

# Derinlikli Nesne Seziciler ve Parçacık Süzgeçleme ile Nesne Takibi

## Object Tracking by Deep Object Detectors and Particle Filtering

Caner Özer Filiz Gürkan Bilge Günsel  
Çoğulortam Sinyal İşleme ve Örüntü Tanıma Grubu  
Elektronik ve Hab. Müh. Bölümü, İstanbul Teknik Üniversitesi, Türkiye  
{ozerc, gurkanf, gunsels}@itu.edu.tr

**Özetçe** —Örtüşme, ani ışıklılık değişimi, bulanıklık ve ölçek değişimlerine karşı gürbüz nesne takip tekniklerinin geliştirilmesine çok sayıda uygulamada gerek duyulmaktadır. Son dönemde derinlikli öğrenme algoritmaları kullanan birçok görsel video nesne takip algoritması önerilmiştir. Bu algoritmalar çoğunlukla görüntülerde nesne tanıma gerçekleştirmekte, video nesne takibinde zamansal bilgiyi kullanmamaktadırlar. Ancak gerçekleşen eğitim aşaması sonrasında, yüksek nesne sezme başarımı sağlamaktadırlar. Parçacık süzgeçleme, probleme uygun durum geçiş ve gözlem modellerinin tanımlanması durumunda, eğitim aşamasına gerek duymaksızın düşük karmaşıklıkla nesne takibine olanak tanıyan alternatif bir yöntemdir. Bu bildiride derinlikli öğrenmeye dayalı nesne seziciler (Faster R-CNN, Mask R-CNN) ve değişken oranlı renk tabanlı parçacık süzgeçleme (VRCPPF) kullanılarak, OTB-50, VOT 2016 ve 2017 veri tabanlarında video nesne takibi performansı raporlanmakta ve her iki yaklaşımın başarılı ve başarısız olduğu durumlar irdelenmektedir. Derinlikli öğrenme ile özellikle ölçek değişimlerine ve örtüşmeye dayanıklılık sağlanabilirken, bulanıklık ve ışıklılık değişimlerinde parçacık süzgeçlemenin daha iyi sonuç verdiği gösterilmektedir. İki yaklaşımın tümleştirilmesi ile nesne yakalama başarımı artırılabilir.

**Anahtar Kelimeler**—Derinlikli öğrenme, parçacık süzgeçleme.

**Abstract**—Developing object tracking techniques robust to blur, scale changes, occlusion and illumination changes is a challenging problem for several applications. Recently many algorithms using deep learning for visual object tracking are proposed. These algorithms mostly perform object detection without using the temporal information for video object tracking. Nevertheless, they provide high object detection accuracy as a result of an extended training scheme. However, particle filtering enables us to track the objects with a lower complexity without requiring any training, when the state transition and observation models are formulated appropriately. In this paper, tracking performance of two visual object trackers (Faster R-CNN, Mask R-CNN) and the variable rate color based particle filtering are tested on OTB-50, VOT 2016 and 2017 datasets. Benefits and deficits of both these approaches are examined. It is concluded that the deep learning methods outperform particle filtering under occlusion and scale changes, whereas particle filtering is more robust to illumination changes and blur. Integration of both approaches improves object tracking accuracy.

**Keywords**—Deep learning, particle filtering.

### I. GİRİŞ

Videoda nesne yakalama ve izleme, nesnelerin ardarda gelen video karelerinde konumlarının ve hareketlerinin kestirilmesi ile gerçekleştirilmektedir. Videoda nesne yakalama sistemlerine, tıbbi görüntü analizinden, mobil uygulamalara, hareket ve aktivite tanımadan, robotik uygulamalarına kadar birçok önemli uygulamada gerek duyulmaktadır. Nesne izlemede ortamdaki ışıklılık değişimleri, ölçek değişimleri, örtüşme ve hızlı nesne hareketlerinden kaynaklanan bulanıklık, nesne sezme ve izleme başarımını düşüren önemli etkenlerdir. Uzun yıllardır yapılan çalışmalar, nesne izleme başarımını artırmanın yanı sıra karşılaşılan bu problemlere gürbüz algoritmalar geliştirmeyi amaçlamaktadır.

Günümüzde, video nesne takibi ve nesne yakalama amacıyla birçok derinlikli öğrenme yöntemi kullanılmaktadır [1], [2], [3], [4], [5], [6]. Önerilen derinlikli öğrenme algoritmaları, nesne takibinde oldukça yüksek başarımlar sağlasa da, çoğunlukla zamansal bilgi kullanmadan ayrı video çerçevelerinde nesne tanıma gerçekleştirmektedir. Bu nedenle, özellikle bulanık video çerçevelerinde hedef nesne seziemekte ve nesne hareketi izlemede kopukluklar ortaya çıkmaktadır. Ayrıca derinlikli öğrenme yöntemlerinin takip edilmek istenilen nesneyi ön eğitim aşaması ile öğrenme gerekliliği öğrenilmemiş nesnelerin izlenememesine neden olmaktadır. Bir Bayesçi süzgeçleme yöntemi olan parçacık süzgecinde ise, ön öğrenme işlemine gerek olmadan ilk video çerçevesinde konumu belirlenen bir nesnenin takibi yapılabilir. Parçacık süzgeçlemede kullanılan hareket ve gözlem modelleri sayesinde düşük bir karmaşıklık ile zamansal bilginin kullanıldığı bir nesne takip yöntemi sunulmaktadır [7], [8], [9]. Parçacık süzgeçlerinin temel eksikliği, nesnenin video çerçevesinde bulunduğu varsayımı ile izlemeyi sürdürmeleri ve bu nedenle örtüşme, sahne giriş/çıkışı olması durumlarında başarımlarının düşmesidir.

Bu bildiri kapsamında, son dönemde nesne yakalamada yüksek başarımlar sağlayan Mask R-CNN [6] ve Faster R-CNN [2] yöntemleri ile değişken oranlı renk tabanlı parçacık süzgecinin (Variable Rate Color Based Particle Filter - VRCPPF) [9] video nesne takip başarımları, OTB50 [1], VOT 2016 ve 2017 [10] veri tabanları üzerinde yapılan testler ile karşılaştırılarak, her bir yöntemin olumlu ve olumsuz yanlarının belirlenmesi amaçlanmıştır. Yapılan testler sonucunda, derinlikli öğrenme yöntemleri ölçek değişimi ve örtüşme problemlerine karşı

daha dayanıklı iken, VRCPPF'in bulanıklık ve ışıklılık değişimi durumlarında daha başarılı olduğu gözlemlenmiştir. Mask R-CNN, VOT veri tabanlarında Faster R-CNN'e göre nesne izleme başarımını ortalama %15 artırmış, OTB veri tabanında başarımları yakındır. VRCPPF in başarımı ortalama düşük olmakla birlikte, bulanıklık ve yüksek ışıklılık değişimi olan videolarda oldukça yüksektir. Bu videolarda derinlikli öğrenmeye dayalı seziciler nesneyi yakalayamamaktadır. Derinlikli nesne seziciler ile paçacık süzgeçlemenin tümleştirilmesinin nesne izleme başarımını artırdığı gözlemlenmiştir [11].

## II. DERİNLİKLİ NESNE TAKİPÇİLERİ

Bu bölümde, Faster R-CNN [2] ve Mask R-CNN [6] derinlikli öğrenme yöntemlerinden bahsedilmektedir.

### A. Faster R-CNN

Faster R-CNN, Fast R-CNN ve olası nesne bölgeleri üreten bölge öneri ağı (region proposal network-RPN) mimarilerinin birleştirilmesi ile oluşan bir nesne sezme yöntemidir. Bölge öneri ağına amaç, olası nesne bölgelerini belirlemek ve Fast R-CNN'in bu olası nesne bölgelerinde işlem yapması ile nesne konumunun daha doğru bir şekilde belirlenmesini sağlamaktır. RPN olası nesne konumlarını belirleyen koordinatları ve bunların nesnelilik skorlarını çıktı olarak verir ve bütünüyle konvolüsyonel katmanlardan oluşmaktadır. Fast R-CNN ise RPN'in ürettiği bölgeleri alarak sınıflandırma ve konumu çevreleyen bölge tahmini (bounding box - BB) belirleme işlemlerini gerçekleştirmektedir. Böylelikle nesne yakalama başarımı daha düşük işlemsel karmaşıklıkla artırılmıştır. Bu çalışmada Faster R-CNN için, [13] de verilen kod geliştirilerek yeni nesne sınıfları eklenmiş ve kullanılmıştır.

### B. Mask R-CNN

Mask R-CNN nesne sezmeye örnek seviyesinde bölütleme (instance segmentation) yaklaşımı getirilerek gerçekleştirilmiştir. Temel olarak Faster R-CNN mimarisine yapılan eklemelerle başarımların artırımı sağlanmıştır.

Mask R-CNN eğitim aşamasında her bir video çerçevesini iki aşamada işler. İlk aşamada CNN yapısı ile çerçeve öznitelik haritaları çıkarılır. Bu çalışmada CNN olarak ResNet 101 [4] ile Öznitelik Piramit Ağları [5] yapıları kullanılmıştır [14]. Mask R-CNN de Faster R-CNN e benzer şekilde video çerçevelerini farklı boyutlarda ankor pencereleri ile filtreleyerek öznitelik çıkarımı gerçekler. Ancak, çokölçekli aramayı iyileştirmek ve lokalizasyon doğruluğunu arttırmak amacıyla, Faster R-CNN'de 3 farklı ölçek ve 3 farklı en-boy oranı ile 9 olan ankor sayısı, Mask R-CNN'de 5 farklı ölçek ve 3 farklı en-boy oranı ile 15 adete çıkartılmıştır. Öznitelik haritaları RPN den geçirilerek olası nesnelere ait bölgeleri belirleyen BB önerileri nesne olabilirlik skorları ile birlikte üretilir. En yüksek olasılıklı sınırlı sayıda (bu çalışmada 100) olası nesne bölgesi RoIAlign katmanı girişine aktarılır. Mask R-CNN de, örnek seviyesinde bölütleme hedeflendiğinden, Faster R-CNN'de kullanılan RoIPool katmanı yerine RoIAlign katmanı önerilmiştir. İkinci aşamada her bir olası nesne bölgesinin öznitelik haritası belirlenip, RoIAlign katmanı ile hizalandıktan sonra [3], Faster R-CNN yöntemine benzer bir sınıflandırıcı katman yardımı ile nesne tanıma gerçekleştirilir. Nesne etiketlemeye paralel olarak sezilen nesneye ait bölütlenmiş imge ya da maske de elde edilir.

Mask R-CNN eğitiminde ikinci aşamada (1)'de formüle edilen maliyet fonksiyonunu enküçükleyerek nesne sezimini gerçekler. Maliyet fonksiyonundaki ilk iki terim Faster R-CNN ile aynıdır. Üçüncü terim eniyilemeye maske bölütlemenin katkısını ekleyen yeni terimdir.

$$L = L_{snf} + L_{BB} + L_{maske} \quad (1)$$

Maliyet fonksiyonundaki sınıflandırma hatalarına karşı düşen birinci terim (2)'de açık olarak yazılmıştır. (2)'de  $N$  her bir iterasyonda eğitim ağına gösterilen nesne içeren BB sayısını,  $t^c$  sınıflandırıcı tarafından öngörülen sınıf etiketlerini,  $t^e$  bu öngörüye ilişkin gerçek sınıf etiketini göstermektedir. Bu nedenle  $L_{snf}$  öğrenme iterasyonlarında sınıflandırma hatalarına karşı düşen maliyeti modellemektedir [2].

$$L_{snf} = \frac{1}{N} \sum_{i=1}^N t_i^c \log(y_i^c) \quad (2)$$

(1)'deki ikinci terim (3)'te formüle edilmiştir ve öğrenilen nesneleri çevreleyen BB lerdeki lokalizasyon hatalarını modellemektedir [2].

$$L_{BB} = \frac{1}{N} \sum_{i=1}^N \text{smooth}_{L_1}(y_i^b - t_i^b) \quad (3)$$

(3)'te  $y^b$  ve  $t^b$  sırasıyla hedef nesneye ilişkin tahmin edilen ve gerçek BB vektörlerine karşı düşmektedir.

Maske bölütleme hataları (4)'te, öğrenilen nesneye ilişkin tahmin edilen ve gerçek bölütleme maskelerinin ikili çapraz entropisi şeklinde modellenmektedir.

$$L_{maske} = \frac{1}{N} \sum_{i=1}^N t_i^m \cdot \log(y_i^m) \quad (4)$$

(4)'te  $(\cdot)$  noktasal çarpım işlemini,  $y^m$  tahmin edilen maskeyi,  $t^m$  ise gerçek bölütleme maskesini göstermektedir.  $y_m$  tahminleri öncelikle Sigmoid aktivasyon fonksiyonundan geçirilerek  $[0, 1]$  aralığına ölçeklenir.

## III. DEĞİŞKEN ORANLI RENK TABANLI PARÇACIK SÜZGEÇİ

Parçacık süzgeci, doğrusal ve Gauss dağılımlı olmayan kestirim problemlerinde kullanılan bir yöntemdir. Parçacık süzgecinde temel fikir, sonsal olasılık dağılımına örneklenmiş ağırlıklı parçacıklar üzerinden yakınsamaktır. Bu çalışmada değişken oranlı parçacık süzgeci (Variable Rate Particle Filter-VRPF) [7] ve renk tabanlı parçacık süzgecinin (color based particle filter-CPF) [8] birlikte kullanıldığı değişken oranlı renk tabanlı parçacık süzgeci (VRCPPF) [9] kullanılarak nesne takibi yapılmakta ve takip başarımı raporlanmaktadır.

Klasik parçacık süzgecinde,  $t$ . çerçeveye ait durum vektörü  $s_t$ , nesne konumunu çevreleyen dikdörtgen bölgeleri (BB) belirtecek şekilde  $s_t = [x, e, \dot{x}, \dot{e}, H_x, H_e, \alpha]$  olarak tanımlanır. Burada BB merkez koordinatları  $(x, e)$ , bölgenin hareket hızları  $(\dot{x}, \dot{e})$ , bölgenin yarı genişliği ve yüksekliği  $(H_x, H_e)$  ve ölçek faktörü  $\alpha$  ile gösterilmektedir. Parçacık filtresinde izlenen nesnenin  $t$ . çerçevedeki konumundan  $(t+1)$ . çerçevedeki konumuna geçiş modeli  $i$ . parçacık için  $s_{t+1}^i = A s_t^i + \varphi_t$  şeklinde ifade edilebilir. Bu gösterimde  $i = 1, 2, \dots, N$  örneklenen

parçacık sayısını,  $A$  durum geçiş matrisini ve  $\varphi_t$  gürültü sürecini modellemektedir.

Değişken oranlı parçacık filtresinde (VRPF), klasik parçacık filtresinden farklı olarak, durum güncellemeleri her gözlem için yapılmaz.  $k$  ayrık durum indisi ve  $\tau_k$  durum geliş anlarını gösteren değişken olmak üzere, iki durum güncelleme anı arasındaki  $\tau_{k+1} - \tau_k$  zaman farkının Gamma dağılımına uyduğu varsayılır. VRPF’de amaçlanan manevra yapılan aralıklarda daha çok durum güncellemesi, düşük manevralı bölgelerde ise daha az durum güncellemesi yapılmasını sağlamaktır. VRPF durum vektörü  $x_k = (s_k, \tau_k)$  olarak gösterilir [9]. VRPF’te gözlem ve durum güncellemeleri senkron olmadıklarından gözlem modeli, gözlem noktası çevresindeki yerel durumların oluşturduğu " $N_t$ " komşuluk kümesine bağlanarak,  $p(y_k | s_{N_t})$  şeklinde ifade edilir. Burada  $s_{N_t} = \{s_k; k \in N_t(1 : H)\}$ ,  $H$  komşuluk sayısı ve  $y_k$   $k$  anında gözlemlenen video çerçevesidir. Bu noktadan sonra gösterim karmaşıklığının önlenmesi amacıyla zaman indisi olarak  $t$  kullanılacaktır.

Bu çalışmada VRPF ile [8]’de önerilen renk gözlem modelini birleştiren değişken oranlı renk tabanlı parçacık filtresi (VRCPF) kullanılmaktadır.  $s_t^i$ ,  $t$  anındaki  $i$ . parçacık durum vektörü ve  $q(s_t^i)$  ilgili durum vektörünün gösterdiği BB’ye ait renk histogramı olmak üzere gözlem modeli eşitlik (5)’tedir.

$$p(y_t | s_t^i = s_{N_t}) \propto e^{-\lambda(d^2(q(s_t^{hed}), q(s_t^i)))} \quad (5)$$

Eşitlik (5)’te  $s_t^{hed}$  ilk çerçevede belirlenen hedef nesne konumuna ait durum vektörünü,  $d[\cdot]$  iki histogram arasındaki Bhattacharya uzaklığını ve  $\lambda$  yumuşatma katsayısını ifade etmektedir [9]. Her bir parçacığa ait ağırlıklar gözlem modeli kullanılarak eşitlik (6) ile güncellenir.

$$w_t^i \propto w_{t-1}^i p(y_t | s_t^i) \quad (6)$$

VRCPF de izlenen nesnenin konumu, her bir çerçevede  $N$  adet parçacığın ağırlıklı ortalaması kullanılarak (7) ile hesaplanır.

$$E[s_t^i] \approx \sum_{i=1}^N w_t^i s_t^i \quad (7)$$

(5) ve (6) incelendiğinde, hedef nesneye en benzer BB lara sahip parçacıkların izlenen nesnenin yerini belirlemede daha fazla söz sahibi olduğu görülmektedir.

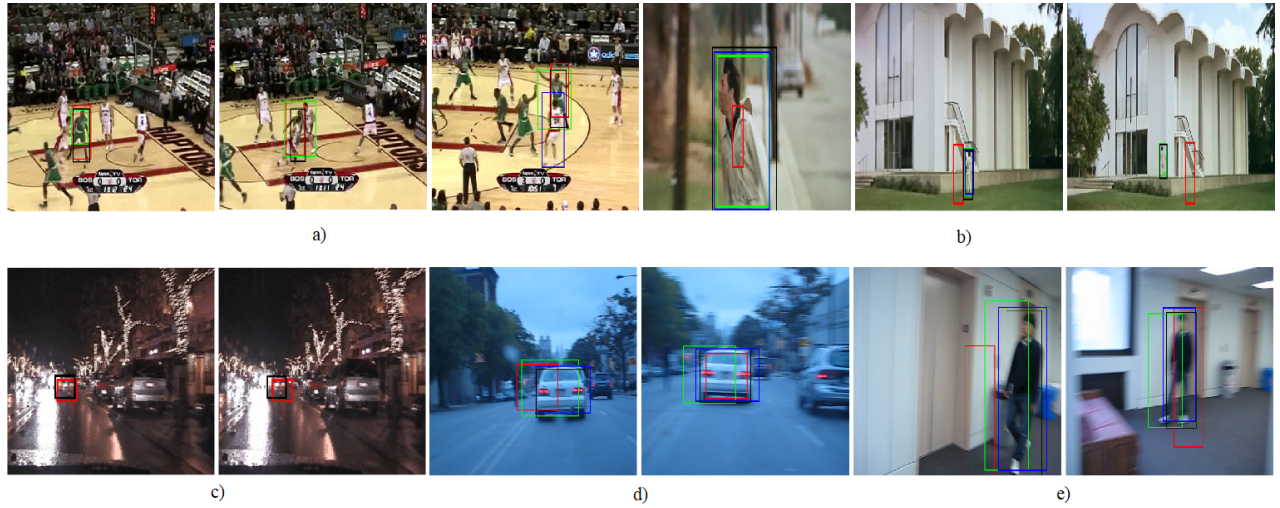
#### IV. PERFORMANS TESTLERİ

Derinlikli nesne takipçileri ve parçacık süzgeci başarımları, nesne takip performans ölçümü için sıklıkla kullanılan ve 50 videodan oluşan TB50 (Object Tracker Benchmark) [1] ve 60’şar videodan oluşan VOT (Visual Object Tracker Benchmark) 2016/2017 [10] veri setleri üzerinde raporlanmaktadır. Derinlikli öğrenme algoritmaları, Microsoft COCO [12] veri tabanında yer alan 80 nesne sınıfına ait görüntüler ile eğitilmiş ve performanslar, ilgili eğitim setinde yer alan nesneleri içeren VOT 2016/2017 ve OTB50 veri tabanlarında bulunan sırasıyla 37, 30 ve 22 video üzerinde raporlanmıştır. Teknik yazında nesne takip başarımları, nesneye ilişkin gerçek BB ile kestirilen BB kesişimlerinin birleşimlerine oranı olarak tanımlanan IoU büyüklüğüne bağlı raporlanmaktadır. IoU, belirli bir eşik değerinden büyük ise ilgili çerçevede nesne başarılı olarak takip edilmiştir denilir ve başarımlar, nesne takibinin başarılı olduğu çerçevelerin tüm çerçevelere oranı ile raporlanır.

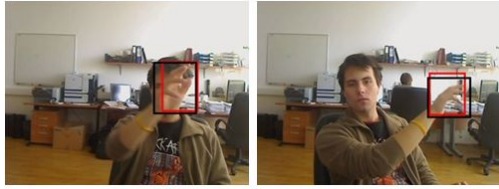
Tablo I’ de, Mask R-CNN, Faster R-CNN ve VRCPF yöntemleri için, ilgili veri tabanlarında, farklı eşik değerleri (IoU th) için başarımlar raporlanmaktadır. Derinlikli öğrenmeye dayalı nesne takipçileri, VRCPF yöntemi ile karşılaştırıldığında bütün veri tabanları için oldukça yüksek başarımlar göstermektedir. VOT veri tabanlarında 0.5 eşik değeri için Mask R-CNN, Faster R-CNN’e göre yaklaşık %15 daha yüksek başarımlar elde ederken, OTB50 veri tabanında başarımları benzerdir. Mask R-CNN’in VOT veri tabanlarındaki başarımları, ilgili veri tabanlarında Faster R-CNN ile nesne sezilemeyen çerçeve sayısının (%32) Mask R-CNN’e göre (%13) yaklaşık %19 daha fazla olmasından kaynaklanmaktadır. Bu fark OTB veri tabanı için %7 olarak hesaplanmıştır. Öte yandan VRCPF ile nesne takip başarımları derinlikli öğrenme yöntemlerine göre daha düşüktür. Ancak VRCPF, Mask R-CNN ve Faster R-CNN den farklı olarak ayrık video çerçevelerinde işlem yapmaz, zamansal video bilgisini de kullanarak belirlenen nesneyi takip eder. Bu nedenle bulanıklık ve ışıklılık değişimi nedeniyle derinlikli sezicilerin nesneyi sezemediği video karelerinde, durum geçiş ve gözlem modellerine bağlı olarak, nesneyi izlemeye devam edebilir. Örnek olarak, Şekil 1’de hand videosa ait video çerçevelerinde derinlikli öğrenme ile nesne konumu kestirilememekte ancak, sırası ile 78. ve 109. çerçevelerde hedef nesne modelinin güncellenmesi ile VRCPF takibe yüksek lokalizasyon başarımları ile devam edebilmektedir. Şekil 2’de her bir yöntem ile elde edilen nesne BB’larına örnekler görülmektedir. Şekil 2(a)’da örtüşme ve küçük nesnenin yer aldığı video çerçevelerinde, Faster R-CNN ile nesne konumunun kestirilemediği ve VRCPF’in Mask R-CNN’den daha doğru bir lokalizasyon ile nesne konum tahmini yaptığı görülmektedir. Şekil 2(b)’de görünüm değişikliklerinden kaynaklı olarak VRCPF nesneyi yakalayamamakta ancak derinlikli seziciler ile nesne doğru olarak takip edilebilmektedir. Son çerçevede Mask R-CNN ile, nesnenin çok küçülmesinden kaynaklı sorunlara karşı daha gürbüz bir izleme gerçekleştirildiği görülmektedir. Şekil 2(c)’de, Mask R-CNN ve Faster R-CNN yöntemleri çözünürlüğün yetersiz kalmasından dolayı, nesne konumunu sezememekte, VRCPF ise hareket modeli sayesinde nesne takibine devam edebilmektedir. Şekil 2(d) ve 2(e)’de ise OTB veri tabanına ait örnekler görülmektedir. Her iki video dizisinde de, Faster R-CNN yöntemi daha doğru bir lokalizasyon ile nesne konumunu belirlerken, Mask R-CNN daha düşük bir başarımlar göstermektedir. Yapılan testlerde VRCPF algoritması için ilk video çerçevesinde takip edilmek istenilen nesnenin gerçek konumu algoritmaya verilmektedir. Ancak uzun videolarda nesne modeli zamanla değiştiği için (ışıklılık, deformasyon vb) bu model yetersiz kalmakta ve nesne takibi bu yöntemle doğru olarak devam edememektedir. Derinlikli öğrenme yöntemlerinin VOT2016 yarışmasında en iyi performansı gösteren ilk 5 yöntem ile karşılaştırılması Şekil 3’de görülmektedir. Bu çalışmada en yüksek başarımları sağlayan derinlikli seziciler olması nedeniyle Mask R-CNN ve Faster R-CNN sonuçları detaylı olarak raporlanmaktadır.

#### V. SONUÇLAR

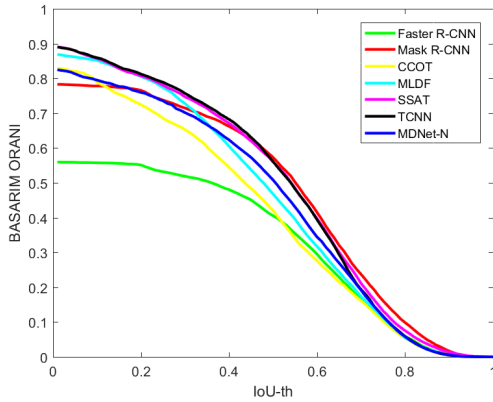
Bu çalışmada, son yıllarda popülerliği artan derinlikli görsel nesne takipçileri ile VRCPF nesne takipçisi karşılaştırılarak, her bir yöntemin olumlu ve olumsuz yanları irdelenmiştir. Parçacık filtrelemenin kullandığı basit durum geçiş modeli ve gözlem modeli ile özellikle bulanık video karelerinde kesintisiz nesne takibine olanak sağladığı görülmüştür. Öte



Şekil 2: Farklı video dizileri için nesne takip sonuçları (a) Basketball (çerçeve no: 6,17,623), (b)Graduate (çerçeve no: 526,612,655), (c) Car2 (çerçeve no: 5,6), (d) BlurCar2(çerçeve no: 99,141), (e) BlurBody2 (çerçeve no: 53,160). VRCPPF(kırmızı), Mask R-CNN(yeşil), Faster R-CNN(mavi), GT(siyah).



Şekil 1: Hand videosu 80 ve 111. çerçeveler. VRCPPF (kırmızı), Faster R-CNN (yeşil) GT (siyah).



Şekil 3: Güncel yöntemlerin nesne izleme başarımlarının VOT2016 videolarında IoU ya bağlı değişimi.

yandan derinlikli öğrenme kullanan nesne sezicileri, nesne görünümünün değişiklikleri, ölçek değişimleri ve örtüşmeye karşı nesne yakalama başarımını arttırmaktadır. Her bir yöntemin olumlu yönlerinden yararlanmak amacıyla iki yöntemin birlikte kullanımı, nesne takip başarımının artırılması ve farklı etmenlere karşı daha gürbüz bir algoritmanın oluşturulması açısından önemlidir.

#### KAYNAKLAR

[1] Yi Wu, Jongwoo Lim, Ming Yang, "Online object tracking: A benchmark", *In Proc. CVPR*, pp.2411-2418, 2013.

TABLO I: Farklı veri tabanlarında nesne izleme başarımları

Dataset	Metot	IoU th			
		0.4	0.5	0.6	0.7
VOT 2016	Mask R-CNN	0.662	0.573	0.415	0.237
	Faster R-CNN	0.480	0.405	0.300	0.163
	VRCPPF	0.080	0.048	0.030	0.018
VOT 2017	Mask R-CNN	0.667	0.566	0.406	0.229
	Faster R-CNN	0.495	0.411	0.296	0.161
	VRCPPF	0.095	0.056	0.034	0.021
OTB50	Mask R-CNN	0.738	0.619	0.467	0.293
	Faster R-CNN	0.729	0.644	0.518	0.359
	VRCPPF	0.209	0.150	0.096	0.048

- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks", *In Proc. NIPS*, pp.91-99, 2015.
- [3] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks", *In Proc. NIPS*, pp.2017-2025, 2015
- [4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition", *In Proc.*, pp. 770-778, 2016.
- [5] T. Y. Lin, P. Dollar, R. Girshick, et.all, "Feature pyramid networks for object detection", *In Proc. CVPR*, pp. 936-944, 2017.
- [6] K. He, G. Gkioxari, P. Dollár, R. Girshick, "Mask R-CNN", *In Proc. ICCV*, pp.2961-2969, 2017.
- [7] S. Godsill and J. Vermaak, "Variable rate particle filters for tracking applications", *In Proc. Statist. Signal Processing*, pp. 1280-1285, 2005.
- [8] K. Nummiaro, E. Koller-Meier, and L. J. V. Gool, "An adaptive color-based particle filter", *Image and Visual Computing*, vol.21, no.1, pp.99-110, 2003.
- [9] D. Kumlu and B. Günsel, "Variable rate adaptive color-based particle filter tracking", *In Proc. ICIP*, pp. 1679-1683, 2016.
- [10] www.votchallenge.net, 'Visual Object Tracking', 2017. URL: <http://www.votchallenge.net/challenges.html> (Erişim Tarihi: 29.03.2018).
- [11] B. Akok, F. Gurkan, O. Kaplan and B. Günsel, "Robust object tracking by interleaving variable rate color particle filtering and deep learning", *in Proc. IEEE ICIP*, pp.3665-3669, 2017.
- [12] T.-Y. Lin, M. Maire, S. Belongie, et. al., "Microsoft COCO: Common objects in context", *In Proc. ECCV*, pp.740-755, 2014.
- [13] Faster R-CNN: <https://github.com/rbgirshick/py-faster-rcnn>
- [14] Mask R-CNN: [https://github.com/matterport/Mask\\_RCNN](https://github.com/matterport/Mask_RCNN)