

Çoklu Nesne Takibi FairMOT Algoritması İçin Optimizasyon Algoritmalarının Karşılaştırılması

Comparison of Optimization Algorithms for Multi-Object Tracking FairMOT Algorithm

Mehmet Fatih Özdemir^{*1} , Ali Arı¹ , Davut Hanbay¹ 

¹Bilgisayar Mühendisliği Bölümü, İnönü Üniversitesi, Malatya, Türkiye

(mfatih.ozdemir@inonu.edu.tr, ali.ari@inonu.edu.tr, davut.hanbay@inonu.edu.tr)

Received: Sep.2, 2021

Accepted: Sep.26, 2021

Published: Oct.20, 2021

Özetçe— Çoklu nesne takibi alanında günümüzde birçok yöntem kullanılmaktadır. Derin öğrenme algoritmaları bu alanda en çok çalışılan yöntemler arasında yer almaktadır. Derin öğrenme tabanlı sistemlerde yüksek performans elde edilebilmesi için uyarlanması gereken birçok parametre vardır. Derin öğrenme sistemlerinde performansı etkileyen önemli parametrelerden birisi de kullanılan optimizasyon algoritmasıdır. Bu çalışmada FairMOT algoritması için Adam, RMSProp, Rprop, SGD optimizasyon algoritmaları karşılaştırılmıştır. Optimizasyon algoritmaları karşılaştırılırken MOT20 veri seti kullanılmıştır. MOT20 doğrulama veri setinde ortalama en yüksek doğruluk değeri, RMSprop optimizasyon algoritması ile %76.7 olarak elde edilmiştir.

Anahtar Kelimeler : FairMOT, Çoklu Nesne Takibi, Optimizasyon Algoritmaları

Abstract— Many methods are used in the field of multi-object tracking today. Deep learning algorithms are among the most studied methods in this field. There are many parameters that need to be adapted in order to achieve high performance in deep learning based systems. One of the important parameters affecting performance in deep learning systems is the optimization algorithm used. In this study, Adam, RMSProp, Rprop, SGD optimization algorithms were compared for FairMOT algorithm. The MOT20 dataset was used when comparing the optimization algorithms. In the MOT20 validation dataset, overall highest accuracy value was obtained as 76.7% with the RMSprop optimization algorithm.

Keywords : FairMOT, Multi-Object Tracking, Optimization Algorithms.

1.Giriş

Çoklu nesne takibi, görüntü çerçeveleri üzerinde aynı veya farklı nesnelerin takibini amaçlayan, gelişen bilgisayar donanımları ile birlikte yükseliş eğilimi de olan çalışma alanlarından biridir. Görüntü üzerinde birden fazla nesneyi aynı anda takip etmeyi amaçlayan sistemlere çoklu nesne takibi yapan sistemler denilmektedir. Çoklu nesne takibi genel anlamda iki aşamadan oluşmaktadır. İlk aşama görüntü çerçevesinde nesnelerin tespit edilmesi, ikincisi ise sıralı görüntüde tespit edilen nesnelerin ilişkilendirilmesi aşamasıdır. Bir algoritmanın başarı oranı bu iki aşamanın başarısına bağlı olarak değişkenlik göstermektedir. Bu alanda geliştirilen algoritmalar, tespit ve ilişkilendirme aşamalarına farklı yaklaşabilmektedirler. İki aşamayı iki ayrı modelde ele alan algoritmaların (Ren *et al.*, 2017; Wojke, Bewley and Paulus, 2018; Yang *et al.*, 2020) yanı sıra tespit ve ilişkilendirme adımlarını tek bir model ile çözen algoritmalar (Redmon *et al.*, 2016; Zhang *et al.*, 2020) da vardır. Tekil modelin gerçek

zamanlı sistemlerde başarı oranının daha yüksek olması bu modellere avantaj sağlamaktadır. Çünkü ikili modellerde öznelilikler paylaşamadığı için tespit aşamasında elde edilen öznelilikler ilişkilendirme aşamasında tekrar hesaplanmak durumundadır (Zhang *et al.*, 2020).

Bu alanda birçok çalışma mevcuttur. Faster R-CNN (Ren *et al.*, 2017), YOLO (Redmon *et al.*, 2016) bu alanda başarıları yüksek algoritmalar arasındadır.

Faster R-CNN (Ren *et al.*, 2017) iki aşamayı iki ayrı modelde sunan evrişimli sinir ağıdır. Bu derin öğrenme algoritmasını anlamak için temellerini aldığı R-CNN (Girshick *et al.*, 2014), Fast R-CNN (Girshick, 2015) algoritmalarını anlamak gerekir. R-CNN (Girshick *et al.*, 2014), teklif bölgelerini nesne kategorilerine veya arka plana göre sınıflandırmak için evrişimli sinir ağlarını uçtan uca eğitir. Fast R-CNN (Girshick, 2015) , paylaşılan evrişimli özellikler hakkında uçtan uca dedektör eğitimi sağlar ve R-CNN (Girshick *et al.*, 2014), modeline göre daha iyi doğruluk ve hız kazanımı sağlar. Faster R-CNN (Ren *et al.*, 2017), Fast R-CNN (Girshick, 2015) algoritmasına ek olarak algılama ağı ile tam görüntü evrişimli özelliklerini çıkaran yöntem sayesinde bağımsız bölge önerileri sağlayan bir bölge teklif ağı (RPN) ekleyerek iyileştirmelere yer vermiştir.

YOLO (Redmon *et al.*, 2016) nesne algılamayı uzamsal olarak ayrılmış sınırlayıcı kutulara ve ilişkili sınıf olasılıklarına bir regresyon problemi olarak yaklaşır. Bir adet sinir ağı ile tek değerlendirmede doğrudan tam görüntülerden sınırlayıcı kutuları ve sınıf olasılıklarını tahmin etmektedir. Bütün algılama hattı tek bir ağı olduğundan, doğrudan algılama performansına göre uçtan uca optimize çalışır.

MOT20 (Dendorfer *et al.*, 2020) veri seti toplamda 8 adet video görüntülerinden oluşmaktadır. Bu veri seti içerisindeki bazı videoların görüntü çerçevesinden 246 adet yaya yoğunluğu bulunmaktadır. Patrick ve arkadaşları bu veri setini ilk olarak Faster R-CNN (Ren *et al.*, 2017) algoritmasıyla derin öğrenme modeli çıkararak sunmuşlardır.

Bu çalışmada MOT20 (Dendorfer *et al.*, 2020) veri seti üzerinde FairMOT (Zhang *et al.*, 2020) nesne takibi algoritmasının daha başarılı çalışması için optimizasyon algoritmaları karşılaştırılması yapılmıştır. Her bir optimizasyon algoritması için ayrı ayrı 20 tur eğitim gerçekleştirilmiştir. Elde edilen sonuçlar çerçevesinde gözlemlenen en başarılı algoritma RMSprop optimizasyon algoritmasıdır.

İkinci bölümde kullanılan yöntemler kısaca anlatılmıştır. Üçüncü bölümde uygulama adımları açıklanmıştır. Dördüncü bölümde ise elde edilen sonuçlar ve tartışma paylaşılmıştır.

2.Kullanılan Yöntemler

2.1. Adam Optimizasyon Algoritması

Stokastik hedef fonksiyonlarının birinci dereceden gradyan tabanlı optimizasyonu için bir algoritma olan Adam (Kingma and Ba, 2015), düşük dereceli ve uyarlanabilir tahminlere dayalı olarak sunulmuştur. Yöntemin uygulanması basittir, hesaplama açısından etkilidir. Çok az bellek gereksinimine ihtiyaç duyar. Gradyanların diyagonal olarak yeniden ölçeklendirilmesiyle değişmez. Veri veya parametreler bakımından büyük problemler için çok uygundur. Yöntem, aynı zamanda, sabit olmayan hedefler ve çok gürültülü veya seyrek gradyanlı problemler için de uygundur. Hiper parametrelerinin sezgisel yorumları vardır ve tipik olarak çok az ayarlama gerektirir. Ampirik sonuçlar, Adam'ın pratikte iyi çalıştığını ve diğer stokastik optimizasyon yöntemleriyle olumlu bir şekilde karşılaştırıldığını göstermektedir.

2.2. Stokastik Gradyan İniş (SGD) Optimizasyon Algoritması

Stokastik gradyan inişi (SGD) (America and Nj, 2010), gradyan inişinin bir çeşididir. Öğrenme oranı düşük olduğunda normal gradyan inişi ile aynı performansı üretir. SGD, bir makine öğrenimi algoritması için optimal parametre değerlerini bulmak veya uygun düzgünlük özelliklerine sahip bir amaç fonksiyonunu optimize etmek için yinelemeli bir yöntemdir. Hata tahminini azaltmak için sık sık parametre değerlerinde küçük ayarlamalar yapar. Bu yöntem, bütün veri yerine parçalı eğitim veri grupları üzerinde gradyanı hesaplayan ve parametre değerlerini güncelleyen gradyan inişinin (GD) stokastik bir versiyonu olarak düşünülebilir. Gradyan inişi büyük veri kümeleri için uygun değildir ve yavaş çalışır çünkü her yinelemede tüm eğitim verileri için bir tahmin hesaplanır. SGD'nin temel avantajları, verimlilik ve kolay uygulanabilir olmasıdır.

2.3. Rprop Optimizasyon Algoritması

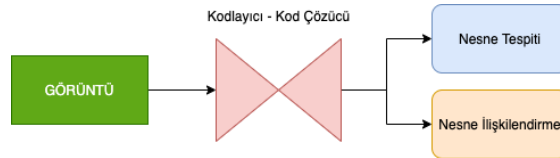
Gradyan inişinin doğal dezavantajlarının üstesinden gelmek için Rprop (Riedmiller and Braun, 1993), hata fonksiyonunun davranışına göre ağırlık güncellemelerinin yerel bir uyarlamasını gerçekleştirir. Diğer tekniklerden önemli bir farkla, Rprop uyarlama sürecinin etkisi, türevin boyutunun öngörülemez etkisiyle bulanıklaşmaz, sadece işaretinin zamansal davranışına bağlıdır. Bu durum verimli ve şeffaf bir uyum sürecine yol açar.

2.4. RMSprop Optimizasyon Algoritması

RMSprop (Tieleman, Hinton and others, 2012), Geoffrey Hinton tarafından tasarlanmıştır. RMSprop, parçalı öğrenme için Rprop (Riedmiller and Braun, 1993), algoritmasının uyarlanması olarak düşünülebilir. RMSprop momentumdan farklı bir şekilde salınımları sönümlemeye çalışmaktadır. RMSprop ayrıca öğrenme oranını ayarlama ihtiyacını ortadan kaldırarak otomatik ayarlanmasını sağlamaktadır. RMSprop her parametre için farklı bir öğrenme oranı seçer.

2.5. FairMOT Çoklu Nesne Takibi Algoritması

FairMOT (Zhang *et al.*, 2020) algoritması çoklu nesne takibi alanında tespit ve ilişkilendirme aşamaları için tekil model sunan derin öğrenme tekniği kullanılarak geliştirilmiş bir algoritmadır. Algoritmada nesne tespiti aşamasında sınır kutusu yerine nesne merkezi yaklaşımı kullanılmıştır (Zhou, Wang and Krähenbühl, 2019). Bu yaklaşım ile nesnenin merkezi noktası bulunmaya odaklanmaktadır. Böylece nesne tespitinde sınırı kutusu yöntemine göre daha kolay ve etkili bir şekilde yapıldığı ifade edilmektedir (Zhou, Wang and Krähenbühl, 2019). Bu algoritma ile tek bir modelde işlem gerçekleştiği için tespit aşamasında çıkarılan öznitelikler aynı şekilde ilişkilendirme aşamasında kullanılmaktadır. Tekil modellenmiş yaklaşım, yüksek boyutlu ilişkilendirme özniteliklerini çıkarımı nesnelere aynı kimliği verme işleminde biraz başarıyı yükseltse de diğer taraftan nesne algılama doğruluğunu düşürmektedir. Bunun toplam başarıyı düşürdüğü belirtilmiştir. Bunun için algoritmada düşük boyutlu öznitelikler kullanılmıştır. Ayrıca düşük boyutlu öznitelik aşırı öğrenme riskini azaltmasından dolayı da avantaj sağlamaktadır ve öğrenme hızını arttırmaktadır. Şekil-1 de gösterildiği gibi FairMOT algoritması görüntüyü kodlayıcı-kod çözücü-kod çözücüden geçirdikten sonra nesne tespiti ve ilişkilendirme adımlarını tamamlamaktadır.



Şekil 1. FairMOT [4]

FairMOT (Zhang *et al.*, 2020) omurga ağı olarak ResNet-34 kullanır. Çok katmanlı öznitelikleri birleştirmek için derin katman toplama (DLA) (Yu *et al.*, 2018) yönteminin daha gelişmiş versiyonu omurgaya uygulanır. Derin katman toplama (DLA) (Yu *et al.*, 2018) yönteminden farklı olarak alt ve üst düzey öznitelikler arasında daha fazla atlama bağlantılarına sahip olmasıdır. Ayrıca, bütün yukarı örnekleme modüllerindeki evrişim katmanları, nesnenin ölçekleri ve pozlarına göre alıcı alanı, dinamik olarak ayarlanabilecek şekilde deforme olabilir evrişim ile değiştirilir. Bu değişiklikler aynı zamanda hizalama sorununu azaltmaya da yardımcı olur.

FairMOT (Zhang *et al.*, 2020) algoritması COCO (Lin *et al.*, 2014) derin öğrenme modelini ile CrowdHuman (Shao *et al.*, 2018) veri setini ön eğitim modeli çıkarılır. Çıkarılan model ile 2DMOT15 (Leal-Taixé *et al.*, 2015), MOT16 (Milan *et al.*, 2016), MOT17 (Milan *et al.*, 2016), ve MOT20 (Dendorfer *et al.*, 2020) görüntü veri setlerinin her biri için tekrar ince ayar eğitimi yapılmıştır. Yapılan çalışmalar doğruluğu değerlendirilirken CLEAR (Bernardin and Stiefelhagen, 2008) ve IDF1 (Ristani *et al.*, 2016) metrikleri kullanılmıştır. Modelin değerlendirme sonuçları Tablo-I de ifade edilmiştir.

Tablo 1. Model Değerlendirme Sonuçları [4]

Veri Seti	MOTA	IDF1	IDS	MT	ML	FPS
2DMOT15	60.6	64.7	591	47.6%	11.0%	30.5
MOT16	74.9	72.8	1074	44.7%	15.9%	25.9
MOT17	73.7	72.3	3303	43.2%	17.3%	25.9
MOT20	61.8	67.3	5243	68.8%	7.6%	13.2

CrowdHuman (Shao *et al.*, 2018) veri seti kullanılarak yapılan ön eğitimde Adam optimizasyon algoritması (Kingma and Ba, 2015) 30 turda eğitilerek model oluşturulmuştur. Eğitimin ilk aşamasında $e^{(-4)}$ öğrenme katsayısı kullanılarak başlanmıştır fakat 20. turdan sonraki eğitimlerde $e^{(-5)}$ öğrenme katsayısı kullanılmıştır. İnce ayar eğitimlerinde ise aynı şekilde Adam optimizasyon algoritması kullanılmıştır fakat 20 turda eğitim gerçekleştirilmiştir (Zhang *et al.*, 2020).

FairMOT (Zhang *et al.*, 2020) çalışmasına incelendiğinde optimizasyon algoritmalarının karşılaştırılması hususunda bir çalışma yapılmadığı gözlemlenmiştir. Bu çalışmada MOT20 (Dendorfer *et al.*, 2020) veri seti üzerinde yapılan ince ayar eğitimi için kullanılan optimizasyon algoritmasının optimize edilmesi için çalışılmıştır.

2.6. Performans Metrikleri

Keni ve arkadaşı Rainer (Bernardin and Stiefelbogen, 2008) çoklu nesne takibi algoritmalarının performanslarını ölçebilmek için 2008 yılında Clear MOT performans metriklerini önermişlerdir. Bu performans metrikleri sırasıyla

Çoklu nesne takibinde hassasiyet (MOTP):

$$MOTP = \frac{\sum_{i,t} h_t^i}{\sum_t b_t} \quad (1)$$

Denklem (1) de t anından mevcut çerçeve için b_t bulunan eşleşme sayısı, h_t^i nesne – hipotez toplam uzaklığı olduğu ifade edilmiştir (Bernardin and Stiefelbogen, 2008).

Çoklu nesne takibinde doğruluk (MOTA):

$$MOTA = \frac{\sum_t (n_t + yp_t + nne_t)}{\sum_t g_t} \quad (2)$$

Denklem (2) de t anında mevcut çerçeve için n_t eksik tespit edilen sayısı, yp_t yanlış pozitiflerin sayısı, nne_t eşleşmeyenlerin sayısı ifade edilmiştir (Bernardin and Stiefelbogen, 2008).

Ergys ve arkadaşları (Ristani *et al.*, 2016) 2016 yılında çoklu hedef ve çoklu kamera takibi için yeni performans metrikleri sunmuşlardır. Kimlik hassasiyetini IDP, kimlik duyarlılığını IDR, kimlik F1 skorunu ise IDF1 olarak tanımlamışlardır.

$$IDP = \frac{IDTP}{IDTP + IDFP} \quad (3)$$

$$IDR = \frac{IDTP}{IDTP + IDFN} \quad (4)$$

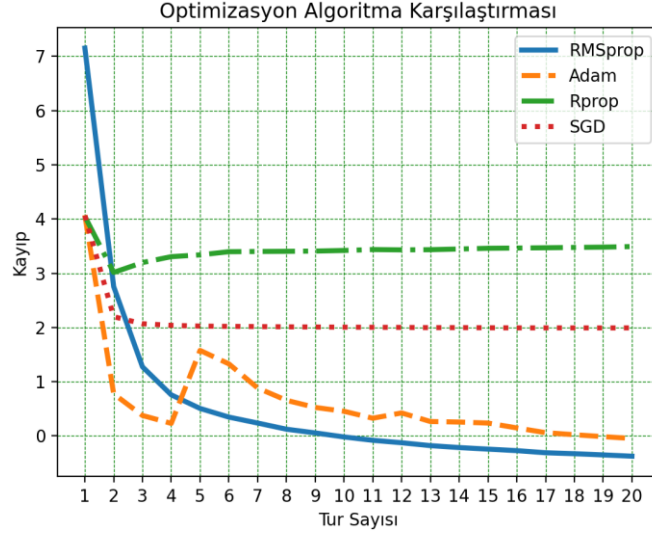
$$IDF1 = \frac{2IDTP}{2IDTP + IDFP + IDFN} \quad (5)$$

Çalışmalarındaki denklemlerde (3), (4), (5) yer alan IDTP kimlik doğru pozitifleri, IDFP kimlik yanlış pozitifleri, IDFN kimlik yanlış negatifleri ifade ettiği belirtilmiştir.

3. Uygulama

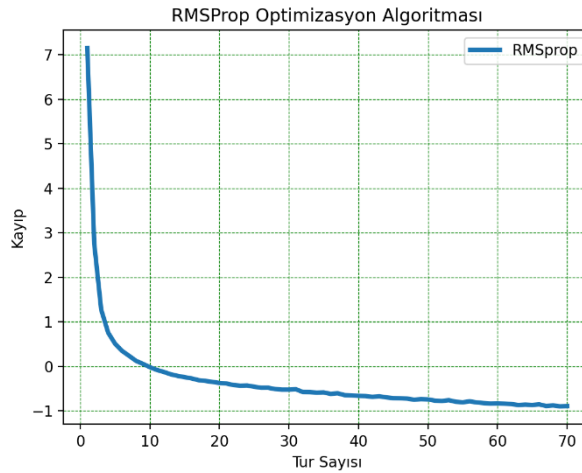
Bu çalışma, PyTorch makine öğrenmesi kütüphanesinin 1.0 versiyonu kullanılarak, Intel(R) Xeon(R) Gold 5122 işlemci, NVIDIA GRID P40 ekran kartı, 16 GB belleğe sahip bir bilgisayarda yapılmıştır. Adam, SGD, RMSProp, Rprop optimizasyon algoritmaları kullanılarak FairMOT (Zhang *et al.*, 2020) algoritması ince ayar eğitimi için en iyi optimizasyon algoritması belirlenmeye çalışılmıştır.

FairMOT (Zhang *et al.*, 2020) algoritması üzerinde ince ayar eğitimi için yapılan dört farklı optimizasyon algoritmaları 20 tur eğitilerek deneme yapılmıştır. Elde edilen sonuçlar Şekil-2 de verilmiştir. Şekil-2 de yatay eksen tur sayısını, dikey eksen kayıp değerini göstermektedir. Şekil-2 de gösterilen deneyde en düşük kayıp değerine sahip RMSprop (Tieleman, Hinton and others, 2012), optimizasyon algoritması olduğu gözlemlenmiştir.



Şekil 2. Optimizasyon Algoritmalarının Karşılaştırması

Model, RMSprop optimizasyon algoritması kullanılarak 20 tur için eğitim başarıları gözlemlendikten sonra 70 tura kadar eğitim devam ettirildi. Şekil-3 yatay eksen tur sayısını, dikey eksen kayıp değerini göstermektedir. Şekil-3 de RMSprop optimizasyon algoritmasının 70 tur eğitim süreci gösterilmiştir.



70 tur eğitilen model MOT20 doğrulama veri seti üzerindeki başarıları Tablo-II de gösterildiği şekilde gözlemlenmiştir.

Tablo 2. MOT20 Doğrulama Veri Seti Değerlendirmesi

Veri Seti	MOTA	MOTP	IDF1	IDP	IDR
MOT20-01	67.9%	0.160	76.4%	91.1%	65.8%
MOT20-02	69.6%	0.153	69.4%	81.8%	60.2%
MOT20-03	77.9%	0.173	85.7%	91.5%	80.7%
MOT20-05	78.4%	0.165	84.7%	90.8%	79.3%
TOPLAM	76.7%	0.166	82.6%	89.8%	76.5%

4. Sonuç

Derin öğrenmede optimizasyon algoritması, modelden modele farklılık gösteren en önemli hiper parametreler arasındadır. Bu çalışmada minimum kayıp değeri ile eğitime başlamak için dört farklı optimizasyon algoritması ile denemeler yapılmıştır. RMSprop [12] algoritması FairMOT modeli için en iyi optimizasyon algoritması olarak gözlemlenmiştir. RMSprop [12] algoritması kullanılarak 70 tur eğitilen model MOT20 doğrulama veri setinde ortalama 76.7% başarı elde edildiği gözlemlenmiştir. Deneme sayısı, test edilen aralık, örneklem sayısı değiştirilerek gelecekte yapılacak olası denemelerde daha iyi sonuç elde edilebilir.

Teşekkür

Bu çalışma İnönü Üniversitesi Bilimsel Araştırma Projeleri Koordinasyon Birimi (BAP) tarafından FYL-2021-2449 kodlu proje ile desteklenmiştir.

Kaynaklar

America, N. E. C. L. and Nj, P. (2010) ‘Large-Scale Machine Learning with Stochastic Gradient Descent (SGD)’, *Proceedings of COMPSTAT’2010*, pp. 3–4. doi: 10.1007/978-3-7908-2604-3.

Bernardin, K. and Stiefelhagen, R. (2008) ‘Evaluating multiple object tracking performance: The CLEAR MOT metrics’, *Eurasip Journal on Image and Video Processing*, 2008. doi: 10.1155/2008/246309.

Dendorfer, P. *et al.* (2020) ‘MOT20: A benchmark for multi object tracking in crowded scenes’, *arXiv*, pp. 1–7.

Girshick, R. *et al.* (2014) ‘Rich feature hierarchies for accurate object detection and semantic segmentation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 580–587. doi: 10.1109/CVPR.2014.81.

Girshick, R. (2015) ‘Fast R-CNN’, *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter, pp. 1440–1448. doi: 10.1109/ICCV.2015.169.

Kingma, D. P. and Ba, J. L. (2015) ‘Adam: A method for stochastic optimization’, *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15.

Leal-Taixé, L. *et al.* (2015) ‘MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking’, pp. 1–15. Available at: <http://arxiv.org/abs/1504.01942>.

Lin, T. Y. *et al.* (2014) ‘Microsoft COCO: Common objects in context’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8693 LNCS(PART 5), pp. 740–755. doi: 10.1007/978-3-319-10602-1_48.

Milan, A. *et al.* (2016) ‘MOT16: A Benchmark for Multi-Object Tracking’, pp. 1–12. Available at: <http://arxiv.org/abs/1603.00831>.

Redmon, J. *et al.* (2016) ‘You only look once: Unified, real-time object detection’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 779–788. doi: 10.1109/CVPR.2016.91.

Ren, S. *et al.* (2017) ‘Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6), pp. 1137–1149.

doi: 10.1109/TPAMI.2016.2577031.

Riedmiller, M. and Braun, H. (1993) ‘Direct adaptive method for faster backpropagation learning: The RPROP algorithm’, *1993 IEEE International Conference on Neural Networks*, pp. 586–591. doi: 10.1109/icnn.1993.298623.

Ristani, E. *et al.* (2016) ‘Performance measures and a data set for multi-target, multi-camera tracking’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9914 LNCS(c), pp. 17–35. doi: 10.1007/978-3-319-48881-3_2.

Shao, S. *et al.* (2018) ‘CrowdHuman: A benchmark for detecting human in a crowd’, *arXiv*, pp. 1–9.

Tieleman, T., Hinton, G. and others (2012) ‘RMSProp’, *COURSERA: Neural networks for machine learning*, 4(2), pp. 26–31.

Wojke, N., Bewley, A. and Paulus, D. (2018) ‘Simple online and realtime tracking with a deep association metric’, *Proceedings - International Conference on Image Processing, ICIP*, 2017-Septe, pp. 3645–3649. doi: 10.1109/ICIP.2017.8296962.

Yang, H. *et al.* (2020) ‘Online multi-object tracking using KCF-based single-object tracker with occlusion analysis’, *Multimedia Systems*, 26(6), pp. 655–669. doi: 10.1007/s00530-020-00675-4.

Yu, F. *et al.* (2018) ‘Deep Layer Aggregation’, *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 2403–2412. doi: 10.1109/CVPR.2018.00255.

Zhang, Y. *et al.* (2020) ‘FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking’, pp. 1–13. Available at: <http://arxiv.org/abs/2004.01888>.

Zhou, X., Wang, D. and Krähenbühl, P. (2019) ‘Objects as points’, *arXiv*.