarda yer alan konvolüsyon ve ResNet omurga ağının filtreleri öğrenilir.

Triplet Kayıp Fonksiyonu. Bir kimliğe ait öznitelik vektörü x_i ankor örneği seçilsin. Aynı kimliğin farklı görüntüsünün ve farklı kimliğin herhangi bir görüntüsünün öznitelik vektörleriyse sırasıyla pozitif x_i^P ve negatif x_i^N örnekler olsun. Amaç ankorun pozitife uzaklığının azaltılırken negatife uzaklığının artırıldığı yeni bir öznitelik uzayına geçmektir. Triplet fonksiyonu, bu uzaklıklar arasındaki farkı en az bir δ margin hiper-parametresine uyduracak cezalandırmayı sağlar.

Triplet kayıp fonksiyonunun öğrenmeye katkısı seçilen örneklere oldukça bağımlıdır. Ankor kimliğe çok benzer pozitif örneklerle hiç benzemez negatif örnekler kolay tripletlerdir ve öğrenmenin etkinliğini düşürmektedir. Cross-Entropide kullanılan sıradan rastgele mini-batch örneklemesi çoğunlukla bu verimsiz tripletleri seçmektedir; oysa, öğrenmenin ilerlemesi için tripletler ankora uzak pozitifler ile yakın negatiflerden örneklenmelidir. Bu sebeple, referans bildiride ID-dengeli zor triplet örneklemesi önerilmiştir.

Etkili tripletlerin seçilebilmesi için öncelikle 8 farklı kimlikten her birinin 8 görüntü içerdiği 64 örnekli rastgele bir batch seçilir. Oluşturulan batch içerisindeki örneklerin tüm piramit dalı çıkış öznitelik vektörleri $1x128$ boyutlu $x(l,k)$'lar hesaplanır ve Denklem 3.8 kuralına göre birleştirilir.

$$x = (x(1,1)^T, \ldots, x(l,k), \ldots, x(n,n-l+1)^T)^T \tag{3.8}$$

Zor tripletlerin belirlenmesi için her örneğe ait birleştirilmiş vektörlerin aynı piramit dalına karşı düşen parçaları işleme alınır. Yani, 1 örnekten 21 farklı piramit dalı için 21×128 boyutlu çıkış alınır; bir batch içindeki 64 örnek için $64 \times 21 \times 128$ öznitelik vektörleri elde edilir; bu 64 piramidin aynı dalları ayrılarak kendi aralarında $21 \times 64 \times 128$ boyutuna uygun biçimde kümelenir. Bir örnekleme kümesindeki üyeler arası uzaklıklar 64×64 matriste tutulur. Matris satırlarında sırayla gezilerek gerçek referans kimlikler gözetiminde en uzak pozitif P ve en yakın negatif N seçilir. Bu yöntemle bir küme içinde 64 triplet örneklenmiş olur. Aynı örnekleme algoritması diğer piramit dalları için de tekrarlanır.

İki öznitelik vektörü arasındaki uzaklık metriği Öklid ile hesaplandığında Triplet kayıp fonksiyonu Denklem 3.9 tanımlanır.

$$L^{tp} = \frac{1}{N_{tp}} \sum_{k,l} \sum_{i} \Big[\big|\big| x_i(l,k) - x_i(l,k)^P \big|\big|_2^2 - \big|\big| x_i(l,k) - x_i(l,k)^N \big|\big|_2^2 + \delta \Big]_+ \tag{3.9}$$

Burada, i alt indisi batch içerisindeki her bir örneği, $x_i(l,k)$ o örneğin l'inci seviye k'ıncı piramit dalı öznitelik vektörünü ve P ile N üst indisleri sırasıyla o örnek için seçilmiş pozitif ve negatif triplet eşlerini temsil eder. $\delta = 0.16$ margin değeridir, $N_{tp} = 21 \times 64$ toplam triplet sayısıdır ve son olarak $[.]_+$ ifadesi ceza puanının sıfırdan küçük olamayacağını belirtir.

Triplet fonksiyonunun geri yayılımı tam bağlı katmanlar olmaksızın direkt öznitelik vektörlerine ulaşır. Bu vektörler ise konvolüsyon işleminin bir sonucudur. Öyleyse, Triplet fonksiyonunun bir batch içindeki öznitelik vektörleri $x_i(l,k)$ için hesapladığı hatanın bu konvolüsyon filtrelerinden ilki $w^{(1)}$'e ulaşan gradyan zinciri Denklem 3.10 gibi yazılabilir.

$$\frac{\partial L^{tp}}{\partial w^{(1)}} = \frac{\partial L^{tp}}{\partial x_i(l,k)} \frac{\partial x_i(l,k)}{\partial w^{(1)}}. \tag{3.10}$$

Konvolüsyon işleminin geri yayılımı kısaca o filtrenin 180° sağa dönmesiyle hesaplanabilmektedir ve $flip(.)$ notasyonu ile gösterilebilir. O halde gradyan çözümü,

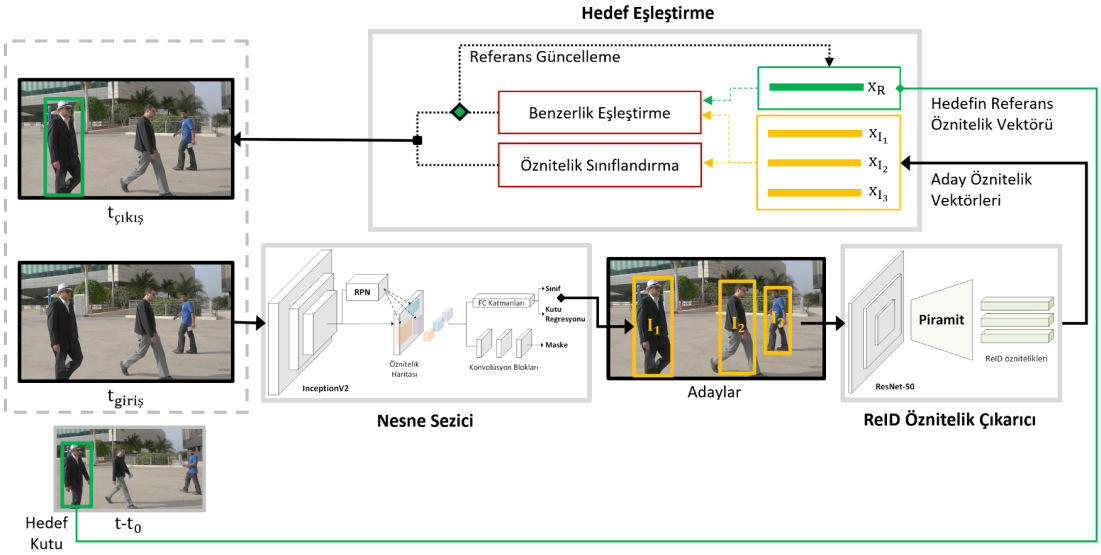
$$\frac{\partial L^{tp}}{\partial w_1} = \frac{1}{N_{tp}} \sum_{k,l} \sum_{i} 2(x_i(l,k)^N - x_i(l,k)^P) \cdot flip(w^{(1)}) \tag{3.11}$$

şeklinde basitçe ifade edilebilir. w_1 filtresi için elde edilen bu gradyan α öğrenme katsayısıyla ağırlıklandırılıp kendisinden çıkarılarak parametre güncellenir:

$$w_{t+1}^{(1)} \leftarrow w_t^{(1)} - \alpha \frac{\partial L^{tp}(w_t^{(1)})}{\partial w_t^{(1)}}. \tag{3.12}$$

Bu şekilde, ReID ağının öğrenilebilir parametreleri, temel operasyonlarda yer alan konvolüsyon bloğu ve ResNet omurga ağının filtreleri, eğitimin her iterasyonunda güncellenmeye devam eder. Durma koşulu sağlandığında modelin öğrenilen ağırlıkları izleyicide kullanılmak üzere kaydedilir.

3.3 ReID ile Tümleştirilmiş Nesne İzleme



Şekil 3.3 : İzleyici mimarisi

Yapılan bitirme çalışmasında literatürde varolan çalışmalardan farklı olarak, ReID ağı Mask R-CNN nesne sezicinin sınıflandırma ve regresyon katmanları ile tümleştirilmiştir. Bu sayede, Mak R-CNN karar vericinin sezilen nesne adaylarından en yüksek sınıf güven skorlu olanını izlenen hedef olarak raporlayan karar verme kuralının değiştirilmesi ve izleme başarımının artırılması hedeflenmektedir. Şekil 3.3, ReID'nin tümleştirilmesi ile gerçeklenen nesne izleme mimarisini göstermektedir. İzleyici t anındaki giriş çerçevesini ilk olarak Mask R-CNN'den ilerletir ve kutu regresyonu çıkışından olası nesnelerin konumunu alır. Aynı zamanda sınıflandırıcının skorlarını değerlendirerek 'kişi nesnesi' olmayan kutuları eler. İzlenen hedef kişi, geriye kalan kutulardan birinde olmalıdır ve bu kutuların her biri I_i, şekilde sarı renkle çizilerek 'aday' ismini almıştır. Adaylar arasından hedefin seçilebilmesi için her birinin ReID öznitelikleri x_{I_i} çıkarılır. Ardından, hedef eşleştirmenin yapılacağı üst bloğa gönderilir. Burada, son kararın verilmesinde 2 alternatif kural önerilmiştir. Benzerlik eşleştirmesi kuralında, hedefin geçmiş $t - t_0$ video çerçevesinde belirlenmiş öznitelik vektörü x_R referans alınarak t anındaki en benzer aday seçilir. Öznitelik sınıflandırma kuralı ise karar vermede referansa ihtiyaç duymadan yalnızca aday öznitelikleri sınıflandırarak doğru hedefi seçebilir ve hedefin çıkışını algılayabilir.

3.3.1 Benzerlik Eşleştirme

Bu kuralda, t çerçevesindeki adayların çıkarılan ReID özniteliklerinin her biri x_{I_i}'nin hedefin referans öznitelik vektörü x_R'ye benzerliği ölçülür ve en yüksek benzerliğe karşılık düşen aday I_i kutusu o çerçevenin son kararı olarak atanır. Şekil 3.3'te görüldüğü üzere referans öznitelik vektörü $t - t_0$ geçmiş çerçevesinde belirlenerek t çerçevesine benzerlik araması yapılması için taşınır. $t - t_0$ çerçevesindeki yeşil hedef kutunun ReID öznitelik vektörü x_R'dir. Benzerlik skorlarının hesaplanması için referans öznitelik vektörü ile aday öznitelikler 'Benzerlik Eşleştirme' bloğuna girer. Burada her aday öznitelik vektörü x_{I_i}'nin $L2$ normalizasyonu sonrasında referans vektör x_R ile arasındaki Kosinüs-benzerliği hesaplanır. Şekildeki örnekte $t - t_0$ çerçevesinde belirlenmiş beyaz şapkalı adam hedefinin öznitelik vektörü t çerçevesinde referans olarak kullanılması amacıyla saklanmıştır. Aday özniteliklerinden en yüksek benzerlik skorlu x_{I_1}'in ait olduğu aday kutu I_1, t çerçevesi çıkışında son karar atanarak yeşillendirilmiştir.

Benzerlik eşleştirme kuralını takiben referans öznitelik vektörünün güncellenmesi kararı verilmektedir. t_0 en son referans güncellemesi yapılan çerçeveden sonra t anına kadar geçen çerçeve sayısıdır. Yani, her çerçevede güncelleme yapılırsa t_0 daima 1'e eşittir. Hiçbir güncellemenin yapılmadığı durumda ise x_R ilk çerçevede gerçek-referans ile başlatılan hedef kutunun ReID öznitelik vektörüdür.

Benzerlik eşleştirme karar kuralının zayıflığı olarak hedefin her zaman adaylar arasında bulunacağı varsayımıyla herhangi bir kontrol sağlanmadan en yüksek skorlu adayla eşleşme yapılması gösterilebilir. Hedefin video akışı süresince sahneyi terk edip yeniden girme olasılığı göz önünde bulundurulduğunda harici bir denetimin gerekliliği anlaşılmaktadır. Ayrıca hedef sahneden ayrılmasa bile Mask R-CNN'in doğru aday kutuları önerebileceği kesin değildir. Akla gelen ilk çözüm yöntemlerinden biri olan benzerlik skoru eşiği belirlemek deneysel bir çaba gerektirir ve yapılan testlerde var olan performansın düşürülmeden 'hedef çıktı' kararı verilmesini sağlayacak optimum bir çözüm sağlanamamıştır.

3.3.2 Öznitelik Sınıflandırma

Benzerlik eşleştirme kuralının belirtilen hedefin çıkışını sezememe zayıflığı, öznitelik sınıflandırması kuralıyla aşılmaya çalışılmıştır. Bu kurala göre, her video çerçevesindeki tüm aday öznitelikler önceden eğitilmiş bir Random Forest [25] sınıflandırıcı modeliyle hedefe ait olup olmaması yönünde ikili-sınıflandırmaya tabi tutulur. Sınıflandırıcı kararına göre hiçbir aday öznitelik hedefle örtüşmüyorsa, izleyici o çerçevede hedefin sahnede bulunmadığı yargısına ulaşır. Öznitelik sınıflandırma kuralının bir diğer avantajı ise hedef eşleştirme için referans öznitelik vektörüne ihtiyaç duyulmamasıdır. Şekil 3.3'te öznitelik sınıflandırması kuralına yalnızca aday özniteliklerin girdiği görülmektedir.

Öznitelik sınıflandırıcısı Random Forest modelinin eğitimi için izleyici VOT-LT videolarında koşturularak öznitelik kayıtları alınmış ve Çizelge 3.3'te örneklerinin dağılımı paylaşılan sınıflandırıcı eğitim veri seti hazırlanmıştır. Mask R-CNN'in her çerçevedeki aday kutu önerilerinin gerçek-referans ile örtüşme oranları karşılaştırılmış ve en az 0.2 IoU skoru koşulunu sağlayan en yüksek örtüşmeye sahip aday POZİTİF diğerleri NEGATİF etiketlenmiştir. Hedefin sahneden çıkış yaptığı çerçevelerdeki tüm adaylar yine NEGATİF sınıfta yer almaktadır. Veri setindeki videolar zamanda iki yarıya bölünerek hepsinin ilk yarısı test ve ikinci yarısı eğitim için ayrılmıştır. Çizelge 3.3 sırasıyla (a) eğitim ve (b) test kümelerindeki her videonun hedefin sahnede olduğu ve olmadığı çerçeve sayılarıyla birlikte bu çerçevelerdeki Mask R-CNN adaylarının POZİTİF ve NEGATİF etiketlenen örnek sayılarını göstermektedir. Bunun yanında, hedefin var olduğu çerçevelerdeki gerçek-referans örnekleri de eğitim kümesi POZİTİF örneklerine dahil edilmektedir. Hedef yokken, gerçek-referans NEGATİF örnekler doğal olarak bulunmaz.

Öznitelik sınıflandırıcısının yanlış adayların öznitelikleri ile hedefin özniteliklerini belirten NEGATİF ve POZİTİF sınıfları öğrenebilme başarımı Çizelge 3.2'de eğitim ve test performansı olarak sırasıyla (a) ve (b) tablolarında gösterilmektedir. Toplam örneklere bakıldığında sınıf örnek sayısı dengesizliği olduğu görülmekle birlikte eğitim performansının kesinlik ve duyarlılık metrikleri iki sınıf için de tam başarıma yakındır. Test verisinde POZİTİF gerçek-referans örnekleri bulunmadığından örnek

Çizelge 3.2 : Öznitelik sınıflandırıcısının NEGATİF ve POZİTİF sınıfları öğrenebilme başarımı

	Kesinlik (PR)	Duyarlılık (RE)	Ortalama (F1)	Toplam Örnek
(a) Eğitim performansı				
NEGATİF	0.998	0.999	0.998	139374
POZİTİF	0.998	0.995	0.996	64883
(b) Test performansı				
NEGATİF	0.931	0.958	0.944	129020
POZİTİF	0.803	0.705	0.751	30913

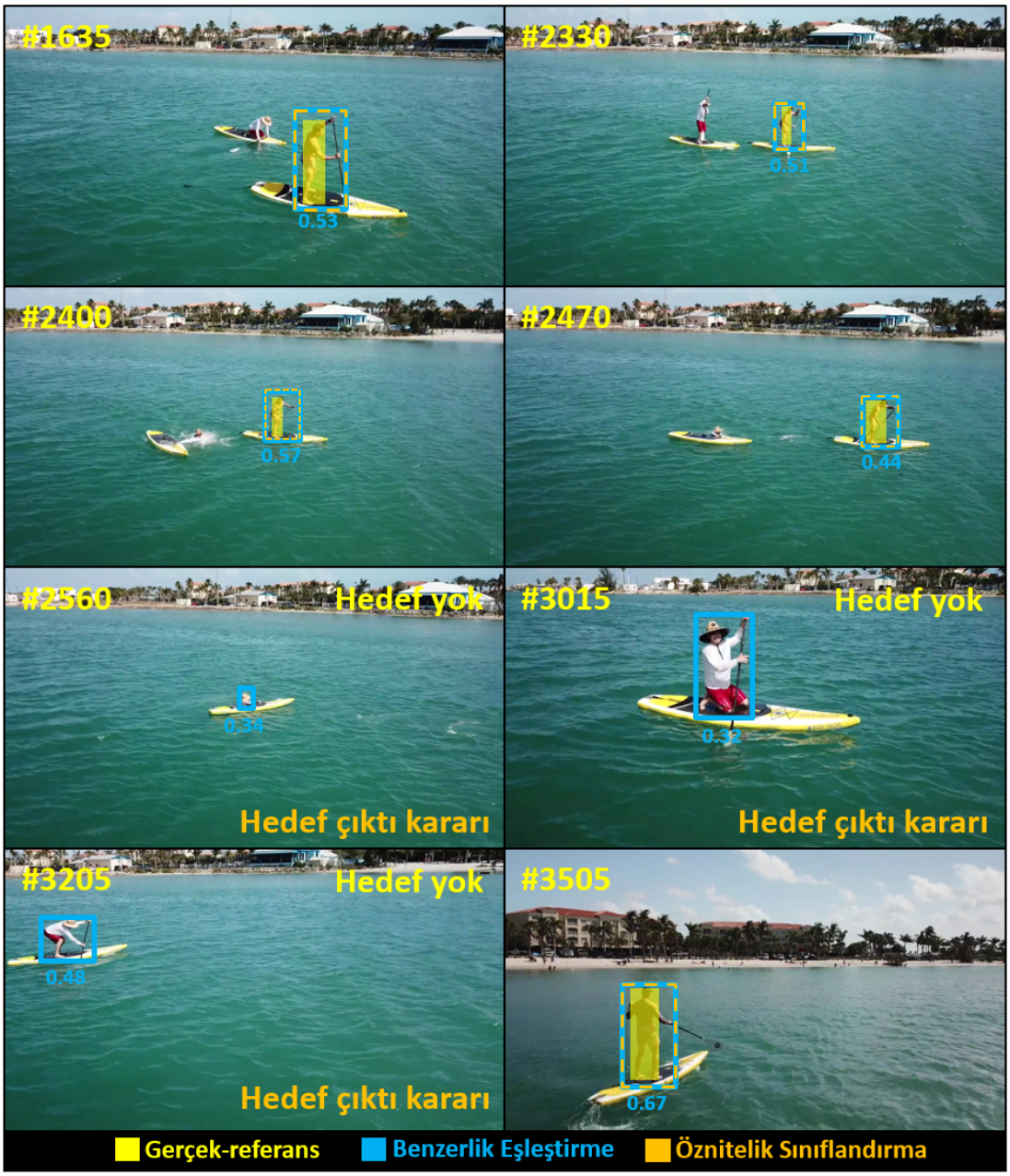
dağılım farkı açılsa da izleyicinin pratikte karşılacağı senaryoya daha uygundur. Test performansının NEGATİF sınıf metrik değerleri yüksek başarımını korumasına rağmen POZİTİF sınıfın özellikle duyarlılık metriğinin 0.705'e düşmesi izleyicinin hedefin sahnede olduğu çerçevelerde dahi çıkış kararı vermesiyle sonuçlanacaktır.

Eğitilen Random Forest sınıflandırıcısı Şekil 3.3 'Öznitelik Sınıflandırma' bloğu içerisinde yer alır ve izleme aşamasında aday özniteliklerin tamamını sınıflandırır. Hedefe ait olan öznitelik vektörünün POZİTİF diğerlerinin NEGATİF sınıflandırılması beklenir. Hiçbir öznitelik POZİTİF sınıflandırılmazsa hedefin sahneden ayrıldığı kararı verilir. Çok nadir karşılaşılan bir durum olmakla birlikte eğer birden fazla POZİTİF sınıflandırılan aday varsa 'Benzerlik Eşleştirme' kuralından yardım alınarak son karara gidilir. Açıklaması yapılan kuralın hedef eşleştirme ve çıkış sezme prosedürü Algoritma 1'de verilmiştir.

Algoritma 1: Öznitelik sınıflandırma kuralıyla hedef eşleştirme ve çıkış sezme algoritması

Veri: Bir çerçevedeki aday öznitelik vektörleri
Sonuç: Hedef kutu veya hedef çıktı kararı
Eğitilen Random Forest modeliyle aday öznitelik vektörlerini sınıflandır
if *tüm adaylar NEGATİF sınıflandırılmışsa* **then**
  return Hedef çıktı kararı
else if *bir aday POZİTİF sınıflandırılmışsa* **then**
  Hedef kutu ← POZİTİF aday kutu
  return Hedef kutu
else
  Hedef kutu ← Benzerlik Eşleştirme kuralına güven
  return Hedef kutu

Şekil 3.4 VOT-LT veri setinin sup videosundan alınmış bazı örnek video çerçeveleri göstermektedir. Sol üst köşelerde belirtilen sayılar ilgili çerçeve numaraları olmak



Şekil 3.4 : Benzerlik eşleştirme ve öznitelik sınıflandırma kurallarının VOT-LT 'sup' videosunda karşılaştırılması

üzere hedefin gerçek konumunu çizen kutular da sarıya boyanmıştır. Hedefin sahneden ayrıldığı bazı çerçevelerin sağ üst köşesinde 'Hedef Yok' yazısı yer almaktadır. 'Benzerlik Eşleştirme' ve 'Öznitelik Sınıflandırma' kurallarının izleme kararlarını sırasıyla mavi ve turuncu kutular temsil etmektedir. Mavi kutuların hemen altında yine mavi renkte benzerlik skorları bulunmaktadır. Örnek olarak seçilmiş 1635, 2330, 2400 ve 2470'nci çerçevelerde iki kural da gerçek-referansa en yakın hedefi seçebilmiştir. 2560, 3015 ve 3205'nci çerçevelerde 'Öznitelik Sınıflandırma' doğru bir şekilde hedefin sahneden ayrıldığı yönünde karar vererek yanlış kişiyi izlemeye

devam etmemiştir. Buna karşın, ‘Benzerlik Eşleştirme’ o sahnede var olan en yüksek benzerlik skorlu adayı hedef seçmiştir. İlerleyen çerçevelerde hedefin sahneye yeniden girişiyle, çerçeve 3505, ‘Öznitelik Sınıflandırma’ konum önermeye dönmüş ve ‘Benzerlik Eşleştirme’ ile aynı doğru karara varmıştır.

Çizelge 3.3 : Öznitelik sınıflandırıcısının eğitiminde kullanılmak üzere oluşturulan veri seti örneklerinin dağılımı

	Gerçek-referansa göre Çerçeve Sayıları		Mask R-CNN adaylarına göre Örnek Sayıları	
Video Dağılımları	Hedef var	Hedef yok	POZİTİF	NEGATİF
(a) Videoların eğitim için ayrılan ikinci yarısı				
ballet (2)	543	152	436	1477
bicycle (2)	1138	284	1069	1466
bike1 (2)	1543	0	1543	2667
group1 (2)	2437	0	2265	13703
group2 (2)	1200	141	972	7953
group3 (2)	2705	58	2688	12937
kitesurfing (2)	2333	0	2314	8203
longboard (2)	3419	110	3003	2881
person2 (2)	1311	0	1311	1093
person4 (2)	1371	0	1371	2786
person5 (2)	1051	0	1051	42
person7 (2)	992	41	970	229
person14 (2)	1461	0	1365	5456
person17 (2)	1147	26	1139	2258
person19 (2)	2003	176	1817	7936
person20 (2)	891	0	891	2624
rollerman (2)	670	186	667	1718
skiing (2)	1196	132	1139	2249
sup (2)	1045	709	1045	4136
tightrope (2)	933	212	829	1678
warmup (2)	1826	0	1825	53224
wingsuit (2)	1023	231	879	1153
yamaha (2)	1110	461	946	1505
	33348	**2919**	**31535**	**139374**
(b) Videoların test için ayrılan ilk yarısı				
ballet (1)	584	110	402	985
bicycle (1)	1142	278	1084	1577
bike1 (1)	1542	0	1538	4070
group1 (1)	2436	0	2430	9400
group2 (1)	1275	67	1252	6277
group3 (1)	2617	147	2479	19678
kitesurfing (1)	2332	0	2317	2361
longboard (1)	2070	1460	1774	4087
person2 (1)	1312	0	1312	162
person4 (1)	1372	0	1372	7464
person5 (1)	1050	0	1050	7
person7 (1)	1003	29	1003	88
person14 (1)	1424	38	1401	2748
person17 (1)	1142	32	1137	934
person19 (1)	2003	175	1951	5870
person20 (1)	892	0	887	1901
rollerman (1)	551	305	544	1978
skiing (1)	976	350	924	3094
sup (1)	1752	0	1752	2716
tightrope (1)	956	190	487	2645
warmup (1)	1855	0	1855	49066
wingsuit (1)	1149	105	1029	1173
yamaha (1)	1194	378	933	739
	32629	**3664**	**30913**	**129020**

Eğitime ayrılan toplam POZİTİF örnek sayısı 31535+33348=64883

4. BAŞARIM ANALİZİ

4.1 Mask R-CNN öneri konfigürasyonlarının izleyici performansına etkisi

Önerilen izleyicide hedefin izlenmesi Mask R-CNN'in sezebildiği aday kutularla yapılmaktadır. Başarılı bir izleme için hedefin adaylar arasında yer alması gerekir. Dolayısıyla, izleyici performansı Mask R-CNN'in sezim performansına bağımlıdır.

Performans ilişkisi açıklanırken izleme kuralı hatalarının gerçek referans konumları kullanılarak elendiği bir test geliştirilmiştir. Bu sebeple, elde edilen skorlar herhangi bir kuralın ulaşabileceği en yüksek performansı da belirtir ve sonraki Çizelgelerda bu skorlar **Mask-En iyi** sütunu altında referans değerler olarak paylaşılmıştır. Test sırasında, hedefin sahnede olduğu her çerçevede Mask R-CNN'in önerilerinden gerçek referansla en yüksek IoU örtüşmesine (minimum 0.2) sahip aday kutu seçilmektedir. Çizelge 4.1'de Mask R-CNN konfigürasyonlarının izlenen nesneyi sezebilme performansı karşılaştırılır.

- Birinci konfigürasyonda RPN'den yukarı 25/250 öneri taşınır ve ağ çıkışındaki 10 adaydan en az 0.3 sınıf güven skorlu 'person' kutuları alınır.
- İkinci konfigürasyonda RPN'den yukarı 1000/6000 öneri taşınır ve ağ çıkışındaki 100 adaydan en az 0.1 sınıf güven skorlu 'person' kutuları alınır.

İki konfigürasyonda da NMS eşik değerleri aynıdır: RPN ve ikinci aşama için sırasıyla 0.69 ve 0.6.

Karşılaştırma metrikleri **Avg IoU** hedefin gerçek referans kutuyla örtüşme oranları ortalaması, **Success Rate @0.5** en az 0.5 IoU örtüşme skoruna sahip çerçeve sayısı oranı, **MissRate** hedefin sahnede olmasına rağmen Mask R-CNN'in hiçbir öneride bulunmadığı çerçeve sayısı oranıdır.

VOT-LT veri seti karşılaştırması Çizelge 4.1.a'da Konfigürasyon-1'den daha fazla önerili Konfigürasyon-2'ye geçilmesiyle birlikte genel **Avg IoU** ortalamasının 0.05, **SR@0.5** ortlamasının 0.06 puan artış gösterdiği ve **MissRate**'in ise 0.06 puan azaldığı

görülmektedir. LASOT veri seti Çizelge 4.1.b'deki iyileşme daha belirgindir. Aynı metriklerdeki olumlu değişimler sırasıyla video oyunlardaki insansı karakterlerin izlenmek istendiği 'gametarget' grubunda 0.11, 0.14, 0.23 iken 'person' grubunda 0.09, 0.12, 0.05'dir. Konfigürasyon 2'nin bariz üstünlüğünden ötürü sonraki çalışmalarda bu konfigürasyon referans alınmıştır.

Çizelge 4.1 : Mask R-CNN öneri konfigürasyonlarının izleyici performansına etkisi

	Konfigürasyon-1			Konfigürasyon-2		
	Avg IoU	SR@0.5	MissRate	Avg IoU	SR@0.5	MissRate
(a) VOT-LT veri seti						
ballet	0.2406	0.195	0.344	0.3653	0.277	0.098
bicycle	0.5951	0.820	0.040	0.6169	0.853	0.020
bike1	0.5866	0.751	0.000	0.5959	0.774	0.000
group1	0.5506	0.663	0.000	0.5841	0.714	0.000
group2	0.5215	0.705	0.013	0.5984	0.822	0.002
group3	0.5133	0.672	0.009	0.5507	0.760	0.005
kitesurfing	0.6167	0.823	0.011	0.6367	0.857	0.006
longboard	0.4615	0.654	0.283	0.5886	0.826	0.092
person2	0.6926	0.965	0.000	0.6943	0.967	0.000
person4	0.7813	0.989	0.000	0.7877	0.993	0.000
person5	0.8069	1.000	0.000	0.8070	1.000	0.000
person7	0.7634	0.975	0.020	0.7704	0.984	0.011
person14	0.5484	0.705	0.031	0.5956	0.783	0.014
person17	0.6934	0.958	0.007	0.7045	0.964	0.002
person19	0.7360	0.908	0.051	0.7615	0.937	0.015
person20	0.7934	0.972	0.011	0.8068	0.989	0.001
rollerman	0.6190	0.865	0.097	0.6895	0.974	0.000
skiing	0.6379	0.855	0.071	0.6711	0.901	0.021
sup	0.6744	0.961	0.007	0.6857	0.980	0.000
tightrope	0.2961	0.348	0.316	0.4174	0.520	0.086
warmup	0.7217	0.943	0.000	0.7531	0.986	0.000
wingsuit	0.6657	0.779	0.214	0.7424	0.877	0.092
yamaha	0.4257	0.522	0.402	0.6006	0.749	0.172
	0.6062	**0.784**	**0.084**	**0.6532**	**0.847**	**0.028**
(b) LASOT veri seti						
gametarget1	0.0661	0.076	0.562	0.1877	0.224	0.346
gametarget3	0.3837	0.496	0.426	0.5282	0.679	0.202
gametarget4	0.0058	0.005	0.830	0.0338	0.035	0.021
gametarget5	0.4509	0.529	0.282	0.6056	0.728	0.054
gametarget6	0.1918	0.223	0.654	0.3951	0.492	0.355
gametarget7	0.0550	0.064	0.795	0.1332	0.158	0.467
gametarget9	0.6716	0.800	0.085	0.7401	0.894	0.010
gametarget10	0.7998	0.920	0.011	0.8268	0.954	0.001
gametarget11	0.7771	0.902	0.079	0.7984	0.932	0.058

	Konfigürasyon-1			Konfigürasyon-2		
	Avg IoU	SR@0.5	MissRate	Avg IoU	SR@0.5	MissRate
gametarget12	0.1745	0.158	0.458	0.4346	0.488	0.052
gametarget14	0.4094	0.537	0.285	0.5115	0.671	0.038
gametarget15	0.5126	0.692	0.182	0.5743	0.781	0.113
gametarget18	0.0142	0.009	0.908	0.0842	0.097	0.656
gametarget19	0.2853	0.311	0.100	0.4591	0.557	0.008
	0.343	**0.409**	**0.404**	**0.451**	**0.549**	**0.170**
person1	0.8108	0.916	0.006	0.8644	0.976	0.000
person2	0.3429	0.343	0.000	0.7024	0.786	0.000
person3	0.5091	0.600	0.000	0.7217	0.896	0.000
person4	0.7977	0.936	0.000	0.8289	0.984	0.000
person5	0.8259	0.951	0.000	0.8654	0.997	0.000
person6	0.8136	0.935	0.051	0.8178	0.941	0.043
person7	0.5193	0.605	0.001	0.6743	0.850	0.000
person8	0.7390	0.886	0.000	0.8042	0.968	0.000
person9	0.6719	0.783	0.000	0.7936	0.948	0.000
person10	0.9082	1.000	0.000	0.9081	1.000	0.000
person11	0.7537	0.904	0.032	0.8027	0.973	0.010
person12	0.1312	0.163	0.536	0.2288	0.292	0.113
person13	0.6757	0.935	0.017	0.7041	0.967	0.002
person14	0.8789	0.983	0.000	0.8885	0.997	0.000
person15	0.8419	0.976	0.005	0.8553	0.989	0.000
person16	0.7495	0.871	0.001	0.8093	0.951	0.000
person17	0.4814	0.560	0.000	0.6542	0.797	0.000
person18	0.6656	0.877	0.000	0.6932	0.924	0.000
person19	0.6558	0.830	0.010	0.7239	0.935	0.000
person20	0.5223	0.620	0.000	0.6861	0.839	0.000
	0.665	**0.784**	**0.033**	**0.751**	**0.900**	**0.008**

4.2 Farklı ReID modellerinin benzerlik eşleştirme performansları

Bu raporlamada ilk çerçevede belirlenen hedefin referans ReID özniteliğiyle benzerlik eşleştirme kuralının VOT-LT ve LASOT veri setleri üzerinde performansı gösterilmektedir. Ayrıca, iki farklı ReID modeli tarafından çıkarılan özniteliklerin izleyici performansına etkisi incelenmektedir. İki model arasındaki en temel fark içerdikleri öznitelik çıkarıcı omurgaların katman sayılarıdır. İlk ReID modelinde 50 katmanlı ResNet omurgası yer alırken diğerinde 101 katmanlı ResNet bulunur. İki modele de aynı adaylar verildiğinde çıkarılan öznitelikler benzerlik eşleştirme kuralıyla izlendiğinde ölçülen performans Çizelge 4.2'de görülmektedir.

Karşılaştırma metrikleri **Success Rate @0.5** en az 0.5 IoU örtüşme skoruna sahip çerçeve sayısı oranı, **PR** izleyiciye göre 'hedef var' denilen çerçevelerdeki ortalama IOU skoru ve **RE** gerçek-referansa göre hedefin sahnede olduğu çerçevelerdeki ortalama IOU skorudur. **F1** ise **PR** ile **RE**'nin harmonik ortalamasıdır.

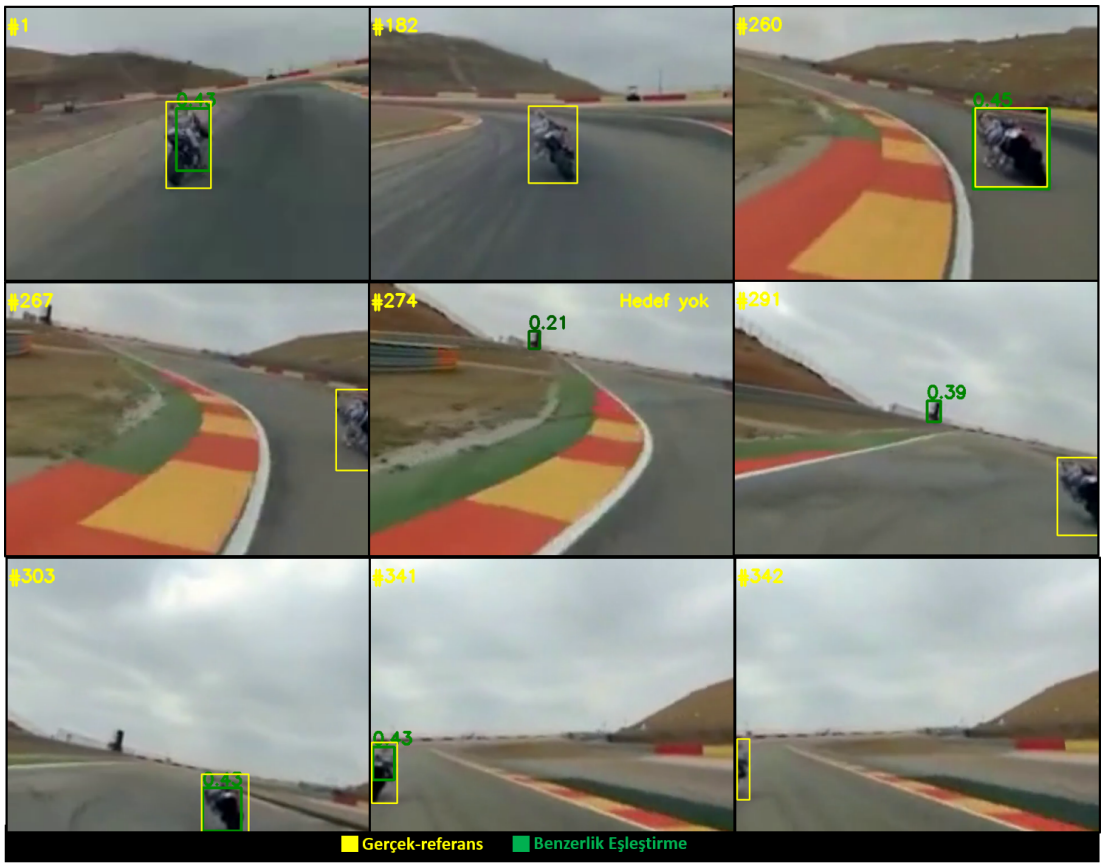
İki ReID modelinin izleme peformanslarında, video özelinde farklılıklar olsa da genel ortalamada Çizelge 4.2.a VOT-LT ve Çizelge 4.2.b LASOT 'gametarget' grubunda tüm metriklerde birbirine çok yakın performans elde edilmiştir. LASOT 'person' grubunda ResNet-101 omurgalı modelin ortalamada biraz daha üstün olduğunu söylemek mümkündür. Bunun yanında, öznitelik çıkarımı için daha çok katman bulundurmasının eklediği hız maliyeti ResNet-50 omurgalı modelin kullanımını öne çıkarmaktadır.

Çizelge 4.2 : Benzerlik eşleştirme kuralında ReID modelinin izleme performansına etkisi

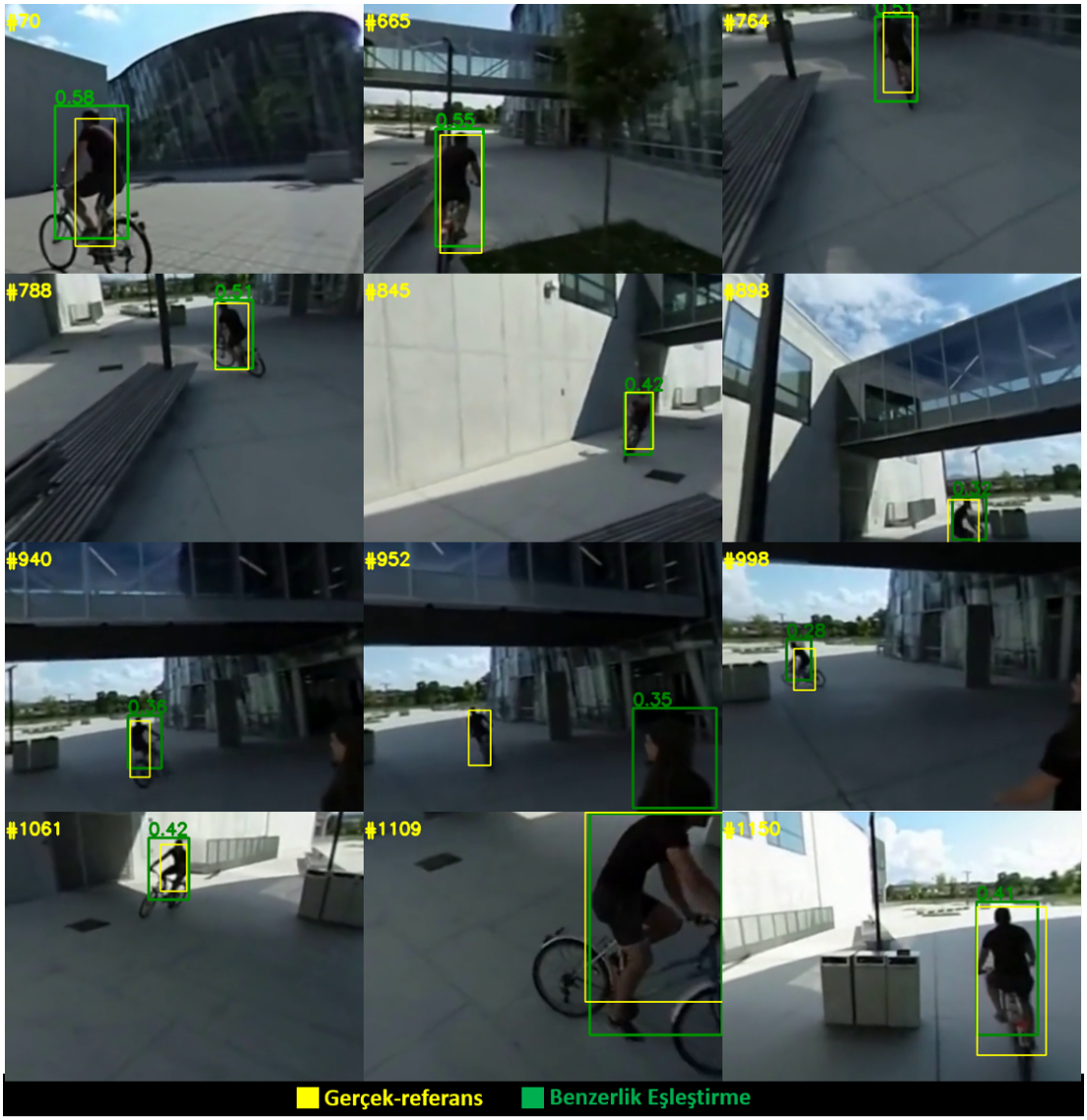
	ResNet-50 omurgalı ReID Benzerlik Eşleştirme				ResNet-101 omurgalı ReID Benzerlik Eşleştirme			
	SR@0.5	F1	PR	RE	SR@0.5	F1	PR	RE
(a) VOT-LT veri seti								
ballet	0.213	0.295	0.287	0.304	0.185	0.274	0.267	0.281
bicycle	0.782	0.534	0.500	0.574	0.770	0.544	0.513	0.578
bike1	0.752	0.588	0.588	0.588	0.756	0.590	0.590	0.590
group1	0.632	0.544	0.544	0.544	0.644	0.540	0.540	0.540
group2	0.729	0.533	0.513	0.555	0.741	0.530	0.510	0.552
group3	0.704	0.533	0.531	0.535	0.671	0.525	0.524	0.527
kitesurfing	0.780	0.596	0.598	0.595	0.806	0.609	0.611	0.607
longboard	0.785	0.561	0.557	0.564	0.793	0.566	0.563	0.569
person2	0.957	0.690	0.690	0.690	0.959	0.689	0.689	0.689
person4	0.985	0.782	0.782	0.782	0.985	0.781	0.781	0.781
person5	1.000	0.806	0.806	0.806	1.000	0.807	0.807	0.807
person7	0.972	0.766	0.769	0.763	0.963	0.762	0.766	0.759
person14	0.749	0.576	0.577	0.576	0.725	0.572	0.573	0.572
person17	0.943	0.684	0.678	0.690	0.936	0.681	0.675	0.687
person19	0.934	0.740	0.725	0.756	0.935	0.742	0.728	0.758
person20	0.988	0.805	0.806	0.805	0.987	0.806	0.806	0.805
rollerman	0.948	0.577	0.501	0.682	0.942	0.576	0.499	0.679
skiing	0.861	0.611	0.578	0.649	0.869	0.616	0.583	0.654
sup	0.886	0.563	0.507	0.634	0.844	0.530	0.476	0.596
tightrope	0.437	0.400	0.401	0.399	0.405	0.386	0.387	0.385
warmup	0.916	0.685	0.661	0.711	0.929	0.690	0.666	0.716
wingsuit	0.660	0.563	0.568	0.558	0.733	0.646	0.671	0.622
yamaha	0.573	0.533	0.554	0.514	0.621	0.555	0.577	0.534
	0.791	**0.607**	**0.597**	**0.621**	**0.791**	**0.609**	**0.600**	**0.621**

	ResNet-50 omurgalı ReID Benzerlik Eşleştirme				ResNet-101 omurgalı ReID Benzerlik Eşleştirme			
	SR@0.5	F1	PR	RE	SR@0.5	F1	PR	RE
(b) LASOT veri seti								
gametarget1	0.2061	0.218	0.276	0.180	0.2110	0.219	0.276	0.181
gametarget3	0.6596	0.573	0.645	0.515	0.6519	0.567	0.639	0.510
gametarget4	0.0160	0.020	0.021	0.020	0.0185	0.021	0.021	0.021
gametarget5	0.3583	0.357	0.367	0.347	0.3635	0.382	0.407	0.360
gametarget6	0.4824	0.479	0.610	0.394	0.4881	0.481	0.613	0.395
gametarget7	0.1417	0.161	0.232	0.124	0.1498	0.170	0.244	0.130
gametarget9	0.6398	0.580	0.583	0.577	0.6324	0.565	0.571	0.560
gametarget10	0.6247	0.568	0.568	0.568	0.6855	0.619	0.620	0.619
gametarget11	0.9096	0.805	0.829	0.781	0.9009	0.800	0.824	0.777
gametarget12	0.3416	0.394	0.405	0.383	0.2925	0.366	0.377	0.356
gametarget14	0.6474	0.509	0.519	0.499	0.6425	0.505	0.515	0.496
gametarget15	0.7321	0.578	0.615	0.545	0.7302	0.582	0.619	0.549
gametarget18	0.0933	0.123	0.241	0.083	0.0940	0.126	0.246	0.084
gametarget19	0.4060	0.369	0.371	0.368	0.4175	0.374	0.376	0.373
	0.447	**0.409**	**0.449**	**0.385**	**0.448**	**0.413**	**0.453**	**0.386**
person1	0.3058	0.284	0.284	0.284	0.2822	0.299	0.299	0.299
person2	0.6638	0.563	0.563	0.563	0.6592	0.560	0.560	0.560
person3	0.5476	0.465	0.465	0.465	0.5278	0.448	0.448	0.448
person4	0.9623	0.815	0.815	0.815	0.9586	0.812	0.812	0.812
person5	0.2700	0.255	0.255	0.255	0.3296	0.299	0.299	0.299
person6	0.8643	0.772	0.790	0.756	0.8852	0.793	0.812	0.776
person7	0.6140	0.514	0.514	0.514	0.5889	0.489	0.489	0.489
person8	0.9603	0.796	0.796	0.796	0.9587	0.795	0.795	0.795
person9	0.3268	0.282	0.282	0.282	0.3575	0.306	0.306	0.306
person10	0.9454	0.854	0.854	0.854	0.9249	0.836	0.836	0.836
person11	0.4391	0.370	0.371	0.368	0.3200	0.286	0.287	0.284
person12	0.1116	0.111	0.118	0.104	0.2101	0.183	0.196	0.171
person13	0.9148	0.671	0.672	0.671	0.9467	0.691	0.691	0.690
person14	0.9883	0.881	0.881	0.881	0.9900	0.883	0.883	0.883
person15	0.3719	0.330	0.330	0.330	0.6105	0.507	0.507	0.507
person16	0.9159	0.780	0.780	0.780	0.8683	0.741	0.741	0.741
person17	0.1378	0.135	0.135	0.135	0.0971	0.100	0.100	0.100
person18	0.7933	0.599	0.599	0.599	0.8815	0.656	0.656	0.656
person19	0.8806	0.689	0.689	0.689	0.8905	0.697	0.697	0.697
person20	0.2415	0.210	0.210	0.210	0.4245	0.353	0.353	0.353
	0.613	**0.519**	**0.520**	**0.518**	**0.636**	**0.537**	**0.538**	**0.535**

Şekil 4.1, 4.2 ve 4.3 VOT-LT veri setinden sırasıyla yamaha, bicycle ve group3 videolarından örnek çerçeveler ve izleyicinin kararlarını göstermektedir. Sarı kutular hedefin gerçek-referans konumunu ifade ederken yeşil kutular en yüksek benzerlik skorlu aday kutuyu çizmektedir. Benzerlik skorları yeşil kutuların üstünde yine yeşil renkle eklenmiştir.



Şekil 4.1 : VOT-LT 'yamaha' videosundaki hareket bulanıklığında izleyici performansı



Şekil 4.2 : VOT-LT 'bicycle' videosundaki ölçek değişikliklerinde izleyici performansı

Şekil 4.3 : VOT-LT ‘group3’ videosundaki birbirine çok benzeyen yakın ve küçük nesnelerde izleyici performansı

4.3 Benzerlik eşleştirmede hedef referansının güncellenmesi

Önerilen izleyici kurallarından benzerlik eşleştirmede her video çerçevesinde adayların hedefin referans öznitelik vektörüne benzerliğine göre karar verilmektedir. Şu ana kadar sunulan çizelgelerde referans vektör güncellemesi yapılmamıştır. Bu başlık altında olası tüm güncellemelerin yapıldığı durum incelenmektedir.

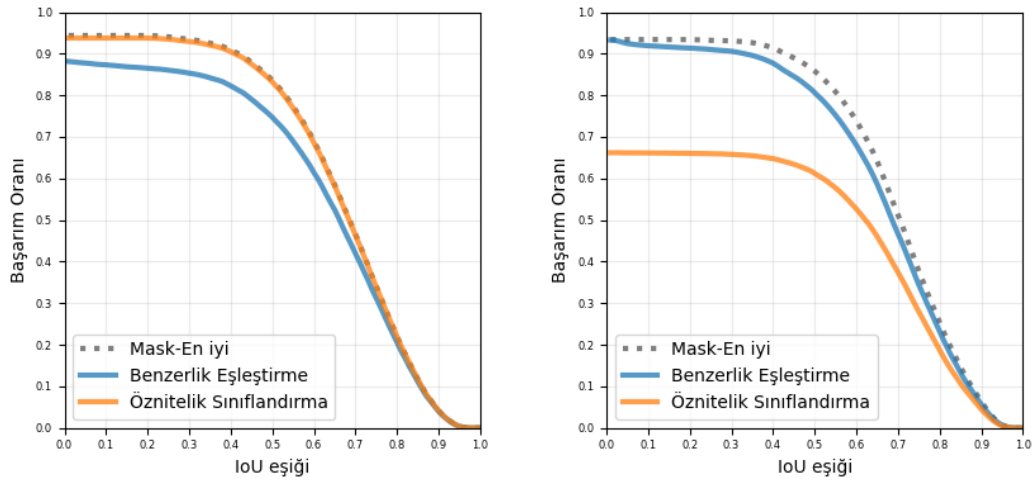
Çizelge 4.3 referans vektörünün yalnızca ilk çerçevede belirlenip hiç güncellenmediği 'Benzerlik Eşleştirme' ile her çerçevede güncellendiği 'Benzerlik Eşleştirme + Güncelleme' kurallarının izleme performanslarını karşılaştırır. Güncellemenin belirgin olumlu etkilerini 'sup', 'winguit' ve 'yamaha' videolarında görebiliyor olsak da ortak özelliği hedefe benzer kişilerin yakın hareket ettiği 'group1', 'group2', 'group3' ve 'warmup' videolarında yanlış güncelleme sonrasında izleme kontrolü kaybedilmektedir. Bu sorun göz önünde bulundurulduğunda güncelleme kararının deterministik bir yöntem yerine öğrenilebilir bir yaklaşımla ele alınması daha doğru olacaktır.

Çizelge 4.3 : Benzerlik eşleştirme kuralında her çerçevede hedef güncellemenin performansa etkisi

	Benzerlik Eşleştirme				Benzerlik Eşleştirme + Güncelleme			
VOT-LT	SR@0.5	F1	PR	RE	SR@0.5	F1	PR	RE
ballet	0.213	0.295	0.287	0.304	0.177	0.264	0.257	0.272
bicycle	0.782	0.534	0.500	0.574	0.737	0.498	0.466	0.535
bike1	0.752	0.588	0.588	0.588	0.658	0.495	0.495	0.495
group1	0.632	0.544	0.544	0.544	0.194	0.190	0.190	0.190
group2	0.729	0.533	0.513	0.555	0.328	0.238	0.229	0.248
group3	0.704	0.533	0.531	0.535	0.471	0.358	0.357	0.359
kitesurfing	0.780	0.596	0.598	0.595	0.822	0.615	0.617	0.613
longboard	0.785	0.561	0.557	0.564	0.762	0.538	0.534	0.541
person2	0.957	0.690	0.690	0.690	0.957	0.690	0.690	0.690
person4	0.985	0.782	0.782	0.782	0.985	0.782	0.782	0.782
person5	1.000	0.806	0.806	0.806	1.000	0.806	0.806	0.806
person7	0.972	0.766	0.769	0.763	0.972	0.766	0.769	0.763
person14	0.749	0.576	0.577	0.576	0.749	0.576	0.577	0.576
person17	0.943	0.684	0.678	0.690	0.943	0.684	0.678	0.690
person19	0.934	0.740	0.725	0.756	0.934	0.740	0.725	0.756
person20	0.988	0.805	0.806	0.805	0.988	0.805	0.806	0.805
rollerman	0.948	0.577	0.501	0.682	0.819	0.501	0.434	0.592
skiing	0.861	0.611	0.578	0.649	0.800	0.566	0.535	0.601
sup	0.886	0.563	0.507	0.634	0.955	0.596	0.536	0.671
tightrope	0.437	0.400	0.401	0.399	0.347	0.330	0.331	0.329
warmup	0.916	0.685	0.661	0.711	0.071	0.081	0.078	0.084
wingsuit	0.660	0.563	0.568	0.558	0.840	0.717	0.724	0.711
yamaha	0.573	0.533	0.554	0.514	0.703	0.600	0.623	0.578
	0.791	**0.607**	**0.597**	**0.621**	**0.705**	**0.541**	**0.532**	**0.552**

4.4 Benzerlik eşleştirme ve öznitelik sınıflandırma karşılaştırması

Bu başlık altında, Bölüm 3.3'te tanıtılan 'Benzerlik Eşleştirme' ve 'Öznitelik Sınıflandırma' hedef eşleştirme kurallarının nesne izleme performansları hem kendi aralarında hem de ulaşılabilecek en yüksek performansı belirten 'Mask - En iyi' ile karşılaştırılmaktadır. Çizelge 4.4 video özelinde sonuçları barındırırken Şekil 4.4 ortalama başarım oranı (SR) eğrilerini göstermektedir. Öznitelik sınıflandırıcısı eğitiminde kullanılan veri setinin eğitim-test dağılımı (Çizelge 3.3) bu raporlama için de geçerlidir.



(a) Eğitim performansı (b) Test performansı

Şekil 4.4 : Hedef eşleştirme kurallarının ortalama başarım oranı eğrileri

Şekil 4.4'de 'Mask En-iyi', 'Benzerlik Eşleştirme' ve 'Öznitelik Sınıflandırma' ortalama başarım oranları sırasıyla noktalı gri, mavi ve turuncu eğrilerle çizilmiştir. Öznitelik sınıflandırıcısının eğitildiği video çerçevelerini belirten eğitim performansı (a) karşılaştırıldığında öznitelik sınıflandırma başarımının erişilebilecek en yüksek performansa yakınsadığı görülürken benzerlik eşleştirme kuralının da bu değere yakın seyrettiği görülmektedir. Test performansı (b)'de öznitelik sınıflandırıcısı belirgin bir performans düşüşü yaşamış; buna karşın, eğitime ihtiyaç duymayan benzerlik eşleştirme yüksek performansını koruyabilmiştir.

Ayrıntılı skorların yer aldığı Çizelge 4.4'te PR ve RE metrikleri incelenerek hedef çıkışlarının performansa etkisi hakkında yorum yapılabilir. PR izleyiciye göre 'hedef var' denilen çerçevelerdeki ortalama IoU skoru ve RE gerçek-referansa göre hedefin

sahnede olduğu çerçevelerdeki ortalama IoU skoru olduğundan başarılı bir izleyicinin bu iki skorunun dengeli olması beklenir. Dengesizlik durumunda, PR yüksekken yanlış çıkış kararları yoğunluktadır. RE'nin yüksekliği ise hedef çıkışlarının kaçırıldığının göstergesidir. Çizelge 4.4'te, çıkış sezemeyen benzerlik eşleştirme kuralının gerek eğitim gerek test performansı ortalamasında az farkla RE skorunun yüksek olduğu görülmektedir. Buna karşın öznitelik sınıflandırma kuralının test performansındaki PR-RE dengesizliği performans kaybının yanlış çıkış kararlarına dayandığını kanıtlar niteliktedir.

Bu rapora göre benzerlik eşleştirme kuralı hedef çıkışlarını sezemiyor olsa da 'Mask En-iyi'ye yaklaşan bir performans sunmaktadır. Öznitelik sınıflandırma, hedefin adaylar arasında yer almadığını algılayabilme yetisiyle bir adım öne geçmekte ancak eğitim verisi dışında yüksek performansını sürdürememektedir.

Çizelge 4.4 : Benzerlik eşleştirme ve öznitelik sınıflandırma kurallarının izleme performansı karşılaştırması

	Mask - En iyi		Benzerlik Eşleştirme				Öznitelik Sınıflandırma			
	SR@0.5	PR/RE	SR@0.5	F1	PR	RE	SR@0.5	F1	PR	RE
(a) Eğitim performansı										
ballet	0.31	0.40	0.23	0.32	0.30	0.35	0.31	0.44	0.50	0.39
bicycle	0.84	0.60	0.76	0.52	0.48	0.56	0.83	0.62	0.64	0.60
bike1	0.78	0.60	0.77	0.60	0.60	0.60	0.78	0.60	0.60	0.60
group1	0.52	0.48	0.07	0.12	0.12	0.12	0.52	0.50	0.52	0.47
group2	0.71	0.52	0.62	0.46	0.43	0.48	0.69	0.57	0.65	0.51
group3	0.78	0.57	0.75	0.55	0.55	0.55	0.78	0.57	0.57	0.56
kitesurfing	0.81	0.61	0.34	0.28	0.28	0.27	0.80	0.61	0.61	0.61
longboard	0.84	0.58	0.66	0.49	0.51	0.47	0.83	0.62	0.67	0.58
person2	0.94	0.67	0.94	0.67	0.67	0.67	0.94	0.67	0.67	0.67
person4	1.00	0.83	0.99	0.83	0.83	0.83	1.00	0.83	0.83	0.83
person5	1.00	0.80	1.00	0.80	0.80	0.80	1.00	0.80	0.80	0.80
person7	0.97	0.77	0.94	0.77	0.77	0.76	0.97	0.78	0.79	0.77
person14	0.70	0.58	0.60	0.53	0.54	0.52	0.69	0.60	0.62	0.58
person17	0.96	0.72	0.91	0.69	0.68	0.70	0.95	0.72	0.72	0.72
person19	0.90	0.64	0.90	0.62	0.61	0.64	0.90	0.67	0.71	0.64
person20	1.00	0.82	1.00	0.82	0.82	0.82	1.00	0.82	0.82	0.82
rollerman	0.97	0.69	0.96	0.61	0.55	0.69	0.96	0.69	0.70	0.69
skiing	0.91	0.69	0.88	0.66	0.66	0.67	0.91	0.70	0.72	0.68
sup	0.98	0.64	0.89	0.44	0.35	0.59	0.97	0.64	0.64	0.64
tightrope	0.73	0.56	0.63	0.51	0.49	0.52	0.72	0.59	0.63	0.56
warmup	0.98	0.73	0.90	0.65	0.62	0.68	0.98	0.73	0.73	0.73
wingsuit	0.86	0.71	0.85	0.71	0.70	0.71	0.85	0.77	0.83	0.71
yamaha	0.76	0.61	0.60	0.53	0.53	0.53	0.75	0.65	0.71	0.60
	0.84	**0.65**	**0.75**	**0.57**	**0.56**	**0.59**	**0.83**	**0.66**	**0.68**	**0.64**
(b) Test performansı										
ballet	0.24	0.33	0.22	0.33	0.33	0.32	0.02	0.06	0.36	0.03
bicycle	0.87	0.63	0.78	0.54	0.51	0.58	0.18	0.22	0.63	0.14
bike1	0.77	0.59	0.73	0.58	0.58	0.58	0.64	0.53	0.58	0.49
group1	0.91	0.69	0.84	0.66	0.66	0.66	0.76	0.63	0.69	0.58
group2	0.93	0.67	0.86	0.62	0.60	0.63	0.69	0.57	0.66	0.50
group3	0.74	0.54	0.66	0.51	0.51	0.51	0.61	0.51	0.55	0.47
kitesurfing	0.91	0.67	0.78	0.60	0.60	0.60	0.31	0.34	0.65	0.23
longboard	0.81	0.60	0.78	0.53	0.49	0.58	0.56	0.50	0.65	0.41
person2	0.99	0.72	0.99	0.72	0.72	0.72	0.92	0.69	0.72	0.67
person4	0.99	0.74	0.98	0.74	0.74	0.74	0.81	0.65	0.71	0.61
person5	1.00	0.81	1.00	0.81	0.81	0.81	1.00	0.81	0.81	0.81
person7	1.00	0.77	0.99	0.76	0.76	0.76	0.91	0.74	0.77	0.71
person14	0.87	0.61	0.83	0.58	0.57	0.59	0.69	0.54	0.61	0.49
person17	0.97	0.69	0.96	0.67	0.67	0.68	0.95	0.68	0.68	0.67
person19	0.97	0.88	0.97	0.86	0.84	0.88	0.78	0.79	0.89	0.71
person20	0.98	0.79	0.98	0.79	0.79	0.79	0.90	0.76	0.80	0.72
rollerman	0.98	0.69	0.94	0.54	0.45	0.67	0.44	0.42	0.66	0.31
skiing	0.89	0.65	0.84	0.56	0.50	0.63	0.67	0.57	0.66	0.50
sup	0.98	0.71	0.91	0.68	0.68	0.68	0.65	0.57	0.71	0.48
tightrope	0.32	0.28	0.27	0.29	0.30	0.28	0.11	0.16	0.50	0.10
warmup	0.99	0.78	0.95	0.73	0.70	0.75	0.81	0.70	0.76	0.64
wingsuit	0.89	0.77	0.68	0.59	0.61	0.58	0.49	0.56	0.83	0.42
yamaha	0.74	0.59	0.60	0.56	0.61	0.52	0.18	0.24	0.69	0.15
	0.86	**0.66**	**0.81**	**0.62**	**0.61**	**0.63**	**0.61**	**0.53**	**0.68**	**0.47**

5. SONUÇLAR

Bu bitirme çalışması kapsamında, uzun süreli videolarda 'insan' nesneleri izlenmek istenmiştir. Bu doğrultuda, nesne sezicinin belirlediği olası konumlar üzerinden veri ilişkilendirme ile hedef konumunu kestirmeyi temel alan derin öğrenme tabanlı yeni bir izleyici çıkarım mimarisi önerildi. Mimari, derin nesne sezici Mask R-CNN ile bu sezicinin önerdiği kişilere ait görüntülerden hedefin ayırt edilmesini sağlayacak konvolüsyonel öznitelikleri çıkaran Piramit ReID ağını içermektedir. Nesne sezicinin aday konumlarının ve gerçek hedefin her biri için ayrı ayrı çıkarılan ReID özniteliklerinin kosinüs benzerliği ile veri ilişkilendirme gerçekleştirilmektedir. Tasarlanan izleyici mimarisi, izleme görevi özelinde ilave bir eğitim gerektirmemesinden dolayı, genel-amaçlı eğitilmiş başka derin nesne sezici ve ReID ağına uyum sağlama esnekliği tanımaktadır.

Önerilen izleyicinin performansı, VOT-LT ve LASOT veri setlerinin 'insan' nesneleri içeren videoları için, başarım oranı (Success Rate), kesinlik (Precision), duyarlılık (Recall) ve harmonik ortalama F1-skoru metrikleri ile raporlandı. Video çerçevelerinde en az 0.5 örtüşme oranının sağlanması koşulu ile, ortalama başarım oranı VOT-LT veri setinde %79.1 ve LASOT veri setinde %61.3 olarak hesaplandı. Örtüşme ortalaması için elde edilen F1-skoru VOT-LT'de %60.7 olup bu skora dayanarak seçili videolarda[1] yapılan güncel nesne izleyiciler (SOTA) ile karşılaştırdığında önerilen izleyicinin performansı, işlem karmaşıklığının düşüklüğü de göz önüne alındığında, umut vericidir. Aynı ölçütün LASOT veri setinde elde edilen skoru ise %51.9'dur.

Bitirme çalışmasında önerilen izleyicinin geliştirilmesi kapsamında ek çalışmalar yapılmıştır.Veri ilişkilendirmenin en yüksek benzerlik skorlu adayla eşleşme kuralına göre yapılması hedefin kamera bakış açısı içerisinde olmadığı durumlarda yanlış kestirime yol açmaktadır. Bu problemin aşılması amacıyla, izleyiciye bir sınıflandırıcı eklenerek aday hedef yerleşimlerine ait özniteliklerin hedef var-yok şeklinde

[1]ballet, bicycle, group1, group2, group3, kitesurfing, longboard, person2, person4, person5, person7, person14, person17, person19, person20, rollerman, skiing, sup, tightrope, wingsuit, yamaha

sınıflandırıldı, bu sayede izleyiciye 'hedef çıktı' kararı verebilme yeteneğinin kazandırılmasının performansa etkisi sınandı. Ancak, aynı videolarda yapılan testlere göre IoU için başarım oranında %20'lik ve F1-skorunda %9'luk bir düşüş gözlemlendi. Sınıflandırıcının genişletilmiş veri setleri ile eğitilmesi çalışmaları devam etmektedir. Önerilen izleyicinin performansının geliştirilmesi için ileriye dönük çalışmalardan ilki hedef çıkışını sezebilen optimum bir karar belirlenmesidir.

Sezme-ile-izleme yötemlerinin performansı nesne sezicinin performansına oldukça bağımlıdır. Bu nedenle bitirme çaışmasında kullanılan Mask R-CNN ağının nesne sezememe oranının düşürülmesi bir diğer devam eden çalışma konusudur. Bu amaçla nesne izleyicideki izleme süreksizliklerinin hedef hareket modeline göre giderilmesi konusunda çalışmalar devam etmektedir.

KAYNAKLAR

[1] **Girshick, R.B.** (2015). Fast R-CNN, *CoRR*, *abs/1504.08083*, http://arxiv.org/abs/1504.08083, 1504.08083.

[2] **Cha, Y.J.**, **Choi, W.**, **Suh, G.**, **Mahmoudkhani, S. ve Buyukozturk, O.** (2017). Autonomous Structural Visual Inspection Using Region-Based Deep Learning for Detecting Multiple Damage Types, *Computer-Aided Civil and Infrastructure Engineering*, *00*, 1–17.

[3] **Zheng, F.**, **Deng, C.**, **Sun, X.**, **Jiang, X.**, **Guo, X.**, **Yu, Z.**, **Huang, F. ve Ji, R.** (2019). Pyramidal Person Re-IDentification via Multi-Loss Dynamic Training, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, s.8514–8522.

[4] **He, K.**, **Gkioxari, G.**, **Dollár, P. ve Girshick, R.B.** (2017). Mask R-CNN, *CoRR*, *abs/1703.06870*, http://arxiv.org/abs/1703.06870, 1703.06870.

[5] **He, K.**, **Zhang, X.**, **Ren, S. ve Sun, J.** (2015). Deep Residual Learning for Image Recognition, *CoRR*, *abs/1512.03385*, http://arxiv.org/abs/1512.03385, 1512.03385.

[6] **Gurkan, F.**, **Cerkezi, L.**, **Cirakman, O. ve Gunsel, B.** (2021). TDIOT: Target-Driven Inference for Deep Video Object Tracking, *IEEE Transactions on Image Processing*, *30*, 7938–7951, http://dx.doi.org/10.1109/TIP.2021.3112010.

[7] **Li, B.**, **Wu, W.**, **Wang, Q.**, **Zhang, F.**, **Xing, J. ve Yan, J.** (2019). SiamRPN++: Evolution of Siamese Visual Tracking With Very Deep Networks, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

[8] **Dai, K.**, **Zhang, Y.**, **Wang, D.**, **Li, J.**, **Lu, H. ve Yang, X.**, (2020), High-Performance Long-Term Tracking with Meta-Updater, 2004.00305.

[9] **Girshick, R.B.**, **Donahue, J.**, **Darrell, T. ve Malik, J.** (2013). Rich feature hierarchies for accurate object detection and semantic segmentation, *CoRR*, *abs/1311.2524*, http://arxiv.org/abs/1311.2524, 1311.2524.

[10] **Uijlings, J.**, **Sande, K.**, **Gevers, T. ve Smeulders, A.** (2013). Selective Search for Object Recognition, *International Journal of Computer Vision*, *104*, 154–171.

[11] **Ren, S.**, **He, K.**, **Girshick, R.B. ve Sun, J.** (2015). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, *CoRR*, *abs/1506.01497*, http://arxiv.org/abs/1506.01497, 1506.01497.

[12] **Kristan, M.**, **Matas, J.**, **Leonardis, A.**, **Vojir, T.**, **Pflugfelder, R.**, **Fernandez, G.**, **Nebehay, G.**, **Porikli, F. ve Čehovin, L.** (2016). A Novel Performance Evaluation Methodology for Single-Target Trackers, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(11), 2137–2155.

[13] **Fan, H.**, **Bai, H.**, **Lin, L.**, **Yang, F.**, **Chu, P.**, **Deng, G.**, **Yu, S.**, **Harshit**, **Huang, M.**, **Liu, J.**, **Xu, Y.**, **Liao, C.**, **Yuan, L. ve Ling, H.**, (2020), LaSOT: A High-quality Large-scale Single Object Tracking Benchmark, 2009.03465.

[14] **Choi, S.**, **Lee, J.**, **Lee, Y. ve Hauptmann, A.**, (2020), Robust Long-Term Object Tracking via Improved Discriminative Model Prediction, 2008.04722.

[15] **Lin, T.Y.**, **Maire, M.**, **Belongie, S.**, **Bourdev, L.**, **Girshick, R.**, **Hays, J.**, **Perona, P.**, **Ramanan, D.**, **Zitnick, C.L. ve Dollár, P.**, (2015), Microsoft COCO: Common Objects in Context, 1405.0312.

[16] **Abadi, M.**, **Agarwal, A.**, **Barham, P.**, **Brevdo, E.**, **Chen, Z.**, **Citro, C.**, **Corrado, G.S.**, **Davis, A.**, **Dean, J.**, **Devin, M.**, **Ghemawat, S.**, **Goodfellow, I.**, **Harp, A.**, **Irving, G.**, **Isard, M.**, **Jia, Y.**, **Jozefowicz, R.**, **Kaiser, L.**, **Kudlur, M.**, **Levenberg, J.**, **Mané, D.**, **Monga, R.**, **Moore, S.**, **Murray, D.**, **Olah, C.**, **Schuster, M.**, **Shlens, J.**, **Steiner, B.**, **Sutskever, I.**, **Talwar, K.**, **Tucker, P.**, **Vanhoucke, V.**, **Vasudevan, V.**, **Viégas, F.**, **Vinyals, O.**, **Warden, P.**, **Wattenberg, M.**, **Wicke, M.**, **Yu, Y. ve Zheng, X.**, (2015), TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, https://www.tensorflow.org/, software available from tensorflow.org.

[17] **Li, S.**, **Liu, X.**, **Liu, W.**, **Ma, H. ve Zhang, H.** (2016). A Discriminative Null Space based Deep Learning Approach for Person Re-Identification.

[18] **Paszke, A.**, **Gross, S.**, **Massa, F.**, **Lerer, A.**, **Bradbury, J.**, **Chanan, G.**, **Killeen, T.**, **Lin, Z.**, **Gimelshein, N.**, **Antiga, L.**, **Desmaison, A.**, **Kopf, A.**, **Yang, E.**, **DeVito, Z.**, **Raison, M.**, **Tejani, A.**, **Chilamkurthy, S.**, **Steiner, B.**, **Fang, L.**, **Bai, J. ve Chintala, S.**, (2019). PyTorch: An Imperative Style, High-Performance Deep Learning Library, Advances in Neural Information Processing Systems 32, Curran Associates, Inc., s.8024–8035.

[19] **Bradski, G.** (2000). The OpenCV Library, *Dr. Dobb's Journal of Software Tools*.

[20] **Harris, C.R.**, **Millman, K.J.**, **van der Walt, S.J.**, **Gommers, R.**, **Virtanen, P.**, **Cournapeau, D.**, **Wieser, E.**, **Taylor, J.**, **Berg, S.**, **Smith, N.J.**, **Kern, R.**, **Picus, M.**, **Hoyer, S.**, **van Kerkwijk, M.H.**,

Brett, M., **Haldane, A.**, **del Río, J.F.**, **Wiebe, M.**, **Peterson, P.**, **Gérard-Marchant, P.**, **Sheppard, K.**, **Reddy, T.**, **Weckesser, W.**, **Abbasi, H.**, **Gohlke, C. ve Oliphant, T.E.** (2020). Array programming with NumPy, *Nature*, *585*(7825), 357–362, https://doi.org/10.1038/s41586-020-2649-2.

[21] **Pedregosa, F.**, **Varoquaux, G.**, **Gramfort, A.**, **Michel, V.**, **Thirion, B.**, **Grisel, O.**, **Blondel, M.**, **Prettenhofer, P.**, **Weiss, R.**, **Dubourg, V.** *ve diğerleri* (2011). Scikit-learn: Machine learning in Python, *Journal of machine learning research*, *12*(Oct), 2825–2830.

[22] **McKinney, W.** *ve diğerleri* (2010). Data structures for statistical computing in python, *Proceedings of the 9th Python in Science Conference*, cilt445, Austin, TX, s.51–56.

[23] **Szegedy, C.**, **Vanhoucke, V.**, **Ioffe, S.**, **Shlens, J. ve Wojna, Z.**, (2015), Rethinking the Inception Architecture for Computer Vision, 1512.00567.

[24] **Lin, T.**, **Goyal, P.**, **Girshick, R.B.**, **He, K. ve Dollár, P.** (2017). Focal Loss for Dense Object Detection, *CoRR*, *abs/1708.02002*, http://arxiv.org/abs/1708.02002, 1708.02002.

[25] **Breiman, L.** (2001). Random Forests, *Machine Learning*, *45*(1), 5–32, http://dx.doi.org/10.1023/A%3A1010933404324.

ÖZGEÇMİŞ

Ad Soyad : Selahaddin HONİ

Doğum Tarihi ve Yeri : 17.12.1997 / İstanbul

E-Posta : honi16@itu.edu.tr

ÖĞRENİM DURUMU

- **Lisans** : 2022, İstanbul Teknik Üniversitesi, Elektrik Elektronik Fakültesi, Elektronik ve Haberleşme Mühendisliği Bölümü

DENEYİM

- 2019'da Baykar Savunma
- 2020'de Havelsan A.Ş
- 2020'de Tübitak Bilgem şirketlerinde yaz stajyeri olarak çalıştı.

S. HONİ	DERİN ÖĞRENME İLE UZUN-SÜRELİ İNSAN İZLEME	2022