T.C.
İNÖNÜ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

GELENEKSEL VE DERİN ÖĞRENME TABANLI NESNE TAKİP YÖNTEMLERİNİN PERFORMANS DEĞERLENDİRMESİ

HÜSEYİN ÜZEN

YÜKSEK LİSANS TEZİ

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

HAZİRAN 2018

Tezin Başlığı : Geleneksel ve Derin Öğrenme Tabanlı Nesne Takip Yöntemlerinin Performans Değerlendirmesi

Tezi Hazırlayan : Hüseyin ÜZEN

Sınav Tarihi : 29.06.2018

Yukarıda adı geçen tez jürimizce değerlendirilerek Bilgisayar Mühendisliği Ana Bilim Dalında Yüksek Lisans Tezi olarak kabul edilmiştir.

Sınav Jüri Üyeleri

Tez Danışmanı **: Doç. Dr. Muhammed Fatih TALU** .....
İnönü Üniversitesi

Doç. Dr. Davut HANBAY .....
İnönü Üniversitesi

Dr. Öğr. Üyesi Nuh ALPASLAN .....
Bingöl Üniversitesi

Prof. Dr. Halil İbrahim ADIGÜZEL
Enstitü Müdürü

ONUR SÖZÜ

Yüksek Lisans Tezi olarak sunduğum "Geleneksel ve Derin Öğrenme Tabanlı Nesne Takip Yöntemlerinin Performans Değerlendirmesi" başlıklı bu çalışmanın bilimsel ahlak ve geleneklere aykırı düşecek bir yardıma başvurmaksızın tarafımdan yazıldığını ve yararlandığım bütün kaynakların, hem metin içinde hem de kaynakçada yöntemine uygun biçimde gösterilenlerden oluştuğunu belirtir, bunu onurumla doğrularım.

Hüseyin ÜZEN

ÖZET

Yüksek Lisans Tezi

GELENEKSEL VE DERİN ÖĞRENME TABANLI NESNE TAKİP YÖNTEMLERİNİN PERFORMANS DEĞERLENDİRMESİ

Hüseyin ÜZEN

İnönü Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

85 + ix sayfa

2018

Danışman: Doç. Dr. Muhammed Fatih TALU

İkinci Danışman: Dr. Öğr. Üyesi Kazım HANBAY

Nesne takibi bir video veya gerçek zamanlı kayıt ortamında hedef nesnenin merkez konumu ve boyutunun otomatik olarak elde edilmesidir. Gelişen teknoloji ile birlikte, farklı uygulama alanlarına yönelik nesne takip yöntemleri geliştirilmektedir. Ancak, literatürdeki bu yöntemlerin birtakım testler yardımıyla avantajlarının ve dezavantajlarının ortaya konulması gerekmektedir. Bu sayede, günümüz problemlerine uygun çözümler geliştirilebilecektir.

Bu tez çalışmasında gerçekleştirilen kıyaslamalı testler ile güncel nesne takip yöntemleri değerlendirilmiştir. Ayrıca, destek vektör makinesi gibi geleneksel makine öğrenmesi yöntemlerinin yanında, günümüzde oldukça popüler olan derin öğrenme tabanlı yöntemler de ele alınmıştır. Yapılan tez çalışmasında öncelikle literatürdeki güncel yöntemler incelenmiştir. Daha sonra uygulamalı karşılaştırmalar yapmak üzere beş geleneksel ve beş derin öğrenme tabanlı olmak üzere on nesne takip yöntem seçilmiştir. Bu yöntemlerin avantaj ve dezavantajlarını ortaya koymak amacıyla birbirinden farklı birçok veri setinden videolar kullanılmıştır. Ayrıca, incelenen yöntemler tez çalışması sırasında oluşturulan özel veri seti ile test edilmiştir. Deneysel sonuçlar, üç ana başlık altında (genel, problem bazlı ve performans bazlı değerlendirme) detaylı bir şekilde incelenmiştir.

Yapılan kıyaslamalı testler sonucunda derin öğrenme tabanlı yaklaşımların oldukça yüksek nesne takip performansı elde ettiği görülmüştür. Bunun yanında korelasyon filtresi kullanan yöntemler performans açısından günümüzde hala vazgeçilemeyen yöntemler arasında yer almaktadırlar. Ayrıca korelasyon filtresinin nesne tespitinde geleneksel ikili sınıflandırıcılara göre çok daha başarılı olduğu gözlemlenmiştir. Son olarak yapılan testlerde derin öğrenme tabanlı ADNet yönteminin sahip olduğu güçlü ağ mimarisi sayesinde gelecekteki birçok çalışmaya yön verebileceği sonucuna varılmıştır.

ANAHTAR KELİMELER: Nesne Takibi, İnsan Takibi, Arka Plan Çıkarma, Nesne Sınıflandırma, Derin Öğrenme

ABSTRACT

Master Thesis

EVALUATING OBJECT TRACKING PERFORMANCE OF CONVENTIONAL AND DEEP LEARNING BASED METHODS

Hüseyin ÜZEN

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

85 + ix pages

2018

Supervisor: Assoc. Prof. Dr. Muhammed Fatih TALU

Co-Supervisor: Asst. Prof. Dr. Kazım HANBAY

Object tracking is automatic obtain of the center position and size of target object in a video or real-time recording environment. Along with the developing technology, object tracking methods for different application areas are being developed. The advantages and disadvantages of these state-of-art methods need to present with the help of several tests. Thus, novel solutions can be developed for today's problems.

In this thesis, state-of-art object tracking methods were evaluated with the carried out comparative tests. In addition to traditional machine learning methods such as support vector machines, deep learning-based methods, which are quite popular today, are also discussed. In this thesis, firstly the state-of-art methods are examined. Then, ten object tracking methods, which are five traditional and five deep learning based, were selected for applied comparisons. In order to demonstrate the advantages and disadvantages of these methods, videos from many different datasets have been used. In addition, the examined methods were tested with a special dataset which is created during the thesis study. Experimental results have been examined in detail under three main topics (general, problem-based and performance-based evaluation).

As a result of the comparative tests, it is seen that the deep learning-based approaches achieve high tracking performance. Besides, methods utilizing correlation filter are still the indispensable methods in terms of performance. It has also been observed that the correlation filter is much more successful in detecting objects than traditional binary classifiers. Finally, it has been concluded with the carried out comparative tests that thanks to the powerful network architecture of deep learning-based ADNet method, it can lead to many future works.

KEYWORDS: Object Tracking, Human Tracking, Background Subtraction, Object Classification, Deep Learning

TEŞEKKÜR

Tez çalışmamın her aşamasında bilgi ve tecrübelerinin yanı sıra ilgi ve teşvikleriyle de bana destek olan danışman hocam Sayın Doç. Dr. Muhammed Fatih TALU ve ikinci danışman hocam Sayın Dr. Öğr. Üyesi Kazım HANBAY'a;

Çalışmalarım boyunca, bana destek olan İnönü Üniversitesi Bilgisayar Mühendisliği Bölümündeki tüm değerli hocalarıma;

Görev yaptığım üniversitede bulunduğum birimdeki iş yoğunluğuna rağmen yüksek lisans tez çalışması yapmamda bana maddi ve manevi her türlü destek veren tüm akademisyen arkadaşlarıma;

Ayrıca yüksek lisans tez çalışmalarım süresince bana karşı sabırla ve anlayışla yaklaşan ve hep yanımda olan sevgili eşim Bahar ÜZEN'e, aileme ve tüm arkadaşlarıma

teşekkür ederim.

İÇİNDEKİLER

SİMGELER VE KISALTMALAR

$\varphi(.)$	Derin öznitelik çıkartma fonksiyonu
θ_d	Duyarlılık eşik değeri
θ_b	Doğruluk eşik değeri
ADNet	Eylem Karar Ağları - Action Decision Network
BACF	Arka Plana Yönelik Korelasyon Filtre Öğrenme - Learning Background Aware Correlation Filters
BIT	Biyolojik Esinlenmiş Takipçi - Biologically Inspired Tracker
HCFT	Görsel İzleme için Hiyerarşik Konvolüsyon Özellikleri - Hierarchical Convolutional Features for Visual Tracking
DCFNet	Görsel İzleme için Diskriminant Korelasyon Filtre Ağı - Discriminant Correlation Filters Network For Visual Tracking
DFT	Ayrık Fourier Dönüşümünü - Discrete Fourier Transform
DVM	Destek Vektör Makineleri
EKF	Entegre Korelasyon Filtresi
FFT	Hızlı Fourier Dönüşümü - Fast Fourier Transform
GOTURN	Regresyon Ağlarını Kullanarak Genel Nesne Takibi - Generic Object Tracking Using Regression Networks
IFFT	Ters Hızlı Fourier Dönüşümü - Inverse Fast Fourier Transform
KCF	Çekirdekleştirilmiş İlinti Süzgeci - Kernelized Correlation Filter
KNN	En Yakın K Komşu Sınıflandırıcısı
MEEM	Azaltma Kullanarak Birden Çok Uzman Vasıtasıyla Gürbüz Takip - Robust Tracking via Multiple Experts Using Entropy Minimization
MUSTer	Çoklu Hafıza Takipçisi - Multi Store Tracker
OPE	Tek Geçişli Değerlendirme - One-Pass Evaluation
SGD	Olasılıksal Dereceli Azalma - Stochastic gradient descent
SINT	Nesne Takibi için Benzer Örnekleri Arama - Siamese Instance Search for Tracking
SiamFC	Nesne Takibi için Tam Bağlı Siyam Ağı - Fully-Convolutional Siamese Networks for Object Tracking
SRE	Konumsal Gürbüzlük Değerlendirmesi - Spatial Robustness Evaluation
TRE	Zamansal Gürbüzlük Değerlendirmesi - Temporal Robustness Evaluation
YGH	Yönlü Gradyan Histogramı
YİO	Yerel İkili Örüntü

ŞEKİLLER LİSTESİ

ÇİZELGELER LİSTESİ

1. GİRİŞ

Görüntü işleme, ölçülmüş veya kaydedilmiş olan elektronik ortamdaki görüntü verilerini işleyerek anlamlı bilgilerin çıkarılmasını sağlayan yöntemler topluluğudur. Günümüzde mühendislik ve bilgisayar bilimlerinin temel araştırma alanı içinde yer almaktadır. Görüntü işleme teknikleri ile ham görüntü içerisindeki verilerin analizi, nesnenin tespiti, sınıflandırılması veya takibi gibi birçok çıkarımlar yapılabilmektedir. Teknolojideki gelişmelere paralel olarak savunma, sağlık, endüstriyel, ulaşım gibi birçok alanda aktif olarak kullanılan görüntü işleme yöntemleri, canlılardaki görme mekanizmasının anlaşılırlığının artması ile beraber daha kapsamlı bir alan olmaya başlamıştır [1].

Görüntü işlemenin en önemli konularından bir tanesi nesne takibidir. Nesne takibinin en basit tanımı; bir video içerisinde belirlenen nesnenin imgeler boyunca konum ve hız gibi nicel bilgilerin otomatik olarak tahmin edilmesidir. Nesne takibi savunma sanayi, robotik, sağlık, insan-bilgisayar etkileşimi [2], arttırılmış gerçeklik, film endüstrisi, mobil uygulamalar gibi birçok alanda kullanılmaktadır [3]–[8].

Nesne takibi için birçok farklı yöntem geliştirilmiş ve halen geliştirilmeye devam etmektedir. Nesne takibi için geliştirilen yöntemlere gelinen zaman boyunca bakıldığında nesneyi daha iyi tanımlayan özniteliklerin çıkarılması [4], nesneyi en iyi tanımlayan modellin belirlenmesi [9] veya nesne harekelerinin daha iyi tahmin edilmesi [10] gibi farklı bakış açılarında oldukça ilerlemeler kaydedilmiştir. Fakat nesne takibinde nesnenin kaybolması (kapanma-occlusion), ortam aydınlanma değişimi, arka plan dinamiği, hareket bulanıklığı, bakış açısı değişimi gibi karşılaşılan zorluklardan dolayı güçlü ve verimli bir yöntem geliştirmek hala zor bir problem olmuştur [11].

Geleneksel nesne takip algoritmaları daha çok manuel olarak elde edilmiş öznitelikleri kullanarak nesne modeli oluşturur. Sonrasında bu model ardışık çerçeveler boyunca izlenir. Genellikle kullanılan modelin güncel tutulması için çevrimiçi model güncelleme işlemleri yapılır [12]. Günümüzde ise zorlu nesne takip problemlerinin çözümüne yönelik derin öznitelik tabanlı yöntemler geliştirilmiştir. Bu yöntemler test aşamasında yüksek hızlara erişmek için çevrim dışı olarak eğitilmektedir [13]–[17]. Çevrim dışı eğitim gören bazı yöntemler aynı zamanda çevrimiçi ince ayarlar yaparak doğruluk değerini arttırmaya çalışmaktadır. Literatür çalışmalarına bakıldığında, geçmişten günümüze birçok nesne takip yöntemi geliştirilmiştir. Bu yöntemlerin incelenmesi ve birbirleri ile kıyaslanması oldukça önemlidir. Zamanla, bu yöntemlerin karşılaştırılmasına yönelik ihtiyaç olacağı görülmektedir.

1.1. Tezin Amacı

Bu tez çalışmasının amacı zorlu nesne takip problemlerine karşı literatürde bulunan güncel nesne takip yöntemlerinin incelenmesi ve değerlendirilmesidir. Özellikle derin öğrenme algoritmalarının geliştirilmesiyle üstün başarılar gösteren derin öğrenme tabanlı nesne takip yöntemlerinin incelenmesi ve diğer geleneksel yöntemler karşısında avantajları ve dezavantajları hakkında değerlendirilmelerin yapılması önem arz etmektedir. Bu doğrultuda öncelikle detaylı literatür araştırması yapılmıştır. İncelenen yöntemler arasında öne çıkan bazı yöntemler uygulamalı olarak daha detaylı analiz edilmek üzere seçilmiştir.

Seçilen yöntemlerin 5 tanesi günümüzde oldukça popüler olan derin öğrenme tabanlı güncel yöntemlerdir. Diğerleri ise daha geleneksel görüntü işleme ve makine öğrenmesi tabanlı yöntemler olmakla beraber yakın zamanda geliştirilmiş ve literatürde güçlü yer edinmiş yöntemlerdir. Bununla birlikte derin öğrenme tabanlı ve geleneksel yöntemlerin hız ve performans açılarından detaylı analizleri yapılmıştır. Özellikle basit veri setlerinin yanında ayrıca nesne görünümünün kısmen engellendiği, ortam ışık miktarının değiştiği, nesnenin kameraya yakınlaştığı veya uzaklaştığı problemlere sahip olan veri setleri seçilmiştir. Yine seçilen yöntemler sadece literatürde mevcut olan veri tabanları ile değil aynı zamanda benzer karakteristiklere sahip olan ancak tez çalışmaları sırasında oluşturulan video veri tabanı üzerinde de test edilmiştir. Bunun sonucunda geliştirilen yöntemlerin günlük kullanımlardaki kararlılığı ve geçerliliği gözlemlenmiştir.

1.2. Tezin Gerekçeleri

Nesne takibi görüntü işleme konuları arasında en önemli ve en zor konulardan biri olarak görülmektedir. Günümüzde güncel yöntemleri ele alıp inceleyen birkaç çalışma bulunmakla birlikte, literatürde geleneksel yöntemler ile derin öğrenme tabanlı yöntemleri uygulamalı ve farklı veri setleri üzerinde kıyaslayan bir derleme çalışmasının olmaması bu konunun seçilmesinde önemli rol oynamıştır. Yapılan en yakın tarihli inceleme çalışmasında [1] sadece derin öğrenme yöntemleri ele alınıp irdelenmiştir. Bu tez çalışmasında yukarıdaki çalışmaya ek olarak bilinen görüntü işleme ve makine öğrenmesi tabanlı nesne takip yöntemleri de ele alınıp farklı açılardan derin öğrenme tabanlı çalışmalar ile karşılaştırılmıştır. Çalışmanın temel dayanak noktası, güncel derin öğrenme tabanlı yöntemlerin sağladığı avantajlar ile birlikte nesne takip çalışmalarının geldiği noktanın ortaya konulmasıdır. 2014 yılında Badillo vd. [2] yaptıkları çalışmada nesne takip yöntemlerini kapsamlı bir şekilde ele almalarına rağmen, Bu çalışma son zamanlardaki yöntemler hakkında bilgi içermemektedir. Mancini vd. [8] nesne takip yöntemlerini genel ve yüzeysel olarak incelenmiş ancak bu yöntemler üzerinde uygulamalı testler gerçekleştirmemiştir. Liv vd. [9] nesne takip yöntemlerini sınıflandırmaya yönelik bir derleme çalışması yapmıştır. Tiwari ve Singhai [10] yaptıkları çalışmada ise nesne

takip yöntemlerinde kullanılan öznitelik çıkarma yöntemleri hakkında kısa ve genel bilgiler sunmuştur. Bu tez çalışmasında ise güncel derin öğrenme tabanlı nesne takip yöntemlerinin detaylı incelenmesi gerçekleştirilmiştir. Bu yöntemlerin güçlü ve zayıf yönleri farklı açılardan uygulamalı bir şekilde ortaya çıkarılmıştır.

1.3. Tezin Çıktıları

Bu tezin ana çıktısı literatürde bulunan nesne takip yöntemlerin incelenmesi ve değerlendirmesini içeren bir derleme çalışması olmuştur. Bu derleme çalışması ile son zamanlarda oldukça popüler olan derin öğrenme yöntemlerinin nesne takip problemine kazandırdığı pozitif etkiler farklı veri setleri üzerinde irdelenmiştir.

Diğer bir ana çıktısı ise seçilen geleneksel nesne takip yöntemleri ile güncel derin öğrenme yöntemlerinin çalışma prensipleri, avantajları ve dezavantajları hakkında nicel verileri içeren değerlendirmelerdir. Değerlendirmeler sırasında literatürde bulunan veri setlerin yanında zor problemler içeren video veri tabanı oluşturularak güçlü bir değerlendirme yapılması sağlanmıştır. Elde edilen bu veri seti ile tüm yöntemler test edilmiştir. Test sonuçları detaylı yorumlar ile birlikte sunulmuştur.

1.4. Tezin Organizasyonu

Bölüm 2'de öncelikle literatürde nesne takibi ile ilgili yapılan derleme çalışmaları incelenmiştir. Daha sonra incelenen derleme çalışmaları doğrultusunda nesne takip yöntemleri gruplandırılmıştır. Yöntemler incelenmeden önce nesne takip yöntemlerinin kullandığı öznitelikler hakkında genel bilgiler verilmiştir. Sonraki adımda literatürde bulunan yöntemler gruplandırılarak incelenmiştir.

Bölüm 3'te geleneksel ve güncel yöntemlerin çalışma biçimini detaylandırmak için kapsamlı bir şekilde 10 farklı yöntem ele alınmış ve çalışma prensipleri ve temel işlem adımları detaylandırılmıştır. Ele alınan yöntemlerden 5 tanesi güncel derin öğrenme tabanlı ve diğer 5 tanesi de geleneksel yöntemlerden oluşacak şekilde seçilmiştir. Burada seçilen yöntemler temel metodolojileri, çalışma prensipleri ve gerekli matematiksel ifadeleri verilerek anlatılmıştır.

Bölüm 4'te ele alınan 10 farklı yöntemin hem literatürde bulunan veri setleri hem de tez çalışmaları sırasında oluşturulan video veri tabanı üzerindeki performans testleri gerçekleştirilmiştir. Elde edilen sonuçlar hız, doğruluk ve zayıf/güçlü yönler açısından yorumlanarak yöntemler hakkında detaylı analizler yapılmıştır. Son olarak Bölüm 5'te tez çalışmasında elde edilen sonuçlara ve önerilere yer verilmiştir.

2. NESNE TAKİP VE TESPİT YÖNTEMLERİ

Nesne takip problemi; bir videodaki ilk çerçevede belirlenen bir hedef nesnenin başlangıç durumu (genellikle bir sınırlayıcı kutu) verildikten sonra ardışık çerçeveler boyunca hedef nesnenin durumunu (konumu ve ölçeği gibi) tahmin etmeyi amaçlar. Son yıllarda önemli ilerlemeler kaydedilmesine rağmen kapanma (occlusion), deformasyon, döndürme, aydınlanma değişiklikleri, ölçek değişiklikleri, hareket bulanıklığı gibi problemler nedeniyle nesne takibi hala zor bir görüntü işleme konusu olarak kabul edilmektedir. Literatürde bulunan birçok nesne takip çalışması, nesne takip problememine farklı bakış açılarıyla yaklaşarak doğru ve hızlı bir nesne takip yöntemi geliştirmeyi amaçlamıştır [3]–[8] .

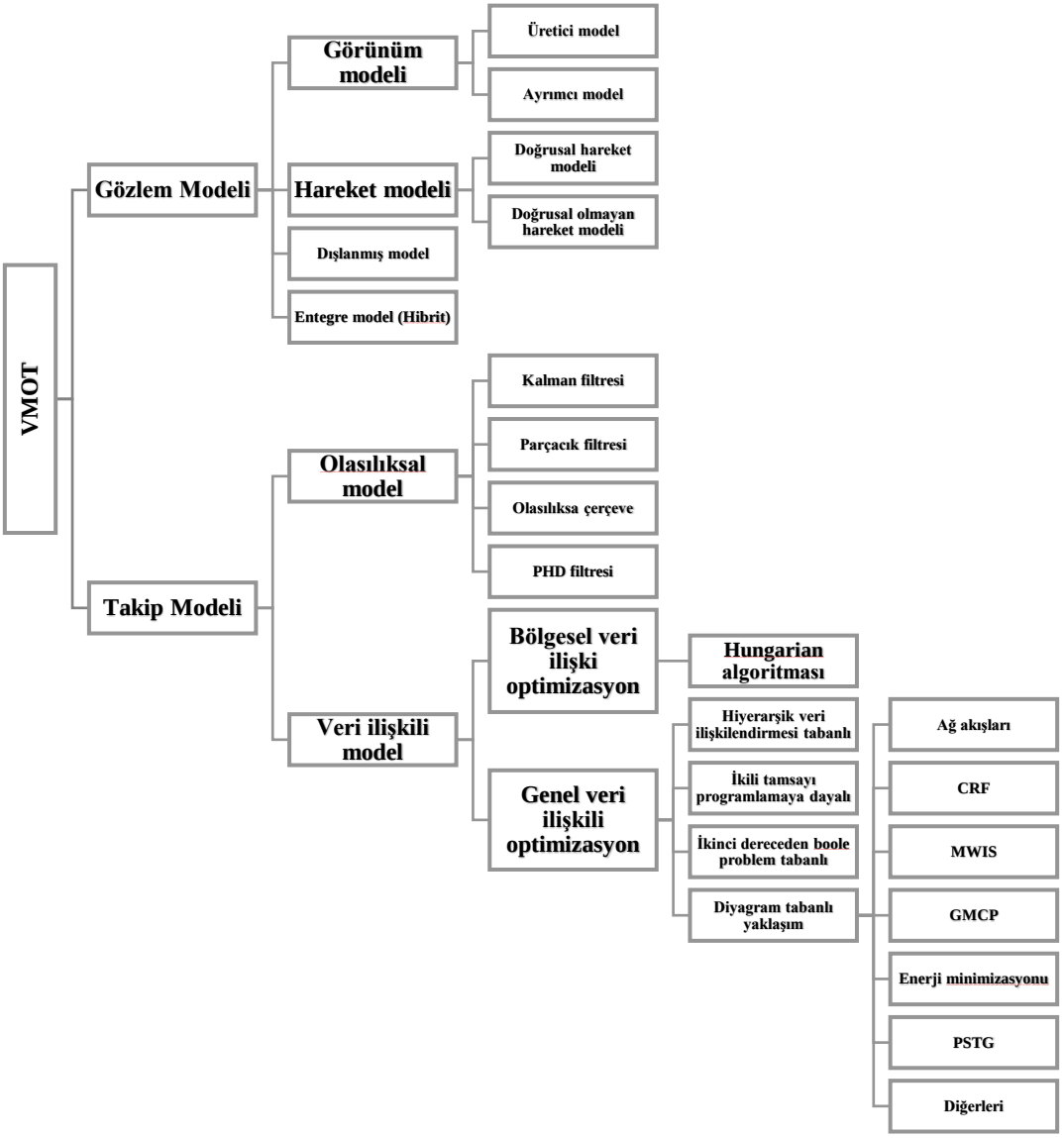
Nesne takibinde son zamanlarda geliştirilen popüler derin öğrenme tabanlı yöntemler oldukça başarılı sonuçlar elde etmiştir [11]. Bunun yanında korelasyon filtre tabanlı yöntemlerde farklı kullanım metotları veya farklı öznitelikler kullanarak hala nesne takibinde başarılı sonuçlar çıkaran yöntemler arasındadır [18]. En başarılı yöntemlere bakıldığında ise benzerlik hesaplamalarına dayanan derin Samese tabanlı ve derin öznitelik kullanan korelasyon tabanlı yöntemlerin öne çıktığı görülmektedir [11]. Bu yöntemlere kıyasla çoklu hafıza gibi daha farklı bir yaklaşım getiren geleneksel çok yapılı nesne takip yöntemleri de mevcuttur [19]. Dolayısıyla bu yöntemlerin incelenmesi, analiz edilmesi ve kıyaslanması ileriki çalışmalar için oldukça önemlidir. Bu bölümün geri kalanında nesne takip çalışmalarını inceleyen derleme çalışmalarından da faydalanılarak yöntemlerin anlaşılırlığını güçlendirecek bir yaklaşım ile nesne takip çalışmaları incelenmiştir.

2.1. Literatürde Bulunan Derleme Çalışmaları

Literatürde bulunan derleme çalışmaları genellikle nesne takibinde geliştirilen yöntemleri *görünüm* ve *hareket modeli* olarak ele alırken, bazı çalışmalar ise yöntemleri *nesne tanıma, sınıflandırma* ve *takibi* şeklinde gruplandırılarak ele alınmıştır. Literatürde bulunan nesne takip yöntemlerinin daha iyi anlaşılması için öncelikle güçlü bir gruplandırma biçiminin benimsenmesi gerekmektedir. Bu tez çalışmasında güçlü bir gruplandırma biçimi seçmek için literatürde bulunan popüler derleme çalışmalarından faydalanılmıştır.

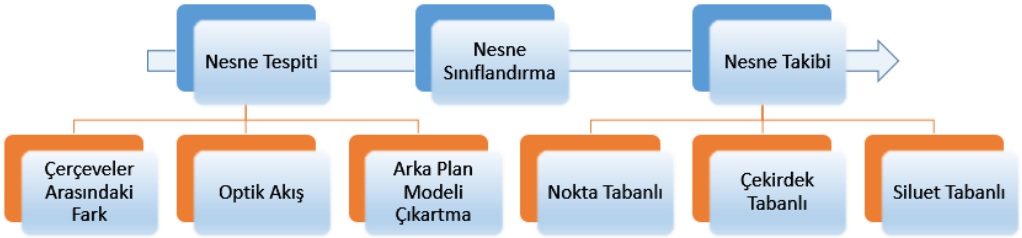
Fan vd. [4] literatürde bulunan çoklu nesne takip yöntemlerinin sistematik bir derleme çalışmasını yapmıştır. Bu çalışmada Şekil 2.1'de görüldüğü gibi nesne takip yöntemleri genel olarak gözlem ve takip olarak iki ana gruba ayrılmıştır. Her grup, çalışma prensibine göre kendi içinde farklı gruplandıra ayrılmıştır. Bu şekilde toplam 19 grup oluşturulmuştur. Her grup altında ilgili alanda yapılan çoklu nesne takip yöntemlerin ilkeleri, avantajları ve dezavantajları kısaca detaylandırılmıştır. Yakın zamanda geliştirilen yöntemler hibrit bir yapıda olduğu için aşırı gruplandırma okuyucularda anlam karmaşasına sebep olabilmektedir.

Fan vd. [4] yaptığı çalışmaya benzer bir başka derleme çalışması Luo vd. [20] tarafından yapılmıştır. Bu çalışmada, önceki çalışmadan farklı olarak seçilen bazı yöntemlerin belirlenen veri setleri üzerindeki deneysel sonuçlarına yer verilmiştir.



Şekil 2. 1. Luo vd. [20] yaptığı derleme çalışmasındaki nesne takip yöntemlerinin gruplandırılması

[3], [7], [10], [21] derleme çalışmalarında literatürde bulunan nesne takip yöntemleri hiyerarşik olarak nesne tanıma, sınıflandırma ve takip şeklinde gruplandırılarak incelenmiştir. Yılmaz vd. [3] yaptığı çalışma en kapsamlı derleme çalışmalarından biridir. Nesne takip probleminde 2006 yılına kadar getirilen çözümler nesne sunumu, tespiti ve takibi şeklindeki başlıklar altında detaylı bir şekilde incelenmiştir. Bir diğer kapsamlı çalışma ise [10] nolu derleme çalışmasıdır. Bu çalışmada Yılmaz vd. yaptığı gruplandırmaya benzer gruplandırmalar yapılarak ilgili başlık altında yapılan yöntemler incelenmiştir (Şekil 2.2).



Şekil 2. 2. Nesne tespit ve takibinde kullanılan temel yöntemlerin gruplandırılması [3], [7], [10], [21]

Li vd. [9] yaptığı derleme çalışmasında görsel nesne takibi için 2 boyutlu görünüm modelleme yöntemlerine odaklanmıştır. Gözlem modeli için kullanılan öznitelikler ve istatistiksel öğrenme yöntemleri ele alınmıştır. İstatistiksel öğrenme yöntemleri kendi içinde üretici, ayırt edici ve hibrit yöntemler olarak gruplandırılmıştır. Öznitelikler ise kendi içinde lokal ve global olarak ele alınmıştır. Daha sonra nesne takibinde kullanılan öznitelikler ve görünüm modeli altında yapılan çalışmalar sunulmuştur.

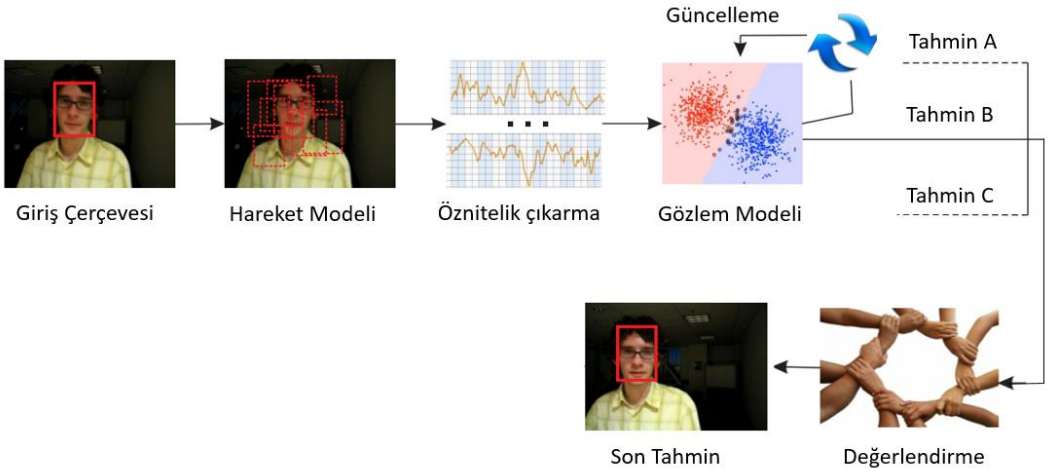
Smeulders vd. [12] nesne takip yöntemlerin performans değerlendirmesinde genellikle kısıtlı video verisi kullanıldığını vurgulanmıştır. Bu nedenle ele aldığı 19 nesne takip yöntemini 315 video veri seti ile detaylı testlere tabi tutmuş ve karşılaştırmalar yapmıştır. Bu çalışmada ele aldığı nesne takip yöntemlerini üretici (model karşılaştırmalı takip, genişletilmiş görünüm modeli karşılaştırmalı takip, kısıtlama ile model karşılaştırmalı takip) ve ayırt edici (ayırt edici sınıflandırma ile takip yöntemi, kısıtlama ile ayırt edici sınıflandırma kullanarak takip yöntemi) yöntemler olarak gruplandırmıştır. [22] çalışması Smeulders vd. [12] yaptığı derleme çalışmasına benzer şekilde gruplandırma yapmış ve ayrıca ayırt edici yöntemlerin altına derin öğrenme yöntemlerini de eklemiştir.

Chen vd. [18] yaptıkları çalışmada son zamanlarda yapılan nesne takip yöntemlerinde oldukça başarılı olan korelasyon filtre yöntemlerini ele almışlardır. Bu çalışmada korelasyon filtresinin çalışma prensibi detaylandırılmış ve bu alanda yapılan güçlü çalışmalar gruplandırılarak değerlendirilmiştir. Ele alınan 11 korelasyon filtresi tabanlı nesne takip yöntemi yapılan deneysel çalışmalar ile nicel olarak incelenmiştir.

Li vd. [11] yaptıkları çalışmada literatürde bulunan derin öğrenme tabanlı nesne takip yöntemleri ağ yapısı, ağ işlevi ve ağ eğitimi şeklinde 3 gruba ayrılarak ele alınmıştır. Ele alınan nesne takip yöntemlerinin OTB-100 [23], TC-128 [24] ve VOT2015 [25] veri setleri üzerindeki çalışma hızları ve gösterdikleri başarılar değerlendirilmiştir.

Bo vd. [26], geliştirilen nesne takip yöntemlerinin performans analizi için yöntemlerin daha sistematik olarak incelenmesi gerektiğini belirtmiştir. Literatürde bulunan veya yeni geliştirilen bir nesne takip yönteminin performans analizi için hareket modeli, öznitelik çıkarma tekniği, gözlem modeli, model güncelleyici yapısı ve birden fazla karar mekanizması

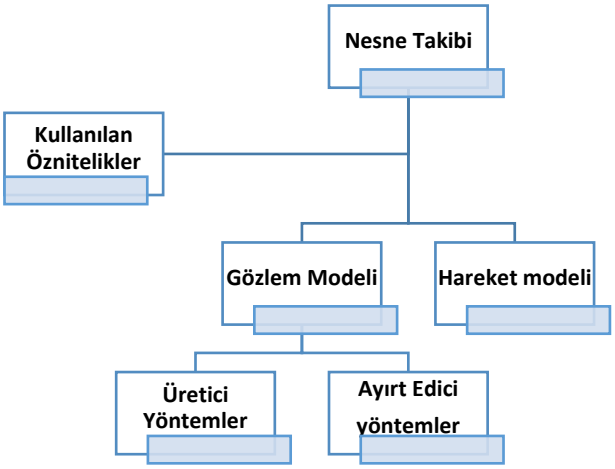
içeren yöntemler olmak üzere yöntemleri beş bileşene ayrılarak, analizlerin bu şekilde yapılması gerektiğini savunmuştur. Bu fikir doğrultusunda yeni bir analiz çerçevesi geliştirilmiştir (Şekil 2.3). Çalışmada elde edilen sonuçlara bakıldığında; kullanılan özniteliklerin nesne takibindeki başarıyı etkileyen en önemli etkenlerden biri olduğu öne çıkmıştır.



Şekil 2. 3. Wang vd. [26] tarafından önerilen sistemin ana yapısı

Literatürde bulunan ve incelenen tüm derleme çalışmalarına bakıldığında nesne takip yöntemlerinin genellikle iki farklı gruplandırma biçimi ile ele alındığı anlaşılmaktadır. Birinci gruplandırma biçiminde yöntemler *arka plan çıkarma, nesne tespiti* ve *nesne takibi* şeklinde ele alınırken, ikinci gruplandırma biçiminde ise yöntemler *hareket modeli* ve *gözlem modeli* şeklinde gruplandırılmaktadır. Özellikle 2010 yılı ve sonrası çalışmalarında daha çok ikinci gruplandırma biçimi kullanılmıştır. Bunun ana sebeplerinden biri yöntemlerin daha hibrit bir yapıda olması ve her nesne takip yönteminin içerisinde aynı zamanda bir nesne tespit işleminin olmasıdır. Nesne takibi için yapılan derleme ve analiz çalışmalarına bakıldığında, kullanılan öznitelik çıkartma yöntemlerinin nesne takip çalışmalarının performansını oldukça önemli derecede etkileyen çok temel bir parametre olduğu sonucuna varılmıştır [9], [20], [26], [27].

Bu tez çalışmasında, literatürdeki nesne takip yöntemleri ve bu yöntemleri inceleyen derleme çalışmaları doğrultusunda Şekil 2.4'te görülen gruplandırma biçiminin kullanılması kararlaştırılmıştır. Bu noktada öncelikle nesne takip yöntemlerinde öne çıkan güçlü öznitelikler ele alınmıştır. Daha sonra yöntemler *gözlem modeli* ve *hareket modeli* şeklinde gruplandırılmıştır. Gözlem modeli de kendi içinde *üretici* ve *ayırt edici* yöntemler şeklinde ele alınmıştır.



Şekil 2. 4. Tez çalışmasında incelenen nesne takip yöntemlerinin gruplandırılması

Hareket modeli, takip edilen nesnenin önceki çerçevelerdeki durumlarını zamana göre tanımlayarak gelecekteki çerçevede nesnenin olası aday durum serisini üretmeyi amaçlar. Kalman ve parçacık filtresi, yapılan birçok yöntemde hareket modeli olarak sıklıkla kullanılmıştır. Parçacık filtresi temel olarak bir Bayes sıralı güçlü örnekleme tekniğidir. Ağırlıklı örnek kümesini kullanarak koşullu dağılımı yinelemeli olarak çıkartır. Yeni gelen örnek bu dağılım kullanılarak değerlendirilir. Nesne takibinde ise önceki konumlarında elde ettiği olasılıksal durumlara göre yeni çerçevedeki konumu tahmin etmeye çalışır [28].

Gözlem modelinde, genellikle takip edilen nesneyi tanımlayan öznitelikler kullanılarak, gelecek çerçevede çıkardığı nesne aday görüntülerinden gerçek nesneyi bulmaya çalışır. Gözlem modeli kendi içinde üretici ve ayırt edici olarak ikiye ayrılmaktadır. Üretici modeller genellikle benzerlik yoluyla nesneyi bulmaya çalışırken, ayrımcı modeller ise hedef nesne görünümü ile arka plan arasında bir sınıflandırıcı kullanarak hedefi konumlandırır [28].

2.2. Nesne Takibinde Kullanılan Öznitelikler

Nesne takip yöntemlerinde ilk adım genellikle nesneyi tanımlayan özniteliklerin çıkarılmasıdır. Çıkarılan özniteliklerin nesne ve arka planı birbirinden ayıracak nitelikte olması gerekmektedir. Takip edilecek nesne çerçeve geçişlerinde farklı değişimlere uğrayabilmektedir. Bu yüzden seçilen özniteliklerin bu değişimlere karşı sağlam kalabilmesi gerekmektedir. En basit tanımıyla bir ışık değişimi neredeyse tüm görüntüdeki piksel değerlerinin değişmesine sebep olabilir. Bu durumlar için nesnenin renk bilgisi yerine kenar bilgisi daha güçlü bir öznitelik olacaktır. Nesnedeki şekil veya kenar bozulmaları durumunda ise nesnenin doku bilgisi ön plana çıkmaktadır.

Nesne takip yöntemlerinde kullanılan özniteliklerin güçlü olması, nesne takip başarısını doğrudan etkileyen bir faktördür. Farklı problemlere karşı daha dayanıklı birçok öznitelik tipi

ortaya çıkmıştır. Bunlardan en çok kullanılanları renk, optik akış, haar, yerel ikili örüntü (YİÖ), yönlü gradyan histogramı (YGH) ve derin özniteliklerdir.

Renk: Renk bilgisi gerçek dünyadaki bir nesnenin sayısal ortama taşındığında nesnenin görüntüsündeki renkleri temsil eden sayısal değerleridir. Bu sayısal değerler farklı formatta bulunabilir. Günümüzde en yaygın kullanılan renk formatı RGB (kırmızı, yeşil, mavi) renk uzayıdır. Buna ek olarak ihtiyaca göre L*u*v*, HSV ve HSI gibi renk uzayları da mevcuttur. Bu değerler nesnelerin birbirilerinden ayrılmasında en basit ve hızlı özniteliklerdir [29].

Optik Akış: Optik akış özniteliği temel olarak nesnelerinin hareket bilgisidir. Bu öznitelik sayesinde hareketli nesneyi arka plandan ayırmak oldukça kolaydır. Optik akış özniteliğinin en temel elde edilme şekli iki çerçeve arasındaki değişimin hesaplanmasıdır. Bu şekilde çerçeveler arasında hareketli nesneler, çerçeve içinde güçlü bir şeklide algılanabilmektedir.

Haar: Belli boyutlarda oluşturulan filtrelerin şekil üzerinde gezdirilmesi ile elde edilen özniteliktir. Kullanılacak problemdeki ihtiyaca göre birçok farklı filtre tipi bulunmaktadır. Nesne üzerinde gezdirilen filtrenin beyaz alanlarının toplamı ile siyah alanlarının toplamının arasındaki fark alınarak Haar özniteliği elde edilmektedir. Haar benzeri öznitelikler, ışık değişimine karşı güçlü bir öznitelik olduğundan, nesne takibinde oldukça tercih edilen bir özniteliktir [30].

Yerel İkili Örüntü (YİÖ, Local Binary Patterns). Yerel ikili örüntü bir görüntünün mekansal yapısını özetlemek için parametrik olmayan bir çekirdek matrisi ile basit fakat çok etkili bir doku özniteliği çıkartma yöntemidir. Belirlenen çekirdek matrisi belirli bir fonksiyondan geçirilerek her piksel için sayısal bir değer elde edilmektedir [30].

Yönlü Gradyan Histogramı (YGH, Histogram Of Gradients): Nesne görüntüsünün kenar eğim açılarınının gradyan yoğunluk dağılımı kullanılarak elde edilen özniteliktir. YGH özniteliğin temel amacı, kenar bilgisini veren bir grup bölgesel histogramlar yardımıyla nesne görünümünü tanımlamaktır [31]. Bu öznitelik sayesinde nesne görüntüsündeki piksel değerlerinin hangi yöne doğru yoğunluk gösterdiği hesaplanabilmektedir.

Derin öznitelik: Derin öznitelikler nesne görüntüsünün çok katmanlı bir yapay sinir ağından geçirilerek elde edilen kalıtsal özellikleridir. Son zamanlarda elle hazırlanmış özniteliklerin yerine birçok uygulamada derin öznitelikler kullanılmıştır. Yapılan [32] ve [33] çalışmalarında nesne takibi ve nesne tanıma için derin öğrenme kullanarak güçlü özniteliklerin çıkarılabileceği kanıtlanmıştır. Bu öznitelikler sayesinde nesne tanıma, sınıflandırma veya takibi gibi birçok uygulamalarda yüksek başarılar elde edilmiştir [34]. Bu öznitelikler bölüm 3.1'de detaylı bir şekilde ele alınmıştır.

2.3. Gözlem Modeli

Gözlem modeli nesnenin renk, şekil ve doku gibi görsel özelliklerini temsil eden modeldir. Gözlem modeli tabanlı nesne takip yöntemleri nesnenin görüntüsü hakkında önceden edindiği bilgileri kullanarak geçerli çerçevede nesnenin konumunu bulmaya odaklanır. Bu yöntemler kendi içinde **üretici** ve **ayırt edici** yöntemler olarak ikiye ayrılmaktadır.

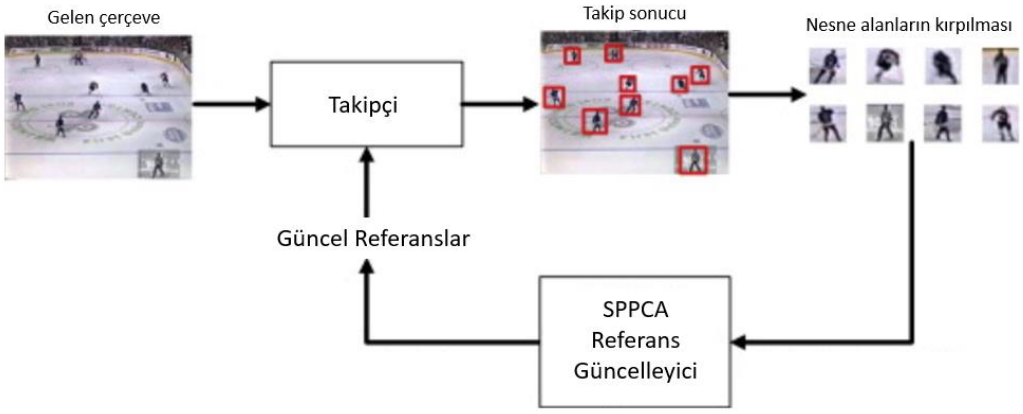
2.3.1. Üretici Model

Üretici model tabanlı yöntemler öncelikle takip edilecek nesnenin özniteliklerini kullanarak nesneyi tanımlayan bir görünüm modeli oluşturulur. Daha sonra çerçeve içinde oluşturulan görünüm modeline en çok benzeyen nesne görünümünü yakalamaya çalışır. Bunun için geçerli çerçevede bir arama işlemi uygulayarak nesne görüntüsüne en çok benzeyen aday görüntü konumunu bulmaya çalışır. Arama işlemlerinde bazı yöntemler tüm çerçeveyi ararken, gelişmiş yöntemler ise nesnenin bulunabileceği belirli bir alan belirleyerek bu alanda bir arama işlemi gerçekleştirir. Belirli bir görüntünün tüm çerçevede aranması oldukça maliyetli bir yaklaşım olduğu için, belli bir bölgenin aranması hız açısından avantaj sağlayabilmektedir.

KLT (Lucas Kanade Takipci - Lucas Kanade Tracker) yöntemi 1981'de Lucas-Kanade algoritması önerildikten sonra bilgisayar görmesi alanında en yaygın kullanılan tekniklerden biri haline gelmiştir. Optik akış ile nesne takibi, katmanlı hareket ve yüz kodlama seçeneklerine kadar geniş bir kullanım alanına sahip olmuştur [35]. Literatürde bulunan birçok yöntem orijinal KLT yöntemine geniş bir çeşitlilik kazandırmıştır. KLT algoritması temel olarak görüntüde belirlenmiş olan bir noktanın ikinci görüntüde hangi noktaya kaydığını bulmayı amaçlamaktadır. Nesne takibinde ise nesnenin belirli noktalarının birlikte nereye kaydıklarını bulmayı amaçlamaktadır. Bunun için nesneye belirli dönüşümler uygulanarak nesne noktalarının ilişkisine bakmaktadır. KLT algoritmasında, çerçeveler arasında belirlenen noktaların geçiş yerlerini en iyi şekilde bulmak için Gauss Newton Gradyan, Gauss-Newton, Levenberg-Marquardt algoritmaları kullanılmıştır [35].

Ross vd. [36] nesne takibi için çevrimiçi olarak artırımlı model kullanan bir nesne takip yöntemi önermiştir. Model güncelleme yöntemi için, eşzamanlı olarak değişen poz ve ışık koşullarında nesne izlemeyi kolaylaştıran etkili bir alt uzay güncelleme algoritması geliştirilmiştir. Temel olarak kullanılan nesne öznitelikleri yeni çerçeve geldikçe güncellenir. Böylece hedef nesnenin görünümü her zaman güncel tutulabilmiştir. Lu vd. [37] yaptıkları çalışmada hokey oyun sahasında oyuncuları otomatik olarak izleyebilen ve hareket yön tahmini yapan parçacık filtre tabanlı bir yöntem önerilmiştir. Takip sırasında algılama, kaydırma, eğme ve yakınlaştırma gibi problemlerle başa çıkabilen güçlü bir yaklaşım

geliştirilmiştir. Oyuncuların görünümünü modellemek için YGH ve HSV renk histogram özniteliği yardımıyla olasılıksal bir model oluşturulmuştur. Test zamanında hokey oyuncularının takibi için arttırılmış parçacık filtresi yöntemi kullanılmıştır. Yöntemin akış şeması Şekil 2.5'te gösterilmiştir.



Şekil 2. 5. [37] nolu çalışmada önerilen yöntemin akış şeması

[38] çalışmasında çevrimiçi yerel yönetim çekirdeği (Local Steering Kernel) yöntemi ile model karşılaştırmalı bir yöntem önerilmiştir. Bu yöntemde nesne görüntüsünün her hareketini (yakınlaştırma veya döndürme) bir nesne örneği olarak düşünülmüş ve bir hafıza alanına depolanmıştır. Bu şekilde nesne görüntü örnekleri kullanılarak nesne modelini oluşturmuştur. Önceki çerçevede nesne konumundan yararlanarak nesnenin geçerli karedeki ortalama konumu (Arama bölgesi) doğrusal bir Kalman filtresi kullanılarak öngörülmüştür. Arama bölgesi ile depolanan nesne görüntüleri arasında renk histogramı ve yerel yönetim çekirdeği [38] yöntemleri kullanılarak benzerlik oranları çıkartılmıştır. Çıkartılan benzerlik oranları ile yeni çerçevedeki nesne konumu belirlenmiştir. [39] çalışmasında nesne takibi için ağırlıklı yerel kosinüs benzerliğini hesaplayan bir fonksiyon önerilmiştir. Yeni gelen çerçevede nesne olabilecek aday görüntü parçaları ile nesne görünüm modeli arasında geliştirilen benzerlik fonksiyonu kullanılarak görüntüler arası benzerlik değerleri hesaplanmıştır. Elde edilen benzerlik oranlarına bakılarak nesne ile en çok uyuşan görüntü parçası nesnenin yeni konumu olarak kabul edilmiştir. Belirlenen yeni nesne görüntüsü parçacık filtresi yardımıyla basit bir güncelleme yapılarak nesne modeli ile ağırlıklı yerel kosinüs benzerliği fonksiyonunun güncel tutulması sağlanmıştır.

Literatürde bulunan nesne takip yöntemlerinde, genellikle hedef nesneyi çıkartmak için sınırlama dikdörtgeni kullanmaktadır. Bu sınırlama kutusu içindeki görüntü nesne olarak kabul edilmektedir. Nesne olarak kabul edilen bu görüntü kısmen de olsa arka plan görüntüsü de içerebilmektedir. Belagiannis vd. [40] bu problemden yola çıkarak örnekleme için bölütleme tabanlı nesne takip yöntemi önermiştir. Bölütleme sonrasında alınan görüntünün

renk ve YGH öznitelikleri hesaplanmıştır. Daha sonra bu öznitelikler parçacık filtre tabanlı bir nesne takip yöntemi ile kullanılmış ve başarılı sonuçlar elde edilmiştir.

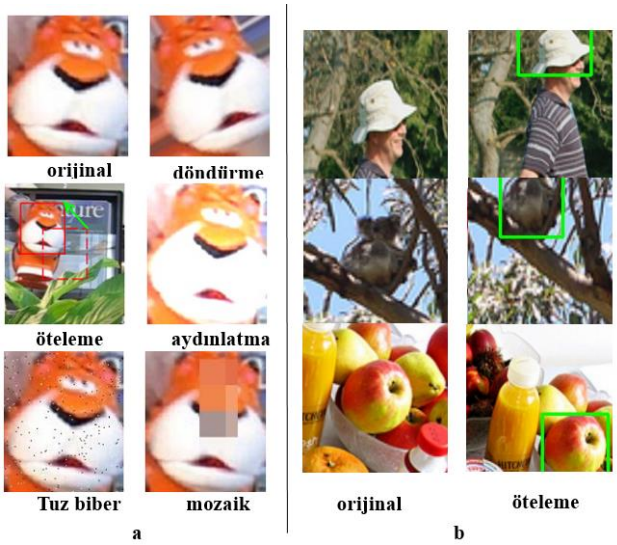
Held vd. [13] nesne takibi için yüksek hızlarda çalışan derin öğrenme tabanlı GOTURN yöntemi geliştirmiştir. Bu çalışmada geliştirilen ağ mimarisi çevrim dışı olarak 41 gigabayt gibi yüksek boyutlardaki veri kümesiyle eğitilmiştir. Daha sonra nesne takibi için öncelikle GOTURN ağ mimarisinin girişine verilen nesne ve bu nesnenin aranacağı bölgenin derin öznitelikleri hesaplanmıştır. Ardından elde edilen derin öznitelikler kullanılarak tam bağlı katmanlarda nesne konumu hesaplanmıştır. Bu yöntemde ağ mimarisinin parametrelerinde herhangi bir güncelleme işlemi yapılmamış ve bu sayede oldukça yüksek hızlarda nesne takibi yapılmıştır.

Tao vd. [41] hedef nesne görüntüsüyle geçerli çerçeve içerisinden alınan nesne aday görüntüleri eşleştirilerek nesne takibini yapan SINT (Benzer Örnekleri Arama - Siamese Instance Search For Tracking) yöntemi önermiştir. Bu yöntemde amaç iki görüntü arasında benzerlik oranını hesaplamak ve en yüksek benzerlik gösteren aday görüntüyü seçmektir. Bu işlem için son zamanlarda oldukça popüler olan Siamse ağ tabanlı bir ağ mimarisi geliştirilmiştir. Geliştirilen ağ mimarisinde öncelikle çevrim dışı olarak çok yüksek boyutlarda veriler ile iki görüntü arasında derin benzerlikleri eşleştiren bir fonksiyon öğrenilir. Daha sonra çevrimiçi nesne takibinde ağ mimarisine verilen nesne görüntüsü ile sırasıyla aday görüntüler kıyaslanır.

Bertinetto vd. [15] yaptıkları çalışmada SINT yöntemine benzer bir prensibe sahip SiamFC ağ mimarisi geliştirmiştir. SiamFC yöntemi takip edilen nesne görüntüsünü SINT yöntemindeki gibi tüm aday görüntüler ile karşılaştırmak yerine sadece belirli bir arama alanı ile karşılaştırır. Bu şekilde işlem zamanında kazanç sağlanmıştır. SiamFC yöntemi iki görüntüden aldığı derin öznitelikleri geliştirilen bir korelasyon filtresinde birleştirmiştir. Burada iki görüntü arası benzerlik hesaplama işleminden sonra 17×17 boyutunda bir ısı haritası elde edilmiştir. En yüksek benzerlik oranı gösteren bölge, çift yönlü enterpolasyon (bilinear interpolation) işleminden geçirilerek arama alanı görüntüsü üzerinden istenilen konum elde edilmiştir.

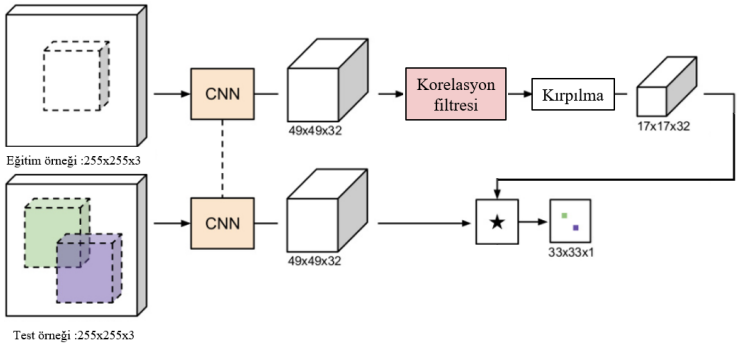
Chen vd. [42] yaptıkları çalışmada GOTURN yöntemine benzer bir yaklaşım olan YCNN (İki Akışlı Konvolüsyonel Sinir Ağı - Two-Flow Convolutional Neural Network) yöntemini önermiştir. Bu yöntemde nesne görüntüsü ve arama bölgesi geliştirilen ağ mimarisine gönderilerek arama bölgesi üzerinde nesnenin konumunu veren olasılıksal ısı haritası elde edilmiştir. GOTURN ve YCNN yöntemlerinde eğitim prosedürü olarak benzer işlemler gerçekleştirilmiştir. Diğer nesne takip yöntemlerinden farklı olarak bu yöntemler video veri setinin yanında büyük boyutlarda görüntü içeren ImageNet [43] veri setindeki görüntüleri kullanılarak da eğitilmiştir. Görüntü dizelerinin temel kullanım amacı ağ mimarilerinde ihtiyaç duyulan geniş ve büyük boyutlardaki eğitim seti ihtiyacının karşılanmasıdır. Kullanım şekilleri

olarak; GOTURN yönteminde arama görüntüsündeki nesne belli oranda kaydırılarak yapay hareketlilik kazandırılırken, YCNN yönteminde ise nesneye ait görüntüde döndürme, kaydırma, ışık değişimi gibi işlemler uygulanarak iki görüntü arasında hareketlilik kazandırılmıştır. Bu şekilde geliştirilen ağ mimarileri daha çeşitli nesneler tanımış ve nesneler arası ilişkiyi öğrenmiş olmaktadır. Şekil 2.6'da yöntemlere uygulanan yapay hareket ve değişimler gösterilmiştir.



Şekil 2. 6. (a) YCNN yöntemi [42] ve (b) GOTURN yöntemi [13] için kullanılan örnek eğitim görüntüleri.

SiamFC yöntemini geliştiren Valmadre vd. yaptıkları yeni bir çalışmada [44] nesne takibinde derin öznitelikleri elde etmek için katmanlar arasına bir korelasyon filtresi eklemiştir. Bu şekilde nesne takip başarısında artış hedeflenmiştir. Şekil 2.7'de gösterildiği gibi geliştirilen CFNet yöntemi SiamFC yönteminden farklı olarak girişte daha geniş bir nesne görüntüsü almıştır. Daha sonrada bu görüntüden elde edilen derin öznitelikler bir korelasyon filtresinden geçirilip nihai derin öznitelikler elde edilmiştir. Bu şekilde derin öznitelik çıkarım sistematiğine farklı ve etkin bir yaklaşım kazandırılmıştır. Yöntemin son adımında nesne öznitelikleri ve arama bölgesinin öznitelikleri arasında başka bir korelasyon filtresi yardımıyla benzerlik değeri hesaplanmıştır. SiamFC yöntemindeki gibi bu benzerlik değeri ile nesne konumu belirlenmiştir.



Şekil 2. 7. CFNet yönteminin ağ mimarisi [44]

Wang vd. [45] yaptığı çalışmada önceki derin öğrenme tabanlı kıyaslama yöntemlerine benzer şekilde nesne görüntülerinden çıkarılan derin öznitelikler arasında korelasyon filtresi yardımıyla nesne takibi yapılmıştır. Bu çalışmada geliştirilen DCFNet yönteminin korelasyon filtresinin parametreleri çevrimiçi olarak güncellenmiştir.

2.3.2. Ayrıt Edici Model

Ayırt edici nesne takip yöntemlerin temel çalışma prensibi hedef nesne ile arka planı birbirinden ayırarak nesne takibinin yapılmasıdır. Genellikle bu yöntemlerde hedef nesneyi arka plandan ayırt etmek için çevrimiçi bir sınıflandırıcı kullanılmaktadır. Temel olarak kullanılan sınıflandırıcı geçerli çerçevede alınan aday görüntü parçalarını, nesne ve arka plan olarak iki sınıfa ayırarak nesnenin yeni konumunu bulmayı amaçlamaktadır. Nesne konumu bulunduktan sonra sınıflandırıcının görünüm değişikliklerine karşı güçlü kalması için güncelleme adımı gerçekleştirilmektedir.

Ayırt edici nesne takip yöntemlerinde kullanılan sınıflandırıcının güncellenmesi için farklı yöntemler mevcuttur. En eski yöntemlerden bir tanesi sınıflandırıcıyı güncellerken bir pozitif örnek (diğer bir deyişle geçerli nesne konumu) ve birkaç negatif örnek kullanarak sınıflandırıcının güncellenmesidir [46], [47]. Genellikle bu yöntemler, takip edilen nesnede görünüm değişikliği veya kapanma (occlusion) problemleri meydana geldiği durumlarda yanlış pozitif örnek alarak hatalı güncellemeler yapabilmektedirler. Bu problemi çözmek için Grabner vd. çevrimiçi yarı denetimli öğrenme yöntemini önermiştir [48]. Bu yöntemde videonun sadece ilk çerçevesindeki örnek (nesne görüntüsü) etiketlenmiştir. Test zamanında elde edilen yeni örnekler yarı denetimli öğrenme problemi şeklinde alınarak güncelleme işlemi yapılmıştır. Güncelleme işlemleri için getirilen bir diğer çözüm ise Babenko vd. [49] tarafından geliştirilen çoklu örnekle öğrenme (multiple instance learning) çerçevesidir. Bu yöntemde test zamanında geçerli çerçeveden alınan görüntü örnekleri pozitif ve negatif çantalar/takımlar içerisinde toplanarak değerlendirilmiştir. Daha sonra oluşturulan çoklu çantalar sınıflandırıcının güncellenmesi için kullanılmıştır. Bu alanda yapılan bir diğer

çalışma [50] ise çevrimiçi yapısal kısıtlamalar ile olumlu ve olumsuz örneklerin oluşturulmasıdır. Örnekler çevrimiçi sınıflandırıcı yardımı ile belli bir güven değerinin geçilmesi durumunda etiketlenmiştir. [51] çalışmasında güncelleme işleminde yanlış etiketlenen örneklerin etkisini azaltmak için yeni bir yöntem önerilmiştir. Burada örneklerin etiket değerleri normal yöntemlerden farklı olarak oluşturulan bir kayıp fonksiyonu ile belirlenmiştir.

Literatürde bulunan birçok sınıflandırıcı nesne takibi için kullanılmıştır. Örneğin çoklu örnekle öğrenme (multiple instance learning) [52], naive bayes [53], metrik öğrenme [54], yapılandırılmış öğrenme [51] gibi sınıflandırıcılar nesne tespitinde güçlü sonuçlar elde edilmesi için denenmiştir. Ayrıca son zamanlarda daha doğru bir tespit için çoklu sınıflandırıcı kullanarak nesne takibi yapan yöntemlerde mevcuttur.

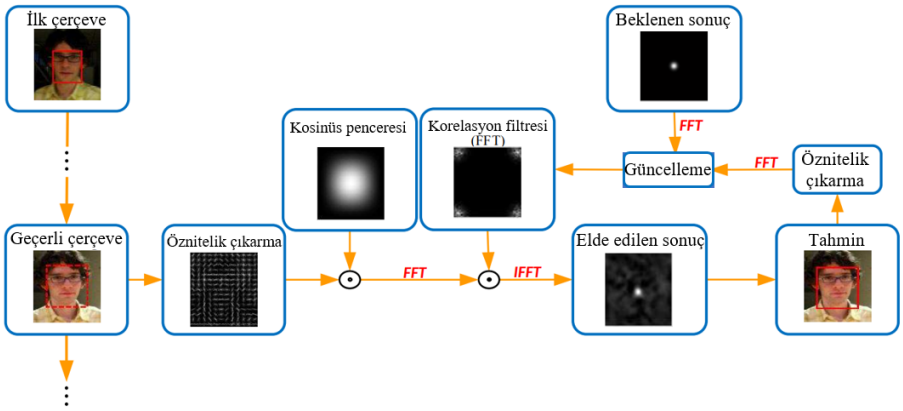
DVT (Destek vektör takibi - Support vector tracking) [46] en eski sınıflandırma tabanlı nesne takip yöntemlerinden birisidir. Bu çalışmada destek vektör makineleri (DVM) ile optik akış yöntemleri birleştirilerek destek vektör takip yöntemi önerilmiştir. Grabner vd. yaptığı [47] çalışmada DVT yöntemine ek olarak birden çok sınıflandırıcı kullanarak farklı bir bakış açısı kazandırmıştır. Burada temel amaç, sınıflandırıcıların elde etiği zayıf sonuçların birleştirilmesi ile daha güçlü bir sonuç elde edilmesidir.

Baben vd. [55] yaptıkları çalışmada nesne takibi için çevrimiçi haar öznitelikleri ile çoklu örnekle öğrenme tabanlı bir nesne takip yöntemi geliştirmiştir. Bu yöntemin temel avantajı, kullanılan sınıflandırıcının çoklu örnekle öğrenme (multiple instance learning) yöntemi ile daha başarılı bir güncelleme yapabilmesidir.

Cao ve Xue [56] nesne takibi için rastgele ormanlar (random forest) sınıflandırıcı kullanan bir yöntem uygulamıştır. Temel olarak karar ağaçları mantığıyla oluşturulan sınıflandırıcılar bağımsız olarak geçerli çerçevede nesne tespiti için bir tahmin üretmiştir. Daha sonra elde edilen sonuçların ortalaması alınarak nihai sonuca ulaşılmıştır. Bu çalışmada daha önce yapılan çevrim dışı Random Forest [52] yöntemine alternatif olarak çevrimiçi güncelleme eklenmiştir.

Zhang vd. [53] sıkıştırılmış görünüm özniteliklerini kullanan naive bayes sınıflandırıcı tabanlı FCT (Hızlı Sıkıştırmalı Takipçisi - Fast Compressive Tracking) yöntemi önermiştir. Nesne görüntü örneklerinden elde edilen öznitelikler sıkıştırılarak bir öznitelik vektörü elde edilmiştir. Ayni işlem arka plan görüntü örnekleri için de uygulanmıştır. Daha sonra bu vektörler yardımıyla çevrimiçi bayes sınıflandırıcısı eğitilmiştir. Sonraki çerçevede konum tespiti için alınan görüntü parçaları aynı şekilde vektör elde edildikten sonra sınıflandırıcı tarafından nesne ya da arka plan olarak ayrılmıştır. Nesne olarak kabul edilen görüntüler nesnenin yeni konumu olarak kabul edilmiştir. Sınıflandırıcı eğitimi ve güncellenmesi için ilk örnekler el ile olarak etiketlenmiştir. Sonraki çerçevelerde örnekler sınıflandırıcı tarafından etiketlenmiştir.

Nesne takip çalışmalarında önemli bir yere sahip bir diğer yöntem grubu ise korelasyon tabanlı yöntemlerdir. Bu yöntemlerde genellikle çevrimiçi veya çevrim dışı olarak eğitilen filtre, geçerli çerçevedeki nesne konumunu bulmak için bir sınıflandırıcı görevi görmektedir. Korelasyon filtrelerinin güncel tutulması için nesne takibi ve filtre eğitimi birlikte çalışır. Korelasyon tabanlı yöntemler hedef nesne görüntüsüne odaklanmış filtreyi, nesnenin aranacağı arama görüntüsü üzerinden ilişkilendirir. Burada DFT (Ayrık Fourier Dönüşümü - Discrete Fourier Transform) ile filtre ve öznitelikler, eleman bazlı çarpma işlemine tabii tutularak korelasyon haritası elde edilir. Genellikle uygulamada DFT hesaplaması için FFT (Hızlı Fourier Dönüşümü - Fast Fourier Transform) yöntemi kullanılmaktadır. Korelasyon haritasında maksimum değere karşılık gelen yer, hedefin yeni konumu olarak belirlenir. Daha sonra bu yeni konuma dayalı bir çevrimiçi güncelleme işlemi gerçekleştirilir. Korelasyon filtresine dayalı bu tür nesne takip çalışmaları günümüzde birçok araştırmacı tarafından oldukça ilgi görmektedir [18]. Korelasyon filtresi yönteminin temel çalışma prensibi Şekil 2.8'de gösterilmiştir. Burada ⊙ sembolü eleman bazlı çarpıma ve FFT ise hızlı hourier dönüşümü göstermektedir.



Şekil 2. 8. Korelasyon filtre tabanlı nesne takip yöntemlerinin iş akışı [18]

Korelasyon filtresi tabanlı nesne takip yöntemlerinde nesne görüntüsü filtre tasarımı için doğrudan kullanılır [18]. Buna karşı, bazı korelasyon filtresi tabanlı yöntemlerde nesne görüntüsünden, pozitif ve arka plan görüntüsünden, negatif örnekler alınarak korelasyon filtresi eğitilmektedir. ASEF (Ortalama Sentetik Tam Filtre - Average Of Synthetic Exact Filters) yöntemi [57] korelasyon filtre takipçilerine farklı bir yaklaşım getirerek, eğitilmiş farklı çoklu filtreleri kullanarak genel bir filtre yapısı oluşturmuştur. ASEF nesne takibi için çok sayıda örneğe ihtiyaç duyduğu için çevrimiçi uygulamalarda kullanışlı değildir [58].

Bolme vd. [59] yaptıkları çalışmada ASEF yönteminin daha kullanışlı ve gelişmiş bir versiyonu olan MOSSE (Karesel Hata Toplamının Minimum Çıkışı - Minimum Output Sum Of Squared Error) yöntemini önermiştir. Burada en uygun koleralasyon filtresini bulmak için MOSSE filtre çıktısı ile nesne takibinde istenilen filtre çıktıları arasında ortalama karesel hataların toplamını en aza indirgeme yöntemi kullanılmıştır. MOSSE yönteminde nesne

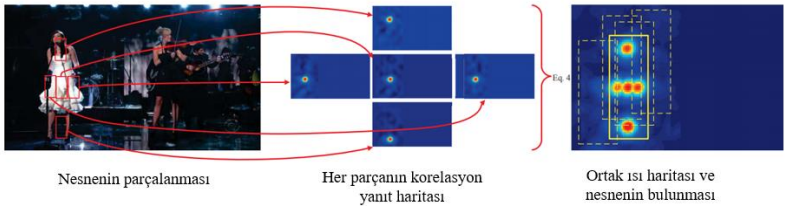
konumunu bulmak için elde edilen korelasyon ısı haritasının tepe noktası kontrol edilmektedir. Bu şekilde nesnenin kaybolması durumunda güncellemeler durdurularak nesne takibinde kayma problemi önlenmiştir.

Henriques vd. [58] doğrusal sınıflandırıcı yerine çekirdek sınıflandırıcılar kullanılarak daha güçlü ve doğru bir nesne takip yöntemi önermiştir. Bu çalışmada dolaşım matrisi teorisini kullanarak, hızlı fourier dönüşümü ile son derece hızlı öğrenme ve algılama imkânı sağlanmıştır. Çalışmada popüler gauss ve polinom çekirdeği dâhil olmak üzere çeşitli çekirdek türleriyle eğitim ve algılama için kapalı formlu çözümler üretmiştir. [60] çalışmasında [58] çalışmasına benzer şekilde KCF (Çekirdekleştirilmiş Korelasyon Filtresi - Kernelized Correlation Filter) yöntemi önerilmiştir. Günümüze kadar birçok yöntem KCF yönteminin alternatifi olarak geliştirilmiştir. Korelasyon filtresi formülünde yaptığı güncellemeler ile oldukça güçlü bir yöntem ortaya çıkmıştır.

Zhang vd. [61] yaptıkları çalışmada geliştirilen STC (Uzaysal-Zamansal Bağlam - Spatio-Temporal Context) nesne takip yöntemi bağlam bilgisini korelasyon filtrelerine uyarlamıştır. Bağlam bilgisi ile nesnenin etrafındaki önemli noktaların olasılıksal değerleri kullanılmış ve korelasyon filtresi eğitilmiştir. Bağlam bilgisinin kullanılması nesne kaybolması durumunda yöntemin kayma problemi ile karşı karşıya kalmasını engellemiştir. STC yöntemi hedef nesnenin boyut değişikliklerini tahmin edebilmektedir. Bu sayede nesne boyutunun değişim problemine karşı güçlü başarılar sağlamıştır.

Li vd. [62] yaptıkları çalışmada önerilen SAMF (Çoklu Özniteliklerle Ölçek Uyarlamalı - Scale Adaptive with Multiple Features) yöntemi, YGH ve renk adları özniteliklerini birlikte kullanarak nesne takibi yapmıştır. Uyarlamalı yöntemin bir diğer yeniliği ise MOSSE ve KCF yöntemlerinden farklı olarak nesne ölçeklerini tahmin etmek için bir arama stratejisi uygulamıştır. Burada temel olarak, farklı boyutlara sahip pencereler hedef etrafında örneklendirilmiş ve daha sonra bu örnekler korelasyon filtresi ile ilişkilendirilmiştir. En yüksek korelasyon puanı elde eden pencere işleme alınarak nesnenin yeni konumu ve durumu olarak kabul edilmiştir.

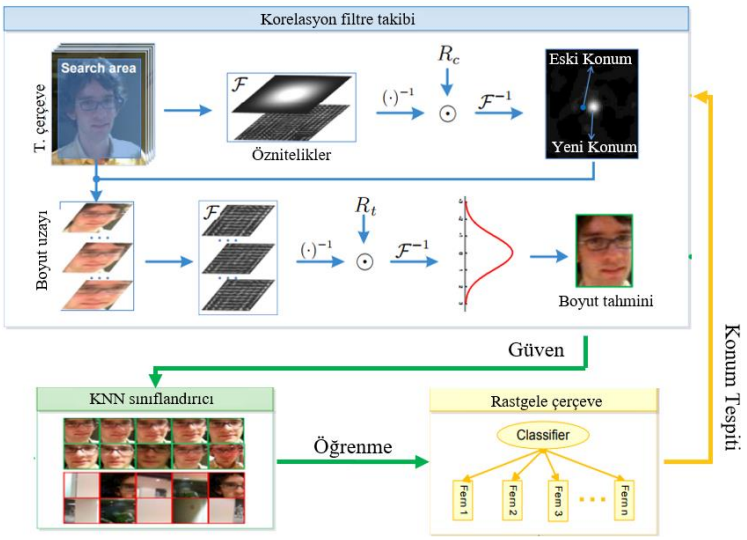
Bazı korelasyon tabanlı nesne takip yöntemleri, nesnenin tam bir görünüm modelini öğrenmek yerine, hedefin parçalı görünüm modelini öğrenme stratejisini uygulamıştır. Bu yöntemler nesne takip başarısını artırmayı hedeflemişlerdir. [63] çalışması nesneyi 5 parçaya bölerek her bir parçasını bir KCF yöntemi ile takip etmiştir. Yeni bir çerçeve geldiğinde her bir parçanın ayrı ayrı güven haritaları hesaplanmıştır (Şekil 2.9). Daha sonra nesnenin yeni durumunu tahmin etmek için parçacık filtresi yöntemini kullanarak bu sonuçlardan ortak bir sonuç haritası oluşturmuştur. Genel olarak bu şekilde kısmi nesne kaybolma durumlarına karşı sağlamlık kazanılmıştır.



Şekil 2. 9. [63] çalışmasında nesnenin 5 parçalı olarak izlenmesi

Çevrimiçi öğrenen nesne takip yöntemlerin karşılaştığı en büyük problemlerden bir tanesi nesneyi kaybettikten sonra nesnenin görünümünü unutmasıdır. Bu problemden yola çıkılarak [19] çalışmasında kısa ve uzun vadeli bellek modeli kullanılarak nesne takibine farklı bir bakış açısı kazandırılmıştır. Bu çalışmada önerilen MUSTer yöntemi insan beyninde kısa süreli bellek ve uzun süreli bellek birimlerinin çalışma prensibinden yola çıkılarak geliştirilmiştir. Kısa süreli bellekte korelasyon filtresi kullanılarak yerel ve zamansal bilgiler depolanmıştır. Uzun vadeli bellek ise genel ve güvenilir bilgilerden oluşan anahtar noktaları depolamıştır. Kısa süreli bellekte nesneyi kaybedilmeye başladığında uzun sureli bellek bu durumu engellemiştir.

[64] nolu çalışmada uzun süreli nesne takibi için KNN (En Yakın K Komşu - K Nearest Neighborhood) destekli korelasyon tabanlı LTC (Uzun Vadeli Korelasyon İzleme - Long-Term Correlation Tracking) yöntemi önerilmiştir. Korelasyon filtresi nesne konumunu belirledikten sonra ilgili konumdaki görüntü bir KNN sınıflandırıcısına gönderilmiştir. Burada tespit edilen aday nesnenin gerçekten nesneye ait olma durumu için bir güven değeri üretilir. Bu değere bakılarak bu görüntünün filtrenin güncellenmesi için kullanılıp kullanılmayacağına karar verilmiştir (Şekil 2.10).



Şekil 2. 10. LTC yönteminin genel görünümü [64]

Danelljan vd. [65] yaptıkları çalışmada daha güçlü korelasyon sağlamak için SRDCF (Mekânsal Olarak Düzenli Ayrımcı Korelasyon Filtreler - Spatially Regularized

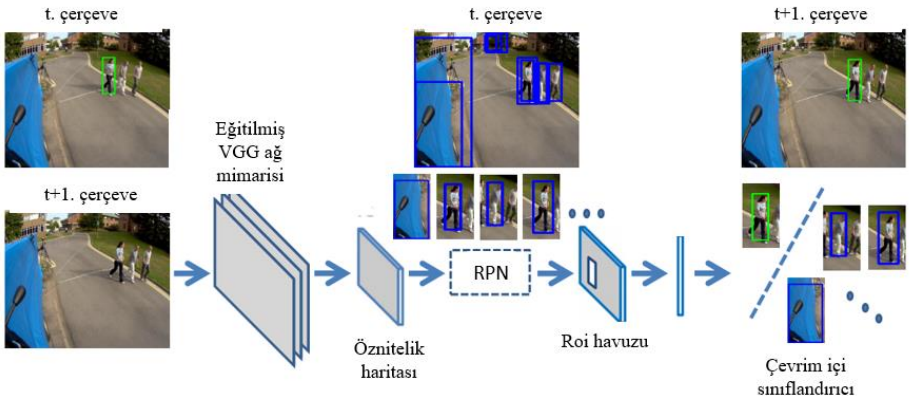
Discriminative Correlation Filters) yöntemini önermiştir. SRDCF yönteminde normalde kullanılan korelasyon filtre tabanlı nesne takip yöntemlerinin formülasyonunda bazı güncellemeler yaparak korelasyon filtresinin eğitiminde daha büyük görüntü bölgelerinde öğrenme işlemi sağlanmıştır. Bu şekilde arka plan örnekleri nesne örneklerinden daha fazla kullanılarak nesne ve arka plan ayrımını daha da artırmayı hedeflemiştir.

Galoogahi vd. [66] SRDCF yöntemine benzer bir yapıya sahip ve arka planı detaylı öğrenen BACF (Arka Plana Yönelik Korelasyon Filtre Öğrenme - Learning Background Aware Correlation Filters) yöntemi önermişlerdir. Çerçeveler boyunca nesne ve arka planının nasıl değiştiğini etkin bir şekilde modelleyen bu yöntem, zor arka plan karmaşasına karşı güçlü bir performans göstermiştir. BACF yönteminin zayıf yönü ise kapanma (occlusion) problemi olma durumuna karşı pozitif örneklerin güvenilirliğinin hesaba katılmamış olmasıdır. Eğer hedef kapanma problemine uğrarsa pozitif örnekler bazı arka plan bilgisi içereceğinden sonraki çerçevelerde kayma problemi olabilir [18].

[33] nolu çalışmada nesne takibi için derin öznitelikleri kullanan korelasyon tabanlı nesne takip yöntemi önerilmiştir. Bu çalışmada, derin öğrenme için geliştirilen ağ mimarilerinin her katmanında elde edilen bilgilerin nesne takibinde önemli olduğu savunulmaktadır. Bu fikir doğrultusunda VGG-Net [32] ağ mimarisinin üç farklı katmanı (Conv3-4, Conv4-4 ve Conv5-4) ile üç korelasyon filtresi birleştirilmiştir. Bu oluşturulan korelasyon filtreleri ile elde edilen sonuçlar belli bir fonksiyondan geçirilerek nihai sonuç elde edilmektedir. Buradaki temel amaç ağ mimarisinin son katmanlarındaki güçlü ayırt edici derin özniteliklerin yanında ilk katmanlardaki mekânsal öznitelikleri de kullanmaktır. Bu şekilde daha güçlü derin öznitelikleri kullanan güçlü bir korelasyon filtre yapısı oluşturulmuştur. [67] çalışmasında, [33] çalışmasına benzer şekilde çalışan Hedged Deep Tracking yöntemi geliştirilmiştir. Yöntemin genel adımları şu şekilde sıralanmıştır: Öncelikle çerçeve görüntüsü önceden eğitilmiş VGG-Net ağ mimarisinden geçirilmiştir. Burada her katman için farklı derin öznitelikler çıkartılmıştır. Önceden eğitilmiş korelasyon filtreleri ile bu derin öznitelikleri kullanarak her katmanda bireysel nesne konumunu gösteren ısı haritaları elde edilmiştir. Ayrı ayrı hesaplanan ısı haritaları daha gelişmiş bir nesne takip sonucu elde etmek için Hedge algoritmasına verilmiştir. Hedge algoritması önceki adımda elde edilen zayıf sonuçları birleştirilerek daha doğru bir sonuç değeri elde etmiştir.

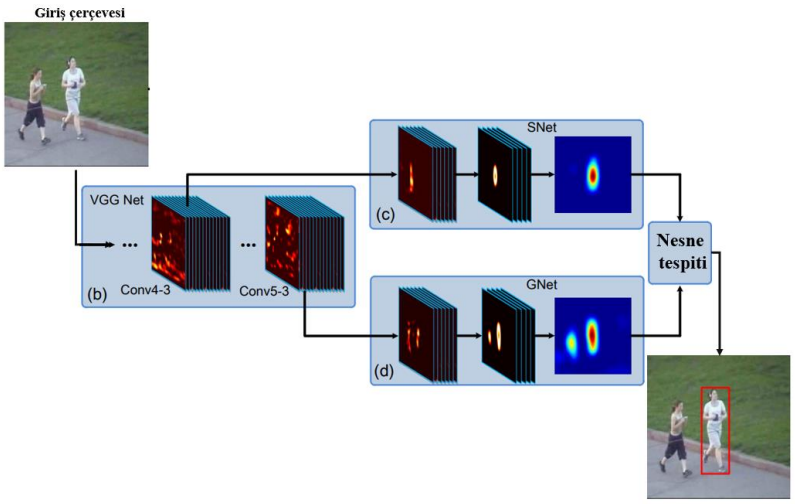
[68] nolu çalışmada derin öğrenme ile DVM sınıflandırıcıyı bir arada kullanan RPNT (Bölge Öneri Ağ Takipçisi - Region Proposal Network Tracker) yöntemi geliştirilmiştir. Yeni gelen bir çerçevede öncelikle VGG-Net ağ mimarisi kullanılarak çerçeveye ait derin öznitelikler hesaplanmıştır. Daha sonra elde edilen öznitelikler nesne tespitinde hızlandırılmış bir R-CNN [69] ağ mimarisinden geçirilmiştir. Burada R-CNN ağ mimarisi geçerli çerçevedeki nesnenin aday konumlarını belirlemiştir. Şekil 2.11'de görülüğü gibi belirlenen

aday nesne konumları bir DVM sınıflandırıcı yardımıyla ile en uygun nesne konumu tespiti için kullanılmıştır.



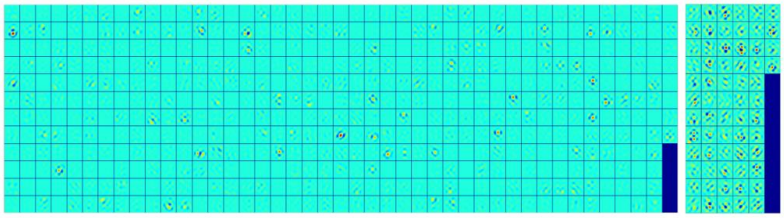
Şekil 2. 11. RPNT yönteminin genel görünümü [69], RPN: Bölge Öneri Ağı

Wang vd. [14] yaptıkları çalışmada FCNT (Tam Bağlı Ağ Mimarisi ile Nesne Takibi - Fully Convolutional Network Based Tracker) nesne takip yöntemi 3 farklı ağ mimarisini birbirine bağlayarak nesne takibi yapmıştır. Öncelikle geçerli çerçeve görüntüsü derin öznitelikleri hesaplanmak üzere VGG-Net [32] ağ mimarisine gönderilmiştir. Daha sonra Şekil 2.12'de görüldüğü gibi VGG-Net ağ mimarisinin Conv4-3 ve Conv5-3 katmanlarında elde edilen derin öznitelikler sırasıyla oluşturulan SNet ve GNet ağ mimarine göndermiştir. Bu ağ mimarileri ile derin öznitelikler kullanılarak her bir nesnenin bulunabileceği konumun ısı haritası elde edilmiştir. En son olarak bu ısı haritası belli bir oranlama fonksiyonu ile birleştirilerek nihai nesne konumu tespit edilmiştir. Bu yöntemde kullanılan VGG-Net, GNet ve SNet ağ mimarileri çevrimdışı olarak eğitilmiştir. Bunun yanında çerçeve boyunca ortaya çıkan arka plan gürültüsünden kaçınmak için GNet ve SNet mimarilerinin ağ parametreleri çevrimiçi olarak güncellenmiştir.



Şekil 2. 12. FCNT yönteminin genel görünümü [32]

Danelljan vd. [70] derin ayırt edici korelasyon filtre yöntemlerinde karşılaşılan 3 farklı problemi ele alıp çözmeye çalışan ECO (İzleme için Verimli Konvolüsyon Operatörleri - Efficient Convolution Operators for Tracking) yöntemi önermişlerdir. Bu çalışmada kullanılan derin özniteliklerin sayısının fazla olduğu ve bunun da hesaplama yükü getirdiği savunulmuştur. Bu doğrultuda hesaplama maliyetini düşürmek ve gereksiz derin özniteliklerin azaltmak için geliştirilen ECO yöntemi uygulanmıştır. Şekil 2.13'te görüldüğü gibi ECO yönteminde kullanılan gereksiz derin öznitelikler elenerek öznitelik sayısında %80'lık bir düşüş sağlanmıştır. İkinci problem olarak eğitim setindeki çerçevelerden alınan ardışık nesne örneklerin kullanılmasıdır. Bu şekilde alınan örneklerin gereksiz eğitim işlemlerine ve yüksek bellek gereksinimine yol açtığı belirtilmiştir. Ardışık eğitim örnekleri kullanmak yerine bu örnekleri Gauss bileşenlerinin birleşimi olarak modelleyen bir yöntem geliştirilmiştir. Bu şekilde her bileşen nesnenin farklı bir yönünü temsil etmiştir. Karşılaşılan son problem ise derin ayırt edici korelasyon filtresi yöntemlerin her çerçevede güncelleme işlemlerini yapmasıdır. Sık sık yapılan güncelleme işlemlerinin aşırı öğrenmeye sebep olabileceği belirtilmiştir. ECO yönteminde bunun yerine her 5 karede bir güncelleme işlemleri gerçekleştirmiştir.



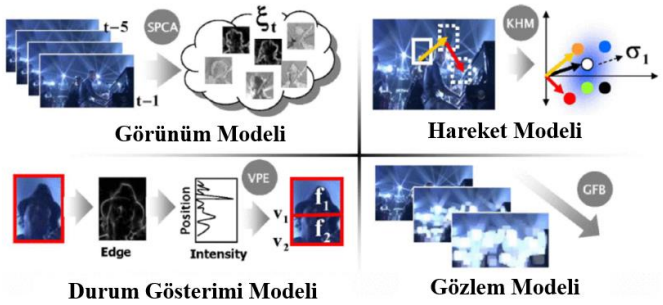
Şekil 2.13. a) Geleneksel derin öğrenme ile elde edilen öznitelikler, b) ECO yöntemi ile %80 oranında azaltılan öznitelikler.

2.4. Hareket Modeli

Nesne hareket modeli, bir nesnenin nasıl hareket ettiğini tanımlayan dinamik model olarak da bilinmektedir [4]. Nesne takibinde hareket modeli, temel olarak hedef nesnenin gelecekteki çerçevede potansiyel konumunu tahmin etmek için kullanılır. Bu şekilde nesnenin kaybolması (kapanma-occlusion) durumunda bile nesnenin olası konumu belirlenebilmektedir. Ayrıca potansiyel konumlar kullanılarak gelecekteki çerçevede arama alanı azaltılabilir veya daha doğru bir arama alanı çıkartılabilir. Bazı çalışmalarda [11] hareket modeli gürünüm modeli kadar önemli görülmese de özellikle kapanma veya ani ışık değişim problemleri olan senaryolarda hareket modeli nesnenin takip edilmesi için güçlü bir çıkarım yapabilmektedir. Hareket modeli için en yaygın kullanılan yöntemler parçacık filtresi ve kalman filtresi gibi olasılıksal yöntemlerdir [28]. Nesne takibinde genellikle hareket modeli gözlem modeline dâhil edilerek kullanılır. Yakın zamanda geliştirilen yöntemlerde daha çok

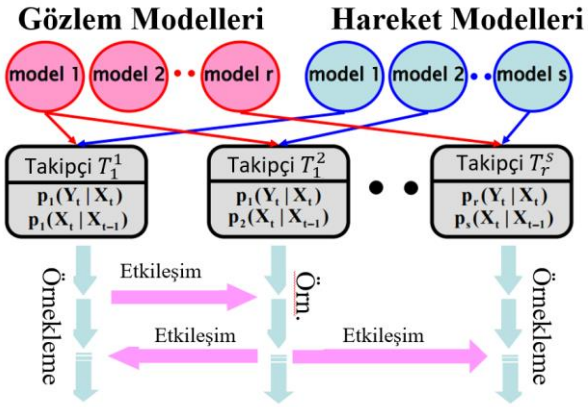
gözlem modeline odaklanılmışsa da ADNet (Eylem Karar Ağları - Action Decision Network) [71] gibi güncel yöntemler hareket modeline dayanan öğrenme tabanlı yöntemlerdir.

Kwon ve Lee [72] yaptıkları çalışmada nesne takibinde çoklu izleyici kullanan ve her izleyici için çoklu model barındıran farklı bir yaklaşım önerilmiştir. Şekil 2.14'de görüldüğü gibi yöntemin içinde bulunduğu her parça izleyici kendi içinde görünüm modeli, hareket modeli, durum gösterimi ve gözlem modeli olmak üzere dört bileşenden oluşmuştur. Görünüm modeli, nesnenin gözlem bilgisini kullanarak aday konum ile ne kadar örtüştüğünü göstermiştir. Hareket modeli, nesnenin önceki çerçevelerdeki hareket bilgisini modelleyerek bir sonraki çerçevelerde durum tahmini yapmıştır. Bu şekilde çerçeve geçişlerinde olası konumlar belirlenmiştir. Durum gösterimi nesnenin çerçevelerdeki merkez konumu, ölçeği ve mekânsal bilgilerini saklamıştır. Gözlem modeli ise video içerisindeki görsel özellikleri modellemektedir. Test zamanında yöntemin içinde bulunduğu her parça izleyici kendi içinde nesnenin yeni konumunu bulmaya çalışır. Daha sonra her bir izleyicinin bulduğu konumları olasılıksal olarak en doğru bulan izleyicinin tahminini geçerli çerçevede nesne konumu olarak seçilir.



Şekil 2. 14. [72] 'de kullanılan her bir izleyicinin içerdiği modeller

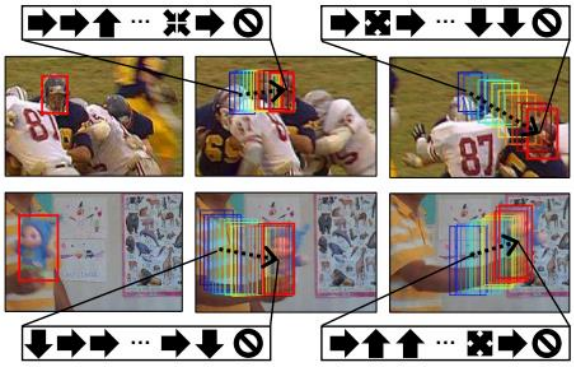
Kwon ve Lee [73] yaptıkları bir diğer çalışmada, temel olarak çoklu gözlem ve çoklu hareket modellerini belirli işlemler ile birleştirerek etkin bir izleme şeması oluşturmuştur. Gözlem modelleri birden fazla temel gözlem modeline ayrıştırılır ve her alt gözlem model, nesnenin belirli bir görünümünü ele alır. Hareket modeli ise her biri farklı bir hareket türünü kapsayan birden çok temel hareket modelinden oluşmuştur. Ardından her bir gözlem modeli ile bir hareket modeli ilişkilendirilerek çoklu temel izleyiciler tasarlanmıştır. Böylece her izleyici nesnenin belirli bir değişiminden sorumlu olmuştur. Şekil 2.15'de gösterildiği gibi en sonunda bütün temel izleyiciler birleşerek etkileşimli Monte Carlo Markov Zinciri çerçevesi oluşturmuştur. Bu şekilde her izleyici diğer izleyiciler ile bilgi alışverişinde bulunarak güçlü bir izleyici topluluğu elde edilmiştir.



Şekil 2. 15. Kwon ve Lee'nin önerdiği çoklu gözlem ve çoklu hareket modellerini birleşim çerçevesi [73]

Ratnayake ve Amer [74] yaptıkları çalışmada önerilen yöntem ile nesne takibi uygulamalarına farklı bir bakış açısı sağlanarak dinamik hareket modeli ile DVM birlikte kullanmıştır. Bu çalışmada hareket modeli iki aşamalı olarak tasarlanmıştır. Birinci adımda nesnenin önceki çerçevede alınan konumsal bilgileri geliştirilen Kernelize Harmonik algoritmasına verilerek gelen çerçevede aday nesne konum tahminleri yapılmıştır. Daha sonra elde edilen tahmini konumları daha da optimize etmek için parçacık filtresi uygulanmıştır. İkinci adımda hareket modeli ile elde edilen güçlü olasılıksal konumlar DVM uygulanarak ağırlıklandırılmıştır. En son olarak bu değerler aktivasyon fonksiyonundan geçirilerek nihai nesne konumu belirlenmiştir.

Yun vd. [71] nesne takibinde nesnenin yapabileceği muhtemel hareket bilgisi kullanılarak nesne takibi yapabilen yeni bir yöntem geliştirmiştir. ADNet yönteminde nesnenin yapabileceği yön (sol, sağ, yukarı, aşağı) ve boyut (küçülme ve büyüme) hareketleri derin öğrenme ile öğrenilerek nesneyi gösteren sınırlama kutusu hareketlendirilmiştir. Bu hareketlendirme Şekil 2.16'da gösterildiği gibi sınırlama kutusunun çerçeveler boyunca nesnenin konumuna taşınmasını sağlamıştır. Nesnenin yapabileceği hareketleri öğrenmek için oluşturulan ağ, çevrim dışı olarak eğitilmiştir. Bunun yanında da nesne takip doğruluğunu artırmak için çevrimiçi olarak güncellemeler yapılmıştır.



Şekil 2. 16. ADNet çalışma prensibi [60].

3. GELENEKSEL YÖNTEMLER İLE DERİN ÖĞRENME TABANLI NESNE TAKİP YÖNTEMLERİNİN İNCELENMESİ

Son zamanlarda derin öğrenme algoritmaları kullanılarak güçlü nesne takip yöntemleri geliştirilmiştir. Bu yöntemlerde genellikle ardışık çerçeve görüntülerinin ilk katmanlarda çıkartılan derin öznitelikleri birbirine tam bağlı ağ katmanlarında geçirilerek nesne takibi yapılmaktadır. Öte yandan geleneksel makine öğrenmesi ve örüntü tanıma yaklaşımlarını kullanan güçlü nesne takip yöntemleri de mevcuttur. Özellikle insan görme sistemini modelleyen BIT [75] yöntemi veya insan hafıza sisteminden yola çıkarak oluşturulan MUSTer [19] yöntemi literatürde bulunan geleneksel yöntemlere oldukça güçlü örnek teşkil eden çalışmalardır. Güncel ve geleneksel yöntemlerin kendi içinde farklı avantajları/dezavantajları bulunmaktadır. Tez çalışmasının bu bölümünde 5 tane güncel derin öğrenme yöntemi ve 5 tane de geleneksel makine öğrenmesi yöntemlerini kullanarak nesne takibi yapan toplam 10 farklı yöntemin temel adımları detaylı olarak ele alınmıştır. Derin öğrenme tabanlı nesne takip yöntemlerinin daha iyi anlaşılması için, ilk olarak derin ağ mimarisi hakkında bazı önemli bilgilendirmeler yapılmıştır. Daha sonra ele alınan 10 yöntem hakkında bilgilendirmelere yer verilmiştir. Yöntemler ele alınırken özellikle yöntemlerin içerdiği temel mantıksal çerçeve ve önemli matematiksel ifadelere değinilmiştir. Bu bölümde ele alınan yöntemler Çizelge 3.1'de verilmiştir.

Çizelge 3. 1. Geleneksel ve güncel nesne takip yöntemleri

	Yöntem	Çalışma İsmi	Yıl	Kaynak
Güncel Derin Öğrenme Tabanlı Yöntemler	**GOTURN**	Learning to Track at 100 FPS with Deep Regression Networks [13]	2016	13
	SiamFC	Fully-Convolutional Siamese Networks for Object Tracking [15]	2016	15
	ADNet	Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning [71]	2017	71
	DCFNet	Discriminant Correlation Filters Network For Visual Tracking [45]	2017	45
	HCFT	Hierarchical Convolutional Features for Visual Tracking [33]	2015	33
Geleneksel Makine Öğrenmesi Tabanlı Yöntemler	**BIT**	Biologically Inspired Tracker [75]	2016	75
	MUSTer	MUlti-Store Tracker (MUSTer): a Cognitive Psychology Inspired Approach to Object Tracking [19]	2015	19
	BACF	Learning Background-Aware Correlation Filters for Visual Tracking [66]	2017	66
	MEEM	Robust Tracking via Multiple Experts Using Entropy Minimization [76]	2014	76
	FCT	Fast Compressive Tracking [53]	2014	53

3.1. CNN Ağ Mimarilerinin Genel Yapısı

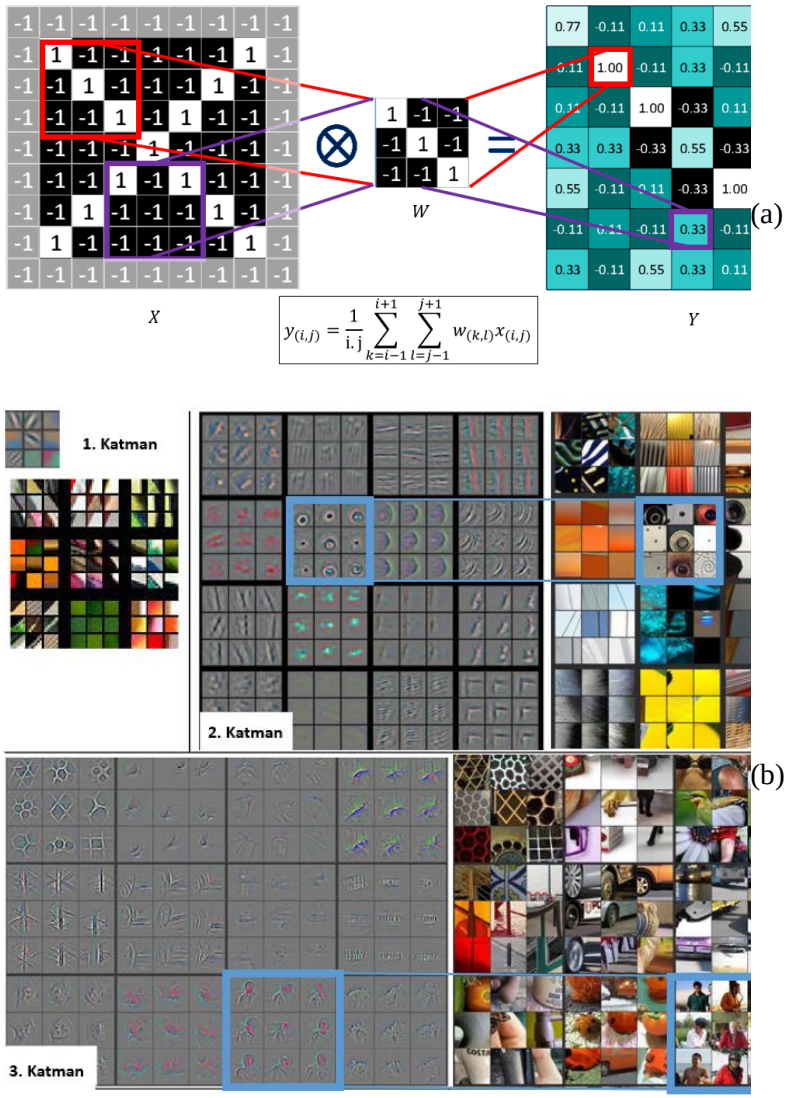
Bir görüntü içerisindeki nesneleri tanımlarken öncelikle ortamda bulunan nesnenin şeklini, boyutunu ya da rengini gösteren özellikler çıkartılır. Daha sonra bu özellikler doğrultusunda nesneler belirlenir ya da gruplandırılır. Bilgisayar ortamında nesnelerin özellikleri genellikle farklı filtreler kullanılarak elde edilir. Geleneksel makine öğrenmesi ve görüntü işleme yöntemlerinde, bu filtreler elle hazırlanmış veya matematiksel fonksiyonlar modellenerek oluşturulmuştur (örn. Gaussian / Laplacian / Canny [77]). Günümüzde oldukça popüler olan derin öğrenme tabanlı CNN ağ mimarisi (Konvolüsyonel Sinir Ağları - Convolutional Neural Network), bu filtreleri manuel olarak çıkartmak yerine otomatik bir şekilde çıkarmaktadır.

Bir CNN ağ mimarisi için filtreler, nöronların birbirine bağlanış biçimini göstermektedir. Filtre içerisindeki parametre ise nöronların katsayısını belirtmektedir. Örneğin bir CNN ağ mimarisinin bir katmanındaki her 9 nöron sonraki katmandaki bir nörona bağlanıyor ise bu katmanda 3×3 boyutunda bir filtre uygulandığı varsayılmaktadır. Ayrıca bu filtredeki elemanlar 9 nöronun ağırlıklarını göstermektedir. Yapılan çalışmalarda CNN ağ mimarileri ile otomatik bir şekilde elde edilen filtrelerin geleneksel yöntemler ile hazırlanmış filtreler kıyasla daha güçlü öznitelikler çıkardıkları tespit edilmiştir [34], [78]. CNN ağ mimarileri temel olarak konvolüsyon, pooling ve tam bağlı katmanlardan oluşmaktadır.

3.1.1. Konvolüsyon Katmanı

Bu katman, CNN mimarilerinde derin özniteliklerin çıkarıldığı katman olarak da bilinmektedir [15], [45]. Bu katmanda yapılan en temel işlem ele alınan filtrelerin önceki katmandan gelen veriler üzerinde gezdirilmesi ile bir filtreleme işleminin uygulanmasıdır. Bu işlem sonucunda derin öznitelikler elde edilir. Ağ mimarilerinde kullanılan filtreler diğer bir adıyla ağ parametreleri, CNN ağ mimarisinin yüksek boyuttaki verileri ile güçlü bir eğitim sonucunda elde edilmektedir.

Konvolüsyon (Conv) katmanında yapılan işlemler ve kullanılan denklemler Şekil 3.1.a'da gösterilmiştir. Gelişmiş bir CNN ağ mimarisinin ilk katmanlarında görüntü içerisindeki kenar veya köşe özniteliklerini çıkartan filtreler kullanılmaktadır. Daha sonraki katmanlarda ise görüntülerdeki nesnelerin veya ortamın bilgilerini daha detaylı ortaya çıkartan (doku öznitelikleri) filtreler kullanılmaktadır (Şekil 3.1.b).



Şekil 3. 1. CNN ağ mimarilerinin conv katmanlarında öğrenilen örnek filtreler (b) [34] ve bunların uygulanma biçimi (a) [79]

Conv katmanında elde edilen çıkış değerleri $W \times H \times D$ (genişlik × yükseklik × derinlik) şeklinde 3 boyutlu bir matristir. Bu matrisin boyutu sırasıyla *derinlik (depth), adım (stride)* ve *genişletme (padding)* parametreleri ile doğrudan ilişkilidir.

Derinlik: CNN ağ mimarisinde kullanılan filtrelerin sayısını göstermektedir. Bunun yanında kullanılan filtre sayısı giriş verisinden çıkartılan derin özniteliklerin sayısını vermektedir. Kullanılan her bir filtre giriş verisinin farklı bir özelliğini çıkartmaktadır. Derinlik yani filtre sayısı çıkış matrisi boyutunun boyutunu (D) vermektedir.

Adım: Filtrenin her bir işlemden sonra ne kadarlık bir kayma yapacağını belirtir. Diğer bir deyişle nöronların bağlanış biçimini göstermektedir. Şekil 3.1.a'da filtre bir piksel kaydırılarak devam etmektedir. Bunun yanında adım aralığı arttırılarak derin öznitelik sayısı azaltılabilir. Adım aralığı çıkış matrisi boyutunun ($W \times H$) boyutunu etkilemektedir.

Genişletme: Filtrenin köşegen elemanlarının değerinin hesaplanması için varsayılan sıfır elemanlarıdır. Şekil 3.1.a giriş verisinin kenarlarına bir piksellik genişletme uygulanmıştır. Bazı durumlarda Genişletme**,** katman çıkışı boyutunu ayarlamak içinde kullanılmaktadır.

3.1.2. Pooling Katmanı

Bu katmanın temel amacı parametre sayısının kontrollü bir şekilde azaltmaktır. Bir pooling katmanında yapılan işlemler en temel anlamıyla belirli bir alandaki değerlerin en büyüğünü veya ortalamasını alma şeklindedir. Pool katmanında en yoğun kullanılan fonksiyon en büyük değeri alma (max pooling) fonksiyonudur. Bu fonksiyon ele alınan 2×2 veya 3×3 lük bir matriste karşılık gelen elemanlar arasından en büyük değeri çıkartır. Daha sonra conv katmanında olduğu gibi ele alınan matris adım genişliği (stride) kadar kaydırılarak bu işleme devam edilir.

Bir CNN ağ mimarisindeki değişkenleri kontrol altına almak ve eğitim zamanını düşürmek için ReLu ve Dropout katmanları kullanılmaktadır. ReLu katmanı gelen veriler arasında negatif olan değerleri 0'a yükseltmektedir. Dropout katmanı ise katmanlar arasındaki bağlantı sayısını azaltmak için bazı nöronların kaldırılması işlemini gerçekleştirir.

3.1.3. Tam Bağlı katman

Tam bağlı katmanların temel işlevi conv katmanlarında elde edilen derin özniteliklerin değerlendirilmesi ve sonuçlandırılmasıdır. İlk olarak bu katmanların girişinde elde edilen öznitelikler tek boyutlu dizi olacak şekilde yeniden boyutlandırılır [80]. Daha sonra elde edilen tek boyutlu matrisin tüm elemanları tam bağlı katmanlara bağlanır. Ağ mimarisinin eğitimi boyunca bu katmanlardaki nöronların ağırlıkları en doğru tahmini verecek şekilde eğitilir. Tam bağlı katmanlardaki işlemler geleneksel yapay sinir ağ mimarilerindeki temel mantıkla aynı çalışmaktadır.

3.1.4. Derin Ağ Mimarisinin Eğitimi

Bir CNN ağ mimarisinin parametrelerinin en doğru tahmini verebilmesi için ağın oldukça güçlü ve yüksek boyutta bir veri kümesi ile eğitilmesi gerekmektedir. Bu eğitim boyunca bir kayıp fonksiyonu ile CNN mimarisinin conv ve tam bağlı katmanlarındaki en uygun ağırlıkların bulunması gerekmektedir. Her katmandaki nöronların sayısı düşünüldüğünde, bulunması gereken binlerce parametre olabilmektedir. Fakat günümüzde gelişen teknoloji sayesinde yüksek hızlarda çalışan GPU birimleri kullanılarak bu problemin üstesinden gelinmiştir [34].

Bir CNN ağ mimarisinin eğitimi temel olarak şu adımlardan oluşmaktadır: İlk olarak kullanılan hata fonksiyonu yardımıyla ağ mimarisinin tahmin ettiği sonuç ile gerçek sonuç arasındaki hata değeri hesaplanır. Daha sonra hesaplanan hata değeri yardımıyla ağ parametreleri güncellenir. Ağ parametrelerinin güncellenmesi bir optimizasyon problemi olarak ele alınmaktadır. CNN ağ mimarilerinde parametrelerin güncellenmesi için en yaygın kullanılan optimizasyon yöntemi SGD (Olasılıksal Dereceli Azalma - Stochastic gradient descent) yöntemidir. SGD yöntemi hakkında daha detaylı bilgi Ruder [81]tarafından yapılan çalışmadan edinilebilir.

Literatürde bulunan ve yaygın bir şekilde kullanılan ağ mimarileri CaffeNet [82], VGG-Net [32], ALexNet [83] ve GoogLeNet [84] şeklinde sıralanabilir. Birçok çalışmada ele alınan problem için bu ağ mimarilerinin ilk katmanları kullanılarak derin öznitelikler elde etmiştir [13], [15], [45].

3.1.5. Korelasyon Filtresi ile Nesne Takip

Nesne tespit/takip problemlerinde korelasyon filtresi son zamanlarda oldukça yaygın bir şekilde kullanılmaktadır. Korelasyon filtresi yönteminin yapısındaki kıyaslama biçimi ve elde ettiği ısı haritası, nesne takibinde ikili sınıflandırıcılara göre daha yüksek başarılar kazanmasını sağlamıştır.

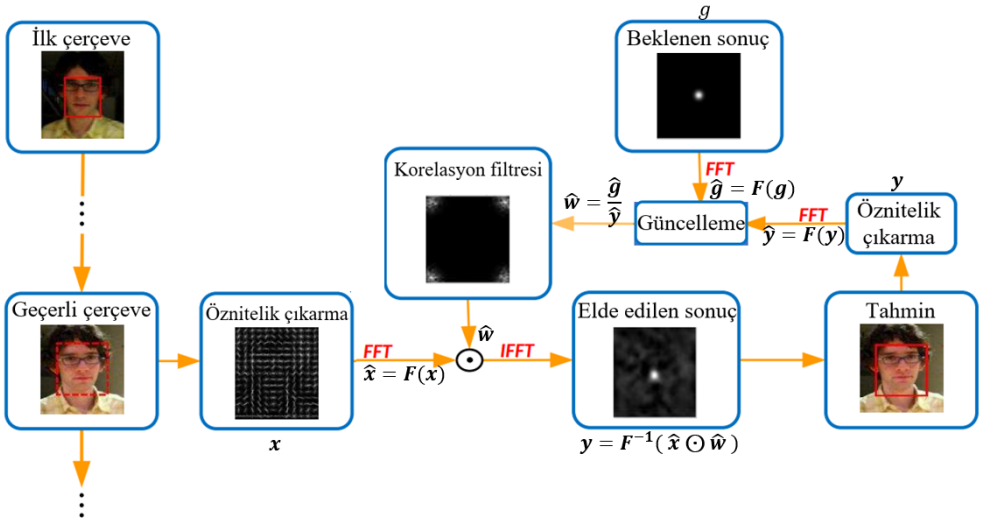
Korelasyon filtresi tabanlı nesne takip yöntemlerinin genel çalışma prensibi şu adımlar ile özetlenebilir: İlk olarak algoritmanın giriş elemanı olarak nesnenin arama alanı seçilir ve bu arama alanının öznitelikleri çıkartılır. Genellikle bu arama alanı önceki çerçevede nesne merkezde iken ve bu nesneden belirli bir oranda daha büyük olan bir dikdörtgen alan şeklindedir. İkinci adımda Ayrık Fourier Dönüşümü (DFT - Discrete Fourier Transform) kullanılarak korelasyon filtresi (w) ile arama alanı öznitelikleri (x) arasında eleman bazlı bir çarpma ($\odot$ sembolü ile gösterilir) işlemi gerçekleştirilir.

Kullanılan korelasyon filtresi eğitim boyunca nesne görüntüsüne odaklanacak şekilde eğitilmiş bir filtredir. Eleman bazlı çarpım işleminin sonucuna Ters Hızlı Fourier Dönüşümü (IFFT, Inverse Fast Fourier Transform) uygulanarak şablon ve arama alanı arasındaki ilişkiyi gösteren tepki (ısı) haritası elde edilmektedir. Bu ısı haritasındaki değerler nesne ile arama görüntüsü arasında mekânsal benzerlik matrisidir. Isı haritasında en yüksek değeri veren konum arama alanındaki nesne ile en çok benzeyen konum olarak değerlendirilir. Dolasıyla bu konum hedef nesnenin arama alanındaki yeni konumu olarak atanmaktadır.

Korelasyon filtre yöntemlerinde nesne boyutundaki değişimi yakalamak için genellikle ölçeklendirilmiş korelasyon çıktısı tekniği kullanılmaktadır. Bu yöntemde temel olarak nesne giriş görüntüsü üzerinde belirli bir oranda ($\%98, \%99.01, \%100, \%101, \%102$ gibi) boyut değişimi yapılır. Daha sonra bu değiştirilmiş görüntüler birçok korelasyon işlemine tabi

tutulur. Elde edilen farklı korelasyon çıktıları kıyaslanarak en yüksek değeri veren korelasyon çıktısı seçilir. Son olarak seçilen korelasyon çıktısı hangi görüntüye ait ise giriş boyutunda yapılan değişim benzer oranda nesne boyutuna uygulanılır. Bu şekilde nesnenin yeni boyutu hesaplanmaktadır.

Nesne tespit işlemlerinden sonra korelasyon filtresinin güncellenmesi işlemleri gerçekleştirilir. En temel anlamıyla bu adımda bir hata fonksiyonu kullanılarak elde edilen ısı haritası ile istenilen (nesnenin gerçek konumu veren) ısı haritası (y) arasındaki hata değeri hesaplanır. Bu hata değerine bakılarak korelasyon filtresinde güncellemeler yapılmaktadır. Yukardaki adımlar Şekil 3.2'de kullanılan denklemler ile birlikte verilmiştir. Burada giriş elemanı x (arama alanının özniteliği), korelasyon filtresi w, elde edilen sonuç ısı haritası y ve beklenen sonuç g (gerçeklilik değerine göre) şeklinde gösterilmiştir. Değişkenin FFT ile dönüşüm fonksiyonu $F(\cdot)$ ile gösterilmiş ve çıkan değişken $'\wedge'$ işareti ile gösterilmiştir. Aynı şekilde IFFT fonksiyonu $F^{-1}(\cdot)$ şeklinde gösterilmiştir.



Şekil 3. 2. Korelasyon filtresi tabanlı nesne takip yöntemlerinin genel yapısı ve adımlarda kullanılan genel formüller

Korelasyon filtresi tabanlı nesne takip yöntemleri kullandıkları öznitelik, hata fonksiyonu ve güncelleme fonksiyonu gibi işlemlerdeki farklılıklar nedeni ile birbirlerinden ayrılmaktadırlar. Örneğin günümüzde en çok bilinen KCF (Çekirdekleştirilmiş İlinti Süzgeci - Kernelized Correlation Filter) adlı nesne takip yöntemi çekirdek tabanlı bir hata ve güncelleme fonksiyonu kullanmıştır [60]. Bunun yanında nesne takibinde son zamanlarda oldukça etkili sonuçlar elde eden korelasyon tabanlı DCFNet [45] ve SiamFC [15] gibi yöntemler ise derin öznitelikler kullanmışlardır.

Derin ağ mimarisinin çalışma prensipleri hakkında yukarıda verilen bilgilerden sonra, bu noktadan itibaren öncelikle derin öğrenme yaklaşımı kullanarak nesne takibi yapan 5 farklı

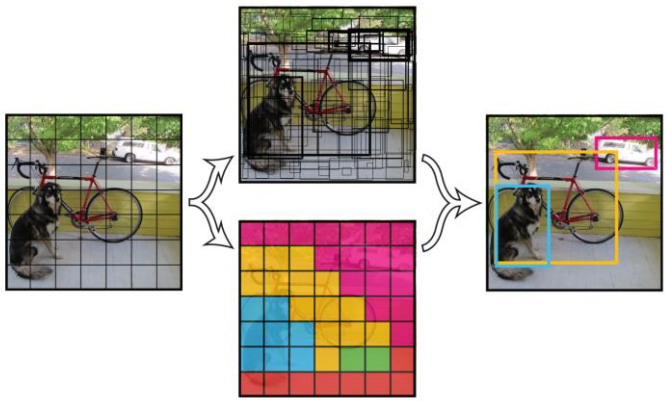
yöntem ile ilgili daha detaylı bilgiler sunulacaktır. Daha sonra geleneksel makine öğrenmesi yaklaşımları ile nesne takibi yapan 5 farklı yöntem hakkında bilgilere yer verilecektir.

3.2. GOTURN

Held vd. [13] yaptıkları çalışmada yüksek hızlarda nesne takibi yapan GOTURN (Regresyon Ağları Kullanarak Genel Nesne Takibi - Generic Object Tracking Using Regression Networks) ağ mimarisini önermiştir. Bu ağ mimarisi çevrim dışı olarak yüksek boyutlu bir veri seti ile eğitilmiştir. Bu sayede ağ mimarisi eğitim boyunca çerçevelerdeki görüntü değişimi ile nesne hareketi arasındaki genel bir ilişkiyi öğrenmiştir. Test zamanında çevrimiçi herhangi bir güncelleme yapmadan nesne takibinde 100 FPS (Saniyedeki Kare Hızı - Frame Per Second) hızına ulaşmıştır [13].

GOTURN ağ mimarisi regresyon tabanlı bir ağ mimarisidir. Bu ağ mimarileri görüntü içeresindeki birçok farklı sınıfa ait nesnelerin konumunu ve nesnenin ait olduğu sınıfı bulabilmektedir. Redmon vd. [85] 2015 yılında yaptıkları çalışmada gerçek zamanlı birçok farklı nesne sınıfını ve konumunu bulabilen YOLO (Sadece Bir Kez Bak - You Only Look Once) ağ mimarisi önermiştir.

YOLO ağ mimarisi nesnelerin konumlarını gösteren sınırlayıcı kutuları bulmak için giriş görüntüsünü S × S boyutunda ızgara şeklinde bölmüştür. Daha sonra her bir ızgara hücresi için bir nesne tanımı yapmıştır. Son olarak bir birine komşu ve aynı sınıfı gösteren ızgara hücresi birleştirilerek sınırlama kutusu elde etmiştir (Şekil 3.3).



Şekil 3. 3. YOLO [85] ağ mimarisinde nesne konumunun bulunması ve sınıflandırılması

Nesne tespit ve takibi için geliştirilen birçok farklı ağ mimarisi (SSD: Single Shot MultiBox Detector [86] , Faster R-CNN: Region Proposal Networks [69] , R-FCN: Region-based Fully Convolutional Networks [87]) nesne konumunu bulmak için çeşitli farklılıklar olsa bile temel anlamda YOLO ile aynı mantığa sahiptirler.

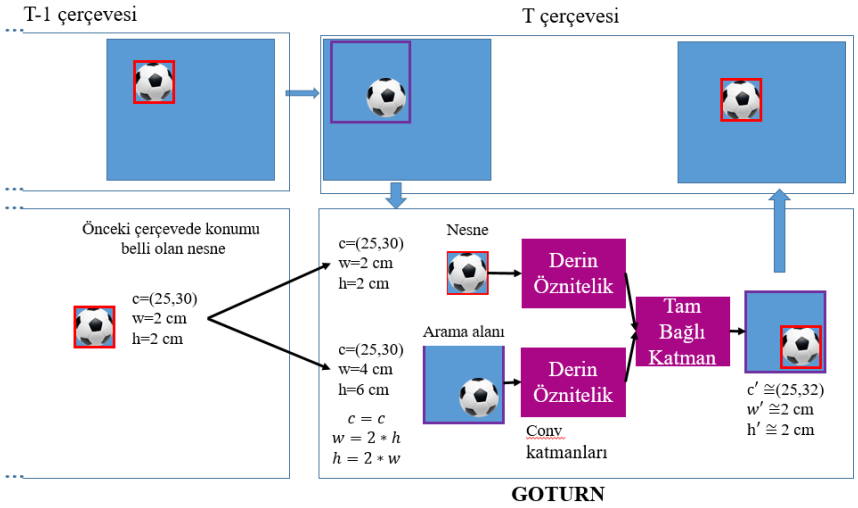
3.2.1. GOTURN Ağ Mimarisinde Giriş ve Çıkış Birimleri

GOTURN ağ mimarisi, nesne takibini yapabilmesi için önceki çerçevedeki nesne konum bilgisini alması gerekmektedir. Bunun için videonun ilk çerçevesinde nesnenin konumu ağ mimarisine manuel olarak verilmiştir. Daha sonraki çerçevelerde, yöntemin önceki çerçevede bulduğu konumlar kullanılmıştır.

Videonun t. çerçevesinde benzerlik yoluyla nesne konumunu bulmak için paralel iki girişe sahip GOTURN ağ mimarisine (Şekil 3.4) iki giriş görüntüsü verilir. Bunlar sırasıyla nesne ve nesnenin aranacağı alan görüntüsüdür. Nesne görüntüsü $t-1$ çerçevesinde manuel biçimde kırpılan görüntüdür. İkinci görüntü ise nesnenin aranacağı arama görüntüsüdür. Arama görüntüsü, t çerçevesinde sırasıyla genişliği $k1w$, yüksekliği $k2h$ ve merkezi c' olan kırpılmış görüntüdür. Buradaki c' merkez koordinatları, önceki çerçevede nesne konumunu veren sınırlayıcı kutu (genişlik w, yükseklik h ve merkezi c olan dikdörtgen) ile aynı merkezlidir ($c' = c$). Aynı şekilde $k1h$ ve $k2h$ değerleri ise nesnenin genişliğinin ve yüksekliğinin k katıdır. Bu değerler nesnenin ne kadarlık bir alanda aranacağını belirtir. GOTURN çalışması için $k1$ ve $k2$ değerleri 2 olarak alınmıştır. Şekil 3.4'te yöntemin genel akış şeması verilmiştir [13].

GOTURN ağ mimarisi regresyon tabanlı geliştirilen ağ mimarilerine benzemektedir. Bu ağ mimarisi temel olarak arama görüntüsü öncelikle ızgara şeklinde parçalanır. Tam bağlı katmanlarda her bir parçanın nesneye ait olup olmama durumu hesaplanır. Elde edilen sonular doğrultusunda nesne olarak belirlenen parçalar birleştirilerek bir dikdörtgen alan elde edilir.

Ağ mimarisinin son katmanı 4 nörondan oluşmaktadır. Bu nöronlardan çıkan değerler, geçerli çerçevede nesne konumunu veren sınırlayıcı dikdörtgenin sol üst ve sağ alt koordinatlarını vermektedir. Bu şekilde nesnenin konumsal bilgileri elde edilmiştir.



Şekil 3.4. GOTURN yönteminin çalışma prensibi

3.2.2. GOTURN Ağ Mimarisi

GOTURN yönteminde görüntü karşılaştırma tabanlı bir ağ mimarisi geliştirilmiştir. Bu mimaride iki farklı görüntüden ilk 5 katmanda elde edilen derin öznitelikler tam bağlı katmanlarda birleştirilerek ortak katmandan çıkışa doğru ilerlemiştir. Tam bağlı katmanların rolü, hedef nesneden gelen derin öznitelikler ile geçerli nesnenin özniteliklerini karşılaştırmak ve hedef nesnenin nereye taşındığını bulmaktır. Tam bağlı katmanlar tarafından öğrenilen fonksiyon, güçlü bir öznitelik karşılaştırma fonksiyonudur. Bu fonksiyon sayesinde izlenen nesnenin takibi yapılırken yöntemin çerçeveler arası ışık değişikliği, kapanma (occlusion) veya deformasyon gibi değişikliklere dayanıklı kalması sağlanmıştır [13].

Ağ mimarisinin öznitelik çıkartma katmanları, CaffeNet mimarisinin ilk beş katmanından alınmıştır [82]. Tam bağlı katmanlar her biri 4096 nöronlu 3 tam bağlı katmandan oluşmaktadır. Son katman, sınırlayıcı kutuyu temsil eden 4 nörondan oluşmaktadır. Doğrulama kümesini kullanarak seçilen 10 faktör ile çıktı değerleri ölçeklendirilmiştir. Ağdaki parametrelerin değerleri CaffeNet için varsayılanlardan alınmış ve tam bağlı katmanlar arasında dropout ve ReLU doğrusal olmayan fonksiyonlar kullanılmıştır. Oluşturulan ağ Caffe derin ağ platformu kullanılarak çalıştırılmıştır [13].

3.2.3. GOTURN Ağ Mimarisinin Eğitimi

Oluşturulan ağ mimarisi, video ve hareketsiz görüntü kombinasyonu kullanılarak eğitilmiştir. Her iki durumda da, yöntemin eğitimi için L1 hata fonksiyonu kullanılmıştır. Burada yöntemin tahmin ettiği sınırlayıcı kutu ile gerçek doğruluk sınırlayıcı kutusu arasında çıkartılan hata değerine göre ağ parametreleri güncellenmiştir. L1 hata fonksiyonu, gerçek değer ile tahmini değer arasındaki farkın mutlak toplamıdır. L1 hata fonksiyonu denklem 3.1'de verilmiştir. Bu denklemde yöntemin bulduğu çerçeve konumu (y^i) ile nesnenin gerçek konumu ($\hat{y}^i$) arasındaki mutlak farkın toplamıdır. Denklemdeki i parametresi çerçeve indeksini göstermiştir.

$$L = \sum_{i=1}^{n} \left| y^i - \hat{y}^i \right| \tag{3.1}$$

Eğitim setinin bir bölümü nesnelerin bulunduğu yerin etiketlendiği ALOV300++ [12] video setinden oluşmuştur. Eğitim kümesindeki ardışık her çerçeve çifti için, nesne ve nesnenin arama bölgesi kırpılmıştır. Eğitim süresi boyunca bu çiftler ağa gönderilmiş ve nesnenin ilk kareden ikinci kareye nasıl taşındığı tahmin edilmeye çalışılmıştır. Daha sonra yapılan tahminlerde oluşan hata değerleri ile ağ mimarisinin parametreleri güncellenmiştir. Eğitim setinin ikinci bölümü ise ImageNet (ImageNet Large Scale Visual Recognition Challenge) [43] görüntü veri setinden alınmıştır. Bu eğitim kümesinin kullanılmasındaki amaç

eğitilen ağın nesnelerin aşırı uyumsuzluğunu öğrenmesi ve oluşturulan ağın daha çeşitli nesneleri takip edebilmesini sağlamaktır. ImageNet görüntü veri seti ortalama 1000'den fazla farklı nesneyi ve 2.3 milyon görüntüyü içermektedir Bu veri seti ile eğitilen bir ağ mimarisi neredeyse takip edebileceği her nesneyi önceden görme imkânına sahiptir. Bu görüntülerde belirli kaydırma ve ölçekleme değişikliği uygulanarak eğitim veri setine yapay hareketlilik kazandırılmıştır.

3.3. SiamFC

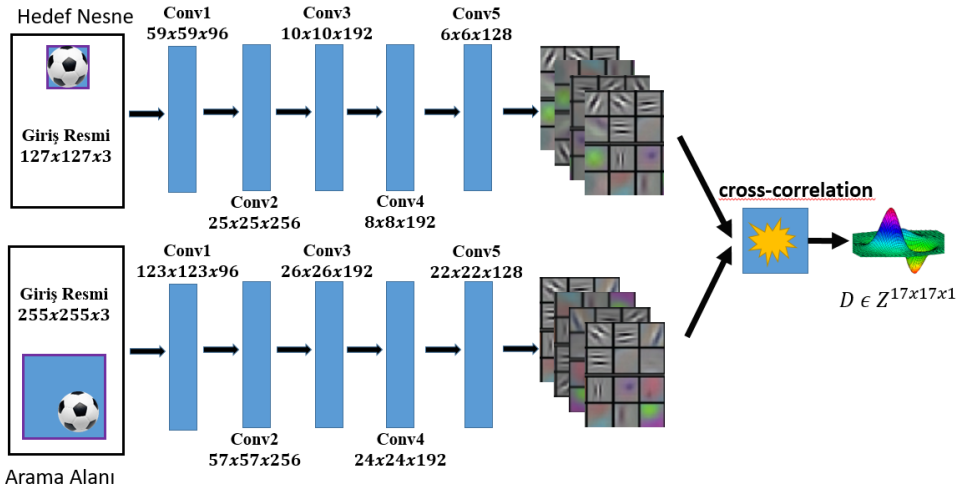
SiamFC (Nesne Takibi için Tam Bağlı Siyam Ağı - Fully-Convolutional Siamese Networks for Object Tracking) yöntemi derin benzerlik (Siamese mimarileri) ağı biçiminde oluşturulan ağ mimarisi ile nesne takibi gerçekleştiren bir yöntemdir. Siamese ağ mimarilerinin temel çalışma prensibi iki aşamadan oluşmaktadır. İlk aşama görüntülerden derin özniteliklerin elde edilmesidir. İkinci aşamada ise bu öznitelikler karşılaştırılarak benzerliklerin hesaplanmasıdır [15]. Benzerlik hesaplamak için nesne görüntüsü ve nesne ile aynı boyuttaki bir aday görüntüsünü karşılaştıran f fonksiyonu öğrenilmektedir [15].

3.3.1. SiamFC Ağ Mimarisi

Geliştirilen mimari nesnenin aranacağı görüntüye (x) tam konvolüsyonel ağ mimarisidir (Şekil 3.5). Tam konvolüsyonel ağın avantajı, nesne ile aynı boyuttaki bir aday görüntü yerine, çok daha büyük bir arama görüntüsü kullanabilmesidir. Bu şekilde arama alanı üzerinden tek bir değerlendirmede alan içindeki tüm alt aday görüntülerin benzerlik değerlerini hesaplayabilmiştir. Benzerlik hesaplama işlemleri için, oluşturulan ağ mimarisinin ilk katmalarında elde edilen öznitelikler denklem 3.2'de matematiksel ifadesi verilen çapraz korelasyon (cross-correlation) katmanı ile birleştirilmiştir. Bu denklemde $b1$ parametresi korelasyon ilişkisi için $b1 \in R$ kümesinde bir sinyali ifade etmektedir [15].

$$f\,(z,x) \;=\; \varphi\,(z) \;*\; \varphi\,(x) \;+\; b1 \tag{3.2}$$

Oluşturulan ağın çıktısı Şekil 3.5'te gösterildiği gibi $D \subset Z^2$ üzerinde tanımlanan bir skor haritasıdır. Denklem 3.2'de $\varphi\,(.)$ fonksiyonunun temel işlevi z nesne ve x arama görüntülerinin öznitelik haritasını çıkartmaktır. D skor haritasının merkezine göre maksimum skorun konumu referans alınarak hedef nesnenin eski konumundan yeni konumuna doğru yer değiştirmesi sağlanmıştır [15]. İki görüntü arasındaki benzerliği bulma açısından oldukça başarılı olan çapraz korelasyon (cross-correlation) yöntemi, öznitelik haritalarının karşılaştırılması için kısa ve basit bir yöntem olmuştur [18].



Şekil 3.5. Tam konvolüsyonel SiamFC ağ mimarisi.

3.3.2. SiamFC Mimarisinin Eğitimi ve Nesne Takibi

SiamFC yöntemi eğitim için ILSVRC [43] veri setinden alınan görüntü çiftlerini kullanmıştır. Bu çiftler, bir videonun belirli aralıktaki çerçevelerinden alınan nesne ve arama görüntüleri çıkarılarak elde edilmiştir. Daha sonra arama görüntüsü üzerinden puan haritası oluşturulmuştur. Puan haritasında nesneden belli bir mesafe içindeki aday görüntülere pozitif (+1), diğer aday görüntülere ise negatif (-1) değerler atanmıştır [15]. SiamFC ağ mimarisi eğitim için pozitif ve negatif örneklere yönelik lojistik regresyondaki sınırlı sayıda sonuç üreten ve lojistik kaybı benimseyen ayırt edici bir yaklaşım kullanmıştır (denklem 3.3).

$$f'(\mathrm{y};\ \mathrm{v}) = \log(1 + \exp(-\mathrm{yv})) \tag{3.3}$$

Bu denklemde v parametresi, arama görüntüsündeki tek bir örnek-aday çifti için elde edilen puanı göstermekte, y ise $\in \{+1, -1\}$ olmak üzere doğruluk değerini göstermektedir. Skor haritası iki görüntüden elde edilen derin özniteliklere çapraz korelasyon uygulanarak iki çerçeve arasındaki benzerliği gösteren ısı haritasıdır. Bu şekilde eğitim sırasında her bir çerçeve çifti için bir skor haritası (v: $D \rightarrow R$) elde edilmiştir. Skor haritasının kaybı, her bir çift için bireysel kayıpların ortalama değeri olarak tanımlanmıştır. Daha sonra bu bireysel kayıplar yardımıyla ağ parametreleri (θ) denklem 3.4'te gösterilen problem şeklinde SGD algoritması uygulanarak eğitilmiştir [15].

$$argmin_{\theta}\ E_{(z,\ x,\ y)}\ L(y;\ f(z;\ x;\ \theta)) \tag{3.4}$$

Eğitim boyunca ağ mimarisindeki öznitelik katmanları ve korelasyon katmanları yukarıdaki gibi eğitilmiştir. Test zamanında ardışık iki çerçeveden alınan nesne ve arama görüntülerinin derin öznitelikleri ağ mimarisinin ilk katmanları ile hesaplanmıştır. Daha sonra

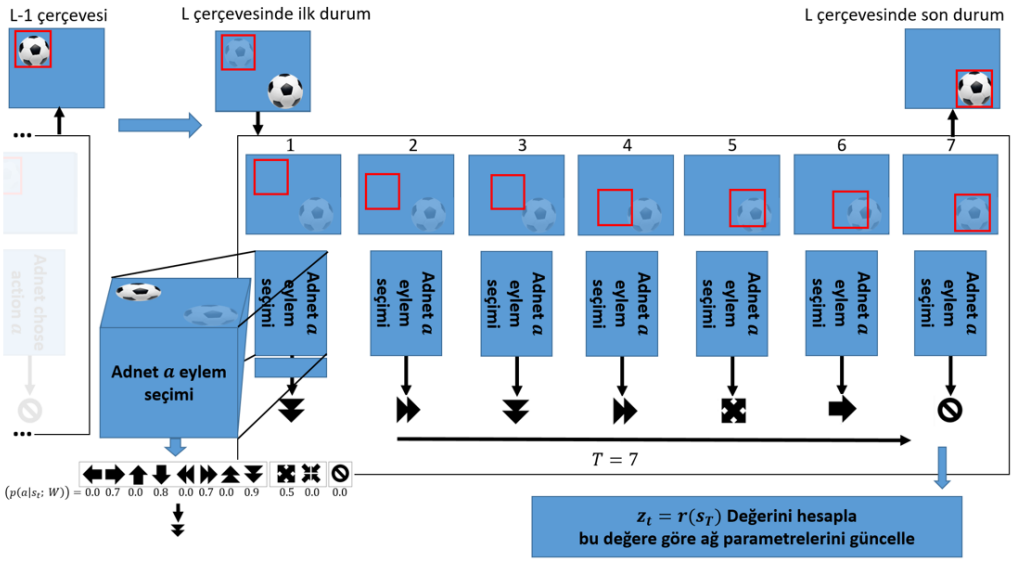
elde edilen bu özniteliklerin benzerliğini hesaplamak için çapraz korelasyon katmanı ile karşılaştırma yapılmıştır. Kıyaslama sonucunda 17×17 boyutunda bir skor haritası elde edilmiştir. Elde edilen skor haritasının boyutu bicubic interpolasyon (yukarı örnekleme) uygulanarak 272×272 boyutuna çıkartılmıştır [15]. Son olarak skor haritasında en yüksek değere karşılık gelen konum nesne merkez konumu olarak kabul edilmiştir.

SiamFC yöntemi, nesneyi gösteren sınırlama kutusunun merkez ve boyutunu bulmak için nesne boyutunu %0.96, %100 ve %103.75 oranında değiştirilerek yukardaki işlemleri tekrarlamıştır. Elde edilen 3 farklı ısı harita en yüksek nokta kime ait ise asıl istenilen sonuç haritası olarak seçilmiştir. Seçilen ısı haritasının en yüksek değere karşılık gelen konum nesne sınırlama kutusunun merkez konumu olarak kabul edilmiştir. Daha sonra nesne boyut değişikliği için seçilen ısı haritası hangi boyut değişim oranından elde edilmiş ise o oranda nesne boyutu değiştirilmiştir.

3.4. ADNet

Yun vd. [71] ardışık eylemler yoluyla nesne takibi yapan ADNet (Eylem Karar Ağları - Action Decision Network) yöntemini geliştirmiştir. Bu yöntemde hedef nesneyi gösteren sınırlama kutusu ikinci çerçevedeki yeni nesne konumuna ulaşmak için nesnenin yapabileceği 11 hareketi öğrenmiştir. Bu hareketler kullanılarak ADNet ağ mimarisi çevrimdışı bir şekilde eğitilmiş ve çevrimiçi belli başlı güncellemeler yapılmıştır. Şekil 3.6'da ADNet yönteminin genel çalışma prensibi görülmektedir [71].

ADNet çalışma prensibi MDP (Markov Karar Süreçleri - Markov Decision Process) yöntemine dayanmaktadır. MDP yöntemi genel anlamıyla belirlenen durumdan başka durumlara geçmek için bir karar mekanizması oluşturmaktadır. Nesne hareketinden sorumlu *faktör (agent)*, yaptığı belli hareketler doğrultusunda durum değişikliği yapar. Bu değişikliklerin doğruluğu hesaplanarak *faktör(agent)* karşılık alır. Buna karşılık ise; aracı doğru hareket yapmışsa ödüllendirilir, aksi durumda cezalandırılır [88].



Şekil 3.6. ADNet nesne takip yönteminin genel akışı

ADNet yönteminde *aracı*, nesnenin yapabileceği 11 farklı hareketi tanımlamıştır. Tanımlanan hareketler yardımıyla çerçeveler arası geçişlerde nesneyi sınırlayan dikdörtgen, nesnenin yeni konumuna doğru taşınmıştır. Çerçeveler boyunca bu işlemler tekrarlanarak nesne takibi yapılmıştır. ADnet yönteminde Eylem, Durum, Durum Geçiş Fonksiyonu ve Karşılık olmak üzere 4 tanım mevcuttur [71].

Aksiyon-Eylem: Nesnenin yapabileceği hareket biçimi olarak tanımlanmıştır. ADNet yönteminde kayma (konum değişikliği) ve boyut hareketi şeklinde toplam 11 hareket biçimi oluşturulmuştur. Bunlar sırasıyla *sol*, *sağ*, *yukarı*, *aşağı* ve bunların iki katı olmak üzere 8 kayma hareketi; *büyült* ve *küçült* şeklinde iki boyut hareketi olarak tanımlanmıştır. Bunlara ek olarak taşıma işleminin bittiğini gösteren *dur* hareketi bulunmaktadır. Her bir hareket (eylem) a ile gösterilmiş ve 11 boyutlu bir vektör ile kodlanmıştır [71].

Durum: $\{p_t, d_t\}$ şeklinde bir tanımlama grubu ile nesneyi temsil eden sınırlama kutusunun (parça olarak tanımlanmıştır) konum ve hareket bilgilerini gösterir. Burada $p_t \in R^{112\times112\times3}$ çerçeve içinden seçilen parça görüntüsüdür. p_t görüntüsünün konumsal değerleri $b_t = [x^{(t)}, y^{(t)}, w^{(t)}, h^{(t)}]$ vektörü ile ifade edilmiştir. b_t vektöründe sırasıyla; t kaçıncı iterasyon olduğunu, $[x, y]$ seçilen görüntünün merkezinin koordinatlarını, $[w, h]$ ise genişlik ve yüksekliğini ifade etmektedir. Tanımlama grubundaki $d_t \in R^{110}$ ise k adet önceki hareketi içeren bir vektörü temsil etmektedir [71].

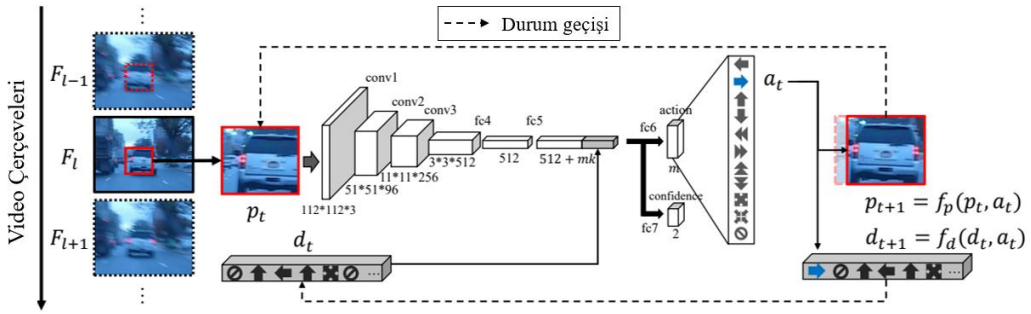
Durum geçiş fonksiyonları: a_t eylem kararı verildikten sonra S_t durumundan S_{t+1} durumuna geçiş fonksiyonları olarak tanımlanmaktadır. Geçiş fonksiyonları, parça görüntüsünün geçiş fonksiyonu $f_p(\cdot)$ ve eylem dinamikleri fonksiyonu $f_d(\cdot)$ olmak üzere iki fonksiyon ile tanımlanmıştır. Parça görüntüsünün geçiş fonksiyonu $b_{t+1} = f_p(b_t, a_t)$ şeklinde tanımlanır. Burada parça görüntüsünün ilgili eyleme göre belli bir miktar konumu değiştirilir.

Eylem dinamiği fonksiyonu ise geçmişteki eylemleri ifade eden $b_{t+1} = f_p(d_t, a_t)$ şeklinde bir fonksiyon ile tanımlanmıştır. $a_t = 'dur'$ eylemi geldiğinde, parça çerçevedeki son konumu alır. Bundan sonra parça nesneyi bulup bulmama durumuna göre karşılığını alır.

Karşılık($r(s_T)$): Durum geçiş fonksiyonunda 'dur' hareketi geldikten sonra çerçevede nesnenin bulunduğu varsayılmaktadır. Bundan sonra ise eğitim setinin doğruluk değerleri yardımıyla nesnenin ne kadar doğru bulunduğuna bakılmaktadır. Doğruluk değeri olarak tanımlanan karşılık değerini hesaplamak için parça (sınırlama kutusu) ve nesnenin gerçek konumsal bilgilerinin kapladığı alanları hesaplayan $IoU(b_t, G) = \frac{kesişim}{birleşim}$ fonksiyonu kullanılır. Doğruluk değeri 0.7 'den büyükse $r(s_T) = +1$ değeri döndürür, değilse $r(s_T) = -1$ değeri döndürür [71].

3.4.1. ADNet Ağ Mimarisi ve Eğitimi

Şekil 3.7'de ADNet ağ mimarisinin genel görünümü verilmiştir. Bu mimaride Fc6 katmanında 11 çıktı birimi bulunmaktadır. Bu çıktılar olasılıksal olarak s_t durumundan s_{t+1} durumuna geçmek için tanımlanan eylemlerin seçilme olasılığını gösterir ($p(a \mid s_t; W)$). Güven katmanı (fc7), hesaplanan parçanın nesneye ait olup olmama olasılığını üretir ($p(nesne \mid s_t; W)$). Burada oluşturulan olasılık değeri, güven puanı olarak tanımlanmış ve bu puan test esnasında çevrimiçi adaptasyon için kullanılmıştır. ADnet yöntemi VOT2015 [25], VOT2014 [89], VOT2013 [90] ve ALOV300 [12] veri kümelerinden alınan ve 360 adet video içeren bir veri kümesi kullanılarak iki aşamada eğitilmiştir. Bu aşamalar sırasıyla denetimli öğrenme (SL, Supervised Learning) ve Takviyeli Öğrenme (RL, Reinforcement Learning) şeklindedir. SL aşamasında, ağ mimarisi denetimli olarak eğitilmiştir. RL aşamasında ise SL aşamasında eğitilen ağın üzerinden doğrulama yapılarak ağ mimarisinde güçlendirmeler yapılmıştır [71].



Şekil 3.7. ADNet ağ mimarisi [71]

Denetimli öğrenme (SL): SL aşamasında eğitim boyunca eylem hareketleri vektörü d_j sıfıra ayarlanarak tüm ağ parametreleri eğitilmiştir. Parça görüntüsünün (sınırlama kutusu)

bulunduğu p_j konumundan nesnenin gerçek konumuna ulaşmak için yaptığı eylem etiketi ($o_j^{(act)}$) denklem 3.5 ile belirlenmiştir.

$$o_j^{(act)} = \underset{a}{\operatorname{argmin}}\, IoU(\bar{f}(p_j, a), G) \tag{3.5}$$

Bu denklemde G nesnenin gerçek konum bilgisini, $\bar{f}(p_j, a)$ fonksiyonu ise nesne sınırlama kutusunun p_j konumundan a eylemi yapıldıktan sonra geleceği yeni konumu üretir. p_j konumuna karşılık gelen sınıf etiketi ($o_j^{(cls)}$), $IoU(p_j, G) = \frac{kesişim}{birleşim}$ denklemi ile belirlenir. Eğer denklem çıkışı değeri (IoU) 0.7 den büyükse 1 (nesne), aksi durumda 0 (arka plan) olarak etiketlemiştir. Eğitim kümesindeki her bir örnek $\{(p_j, o_j^{(act)}, o_j^{(cls)})\}_{j=1}^{m}$ parametre topluluğu ile gösterilmiştir. Burada ADNet ağ parametreleri denklem 3.6'da tanımlanan kayıp fonksiyonu SGD ile minimize edilerek eğitilmiştir [71].

$$L_{SL} = \frac{1}{m}\sum_{j=1}^{m} L(o_j^{(act)}, \hat{o}_j^{(act)}) + \frac{1}{m}\sum_{j=1}^{m} L(o_j^{(cls)}, \hat{o}_j^{(\mathrm{cls})}) \tag{3.6}$$

m parametresi örnek sayısını, L Çapraz entropiyi ve $o_j^{(act)}/o_j^{(cls)}$ ise sırasıyla hareket/sınıf olasılıksal değerlerini belirtmektedir.

Takviyeli Öğrenme (RL) : Bu eğitim aşamasının temel mantığı yarı-denetimli olarak durumlar arasındaki geçişleri yaptıktan sonra olumlu ya da olumsuz sonuçlar neticesinde ağ parametrelerin cezalandırılması veya ödüllendirilmesidir. RL eğitim aşamasında hareket tahmini [91] eğitim yöntemi kullanılarak SL eğitiminden geçmiş ağ parametreleri tekrar bir eğitimden geçirilmiştir. RL'nin amacı hareket tahminini [91] öğrenmek olduğundan, izleme aşamasında gerekli olan güven katmanı ($fc7$) yok sayılmıştır [71].

Eğitim süresince iki çerçeve arası nesne konumunu bulmak için ağ parametreleri (W_{RL}) ve durum ($s_{t,l}$) değerleri kullanılarak bir eylem $a_{t,l}$ seçilmektedir. Seçilme işlemi $fc6$ katmanından gelen derin öznitelikler değerlendirilerek yapılmaktadır. Bu değerlendirme sonucu tanımlanan her bir eylemin ($p(a|s_{t,l}; W_{RL})$) olasılığı hesaplanarak en yüksek olasılık veren eylem seçilir. Daha sonra seçilen eyleme göre parça hareket ettirilir. Eğer eylem seçiminde 'dur' hareketi gelirse, hareket işlemi tamamlanmış olur ve nesne konumunun gerçek değerine bakılarak karşılık değeri hesaplanır. Karşılık değeri $1 \dots t$ iterasyonlar sonunda nesnenin ikinci çerçevedeki konumunu bulma durumuna göre +1 veya -1 değerini almaktadır. Daha sonra $z_{t,l}$ karşılık değerleri kullanarak W_{RL} ağ parametreleri denklem 3.7 ile güncellenir [91].

$$\Delta W_{RL} \propto \sum_{l}^{L}\sum_{t}^{T_l} \frac{\partial \log p(a|s_{t,l}; W_{RL})}{\partial W_{RL}} z_{t,l} \tag{3.7}$$

Takviyeli öğrenme aşamasında sistem kısmen doğruluk değerleri verilerek eğitilmektedir. Örneğin bir videoda 162-189 aralığındaki çerçevelerde doğruluk etiketinin

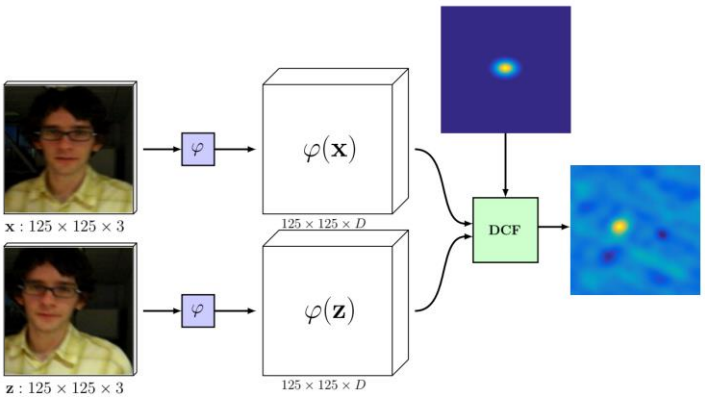
verilmediği durumlarda $\{z_{t,l}\}$ karşılık değerini hesaplamak için beklenmektedir. Etiketli bir çerçeve geldiği anda (örneğin 190'nıncı çerçeve) $\{z_{t,l}\}$ değerini hesaplar. Çıkan değere bakılarak geçmişteki çerçevelere de aynı karşılık değeri verilmektedir [71].

3.4.2. Çevrimiçi Güncelleme

ADNet yönteminin ağ parametreleri görünüm değişikliklerine veya deformasyonlarına karşı daha güçlü kalabilmeleri için çevrimiçi bir şekilde güncellenmiştir. Güncelleme için $fc7$ katmanında nesne bulma olasılığı ($p\,(nesne \mid s_t;\, W)$) değerine bakılmıştır. Eğer bu değer 0.5 değerinden büyükse "nesne var" olarak kabul edilir ve SL aşamasındaki gibi ağ parametreleri güncellenir. Fakat 0.5 altındaysa "nesne kaybolmuş" olarak kabul edilir ve tekrar nesne tespit işlemi gerçekleştirilir. Nesne tespit işlemi için rassal Gauss gürültüsü dağılımı yardımıyla nesnenin bulunabileceği konumların kümeleri oluşturulur. Bu kümeden güven değeri en yüksek olan parça nesnenin yeni konumu olarak seçilir.

3.5. DCFNet

Wang vd. [45] nesne takibi için derin öznitelikleri kullanan ayrımcı korelasyon filtresi tabanlı DCFNet (Görsel İzleme için Diskriminant Korelasyon Filtre Ağı - Discriminant Correlation Filters Network For Visual Tracking) yöntemini önermiştir. Bu çalışmada diğer ayrımcı korelasyon filtre tabanlı yöntemlerden farklı olarak derin öznitelikler kullanmıştır. DCFNet mimarisi iki görüntü arası benzerliği bulan Siamese ağ mimarisinin sonuna bir ayrımcı korelasyon filtresi katmanı eklenerek oluşturulmuştur (Şekil 3.8). Geliştirilen ağ mimarisi çevrimdışı eğitilmenin yanında, aynı zamanda çevrimiçi de eğitilerek ağ parametreleri güncellenmiştir [45].



Şekil 3.8. DCFNet yönteminin genel görünümü [7]

DCFNet yöntemi iki aşamadan oluşmaktadır. Bunlar sırasıyla derin özniteliklerin çıkartılması ve bu özniteliklerden korelasyon yardımıyla nesne konumunun bulunmasıdır. Ağ

mimarisine öncelikle belli boyutta kırpılmış iki görüntü girdi olarak verilmektedir. Bunlar sırasıyla $t-1$ çerçevesinden seçilen nesne görüntüsü ile t çerçevesinden nesnenin aranacağı arama görüntüsüdür. İki görüntü ilk katmandan geçirilerek derin öznitelikler elde edilmiştir [45]. Bu derin özniteliklere daha sonra korelasyon filtresi uygulanarak nesne konumunu veren bir ısı haritası elde edilmiştir. Öznitelikleri çıkartmak için VGG-Net [32] ağ mimarisinin ilk katmanı olan conv1 katmanı kullanılmıştır.

Ele alınan DCFNet yönteminde korelasyon filtresi için kullanılan hata fonksiyonu denklem 3.8'de verilmiştir.[45]. Nesne takibinde istenilen filtre (w) bu hata fonksiyonu kullanılarak eğitilmiştir.

$$\epsilon = \left\| \sum_{l=1}^{D} w^l * \varphi^l(x) - y \right\|^2 + \lambda \sum_{l=1}^{D} \|w^l\|^2 \tag{3.8}$$

Bu denklemde w^l parametresi w korelosyon filtresinin l'ninci kanalını, $*$ parametresi dairesel korelasyonu ve $\lambda \geq 0$ ise sabit düzenleştirme katsayısını ifade etmektedir. $\varphi(x) \in R^{M \times N \times D}$ fonksiyonu ise video çerçevesinden çıkarılan derin öznitelikleri göstermektedir. Buradaki M, N, D değerleri sırasıyla öznitelik matrisinin yükseklik, genişlik ve kanal sayısını ifade etmektedir.

DCFNet yönteminin kullandığı filtrenin her bir kanaldaki (w^l) değerlerinin eşitliği denklem 3.9'da gösterilmiştir. Bu denklem yardımıyla filtre eğitilmiş ve güncellenmiştir [45].

$$\hat{w}^l = \frac{\hat{\varphi}^l(x) \odot \hat{y}^*}{\sum_{k=1}^{D} \hat{\varphi}^l(x) \odot (\hat{\varphi}^l(x))^* + \lambda} \tag{3.9}$$

Burada $\hat{y}$ ayrık Fourier dönüşümünü, y^* ise kompleks sayı olan y'nin karmaşık eşleniğini ve son olarak $\odot$ ise Hadamard eleman bazlı çarpımı temsil etmiştir. Eğitim boyunca denklem 3.9 kullanılarak nesne görüntüsüne odaklanmış olan korelasyon filtresi eğitilir.

Test aşamasında nesne konumu tespiti için öncelikle aranacak bölgenin derin öznitelikleri ($\varphi(z)$) çıkarılmıştır. Arama bölgesi 125×125 boyutunda olup nesnenin eski konumunu merkezleyen ve mevcut çerçeveden kırpılmış olan görüntü parçasıdır. Daha sonra çıkarılan özniteliklere korelasyon filtresi uygulanarak ısı haritası hesaplanmıştır (denklem 3.10). Isı haritasında maksimum değeri veren konumlar gerçek nesne konumu olarak seçilmiştir.

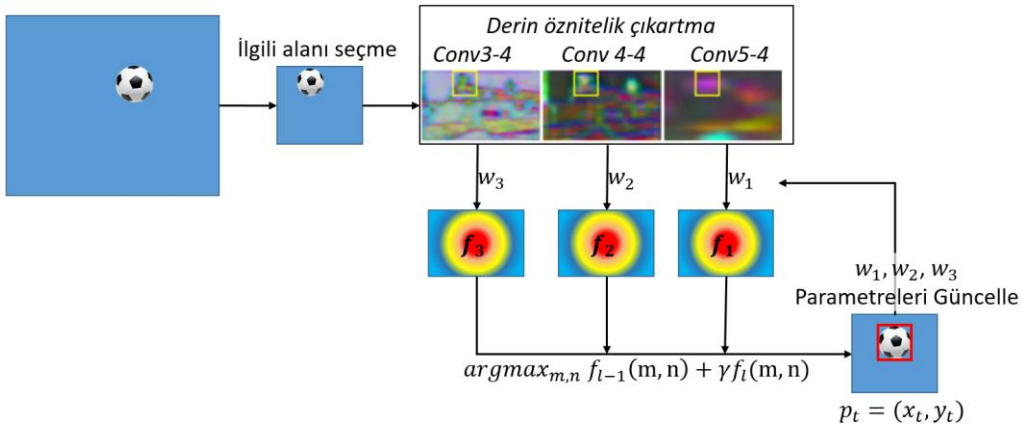
$$g = F^{-1}\left(\sum_{l=1}^{D} \hat{w}^{l*} \odot \hat{\varphi}^l(z)^* \right) \tag{3.10}$$

DCFnet yöntemi test zamanında ağ mimarisinde geri yayılım yaparak filtre ve derin öznitelik çıkartma parametrelerini güncellemiştir. Geri yayılımda parametre optimizasyonu için ağırlık azaltma (weight decay) yöntemi kullanılmıştır [45]. Ayrıca öznitelik haritasındaki değerlerin büyüklüğünü kısıtlamak ve eğitim sürecindeki kararlılığı arttırmak için konvolüsyon katmanının sonuna bir normalizasyon katmanı eklenmiştir [45].

Eğitim seti NUS-PRO [92], TempleColor128 [24] ve UAV123 [93]'den alınmış videolardan oluşmaktadır [45].

3.6. HCFT

Genel olarak derin öğrenme yöntemleri, oluşturulan ağ mimarisinin son katmandaki öznitelikleri kullanmaktadır. Fakat ağ mimarisinin son katmanlarına doğru ilerledikçe daha güçlü ayırt edici bilgiler kazanılmasına rağmen nesnelere ait mekânsal bilgiler kaybedilebilmektedir. Ma vd. [33] nesne takibi için mekânsal bilgileri barındıran ilk katmanlardaki özniteliklerin de önemli olabileceğini belirtmiştir. Bu çalışmaya göre ağ mimarisine giren bir görüntü ağ katmanlarında ilerledikçe özniteliklerdeki anlamsal ayırt edicilik artmaktadır. Bunun yanında mekânsal çözünürlük ise kademeli olarak azalmaktadır. Bu fikir doğrultusunda nesne takibi için ağ mimarisinin her bir katmanındaki özniteliklerin birlikte kullanılmasının başarı oranını arttıracağı belirtilmiştir. Geliştirilen HCFT (Görsel İzleme için Hiyerarşik Konvolüsyon Özellikleri - Hierarchical Convolutional Features for Visual Tracking) yöntemi nesne takibi için yukardaki fikir doğrultusunda hem anlamsal hem de konumsal bilgileri kullanarak nesne takibi yapmıştır. Bunun için her konvolüsyon katmanın sonuna bir uyarlanabilir korelasyon filtresi eklenmiştir. Bu filtrelerden elde edilen korelasyon tepki haritalarının sonuçları geliştirilen bir yöntem (denklem 3.14) ile birleştirilerek nesnenin konumu elde edilmiştir. Şekil 3.9'da yöntemin ana adımları gösterilmiştir.



Şekil 3.9. HCFT yönteminin akış diyagramı

3.6.1. HCFT Çalışma Metodolojisi

Öznitelik olarak eğitilmiş VGG-Net [32] ağ mimarisinin conv3-4, conv4-4 ve conv5-4 konvolüsyon katmanlarından elde edilen öznitelikler yeniden boyutlandırılarak kullanılmıştır. Yeniden boyutlandırma için bilinear enterpolasyon uygulanarak öznitelikler sabit ve eşit boyuta getirilmiştir. HCFT yönteminde conv3-4, conv4-4 ve conv5-4 katmanlarından

çıkarılan özniteliklerin (x) her birisi için farklı bir korelasyon filtresi öğrenilmiştir. Ağ mimarisinin i'ninci katmanında elde edilen öznitelikler $X \in Z^{\mathrm{M} \times \mathrm{N} \times \mathrm{D}}$ şeklinde gösterilmiştir. Burada M, N ve D parametreleri sırasıyla katmanın genişliğini, yüksekliğini ve derinliğini belirtmektedir. Her bir katman için korelasyon filtresinin (w) eğitiminde denklem 3.11.a'daki hata fonksiyonu kullanılmıştır [33].

$$w^* = \underset{w}{\operatorname{argmin}} \sum_{k} \left\| w \cdot x_{m,n} - y(m,n) \right\|^2 + \lambda \left\| w \right\|_2^2 \tag{3.11.a}$$

$$w \cdot x_{m,n} = \sum\nolimits_{d=1}^{D} w_{m,n,d}^{T} x_{m,n,d} \tag{3.11.b}$$

Bu denklemde λ düzenleme parametresi ve $"\cdot"$ çarpımı Hilbert uzayında doğrusal bir çekirdek çarpımını göstermektedir (denklem 3.11.b). d'ninci kanalda frekans alanında eğitilen korelasyon filtresinin eşitliği denklem 3.12'deki gibi gösterilmiştir [33].

$$W^d = \frac{Y \odot \overline{X}^{\,d}}{\sum_{i=1}^{D} X^{\,i} \odot \overline{X}^{\,i} + \lambda} \tag{3.12}$$

Y değeri y'nin Fourier dönüşümü formu ve $\odot$ operatörü ise Hadamard eleman bazlı çarpımıdır. Oluşturulan ağın l'ninci katmanındaki derin öznitelikler kullanılarak buradaki korelasyon sonuç haritası denklem 3.13'teki gibi hesaplanmıştır. Burada l'inci katmandaki derin öznitelikler Z ile gösterilmiştir.

$$f_l = F^{-1}\left(\sum_{l=1}^{D} W^d \odot \overline{Z}^{\,d} \right) \tag{3.13}$$

F^{-1} operatörü, ters FFT dönüşümünü gösterir ve l'inci katman üzerindeki hedef konum, $M \times N$ boyutundaki korelasyon haritası üzerinden f_l'inin maksimum değerini veren konum olarak tahmin edilmiştir [33].

Normal şartlarda her korelasyon çıktısı nesne konumunu tahmin edebilir. Fakat daha doğru tahmin için hiyerarşik olarak yanıt haritaları birleştirilmiştir. Bu şekilde nesne konumunun daha güçlü bir tahmin ile elde edilmesi sağlanmıştır. Sırasıyla l'inci katmandaki f_l yanıt haritasının en büyük değerleri $(\hat{\mathrm{m}}, \hat{\mathrm{n}}) = \operatorname{argmax}_{m,n} f_l(\mathrm{m}, \mathrm{n})$ fonksiyonu ile hesaplanmıştır [33]. Nihai son konumu bulmak için önceki katmanların bulduğu en büyük değerler denklem 3.14 kullanarak birleştirilir.

$$\underset{m,n}{\operatorname{argmax}}\, f_{l-1}(m,n) + \gamma f_l(m,n) \tag{3.14}$$

Bu denklemde $l-1$'inci korelasyon yanıt haritasında $(\hat{\mathrm{m}}, \hat{\mathrm{n}})$ değerleri yalnızca r × r komşu bölgelerinde aranır. l katmanından gelen değerler bir γ düzenleyici terimi ile ağırlıklandırılarak $l-1$'inci korelasyon çıktısına eklenir. Son katmanda nesne konumu denklem 3.14'yı maksimize edecek değerlerin konumu şekilde tahmin edilir. Nesne takibi

boyunca korelasyon filtreleri sonuç değerlerine bakılarak elde edilen hatayı en aza indirgeyerek güncellenmiştir [33].

3.7. BACF

Çevrimiçi nesne takip probleminde ilk çerçevede verilen tek nesne görüntüsü ile korelasyon filtresini eğitmek oldukça zordur. Bunun yanında bu çerçevede büyük miktarda arka plan görüntüsü mevcuttur. Kiani vd. yaptıkları [66] çalışmada bu yüksek boyutlu arka plan bilgisini kullanmanın büyük avantaj olacağını öne sürerek bu prensibe dayanan BACF (Arka Plana Yönelik Korelasyon Filtre Öğrenme - Background-Aware Correlation Filters) yöntemini önermiştir. Bu bakış açısı son zamanlarda oldukça güçlü performanslar gösteren derin öğrenme tabanlı yöntemlerle benzer başarıda performans elde etmiştir. Bu yöntemde manuel olarak belirlenen 31 kanallı YGH öznitelikleri kullanmıştır.

BACF yönteminin temel çalışma prensibi geleneksel korelasyon filtre yöntemleri ile aynı adımlardan olmuştur. Öncelikle ilk çerçevede yoğun arka plan ve nesne öznitelikleri ile korelasyon filtresi eğitilmiştir. Daha sonraki çerçevelerde bu korelasyon filtresini kullanılarak nesne merkez konumu tespit edilmiştir. Nesne boyut tahmini için ölçeklendirilmiş korelasyon çıktılarından en yüksek değeri veren korelasyon çıktısına göre nesne boyutu ölçeklendirilmiştir.

3.7.1. Korelasyon Filtresi

BACF yönteminde kullanılan çok kanallı korelasyon filtreleri denklem 3.15'deki hata fonksiyonu kullanılarak eğitilmiştir.

$$E(h) = \frac{1}{2}\left\| y - \sum_{k=1}^{K} h_k * x_k \right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 \tag{3.15}$$

Bu denklemde $x_k \in R^D$ görüntü özniteliği ve $h_k \in R^D$ korelasyon filtresinin k'inci kanalını göstermektedir. k parametresi öznitelik kanallarının sayısını ifade eder. $y \in R^D$ istenen korelasyon (ısı haritası) tepkisidir. λ düzenleme katsayısını ve $*$ mekansal korelasyon operatörü şeklinde tanımlanmıştır. Denklem 3.15'deki problemin Ridge regresyon [18], [66] yöntemi ile çözülebilmesi için denklem 3.16'daki gibi ifade edilebilir:

$$E(h) = \frac{1}{2}\sum_{j=1}^{D}\left\| y(j) - \sum_{k=1}^{K} h_k^T x_k[\Delta\tau_j] \right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 \tag{3.16}$$

Buradaki $[\Delta\tau_j]$ dairesel kaydırma operatörünü ifade etmektedir. $[\Delta\tau_j]$ tüm $j = [0, \ldots, T-1]$ çerçevelerinin içindeki parça görüntülerinin dairesel kayma değerini üretir. BACF yönteminde denklem 3.16'daki $x_k[\Delta\tau_j]$ yerine $Px_k[\Delta\tau_j]$ yazılarak çok-kanallı arka plan

ayırıcı olan korelasyon filtreleri öğrenilmiştir. Denklem 3.16'nın güncel hali denklem 3.17'de gösterilmiştir:

$$E(h) = \frac{1}{2}\sum_{j=1}^{T}\left\| y(j) - \sum_{k=1}^{K} h_k^T \boldsymbol{P} x_k[\Delta\tau_j] \right\|_2^2 + \frac{\lambda}{2}\sum_{k=1}^{K}\|h_k\|_2^2 \qquad (3.17)$$

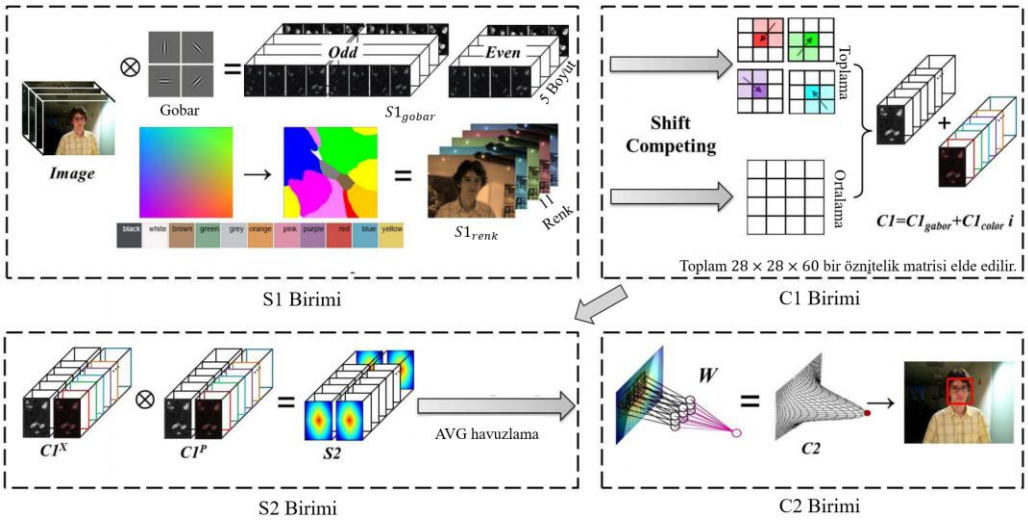
Bu denklemdeki P değeri çerçevenin kaydırılmış her görüntüsünün orta noktasını seçen bir ikili matristir. YGH özniteliği için T kaydırma adetini, $x_k \in R^T$ geniş mekansal destekli bir eğitim örneğini, k öznitelik kanallarının sayısını ve $y \in R^T$ istenilen korelasyon çıktısını ifade etmektedir. Son olarak $h \in R^D$ ise korelasyon filtresini göstermektedir.

Korelasyon filtresinin eğitiminde işlem maliyetini düşürerek gerçek zamanlı nesne takibini sağlamak için ALM (Artırılmış Lagrangian Metodu - Augmented Lagrangian Method) [94] kullanılmıştır. ALM yöntemi genel anlamda kısıtlı problemi kısıtlanmamış alt problemlere ayırarak bir çözüm aramaktadır. Bu yöntem ile gerçek zamanlı nesne takibi için denklemin çözümünde algoritma karmaşıklığı büyük O notasyonu cinsinden $O(DK^3)$'den $O(DK)$'ya kadar düşmüştür.

Nesne takibinde ortam değişikliklerine uyum sağlamak için oluşturulan filtre çevrimiçi olarak güncellenmiştir. Nesnenin yeni konumu tespit edildikten sonra nesne konumu ve civarındaki bilgiler yardımıyla eğitim prosedürüne benzer işlemler uygulanarak korelasyon filtresi güncellenmiştir.

3.8. BIT

Cai vd. [75] insan görme sisteminin nesne takip performansını göz önüne alarak BIT (Biologically Inspired Tracker-Biyolojik Esinlenmiş Takipçi) yöntemini önermiştir. BIT yöntemi görünüm ve izleme modeli olmak üzere 2 ana birime ayrılmaktadır. Görünüm modeli hedef nesne görünümü için düşük seviyeli biyolojik öznitelikleri hesaplar. İzleme modeli ise hesaplanan öznitelikleri gelişmiş bir öğrenme mekanizmasında değerlendirerek nesne konumunu tespit etmektedir. Görünüm modeli klasik basit hücrelere (S1 birimi) ve kortikal kompleks hücrelere (C1 birimi) bölünmüştür. Aynı şekilde izleme modeli de görüşe dayalı öğrenme (S2 birimleri) ve göreve bağlı öğrenme (C2 birimleri) olarak ayrılmıştır. BIT yönteminin çalışma adımları Şekil 3.10'da gösterilmiştir.



Şekil 3.10. BIT yönteminin akış şeması [75]

Nesne takibinde öncelikle nesnenin aranacağı alan görüntüsü seçilmiştir. t çerçevesindeki arama alanın seçilmesi için $t-1$ çerçevesindeki nesne konumu ile aynı merkezli bir dikdörtgen alınır. Bu dikdörtgenin genişlik ve yüksekliği değerleri $t-1$ çerçevesindeki nesne sınırlama kutusunun k katı olarak alınmıştır. Daha sonra sırasıyla S1 ve C1 katmanlarında t çerçevesindeki arama alanın öznitelikleri çıkartılmıştır. Ardından çıkartılan öznitelikler S2 katmanında nesne özniteliklerileriyle oluşturulan bir prototip ile doğrusal korelasyon işlemine tabi tutulmuştur. Son adımda ise elde edilen korelasyon ilişkisi C2 katmanında tek katmanlı tam bağlı konvolüsyonel yapay sinir ağı katmanından geçirilerek nesnenin konumuna odaklı bir ısı haritası elde edilmiştir.

3.8.1. S1 Birimi - Klasik Basit Hücreler:

Birincil görsel korteks olarak adlandırılan bu bölümde çift ve tek Gabor filtreleri görüntü üzerinde gezdirilerek görüntülerin öznitelikleri elde edilmiştir. Bu özniteliklerin yanında renk adları özniteliği de kullanılmıştır. Görüntüdeki her piksel için renk adları olan siyah, kahverengi, yeşil, pembe, kırmızı, sarı, mavi, gri, turuncu, mor ve beyaz olmak üzere 11 boyutlu olasılıksal renk özniteliği oluşturulmuştur. Bu birimde girişte alınan arama görüntüsünün $S1_{gabor}$ ve $S1_{renk}$ öznitelikleri ile birlikte toplam 60 özniteliği çıkarılmıştır [75].

3.8.2. C1 Birimleri - Kortikal Kompleks Hücreler

Kortikal kompleks hücreler $S1$ biriminden gelen özniteliklerin dönme gibi değişimlere karşı güçlendirilmesi için doğrusal öznitelik entegrasyonu işleminden geçirilmiştir. Bunun için $S1$ biriminde elde edilen $S1_{gabor}$ özniteliklerine doğrusal olmayan standart sapma yöntemi

uygulanmıştır (denklem 3.18.a). $S1_{\text{renk}}$ özniteliklerine ise gürültüye karşı dayanıklı kalmaları için ortalama pooling yöntemi (AVG) uygulanmıştır (denklem 3.18.b). Ortalama pooling işlemi için 4×4 boyutundaki bir filtre, $S1_{\text{renk}}$ öznitelikleri üzerinden gezdirilerek $C1_{\text{renk}}$ öznitelikleri elde edilmiştir [75].

$$C1_{gabor}(x,y) = \sum_{(x,y)} \frac{S1_{gabor}(x,y)}{N_{\delta_x,\delta_y}(x,y)}$$
$$N_{\delta_x,\delta_y}(x,y) = (S1^2_{gabor}(x,y) + S1^2_{gabor}(x+\delta_x, y+\delta_y) + S1^2_{gabor}(x+\delta_x, y) + S1^2_{gabor}(x, y+\delta_y))^{0.5} \quad (3.18.a)$$

$$C1_{\text{renk}}(x,y) = \frac{1}{n_s \times n_s} \sum S1_{\text{renk}}(x,y) \quad (3.18.b)$$

Denklem 3.18.a'da ki N_{δ_x,δ_y} parametresi normalizasyon faktörü ve δ_x, δ_y parametreleri ise kayma eğilimidir. Denklem 3.18.b'de ki $n_s \times n_s$ ise 4×4 boyutundaki ortalama filtredir.

3.8.3. S2 Birimleri - Görünüm Ayarlı Öğrenme

Bu birimde yeni gelen çerçeveden alınan X görüntü öznitelikleri ile depolanmış prototip (P) öznitelikleri arasında Öklid uzaklığına dayalı bir tepki değeri hesaplanmıştır. Bu tepki haritası için denklem 3.19'da verilen doğrusal korelasyon [75] işlemi uygulanmıştır.

$$S2(x,y) = \frac{1}{K} \sum_{k=1}^{K} C1^X(x,y,k) \otimes C1^P(x,y,k) \quad (3.19)$$

X görüntüsünün C1 biriminden elde edilen öznitelikleri $C1^X(x,y,k)$ ve aynı şekilde depolanan prototip P'nin öznitelikleri de $C1^P(x,y,k)$ şeklinde ifade edilmiştir. Burada k parametresi 12 yön ve 5 ölçeğe karşılık gelen 60 öznitelik haritasının indeksidir. $S2$ biriminden elde edilen ilişki denklem 3.18.b'de ki ortalama pooling işleminden geçirildikten sonra $C2$ birimine gönderilmiştir [75].

3.8.4. C2 Birimleri - Görev Bağımlı Öğrenme

Elde edilen $S2$ sonuç matrisi tek katmanlı tam bağlı konvolüsyonel sinir ağı mimarisinden geçirilerek sonuç ısı haritası elde edilir. 28×28 boyutundaki ısı haritasında en büyük değeri veren konum, arama görüntüsünde nesnenin merkezini vermektedir. Nesne merkezi belirlendikten sonra önceki çerçevedeki nesne boyutuyla aynı olacak şekilde bir sınırlama kutusu çizilmektedir. Bu şekilde t çerçevesindeki nesne konumu belirlenir.

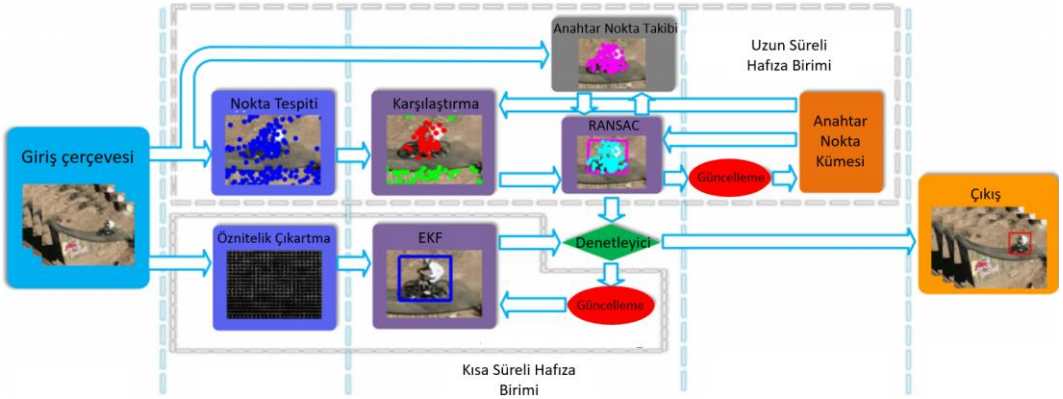
Biyolojik ilham modelinde $S2$ ve $C2$ birimlerinin eğitimi için ilk karedeki nesne görüntüsü referans alınır. Bundan sonra gelen her çerçevede nesne konumu bulunduktan sonra

$S2$ ve $C2$ birimlerindeki ağ parametreleri güncellenmiştir. BIT yönteminde işlem maliyetini düşürmek için $S1$ biriminde hızlı Gabor yaklaşımı, $S2/C2$ birimlerinde ise hızlı Fourier dönüşümü yöntemleri kullanılmıştır [75].

3.9. MUSTer

Hong vd. [19] nesne takibi için önerdikleri MUSTer (Çoklu Hafıza Takipçisi - Multi Store Tracker) yöntemi kısa vadeli hafıza, uzun vadeli hafıza ve ilgili işlem birimlerinden oluşan *Atkinson-Shiffrin Bellek Modeline* dayanmaktadır (Şekil 3.11). Yöntemin temel amacı kısa vadeli hafıza biriminde yapılan nesne takibinin başarısız olduğu durumlarda, uzun vadeli hafıza birimini kullanarak sistemin güncelleştirilmesi ya da yeniden başlatılmasını sağlamaktır. Nesne takibinin asıl yapıldığı birim olan kısa vadeli hafıza biriminde EKF (Entegre Korelasyon Filtresi - Integrated Correlation Filter) kullanılarak güçlü bir şekilde nesne takibi yapılmaktadır. Bunun yanında nesne takibinde kayma veya nesne görünümünde değişiklik olduğu durumlarda, nesne özelliklerinin bir araya getirildiği anahtar nokta kümesi ile EKF takipçisini destekleyen uzun süreli hafıza birimi bulunmaktadır. Bu birim anahtar nokta eşleştirme ve RANSAC tahmin bileşenlerinden oluşmuştur. Anahtar nokta kümesi oluşturmak için *ileri-geri* [95] izleme tekniği kullanılmıştır.

Nesne takibi sırasında, kısa ve uzun vadeli hafızalarda oluşan sonuçlar bir denetleyiciye gönderilir ve bu denetleyici sonuçları işleyerek nesneyi tespit eder. Denetleyici modülü kısa vadeli hafızadaki birimin güncellenmesi gerekiyor ise EKF güncellemesini aktifleştirir. Eğer birimlerdeki sonuçlar aşırı tutarsızlık gösteriyor ise, denetleyici kısa vadeli hafızada bulunan EKF parametrelerini sıfırlar. Bu şekilde EKF takipçisinin uzun vadede başarılı sonuçlar vermesi sağlanmıştır. Bu çalışmada 31 boyutlu YGH özniteliği kullanılmıştır. Ayrıca renkli görüntüler için daha iyi performans sağlamak amacıyla 10 boyutlu renk öznitelikleri [96] çıkarılmış ve bunlar YGH tanımlayıcılarıyla birleştirilmiştir.



Şekil 3.11. MUSTer yönteminin genel akış şeması [19]

3.9.1. EKF ve Anahtar Noktaların Kısa Süreli İşlenmesi

Entegre korelasyon filtreleri nesne takibi için KCF (Çekirdekleştirilmiş İlinti Süzgeçleri - Kernelized Correlation Filters) [60] ve DSSCF (Ayırt edici Ölçek Alan Korelasyon Filtresi - Discriminative Scale Space Correlation Filter) [97] yöntemlerini bir arada kullanmıştır. EKF sistemi Danelljan vd. [97] tarafından tanımlanmış ve boru hattına benzer bir yapıda oluşturulmuştur. EKF yöntemi nesne takibinde konum değişiklikleri için KCF ve boyut tahmini için DSSCF yöntemlerini kullanan iki aşamalı bir filtreleme sürecinden oluşmuştur. EKF biriminde kullanılan korelasyon filtrelerinin eğitimi BACF yöntemine benzer aşamalar içermektedir [19].

Uzun vadeli hafıza biriminde hedef nesne görünümü ile ilgili oluşturulan öznitelik veri tabanı $\mathcal{M}$ ile gösterilmiştir. Bu veri tabanı hedefin öznitelikleri ($\mathcal{T}$) ile arka plan özniteliklerinin ($\mathcal{B}$) birleşim kümesi ile oluşmaktadır (denklem 3.20).

$$\mathcal{T} = \{(\mathrm{d}_i\,,\, \mathrm{p}_i^o)\}_{i=1}^{N_T}, \mathcal{B} = \{\mathrm{d}_i\}_{i=1}^{N_B}\, \mathcal{M} = \mathcal{T} \cup B \tag{3.20}$$

Bu denklemde $\mathrm{d}_i \in \mathcal{R}^{128}$ anahtar noktalarının 128-boyutlu SIFT (Scale-invariant Feature Transform) tanımlayıcılarıdır [98]. N_T ve N_B değerleri tanımlayıcı sayısını göstermektedir. Her bir hedef tanımlayıcı $\mathrm{d}_i \in \mathcal{T}$ koordinatları $\mathrm{p}_i^o \in \mathcal{R}^2$ ile de ilişkilendirilmiştir. p_i^o bilgisi orijinal hedef şablondaki anahtar nokta konumunu hatırlamak için kullanılmıştır.

EKF birimin bir diğer işlevi, $\mathcal{M}$ kümesindeki anahtar noktaları kısa süreli belleğe alarak ardışık olarak işlemektir. Her çerçevede SIFT dedektörü bir dizi anahtar noktalar kümesi çıkarmak için bir görüntü arama alanına uygulanmıştır. SIFT tanımlayıcılarıyla ilişkili tespit edilen anahtar nokta kümesi $\mathrm{P}_D = \{(\mathrm{d}_k, \mathrm{p}_k)\}_{k=1}^{N_D}$ olarak gösterilmiş ve $p_k \in \mathcal{R}^2$ bir anahtar noktasının koordinatını göstermiştir [19].

Eşleşen Anahtar Noktalar: Öklid uzaklığına dayalı olarak, her bir $\mathrm{d}_k \in \mathrm{P}_D$ 'nin en yakın komşu noktalarını bulmak için $\mathcal{M}$ bellek veri tabanında arama yapılmıştır. Kosinüs benzerliği $C(d_k, d_k^{1N})$ kullanılarak d_k ve onun en yakın komşusu d_k^{1N} arasındaki eşleşme güven değeri tanımlanmıştır. Eşleşen güven değeri önceden tanımlanmış eşik değerinden büyükse, aday nokta, *eşleşen anahtar nokta* olarak tespit edilmiştir. Hedef noktalar eşik değeri θ_T ve arka plan noktalarının eşik değeri θ_B olarak gösterilmiştir. Hedef noktası eşleştirmesi sırasında parametrik olmayan en yakın komşu sınıflandırıcı kullanılmıştır [98]. Son olarak tespit edilen öznitelik noktaları ($(\mathrm{d}_k, \mathrm{p}_k) \in \mathrm{P}_D$) denklem 3.21'e göre üç gruptan birine eklenmiştir. Bu denklemde $\mathcal{P}_T$ eşleşen hedef anahtar noktaları, $\mathcal{P}_B$ eşleşen arka plan anahtar noktaları ve $\mathcal{P}_N$ ise eşsiz anahtar noktaları göstermiştir [19].

$$\begin{cases} \mathrm{p}_k \in \mathcal{P}_T & d_k^{1N} \in T, \quad C(d_k, d_k^{1N}) > \theta_T, \quad \mathrm{r}(d_k) < \theta_r \\ \mathrm{p}_k \in \mathcal{P}_B & d_k^{1N} \in B, \quad C(d_k, d_k^{1N}) > \theta_B \\ \mathrm{p}_k \in \mathcal{P}_N & otherwise \end{cases} \tag{3.21}$$

Eşleşen nesne anahtar noktalar $(\mathrm{d}_k, \mathrm{p}_k) \in \mathrm{P}_T$ belirlendikten sonra orijinal şablondaki p_k^o'a karşılık gelen koordinatlar bulunmuştur. Bulunan $\mathcal{P}_T = \{(\mathrm{d}_k\ ,\ \mathrm{p}_k^o,\ p_k)\}_{k=1}^{N_m}$ noktası $\mathcal{P}_T$ kümesine eklenerek kümenin son hali elde edilmiştir.

Uzun süreli hafıza biriminde anahtar noktaları eşleştirmenin yanı sıra Nebehay ve Pflugfelder'in [99] yaptıkları çalışmadan yararlanılarak nesneyi tanımlayan *aktif bir anahtar nokta kümesi* oluşturulmuştur. Nesne takip başarısını arttırmak ve güvenilir anahtar noktaları elde etmek için İleri-Geri nesne takip [95] yöntemi kullanılmıştır. *İleri-Geri* nesne takip yönteminde $t-1$'den t'ye ileri optik akış ve t'den $t-1$'e geriye doğru optik akış tekniği ile nesne konumu hesaplanmıştır. Elde edilen iki sonuç arasındaki fark belli bir eşik değerden büyük ya da küçük olma durumuna göre herhangi bir izleme hatasının olup/olmama durumu belirlenmiştir [19]. Daha sonra *İleri-Geri* izleme tekniğinde başarılı olan sonuçlar P_t^A kümesinde toplanmıştır.

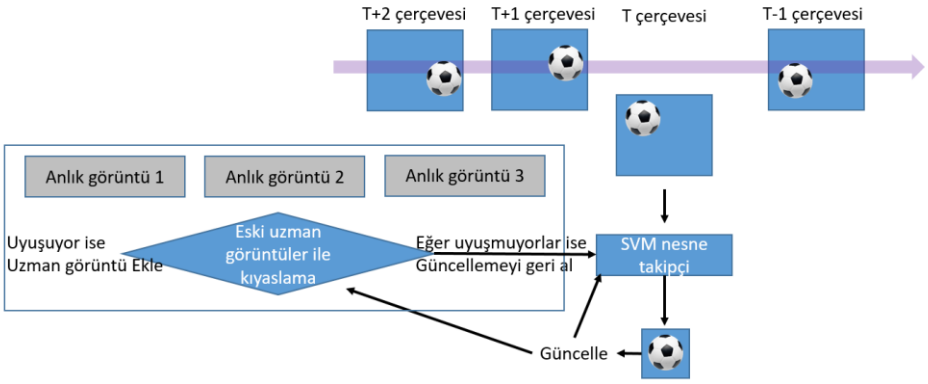
RANSAC Tahmini: Eşleşen ve izlenen anahtar noktalar elde edildikten sonra nesnenin durum bilgisini sağlamak için P_t^A ve P_T anahtar noktalarından oluşan bir $\mathcal{P}_C$ aday seti oluşturulmuştur. Benzerlik dönüşümü uygulanarak hedefin durumu $s_t = \{(x_t, y_t), s_t, \beta_t\}$ olarak tanımlanmıştır. Bu değerler sırasıyla yer değiştirme, ölçek ve dönme açısını ifade etmektedir. Hedef durumu $\mathcal{P}_C = \{(\mathrm{p}_i^o,\ p_i)\}_{i=1}^{N_C}$ tahmin etmek için $F_{S_t}(\mathrm{p}_i^o) \rightarrow p_i$ sağlayan bir dönüşüm uygulanmıştır. Anahtar nokta eşleştirmede oluşabilecek problemler karşısında RANSAC tahmincisi rastgele varsayımlarda bulunarak çözümler önermekte ve "doğru örnekler" ya da "yanlış örnekler" hesaplamaktadır. F_{S_t} tahmini yapıldıktan sonra nesne konumunu gösteren sınırlayıcı kutusunun (B) konumu ve boyutu hesaplanmıştır. Bu esnada, aykırı olmayan noktalar kümesindeki nokta sayısı izleme başarısında güçlü bir etken olmuştur.

Önerilen yapı sayesinde uzun süreli bellek birimi kademeli olarak güncellenmiştir. Nesne görünümünün kaybolması durumlarını kontrol etmek için kapanma anahtar noktalarının (occluding keypoints) $\mathcal{P}_o$ kümesi $\mathcal{P}_o = \mathcal{P}(B_I) \cap \mathcal{P}_B$ tanımlanmıştır. Buradaki $\mathcal{P}(B_I)$ hedef nesnenin sınırlayıcı kutu içindeki anahtar noktalar kümesidir. Sezgisel olarak kapanma yoksa $\mathcal{P}_o$ kümesinin eleman sayısı sıfıra yakın olmalıdır. Buna karşılık, hedef nesne kapanma problemine uğradığında, $\mathcal{P}_o$ kümesinin eleman sayısının yüksek çıkması gerekmektedir. Bu fikir doğrultusunda $\mathcal{P}_o$ nokta sayısı $\mathcal{P}_G = \mathcal{P}(B_I) \cap \mathcal{P}_t$ şeklinde gösterilen sınırlama kutusundaki anahtar nokta sayısına bölünerek hedef görünümün kaybolma (kapanma-occlusion) oranı hesaplanmıştır. Bu değer 0.5'ten büyükse "hedef kayboldu" şeklinde yorumlanıp güncellemeler durdurulmuştur [19].

MUSTer yönteminde hafıza birimindeki anahtar noktaların aşırı büyümesini engellemek için Hermann'ın [100] önerdiği unutma eğrisi kullanılmıştır [100]. Bu sayede nesne anahtar noktaları $\mathcal{T}$ ve arka plan anahtar noktaları B belli bir kapasitede ayarlanmış ve zaman içinde kullanılmayan bazı noktalar unutulmuştur [19].

3.10. MEEM

Zorlu nesne takip problemine getirilen çözümlerden bir tanesi de birden fazla sınıflandırıcı kullanarak nesne takibinin yapılmasıdır. Bu sınıflandırıcıların parametrelerini güncellemek için test işlemleri sırasında sınıflandırıcı tarafından nesne olarak etiketlenen örnekler kullanılmaktadır. Bu şekilde çalışan yöntemler her zaman kendi tahminleri hakkında daha emin olma eğilimindedir. Bu eğilim, yöntemler nesneyi kaybetmeye başladıklarında tamamen arka plana yönelme gibi problemlere yol açmaktadır. Zhang vd. [76] bu problemi çözmek için farklı bir yaklaşım getiren MEEM (Azaltma Kullanarak Birden Çok Uzman Vasıtasıyla Gürbüz Takip - Robust Tracking via Multiple Experts Using Entropy Minimization) yöntemini önermiştir (Şekil 3.12). Bu yöntemde güncelleme işleminde kullanılacak uzman seçiminde hata hesaplama fonksiyonu için minimum entropi yöntemi kullanılmıştır. Nesne takip sırasında DVM (destek vektör makineleri) sınıflandırıcısı tarafından yüksek olasılıkla nesne olarak kabul edilen görüntü parçaları uzman olarak seçilmektedir. Seçilen uzman, depolanan diğer uzmanlar ile uyuşmuyorsa DVM için yapılan güncelleme iptal edilmektedir (Şekil 3.12) [76].



Şekil 3.12. MEEM yönteminin temsili algoritması (Güncellemeyi geri almak için depolanan uzmanlar arasında en iyi uzman DVM parametrelerini tekrar günceller.)

MEEM nesne takip yöntemi tamamen çevrimiçi olarak çalışmaktadır. Takip edilecek nesnenin önceki konumuna bakılarak aynı merkezli ve $\sqrt{wh}$ (w/h önceki çerçevedeki genişlik ve yükseklikliktir) yarıçaplı bir çember içindeki alan, kartezyen ızgara şeklinde parçalara ayrılmıştır. Daha sonra bu parçalanmış alan içinden DVM sınıflandırıcı ile nesneye ait parçalar seçilerek nesne konumu belirlenmiştir. MEEM yöntemi öncelikle görüntüleri $CIELab$ renk uzayına dönüştürmüştür. Aşırı aydınlanma değişimlerine karşı yöntemi güçlendirmek için

görüntünün L kanalı üzerine bir parametrik olmayan yerel sıralama dönüşümü (non-parametric local rank transform) [101] uygulanmıştır. Bu dönüşüm, piksel yoğunluklarının monoton bir şekilde artmasına karşı değişmez bir öznitelik haritası üretir. Bu öznitelik haritasından ($F(L)$) ve CIE Lab renk kanalları ($[L, a, b]$) ile birlikte 4 boyutlu öznitelik kümesi ($[F(L), L, a, b]$) elde edilmiştir [76].

3.10.1. Çevrimiçi Doğrusal DVM Sınıflandırıcı

MEEM yöntemi çevrimiçi sınıflandırma için DVM algoritmasını nesne takibine daha iyi adapte olacak şekilde yeniden formüle etmiştir. Çevrimdışı DVM sınıflandırıcıya gelişmiş bir yaklaşım kazandırmak için prototip kümeleri kullanılmıştır. MEEM yönteminde kullanılan takipçi (DVM sınıflandırıcısı) T ile gösterilmiş ve önceki eğitim verilerini özetlemek için prototip kümesini $Q = \{\zeta_i = (\varphi\ (\mathrm{q}_i), \omega_i, \mathrm{s}_i\}_1^B$ şeklinde gösterilmiştir. q_i bir görüntü parçasını ve bu görüntünün öznitelik vektörü ise $\varphi(q_i)$ ile belirtilmiştir. w_i ikili bir etiket ve s_i kaç tane destek vektörünün (DV) kullanıldığını göstermektedir. Yeni gelen bir görüntü ($\mathcal{L} = \{\mathrm{x}_i, \mathrm{y}_i\}_1^J$) denklem 3.22'deki hatayı en aza indirgeyerek her karede DVM sınıflandırıcısını güncellemiştir.

$$\min_{w,b} \frac{1}{2}\|w\|^2 + C\left\{\sum_{i=1}^{B} \frac{\mathrm{s}_i}{N_{\mathrm{w}_i}} \mathrm{L}_h\left(\omega_i, \mathrm{q}_{i;} w\right) + \sum_{i=1}^{J} \frac{1}{N_{\mathrm{y}_i}} \mathrm{L}_h\left(\mathrm{y}_i, \mathrm{x}_{i;} w\right)\right\} \tag{3.22}$$

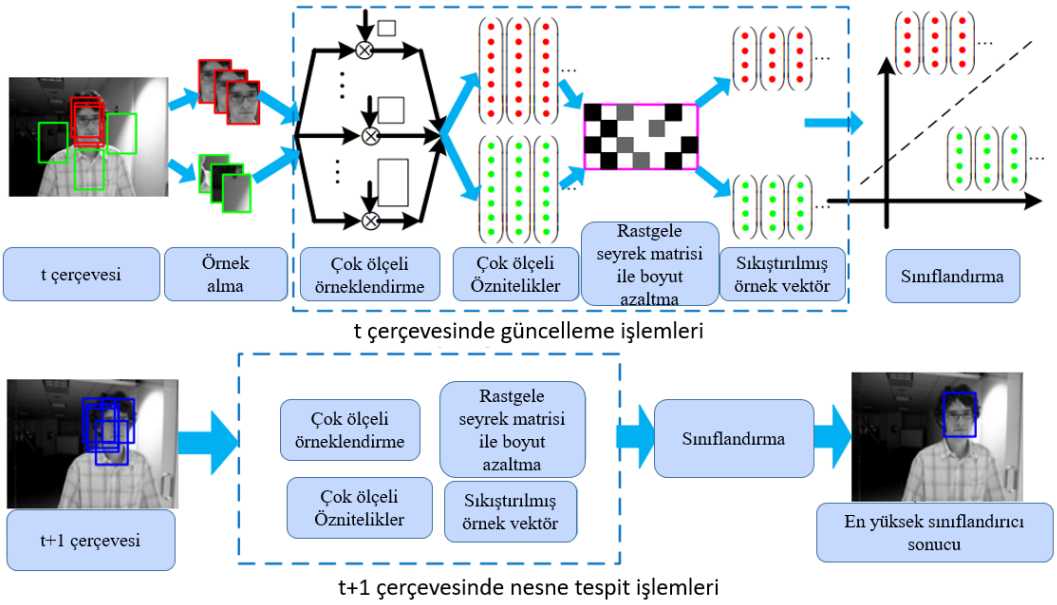
Bu denklemde L_h hinge kaybı [102] olarak belirtilmiştir. Genel olarak hinge kaybı örnek kümelerini eşitlemek ve aykırı örnekleri önlemek için kullanılmıştır [102]. Güncellemelerden sonra, çerçevedeki güncelleme verisi destek vektörlerin prototip kümesine eklenir. Prototip kümesinin boyutu önceden tanımlanmış bir değerden (B) daha büyük olduğunda, aynı etiketli prototip örnekleri minimum mesafeyle birleştirilir. Pozitif örnekler nesne içerdiğinden, genellikle izleme probleminde negatif olanlardan daha düşük çeşitliliğe sahip olmaktadırlar. MEEM yöntemi DVM takipçisinde bu durumu önlemek için, pozitif prototip örnek sayıları önceden tanımlanmış bir B^+ değerine ulaşana kadar birleştirilmemiştir. Doğrusal DVM ile doğrusal olmayan karar sınırlarını elde etmek için Maji ve Berg [103] tarafından önerilen öznitelik haritalama tekniği kullanılmıştır [76].

Çevrimiçi güncelleme yapan yöntemlerden farklı olarak MEEM yönteminde uzman (nesne görüntüsü) seçiminde yarı denetimli öğrenme problemi için geliştirilen minimum entropi [104] yöntemi kullanılmıştır [76]. Nesne takibinde nesne tespit edildikten sonra sınıflandırıcının kendi seçtiği örneklere minimum entropi uygulanarak daha farklı etiketlenmiş örnekler ile sınıflandırıcı güncellenmiştir. Minimum entropi yöntemi test zamanında örnekleri ağırlıklandırarak sınıflandırıcı başarısını olumsuz etkileyecek örnekler engellemiştir. Bu şekilde sınıflandırıcının doğruluğu korunmuştur.

3.11. FCT

Zhang ve vd. tarafından önerilen FCT (Hızlı Sıkıştırmalı Takipçisi - Fast Compressive Tracking) nesne takip yöntemi [53] 2012 yılında aynı yazarlar tarafından önerilen Real-Time CT (Sıkıştırmalı Takipçisi - Compressive Tracking) [105] yönteminin geliştirilmiş halidir. Bu yüzden öncelikle CT yöntemi ele alınmış ve daha sonra iki yöntem arasındaki fark detaylandırılmıştır.

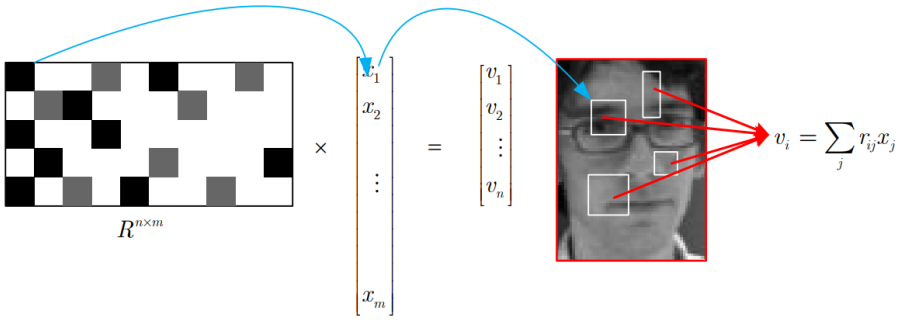
CT yönteminde temel olarak çerçeveden alınan örnek görüntüler sıkıştırılarak Naive Bayes sınıflandırıcısına gönderilmiş ve sınıflandırıcı sonucuna göre nesne konumu belirlenmiştir. Şekil 3.13'te CT algoritmasının ana bileşenleri gösterilmiştir.



Şekil 3.13. CT yönteminin genel akışı [105]

CT yöntemi izleme ve güncelleme olmak üzere iki aşamadan oluşmaktadır. İzleme aşamasında, $t-1$ çerçevesindeki nesne konumu referans alınarak t çerçevesindeki nesne görüntü adayları alınmıştır. Daha sonra bu parçaların integral vektörleri hesaplanmıştır [106]. Sıkıştırma algılama teorisine dayanılarak, örneklerin statik bir ölçüm matrisi yardımıyla çıkartılan haar öznitelikleri sıkıştırmıştır. Sıkıştırma işlemi $v = Ru$ şeklinde gösterilmiştir. Burada $u \in R^m$ integral vektörünü ve $v \in R^n$ değeri $n \ll m$ boyutlarına sahip sıkıştırılmış öznitelik vektörünü ifade etmektedir. R parametresi denklem 3.23'te gösterildiği gibi tanımlanmış seyrek rastgele matristir. Bu denklemde s değeri $\frac{m}{4}$ olarak alınmıştır [105]. Belirli bir görüntüye yukardaki işlemlerin uygulanmış hali Şekil 3.14'te gösterilmiştir [105].

$$r_{i,j} = \sqrt{s} \times \begin{cases} +1, & with\ probability\ \frac{1}{2s} \\ 0, & with\ probability\ 1-\frac{1}{s} \\ -1, & with\ probability\ \frac{1}{2s} \end{cases} \qquad v_i = \sum_j r_{ij} x_j \qquad (3.23)$$



Şekil 3.14. Rastgele seyrek matrisi ile örneklerin sıkıştırılması

Elde edilen düşük boyutlu öznitelik vektörleri çevrimiçi Naive Bayes sınıflandırıcıya gönderilmiştir (denklem 3.24). Maksimum sınıflandırıcı çıkışı sağlayan örnek, t çerçevesinde nesne konumu olarak atanmıştır.

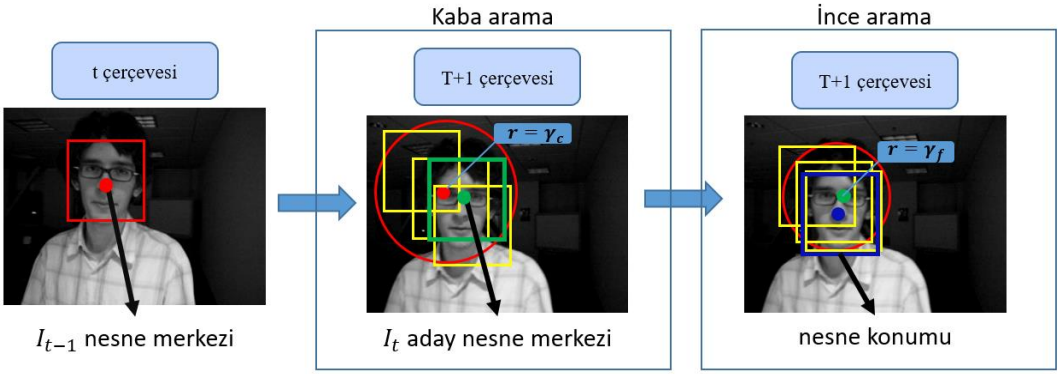
Güncelleme aşamasında nesne ve arka planın eğitim örnekleri, t çerçevesinde takip sonucuna göre örneklenir. Eğitim örneklerinin sıkıştırılmış öznitelik vektörleri, sınıflandırıcı parametrelerini güncellemek için kullanılmıştır [105].

$$H(v) = \log\left(\frac{\prod_{i=1}^{n} p(v_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(v_i|y=0)p(v_i|y=0)}\right) = \sum_{i=1}^{n} \log\left(\frac{p(v_i|y=1)}{p(v_i|y=0)}\right) \qquad (3.24)$$

Nesne konumunu daha doğru ve hızlı bulmak için CT [105] yöntemine kaba-ince (coarse to fine) arama stratejisi uygulanarak FCT [53] algoritması önerilmiştir. FCT yöntemi CT yönteminden farklı olarak iki aşamada nesne tespiti yapmıştır. Öncelikle geniş bir arama alanı alınarak bu alanda aday nesne konumu belirlenmiştir. Daha sonra aday nesne konumlarının ortalaması alınır ve birinci arama alanına göre daha küçük bir arama alanı belirlenerek tekrar bir nesne tespit adımı gerçekleştirilir. Buradaki temel amaç, ilk adımda büyük arama alanı ile daha genel bir nesne tespiti yapılması ve ardından bu sonuç doğrultusunda daha küçük bir arama alanı ile daha kesin bir nesne tespiti yapılmasıdır. Öznitelik çıkartılmasında ise seçilen arama alanının Haar özniteliklerinden faydalanılmıştır.

Arama alanlarının seçilmesi Şekil 3.15'te gösterilmiştir. Burada temel olarak önceki çerçevedeki nesne konumu merkezdeyken γ_c yarıçaplı bir alan alınmıştır. Bu arama alanında CT yöntemi ile aynı fakat daha az örnekler ile aday nesne konumu bulunmuştur. İkinci arama alanı seçiminde ise tespit edilen aday nesne konumu merkezdeyken γ_f yarıçaplı bir alan seçilmiştir. Bu alanda tekrar bir tespit işlemi gerçekleştirilerek son nesne konumu belirlenmiştir. Burada önemli olan nokta alan yarıçaplarının $\gamma_f < \gamma_c$ şeklinde ikinci arama alanının oldukça küçük olmasıdır. Bu şekilde CT yönteminin kullandığı geniş alana kıyasla

daha kesin sonuç veren iki aşamalı bir arama yöntemi kullanılmıştır. FCT yönteminin kaba-ince arama tekniği, CT yöntemine göre daha az sayıda aday nesne konumu elde etmiştir. Bu sayede CT yöntemine göre hız performansı iyileştirilmiştir.



Şekil 3.15. FCT yönteminin kaba ince arama yaklaşımı [53]

4. DENEYSEL SONUÇLAR

Bu tez çalışmasında ele alınan yöntemlerin detaylı analizi için güçlü bir veri kümesi ile kapsamlı testler gerçekleştirilmiştir. Test veri kümesi literatürde bulunan farklı veri kümelerinden seçilen 18 video ve bu tez çalışması için alınan 7 videodan oluşmaktadır. Ele alınan 5 güncel derin öğrenme ve 5 geleneksel nesne takip yöntemi için gerekli kodlar ilgili çalışmalardan referans alınarak temin edilmiştir.

Yöntemlerin kodlamaları orijinal algoritma yapıları bozulmadan ortak bir yazılım alanında çalışacak şekilde biçimlendirilmiştir. Ekran kartı üzerinde çalıştırılan derin öğrenme tabanlı yöntemler için CUDA 8.0 ekran kartı yazlımı ile Cudnn 6.0 ve matconvnet kütüphaneleri kullanılmıştır. Ele alınan yöntemler mümkün olduğu kadar ortak çalışma platformuna (Windows, Matlab 2017 ve ilgili kütüphaneler) bağlı kalınarak test edilmiştir. Caffe derin öğrenme platformuna bağlı çalışan GOTURN yöntemi Windows işletim sistemi üzerinde beklenen performansı göstermediği için Linux Ubuntu 16.0 versiyonu üzerinde çalıştırılmıştır. Ayrıca, çevrimdışı ön eğitim isteyen yöntemlerde orijinal ön eğitim modeli kullanılmıştır. Ele alınan yöntemler ve yöntemlerin alındığı web adresleri Çizelge 4.1'de verilmiştir.

Çizelge 4. 1. Ele alınan yöntemler ve kaynak kodların alındığı web adresleri

Yöntem	**Kodların Temin Edildiği Web Adresleri**	**Erişim Tarihi**
GOTURN	http://davheld.github.io/GOTURN/GOTURN.html	01.12.2017
SiamFC	http://www.robots.ox.ac.uk/~luca/siamese-fc.html	01.01.2018
ADNet	https://sites.google.com/view/cvpr2017-adnet	01.02.2018
DCFNet	https://github.com/foolwood/DCFNet#dcfnet-discriminant-correlation-filters-network-for-visual-tracking	01.02.2018
HCFT	https://sites.google.com/site/jbhuang0604/publications/CF2	01.01.2018
BIT	http://caibolun.github.io/BIT/index.html	01.12.2017
MUSTer	https://sites.google.com/site/multistoretrackerMUSTer/	01.02.2018
BACF	http://www.hamedkiani.com/bacf.html	01.12.2017
MEEM	http://cs-people.bu.edu/jmzhang/MEEM/MEEM.html	01.01.2018
FCT	http://www4.comp.polyu.edu.hk/~cslzhang/FCT/FCT.htm	01.12.2017

4.1. Test Veri Kümesinin Hazırlanması

Yeni geliştirilmiş bir nesne takip yönteminin performansı hakkında güçlü bir analiz ve değerlendirmenin yapılabilmesi için kullanılacak veri kümesinin özenle seçilmesi gerekmektedir [12]. Test veri kümesinin nesne takibi sırasında karşılaşılması muhtemel ışık değişimi (ID), boyut değişimi (BD), kapanma (KP - occlusion), nesne görünüm değişimi (GD), ani hareket değişimi (AHD) ve arka plan dinamikliği (APD) gibi problemleri içermesi gerekmektedir. Ayrıca çevrim dışı eğitim gören yöntemlerin eğitimi için kullanılan verilerin, test veri kümesinin dışında tutulması gerekmektedir.

Bu tez çalışmasında test veri kümesinin oluşturulması için literatürde yaygın bir şekilde kullanılan CVPR2013 [107], VOT2013 [90], VOT2014 [89], VOT2015 [25] ve ALOV [12] nesne takip veri tabanları analiz edilmiştir. Analiz işleminde özellikle üç noktaya dikkat edilmiştir. İlk olarak seçilecek videoların içeriği ve takip edilecek nesne biçimi yöntemin geliştirilme amacına uygun olmalıdır. İkincisi ise çevrim dışı eğitim gören yöntemlerin kıyaslanması için kullanılacak verilerin daha önce yöntemin eğitiminde kullanılmamış olması gerekmektedir [15], [71]. Son olarak video veri kümesinde kullanılan video içeriklerin birbirinden farklı olması gerekmektedir [12]. Bu analizler doğrultusunda ele alınan veri tabanlarından belirlediğimiz 18 video seçilmiştir. Bunun yanında ikinci önemli nokta ise eğitim ve test örneklerinin benzerliğinin olabildiğince engellenmesinin sağlanmasıdır. Bu amaçla tez çalışmaları sırasında içerisinde nesne takip problemleri içeren 7 farklı video görüntüsü kaydedilerek yöntemler için ortak bir test veri tabanı oluşturulmuştur.

Yöntemlerin zorluklar karşısındaki başarılarını ölçmek için [12], [25], [33], [66], [71], [75], [76], [107] çalışmaları referans alınarak her videonun içerdiği problemler belirlenmiştir. Çizelge 4.2'de oluşturulan test veri kümesi ve içerdiği problemeler gösterilmiştir. Bu tez çalışması için alınan video sahneleri koyu renk ile belirtilmiştir. Oluşturulan test veri kümesi toplam 10438 çerçeveden oluşan 25 video içermektedir.

Çizelge 4. 2. Test veri kümesi

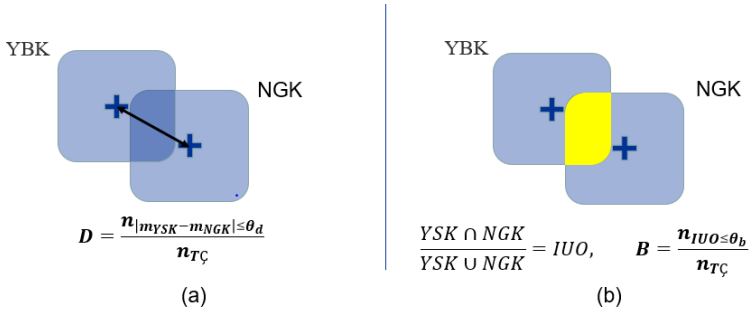
Video		Toplam Çerçeve	Nesne Takip Problemleri					
			KP	ID	AHD	GD	APD	BD
1	**ayi1**	**250**			✓	✓		✓
2	**ayi2**	**221**	✓		✓	✓		
3	ball	603			✓	✓		
4	ball1	105	✓		✓		✓	
5	basketball	725	✓		✓		✓	
6	birds1	339			✓	✓	✓	✓
7	bmx	76			✓	✓	✓	✓
8	book	175	✓	✓	✓	✓	✓	✓
9	**yuzTakip**	**617**		✓			✓	✓
10	fish2	310	✓		✓	✓	✓	✓
11	gymnastics3	118		✓	✓	✓	✓	✓
12	helicopter	708				✓		✓
13	human	412				✓		✓
14	iceskater2	707	✓		✓	✓	✓	
15	jogging	307	✓				✓	
16	lemming	1336	✓			✓		
17	motocross2	61			✓	✓	✓	
18	pedestrian1	140			✓		✓	
19	road	558	✓		✓			
20	**ozCekim**	**921**		✓		✓		
21	sheep	251				✓	✓	
22	soldier	138	✓		✓			✓
23	**surahi**	**448**	✓					✓
24	**morKutu**	**246**	✓		✓	✓	✓	✓
25	**yuruyus**	**666**	✓		✓	✓		✓

4.2. Değerlendirme Ölçütleri

Yöntemlerin nesne takibindeki başarılarını hesaplamak için literatürde yaygın bir şekilde kullanılan duyarlılık (Precision) ve doğruluk (Sucsees rate) parametreleri kullanılmıştır [75], [107]. Duyarlılık merkez konum hatasına göre nesnenin bulunduğu çerçeve sayısının tüm çerçeve sayısına oranını göstermektedir (Şekil 4.1.a). Merkez konum hatası nesne takip yönteminin bulduğu sınırlama kutusu (YBK) ile nesnenin gerçek sınırlama kutusunun (NGK) merkezleri arasındaki Öklid uzaklığı olarak tanımlanmıştır [107]. Belirlenen bu mesafe belirli bir eşik değerin (θ_d) altında ise nesne bulunmuş varsayılmaktadır (denklem 4.1).

$$D = \frac{n_{|m_{YSK} - m_{NGK}| \leq \theta_d}}{n_{TÇ}} \tag{4.1}$$

Bu denklemde $|m_{YSK} - m_{NGK}|$ ele alınan her hangi bir çerçevedeki YBK ve NGK sınırlama kutuların merkezler arasındaki Öklid uzaklığını hesaplamıştır. Bu uzaklık belirlenen θ_d değerinden küçük olması durumunda nesnenin bulunduğu varsayılmıştır. $n_{|m_{YSK}-m_{NGK}|\leq\theta_d}$ parametresi ise nesnenin bulunduğu varsayılan toplam çerçeve sayısını ve $n_{TÇ}$ değeri video içeresinde bulunan toplam çerçeve sayısını göstermiştir.



Şekil 4. 1. Nesne takip yöntemlerinde (a) duyarlılık ve (b) doğruluk değerlerin hesaplanması

Bir diğer değerlendirme tekniği olan doğruluk değeri ise YBK ile NGK'nın $kesişim/birleşim(IUO)$ oranına bakılarak hesaplanmaktadır (Şekil 4.1.b). Hesaplanan IUO değerinin belirlenen eşik (θ_b) değerinden küçük olduğu çerçevelerde nesnenin tespit edildiği varsayılmaktadır [107]. Doğruluk değeri, IUO yardımı ile nesnenin başarılı bir şekilde tespit edildiği çerçeve sayısının toplam çerçeve sayısına göre oranı ile ifade edilmektedir (denklem 4.2).

$$\frac{YSK \cap NGK}{YSK \cup NGK} = IUO, \quad B = \frac{n_{IUO\leq\theta_b}}{n_{TÇ}} \tag{4.2}$$

Bu tez çalışması için eşik değerler, Wu vd. [107] önerdiği değerler ile aynı olacak şekilde $\theta_d = \{1, 2, 3, \ldots, 50\}$ ve $\theta_b = \{0.01,\ 0.02, \ldots, 1\}$ olarak alınmıştır. θ_d değerinin artması ve θ_b değerinin azalması hata payını arttırmaktadır. Yöntemlerin başarı sırasını elde etmek için nesnenin bulunduğu varsayılan eşik değerler sırasıyla $\theta_d = 20$ ve $\theta_b = 0.5$ şeklinde alınmıştır [107].

Duyarlılık ve doğruluk analizleri için Wu vd. tarafından önerilen 3 farklı duyarlılık (Precision) parametresi ve bunların sırasıyla doğruluk (Sucsees rate) çizelgeleri kullanılmıştır. Bunlar sırasıyla OPE (Tek Geçişli Değerlendirme - One-Pass Evaluation) , TRE (Zamansal Gürbüzlük Değerlendirmesi - Temporal Robustness Evaluation), SRE (Konumsal Gürbüzlük Değerlendirmesi - Spatial Robustness Evaluation) şeklindedir. Detaylandıracak olursak; OPE ilk çerçevede nesne görüntüsü verildikten sonra video boyunca nesne tespit etme başarı oranını hesaplamaktadır. TRE değerlendirme biçimi, takip işlemini videodaki rastgele seçilen bir çerçeveden başlatarak nesnenin takip edilme oranını hesaplamaktadır. Son olarak SRE ise, nesneyi çerçeveleyen sınırlama kutusunun boyutunda yapılan değişimler ile yöntemin nesneyi takip etme oranını hesaplamaktadır.

SRE sonuçları için nesne takip yöntemlerine başlangıç konumu olarak nesne konumunu gösteren sınırlama kutusunun boyutu %2 oranında küçültülüp/büyültülerek verilmiştir. Bu şekilde elde edilen iki farklı sonucun ortalaması alınarak ortalama SRE sonucu elde edilmiştir [107]. TRE sonuçları için yöntemlere ilk çerçeve olarak videonun rastgele bir çerçevesi verilerek nesne takibi başlatılmıştır. Bundan sonra gelen çerçevelerdeki elde edilen ortalama sonuç TRE sonucunu göstermiştir. Testler Intel i7 6700HQ işlemci, 16 GB RAM ve GeForce GTX 960M ekran kartı donanımına sahip bir bilgisayarda gerçekleştirilmiştir.

Bu tez çalışmasında yöntemlerin değerlendirmeleri elde edilen sonuçlar doğrultusunda 3 ana başlık altında ele alınmıştır. İlk olarak genel performans için yöntemlerin OPE, TRE, SRE duyarlılık ve doğruluk sonuçları incelenerek genel performans değerlendirmesi yapılmıştır. İkinci başlık altında nesne takip yöntemlerinin problem bazlı sonuçlarını incelemek üzere ilgili problemi içeren sınırlı sayıda video kullanılmış ve değerlendirmeler yapılmıştır. Buradaki temel amaç geliştirilen yöntemlerin ışık değişimi, boyut değişimi, kapanma, nesne görünüm değişimi, ani hareket değişimi ve arka plan dinamikliği problemleri karşısındaki performanslarını ayrı ayrı değerlendirilmiştir. Son başlıkta ise yöntemlerin hız performansı analiz edilmiştir.

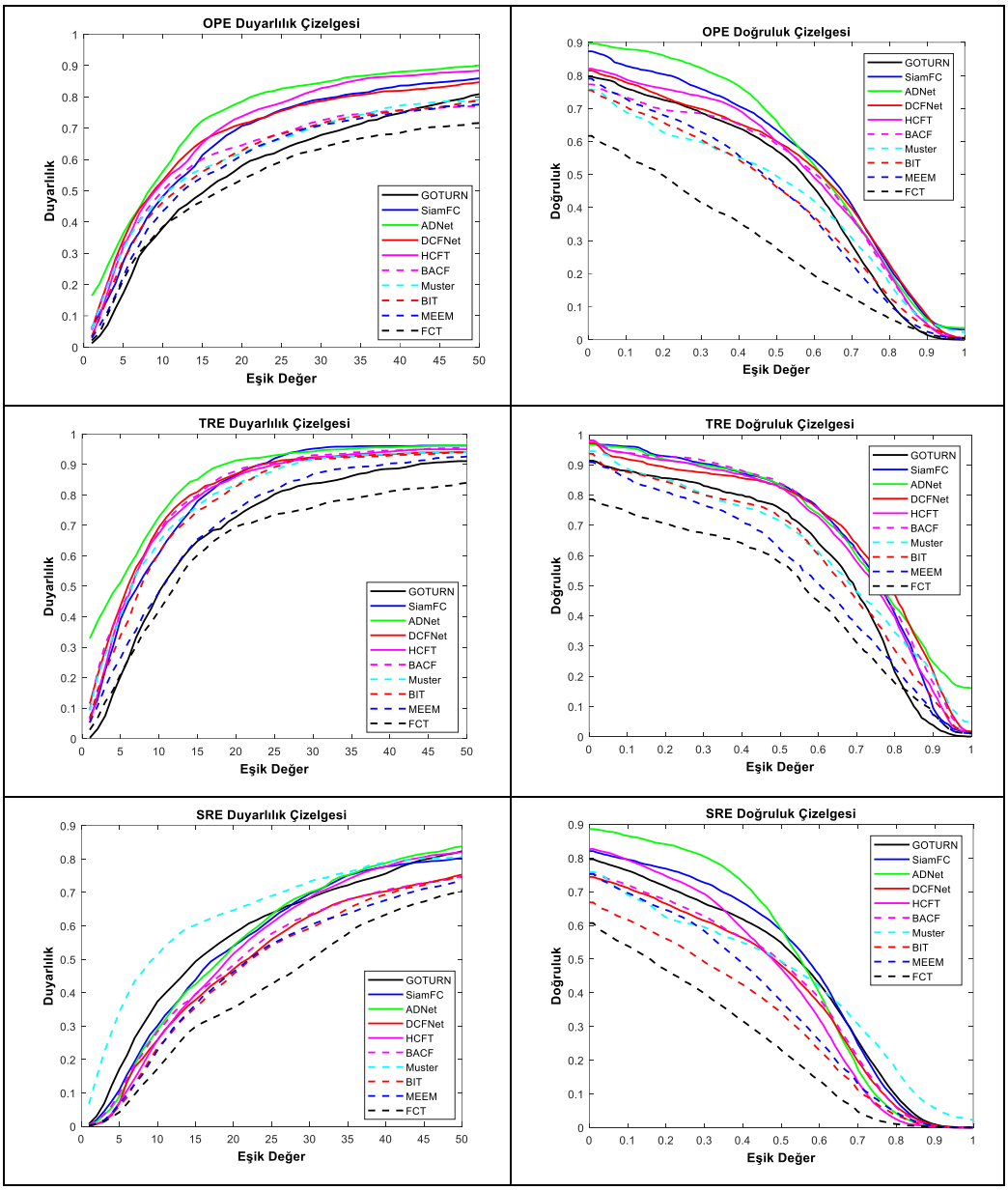
4.3. Genel Performans

Nesne takip yöntemlerinin tüm veri kümesi üzerinde elde edilen ortalama performans sonuçları Şekil 4.2'de gösterilmiştir. OPE duyarlılık sonuçlarına ($\theta_d = 20$ için) bakıldığında ADNet yöntemi diğer yöntemlere kıyasla daha başarılı bir sonuç elde etmiştir. Başarı sıralaması HCFT, DCFnet ve SiamFC şeklinde devam etmektedir. SiamFC yöntemi gelişmiş ağ mimarisi sayesinde nesne boyut değişimine daha iyi uyum sağlayabildiği için SRE doğruluk sonuçlarında DCFnet ve HCFT yöntemlerini geride bırakmıştır.

Elde edilen OPE sonuçlarında derin öğrenme tabanlı yöntemler kullandıkları derin öznitelik sayesinde nesne görünümüne karşı daha duyarlı olmuş ve daha güçlü başarılar elde etmişlerdir. Bunun yanında korelasyon tabanlı sınıflandırıcı kullanan yöntemler (BACF, DCFNet, HCFT gibi) nesne ile arka planı güçlü bir şekilde ayırt edebildikleri için başarılı sonuçlar elde etmişlerdir.

Elde edilen en şaşırtıcı sonuç GOTURN yönteminde gözlemlenmiştir. Kullanılan eşik değerlerin artmasıyla diğer yöntemlere göre doğruluk daha hızlı bir şekilde düşmüştür. Bunun temel sebebi GOTURN yönteminin çevrimiçi güncelleme yapmaması ve nesnenin kaybolması veya görünüm değişikliğine uğraması durumunda bu yöntemin nesneyi unutma ihtimalinin yüksek olmasıdır. Bununla birlikte uzun hafıza birimindeki anahtar nokta tanımlayıcı sayesinde MUSTer yöntemi doğruluk sonuçlarındaki eşik değerin (θ_b) artmasına karşı, elde etiği başarıyı diğer yöntemlere göre daha iyi korumuştur. Geleneksel nesne takip

yöntemlerinden olan ve derin öğrenme kullanmayan BACF yöntemi arka plan örnekleri ile güçlü bir şekilde eğitilen korelasyon filtresi sayesinde, derin öğrenme yöntemleri ile yarışacak bir düzeyde başarı yakalamıştır.



Şekil 4.2. İncelenen yöntemlerin OPE, TRE ve SRE parametreleri açısından duyarlılık ve doğruluk sonuçları

Şekil 4.3'te ki SRE duyarlılık sonuçları incelendiğinde sınırlama kutusunun boyut değişikliğinin MUSTer yöntemini olumlu yönde etkilediği gözlemlenmektedir. Fakat buna karşı SRE başarı hesaplama tekniğinde sınırlama kutusu hatalı verildiği için, MUSTer yöntemi nesnenin gerçek boyutunu ayarlayamamakta ve SRE doğruluk sonucu düşük kalmaktadır.

Çizelge 4.3'te yöntemlerin $\theta_d = 20$ ve $\theta_b = 0.5$ eşik değerleri için duyarlılık ve doğruluk sonuçları verilmiştir. Burada kendi alanında en yüksek başarıları elde eden yöntemlerin sonuçları kalın ve italik olarak belirtilmiştir. Yapılan testler sonucunda ortalama

en yüksek başarıyı sağlayan yöntem ADNet yöntemi olmuştur. Bunun temel sebebi; ADNet yöntemi hareket modelini öğrenen gelişmiş ağ mimarisi kullanması ve ADNet yöntemindeki sırası gelen çerçevede nesne bulunduktan sonra bulunan nesne konumun doğruluğunu denetleyen güven katmanı ile hatalı sonuçları engellemesidir. Özellikle kapanma problemine karşı ADNet yönteminin kullandığı güven katmanı sayesinde, nesne ortaya çıktığında tekrar nesneyi tespit edip takibe devam edebilmiştir. Başarı sonuçlar elde eden bir diğer yöntem ise SiamFC yöntemi olmuştur. Bu yöntemin özniteliklerin çıkarıldığı katmanların test aşamasında güncellenmemesi hız performansı açısından avantaj sağlamıştır. Öznitelik katmanından hemen sonra gelen korelasyon katmanı çevrim dışı olarak güçlü bir şekilde eğitildiği için, bu yöntem nesne görünüm değişikliği, ışık değişimi ve boyut değişimi gibi problemler karşısında oldukça dayanıklı kalabilmiştir. Fakat SiamFC yönteminin ani hareket değişimleri karşısında arama alanını ayarlayamaması nesneyi kaybetmesine sebep olmuştur. SiamFC yöntemine benzer bir yapıda olan DCFnet yöntemi gelişmiş ve basit ağ mimarisi sayesinde ilk sıralarda yer alan ADNet ve SiamFC yöntemlerine yakın bir başarı elde etmiştir. BACF yöntemi değişken çerçeve boyları ile yapılan TRE doğruluk sonuçlarında $\theta_b = 0.5$ eşik değeri için tüm yöntemleri geride bırakarak en yüksek skoru elde etmiştir. SRE duyarlılık sonuçlarında ise MUSTer yöntemi 0.646 duyarlılık sonucu ile en iyi skoru elde etmiştir. Bunun temel sebebinin yapılan SRE testinde sınırlama kutusunun boyutunun değişmesi ile MUSTer yönteminin daha güçlü anahtar noktalar toplayabilmesidir.

Çizelge 4. 3. $\theta_d = 20$ ve $\theta_b = 0.5$ eşik değerlerindeki OPE, TRE, ve SRE için duyarlılık ve doğruluk sonuçları

Yöntem	**Duyarlılık ($\theta_d = 20$)**			**Doğruluk ($\theta_b = 0.5$)**		
	OPE	**TRE**	**SRE**	**OPE**	**TRE**	**SRE**
GOTURN	0,580	0,728	0,577	0,582	0,763	0,557
SiamFC	0,708	0,865	0,539	0,643	0,838	0,594
ADNet	***0,786***	***0,913***	0,541	***0,674***	0,839	***0,605***
DCFNet	0,714	0,869	0,471	0,608	0,834	0,491
HCFT	0,738	0,861	0,517	0,611	0,832	0,483
BACF	0,646	0,879	0,484	0,601	***0,848***	0,502
MUSTer	0,620	0,833	***0,646***	0,504	0,721	0,501
BIT	0,628	0,827	0,456	0,473	0,736	0,350
MEEM	0,614	0,747	0,463	0,479	0,637	0,386
FCT	0,536	0,696	0,355	0,288	0,584	0,241

Derin öznitelikleri kullanması açısından hiyerarşik bir bakış açısı sağlayan HCFT yöntemi nesneyi oldukça iyi bir şekilde tanımış ve bu sayede görünüm değişikliklerine karşı avantaj sağlamıştır. Fakat bu işlemlerin zaman alması hız performansının zayıf kalmasına sebep olmuştur (bkn. 4.5. bölüm). GOTURN yöntemi çevrimiçi gü!ncelleme yapmadığı için

kapanma problemine karşı zayıf kalmıştır. Çevrimiçi güncelleme yapılmış olsa idi, GOTURN yöntemi ardışık çerçevelerde meydana gelen kapanma gibi durumlar ile güncelleme yapacak ve yöntemin bu durum ile başa çıkmasına katkı sunabilecekti.

Yapılan testlerde oldukça şaşırtıcı sonuçlar elde eden ve geleneksel makine öğrenmesi algoritmaları ile nesne takibi yapan BACF yöntemi ağırlıklı olarak arka plan örnekleri ile eğitilen korelasyon filtresi sayesinde derin öğrenme yöntemleri ile yarışacak bir başarı yakalamıştır. Bu yöntem günümüzdeki zorlu nesne takip problemi için korelasyon filtresinin tek bir nesne örneği ile eğitilmesi yerine geniş arama alanındaki bir çok sayıda arka plan örnekleri eğitilmiştir. Bu sayede eğitilen korelasyon filtresi nesne ile arka planı daha güçlü bir şekilde ayırabilmiştir.

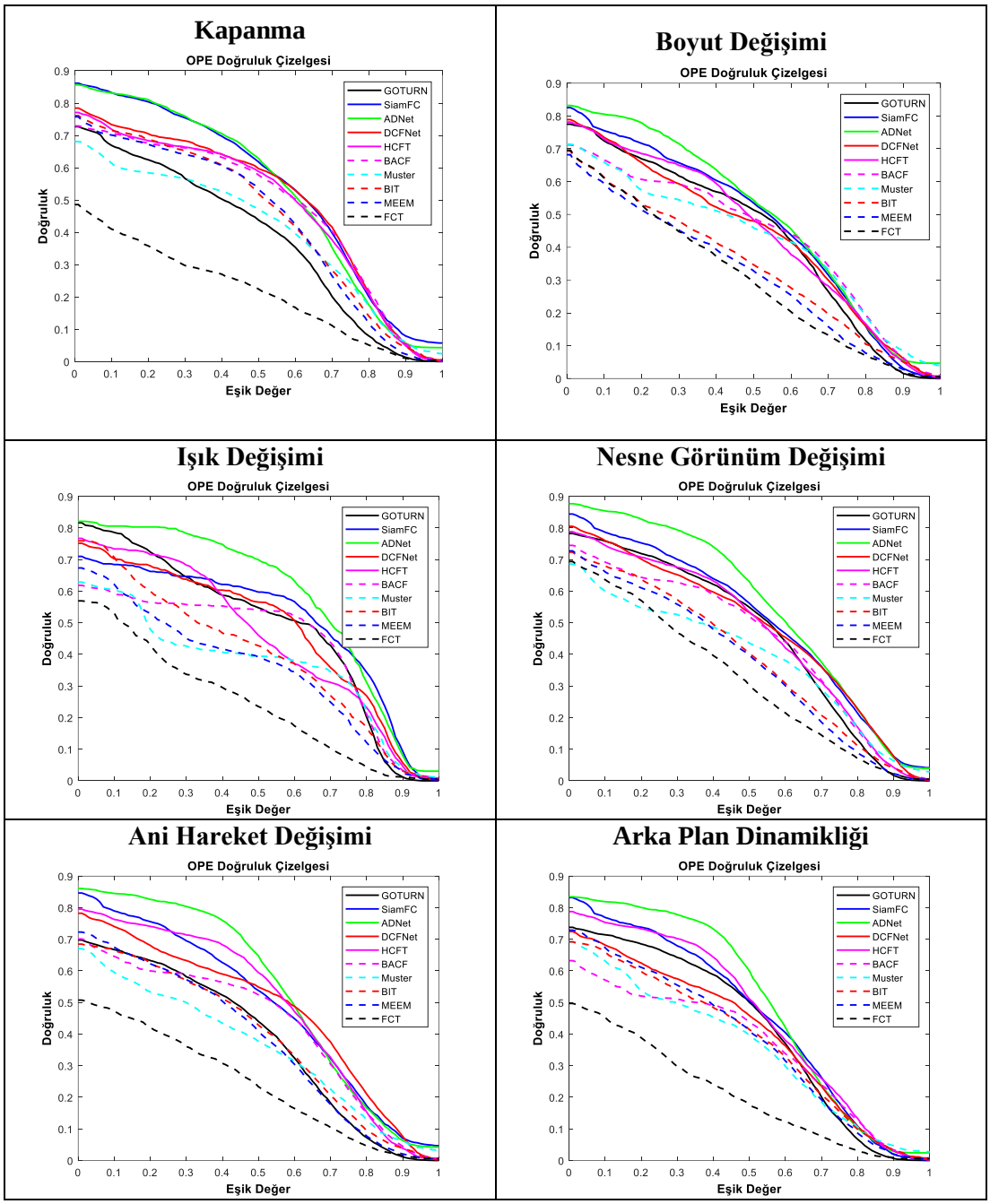
MUSTer yönteminde kısa hafıza birimindeki KCF (Çekirdekleştirilmiş İlinti Süzgeci - Kernelized Correlation Filters) [60] ve DSSCF (Ayırt edici Ölçek Alan Korelasyon Filtresi - Discriminative Scale Space Correlation Filter) [97] yöntemlerini bir arada kullanan EKF (Entegre Korelasyon Filtresi - Integrated Correlation Filter) kısa süreli nesne takibinde güçlü bir yöntem olmuştur [19]. Bunun yanında MUSTer yöntemi kullandığı denetleyici sayesinde EKF birimindeki hatalı güncellemeleri engelleyerek uzun süreli nesne takip senaryolarında başarılı sonuçlar elde etmiştir.

Kapanma problemine karşı en başarılı sonuçları elde eden yöntemlerden biri de MEEM yöntemi olmuştur. Kapanma problemi olduğu durumlarda geliştirilen çoklu uzman seçim algoritması sayesinde hatalı güncellemeleri geri alarak arka plan kaymalarını engellemiş ve nesne tekrar ortaya çıktığında nesneyi takibe devam etmiştir. Bu da yöntemin kapanma problemine karşı oldukça dayanıklı olmasını sağlamıştır. BIT yöntemi insan görme sistemini taklit eden ve dört bölümden oluşan bir nesne takip algoritmasıdır. Bölüm 3'te detaylandırılan ve yönteme ait olan S1 ve C1 birimlerinde çıkartılan kenar ve renk adları öznitelikleri sayesinde, başarı sıralamasında ilk sıralarda yer almasa da geleneksel nesne takip yöntemleri arasında yüksek bir başarı sağlamıştır. FCT yöntemi yapılan testlerde diğer yöntemler arasında en düşük başarı gösteren yöntem olmuştur. Bunun temel sebebi kullandığı öznitelikler ve DVM sınıflandırıcının nesne takibindeki değişen arka plan karşısında yetersiz kalmasıdır. Özellikle Haar öznitelikleri kapanma ve gürültü gibi durumlarda ayırt edici öznitelikler çıkartamadığı için, yöntemin sınıflandırıcısı kaliteli öznitelik ile eğitilememektedir. Fakat önerilen öznitelik sıkıştırma algoritması sayesinde, FCT yöntemi hız performansı açısından ikinci sırada yer almaktadır (bölüm 4.5 bkn.)

4.4. Yöntemlerin Nesne Takip Problemleri Karşısındaki Başarıları

Güçlü bir nesne takip yöntemi, nesne takibi sırasında karşılaşılan problemlere rağmen nesneyi kaybetmeden takibe devam edebilmelidir. Bu bölümde değerlendirmeye alınan

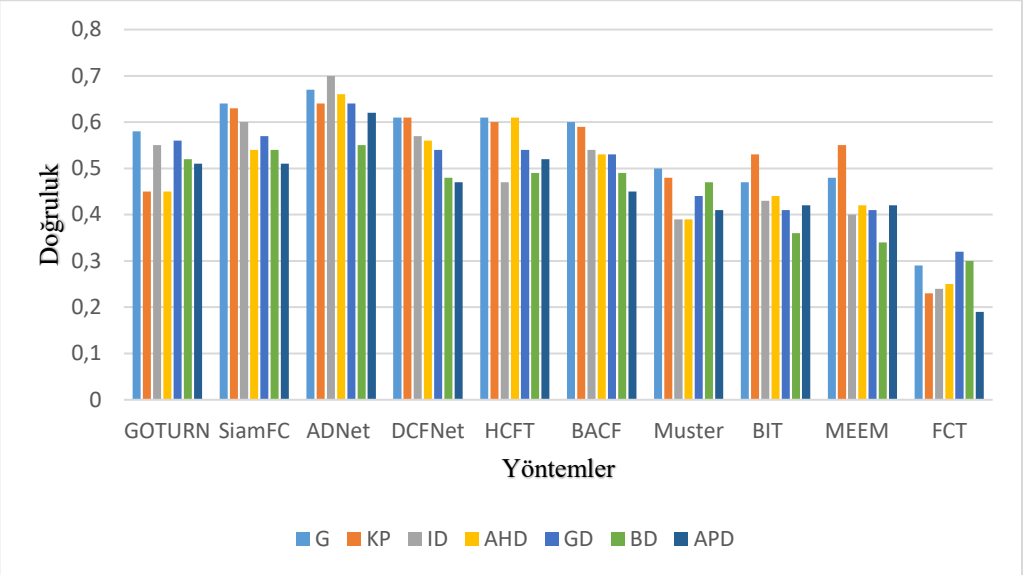
yöntemlerin nesne takibinde karşılaşılan ışık değişimi (ID), boyut değişimi (BD), kapanma (KP: occlusion), nesne görünüm değişimi (GD), ani hareket değişimi (AHD) ve arka plan dinamikliği (APD) problemleri karşısındaki başarılarını incelenmiştir. Tüm bu problemleri içerecek şekilde inşa edilen alt test veri grupları ile kapsamlı testler yapılmıştır. Her bir problem için test veri setindeki ilgili problemi içeren videolar bir araya getirilerek 6 farklı alt veri kümesi elde edilmiştir (Çizelge 4.2). Daha sonra ele alınan yöntemler bu veri kümeleri kullanılarak ayrı ayrı testlere tabi tutulmuştur. Bu şekilde yöntemlerin nesne takip problemleri karşısındaki ortalama başarıları hesaplanmıştır. Her bir problemi içeren veri kümeleri için ayrı ayrı yapılan testler sonucunda elde edilen OPE doğruluk sonuçları Şekil 4.3'te verilmiştir. Sonuçlara bakıldığında, neredeyse her problem için ADNet yöntemi oldukça güçlü sonuçlar elde etmiştir. Bunun temel sebebi ağ mimarisindeki güven katmanı sayesinde nesne takibinde yapılan hatalı sonuçları engelleyebilmesidir. MUSTer yöntemi ADNet ile benzer yapıda olmasına rağmen, kullandığı geleneksel YGH öznitelikleri yetersiz kaldığı için daha düşük bir başarı elde etmiştir.



Şekil 4. 3. Problem bazlı olarak yöntemlerin OPE doğruluk sonuçları.

OPE doğruluk çizelgelerinin eşik değeri 0.5 için elde edilen problem bazlı sonuçlar ise Şekil 4.4'te verilmiştir. Genel başarı (G) ile kıyaslandığında, MEEM yöntemi uzman seçme algoritması sayesinde kapanma problemi karşısında güçlü bir başarı yakalamıştır. Uzman seçim algoritması temel olarak MEEM yönteminin içerdiği DVM sınıflandırıcının güncellenmesi için nesne olarak belirlenen örneklerin (uzman) yarı denetimli olarak ele alması ve güncellenmek için kabul edilen örneklerin kendi içerisindeki bulunduğu örnekler ile kıyaslamasıdır. Bu şekilde nesne olarak belirlenen örnekler hatalı olma durumunda güncellemeler engellemiştir.

Yapılan testlerde elde edilen en şaşırtıcı sonuçlardan bir tanesi de kapanma problemine karşı en hassas olan yöntemin GOTURN olmasıdır. Bir diğer şaşırtıcı sonuç ise kullandığı 3 katmanlı hiyerarşik özniteliğe rağmen, HCFT yöntemi ışık değişimi problemine karşı zayıf kalmış olmasıdır. Bu sonuçlara göre ADNet yönteminin ışık değişimi problemine karşı 0.7 oranında bir başarı elde ettiği görülmektedir.



Şekil 4. 4. Ele alınan yöntemlerin problem bazlı ve genel (G) OPE doğruluk sonuçları (eşik değer 0.5 için)

Bu bölümün devamında her problem ayrı ayrı ele alınarak yöntemlerin değerlendirilmesi yapılmıştır.

4.4.1. Kapanma Problemi

Kapanma (occlusion) problemi hedef nesnenin çerçeve içinde kısmen ya da tamamen görünümünün engellenmesi ile ortaya çıkan olumsuz durumu ifade etmektedir. Bu problem nesne takibindeki karşılaşılan en zor problemlerden bir tanesidir. Gelişmiş güçlü bir nesne takip yöntemi, nesne kapanma problemine uğradığında, yapmakta olduğu çevrimiçi güncellemeleri durdurması beklenmektedir. Kapanma problemi sırasında güncellemelerin durdurulmaması arka plandaki örnekler ile hatalı güncellemeler yapılmasına ve bunun sonucunda nesne görünümünün unutulmasına sebep olacaktır. Bununla birlikte, geliştirilen nesne takip yöntemi ileri ki çerçevelerde hedef nesne tekrar ortaya çıktığında, hedef nesneyi tespit edip nesne takibine devam etmesi gerekmektedir.

Ele alınan yöntemlerin kapanma problemine (KP) karşı elde ettikleri OPE başarı sonuçları Çizelge 4.4'te verilmiştir. Genel olarak sonuçlar incelendiğinde derin öznitelikler kullanan yöntemlerin nesne görünümündeki ince ayrıntıları da öğrenerek kapanma problemi karşısında başarı elde ettikleri görülmüştür. Geleneksel yöntemlerde (MUSter ve MEEM

yöntemleri gibi) ise bu problem karşısında başarı elde edebilmek için önceki çerçevelerde saklanan nesne örnekleri ile hatalı güncellemeler engellenmeye çalışılmıştır. Bu şekilde nesne tekrar ortaya çıktığında nesne tespit edilerek takibe devam edilmiştir.

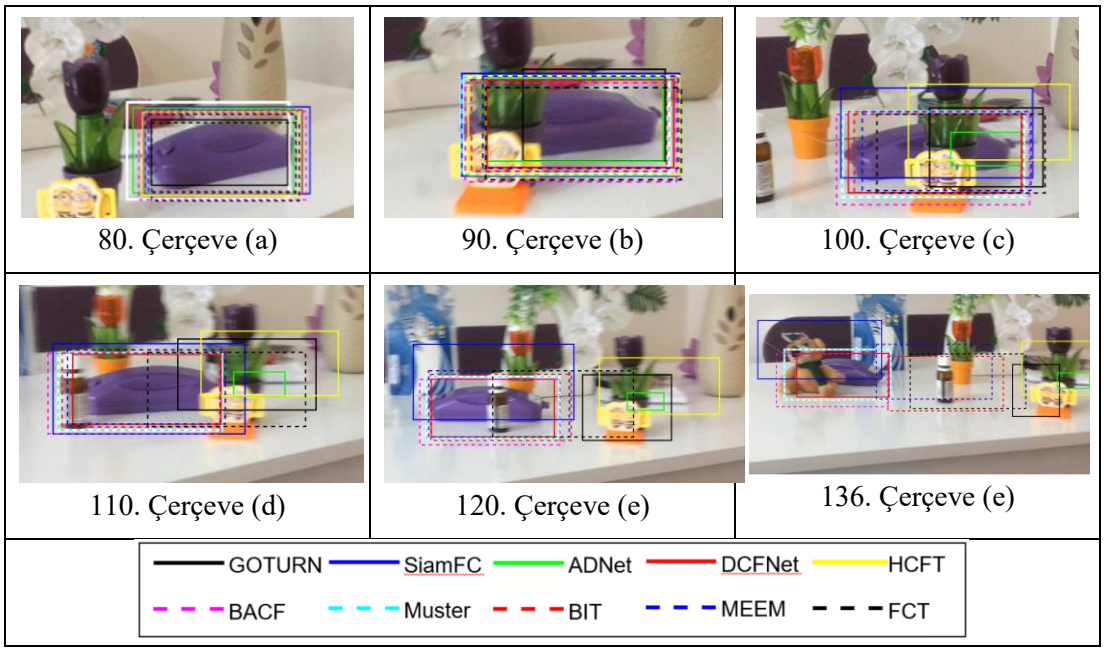
Çizelge 4. 4. Kapanma problemine karşı $eşikdeğerler\ \theta_b = 0.5$ ve $\theta_d = 20$ için OPE duyarlılık ve doğruluk sonuçları

	Kapanma Problemi	
Yöntem	**OPE Duyarlılık Sonuçları**	**OPE Doğruluk Sonuçları**
GOTURN	0,54	0.48
SiamFC	0.73	0.64
ADNet	**0.85**	0.**65**
DCFNet	0.73	0.63
HCFT	0.77	0.62
BACF	0.65	0.61
MUSTer	0.59	0.51
BIT	0.67	0.57
MEEM	0.59	0.57
FCT	0.48	0.26

Elde edilen OPE doğruluk sonuçlarına göre kapanma problemine karşı en başarılı yöntemler ADNet ve SiamFC olmuştur. Geleneksel yöntemlerden BIT, MEEM ve BACF yöntemleri derin öğrenme yöntemleri ile kıyaslanabilecek başarılar elde etmiştir. BACF yöntemi, korelasyon filtresinin eğitiminde arka plan örneklerini ağırlıklı kullanması nedeniyle nesne ile arka planı güçlü bir şekilde ayırabilmiştir. Bunun sonucunda geleneksel nesne takip yöntemleri arasında en iyi performansı elde etmiştir. Bu problem karşısında bir diğer başarılı yöntem ise MEEM yöntemi olmuştur. MEEM yöntemi geliştirdiği uzman seçim algoritması sayesinde kapanma problemi sırasında oluşan yanlış güncellemeleri engellemiştir. Bu şekilde ileriki çerçevelerde nesne tekrar ortaya çıktığında nesneyi doğru bir şekilde tespit etmiş ve takibe devam etmiştir. Geleneksel yöntemler içerisindeki MUSTer yöntemi içerdiği çoklu hafıza modeli ile nesneyi tanımlayan ve nesne görüntülerinden oluşan anahtar nokta kümesi içermesine rağmen, tez çalışmaları sırasında kaydedilen ve kapanma problemi içeren videolarda beklenen başarıyı sağlayamamıştır. Bunun temel sebebi, içerdiği denetleyici zor kapanma problemleri karşısında zayıf kalması ve hatalı güncellemelere izin vermesidir.

GOTURN yöntemi nesne takibi sırasında herhangi bir güncelleme yapmadığı için kapanma problemi karşısında zayıf kalmıştır. Bunun temel sebebi, kapanma problemi oluştuğu sırada, nesneyi arama alanı içinde bulamadığı için arka planda nesneye en çok benzeyen alana (nesne ya da nesneye benzeyen) yönelmesidir. İleriki çerçevelerde nesne ortaya çıktığında eski nesne görünümüyle ilgili herhangi bir bilgi saklamadığından nesneyi bulamamıştır.

Test veri setindeki “morKutu” videosu ve yöntemlerin hareket biçimleri Şekil 4.5’de verilmiştir. Burada temel amaç mor renkte görünen kutu nesnesinin çerçeve boyunca takip edilmesidir. Bu senaryoda arka arkaya gelen yarı kapanma problemi birçok yöntemi oldukça zorlamıştır. Başarılı olarak nesneyi takip eden yöntemler sırasıyla MUSTer, BACF ve DCFNet şeklindedir. En iyi sonucu ise MUSTer yöntemi yakalamıştır. Bu sahnede SiamFC yöntemi nesneyi kısmen içine alan geniş bir sınırlama kutusu ile nesneyi takip etmiştir. Bunun temel sebebi; SiamFC yönteminin görüş açısında kapanma problemi oluştukça nesne ile aynı renkte olan arka plandaki mor alanı nesne olarak görmesidir. En şaşırtıcı sonuç testlerde yüksek performans gösteren ADNet yönteminde gözlemlenmiştir. Bu yöntem 90’nıncı çerçeveden sonra oyuncak yeşil çiçek görüntüsüne takılıp hedef nesne olan mor kutuyu takip etmeyi bırakmıştır.



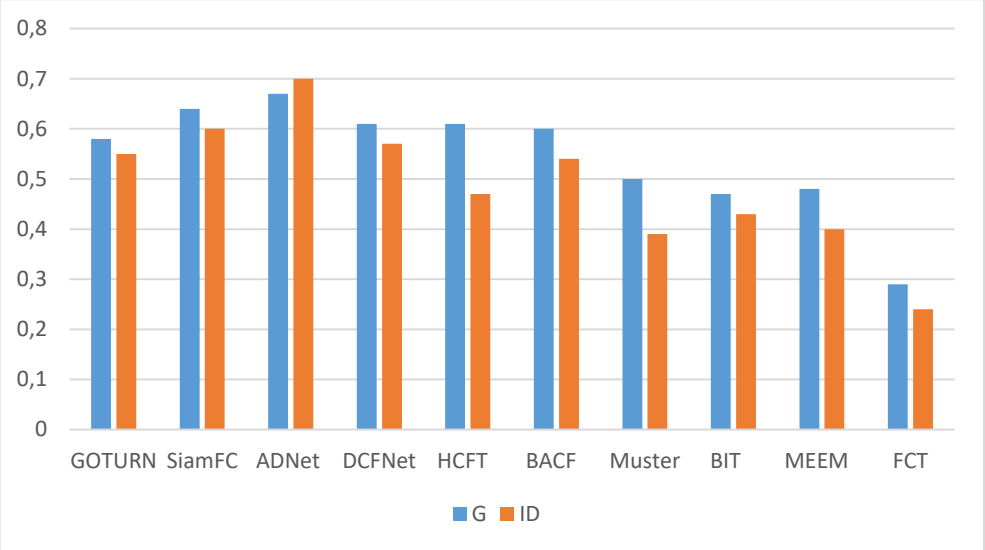
Şekil 4. 5. “morKutu” video verisinin nesne takip sonuçları

4.4.2. Işık Değişimi

Işık değişim problemi video içerisindeki nesnenin bulunduğu ortam içeresindeki ışık miktarının değişmesi olarak tanımlanabilir. Bu değişim özellikle nesnenin renk bilgisini ciddi şekilde etkileyebilmektedir. Günümüzdeki geliştirilen nesne takip yöntemleri genellikle ışık değişim problemine karşı güçlü başarılar elde etmektedir.

Ele alınan yöntemlerin ışık değişim problemine (ID) karşı elde edilen OPE doğruluk sonuçları Şekil 4.6’da bir arada verilmiştir. Yapılan test çalışmalarında ışık değişimine karşı ADNet ve SiamFC yöntemleri kullandıkları derin öznitelikler ve gelişmiş ağ mimarileri sayesinde ışık değişim problemine karşı en güçlü yanıt verebilen yöntemler olmuştur (Şekil

4.6). MEEM yöntemi yapısındaki kısıtlı güncellemeler (uzman seçim algoritması) yüzünden ışık değişimi problemine karşı zayıf kalmıştır. Normal şartlarda renk özniteliği ışık değişim probleminden en çok etkilenen özniteliktir. BIT yöntemi renk öznitelikleri kullanmasına rağmen C1 biriminde özniteliklerin 4×4'lük bir filtre ile ortalamasının alınması sayesinde bu problem karşısında performans kaybına uğramamıştır.



Şekil 4. 6. Genel (G) ve ışık değişim (ID) problemi için OPE doğruluk sonuçları (eşik değer $\theta_b = 0.5$ için)

4.4.3. Ani Hareket Değişimi

Nesnenin yaptığı hareketin düzgünlüğü (akışkan olması) nesne takip performansını doğrudan etkileyebilmektedir. Özellikle hareket modeli ile geliştirilen yöntemler ani hareket değişimine karşı hassas olabilmektedirler. Aynı şekilde belirli bir arama alanı içinde nesneyi arayan yöntemler ani hız artışı (iki çerçeve arasında ani konum değişiklikleri) olma durumunda arama alanın bu atlayışa karşı ayarlanamaması nesnenin kaybolmasına sebep olmaktadır. Bunun yanında ani hız değişimi nesnede bulanıklaşmaya ve dolayısıyla görünüm değişikliğine sebep olabilmektedir.

Çizelge 4.5'te ele alınan yöntemlerin OPE duyarlılık ve doğruluk sonuçları verilmiştir. Hiyerarşik bir yapıda mekânsal ve ince ayrıntılı öznitelikleri kullanan HCFT yöntemi ani hareket değişimi problemi karşısında oldukça başarılı sonuçlar elde etmiştir. Şekil 4.4'te ki genel sonuçlar ile kıyaslandığında bu problem karşısında dayanıklı kalabilen bir diğer yöntem ise BIT yöntemidir. Bunun temel sebebi; S1 biriminde alınan başta doku ve renk adları olmak üzere toplam 60 özniteliğin S2 biriminde korelasyon filtresini güçlü bir şekilde sınıflandırıcı için kullanması olarak izah edilebilir. Yapılan testlerde hareket değişiminden en olumsuz etkilenen yöntemlerin SiamFC ve BACF yöntemleri olduğu görülmüştür.

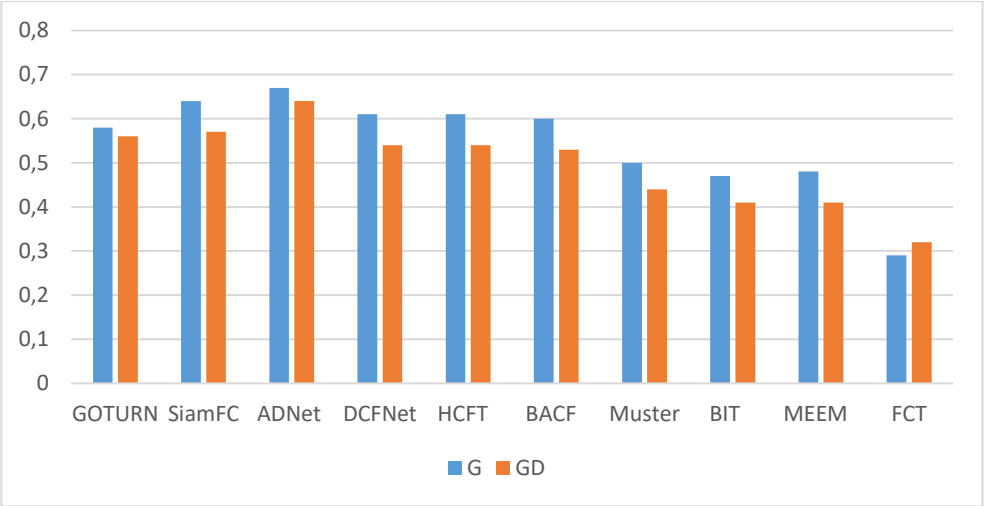
Çizelge 4. 5. Ani hareket değişimi problemine karşı elde edilen OPE duyarlılık ve doğruluk sonuçları.

	Ani Hareket Değişimi	
Yöntem	**OPE Duyarlılık Sonuçları**	**OPE Doğruluk Sonuçları**
GOTURN	0.48	0.45
SiamFC	0.61	0.54
ADNet	**0.70**	**0.66**
DCFNet	0.61	0.56
HCFT	0.67	0.61
BACF	0.52	0.53
MUSTer	0.48	0.39
BIT	0.47	0.44
MEEM	0.46	0.42
FCT	0.34	0.25

4.4.4. Görünüm Değişikliği

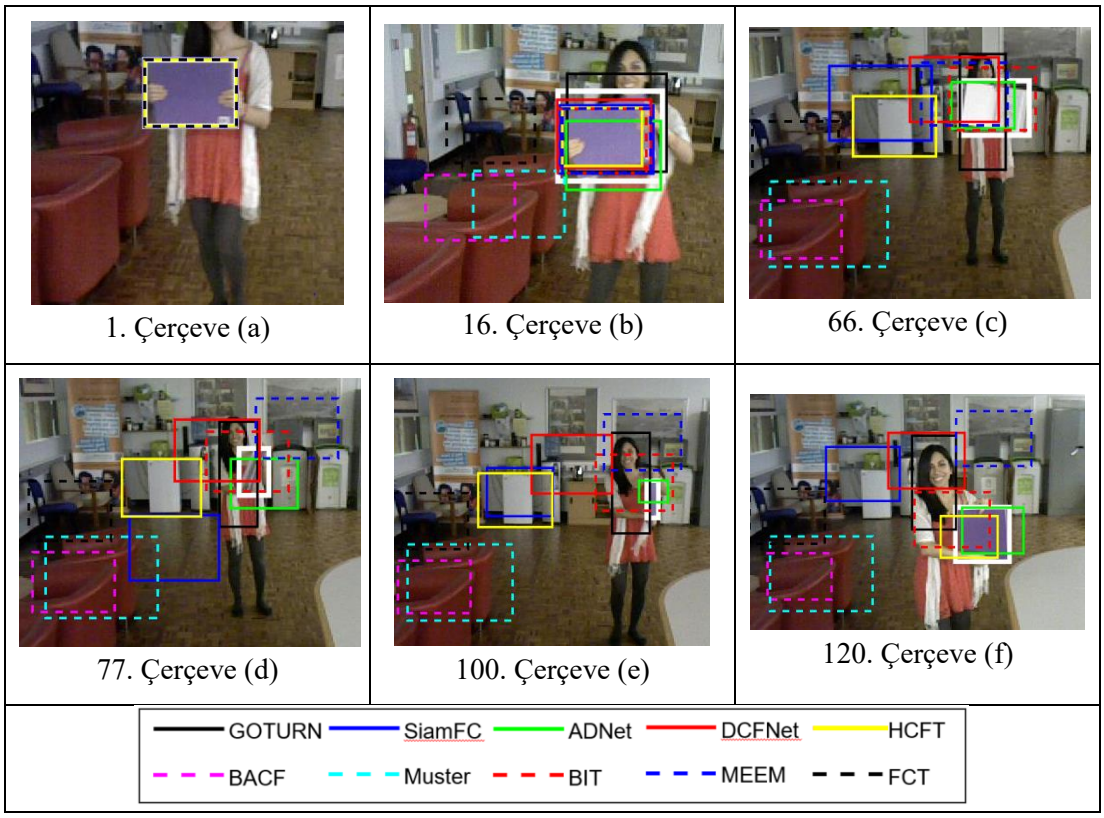
Görünüm değişikliği nesne takibinde en çok karşılaşılan problemlerden bir tanesidir. Nesnenin dönmesi, şekil değiştirmesi (orantısız olarak genişlemesi) ya da nesnenin yüksek hızlarda hareket etmesi doğrudan nesne görünümü etkileyebilmektedir. Literatürde bulunan geleneksel birçok yöntem bu problem ile başa çıkabilmek için doku veya kenar öznitelikleri kullanmıştır [12], [21]. Bunun yanında güncel derin öğrenme tabanlı yöntemler derin öznitelikler veya hareket modeli öğrenerek bu problem karşısında başarılar elde etmeye çalışmışlardır [11]. Literatürdeki birçok yöntem görünüm değişikliklerine adaptasyon için genellikle çevrimiçi güncellemeler yapmaktadır [11], [26].

Ele alınan yöntemlerin görünüm değişikliği problemine karşı elde edilen OPE doğruluk sonuçları Şekil 4.7'de verilmiştir. ADNet yöntemi hareket modeli ve yapısındaki güncellemeler sayesinde görünüm değişimi problemine karşı en iyi sonuçları elde etmiştir. Korelasyon tabanlı SiamFC, DCFnet ve BACF yöntemleri genel olarak başarılı olmakla birlikte bu problem karşısında zayıf kalmışlardır. Bunun yanında MUSTer yöntemi kısa hafıza biriminde bulunan ICF korelasyon filtresini, uzun hafıza birimini kullanarak güncellemesi sayesinde görünüm değişikliğinden en az etkilenen yöntemlerin arasına girmiştir. FCT yöntemi genel nesne takip başarısına göre bu problem karşısında daha başarılı sonuçlar elde etmiştir. Bunun temel sebebi yaptığı çevrimiçi güncellemeler sayesinde DVM sınıflandırıcısını nesne görünümüne karşı her zaman güncel tutmasıdır.



Şekil 4. 7. Genel (G) ve görünüm değişimi (GD) problemi için yapılan testleri OPE doğruluk sonuçları (eşik değer $\theta_b = 0.5$ için)

Şekil 4.8'de gürünüm değişimi problem içeren "book" video senaryosu ve yöntemlerin nesne takip sonuçları verilmiştir. Burada ADNet yöntemi hariç tüm yöntemler videodaki 66'ncı çerçevede bulunan kitabın açılması ile birlikte hedef nesneyi kaybetmiş ve çoğu yöntem nesne görünümüne en çok benzeyen arka plandaki farklı nesnelere odaklanmıştır. Fakat ADNet yöntemi hareket modeli öğrenmesi sayesinde nesne boyutunu tam olarak ayarlayamasa da video boyunca nesneyi takip edebilmiştir. ADNet yönteminin bu başarısındaki önemli etkenlerden bir tanesi ağ mimarisinin fc7 katmanındaki güven katmanıdır. Bu katman hedef nesneyi kaybetse bile nesneyi mevcut çerçevede devamlı arayan bir denetleyici görevini yürütmektedir. Bu şekilde nesne görünümü eski halini aldığında tekrar nesneye odaklanarak takibe devam etmiştir. "book" video senaryosunda 16'ncı çerçeveye kadar BACF, MUSTer ve FCT yöntemi arka plandaki "koltuk" görüntüsüne odaklanmıştır. 66'ncı çerçeve geldiği sırada kitabın açılması ile birlikte SiamFC ve HCFT yöntemi arka plandaki geniş bir alana ve GOTURN, BIT ve MEEM yöntemleri ise videodaki kişiye odaklanmıştır. Daha sonra gelen 120'nci çerçevede kitabın eski haline gelmesi ile birlikte HCFT yöntemi hedef nesne konumunu tekrar bulmuş ve takibe devam etmiştir. HCFT yöntemi nesneyi tekrar bulması sayesinde "book" video senaryosunda ADNet yönteminden sonra en iyi sonucu veren yöntem olmuştur.



Şekil 4. 8. Yöntemlerin Vot2015 veri kümesindeki zorlu "book" video üzerindeki test sonuçları

4.4.5. Nesne Boyut Değişikliği

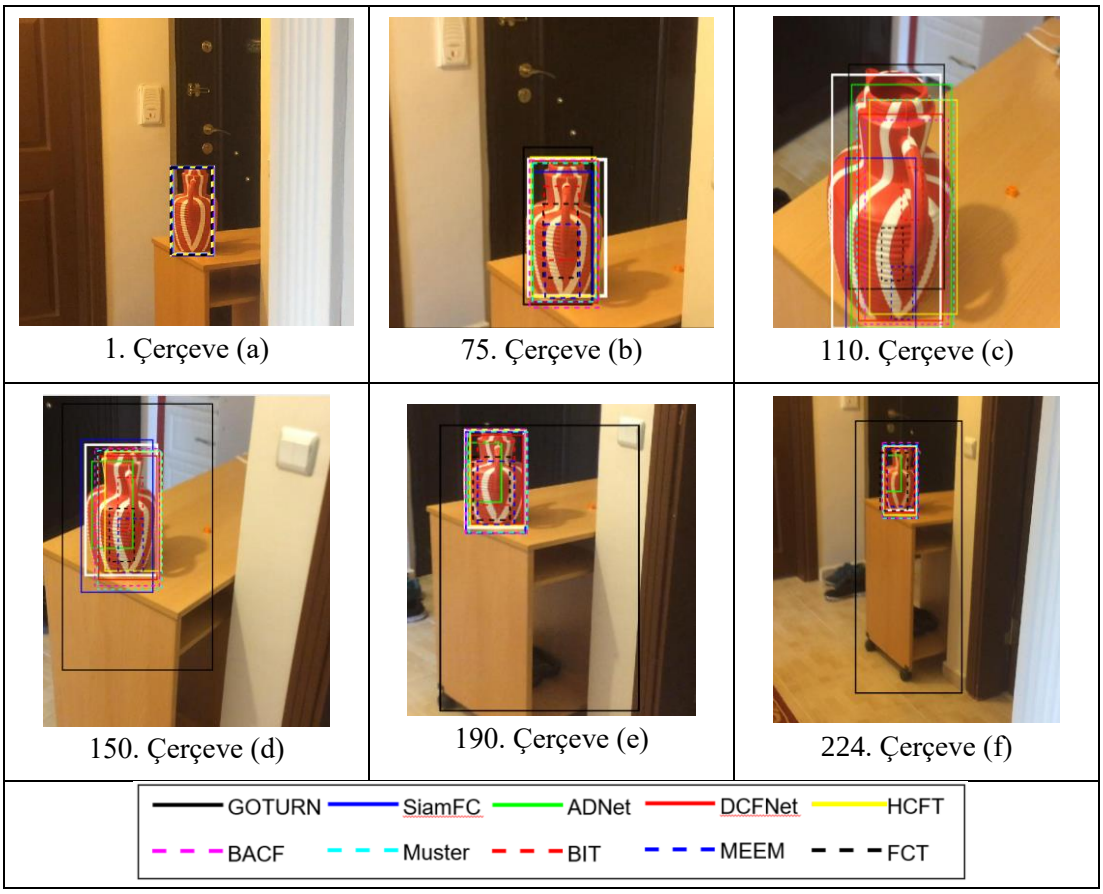
Nesne takibinde nesnenin kameraya yaklaşması veya nesnenin yapısal olarak boyutunun değişmesi durumunda yöntemlerin bu değişime ayak uydurması gerekmektedir. Yani nesne boyutu arttığında nesneyi gösteren sınırlama kutusunun da nesne ile aynı oranda büyümesi gerekmektedir.

Boyut değişim problemi karşısında ele alınan yöntemlerin elde ettiği sonuçlar Çizelge 4.6'da verilmiştir. Buradaki OPE doğruluk sonuçlarına göre sırasıyla ADNet, SiamFC ve GOTURN yöntemi en iyi sonuçları elde etmiştir. ADNet yöntemi en iyi performansı göstermesine rağmen diğer problemler karşısındaki başarısı ile kıyaslandığında boyut değişimine karşı daha düşük performans sergilemiştir. Geleneksel nesne takip yöntemi olan BACF yöntemi HCFT ile aynı sonuçları elde ederek başarı sıralamasında dördüncü sırada yer almıştır. Korelasyon tabanlı DCFNet yöntemi kullandığı derin özniteliklere rağmen BACF yönteminden daha düşük sonuçlar elde etmiştir. Buradaki en düşük sonuçları MEEM ve FCT yöntemi elde etmiştir. Özellikle MEEM, BIT ve FCT yöntemleri hedef nesnenin boyut değişim senaryosuna karşı hiçbir öngörüye sahip değildir. Bunun sonucunda bu yöntemler boyut değişikliği olan senaryolarda başarısız kalmışlardır.

Çizelge 4. 6. Boyut değişim problemine karşı elde edilen OPE duyarlılık ve doğruluk sonuçları.

	Boyut Değişikliği	
Yöntem	**OPE Duyarlılık Sonuçları**	**OPE Doğruluk Sonuçları**
GOTURN	0.52	0.52
SiamFC	0.62	0.54
ADNet	**0.75**	**0.55**
DCFNet	0.66	0.48
HCFT	0.71	0.49
BACF	0.56	0.49
MUSTer	0.61	0.47
BIT	0.58	0.36
MEEM	0.53	0.34
FCT	0.54	0.3

Şekil 4.9'da "surahi" video senaryosu ve yöntemlerin farklı çerçevelerdeki takip sonuçları verilmiştir. Bu video senaryosunda kamera yaklaşma hareketi ile yaklaşık olarak hedef nesne boyotu 4-5 kat büyütülmüştür. Daha sonra kameranın uzaklaşması ile hedef nesne küçülerek eski halini almıştır. Bu senaryoda nesne boyutu arttıkça yöntemlerin bulduğu sınırlama kutusunun boyutunun da artması beklenmektedir. ADNet, MEEM, FCT ve BIT yöntemlerinde nesne boyutu belli bir oranda arttıktan sonra nesnenin bir kısmına odaklanmış ve geri kalan çerçevelerde bu alanı takip etmeye devam etmişlerdir. GOTURN yöntemi videonun 110'uncı çerçevesine kadar hedef nesne olan sürahiyi başarılı bir şekilde tespit ve takip edebilmiştir. Fakat nesne boyutu küçülmeye başladığında bu değişime ayak uyduramamış ve nesneyi de içeren büyük bir alanı takip etmeye başlamıştır. Bu senaryo için DCFNet, HCFT ve MUSTer yöntemleri başarılı bir şekilde nesneyi takip edebilmiştir.



Şekil 4. 9. "surahi" videosunun test sonuçları

4.4.6. Arka Plan Dinamikliği

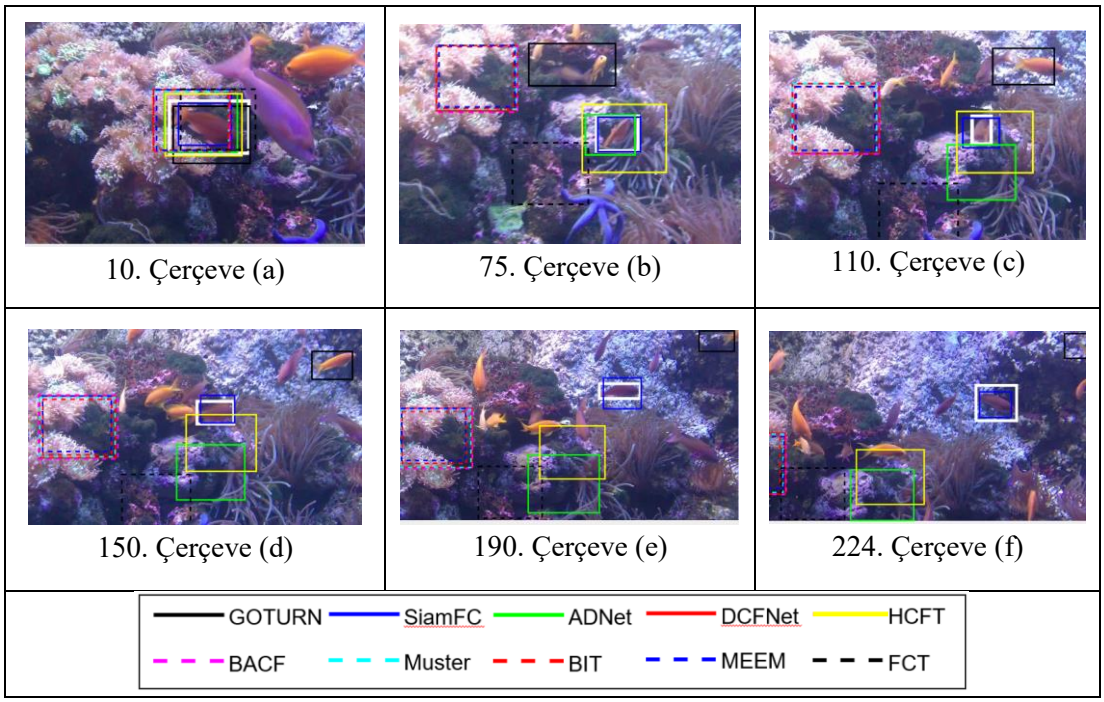
Arka plan dinamikliği diğer bir adıyla sahne dinamikliği arka plandaki nesnelerin değişken, hareketli ya da hedef nesne ile benzer olma durumu şeklinde tanımlanmaktadır. Bunun yanında nesnenin takip edileceği video sahnesinin kameraya alınması sırasında kameranın hareket etmesi de sahne dinamikliği problemi tanımı içerisine girebilmektedir. Bu durumlar nesne ile arka planının güçlü bir şekilde ayırt edilmesini engellediği için nesne takibini oldukça zorlaştırabilmektedir.

Ele alınan yöntemlerin arka plan dinamikliği problemi için elde etiği sonuçlar Çizelge 4.7'de verilmiştir. Bu problem için kullanılan özniteliklerin güçlü olması başarıyı doğrudan etkilemektedir. Derin öznitelik kullanan yöntemler geleneksel yöntemlere göre kullandıkları öznitelikler sayesinde daha yüksek başarılar sağlamıştır. Bu problem karşısında en yüksek sonuçları veren yöntemler arasında ADNet yöntemi ilk sırada yer almaktadır. Hemen arkasından HCFT yöntemi başarı sıralamasında ikinci sırada yer almıştır. BACF yöntemi arka plan ile doğrudan etkileşimli bir yapıya sahip olduğundan bu problem karşısında etkinlik gösterememiştir. SiamFC ve DCFNet yöntemleri de bu tür problemler içeren video setlerinde istenen düzeyde başarı elde edememiştir.

Çizelge 4. 7. Arka plan dinamikliği problemine karşı elde edilen OPE duyarlılık ve doğruluk sonuçları

	Arka plan dinamikliği	
	OPE Duyarlılık Sonuçları	**OPE Doğruluk Sonuçları**
GOTURN	0.50	0.51
SiamFC	0.59	0.51
ADNet	**0.68**	**0.62**
DCFNet	0.57	0.47
HCFT	0.62	0.52
BACF	0.49	0.45
MUSTer	0.51	0.41
BIT	0.52	0.42
MEEM	0.52	0.42
FCT	0.41	0.19

Şekil 4.10'da test veri setindeki "fish2" video senaryosu ve yöntemlerin bazı çerçevelerdeki davranış biçimleri görsel olarak verilmiştir. Akvaryumdaki benzer balıkların görüntüsü ve arka plandaki aşırı nesne değişkenliği birçok yöntemi oldukça zorlamıştır. Sonuçlara bakıldığında SiamFC yöntemi video boyunca nesneyi kaybetmeden oldukça başarılı bir şekilde nesneyi takip edebildiği görülmüştür. ADNet ve HCFT yöntemi ise 75'inci çerçeveden sonra balık dönme hareketi yaptıktan sonra görünümünde oluşan değişim nedeniyle iki yöntemde nesneyi kaybederek arka plandaki taş görüntüsüne kaymıştır. DCFnet, MUSTer, FCT ve MEEM yöntemleri ise daha ilk çerçevelerde arka plandaki aşırı değişiklik yüzünden nesneyi tam tespit edemeden arka plandaki beyaz bir bitkiye kaymıştır. GOTURN yöntemi ise takip edilen nesne ile aşırı benzeyen başka bir balığa odaklanıp video boyunca o balığı takip etmiştir.



Şekil 4. 10. Yöntemlerin Vot2015 veri kümesindeki zorlu "fish2" video üzerindeki test sonuçları

4.5. Hız Performansı

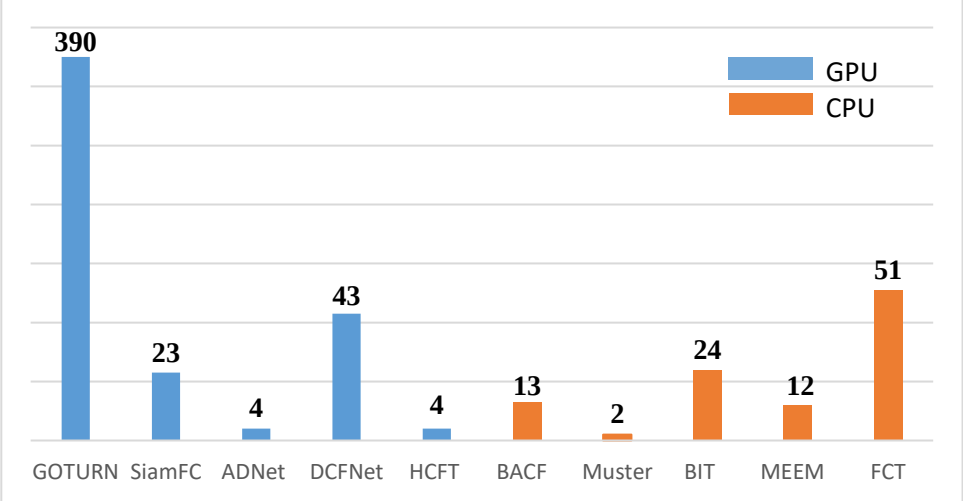
Güçlü bir nesne takip yönteminin gerçek zamanlı çalışma performansının en temel ölçütlerinden birisi, yöntemin çalışma hızının 24 FPS (Saniyedeki Kare Hızı-Frame Per Second) hızına eşit veya büyük olmasıdır [75]. Nesne takibinde geliştirilen yöntemin karmaşıklığı, işlem yoğunluğu ve güncelleme sıklığı hız performansını doğrudan etkilemektedir. Özellikle nesne takip yöntemleri derin ağ mimarisi kullanılarak özniteliklerin çıkartılması, özniteliklerin farklı makine öğrenmesi ve örüntü tanıma yöntemleri ile çıkartılmasına kıyasla çok daha fazla zamana ihtiyaç duymaktadır [11]. Bunun yanında derin özniteliklerin çıkartıldığı katmanların güncellenmesi oldukça zaman alan süreçler olarak öne çıkmaktadır [15].

Günümüzdeki birçok nesne takip yöntemi yüksek hızlara ulaşmak için farklı teknikler kullanmaktadır. En yaygın kullanılan teknik işlem zamanını düşürmek için işlemlerin Fourier uzayında gerçekleştirilmesidir [75]. Özellikle korelasyon tabanlı yöntemlerde eleman bazlı çarpma işlemi için bu tekniği kullanarak işlem maliyeti büyük oranda düşürülebilmiştir [18]. Hız performansını arttırmak için bir diğer teknik ise güçlü işlemcilere sahip ekran kartlarındaki grafik işlemcilerin kullanılmasıdır. Derin öğrenme gibi yöntemlerin yapısındaki çok katmanlı ağ yapısı ve bu katmanlarda gerçekleştirilen işlemlerin normal işlemcilerde (CPU) yapılması çok fazla zaman alabilmektedir. Bunun yanında 40 Gigabayt'ı aşan veriler ile eğitilen ağ mimarileri için (GOTURN, VGG-Net gibi) çoğu zaman işlemcilerin yanındaki hafıza alanları

da yetersiz kalmaktadır. Bu sebeplerden dolayı derin öğrenme yöntemleri yüksek kapasiteye sahip gelişmiş ekran kartlarının işlemcileri (GPU) üzerinde çalıştırılmaktadır [11].

Ele alınan yöntemlerin hız performanslarına ait sonuçlar Şekil 4.11'de verilmiştir. Yöntemlerin hız performanslarını hesaplamak için her bir yöntemin test kümesindeki toplam çerçeve sayısı toplam test süresine bölünmüştür. Bu şekilde yöntemlerin ortalama fps cinsinden hız değerleri elde edilmiştir. Elde edilen sonuçlar incelediğinde en yüksek hız performansına 395 fps hızı ile GOTURN yöntemi ulaşarak ilk sırada yer almıştır. GOTURN yöntemi geliştirdiği ağ mimarisindeki işlemlerin basitliği ve ağ katmanlarında herhangi bir güncelleme yapmaması bu başarının temel sebebi olmuştur. FCT [53] yöntemi geliştirdiği algoritma sayesinde yüksek hızlarda öznitelik çıkarıp sınıflandırıcı işlem maliyetini oldukça düşürebilmiştir. FCT yöntemi CPU üzerinde çalışmasına rağmen ortalama 51 fps hızına ulaşarak hız performansı kıyaslamasında ikinci sırada yer almıştır.

Derin öznitelik kullanan korelasyon filtre tabanlı SiamFC ve DCFNet yöntemleri sırasıyla 23 ve 43 fps hızına ulaşmıştır. BIT yöntemi nesne takibinde işlem maliyetlerini düşürmek için S1 biriminde hızlı gabor yaklaşımı ve S2/C2 birimlerinde ise hızlı fourier dönüşümü yöntemleri kullanılmıştır [75]. Bu sayede nesne takibinde 24 fps hızına ulaşmıştır. Hız açısından en geride kalan yöntem yoğun işlemler ve bu işlem maliyetlerinin yüksek olması yüzünden MEEM yöntemi olmuştur. ADNet yönteminin nesne takip başarısı oldukça iyi olmasına rağmen çevrimiçi güncellemeler yüzünden hız performansı açısından istenen düzeyde sonuçlar verememiştir.



Şekil 4. 11. Ele alınan yöntemlerin ortalama saniyedeki işlenen kare sayısı (fps)

5. SONUÇLAR VE ÖNERİLER

Bu tez çalışmasında nesne takibinde başarılı olan 5 geleneksel nesne takip yöntemi ve 5 güncel derin öğrenme tabanlı nesne takip yöntemi ele alınarak yöntemler hakkında detaylı testler ve değerlendirmeler yapılmıştır. Bu değerlendirmeler sonucunda nesne takip alanında çalışacak araştırmacılar için önemli sonuçlara varılmıştır. Yöntemlerin test edilmesinde hem literatürde bulunan veri tabanları hem de tez çalışmaları sırasında oluşturulan video veri tabanı kullanılmıştır.

Yapılan deneysel çalışmalar sonucunda derin öğrenme tabanlı yaklaşımların nesne takibinde oldukça yüksek başarılar elde ettikleri tespit edilmiştir. Bunun yanında geleneksel makine öğrenmesi yöntemleri kullanan korelasyon filtre tabanlı yöntemler (BACF gibi), derin öğrenme tabanlı yöntemler ile benzer yüksek başarılar elde etmiştir. Yöntemlerin genel performansları incelendiğinde, DCFNet ve SiamFC gibi nesne takip yöntemleri hem korelasyon filtresi hem de derin öznitelikleri birleştirdikleri için gerek hız gerekse nesne tespit başarısı açısından oldukça yüksek sonuçlar elde etmişlerdir. En başarılı sonuçları elde eden yöntem ise gelişmiş ağ mimarisi ile hareket tahmini yapan ADNet yöntemi olmuştur. Testlerde düşük başarılar elde eden yöntemler ise DVM ve Naive Bayes sınıflandırıcılarını kullanan MEEM ve FCT yöntemleri olmuştur. Her ne kadar düşük performans göstermiş olsa da MEEM yönteminin hatalı güncellemelerini engelleyen uzman seçim algoritması ve FCT yönteminin yüksek hızlarda çalışmasını sağlayan öznitelik sıkıştırma tekniği sayesinde her iki yöntem literatürde önemli çalışmalar arasında yer almıştır. Deneysel çalışmalar sırasında göze çarpan bir diğer önemli yöntem ise korelasyon filtresi yönteminin yüksek potansiyelini ortaya koyan BACF yöntemi olmuştur.

Yapılan testler ve değerlendirme sonuçlarına bakıldığında, bir yöntemin nesne takip problemindeki başarısını etkileyen 3 temel unsur olduğu belirlenmiştir. Bunlar sırasıyla kullanılan öznitelikler, yöntemin özniteliklerini değerlendirip karar veren sınıflandırıcı (takipçi) ve bu sınıflandırıcının güncellenmesidir.

DCFNet, BACF, HCFT yöntemlerinin test sonuçlarına bakıldığında, aynı takipçiyi kullanmalarına rağmen kullandıkları özniteliklerin farklı olmasından dolayı aralarında başarı oranları birbirinden farklıdır. Bu yöntemler, ortak takipçi olarak korelasyon filtresi kullanmışlardır. Fakat DCFNet yönteminin kullandığı derin özniteliklerin nesne ve arka planı ayırtedebilme gücü daha yüksek olduğundan oldukça yüksek başarılar elde etmiştir. Bunun yanında, çok fazla özniteliğin kullanılması aynı zamanda dezavantaja da sebep olabilmektedir. HCFT yöntemi, daha güçlü başarı elde etmek için VGG-Net ağ mimarisinin 3 farklı katmanında 3 farklı öznitelik grubu kullanmıştır. Buna karşı olarak DCFNet yöntemi ise VGG-Net mimarisinin tek katmanını kullanmasına rağmen daha yüksek başarılar elde etmiştir.

Ayrıca HCFT yönteminde, çok fazla öznitelik kullanılmasından dolayı yöntem çalışma hızı açısından zayıf kalmaktadır. Bu yöntemler arasında en düşük performansı elde eden ve 31 boyutlu YGH özniteliğini kullanan BACF yönteminde ise, yüksek başarı elde edebilmek için arka plan örnek sayısının arttırılmış olduğu göze çarpmaktadır. Bununla birlikte yöntemin kullandığı öznitelikler yetersiz kalmış ve takip sırasında yöntem başarısı düşmüştür. Bu değerlendirme sonucunda, nesne takibinde kullanılması gereken özniteliklerin hedef nesne açısından ayırt edici bilgiler içermesi gerektiği sonucuna ulaşılmıştır. Ayrıca 31 boyutlu özellik vektörü boyutu hız açısından avantaj sağlamakla birlikte ayırt edici ve kapsayıcı bir öznitelik bilgisi sağlayamamıştır.

Yöntemlerin nesne takip başarılarını etkilen ikinci temel unsur kullanılan sınıflandırıcının yapısı, niteliği ve çalışma mantığıdır. Bazı yöntemlerde sınıflandırıcı yerine nesnenin gerçekten içinde bulunduğu referans çerçeve ile mevcut görüntüde tespit edilen çerçeve arasındaki benzerliği hesaplayan bir ağ mimarisi kullanılmıştır. Burada önemli olan nesne takibinde karşılaşılabilecek görünüm değişikliklerine karşı sınıflandırıcının başa çıkabilmesidir. Özellikle günümüzde dinamik ortamlardaki nesne takibinin yapılması hedeflendiğinde, sınıflandırıcı olarak makine öğrenmesindeki DVM veya Naive Bayes gibi ikili sınıflandırıcılar nesne takibinde zayıf kalmaktadırlar. Buna karşı korelasyon filtresi tabanlı yöntemler, nesne takibinde nesneye en çok benzeyen konumları belirleyerek oldukça yüksek performans göstermiştir. Bu nedenle yakın geçmişte geliştirilen bir çok yöntem korelasyon filtresi tabanlı olarak geliştirilmiştir [11, 18]. Ele alınan yöntemler arasında ADNet yöntemi, sınıflandırıcı olarak kullandığı hareket modelini öğrenen ağ mimarisi sayesinde en başarılı sonuçları elde etmiştir. Elde edilen test sonuçlarına bakıldığında ADNet ağ mimarisi için kullanılan yaklaşımın yakın gelecekte oldukça popüler olacağı öngörülmektedir.

Nesne takip yöntemlerin başarısını etkileyen üçüncü önemli unsur sınıflandırıcının güncellenme biçimidir. Nesne takibinde ortamın değişmesiyle nesne görünümü ve arka plan sıklıkla değişebilmektedir. Bu değişikliklere sınıflandırıcının hızlı bir şekilde adapte olması gerekmektedir. Genellikle güncelleme işlemleri nesne takibinin sonucunda nesne olarak belirlenen konumdan alınan örnekler kullanılarak yapılmaktadır. Burada önemli olan alınan örneklerin gerçekten nesne olup olmadığıdır. MEEM yöntemi nesne olarak alınan örnekleri denetleyerek hatalı güncellemeleri engellemiştir. Bu sayede özellikle kapanma problemine karşı oldukça başarılı olmuştur. Bir diğer başarılı sonucu elde eden ADNet yöntemi ise ağ mimarisinin son katmanına bir güven katmanı eklemiştir. Bu katman sayesinde nesne olarak belirlenen alan denetlenerek hatalı sonuçlar engellenmiştir. Bu şekilde hatalı sonuçlardan kaçınılmış ve ağ mimarisinin güncellenmesi için daha başarılı örnekler kullanılmıştır.

Yapılan deneysel çalışmalar ve değerlendirmeler sonucunda yeni geliştirilecek bir nesne takip yöntemi için birkaç öneriye ulaşılmıştır. İlk olarak yeni geliştirilecek bir nesne takip yönteminde, derin özniteliklerin kullanılması büyük bir avantaj sağlayacaktır. Bu sayede

nesne ve arka planın görüntülerin arasındaki ince detaylar ortaya çıkartılacak ve birbirlerinden ayrılması kolaylaştırılacaktır. Kullanılabilecek sınıflandırıcının korelasyon filtre tabanlı olması ve bunun sonucunda elde edilen ısı haritası sayesinde güçlü bir ayırt edicilik ve yüksek hız performansı yakalanacaktır. Bunun yanında bir diğer güçlü takipçi tasarımı olarak ADNet yönteminin hareket öğrenen ağ mimarisi göze çarpmaktadır. Bu ağ mimarisi daha düşük hız performansı elde etmekle birlikte oldukça yüksek başarılar elde etmiştir. Son olarak, geliştirilecek nesne takip yönteminin güncelleme işleminde mutlaka bir denetleyici kullanması gerekmektedir. Bu denetleyici ağ mimarisi için bir güven katmanı ya da derin öğrenme kullanmayan geleneksel yöntemler için sonuçları denetleyen bir eşik değer olabilir. Bu denetleyici sayesinde sınıflandırıcının daha doğru örnekler ile daha güncel bir takip yapması sağlanacaktır. Bu şekilde geliştirilen nesne takip yöntemi, nesne takibinde karşılaşılabilecek birçok problem karşısında ve uzun süreli senaryolarda nesne takibi için istenilen sonuçları sağlayabilecektir.

6. KAYNAKLAR

[1] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, *Image Quality Assessment: From Error Visibility to Structural Similarity*, **IEEE Trans. Image Process.**, 13:4 (2004) 600–612.

[2] B. Badillo, D. A. Bowman, W. McConnel, T. Ni, and M. G. da Silva, *Literature Survey on Interaction Techniques for Large Displays*, (2006).

[3] A. Yilmaz, O. Javed, and M. Shah, *Object Tracking: A Survey*, **ACM Comput. Surv.**, 38:13 (2006).

[4] L. Fan *et al.*, *A survey on multiple object tracking algorithm*, 2016 IEEE International Conference on Information and Automation (ICIA), (2016) pp.1855–1862.

[5] Tao Zhang, Zaiwen Liu, Xiaofeng Lian, and Xiaoyi Wang, *Study on moving-objects detection technique in video surveillance system*, 2010 Chinese Control and Decision Conference, (2010) pp.2375–2380.

[6] W. Hu, T. Tan, L. Wang, and S. Maybank, *A Survey on Visual Surveillance of Object Motion and Behaviors*, **IEEE Trans. Syst. Man Cybern. Part C (Applications Rev.**, 34:3 (2004) 334–352.

[7] S. R. Balaji and S. Karthikeyan, *A survey on moving object tracking using image processing*, 2017 11th International Conference on Intelligent Systems and Control (ISCO), (2017) pp.469–474.

[8] F. Mancini, M. Dubbini, M. Gattelli, F. Stecchi, S. Fabbri, and G. Gabbianelli, *Using Unmanned Aerial Vehicles (UAV) for High-Resolution Reconstruction of Topography: The Structure from Motion Approach on Coastal Environments*, **Remote Sens.**, 5:12 (2013) 6880–6898.

[9] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. Van Den Hengel, *A survey of appearance models in visual object tracking*, **ACM Trans. Intell. Syst. Technol.**, 4:4 (2013) 1–48.

[10] M. Tiwari and R. Singhai, *A Review of Detection and Tracking of Object from Image and Video Sequences*, **Int. J. Comput. Intell. Res. ISSN**, 13:5 (2017) 973–1873.

[11] P. Li, D. Wang, L. Wang, and H. Lu, *Deep visual tracking: Review and experimental comparison*, **Pattern Recognit.**, 76 (2018) 323–338.

[12] A. W. Smeulders, D. M. Chu, R. Cucchiara, S. Calderara, A. Dehghan, and M. Shah, *Visual Tracking: An Experimental Survey*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 36:7 (2014) 1442–1468.

[13] D. Held, S. Thrun, and S. Savarese, *Learning to Track at 100 FPS with Deep Regression Networks*, **arXiv Prepr. arXiv1604.01802**, (2016).

[14] L. Wang, W. Ouyang, X. Wang, and H. Lu, *Visual Tracking with Fully Convolutional Networks*, 2015 IEEE International Conference on Computer Vision (ICCV), (2015) pp.3119–3127.

[15] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, *Fully-Convolutional Siamese Networks for Object Tracking*, **arXiv Prepr. arXiv1606.09549**, (2016).

[16] D. Zhang, H. Maei, X. Wang, and Y.-F. Wang, *Deep Reinforcement Learning for Visual Object Tracking in Videos*, **arXiv Prepr. arXiv1701.08936**, (2017).

[17] H. Fan and H. Ling, *SANet: Structure-Aware Network for Visual Tracking*, **arXiv Prepr. arXiv1611.06878v3**, (2016).

[18] Z. Chen, Z. Hong, and D. Tao, *An Experimental Survey on Correlation Filter-based Tracking*, **arXiv Prepr. arXiv1509.05520**, (2015).

[19] Z. Hong, Zhe Chen, C. Wang, X. Mei, D. Prokhorov, and D. Tao, *MUlti-Store Tracker (MUSTer): A cognitive psychology inspired approach to object tracking*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015) pp.749–758.

[20] W. Luo *et al.*, *Multiple Object Tracking: A Literature Review*, **arXiv Prepr. arXiv1409.7618**, (2014).

[21] H. S. Parekh, D. G. Thakore, and U. K. Jaliya, *A Survey on Object Detection and Tracking Methods*, **Int. J. Innov. Res. Comput. Commun. Eng. (An ISO Certif. Organ.**, 3297:2 (2007).
[22] B. Maraş, N. ARICA, and A. B. ERTÜZÜN, *Görsel Hedef Takibi Yöntemlerine Genel Bakış*, **Emo Bilim. Dergi**, 7:13 5–14.
[23] Y. Wu, J. Lim, and M.-H. Yang, *Object Tracking Benchmark*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 37:9 (2015) 1834–1848.
[24] P. Liang, E. Blasch, and H. Ling, *Encoding Color Information for Visual Tracking: Algorithms and Benchmark*, **IEEE Trans. Image Process.**, 24:12 (2015) 5630–5644.
[25] *The Visual Object Tracking VOT2015 Challenge Results*, 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), (2015) pp.564–586.
[26] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, *Understanding and Diagnosing Visual Tracking Systems*, **arXiv Prepr. arXiv1504.06055**, (2015).
[27] H. Yang, L. Shao, F. Zheng, L. Wang, and Z. Song, *Recent advances and trends in visual tracking: A review*, **Neurocomputing**, 74:18 (2011) 3823–3831.
[28] C. Bo, J. Zhang, J. Liu, and Q. Yao, *Robust online object tracking via the convex hull representation model*, **Neurocomputing**, (2018).
[29] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu, *Shape and Appearance Context Modeling*, 2007 IEEE 11th International Conference on Computer Vision, (2007) pp.1–8.
[30] V. V Nabiyev and A. Günay, *LBP Yardımıyla Goruntudeki Kisinin Yasının Bulunması*, **Cankaya Univ. J. Sci. Eng.**, 8:1 (2011) 27–41.
[31] M. Peker, H. Altun, and F. Karakaya, *HOG Temelli Bir Yöntem ile Ölçek ve Yönden Bağımsız Gerçek Zamanlı Nesne Tanıma*, Otomatik Kontrol Türk Milli Komitesi'nin Otomatik Kontrol Ulusal Toplantısı, (2012).
[32] K. Simonyan and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, **arXiv Prepr. arXiv1409.1556**, (2014).
[33] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, *Hierarchical Convolutional Features for Visual Tracking*, 2015 IEEE International Conference on Computer Vision (ICCV), (2015) pp.3074–3082.
[34] M. D. Zeiler and R. Fergus, *Visualizing and Understanding Convolutional Networks*, **arXiv Prepr. arXiv1801.10496**, (2013).
[35] S. Baker and I. Matthews, *Lucas-Kanade 20 Years On: A Unifying Framework*, **Int. J. Comput. Vis.**, 56:3 (2004) 221–255.
[36] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, *Incremental Learning for Robust Visual Tracking*, **Int. J. Comput. Vis.**, 77:1–3 (2008) 125–141.
[37] W.-L. Lu, K. Okuma, and J. J. Little, *Tracking and recognizing actions of multiple hockey players using the boosted particle filter*, **Image Vis. Comput.**, 27:1–2 (2009) 189–205.
[38] Hae Jong Seo and P. Milanfar, *Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 32:9 (2010) 1688–1704.
[39] Dong Wang, Huchuan Lu, and Chunjuan Bo, *Visual Tracking via Weighted Local Cosine Similarity*, **IEEE Trans. Cybern.**, 45:9 (2015) 1838–1850.
[40] V. Belagiannis, F. Schubert, N. Navab, and S. Ilic, Segmentation Based Particle Filtering for Real-Time 2D Object Tracking. Springer, Berlin, Heidelberg, pp. 842–855, 2012.
[41] R. Tao, E. Gavves, and A. W. M. Smeulders, *Siamese Instance Search for Tracking*, **arXiv Prepr. arXiv1605.05863**, (2016).
[42] K. Chen and W. Tao, *Once for All: a Two-flow Convolutional Neural Network for Visual Tracking*, **arXiv Prepr. arXiv1604.07507**, (2016).
[43] O. Russakovsky *et al.*, *ImageNet Large Scale Visual Recognition Challenge*, **Int. J. Comput. Vis.**, 115:3 (2015) 211–252.
[44] J. Valmadre, L. Bertinetto, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, *End-to-end representation learning for Correlation Filter based tracking*, **arXiv Prepr.**

arXiv1704.06036, (2017).
[45] Q. Wang, J. Gao, J. Xing, M. Zhang, and W. Hu, *DCFNet: Discriminant Correlation Filters Network for Visual Tracking*, **arXiv Prepr. arXiv1704.04057**, (2017).
[46] S. Avidan, *Support vector tracking*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 26:8 (2004) 1064–1072.
[47] H. Grabner and H. Bischof, *On-line Boosting and Vision*, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06), pp.260–267.
[48] H. Grabner, C. Leistner, and H. Bischof, Semi-supervised On-Line Boosting for Robust Tracking. Springer, Berlin, Heidelberg, pp. 234–247, 2008.
[49] B. Babenko, Ming-Hsuan Yang, and S. Belongie, *Robust Object Tracking with Online Multiple Instance Learning*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 33:8 (2011) 1619–1632.
[50] Z. Kalal, J. Matas, and K. Mikolajczyk, *P-N learning: Bootstrapping binary classifiers by structural constraints*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2010) pp.49–56.
[51] S. Hare, A. Saffari, and P. H. S. Torr, *Struck: Structured output tracking with kernels*, 2011 International Conference on Computer Vision, (2011) pp.263–270.
[52] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, *On-line Random Forests*, 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, (2009) pp.1393–1400.
[53] K. Zhang, L. Zhang, and M.-H. Yang, *Fast Compressive Tracking*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 36:10 (2014) 2002–2015.
[54] N. Jiang, W. Liu, and Y. Wu, *Learning adaptive metric for robust visual tracking*, **IEEE Trans. Image Process.**, 20:8 (2011) 2288–2300.
[55] B. Baben, Ming-Hsuan Yang, and S. Belongie, *Visual tracking with online Multiple Instance Learning*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, (2009) pp.983–990.
[56] S. Cao and W. Xue, *Robust visual tracking via adaptive forest*, 2013 Fourth International Conference on Intelligent Control and Information Processing (ICICIP), (2013) pp.30–34.
[57] D. S. Bolme, B. A. Draper, and J. R. Beveridge, *Average of Synthetic Exact Filters*, 2009 IEEE Conference on Computer Vision and Pattern Recognition, (2009) pp.2105–2112.
[58] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, Exploiting the Circulant Structure of Tracking-by-Detection with Kernels. Springer, Berlin, Heidelberg, pp. 702–715, 2012.
[59] D. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, *Visual object tracking using adaptive correlation filters*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2010) pp.2544–2550.
[60] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, *High-Speed Tracking with Kernelized Correlation Filters*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 37:3 (2015) 583–596.
[61] K. Zhang, L. Zhang, Q. Liu, D. Zhang, and M.-H. Yang, Fast Visual Tracking via Dense Spatio-temporal Context Learning. Springer, Cham, pp. 127–141, 2014.
[62] Y. Li and J. Zhu, A Scale Adaptive Kernel Correlation Filter Tracker with Feature Integration. Springer, Cham, pp. 254–265, 2015.
[63] T. Liu, G. Wang, and Q. Yang, *Real-time part-based visual tracking via adaptive correlation filters*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015) pp.4902–4912.
[64] C. Ma, X. Yang, Chongyang Zhang, and M.-H. Yang, *Long-term correlation tracking*, 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015) pp.5388–5396.
[65] M. Danelljan, G. Hager, F. S. Khan, and M. Felsberg, *Learning Spatially Regularized Correlation Filters for Visual Tracking*, 2015 IEEE International Conference on

Computer Vision (ICCV), (2015) pp.4310–4318.
[66] H. Kiani Galoogahi, A. Fagg, and S. Lucey, *Learning Background-Aware Correlation Filters for Visual Tracking*, **arXiv Prepr. arXiv1703.04590**, (2017) 1135–1143.
[67] Y. Qi *et al.*, *Hedged Deep Tracking*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016) pp.4303–4311.
[68] G. Zhu, F. Porikli, and H. Li, *Robust Visual Tracking with Deep Convolutional Neural Network Based Object Proposals on PETS*, 2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2016) pp.1265–1272.
[69] S. Ren, K. He, R. Girshick, and J. Sun, *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks*, **arXiv Prepr. arXiv1506.01497**, (2015).
[70] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, *ECO: Efficient Convolution Operators for Tracking*, **arXiv Prepr. arXiv1611.09224**, (2016).
[71] S. Yun, J. Choi, Y. Yoo, K. Yun, and J. Y. Choi, *Action-Decision Networks for Visual Tracking with Deep Reinforcement Learning*, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) pp.1349–1358.
[72] J. Kwon and K. M. Lee, *Tracking by Sampling Trackers*, 2011 International Conference on Computer Vision, (2011) pp.1195–1202.
[73] J. Kwon and K. M. Lee, *Visual tracking decomposition*, 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (2010) pp.1269–1276.
[74] K. Ratnayake and M. A. Amer, *Object tracking with adaptive motion modeling of particle filter and support vector machines*, 2015 IEEE International Conference on Image Processing (ICIP), (2015) pp.1140–1144.
[75] B. Cai, X. Xu, X. Xing, K. Jia, J. Miao, and D. Tao, *BIT: Biologically Inspired Tracker*, **IEEE Trans. Image Process.**, 25:3 (2016) 1327–1339.
[76] J. Zhang, S. Ma, and S. Sclaroff, *MEEM: Robust Tracking via Multiple Experts Using Entropy Minimization*, European Conference on Computer Vision, (2014) pp.188–203.
[77] M. Sharifi, M. Fathy, and M. T. Mahmoudi, *A classified and comparative study of edge detection algorithms*, Proceedings. International Conference on Information Technology: Coding and Computing, pp.117–120.
[78] Anonymous. (2018). http://cs231n.github.io/convolutional-networks/#conv (on-line access on 21, May, 2018).
[79] B. Rohrer. (2018). http://brohrer.github.io/how_convolutional_neural_networks _work.html (on-line access on 21, May, 2018).
[80] A. Radford, L. Metz, and S. Chintala, *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*, **arXiv Prepr. arXiv1511.06434**, (2015).
[81] S. Ruder, *An overview of gradient descent optimization algorithms*, **arXiv Prepr. arXiv1609.04747**, (2016).
[82] Y. Jia *et al.*, *Caffe: Convolutional Architecture for Fast Feature Embedding*, **arXiv Prepr. arXiv1408.5093**, (2014).
[83] A. Krizhevsky, I. Sutskever, and G. E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, **Adv. Neural Inf. Process. Syst. 25**, (2012) 1106–1114.
[84] C. Szegedy *et al.*, *Going Deeper With Convolutions*, Proceedings of the IEEE conference on computer vision and pattern recognition, (2015) pp.1–9.
[85] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, *You Only Look Once: Unified, Real-Time Object Detection*, **arXiv Prepr. arXiv1506.02640**, (2015).
[86] W. Liu *et al.*, *SSD: Single Shot MultiBox Detector*, **arXiv Prepr. arXiv1512.02325**, (2015).
[87] J. Dai, Y. Li, K. He, and J. Sun, *R-FCN: Object Detection via Region-based Fully Convolutional Networks*, **arXiv Prepr. arXiv1605.06409**, (2016).
[88] G. E. Monahan, *State of the Art—A Survey of Partially Observable Markov Decision Processes: Theory, Models, and Algorithms*, **Manage. Sci.**, 28:1 (1982) 1–16.
[89] M. Kristan *et al.*, *The Visual Object Tracking VOT2014 Challenge Results* (2015) pp.191–217.

[90] M. Kristan *et al.*, *The Visual Object Tracking VOT2013 Challenge Results*, 2013 IEEE International Conference on Computer Vision Workshops, (2013) pp.98–111.
[91] R. J. Williams, *Simple statistical gradient-following algorithms for connectionist reinforcement learning*, **Mach. Learn.**, 8:3–4 (1992) 229–256.
[92] A. Li, M. Lin, Y. Wu, M.-H. Yang, and S. Yan, *NUS-PRO: A New Visual Tracking Challenge*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 38:2 (2016) 335–349.
[93] M. Mueller, N. Smith, and B. Ghanem, A Benchmark and Simulator for UAV Tracking. Springer, Cham, pp. 445–461, 2016.
[94] Boyd, Stephen, et al. *Distributed optimization and statistical learning via the alternating direction method of multipliers*. **Foundations and Trends in Machine learning**, 3.1 (2011) 1-122.
[95] Z. Kalal, K. Mikolajczyk, and J. Matas, *Tracking-Learning-Detection*, **IEEE Trans. Pattern Anal. Mach. Intell.**, 34:7 (2012) 1409–1422.
[96] M. Danelljan, F. S. Khan, M. Felsberg, and J. van de Weijer, *Adaptive Color Attributes for Real-Time Visual Tracking*, 2014 IEEE Conference on Computer Vision and Pattern Recognition, (2014) pp.1090–1097.
[97] M. Danelljan, G. Häger, F. Shahbaz Khan, and M. Felsberg, *Accurate Scale Estimation for Robust Visual Tracking*, Proceedings of the British Machine Vision Conference 2014, (2014) pp.65.1-65.11.
[98] D. G. Lowe, *Distinctive Image Features from Scale-Invariant Keypoints*, **Int. J. Comput. Vis.**, 60:2 (2004) 91–110.
[99] G. Nebehay and R. Pflugfelder, *Consensus-based matching and tracking of keypoints for object tracking*, IEEE Winter Conference on Applications of Computer Vision, (2014) pp.862–869.
[100] H. Ebbinghaus, *Memory: a contribution to experimental psychology.*, **Ann. Neurosci.**, 20:4 (2013) 155–6.
[101] R. Zabih and J. Woodfill, Non-parametric local transforms for computing visual correspondence, *Computer Vision — ECCV '94*. Berlin/Heidelberg: Springer-Verlag, pp. 151–158, 1994.
[102] Y. Wu and Y. Liu, Robust Truncated Hinge Loss Support Vector Machines, *Journal of the American Statistical Association*, 102. Taylor & Francis, Ltd.American Statistical Association, pp. 974–983.
[103] S. Maji and A. C. Berg, *Max-margin additive classifiers for detection*, 2009 IEEE 12th International Conference on Computer Vision, (2009) pp.40–47.
[104] Y. Grandvalet and Y. Bengio, *Semi-supervised Learning by Entropy Minimization*, **Adv. Neural Inf. Process. Syst.**, (2005) 529–536.
[105] K. Zhang, L. Zhang, and M.-H. Yang, *Real-Time Compressive Tracking*, European conference on computer vision, (2012) pp.864–877.
[106] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, pp.I-511-I-518.
[107] Y. Wu, J. Lim, and M.-H. Yang, *Online Object Tracking: A Benchmark*, Proceedings of the IEEE conference on computer vision and pattern recognition, (2013) pp.2411–2418.

7. ÖZGEÇMİŞ

Ad Soyad **:** Hüseyin ÜZEN

Doğum Yeri ve Tarihi **:** Silopi/Şırnak – 25.02.1992

Adres **:** Bingöl Üniversitesi Bilgisayar Mühendisliği Bölümü - Bingöl

E-Posta **:** huzen@bingol.edu.tr

Lisans **:**Süleyman Demirel Üniversitesi, Mühendislik Mimarlık Fakültesi, Bilgisayar Mühendisliği (2015)

Mesleki Deneyimler **:**Bingöl Üniversitesi Bilgisayar Mühendisliği, Araştırma Görevlisi (2017- devam ediyor)

Yayın listesi:

TEZDEN TÜRETİLEN YAYINLAR/SUNUMLAR

H. ÜZEN, K. HANBAY, Evaluation of Object Tracking Performance of ADNet Method with Different Datasets and Color Spaces, International Conference on Advanced Technologies, Computer Engineering and Science (ICATCES 2018), Safranbolu, 11-13.05.2018

H. ÜZEN, K. HANBAY, Comparison of Deep Learning and Correlation Filter-Based Methods in Object tracing, 2nd International Vocational Science Symposium (IVSS – 2018), Belek/Antalya, 26-29.04.2018

H. ÜZEN, K. HANBAY, Impact on The Performance of The Search Area in Object Tracking, 2nd International Vocational Science Symposium (IVSS – 2018), Belek/Antalya, 26-29.04.2018

K. HANBAY, **H. ÜZEN**, Nesne Tespit Ve Takip Metotları: Kapsamlı Bir Derleme, Türk Doğa ve Fen Dergisi, 2017, 2149-6366, 6, 2, 40-49.