

# Otonom Araçlar için Derin Öğrenme Tabanlı, Gerçek Zamanlı Nesne Tespiti Deep Learning Based, Real-Time Object Detection for Autonomous Driving

Gamze Akyol\*, Alperen Kantarcı\*, Ali Eren Çelik\*, Abdullah Cihan Ak\*

Department of Computer Engineering

Istanbul Technical University

Istanbul, Turkey

{akyolga, kantarcia, celikali1, akab}@itu.edu.tr

**Özetçe** —Bilgisayarlı Görü alanındaki popülerliğini koruyan ve aktif araştırma konularından birisi, otonom araçlarda nesne tespiti problemidir. Nesne tespiti zor bir problem olmasından ötürü yüksek başarımlı çözümler çok hızlı çalışmamaktadır. Benzer şekilde gerçek zamanlı çözümler ise başarımdan ödün vermektedir. Ancak otonom sürüşün doğası gereği nesne tespit sistemlerinin gerçek zamanlı ve yüksek başarımlı performanslar sergilemeleri gerekmektedir. Bu çalışmada en başarılı nesne tespit mimarilerinden biri olan Tiny YOLOv3, klasik nesne takip yöntemlerinden biri olan Kalman Süzgeci ile birleştirilmiştir. Modelin hızından bir kayıp yaşamadan başarımlarını artıran, gerçek zamanlı ve küçük bir nesne tespiti sistemi önerilmiştir.

**Anahtar Kelimeler**—Bilgisayarlı Görü, Derin Öğrenme, Nesne Tespiti, Kalman Süzgeci, Otonom Sürüş.

**Abstract**—One of the active research topics that maintains its popularity in the field of Computer Vision is the problem of object detection in autonomous cars. Since object detection is a difficult problem, high performance solutions do not work very quickly. Similarly, real-time solutions make compromise on performance. However, due to the nature of autonomous driving, object detection systems must perform in real time and high performance. In this study, Tiny YOLOv3, one of the most successful object detection architectures, was combined with one of the classical object tracking methods, the Kalman filter. A small and real-time object detection system, which increases the model's accuracy without losing its speed, is proposed.

**Keywords**—Computer Vision, Deep Learning, Object Detection, Kalman Filter, Autonomous Driving.

## I. GİRİŞ

Otonom sürüş son yıllarda hem ticari hem de akademik çevrede popülerleşmiş, içinde birçok zorlu ve hala çözülmemiş problemler barındırmakta olan bir teknolojidir [3]. Bu zorlu problemlerden birisi sürüş sırasında nesne tespitidir ve hem kamera hem de LIDAR verileri kullanılarak çözülmeye

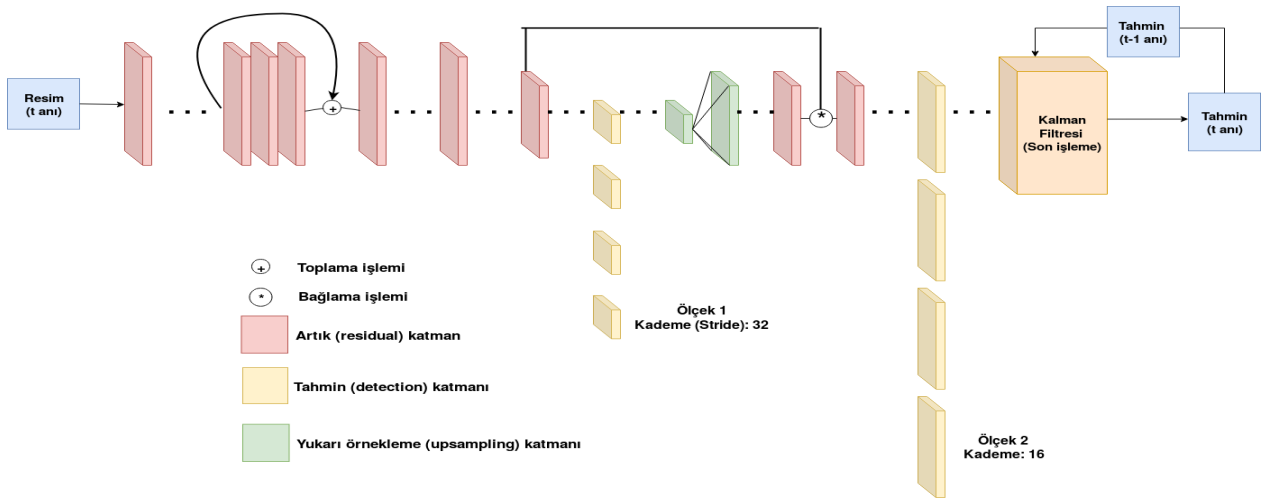


Şekil 1: Kalman Süzgeci [1] son işlemi, Tiny YOLOv3'ün [2] tespit edemediği nesneyi sonuca dahil ediyor - gerçek referans değer (kırmızı), Tiny YOLOv3 çıktısı (yeşil), Kalman çıktısı (mavi)

çalışılmaktadır. LIDAR etraftaki nesnelerin konumlarını üç boyutlu uzayda tespit edebilse de nesnelerin sınıflandırılması konusunda kullanılamamaktadır ve ayrıca yüksek ücretleri ile araç maliyetlerini artırmaktadır. Bu yüzden son yıllarda araştırmalar araç etrafına yerleştirilen kameralar ile nesne tespiti üzerine yoğunlaşmıştır.

Derin öğrenme modelleri, yeterli veri ile eğitildinde nesne tespiti probleminde yüksek başarılar elde edebilmektedir. Günümüzde otonom sürüş için kullanılan veri kümeleri daha çok yurt dışında yapılan sürüşlerde kaydedilmiştir. Bu popüler veri kümelerine Almanya'dan KITTI [4], Amerika'dan nuScenes

\*Yazarlar eşit katkıda bulunmuştur



Şekil 2: Model Mimarisi

[5] ve İngiltere'den Oxford Robotcar [6] örnek verilebilir. Her ülkenin yol yapısı, trafik akışı ve araç trafiği gibi sürüşü etkileyen özellikler farklı olduğu için modellerin eğitildiği verilerin hangi ülkeden alındığı ve modelin hangi ülkede kullanılacağı da model başarımı için önemlidir.

Otonom araçlara yerleştirilen işlemci ve grafik işlemci birimlerinin veri işleme gücü sınırlı olmasına rağmen derin öğrenme modellerinin başarımlarından dolayı bu modeller araçlarda kullanılmaktadır. Araç sürüşünün doğası gereği gerçek zamanlı çalışması gereken bu modeller kapasitesi azaltılarak hızlandırılmakta fakat başarımlarından taviz verilmektedir. Bu yüzden son yıllarda derin öğrenme modelleri ile uyumlu çalışabilecek ve modellerin hızını etkilemeden başarımlarını artıracak klasik sinyal işleme ve bilgisayarlı görü metotları bu modellerle beraber kullanılmaya başlanmıştır.

Bu çalışmada, en hızlı çalışan nesne tespit mimarilerinden biri olan Tiny YOLOv3 [2] mimarisi, İstanbul yollarından yolculuk görüntülerini ve görüntülerdeki nesnelerin etiketlerini içeren özel bir veri kümesi ile eğitilmiştir. Nesne takip metotlarından Kalman Süzgeciyle [1] birleştirilerek orijinal mimari hızının korunup başarımın artırılması sağlanmıştır.

Çalışmanın ikinci bölümünde literatürdeki eski çalışmalar tanıtılmıştır. Üçüncü bölümde önerilen sistemin ayrıntıları anlatılmıştır. Dördüncü bölümde deneysel sonuçlar sunulmuş ve son olarak beşinci bölümde çalışmanın özetine yer verilmiştir.

## II. LİTERATÜR ÖZETİ

YOLOv3 [2], gerçek zamanlı çalışabilen ve başarımı yüksek bir nesne tespit sistemidir. İlk olarak ImageNet [7] veri kümesinde sınıflandırma üzerine eğitilip daha sonra COCO [8] veri kümesinde nesne tespit görevine adapte edilmiştir. Bağlamalı (concatenation), atlama bağlantılı (skip connection) aynı zamanda yukarı örnekleme (upsample) yöntemlerini kullanarak tespit işlemini 3 farklı ölçekte yapan YOLOv3, literatürdeki güncel olan yüksek başarımli modellere göre ve yaygın olarak kullanılmaktadır. Tiny YOLOv3 bu sistemin küçük sürümü olup bir miktar başarım kaybı ile hızını yaklaşık 10 katına çıkarmaktadır. YOLOv3 ile Tiny sürümünün COCO [8]

TABLO I: YOLOv3 ile Tiny Sürümünün Karşılaştırılması

Modeller	mAP	FLOPS	FPS
YOLOv3-608	57.9	140.69 Milyar	20
YOLOv3-tiny	33.1	5.56 Milyar	220

veri kümesi üzerindeki performansları karşılaştırılması Tablo I'deki gibidir.

Otonom sürüşte nesne tespitine yönelik başka bir çalışma olan SqueezeDet [3], KITTI [4] veri kümesi üzerinde güncel olan en iyi nesne tespit sistemi olmuştur. SqueezeDet çalışmasında, var olan YOLO mimarisindeki birçok evrimsel katman çıkarılıp yerine özellik çıkarma, sınırlayıcı kutu dönüşümü (bounding box transformation) gibi farklı işlemlere sahip katmanlar eklenerek ve sonuçta boyut olarak küçük ve aynı zamanda başarım olarak yüksek bir sistem elde edilmiştir. Bu çalışmada ise YOLO mimarisi değiştirilmeden başarım doğrudan artırılmıştır.

Kalman Süzgeci [1] sürecin yeni durumunu tahmin etmek için kullanılan bir dizi matematiksel denklemlerdir. Bu denklemler, sonraki durumun tahmini ve durumun gözlemi ile tahminin düzeltilmesi olmak üzere iki parçadan oluşur. Derin öğrenme yöntemleri ile Kalman Süzgecinin kullanımında iki yaklaşım bulunmaktadır. Birinci yaklaşımda [9], [10] Kalman Süzgeci derin öğrenme katmanları olarak gerçekleştirilmiştir. İkinci yaklaşımda [11] Kalman Süzgeci bir derin öğrenme modelinin son işleme adımı olarak kullanılmıştır. Yang ve diğerleri [11] Kalman Süzgeci son işlemesi ile yaya tespiti başarısında derin öğrenme modelinin başarısını artırmışlardır. Bu çalışma kameranın sabit olduğu bir veri kümesi kullanmıştır ve gerçekçi bir otonom sürüşü temsil edememektedir.

## III. YÖNTEM

Hızlı ve aynı zamanda başarısı yüksek bir sistem elde etmek için farklı yaklaşımlar üzerine çalışılmıştır. Daha önce bahsedilen sebeplerden ötürü mimari olarak Tiny YOLOv3 kullanılmıştır. YOLOv3, COCO veri kümesi üzerine eğitildiği için 80 sınıf çıktısı vermektedir. Bu çalışmada kullanılan veri kümesi otonom sürüş için özelleştirilmiş olduğundan dolayı sadece

6 farklı sınıf içermektedir. Bu sebeple model 6 sınıf arasında nesne tespiti yapabilecek şekilde değiştirilmiştir. Bu bölümde ilk olarak kullanılan veri kümesi açıklanıp, daha sonra modeli son haline getirirken yapılan değişiklikler ayrıntılı bir biçimde açıklanacaktır.

#### A. Veri Kümesi

Çalışmada kullanılan veri kümesi, Eatron Technologies şirketi tarafından sağlanmıştır. Aracın ön camına yerleştirilmiş bir kamera ile farklı gün ve zamanlarda İstanbul içinde yapılmış 10 farklı yolculuk görüntülerinden oluşmaktadır. Önerilen sistemin doğruluğu kanıtlamak için bu veri kümesinden 2 yolculuk sına kümesi olarak ayrılmıştır ve model eğitimi kalan 8 yolculuk ile gerçekleştirilmiştir. Veri kümesinde toplam 11793 görüntü bulunmaktadır. Bunların 10471 tanesi eğitim, 1322 tanesi ise sına kümesindedir. Yolculuk boyunca 6 farklı nesnenin (insan, araba, kamyon, otobüs, motosiklet, bisiklet) çerçeve içindeki piksel konumları elle etiketlenmiştir. Farklı hava, ısklandırma ve ortam koşullarında kaydedilmiş yolculukların bulunduğu veri kümesi, otonom araç teknolojisinin Türkiye’de daha başarılı hale gelmesi için önem arz etmektedir.

#### B. Örneklem Katmanlarını Evrişim Katmanları ile Değiş-tirme

Denenen yöntemlerden biri, Tiny YOLOv3 mimarisinde bulunan ve kademesi (stride) iki olan örneklem (pooling) katmanlarının yerine kademesi iki olan evrişim katmanlarının eklenmesi ile yöntem geliştirilmeye çalışıldı. Bu sayede örneklem esnasındaki veri kaybı sorununun en aza indirilmesi hedeflenmiştir.

Model değiştirildiğinden, COCO veri kümesi üzerinde eğitilmiş ağırlıklar (pretrained) kullanılamamıştır. Değiştirilmemiş olan katmanlardaki ağırlıklar önceden eğitilmiş olan ağırlıklardan kopyalanmıştır. Eğitim sırasında öğrenme oranı, COCO veri kümesi üzerinde eğitilmiş ağırlıklarda düşük, yeni oluşturulan katmanlardaki ağırlıklarda ise daha yüksek tutulmuştur. Değiştirdiğimiz modelin eğitilmiş ağırlıklarının kullanılabilir bir sonuç çıkarmaması sebebiyle bu model kullanılmamıştır.

#### C. İnce Ayar Yöntemleri

Tiny YOLOv3 mimarisinde ince ayarın hangi katmanlar üzerinde yapılacağı alttaki seçenekler arasında deneysel olarak belirlenmiştir.

- Yalnızca çıktı katmanlarına ince ayar yapma
- Çıktı katmanlarına ve öncesindeki birer evrişimsel katmana ince ayar yapma (toplamda 4 katman)
- Çıktı katmanlarına ve öncesindeki ikişer evrişimsel katmana ince ayar yapma (toplamda 6 katman)
- Tüm mimariye ince ayar yapma

Her seçeneği eğitirken yitim ve başarımların sına malarının sonucunda ince ayar tüm katmanlara yapılmıştır.

#### D. Ön İşleme

Veri kümesinde gece ve gündüz, açık ve kapalı hava gibi değişik ortam koşullarına sahip görüntüler bulunmaktadır (Şekil 1). Çıkarım sırasında bu durumdan en az etkilenmek

için renk seçirme (jitter) ön işlemi kullanılmıştır. Bazı görsellerde renk kaymaları olduğundan ton parametreleri veriye uygun olarak seçilmiştir. Kullanılan renk seçirme parametreleri şu şekildedir: Parlaklık=0.3, karışıklık=0.2, doygunluk=0.4, ton=0.05.

#### E. Kalman Süzgeci

Kalman Süzgeci son işleme kısmı, yukarıda anlatılan Tiny YOLOv3 modelini Şekil 2’deki gibi güncellenmiştir. Kalman Süzgeci son işlemesi Tiny YOLOv3 modeli çıktılarını gözlem olarak kullanıp tespit sonuçlarını düzenlemektedir. Kullandığımız Kalman son işleme algoritması Tablo II’de verilmiştir. Kullandığımız Kalman Süzgecinin nesneler hem konumu hem de boyutu temsil edebilmesi için sınırlayıcı kutunun sağ üst ve sol alt konumları ile temsil edilmiştir. Giriş vektörü bu konumları içerir. Sistem matrisi bu konumların hareket süreci ile aynı nesne için tespit edilmiş konum değişikliklerini içerir. Ölçüm matrisi YOLOv3 modeli çıktısı sınırlayıcı kutunun sağ üst ve sol alt konumlarını içerir ve Kalman tahminini düzeltir. Bunun sonucunda Kalman Süzgecini son işleme olarak kullanıp nesne konumu güncellenir. Sistem ve ölçümdeki gürültü bütün girişlerin birbirini etkilediği bir matrisle temsil edilmiştir. Etkinin 5 pikselden fazla olduğu ve en iyi sonucu veren gürültü şiddeti kullanılmıştır.

Bu algoritma bazı ön kabullerle çalışmaktadır ve şartlar sağlanmadığında Kalman Süzgeci son işlemesi atlanır. Kalman Süzgecinin bir düzeltme yapabilmesi için arka arkaya olan iki görüntüde birbirini takip eden iki nesne tespit edilmesi gerekmektedir. İki görüntüdeki iki nesnenin aynı nesne olarak kabul edilebilmesi için tespitlerin kesişim bölgelerinin birleşim bölgesine oranının eşik değerinin üzerinde olması gerekmektedir. Eşik değeri bu çalışmada 0.3 olarak seçilmiştir. Bazı durumlarda birden fazla tespit bir geçmiş tespitle aynı bölgeyi paylaşabilir. Bu durumda kesişim bölgesinin birleşim bölgesine oranı en yüksek olan tespitler ikilisi eşleştirilmiştir. Bir tespit ile eşleşen eski tespit bulunamazsa yeni tespit olarak kabul edilir ve sadece Tiny YOLOv3 tahmini kullanılır. Önceki tespitler ile eşleşen yeni bir tespit bulunamazsa, derin öğrenme modelinin tespit edemediği kabul edilir ve sadece Kalman Süzgecinin tahmin kısmı ile nesne tespiti yapılır.

TABLO II: Kalman Süzgeci Son İşlemesi Algoritması

Sonuç: Sınırlayıcı_Kutu
Tiny YOLOv3 tahminlerini al(Tahmin)
Önceki Tiny YOLOv3 tahminlerini al
bütün tahmin Sınırlayıcı Kutuları için:
$X = AX$
$P = APA^T + Q$
En iyi IOU ile:
$K = PH^T(HPH^T + R)^{-1}$
$X = X + K(Tahmin - HX)$
Sınırlayıcı_Kutu = X
eğer yeni nesne:
Sınırlayıcı_Kutu = Tahmin

#### IV. DENEYSEL SONUÇLAR

Geliştirilen modelin sonuçları, Kalman Süzgeciyle ve Kalman Süzgeçsiz elde edilen ortalama hassasiyet (mean average precision, mAP) ve saniye başına kare (frame per second, FPS)

değerleri, Tablo III'te gösterilmiştir. Sonuçlarda görüldüğü üzere Kalman Süzgeciyle görüntü işleme hızında küçük bir azalma ve başarımda artış olmuştur.

TABLO III: Eğitim Sonuçları

Veri Kümesi	Orijinal FPS	Kalman FPS	Orijinal mAP	Kalman mAP
0	145.32	142.22	44.326	44.332
1	151.12	146.885	34.726	34.726
2	143.8	138.8	44.197	44.27
3	112.26	112.51	54.275	54.314
5	121.66	114.71	49.87	49.85
6	142.45	138.78	49.66	49.97
7	158.6	163.1	35.29	35.29
9	136.61	131.71	55.83	55.92
<b>Ortalama</b>	<b>138.978</b>	<b>136.089</b>	<b>46.022</b>	<b>46.084</b>

Eğitim başarımında (mAP) sayısal olarak büyük bir artış gözlemlenmemiş olsa da Tiny YOLOv3'ün tespit edemediği bazı nesneler Kalman Süzgeci son işlemesi ile tespit edilebilmiştir, bir örnek Şekil 1'de görülebilir. İşleme hızındaki düşüşün ihmal edilebilecek kadar az olmasına rağmen başarıda kayda değer gelişmeler gözlemlenmiştir.

Kalman Süzgecinin başarıma olan etkisi incelendiğinde olumlu ve olumsuz sonuçlar bir arada gözlemlenmiştir. Kalman Süzgecinin başarıyı ciddi derecede artırdığı durumların olmasına karşın tespit edilmiş nesnenin konumunu yanlış konuma doğru kaydırıldığı durumlar da gözlemlenmiştir. Veri kümesinin canlı kamera görüntüsünden ziyade örneklenmiş sürümü üzerinde çalışıldığından dolayı buna benzer durumlarla karşılaşılması beklenmiştir. Bu veri kümesindeki gibi aracın sabit olmayacağı göz önünde bulundurulduğunda saniyede en az 15 gibi yüksek bir kare hızının, Kalman Süzgecinin başarıları için esas bir gereklilik olduğu ortaya çıkmıştır.

TABLO IV: Sınama Sonuçları

Veri Kümesi	Orijinal FPS	Kalman FPS	Orijinal mAP	Kalman mAP
4	120.3	115.52	42.583	42.701
8	156.4	150.8	30.77	31
<b>Ortalama</b>	<b>138.35</b>	<b>133.16</b>	<b>36.6765</b>	<b>36.8505</b>

Sayısal olarak bakıldığında Kalman Süzgeci son işlemi, başarıyı tespit başına en fazla 0.4 yükseltirken en çok 0.6 puan düşürmüştür. Genel ortalama ise Tablo IV'te görüldüğü üzere başarımda 0.18'lik bir artış elde edilmiştir. Başarımın %40 olduğu bir durumda bu artış miktarı kayda değer bir gelişme olarak görülebilir. Sistemin örnek bir çıktısı Şekil 3'te görülebilir.

Yang vd. [11] Kalman Süzgeci son işlemesinin uzaktaki nesneler için başarıyı artırdığını öne sürmüştür. Ancak uzaktaki nesneler üzerinde Kalman son işlemesinden başarılı sonuçlar alınmadığı görülmüştür. Bunun sebebi, bu çalışmada uzak nesnelerin sınırlayıcı kutularının diğer çalışmadaki uzak nesnelerin sınırlayıcı kutularından çok daha küçük olmasıdır. Bunun dışında olarak saniye başına kare sayısı yüksek olan verilerde Kalman Süzgecinin modelin başarısını artıracak kanısına varılmıştır.

## V. SONUÇ

Bu çalışmada, otonom araçlarda nesne tespiti için Tiny YOLOv3 modeline çeşitli yenilikler getirerek daha yüksek ortalama hassasiyet (mAP) ve daha yüksek saniye başına kare (FPS) elde edilmiştir. Tiny YOLOv3 modeli yüksek hızda



Şekil 3: Sistemin zor bir görüntüdeki çıktısı

sonuç vermesine rağmen YOLOv3 modelinden daha düşük ortalama hassasiyete sahiptir ve ortalama hassasiyet değerini arttırmak için modele Kalman Süzgeci son işlemesi uygulanmıştır. Bunun sonucunda ortalama hassasiyet değerinde bir miktar artış görülmüştür. Gelecek çalışma olarak, saniye başına kare değerindeki kaybın, Kalman Süzgeci son işlemesinin geliştirilmesiyle azaltılması amaçlanmıştır. Problem doğrusal olmadığı için karşılaşılan problemlerin Kokusuz Kalman Süzgeçleri ile çözümü aranacaktır.

## TEŞEKKÜR

Bu çalışmada kullanılan veri, Eatron Technologies şirketi tarafından sağlanmıştır.

## KAYNAKLAR

- [1] E. Kalman, R. "A new approach to linear filtering and prediction problem,," in *Trans. ASME, J. Basic Engr., Ser. D.*, 1960.
- [2] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
- [3] B. Wu, F. Iandola, P. H. Jin, and K. Keutzer, "Squeezednet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *CVPR Workshops*, 2017.
- [4] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.
- [5] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.
- [6] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *CVPR. IEEE*, 2009, pp. 248–255.
- [8] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 740–755.
- [9] K. Kang, O. Ouyang, H. Li, and X. Wang, "Object detection from video tubelets with convolutions neural network,," in *in Proceedings of CVPR*, 2016.
- [10] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," 2015.
- [11] F. Yang, H. Chen, J. Li, F. Li, L. Wang, and X. Yan, "Single shot multibox detector with kalman filter for online pedestrian detection in video," in *IEEE Access, Volume 7*, 2018.