

# İnsansız Hava Araçlarının Konumlandırılarak Takip Edilmesi

## Detection Based Tracking of Unmanned Aerial Vehicles

Bedirhan Uzun\*, Onur Eker\*, Hasan Sarıbaş†, Hakan Çevikalp\*

\*Elektrik ve Elektronik Mühendisliği Bölümü, Eskişehir Osmangazi Üniversitesi, Eskişehir, Türkiye  
bedirhan@ogu.edu.tr, {onureker34, hakan.cevikalp}@gmail.com

†Havacılık Elektrik ve Elektronik Bölümü, Eskişehir Teknik Üniversitesi, Eskişehir, Türkiye  
hasansaribas@eskisehir.edu.tr

**Özetçe**—Nesne takibi hızlı kamera hareketi, örtüşme, benzer nesnelere gibi içerisinde birçok zorluk bulunduran bilgisayarlı görüşün çok çalışılan temel problemlerinden birisidir. Gelişen teknoloji ile birlikte çok yüksek işlemci gücüne sahip, küçük ve hafif kart bilgisayarlar geliştirilmiştir. Bu kart bilgisayarların insansız hava araçlarına (İHA) entegre edilmesiyle birçok bilgisayarlı görüş uygulaması gerçek zamanlı olarak İHA'larda uygulanabilir hale gelmiştir. Bu çalışmada İHA'ların bir başka İHA tarafından algılanarak konumlandırılması ve takip edilmesi için hibrit bir yöntem tasarlanmıştır. İHA'yı videonun başlangıcında ve takip yönteminin başarısız olduğu durumlarda konumlandırmak için literatürdeki en hızlı ve en yüksek doğruluğa sahip yöntemlerden birisi olan derin öğrenme (deep learning) tabanlı YOLOv3 ve YOLOv3-Tiny (You Only Look Once) modelleri kullanılmıştır. Algılanan nesnelere gerçek zamanlı takibi için ise Çekirdeklesmiş İlinti Filtresi (Kernelized Correlation Filter (KCF)) kullanılmıştır. Ayrıca YOLOv3 modellerinin eğitilmesi ve test edilmesi amacıyla farklı İHA'ların bulunduğu veri seti oluşturulmuştur. Oluşturulan veri seti üzerinde önerilen yöntemler test edilerek performansları değerlendirilmiştir.

**Anahtar Kelimeler**—nesne takibi, nesne tanıma, insansız hava aracı, derin öğrenme, ilinti filtresi.

**Abstract**—Object tracking is one of the fundamental problems of computer vision, which has many difficulties such as fast camera motion, occlusion and similar objects. Today, small and lightweight single board computers with very high processing power have been developed. Real-time processing of the computer vision applications on unmanned aerial vehicles has become possible with the integration of such single board computers within UAVs. In this study, a hybrid method is developed to detect and track UAVs by another UAV. A deep learning based approach which is one of the fastest and most accurate method in the literature, YOLOv3 and YOLOv3-Tiny (You Only Look Once), are utilized to detect the UAV at the beginning of the video and when tracking of the UAV is failed. Kernelized Correlation Filter (KCF) is used for real time tracking purpose of the detected UAVs. A dataset is created that consists different UAVs to train and test YOLOv3. Performance of the proposed methods are evaluated on this dataset.

**Keywords**—object tracking, object detection, unmanned aerial vehicle, deep learning, correlation filter.

### I. GİRİŞ

Görsel nesne takibi, kameradan alınan görüntülerden takip edilmesi istenen nesnenin konumunun belirlenmesi işlemidir. Nesne takibinin insansız hava araçlarıyla takip ve gözetleme, güvenlik kamera sistemleri, trafik gözetleme gibi birçok farklı alanda uygulamaları mevcuttur. Bilgisayarlı görüşte temel bir problem olan ve araştırmacılar tarafından üzerinde çok çalışılan nesne takibi, kontrollü ortamlarda tatmin edici sonuçlar verse de ışıklandırma değişikliği, hızlı kamera ve nesne hareketleri, örtüşme, karışık arka plan gibi pratik uygulamalarda sıklıkla karşılaşılan problemlere hala çözüm getirilememesi nedeniyle güncelliğini hala korumaktadır.

Önceki yıllarda nesne takip yöntemlerinin çoğu, üretici ya da ayırt edici yaklaşımı benimsemektedir. Üretici yöntemler [1]–[3] başlangıçta verilen nesnenin konum bilgisini kullanarak sonraki framerde takip işlemini gerçekleştirir. Üretici yöntemlerde takip işlemi sırasında, nesnenin önceki framedeki konumunun etrafında pencereler oluşturularak nesne sonraki framede aranır. Arama sonuçlarında kullanılan ölçütün en yüksek benzerliği verdiği bölge yeni konum olarak seçilir. Bu yöntemlerde, takip edilecek olan nesnenin konum bilgisi başlangıçta verilir ve sistem önceden eğitilmez. Ayırt edici yöntemlerde [4], [5] ise üretici yöntemlere karşı takip edilecek nesnelere sisteme önceden eğitilerek öğretilir. Eğitim sonucunda sisteme öğretilen nesnenin bulunması ve takip edilmesi istenir. Bu yaklaşımı benimseyen yöntemlerin doğrulukları üretici yöntemlere nazaran oldukça yüksektir. Ancak ayırt edici yöntemlerde sadece eğitilen nesnelere takibinin mümkün olması, nesnenin görünüm değişimlerinde eğitilen modelin nesneyi tanıyamaması ve üretici yöntemlere göre oldukça yavaş çalışması gibi dezavantajlar vardır.

Günümüzde ise nesne takibinde en iyi sonuçları veren yöntemler genellikle derin öğrenme tabanlı ve ilinti filtreleri olarak iki farklı kategoride incelenmektedir. Derin öğrenme tabanlı yöntemlerde [6]–[8] büyük CNN mimarilerinin son katmanları anlam çıkarmada oldukça başarılı olsa da nesnelere ayrıntıları yakalamada yetersiz kalmaktadırlar. Bundan dolayı, nesne takibinde CNN mimarisinde ilk katmanların nesnelere görünüm değişikliklerine karşı daha hassas olması ve bu da hedef konumlarının belirlenmesinde daha etkili olacağından

genellikle küçük CNN mimarileri kullanılır. Ayrıca, büyük mimariye sahip ağlarda parametrelerin güncellenmesi zaman alacağından gerçek zamanlı çalışan bir nesne takip sistemi geliştirilemez. İlinti filtresi tabanlı yöntemlerde [9]–[11] takip edilmek istenen nesnenin konumunu belirlemek için Fourier frekans uzayında sırt (ridge) regresyon problemi çözülerek bir ilinti filtresi öğrenilir. Nesnenin video serisinde bir sonraki framede konumunu bulmak için öğrenilen filtre ilgi bölgesine (ROI) uygulanır ve maksimum ilinti filtresi tepkisini veren bölge nesnenin yeni konumu olarak belirlenir. Aynı zamanda, nesnenin belirlenen yeni konum bilgisi kullanılarak filtre güncellenir. İlinti filtresi tabanlı yöntemler derin öğrenme tabanlı yöntemlere kıyasla oldukça hızlıdır.

Nesne konumlandırma, genel nesne sınıflarında bulunan örneğin imgedeki konum ve ölçeğinin bulunması işlemidir. Nesne konumlandırmada iki kritik faktör vardır; nesneyi belirlemede kullanılan öznitelikler ve konumlandırma işleminde kullanılan öğrenme algoritması. Konumlandırma işleminde öznitelik olarak HOG [12], LBP [13] ve CNN öznitelikleri kullanılmaktadır. Öznitelik çıkartma işleminin ardından nesnenin tespit edilmesi için sistemin eğitilmesi gerekmektedir. Günümüzde, en yüksek başarımlara ulaşan nesne konumlandırma algoritmaları derin öğrenme mimarisine sahip ve ekran kartları sayesinde gerçek zamanlı çalışabilen sistemlerdir. Bu yöntemler arasında en popüler olanlar YOLO [14], SSD [15], Faster R-CNN [16] olarak belirtilebilir. Bu çalışmada nesne konumlandırma işlemi için yüksek başarımlı ve gerçek zamanlı çalışan derin öğrenme mimarisine sahip YOLOv3 ve YOLOv3-Tiny modelleri kullanılmıştır [17].

## II. YÖNTEM

Bu bildiriye insansız hava araçlarının takibini gerçek zamanlı olarak yürüten hibrit yöntem geliştirilmiştir. Amaç doğrultusunda literatürde yüksek hızıyla öne çıkan Çekirdekleşmiş İlinti Filtresi (Kernelized Correlation Filter (KCF)) [10] görsel takip yöntemi ile nesne konumlandırma problemine hızlı ve iyi sonuçlar veren yöntemlerden biri olan YOLOv3 (You Only Look Once) ve YOLOv3-Tiny [17] birlikte kullanılmıştır. Ayrıca YOLOv3 yönteminin eğitilmesi ve tasarlanan yöntemin test edilmesi için farklı İHA'ları içeren veri seti oluşturulmuştur.

### A. Çekirdekleşmiş İlinti Filtresi

İlinti, iki sinyal arasındaki benzerliğin ölçüsüdür. Birbirine daha çok benzeyen sinyallerin ilinti değeri daha yüksek çıkacaktır. Görsel hedef takibi algoritmalarında, ilinti filtresi takip edilecek hedefin üzerine uygulandığında maksimum tepkiyi verecek şekilde tasarlanır. Algoritmayı hızlandırmak ve eğitilen filtrenin hesaplama karmaşıklığını azaltmak için, KCF dolanır matrislerin (circulant matrices) [18] özelliklerinden faydalanır. Ayrıca takip doğruluğunu arttırmak için HOG [12] öznitelikleri kullanılır. Bu algoritmanın avantajlarına değinmek gerekirse, literatürdeki yüksek hıza sahip takip algoritmalarından biri olmasının yanı sıra yönelme ve ölçek değişimlerine karşı gürbüzlük ve doğruluk oranı ile öne çıkmaktadır. Buna rağmen hedefi sadece önceki konumunun etrafında aramasından dolayı örtüşme ve hedefin görüş dışı kaldığı durumlarda, kendini düzeltme mekanizması olmadığı için izleyen takip etme görevi başarısızlıkla sonuçlanır. Kusurlarına ek olarak en-boy oranı dinamik bir şekilde değişmemektedir.

### B. YOLO

Redmon ve arkadaşları [17], giriş olarak ham görüntü piksellerini kullanıp nesneyi çevreleyen pencerenin koordinatlarını ve sınıf olasılıklarını tahmin eden baştan-başa eğitilebilir tek aşamalı bir yapay sinir ağı tasarlamışlardır. Bundan dolayı, YOLO aşırı miktarda hızlı nesne konumlandırma algoritmasıdır. Takip amacıyla YOLO kullanıldığında karşılaşılan sorunlara bakılırsa, takip görevi için gönderilen imge, hedef nesneden birden fazla içeriyorsa, algoritma doğası gereği hepsini tespit etmeye çalışacaktır bu da genel görsel takip uygulamalarının usulüne uygun değildir. Ayrıca ekran kartı üzerinde çalıştığında ne kadar hızlı sayılsa da, KCF gibi işlemci üzerinde 100+ fps hıza ulaşabilen bir yöntem değildir ve fazla hesaplama kaynağı gerektirir. Ek olarak takip edeceği nesneyi bir sonraki framede tahmin edememesi olasılık dahilinde olduğundan sürekli sonuç üretmesi kesin değildir.

### C. Önerilen Yöntem

Yukarıdaki yöntemlerin avantajları ve dezavantajları ele alındığında, bahsedilen algoritmaların birbirinin kusurlarını örttüğü gözlenmiştir. KCF yöntemini kötü durum kurtarıcısına ihtiyacı varken, YOLO algoritmasının da sadece takip edilecek hedefe odaklanmaya ve tek kartlı bilgisayarlarda daha hızlı çalışması için optimize edilmeye ihtiyacı vardır. Bu sebeple önerilen hibrit takipçide (ÖHT) YOLO ilk framede nesnenin konumunu belirlemede ve takip edicinin başarısız olduğu durumlarda kullanılırken, KCF yöntemi ise diğer durumlarda nesneyi takip amaçlı kullanılmıştır. Bu şekilde hem YOLO konumlandırıcısını hem de KCF takipçisini kullanan yeni bir hibrit takip edici sistem oluşturulmuştur. Bunu gerçekleştirebilmek için öncelikle YOLO ağı, İHA imgelerinden oluşturulan veri seti ile eğitilmiştir. Hızdan taviz vermemek için YOLO yöntemini C dilinde yazılmış orjinal implementasyonunun derlenmiş dinamik kütüphanesindeki fonksiyonları, Python dilinde çağrılarak kullanılmıştır. Bu kısımdan sonra KCF ve YOLO yöntemleri için sırasıyla, takipçi ve dedektör sözcükleri kullanılmıştır. KCF yönteminden farklı olarak, takipçinin güvenilirlik değeri belirlenen eşik değerden düşükse, dedektörü tüm imgedeki İHA'ları bulması için çağırılıp Denklem (1)'de gösterildiği gibi, tahmin edilenlerden ( $k$  tane) ( $P_{D_i}$ ), takipçinin son ürettiği sonuç ( $P_T$ ) ile en yüksek IoU (Intersection-over-Union) skoru vereni kullanarak takipçi eğitilmiştir.

$$\operatorname{argmax}_{i \in k} \left( \frac{|P_T \cap P_{D_i}|}{|P_T \cup P_{D_i}|} \right) \quad (1)$$

Şu ana kadarki durumda takipçinin hata durumunda kurtarılması ve dinamik olarak güncellenen en-boy oranı algoritmaya katılmıştır. Fakat çok ani frame atlamaları sebebiyle, dedektörün bulunduğu sonuçlardan hiçbiri takipçinin sonucu ile kesişmeyebilir. Bahsi geçen durum oluşmuşsa Denklem (2)'de gösterildiği gibi, dedektör tarafından tahmin edilenlerin merkez koordinatları ( $k$  tane) ( $x_{D_i}, y_{D_i}$ ) ile takipçinin son ürettiği sonucun merkez koordinatları ( $x_T, y_T$ ) arasındaki Öklid uzaklığına bakılarak metrik bazında en yakın olanı seçilip, takipçi eğitilir.

$$\operatorname{argmin}_{i \in k} \left( \sqrt{(x_T - x_{D_i})^2 + (y_T - y_{D_i})^2} \right) \quad (2)$$

Sadece güvenilirlik değeri belirlenen eşik değerinden düşük olduğunda dedektör çalıştığı için, önerilen yöntem YOLO'dan

daha hızlı çalışmaktadır. Sonuç olarak, önerilen yöntem KCF'den hız, tek hedefe kilitlenme ve sürekli sonuç üretme avantajlarını alırken, YOLO'yu da takipçide oluşacak kötü durumları kurtarması ve en-boy oranını dinamik olarak güncellemesi için kullanmıştır.

### III. DENEYLER

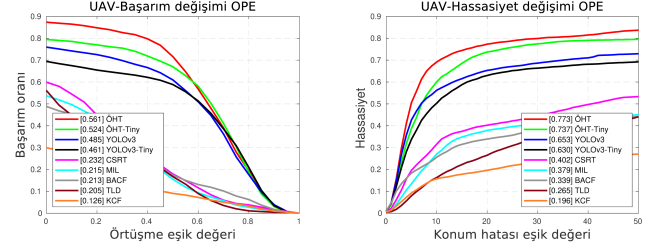
Videodan nesne takibi için OTB (Object tracking benchmark), VOT (Visual object tracking) gibi birçok veri seti bulunmasına rağmen sırf İHA'lardan oluşan veri seti bulunmamaktadır. Bu sebepten dolayı kendi spesifik problemimiz için yaklaşık 7500 imge etiketlendirilerek veri seti oluşturulmuştur. Bu veri setinin yaklaşık 2000 imgelik kısmı YOLOv3'ün eğitimi için, 15 videodan oluşan yaklaşık 5500 imgelik kısmı ise test için ayrılmıştır. Şekil 1'de oluşturulan veri setinden örnekler mevcuttur.

Yöntemler test edilirken adil bir karşılaştırma olması için, OTB değerlendirme protokolü kullanılmıştır. Bu protokolde sonuçlar hem başarı (success) hem de hassasiyet (precision) skorları ile birlikte verilmektedir. Başarı skoru hesaplanırken hedef nesnenin gerçek pozisyonu (ground-truth) ( $R_{GT}$ ) ile yöntemin ürettiği pozisyon (predicted bounding box) ( $R_P$ ) arasında ki örtüşmeye bakılır,  $IoU = \frac{|R_{GT} \cap R_P|}{|R_{GT} \cup R_P|}$ . IoU belirli bir eşik (threshold) değerinden büyükse başarılı olarak ifade edilir ve bu tüm frame'ler boyunca tekrarlanarak başarılı olan frame'lerin toplam frame oranı başarı skoru olarak hesaplanır. Bu işlem veri setindeki tüm videolar için yapılarak toplam başarı skoru elde edilmiş olur. Bir diğer metrik olan hassasiyet hesaplanırken ise nesnenin gerçek pozisyonu ile tahmin edilen pozisyonun merkezleri arasındaki mesafeye bakılır. Bu mesafenin 20 pikselden küçük olduğu frame'lerin toplam frame'lere oranı hassasiyet skorunu verir.



Şekil 1: Oluşturulan veri setinden seçilmiş örnek imgeler.

Deneyler hem YOLOv3 hem de YOLOv3-Tiny modeli için yapılmıştır. Ayrıca önerilen yöntemin İHA üzerine entegre edilebilen Jetson TX2 işlemcisinde gerçek zamanlı çalışabilmesi için YOLOv3-Tiny modeli de ayrıca eğitilerek, daha hızlı çalışabilen bir yöntem (ÖHT-Tiny) oluşturularak test edilmiştir. Modeller eğitilirken NVIDIA Quadro P5000 ekran kartı kullanılmıştır. Sonuçlar değerlendirilirken ise Xeon E-2186M işlemciye ve Quadro P4200 ekran kartına sahip dizüstü bilgisayar kullanılmıştır.



Şekil 2: Oluşturulan veri seti üzerinde yöntemlerin tek geçiş değerlendirilmesi (One Pass Evaluation-OPE) protokolü ile performanslarının karşılaştırılması.

Tablo I: ÖNERİLEN YÖNTEM İLE LİTERATÜRDEKİ YÖNTEMLERİN İHA VERİ SETİ ÜZERİNDE PERFORMANSLARININ KARŞILAŞTIRILMASI (SIRASIYLA KIRMIZI, YEŞİL VE MAVİ RENKLER EN YÜKSEK SKORLARI GÖSTERMEKTEDİR)

Yöntem	Başarı	Hassasiyet	Hız (fps)
ÖHT	0.561	0.773	53.5
ÖHT-Tiny	0.524	0.737	69.2
YOLOv3	0.485	0.653	16.5
YOLOv3Tiny	0.461	0.630	47.1
CSRT	0.232	0.402	111
TLD	0.205	0.265	20.1
MIL	0.215	0.379	12.3
KCF	0.126	0.196	115
BACF	0.213	0.339	25.8

Test için ayrılan videolar kullanılarak literatürdeki son yıllarda yayınlanmış yüksek başarımlara sahip yöntemler olan; BACF [19], KCF [10], TLD [5], MIL [20], CSRT [21] ve ayrıca eğitim seti kullanılarak eğitilmiş YOLOv3 ve YOLOv3-Tiny modelleri geliştirilen yöntemlerle karşılaştırılmıştır. Tablo I ve Şekil 2'de sıralanan yöntemlere ait başarımlı ve hassasiyet skorları verilmiştir. Sonuçlara bakıldığında önerdiğimiz hibrit yöntemlerin en yüksek başarımlı ve hassasiyet skorlarına sahip olduğu, hız açısından ise en hızlı üçüncü yöntem olduğu gözlemlenmiştir. Ayrıca daha hızlı çalışan yöntemler olan, KCF ve CSRT yöntemlerinin ise test edilen veri seti üzerindeki hem başarımlı hem de hassasiyet skorlarının çok düşük olduğu görülmüştür. Bunun yanı sıra önerilen hibrit yöntem kendini oluşturan YOLOv3 ve KCF takip yöntemlerine oranla çok daha iyi sonuçlar vermiştir. Şekil 3'de tasarlanan yöntemler ile CSRT ve KCF yöntemlerinin, oluşturulan veri seti üzerindeki görsel karşılaştırılması verilmiştir.

### IV. SONUÇLAR

Bu çalışma ile İHA'ların tespit ve takip edilmesi amacıyla YOLOv3 ve KCF yöntemleri kullanılarak hibrit bir yöntem tasarlanmış ve bu yöntem literatürdeki gerçek zamanlı ve yüksek başarımlı ile çalışabilen diğer yöntemlerle karşılaştırılmıştır. Bu amaçla hem yöntemin eğitilmesi hem de test edilmesi için farklı İHA'lar içeren veri seti oluşturulmuştur. Tasarlanan yöntemin hızının artırılması amacıyla YOLOv3-Tiny modeli de ayrıca eğitilerek test edilmiştir. Sonuçlardan ÖHT yönteminin beklendiği şekilde ÖHT-Tiny ve diğer yöntemlere oranla daha yüksek başarımlı elde ettiği ancak ÖHT-Tiny modelinden daha yavaş çalıştığı gözlemlenmiştir. Gerçeklenen yöntem şu anda sadece İHA'ları tespit ve takip edilebilecek şekilde tasarlanmıştır ancak karşılaştırıldığı diğer yöntemler ise



Şekil 3: Oluşturulan veri seti üzerinde literatürdeki yöntemlerin sonuçlarının karşılaştırılması (sırasıyla siyah, kırmızı, mavi, yeşil ve sarı renkli çerçeveler, nesnenin gerçek pozisyonu, ÖHT, ÖHT-Tiny, CSRT ve KCF yöntemlerini temsil etmektedir).

genel nesne takibi amacıyla tasarlanmıştır. Sonuç olarak gerçek zamanlı çalışabilen yüksek performanslı bir takip yöntemi geliştirilmiştir.

#### BİLGİLENDİRME

Bu çalışma Tübitak tarafından 116E080 Nolu proje kapsamında desteklenmiştir.

#### KAYNAKLAR

- [1] X. Mei and H. Ling, "Robust visual tracking using l1 minimization," in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1436–1443.
- [2] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *International journal of computer vision*, vol. 77, no. 1-3, pp. 125–141, 2008.
- [3] T. Zhang, B. Ghanem, S. Liu, and N. Ahuja, "Low-rank sparse learning for robust visual tracking," in *European conference on computer vision*. Springer, 2012, pp. 470–484.
- [4] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 10, pp. 2096–2109, 2016.
- [5] Z. Kalal, K. Mikolajczyk, J. Matas *et al.*, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, p. 1409, 2012.
- [6] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4293–4302.
- [7] H. Li, Y. Li, and F. Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1834–1848, 2016.
- [8] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3074–3082.
- [9] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 2544–2550.
- [10] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
- [11] M. Danelljan, A. Robinson, F. S. Khan, and M. Felsberg, "Beyond correlation filters: Learning continuous convolution operators for visual tracking," in *European Conference on Computer Vision*. Springer, 2016, pp. 472–488.
- [12] H. Cevikalp and B. Triggs, "Visual object detection using cascades of binary and one-class classifiers," *International Journal of Computer Vision*, vol. 123, no. 3, pp. 334–349, 2017.
- [13] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 12, pp. 2037–2041, 2006.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [17] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [18] P. J. Davis, *Circulant matrices*. American Mathematical Soc., 2012.
- [19] H. K. Galoogahi, A. Fagg, and S. Lucey, "Learning background-aware correlation filters for visual tracking," in *ICCV*, vol. 3, 2017, p. 4.
- [20] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2011.
- [21] A. Lukezic, T. Vojir, L. Cehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter with channel and spatial reliability," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6309–6318.