ANKARA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

YÜKSEK LİSANS TEZİ

ÖN PLAN BÖLÜTLENMESİNDE DENETİMLİ ÇOK ÖLÇEKLİ KONVOLÜSYONEL SİNİR AĞLARI YAKLAŞIMININ KULLANIMI

Long Ang LİM

BİLGİSAYAR MÜHENDİSLİĞİ ANABİLİM DALI

ANKARA
2018

TEZ ONAYI

Long Ang LİM tarafından hazırlanan **"Ön Plan Bölütlenmesinde Denetimli Çok Ölçekli Konvolüsyonel Sinir Ağları Yaklaşımının Kullanımı"** adlı tez çalışması 15/08/2018 tarihinde aşağıdaki jüri tarafından oy birliği ile Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Anabilim Dalı'nda **YÜKSEK LİSANS TEZİ** olarak kabul edilmiştir.

Danışman : Dr. Öğr. Üyesi Hacer YALIM KELEŞ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Jüri Üyeleri :

Başkan : Doç. Dr. Süleyman TOSUN
Hacettepe Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Doç. Dr. Üyesi Semra GÜNDÜÇ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Üye : Dr. Öğr. Üyesi Hacer YALIM KELEŞ
Ankara Üniversitesi Bilgisayar Mühendisliği Anabilim Dalı

Yukarıdaki sonucu onaylarım.

Prof. Dr. Atila YETİŞEMİYEN
Enstitü Müdürü

ETİK

Ankara Üniversitesi Fen Bilimleri Enstitüsü tez yazım kurallarına uygun olarak hazırladığım bu tez içindeki bütün bilgilerin doğru ve tam olduğunu, bilgilerin üretilmesi aşamasında bilimsel etiğe uygun davrandığımı, yararlandığım bütün kaynakları atıf yaparak belirttiğimi beyan ederim.

15/08/2018

Long Ang LİM

ÖZET

Yüksek Lisans Tezi

ÖN PLAN BÖLÜTLENMESİNDE DENETİMLİ ÇOK ÖLÇEKLİ KONVOLÜSYONEL SİNİR AĞLARI YAKLAŞIMININ KULLANIMI

Long Ang LİM

Ankara Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Hacer YALIM KELEŞ

Ön plan segmentasyonu alanında birkaç yöntem önerilmiştir; ancak, bu yöntemler aydınlatma değişiklikleri, arka plan veya kamera hareketi, kamuflaj etkisi, gölge gibi çeşitli zor senaryolarla başa çıkamamaktadır. Bu sorunları çözmek amacıyla, az sayıda eğitim örneğini kullanarak uçtan uca eğitilebilen üç farklı güçlü enkoder-dekoder tipi derin sinir ağı önerilmiştir. İlk olarak bu çalışmada, bir görüntüyü, çoklu ölçekte öznitelik uzayına dönüştürmek için kodlayıcı kısımında üçlü bir yapı altında önceden eğitilmiş bir konvolüsyonel (evrişimsel) ağı (VGG-16 Net) veriye uyarlanarak kullanılmıştır; öznitelik uzayındaki ifadeden görüntü uzayına projeksiyonu öğrenmek için, dekoder kısmında, dönüştürülmüş bir konvolüsyonel sinir ağı kullanılmıştır. İkinci olarak, çoklu ölçek özniteliklerini çıkarmak için tek bir giriş kodlayıcının üstüne takılabilen bir Feature Pooling Module (FPM) önerilmiştir ve görüntü uzayına projeksiyonu öğrenmek için bu özniteliklerin üstüne aynı dekoder yerleştirilmiştir. Üçüncü olarak FPM modülün yapısına öznitelikler füzyonu eklenerek bu modül genişletilmiştir ve sonuç olarak kamera hareketlerine karşı gürbüz bir modül oluşturulmuştur. Daha ileri performans iyileştirmesi için genişletilmiş FPM'in üzerine yeni bir dekoder ağı önerilmiştir. Önerilen FgSegNet_M, FgSegNet_S ve FgSegNet_v2 olarak adlandırılan yöntemlerle geliştirilmiş modeller, Change Detection 2014 Challenge (changedetection.net)'de, sırasıyla 0.9770, 0.9804 ve 0.9847 ortalama F-Measure ile, mevcut tüm yöntemlerinden daha iyi performansla çalıştırmaktadır. Modellerimiz SBI2015 ve UCSD Background Subtraction veri setlerinde de değerlendirilmiştir. Tez çalışması kapsamında, yukarıda özetlenen çalışmalara ek olarak ön plan nesnelerinin segmantasyonu alanında yama-tabanlı (patch-wise) öğrenme hakkında yürüttgümüz çalışmalar da sunulmaktadır. Ek olarak, ön plan segmentasyon kapsamında geliştirdiğimiz yöntemlerin anlamsal (semantik) segmentasyon alanındaki etkinliğini değerlendirmek için yürüttüğümüz iki yöntem çalışması da detaylı olarak tartışılmaktadır.

Ağustos 2018, 88 sayfa

Anahtar Kelimeler: Ön plan segmentasyonu, arka plan çıkarması, konvolüsyonel sinir ağları, derin öğrenme, semantik segmentasyonu, video gözetim sistemi

ABSTRACT

Master Thesis

UTILIZATION OF SUPERVISED MULTI-SCALE CONVOLUTIONAL NEURAL NETWORKS APPROACH FOR FOREGROUND SEGMENTATION

Long Ang LIM

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

Supervisor: Asst. Prof. Dr. Hacer YALIM KELEŞ

Several methods have been proposed in foreground segmentation domain. However, they lack the ability of handling various difficult scenarios such as illumination changes, background or camera motion, camouflage effect, shadow etc. To address these issues, we propose three different robust encoder-decoder type deep neural networks that can be trained end-to-end using only a few training examples; first, we adapt a pre-trained convolutional network, i.e. VGG-16 Net, under a triplet framework in the encoder part to embed an image in multiple scales into the feature space and use a transposed convolutional network in the decoder part to learn a mapping from feature space to image space. Second, we propose a Feature Pooling Module (FPM) that can be plugged on top of a single input encoder to extract multiple scale features and the same decoder is embedded on top of these features to learn upsampling to image space. Third, we extend the FPM module by introducing features fusion inside this module, resulting in a robust module against camera motions and we further propose a novel decoder network on top of the extended FPM for further performance improvement. In order to evaluate our models, we entered the Change Detection 2014 Challenge (changedetection.net) and our methods called FgSegNet_M, FgSegNet_S and FgSegNet_v2 outperformed all the existing state-of-the-art methods by an average F-Measure of 0.9770, 0.9804 and 0.9847, respectively. We also evaluate our models on SBI2015 and UCSD Background Subtraction datasets. In the context of this study, we also provide a comprehensive study about patch-wise learning in foreground segmentation domain. Furthermore, in order to evaluate the methods that we developed in the context of the foreground segmentation problem in semantic segmentation domain, we present two semantic segmentation method studies in detail.

August 2018, 88 pages

Key Words: Foreground object segmentation, background subtraction, convolutional neural networks, deep learning, semantic segmentation, video surveillance

TEŞEKKÜR

İlk olarak, bu araştırma sayesinde beni araştırma dünyasına tanıtan, tez çalışmanın her aşamasında yardım eden, destek sağlayan ve çalışmalarım sırasında beni teşvik edip motivasyon sağlayan danışmanım Sayın Dr. Öğr. Üyesi Hacer YALIM KELEŞ'e çok teşekkür ederim.

Aynı zamanda tez komitemize zamanları ve özverileri için çok teşekkür ederim.

Çalışmalarım sırasında hem burada çalışma fırsatı veren hem de maddi destek sağlayan Türkiye bursları komitesine çok müteşekkirim.

Tezimin Türkçe dilbilgisi düzeltilmesinde yardımcı olan meslektaşım Anıl Osman Tur' a teşekkür ederim.

Son olarak, çocukluğumdan beri tüm destek, ilham, sevgi ve yüksek lisans bitirene kadar sürdürme kararımı destekledikleri için aileme çok teşekkür etmek isterim.

Long Ang LİM
Ankara, Ağustos 2018

İÇİNDEKİLER

SİMGELER DİZİNİ

σ	Sigma
δ	Delta
λ	Lamda

Kısaltmalar

ANN	Artificial Neural Network
BN	BatchNormalization
CNN	Convolutional Neural Network
CDnet2014	Change Detection net 2014
CRF	Conditional Random Field
CamVid	Cambridge-driving Video
DenseNet	Densely Network
ETS	Entire Set Training
Enet	Efficient Network
FCN	Fully Convolutional Network
FgSegNet	Foreground Segmentation Network
FN	False Negative
FP	False Positive
FPM	Feature Pooling Module
GAP	Global Average Pooling
GMM	Gaussian Mixture Model
IN	Instance Normalization
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
K	Kilo veya Thousands
LBP	Local Binary Pattern
LBSP	Local Binary String Pattern
M	Million
M-FPM	Modified Feature Pooling Module
MCC	Matthews Correlation Coefficient
PAWCS	Pixel-based Adaptive Word Consensus Segmenter
PWC	Percentage of Wrong Classification
RGB	Red-Green-Blue
RST	Randome Subset Training
ReLU	Rectified Linear Unit
ResNet	Residual Network
SD	Spatial Dropout
SegNet	Segmentation Network
SuBSENSE	Self-Balanced SENsitivity SEgmenter
TCNN	Transposed Convolutional Neural Network
TN	True Negative
TP	True Positive
VGG	Visual Geometry Group
ViBe	Visual Background Extractor

ŞEKİLLER DİZİNİ

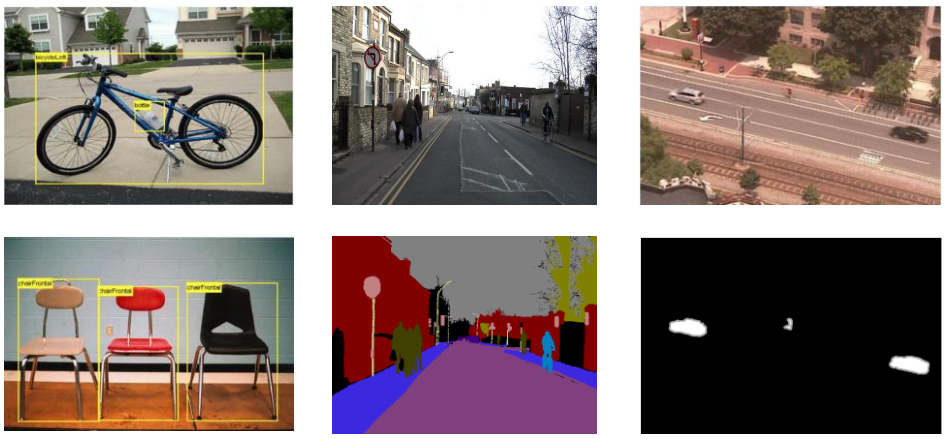
ÇİZELGELER DİZİNİ

1. GİRİŞ

1.1 Genel Bakış

İnsan, etrafındaki dünyayı görsel olarak algılamak için gözlerini ve beyinini kullanmaktadır. Ancak, bilgisayarla görmenin amacı dijital görüntüler veya video dizileri gibi gerçek dünya verilerinden yüksek düzeyde bir anlayış elde ederek insan görmesini taklit etmektedir. Gerçek dünya verilerinden herhangi bir karar vermeden önce, bilgisayarın ana üç bileşenle işlenmesi gerekmektedir; görüntü edinmesi, görüntü işlemesi, görüntü analizi ve anlamadır. Bilgisayar vizyonu, görüntülerden faydalı bilgiler elde etmek için algoritma oluşturması ile ilgilidir.

Tanıma (örneğin: nesne algılaması, nesne segmentasyonu), hareket analizi, görüntü restorasyonu ve sahne rekonstrüksiyonu gibi birçok tipik bilgisayar vizyonu görevi vardır. Günümüzde görüntü tanıma görevi en aktif araştırma alanlarından biridir ve çoğu araştırmacı yıllarca kapsamlı bir şekilde incelenmiştir. Şekil 1.1'de üç görüntü tanıma alt grubunu göstermektedir; bunlar nesne sınıflandırması, semantik segmentasyon ve ön plan segmentasyonudur.



(a) (b) (c)

Şekil 1.1 Nesne tanıma görevleri (Görüntüler Everingham vd. (2015), Brostow vd. (2009) ve Wang vd. (2014)'ten alınmıştır)

a. nesne algılaması; b. semantik segmentasyon; c. ön plan segmentasyonu

- ***Nesne Algılaması***: bir görüntüdeki her nesneye sınıf etiketleri ve sınırlayıcı kutular (bounding box) vermekle ilgilidir.
- ***Semantik Segmentasyon***: bir görüntüyü kaldırım, yol, yaya, gökyüzü, bina veya ağaç gibi semantik kategorilerle ilişkilendiren farklı bölgelere ayırmakla ilgilenmektedir.
- ***Ön Plan Segmentasyonu***: video dizilerinden ön plan nesnelerinin (veya hareketli nesnelerin) çıkarılmasıyla ilgilidir. Ön plan segmentasyonu, arka plan çıkarması veya hareketli nesneler segmentasyonu olarak adlandırılmaktadır. Bu tez boyunca bahsedilen terimler birbirinin yerine kullanılmaktadır.

Tezin ilk bölümünde ana tartışma olarak ön plan segmentasyonu derinlemesine tartışılacaktır. İkinci bölümde ise ön plan segmentasyonu algoritmasında yapılan bazı değişikliklerle tasarlanan yeni bir mimari üzerinden semantik segmentasyon yaklaşımı tartışılacaktır.

Bilgisayar vizyonunda ön plan nesne segmentasyonu (veya arka plan çıkarması) en aktif araştırma alanlarından biridir. Arka plan çıkarma algoritmaları çeşitli uygulamalarda kullanılmaktadır. Güvenlik ve gözetim en dikkate değer uygulamalardan biridir. Bu durumda herhangi bir sahnede veya belirli bir sahnede hareketli nesneleri belirlenmek için yazılım algoritması kameranın donanımına yerleştirilmektedir. Güvenlik kameraları genellikle trafiği gözlemek için yol üzerinde belirli noktalara, suç veya alışılmadık faaliyetleri tespit edilmek için binaların içlerine yerleştirilmektedir. Video dizilerindeki hareketli nesneleri hassas bir şekilde segmentlere ayırmak için sağlam bir algoritma gereklidir.

Birçok konvansiyonel arka plan çıkarma algoritması benzer adımlar paylaşmaktadır:

- ***Arka Plan İlklendirilmesi***: Bu adımda bir ilk arka plan modeli kurulmaktadır (Şekil 1.2.a). Bu arka plan modeli, video dizisinden gelen çerçeveler ile herhangi bir karşılaştırmayı gerçekleştirmek için kullanılmaktadır.
- ***Ön Plan Segmentasyonu***: Bu adımda video dizisinden gelen çerçeveler ile arka plan modeli arasındaki karşılaştırmalar yapılmaktadır (Şekil 1.2.b). Bu işlemin sonuçları, ön plan pikselleri ve arka plan piksellerini içeren ön plan maskeleridir

(Şekil 1.2.c).

- ***Arka Plan Güncellemesi***: Bu adımda arka plandaki değişikliklere uyum sağlamak için arka plan modelinin güncellenmesi gerekmektedir.

(a) (b) (c)

Şekil 1.2 CDnet2014 veri kümesinden bazı çerçeveler (Wang vd. 2014)

a. medyan filtresi kullanarak oluşturulan arka plan modeli; b. video dizisinden gelen çerçeveler; c. ön plan maskesi

1.2 Ön Plan Segmentasyon Problem Tanımı

Güçlü bir arka plan çıkarma algoritmasını geliştirmekte birçok zorluk vardır. Bunlar kademeli veya ani aydınlatma değişiklikleri, ön plan nesnelerinden kaynaklanan gölgeler, dinamik arka plan hareketi (sallanan ağaç(lar), yağmur, kar veya hava türbülansı), kamera hareketi (kamera titreşimi, kamera kaydırma-devirme-yakınlaştırma), kamuflaj efekti ya da anlaması zor bölgelerdir (yani ön plan pikselleri ve arka plan pikselleri arasındaki benzerlik). Güçlü algoritmalar bu zor durumlara karşı sağlam olmalı ve yüksek kaliteli segmantasyon maskeleri oluşturabilmelidir.

Arka plan çıkarma problemini çözmek için birçok konvansiyonel bilgisayar görüşü yaklaşımı önerilmiştir. Ancak, önerilen metotlar sadece bazı belirli durum türleri üzerinde iyi çalışır ama tüm durumlarda işleme yeteneği eksikliktir. Trafik izleme ve video gözetimi alanlarını düşünelim: güçlü bir arka plan çıkarma algoritması, kameranın nereye veya hangi hava koşuluyla yerleşeceğine bağlı kalmadan çalışmalı, hareketli nesneleri sağlam bir şekilde izleyebilmeli ve bölütlemelidir. Üstelik, klasik metotların çoğu tamamen bir arka plan modelini oluşturmaya dayanmaktadır. Bu durumda eğer arka plan modeli yeterince sağlam değilse, istenmeyen yanlış tahminler yapılabilmektedir.

Bu alanda oluşturulmuş birçok yöntem yaygın yaklaşımlara dayanmaktadır; bulanık tabanlı yöntemler (Zhang ve Xu 2006, El baf vd. 2008), Gaussian kullanılan istatistiksel yöntemler (Stauffer ve Grimson 1999, Zivkovic ve Van Der Heijden 2006), renk ve doku özellikleri kullanılan istatistiksel yöntemler (Yao ve Odobez 2007), eigenvalue ve eigenvector kullanılan yöntemler (Oliver vd. 2000), ve parametrik olmayan yöntemler (Barnich ve Van Droogenbroeck 2011, Hofmann vd. 2012) gibi. Ancak, bu geleneksel yöntemler, zor senaryolara karşı güçlü değildir.

Son zamanlarda derin öğrenme yöntemleri, özellikle konvolüsyonel sinir ağları (CNN'ler), nesne tanıması (Krizhevsky vd. 2012, Zeiler ve Fergus 2014, Simonyan ve Zisserman 2014, Szegedy vd. 2015, He vd. 2016), sahne etiketlemesi ve semantik segmentasyonu (Farabet vd. 2013, Long vd. 2015, Pinheiro ve Collobert 2014), metin sınıflandırması ve resim manşetleme (Lai vd. 2015, Venugopalan vd. 2014, Vinyals vd. 2015, Xu vd. 2015, Karpathy ve Fei-Fei 2015) alanlarında popüler hale getirilmiş ve başarılı bir şekilde uygulanmıştır. CNN'ler, ön plan veya arka plan segmentasyonu da dahil olmak üzere çeşitli bilgisayarlı görme problemlerinde görüntülerden düşük, orta ve yüksek düzey öznitelik temsillerinin çıkarılmasında yararlı olduğu ortaya koyulmuştur.

1.3 Semantik Segmentasyon Problem Tanımı

Günümüde bilgisayar görüşü alanında semantik video segmentasyonu aktif bir araştırma alanıdır. Ön plan nesnelerini (veya hareketli nesneleri) bölütlenmekle ilgilenen ön plan nesnelerinin segmentasyonunun aksine semantik video segmentasyonu, video sekanslarındaki görüntü karelerinin çoklu bölgelere ayrılmasıyla ilgilidir. Sahnenin bölgelerine kaldırım, yol, yaya, gökyüzü, bina ve ağaç gibi semantik kategorilere atanan yol sahnesi içeren video dizilerini düşünelim. Semantik segmentasyon, bir görüntünün her pikselinin bir sınıf etiketiyle ilişkilendirildiği, çok sınıflı bir segmentasyon problemidir. Semantik segmentasyonun amacı imgeleri temsili anlamlı ve anlaşılması kolay bir şeye dönüştürmektir. Pixel-wise semantik segmentasyonu otonom sürüş, tıbbi görüntüler ve robot navigasyonu vb. gibi çeşitli uygulamalarda faydalıdır.

1.4 Araştırma Amacı

Bu tezin ana amacı, CNN'ler görsel verilerden gizli öznitelik temsilini öğrenbildikleri için, CNN'leri kullanarak ön plan nesnelerinin segmentasyonu için sağlam derin öğrenmeye tabanlı bir metot önerisi getirmektir. Bu kapsamda üç farklı mimari tasarım önerisi kapsamlı olarak sunulmaktadır.

Ek olarak, ikincil amaç, benzer bir yaklaşımla semantik segmentasyon alanında bir çalışma gerçekleştirmektir. Bu kapsamda iki yöntem önerisi getirilmiştir: ilkinde mevcut ön plan segmentasyon ağı, semantik segmentasyon alt problemine uyarlanmıştır, ikincisinde, performansı daha da geliştirmek için yeni bir ağ mimarisi tasarlanmıştır.

Bu çalışmaların sonuçları, yapılacak akademik araştırmalar için değerli olacaktır ve iş süreçlerinde yöntemimizi kullanan tüm endüstrilere pratik bir araç sağlayacaktır.

1.5 Tez Katkısı

Bu tezde ön plan nesnelerinin segmentasyonu ve semantik segmentasyon için farklı katkılar içermektedir. Katkılarımız aşağıdaki gibidir:

- ***Yama-tabanlı (Patch-wise) ön plan segmentasyonu***: Bu alt problemin amacı, belirli bir video dizisindeki video çerçevelerinden her piksele ortalanmış olan parçaların (patches) çıkarılmasıdır. Bu parçalar, daha sonra merkezlenmiş pikselleri ön plan veya arka plan pikselleri olarak sınıflandırma için ağa girdi olarak gönderilmektedir. Bu alt problemde iki eğitim stratejisi sağlanmaktadır: bunlar rastgele alt grup eğitimi (Random Subset Training) ve tüm grup eğitimi (Entire Set Training).
- ***İmge-tabanlı (Image-wise) ön plan segmentasyonu***: Bu alt problemin amacı ham RGB görüntülerinden ön plan maskeleri oluşturmaktır. Bu durumda, tüm görüntüler doğrudan ağı beslenmektedir. Bu yöntem genellikle pikselden-piksele

yöntemi (pixel-to-pixel) olarak adlandırılmaktadır (Şekil 1.1.c). Bu alt problemde önerilen metotların mevcut tüm modern yöntemlerden daha iyi performans gösterdiği üç yöntem önerilmiştir.

- ***Semantik segmentasyon***: Bu problemde bir sahnedeki bütün piksellerin kendi semantik etiketleri vardır. Semantik segmentasyon, ikili sınıflandırma problemi (binary classification problem) olan ön plan segmentasyonunun aksine çok sınıflı sınıflandırma problemidir (Şekil 1.1.b). Bu alanda önceki bazı yöntemlerden daha iyi performans gösteren iki yöntem önerilmiştir.

2. ÖN PLAN SEGMENTASYONU

Bu bölümde ön plan segmentasyon problemi ile ilgili yürüttüğümüz çalışma detaylı biçimde sunulmaktadır. Bu kapsamda geliştirdiğimiz iki yaklaşımla ilgili detaylar, Alt bölüm 2.5 ve 2.6'da yama-tabanlı (patch-wise) öğrenmesi ve imge-tabanlı (image-wise) öğrenmesi başlıkları altında açıklanmaktadır. Gerçek dünya görüntüleri verildiğinde hedefimiz sahnelerden ilgilenilen ön plan nesnelerini çıkarmaktadır. Şekil 2.1, bu görevin başka bir görselleştirmesini sağlamaktadır.

Şekil 2.1 Ön plan nesneler segmentasyon probleminin görselleştirilmesi (Görüntüler Wang vd. (2014)'ten alınmıştır)

Deneylerimizde eğitim çerçeveleri olarak rastgele 25, 50 ve 200 çerçeve kullanılmıştır. Şu andan itibaren, 25-çerçeve deneyleri, 25 eğitim çerçevesi kullandığımız deneyleri, benzer şekilde, 50-çerçeve deneyleri, 50 eğitim çerçevesi kullandığımız deneyleri ve 200-çerçeve deneyleri, 200 eğitim çerçevesi kullandığımız deneyleri ifade etmek için kullanılacaktır.

2.1 Literatür İncelemesi

Son birkaç yılda ön plan nesnelerinin segmentasyonu probleminde çeşitli yöntemler önerilmiştir. Bu problem, bir görüntü dizisinden ön plan maskesinin belirlenmesi olarak da ifade edilebilmektedir; burada sahnede maskelenmiş bölgeler hareketli nesneler olarak adlandırılmaktadır. Belirli bir sahneden bir ön plan maskesini çıkarmak için o sahnenin ön plan bölgelerini belirlemek gerekir. Bunun için de bir görüntü dizisinin her bir karesinde kullanılabilecek sağlam ve esnek bir arka plan modeli oluşturulmalıdır. Bu

bölümde önerilmiş yöntemler kendi kategorilerine göre tartışılacaktır.

2.1.1 Temel yöntemler

Ön plan nesnelerinin segmentasyonunu gerçekleştirmeden önce arka plan modeli olarak hareketli nesnesi olmayan sabit bir görüntü kullanılabilmektedir. Her gelen video karesi için sabit arka plan modeli ile bu gelen çerçeve arasındaki mutlak fark hesaplanır. Bu yaklaşım *Durağan Çerçeve Farkı* (*Static Frame Difference)* olarak adlandırılmıştır (Piccardi 2004). Fakat, aydınlatma, yağmur, sallanan ağaçlar gibi arka planda değişiklikler meydana gelirse, bu yöntem başarısız olabilmektedir. Bu sorunun üstesinden gelmek için sabit çerçeveyi kullanmak yerine bir önceki çerçeve arka plan modeli olarak kullanılabilmektedir. Bu yaklaşım *Çerçeve Farkı* (*Frame Difference)* olarak adlandırılmıştır (Lipton vd. 1998). Ancak, nesneler bir sahnede uzun bir süre durup aniden hareket etmeye başlarsa, bu yöntem de başarısız olabilmektedir.

Rao ve Darwin (2012) tarafından *Kalman* filtresinin *Frame Difference*'den daha iyi performans gösterdiği iddia edilmiştir. Fakat, bu yöntem arka plan piksellerini ön plan pikselleri olarak sınıflandırmıştır. Lai ve Yung (1998) tarafından arka plan modelini tahmin etmek için *running mode, running average* ve *scoreboard* algoritması önerilmiştir. Sonuçlar, *scoreboard* algoritmasının arka plan modelini daha doğru tahmin edebileceğini göstermiş ve bu algoritma *running mode* ve *running average* algoritmalarından daha hızlı çalışmıştır.

2.1.2 Çoklu gausslar kullanan istatistiksel yöntemler

Arka plan modelindeki varyansı daha efektif bir şekilde modellemek için olasılıksal yaklaşımlar uyarlanmıştır. En yaygın kullanılan olasılık modellerinden biri Gaussian Mixture Model (GMM)'dir (Stauffer ve Grimson 1999). Bu yaklaşımda Stauffer ve Grimson (1999) tarafından her pikselin tüm piksel değerlerini tek bir dağılım olarak modellemek yerine bir arka plan pikseli ya da bir ön plan pikseli olup olmadığını modellemek için bir Gauss karışımı önerilmiştir. Bu yöntem, aydınlatma

değişikliklerine, kümelenmiş bölgelere ve yavaş hareket eden nesnelere karşı dayanıklıdır. Ancak, aydınlanmada hızlı değişikliklere ve gölgelere karşı sorunludur.

Kaewtrakulpong ve Bowden (2001) tarafından segmentasyonun doğruluğunu artırmak için Stauffer ve Grimson (1999)'un denklemi değiştirilmiştir. Bundan başka, var olan GMM'i kullanarak gölgeleri ortadan kaldırmak için bir gölge algılama düzeni önerilmiştir. Önerilmiş olan yöntem Mixture of Gaussian V1 (MoG) olarak adlandırılmıştır.

Zivkovic (2004) tarafından her piksel için Gauss dağılımının sayısını sürekli olarak uyarlayarak GMM algoritması geliştirilmiştir. Zivkovic (2004)'in yöntemi segmentasyon sonuçlarını biraz geliştirmektedir ancak bu algoritmanın işlem süreleri açısından Stauffer ve Grimson (1999)'un algoritmasından daha hızlı işlemektedir. Diğer yazarlar tarafından da farklı parametrik yöntemler geliştirilmiş ve bu yöntemler, Tuzel vd. (2005), Zivkovic ve Van Der Heijden (2006), Bouwmans vd. (2008), ve Benezeth vd. (2008) tarafından önerilmişlerdir.

2.1.3 Parametrik olmayan yöntemler

Önceki bölümde tartışıldığı gibi çok sayıda hesaplama içermesi nedeniyle parametrik yöntemler (veya istatistiksel yöntemler) hesaplama açısından pahalıdır. Bu bölümde parametrik olmayan yöntemlerle ilgili bazı literatürler incelenmektedir.

Barnich ve Van Droogenbroeck (2011) tarafından mevcut piksel değerinin bir örnek koleksiyonu içinde en yakın örneğiyle karşılaştırıldığı piksel tabanlı bir yöntem önerilmiştir. Bu yöntem ViBe (Visual Background Extractor) olarak adlandırılmıştır. ViBe, piksel modellerini her bir pikselin uzaysal mahallelerinde rastgele alınan değerlerle doldurarak arka plan modelini tek bir kareden başlatmıştır. Bu yöntem küçük kamera hareketlerine hızlı uyum sağlamıştır. Üstelik, zor gürültülere karşı dayanıklıdır ve düşük hesaplama maliyeti vardır. Bu yöntem piksel modelinin hareketli nesnelerden kötü başlatılması nedeniyle ilk karede gürültüler oluşturmuştur.

Van Droogenbroeck ve Paquot (2012) tarafından arka plan örnekleri ile var olan piksel arasındaki mesafe ölçümü, thresholding gibi parametrelerin ayarlarını ayarlayarak ViBe'nin orijinal çalışmasına çeşitli değişiklikler önerilmiştir. Bu yöntem ViBe+ olarak adlandırılmıştır. ViBe+, CDnet2014 veri kümesinin ana kategorilerinde orijinal çalışmadan daha iyi performans göstermiştir.

Son zamanlarda parametrik olmayan yöntemlerden biri SuBSENSE (Self-Balanced SENsitivity SEgmenter)'dir. Bu yöntem St-Charles vd. (2015a) tarafından önerilmiştir. St-Charles vd. (2015a) tarafından kamufle edilmiş objeleri tespit edebilen Local Binary String Pattern (LBSP) geliştirilmiştir. Ayrıca, aydınlatma varyasyonunu ele alabilmektedir ve düşük hesaplama maliyeti vardır.

St-Charles vd. (2015b) tarafından de PAWCS (Pixel-based Adaptive Word Consensus Segmenter) adlı başka bir yöntem önerilmiştir. Bu çalışmada yazarlar, arka plan modelinin renk ve doku bilgileri kullanılarak oluşturulduğu kelime tabanlı bir model sunmaktadır ve piksellerin görünümü arka plan kelimeleri olarak zamandan zamana kaydedilmiştir. Daha sonra, eğer bu arka plan kelimeri kalıcı ise bunu arka plan olarak kabul edilmiştir. Bu yöntem CDnet2012 veri kümesindeki (Goyette vd. 2012) tüm yöntemlerinden daha iyi performans göstermiştir.

Bianco vd. (2017) tarafından, var olan *Change Detection* yöntemlerinden en iyi yaklaşımları seçilmek için bir Genetik Programlama kullanılmıştır ve daha sonra bu metotlardan elde edilen sonuçlar birleştirilmiştir. Son etiketleri belirlemek için post-processing tekniği uygulanmıştır.

2.1.4 Derin öğrenme yöntemleri

Son zamanlarda derin öğrenme yaklaşımları, pek çok araştırmacı tarafından sahnelerdeki gizli öznitelikleri öğrenmeye ve bu öznitelikleri kullanarak video dizilerindeki ön plan nesnelerini bölütlemeye dayalı olarak önerilmiştir. Braham ve Van Droogenbroeck (2016) tarafından CNN'ler kullanılarak sahneye özel bir yöntem

önerilmiştir. Biraz daha açmak gerekirse, belirli bir sahne için tek bir arka plan modeli oluşturulmuştur. Bir video sekansındaki her bir çerçeve için her bir piksel üzerinde ortalanmış olan görüntü yamaları çıkarılmış ve daha sonra arka plan modelinden gelen ilgili yamalar ile birleştirilmiştir. Bundan sonra, bu birleştirilmiş yamalar ön plan piksellerinin olasılığını tahmin etmek için ağı beslenmiştir. Eğitim örneklerinin yarısı ağı eğitmek için kullanılmış (ground-truth etiketlerini içeren çerçevelerin aralığını dikkate alarak) ve kalan çerçeveler test için alınmıştır. Bu yöntem CDnet2014 veri kümesi üzerinde 0.9046 ortalama F-Measure elde etmiştir. Fakat, bu metot her bir çerçeveden çıkarılan çok sayıda yama nedeniyle hesaplama açısından pahalıdır.

Babaee vd. (2017) tarafından arka plan çıkarma alanında bir CNN yöntemi de önerilmiştir. Ağ modeli çeşitli video dizilerinden eğitim çerçevelerini birleştirerek bir kerede eğitilmiştir; özellikle, her video sekansından çerçevelerin % 5'i dahildir. Braham ve Van Droogenbroeck (2016)'daki gibi aynı eğitim yolunu takip edilmiştir, burada görüntü yamaları arka plan yamaları ile birleştirilmiş ve daha sonra ağı beslenmiştir. Segmentasyon maskelerini yumuşatmak için bir post-processing tekniği uygulanmıştır. Bu yöntem CDnet2014 veri kümesi üzerinde 0.7548 ortalama F-Measure elde etmiştir. Fakat, görüntü yamaları kullanılarak yapılan eğitimle hesaplamanın pahalı olmasından kaynaklı dezavantajı vardır.

Wang vd. (2017) tarafından Braham ve Van Droogenbroeck (2016)'nın metoduna benzer bir arka plan çıkarma için derin bir öğrenme yaklaşımı da önerilmiştir. Bu yöntem sahneye spesifik strateji kullanılarak eğitilmiştir. Aynı araştırmacı tarafından bir video dizisinde 200 kare kullanarak yönteminin mevcut tüm yöntemlerden daha iyi performans gösterdiğini ve CDnet2014 veri kümesi üzerinde 0.95 ortalama F-Measure elde ettiğini iddia edilmiştir. Bu yöntem küçük hareketli nesneler nedeniyle zayıf performans göstermiştir.

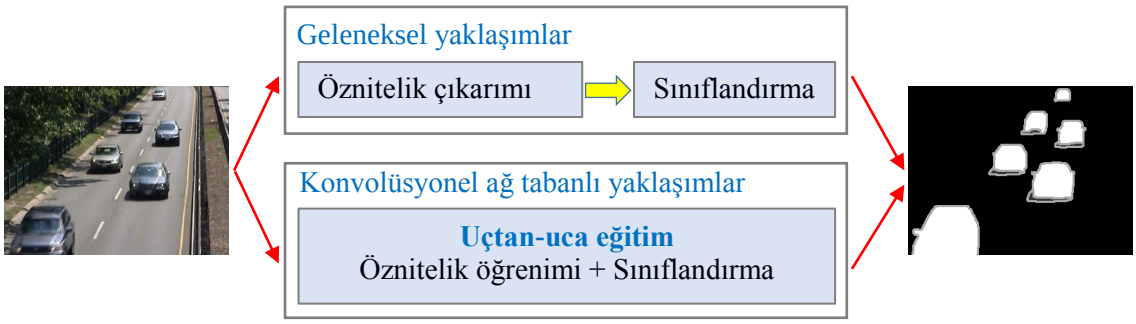
Sakkos vd. (2017), eğitimde herhangi bir arka plan modeli kullanmadan video dizilerindeki zamansal değişiklikleri izlemek için bir 3D konvolüsyon tekniği kullanmıştır. Bu yaklaşım, CDnet2014 veri kümesinde ortalama 0.9507 F-Measure ile gerçekleştirilmiştir.

Lim vd. (2017) tarafından VGG-16 Net uyarlanmış ve birleştirilen 3 gri tonlamalı görüntü (önceki çerçeve, hedef çerçeve, arka plan modeli) alıp tek bir çıktı üreten bir üretici-kodlayıcı (enkoder-dekoder) ağı önerilmiştir. Son segmentasyon maskesindeki boşlukları doldurmak için kenar dedektörü ve süper pikselle elde edilen kontur bilgileri kullanmışlardır.

Cinelli vd. (2017) tarafından bir kodlayıcı olarak ResNet (He vd. 2016) kullanılmış ve kod çözücü varyasyonları hakkında kapsamlı bir çalışma sunulmuştur. Önerilen ağ, oluşturulan arka plan modeliyle eğitilmiş ve çerçevelerin % 70'i eğitim ve % 30'u da test için bölünmüştür.

2.2 Materyal

Bu bölümde derin öğrenme yaklaşımını kullanarak görsel öznitelik temsili öğrenimine genel bir bakış sunulmaktadır. Geleneksel bilgisayarda görü yaklaşımlarında el yapımı bir öznitelik çıkarıcısı, kenarlar, köşeler, dokular veya renk yoğunlukları gibi görüntülerden ilgili bilgileri toplamaktadır. Daha sonra, ortaya çıkan öznitelik vektörleri eğitilebilir bir sınıflandırıcı tarafından sınıflara ayrılmaktadır. Yukarıda açıklandığı gibi bu yaklaşım iki adımda yapılmaktadır. Fakat, artık manuel öznitelik çıkarma adımı ortadan kaldırılabilmektedir. Konvolüsyel sinir ağları (CNN'ler) ham görüntülerden gizli öznitelik temsillerini öğrenebilme ve ilk birkaç katmanı bir öznitelik çıkarıcıya dönüştürebilmektedir. Bu yaklaşım güçlü bir şekilde ortaya çıkmaktadır çünkü el yapımı öznitelik çıkarıcıları için gereksinimi ortadan kaldırmaktadır. Şekil 2.2'de yukarıdaki belirtilen öznitelik tabanlı ve temsil öğrenmeyi göstermektedir.



Şekil 2.2 Ön plan segmentasyon problemi için öğrenme özelliği

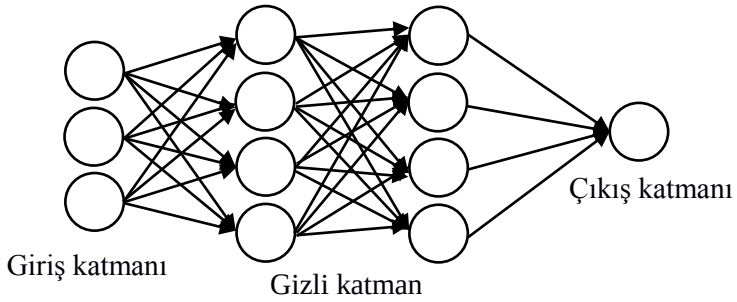
Üst kutu, özelliklerinin elle çıkarıldığı geleneksel bilgisayar görme yaklaşımını göstermmek ve daha sonra bu özellikler sınıflandırıcıyı eğitmek için kullanılmaktadır; Hiyerarşik özellikler ve sınıflandırıcıların birlikte eğitildiği derin öğrenme yaklaşımının aksine (alt kutu)

2.2.1 Çok katmanlı perseptron

Çok katmanlı perseptron (Multilayer Perceptron veya MLP) ileri beslemeli yapay sinir ağlarının bir türüdür. Perseptron, tek bir y_i çıktısını çoklu x_i girişi ve w_i ağırlığı ile lineer bir kombinasyon ile hesaplar ve buna *bias* b_i ekler. Sonra, isteğe bağlı olarak lineer olmayan aktivasyon fonksiyonu σ takip edilmektedir. Matematiksel olarak şu şekilde yazılabilmektedir:

$$y_i = \sigma(w_i^T x_i + b_i)$$

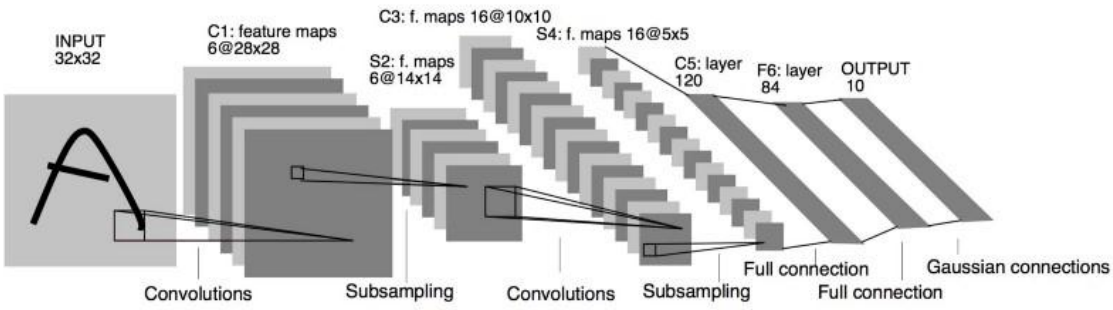
MLP en az üç katman içermektedir; bunlar giriş katmanı, gizli katman ve çıktı katmanıdır. Şekil 2.3'de üç katmanlı bir sinir ağını göstermektedir. Her nöronun, önceki katmandaki tüm nöronlara bağlı olduğu bir yapıya sahiptir. Bu şekilde sadece 3 giriş nöronu yani 3 boyutlu bir giriş vektörü içermektedir. Fakat, gerçek dünyada giriş görüntüsünün 224x224x3 boyutunda olduğunu varsayalım; yani 224 genişlik, 244 yükseklik, ve 3 kanal. Bu durumda tek bir katmanda 224x224x3 ~ 150K boyutunda bir giriş vektörü vardır. Çok katmanlı sinir ağı ikiden fazla katmana sahip olamalıdır. Böylece, ağdaki parametrelerin sayısı hızla artacak ve normal bir sinir ağı büyük görüntüler iyi ölçeklenememektedir. Bu nedenle, bu tür bir sorunu çözmek bilgisayarda görü alanında yaygın olarak evrişimsel sinir ağları kullanılmaktadır.



Şekil 2.3 Üç katmanlı bir sinir ağı (giriş katmanı hariç)
Bir giriş katmanı (3 nöron); iki gizli katman (8 nöron); bir çıkış katmanı (1 nöron)

2.2.2 Konvolüsyonel sinir ağları

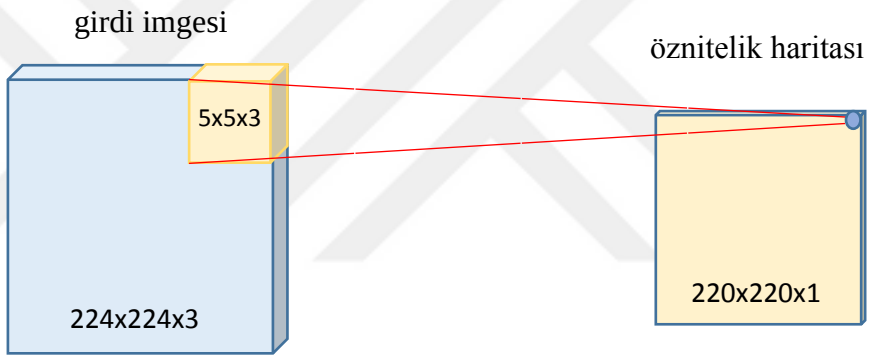
Evrişimsel sinir ağları (CNN'ler) veya konvolüsyon ağları, memeli hayvanların görsel korteksinde bulunan biyolojik süreçlerden ilham almaktadır. CNN'ler ilk olarak LeCun vd. (1998) tarafından rakam tanımada tanıtılmıştır (Şekil 2.4). CNN'ler, görüntüler gibi 2D yapı verilerinin avantajlarından yararlanmak için tasarlanmış özel bir yapay sinir ağıdır (Artificial Neuron Network veya ANN). CNN'ler, görüntü sınıflandırması (Krizhevsky vd. 2012, Szegedy vd. 2015, He vd. 2016), semantik segmentasyon (Long vd. 2015), konuşma tanıması (LeCun ve Bengio 1995, Graves vd. 2013) gibi birçok pratik uygulamada başarıyla kullanılmıştır. Tıpkı normal sinir ağı gibi, CNN'ler öğrenilebilir ağırlıklardan ve bias'lardan oluşmaktadır. CNN'ler mimarisini oluşturmak için başlıca dört ana katman türü kullanmaktadır: Konvolüsyonel Katman, Havuzlama Katmanı (Pooling Katmanı), Aktivasyon Katmanı ve Tam-bağlı Katman (Fully-connected Katman).



Şekil 2.4 Rakam tanıma için tipik bir CNN mimarisi (Görüntü LeCun vd. (1998)'den alınmıştır)

2.2.2.1 Konvolüsyonel katman

Girişe konvolüsyon operasyonunu uygulayan ve öznitelik haritalarıyla (veya aktivasyon haritalarıyla) sonuçlanan konvolüsyonel katmanlar CNN'lerin çekirdekleridir, ve daha sonra bu öznitelikler sonraki katmanlara geçirilmektedir. Konvolüsyonel katmanlar isteğe bağlı olarak aktivasyon katmanları ile takip edilmektedir. 224x224x3 boyutunda bir görüntü (genişlik 224, yükseklik 224 ve derinlik 3) ve 5x5x3'lik bir filtre boyutu (genişlik 5, yükseklik 5 ve derinlik 3) olduğunu varsayalım. Bir filtre giriş görüntüsünün küçük bölgeleri üzerinde kaymak ve çarpma işlemi gerçekleştirir, sonuçta çıkış nöronları ortaya çıkmaktadır. Bu çıkış nöronları 3D öznitelik haritaları oluşturmak için yeniden düzenlenmektedir (Şekil 2.5).



Şekil 2.5 Konvolüsyonlu Yapay Sinir Ağlarının konvolüsyonel işlemi
Filtre, öznitelik haritasını oluşturmak için giriş görüntüsündeki uzaysal konumlara kaydırılmaktadır

Tıpkı normal sinir ağı gibi, konvolüsyon katmanları, filtre girdinin küçük bölgeleri arasında çarpmayı gerçekleştirmektedir. Matematiksel olarak bu işlem şu şekilde yazılabilmektedir:

$$y_i = \sigma(w_i^T x_i + b_i)$$

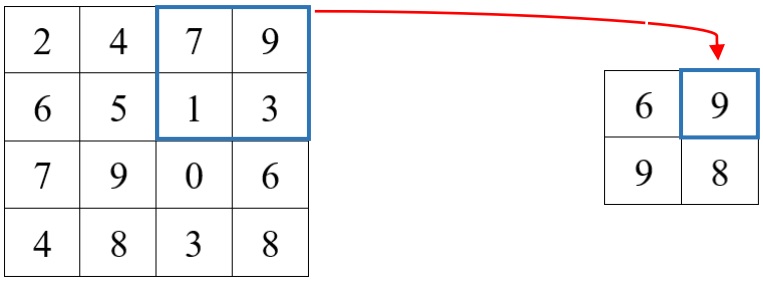
burada, w_i öğrenilebilir ağırlık veya filtredir, x_i giriş görüntüsünün lokal bölgesidir, b_i bias'dir, σ aktivasyon fonksiyonudur ve y_i elde edilen matris çarpımı sonucudur.

Parametre Paylaşımı: CNN'lerin ilginç bir özelliği ağırlık paylaşım özelliğidir. CNN'ler bellek gereksinimleri açısından normal sinir ağlarından daha verimlidir. Yukarıdaki örnek için parametre paylaşımı olmadan ilk konvolüsyonel katmanında 224x224x64 = 3,211,264 nöron vardır, ve her öznitelik haritasında 5x5x3 = 75 ağırlık ve 1 bias vardır. Bu durumda ilk konvolüsyonel katmanda 3,211,264 x 76 = 244,056,064 parametre vardır ve bu parametreler çok büyüktür. Ancak, her öznitelik haritasındaki nöronların aynı ağırlığı kullanmasını sağlanarak parametre sayısı azaltılabilmektedir. Bu durumda 5x5x3 = 75 benzersiz ağırlık ve 1 benzersiz bias vardır. Bütün olarak ilk konvolüsyonel katmanda 64 x 76 = 4.864 parametre vardır. Parametre paylaşımı, konvolüsyonel katmanların ***Equivariance*** çeviri özelliğine sahip olmasını sağlamaktadır; yani giriş değişirse çıkış değişmektedir (Goodfellow vd. 2016).

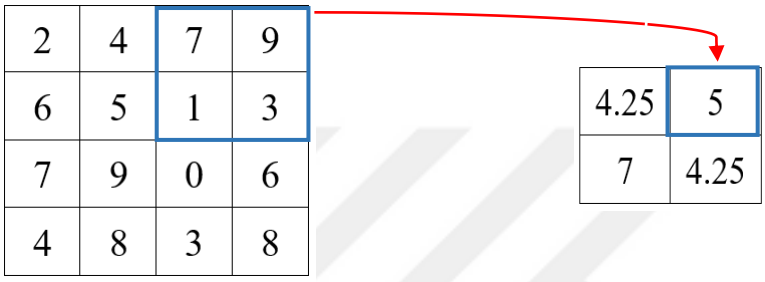
1x1 Konvolüsyonu: 1x1 konvolüsyonu, başka bir adıyla point-wise konvolüsyonu, sınıflandırma görevinde Lin vd. (2013) tarafından önerilmiştir. 1x1 konvolüsyonu ağdaki parametre sayısının azaltılmasında yararlı olduğu ortaya çıkmıştır. Son zamanlarda majör yazarlar görüntü tanıma görevinde 1x1 konvolüsyonu kullanmışlardır (Szegedy vd. 2015, He vd. 2016). Ağımızda parametrelerin sayısını azaltmak için 1x1 konvolüsyonu kullanılmıştır.

2.2.2.2 Havuzlama (pooling) katmanı

Pooling katmanı genellikle konvolüsyonel katmanlardan sonra eklenmektedir. Her bir giriş öznitelik haritasında bağımsız olarak çalışmak ve parametrenin sayısını artırmadan temsilin uzaysal boyutlarını azaltmaktadır. Pooling katmanı küçük çeviri değişmezlerinin artmasına yardımcı olabilmek ve ağdaki parametrelerin miktarını önemli ölçüde azaltmaktadır. CNN'lerde iki ana pooling katmanı vardır: bunlar maksimum pooling ve ortalama pooling katmanıdır. Max pooling katmanı maksimum operasyonu kullanmaktadır, burada sadece maksimum aktivasyonlar tutulur ve minimum aktivasyonlar elenmektedir (Şekil 2.6). Ortalama pooling katmanı, küçük bölgedeki aktivasyonların üzerinden bir ortalama hesaplamaktadır (Şekil 2.7).



Şekil 2.6 Konvolüsyonel sinir ağların maksimum pooling operasyonu



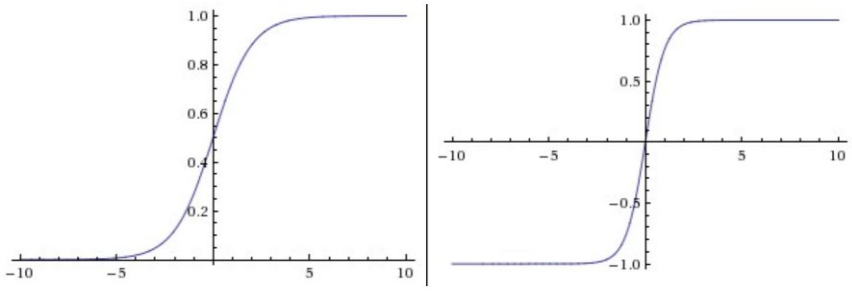
Şekil 2.7 Konvolüsyonel sinir ağların ortalama pooling operasyonu

2.2.2.3 Aktivasyon katmanı

Ağda aktivasyon katmanı genellikle konvolüsyon katmanlarından veya batch-normalizasyon katmanlarından sonra eklenmektedir. Aktivasyon fonksiyonu (veya transfer fonksiyonu), ağa doğrusal olmayan (non-linearity) bir yapı kazandırır. Özellikle, aktivasyon fonksiyonu öznitelikleri başka bir öznitelik uzayına taşır ve bu da karar fonksiyonunu daha ayrımcı yapmaktadır. CNN'lerde yaygın olarak kullanılan üç aktivasyon fonksiyonu vardır:

- **Sigmoid Fonksiyonu:** Bu doğrusal olmayan fonksiyon, gerçek değerleri [0, 1] aralığına sıkıştırmaktadır. Sigmoid doğrusal olmayan değerleri, nöronların ateşleme oranları olarak yorumlanabilmektedir. Burada 0 değeri nöronların ateşlenmediğini veya değerin 1 olması durumunda tamamen doymuş olduğunu göstermektedir. Sigmoid fonksiyonu, gradyanın geri yayılım sırasında sönümlenmesine ve ağa neredeyse hiç gradyan akamamasına neden olabilmektedir. Sigmoid fonksiyonu (Şekil 2.8. a), şu şekilde yazabilmektedir:

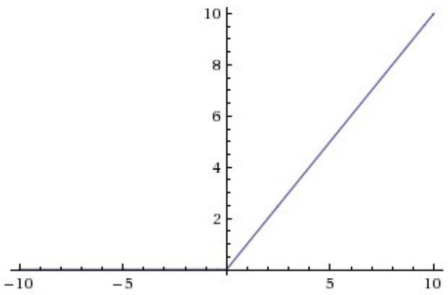
$$\sigma(x) = \frac{1}{1+e^{-x}}$$



(a) (b)

Şekil 2.8 a. Sigmoid aktivasyon fonksiyonu, b. Tanh aktivasyon fonksiyonu

- **Tanh Fonksiyonu:** Bu doğrusal olmayan fonksiyon, gerçek değerleri [-1, 1] aralığına sıkıştırmaktadır. Sigmoid fonksiyonundan farklı olarak bu fonksiyonun çıkışı sıfır merkezlidir (Şekil 2.8.b). Tanh fonksiyonu şu şekilde yazılabilmektedir: $tanh(x) = 2\sigma(2x)-1$, burada $\sigma(x)=(1+e^{-x})^{-1}$
- **ReLU (Rectified Linear Unit) Fonksiyonu:** Son yıllarda bu doğrusal olmayan aktivasyon yaygın olarak kullanılmaktadır ve günümüzede de popülerdir. Bu fonksiyon şu şekilde yazılabilmektedir: $f(x) = max(0, x)$. ReLU, yakınsama ve antrenman hızı açısından sigmoid ve tanh aktivasyon fonksiyonundan daha iyi bulunmuştur (Krizhevsky vd. 2012). ReLU fonksiyonu şekil 2.9'de gibi gösterilmektedir.



Şekil 2.9 ReLU aktivasyon fonksiyonu

2.2.2.4 Kayıp fonksiyonu, regularization ve optimizasyon

Ön plan nesneler segmentasyon problemi için x_i görüntüsünün y_i ground-truth'u ile ilişkilendirilmesi durumunda, tahmin edilen skor $s_i = W^T x_i + b$ ile tanımlanmaktadır. Basitlik için bias b, ağırlık matrisine W genişletilir ve 1 sabit değeri x_i matrisine eklenebilmektedir. Bu genellikle *bias trick* olarak adlandırılmıştır.

Kayıp Fonksiyonu (**L**): maliyet fonksiyonu, amaç fonksiyonu olarak da bilinir, tahmin edilen s_i skoru ile y_i ground-truth'u arasındaki anlaşmayı ölçmektedir. Böyle bir anlaşmayı yapmak için ağ tarafından kayıp fonksiyonunu (L) en aza indirecek şekilde W ağırlığı öğrenilmesi gerekmektedir. Bu bölümde tartışacağımız iki kayıp fonksiyonu vardır: bunlar Binary Cross Entropy kayıp fonksiyonu ve Softmax Cross Entropy kayıp fonksiyonudur.

Elimizde N eğitim örneği olduğunu ve her bir örneğin M piksel içerdiğini varsayalım:

$$\left\{\left\{x_j^i, y_j^i\right\}_{j=0}^{M-1}\right\}_{i=1}^{N}, j \in \{0,1,2,...,M-1\}, i \in \{1,2,3,...,N\}$$

, burada x^i_j, i örneğinin j konumunda bir ham pikseldir, y^i_j ise i örneğinin j konumundaki ham pikselin doğru sınıfını belirten ayrık bir değişkendir.

- Binary Cross Entropy Kayıp fonksiyonu gerçek etiketi y^i_j ve tahmin edilen değeri karşılaştırmak için kullanılmaktadır. i örneğinin binary cross entropy kayıp fonksiyonu şu şekilde tanımlanmaktadır:

$$L_i = \frac{-1}{M}\sum_{j=1}^{M}[y_j^i log(p_j^i) + (1-y_j^i) log(1-p_j^i)]$$

, burada p^i_j , i örneğinin j konumunda pikselin tahmin edilen olasılık değeridir.

- Softmax Cross Entropy Kayıp fonksiyonu şu şekilde tanımlanmaktadır:

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right)$$

, burada s_{yi} doğru sınıfın skorudur, s_j ise her sınıfın skorudur. Amacımız doğru sınıfın negatif log olasılığını en aza indirmektedir, böylece kayıp minimum olacaktır. Yukarıdaki iki kayıp için, tüm eğitim örnekleri üzerindeki tam kayıp şöyle tanımlanmaktadır:

$$L = \frac{1}{N}\sum_{i=0}^{N-1} L_i$$

Regularization: Makine öğrenmesinde gözlemlenememiş verilere iyi genelleme kabiliyetine sahip modeller geliştirmekle ilgilenmektedir. Makine öğrenmesi modellerini eğitirken iki tür problem vardır: bunlar, underfitting ve overfitting'dir. Modelin kapasitesini değiştirerek modelin underfit veya overfit olup olmadığını kontrol edebilmektedir. Yüksek kapasiteli model, eğitim setini ezberleyerek overfit edebilmek ve görünmeyen verilere zayıf bir genelleme sağlamaktadır. Ağda ezberlemeyi önlemek için regularization yöntemleri uygulanabilmektedir; bunlar L1 veya L2 regularization, dropout, ve eğitim setinin yapay genişlemesidir.

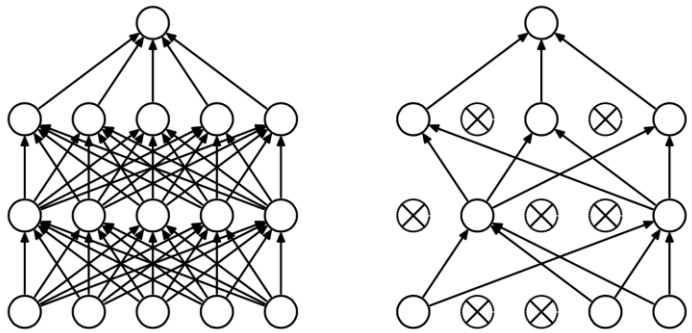
- **L2 regularization:** En yaygın regularization tekniği, model ağırlıklarının küçük olmasını sağlayan L2 regularization'udur. Bu regularization fonksiyonu model ağırlıklarına dayanır ve veri kaybı fonksiyonu ile bir regularization kaybı olarak genişletilebilmektedir. Matematiksel olarak bu kombinasyon şu şekilde yazılabilmektedir:

$$L = \frac{1}{N}\sum_{i=0}^{N} L_i + \lambda R(W), \quad where\ R(W) = \sum_j \sum_k (W_{j,k}^2)$$

, burada λ [0, 1] arasındaki aralıktaki regularization gücüdür.

- **Dropout regularization**: Dropout, Srivastava vd. (2014) tarafından tanıtılan başka bir regularization tekniğidir. Bu ağda ezberlemeyi önlemek için basit ve etkili bir tekniktir. Dropout nöronların eğitim sırasında bazı olasılıklarla aktif

tutulacak şekilde uygulanmıştır. Test süresinin boyunca hiçbir Dropout uygulanmamaktadır. Şekil 2.10'da çıkışın normal sinir ağına nasıl uygulandığını göstermektedir.



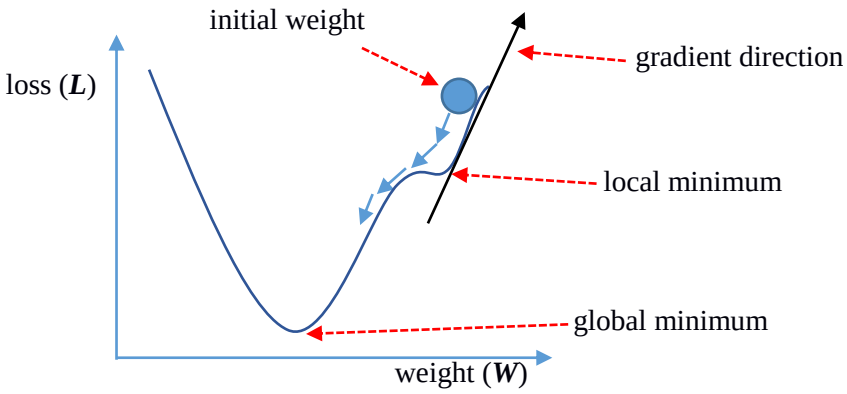
(a) (b)

Şekil 2.10 Dropout sinir ağı hakkında bir illüstrasyon (Görüntü Srivastava vd. (2014)'ten alınmıştır)
a. standart bir sinir ağı, b. Dropout bir sinir ağı

Optimizasyon: CNN'lerde optimizasyonun amacı kayıp fonksiyonunu en aza indirgeyen bir ağırlık seti (*W*) bulmaktır. Minimum kaybı bulmak için basit ve etkili bir yol, gradyanin negatif yönünü takip etmektedir. Bu işlemi *gradient descent* olarak da bilinmektedir. Gradient descent'in ağırlıklar güncellemeleri şu şekilde tanımlanmaktadır:

$$W \leftarrow W - \alpha \frac{\partial L}{\partial W}, \quad b \leftarrow b - \alpha \frac{\partial L}{\partial b}$$

, burada α öğrenme oranıdır (veya adım büyüklüğü). $\frac{\partial L}{\partial W}$, ***W*** ağırlıklarına göre ***L*** kayıplarının birinci derece türevidir, $\frac{\partial L}{\partial b}$ için de benzerdir. Şekil 2.11, ağırlıkların nasıl güncellendiğini göstermiştir. İlk olarak, ağırlıklar küçük rastgele değerlerle başlatılmıştır. Negatif gradyan yönünü takip ederek ağırlık güncellemeleri, lokal minimumdan kaçınır ve kaybın global minimumuna ulaşacak şekilde adım atabilmektedir.



Şekil 2.11 Ağırlık güncellemelerinin nasıl adım attığının görselleştirilmesi

2.2.3 Veri kümeleri

2.2.3.1 CDnet2014 veri seti

Deneylerimizde CDnet2014 veri kümesi (Wang vd. 2014) kullanılmaktadır. CDnet2014 ver seti *baseline*, *camera jitter*, *bad weather*, *dynamic background*, *intermittent object motion*, *low frame rate*, *night videos*, *PTZ* (panning-tilting-zooming), *shadow*, *thermal* ve *turbulence* gibi 11 kategori içermektedir. Her kategori 4 ila 6 video dizisi içermektedir. Bütün olarak 53 farklı video dizisi vardır. Video karelerinin mekansal çözünürlükleri 320x240 ila 720x576 piksel arasında değişmektedir. Üstelik, bir video dizisi 600 ila 7999 kare içerebilmektedir. Neredeyse tüm video dizileri, bu veri setini her bir durumda bir modelin sağlamlığını ölçmek için uygun kılan farklı zorlu senaryolar içermektedir.

2.2.3.2 SBI2015 veri seti

Wang vd. (2017) tarafından sağlanan groundtruth etiketli 14 video sekansı içeren Scene Background Initialization 2015 (SBI2015) veri seti (Maddalena ve Petrosino 2015) üzerinde daha fazla deney yapılmaktadır. Ek olarak, bir video dizisi 6 ila 740 kare içerebilmektedir.

2.2.3.3 UCSD Background Subtraction veri seti

Groundtruth etiketleriyle 18 video dizisi içeren UCSD Background Subtraction (Mahadevan ve Vasconcelos 2010) veri kümesinde başka bir deney gerçekleştirilmektedir. Bu veri kümesi, arka plan çıkarma alanında son derece zorlayıcı dinamik arka planlar içermektedir. Ek olarak, bir video dizisi 4 ila 111 kare içerebilmektedir.

2.2.4 Değerlendirme metrikleri

Bu bölümde model performans değerlendirmesi için üç farklı metrik kısaca tartışılmaktadır: bunlar F-Measure, MCC ve Percentage of Wrong Classifications (PWC). True negative (TN) dahil etmeden true positive (TP), false positive (FP), false negative (FN) verilen precision ve recall'ın harmonik ortalaması (F-Measure), şu şekilde tanımlanmaktadır:

$$F - Measure = \frac{2 \times precision \times recall}{precision + recall}$$

,burada *precision=TP/(TP+FP)* ve *recall=TP/(TP+FN)*. *F*-Measure, [0,1] aralığındadır, burada 1 değeri, tahmin edilen maskenin ground-truth ile tamamen aynı fikirde olduğunu göstermektedir. Ancak, 0 değeri anlaşmazlık göstermektedir. TN dahil ederek PWC şu şekilde tanımlanmaktadır:

$$PWC = \frac{100 \times (FP + FN)}{TP + FP + TN + FN}$$

Fakat, daha önce belirtildiği gibi F-Measure dengesiz sınıflara duyarlıdır. Üzerinde düşünürsek; bir çerçevede ön plan pikseli yoktur, bu durumda modelimiz arka plandaki tüm arka plan piksellerini doğru şekilde sınıflandırmasına rağmen F-Measure sıfır olacaktır. Bu sorunun üstesinden gelmek için MCC metriği dengesiz sınıf problemi için stabil olaması nedeniyle performans ölçümlerinde kullanılmaktadır. MCC metriği TP, FP, FN ve TN açısından tanımlanmaktadır:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FN)(TN+FP)}}$$

, burada MCC'nın değeri [-1, +1] aralığındadır, burada +1 değeri tahmin edilen maskenin ground-truth ile tamamen aynı fikirde olduğunu ve -1 değerinin de anlaşmazlık olduğunu belirtmektedir.

2.3 Eğitim Örneği Seçimi

Sahneye özgü modeller için çerçevelerin seçilmesi çok önemli olabilir eğer ki arka plan dinamikse ve video sekansındaki görüntüler CDnet2014 veri kümesinde *thermal, dynamicBackground, badWeather,* veya *turbulence* kategorisi gibi önemli nesneler içeriyorsa dikkat gerekebilmektedir. Hafifçe sallanan ağaçlar gibi daha az arka plan hareketine sahip bir statik video dizisi için sadece bir dizi eğitim örneği, yani 50 çerçeve yeterli olacaktır. Çerçeveler, bazı ön plan nesneleri içeren karelere daha fazla odaklanarak rasgele seçilebilmektedir. Bu strateji, ağın ön plandaki pikselleri daha doğru şekilde öğrenmesine ve bölütlemesine yardımcı olmaktadır. Fakat, daha karmaşık sahneler ve dinamik arkaplanlar veya kamera kaydırma-eğme-yakınlaştırma video dizileri için seçilen örneklerde sahnenin farklı bölümlerini dahil ederek daha fazla eğitim örneği, yani 200 kare seçmek daha iyidir. Eğitim çerçevelerinin içeriği, ön plan veya arka plan parçalarını veya her ikisini de içerebilmektedir. ***n*** çerçeve sayısı seçilebilir, burada ***n*** << ***N*** ve ***N*** bir video dizisindeki toplam çerçeve sayısıdır. Deneylerimizde CDnet2014 veri seti için iki ayrı eğitim için 50 ve 200 çerçeve manuel ve rasgele seçilmiştir. Üstelik, SBI2015 ve UCSD veri kümeleri için eğitim çerçevelerinin sınırlı sayıda olmasından dolayı, veri kümesinin % 20'si eğitim ve kalan % 80'i test olacak şekilde bölüştürülmüştür. UCSD veri kümesi için % 50'lik eğitim bölütlemesiyle başka bir deney de sunulmuştur. Bir sonraki bölümde, bu alandaki gürbüz modellerin oluşturulmasında dikkat edilmesi gereken denetimli ikili sınıflandırmada dengesiz sınıf problemi tartışılacaktır.

2.4 Dengesiz Veri ile Çalışma

Denetimli bir eğitim ortamında farklı sınıf kategorileri için dengesiz eğitim örneklerinin sayısı, sınıflandırmada yanlılık sorunlarına neden olabilmektedir; bu aktif bir araştırma problemidir (He ve Garcia 2009, Chawla vd. 2004). Geniş görüş alanı gözetleme kamerası ayarlarında arka plan piksellerinin dağılımı genellikle ön plan piksellerinden daha ağır basdığından bu problemi ön plan nesnelerinin segmentasyonu eğitiminde ortaya çıkabilmektedir. Bu ikili sınıflandırma (veya binary classification) alanında 100: 1, 10000: 1 veya hatta 10000: 0 gibi ciddi oranlarda dengesiz veri problemi ortaya çıkabilmekte ve bu da sınıflandırmada optimal performansa neden olmaktadır. Bu sorun iki düzeyde hafifletilebilmektedir: bunlar veri seviyesi ve algoritmik seviyedir. Bizim problemimizde, veri düzeyinde dengesiz sınıflar ile uğraşmak zordur, dolayısıyla bu problem algoritmik düzeyde hafifletilmektedir. Ön plan bir piksel bir arka plan pikseli olarak sınıflandırılırsa, hesaplanan kaybı daha fazla cezalandırarak uygulanmıştır. Her bir eğitim çerçevesi için ground-truth'u bağımsız olarak kullanarak ön plan/arka plan piksel dağılımını kullanarak sınıf-ceza-ağırlıkları hesaplanmıştır. Özellikle, CDnet2014 veri kümesinin video dizilerinde, arka plan pikselleri ön plan piksellerinden ciddi şekilde daha fazladır, dolayısıyla eğitim sırasında ağırlıklı kayıp hesaplaması uygulanmıştır.

Ön plan segmentasyonu (veya arka plan çıkarması) probleminde F-Measure (veya F1-score), model performanslarını değerlendirmek için yaygın olarak kullanılmaktadır. Fakat, Boughorbel vd. (2017) tarafından da iddia edildiği gibi F-Measure, True Negativeleri (TN) dikkate almadığı için dengesiz verilere duyarlıdır. Boughorbel vd. (2017), Matthews (1975) tarafından önerilen MCC'nin (Matthews Correlation Coefficient) True Negatiflerin dahil edilmesine bağlı olarak dengesiz sınıfları sınıflandırma problemlerinde daha uygun olduğunu iddia etmiştir. Sonraki bölümde yama-tabanlı öğrenme kapsamlı olarak tartışılacaktır.

2.5 Yama-tabanlı (Patch-wise) Öğrenme

Ön plan nesnelerin segmentasyonu için genel olarak kullanılan bir yaklaşım patch-wise öğrenmesidir. Bu durumda bir video dizisindeki her çerçeveden her piksele ortalanmış görüntü yamaları çıkarılmaktadır. Çıkarılan yamalar, merkezlenmiş pikselin ön plan veya arka plan pikseli olarak tahmin edilmesi için ağı beslemektedir. Yama boyutu, bağlamsal bilgileri kapsayacak ve daha az hesaplama maliyeti olan bir şekilde tanımlanmalıdır. Çokca kullanılan yama boyutları, [25-39] arasındaki aralıktaki tamsayı değerleridir. Bu bağlamda küçük bir yama boyutu ağ tarafından ilgili bilgileri öğrenmek için mücadele edilen daha az küresel bilgiyi kapsamaktadır. Bunun dışında büyük bir yama boyutu yeterli bilgiyi kapsamaktadır, fakat hesaplama açısından pahalıdır. Şekil 2.12, belirli bir görüntüden çıkarılan bazı örnek yamaları göstermektedir.

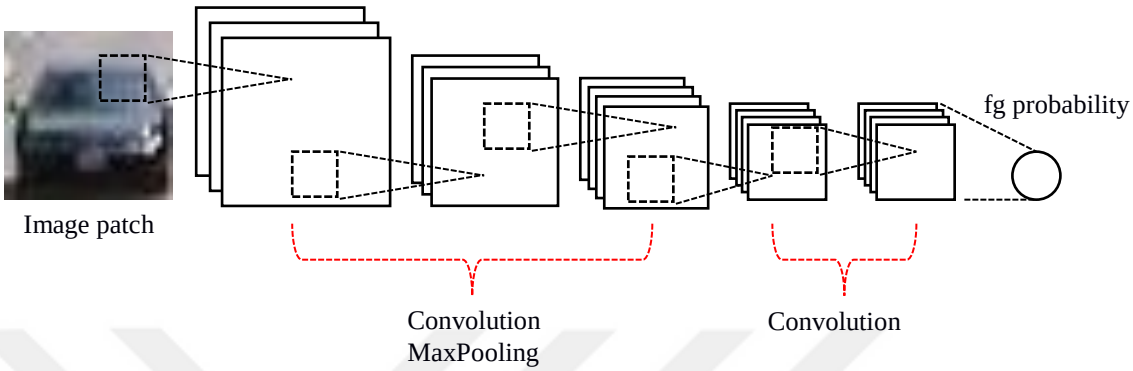
(a) (b)

Şekil 2.12 Bir görüntüden çıkarılan rasgele yamaların görselleştirilmesi
a. ham imge; b. çıkarılan yamalardan rastgele alınan bazı yamaları göstermektedir. Etiket 0, bu yamadaki ortalanmış pikselin bir arka plan pikseli içerdiğini göstermek, etiket 1 ise ortalanmış pikselin bir ön plan pikseli olduğunu belirtmektedir

2.5.1 Patch-wise ağ mimarisi

Patch-wise eğitimi için bir vanilya CNN'ler ağı şekil 2.13'te gösterildiği gibi oluşturulmuştur ve çizelge 2.1'de katmanlar detaylandırılmıştır. BatchNormalization katmanı tarafından takip edilen her bir konvolüsyonel katmanın bulunduğu 5 tane konvolüsyonel katman vardır. Bu ağ, görüntü yamaları alarak görüntü yamalarının merkezli piksellerinin ön plan veya arka plan pikselleri olup olmadığı konusunda bazı

olasılıklar çıkarmaktadır. Not olarak; bu deney için en uygun mimari veya hiper parametreler aranmamıştır.



Şekil 2.13 Vanilya ağ mimarisi
Görüntü yamaları, arka plan veya ön plan pikseli olarak bu yamaların ortalanmış piksellerini tahmin etmek için 5 katmanlı konvolüsyon ağlarına beslenmiştir

Çizelge 2.1 Vanilya patch-wise ağ mimarisinin detayları

Blok	Katman	Açıklama
1	Convolutional	feature=32, filter=3x3, stride=1
	BatchNormalization	momentum=0.99, epsilon=0.001
	ReLU	
	MaxPooling	filter=3x3, strides=2
2	Convolutional	feature=32, filter=3x3, stride=1
	BatchNormalization	momentum=0.99, epsilon=0.001
	ReLU	
	MaxPooling	filter=3x3, strides=2
3	Convolutional	feature=64, filter=3x3, stride=1
	BatchNormalization	momentum=0.99, epsilon=0.001
	ReLU	
	MaxPooling	filter=3x3, strides=2
4	Convolutional	feature=128, filter=3x3, stride=1
	BatchNormalization	momentum=0.99, epsilon=0.001
	ReLU	
5	Convolutional	feature=128, filter=3x3, stride=1
	BatchNormalization	momentum=0.99, epsilon=0.001
	ReLU	
6	Fully connected	feature=1
7	Sigmoid	foreground probability

2.5.2 Eğitim detayları

CDnet2014 veri kümesinde en düşük çözünürlüğe sahip 1700 çerçeve içeren belirli bir video dizisini ele alalım, ör. 320x240. 50-çerçeve deneylerini seçmemiz durumunda patch-wise eğitimi için 320x240x50 ~ 3.8M yama vardır. Depolama açısından idare edilebilmektedir. Fakat, aynı veri kümesinde çerçevelerin büyük çözünürlüğünü düşünmemiz durumunda, ör. 720x576, 720x576x50 ~ 21M yama vardır. Bu yamalar depolama ve işleme için çok büyük miktarda bellek gerektirmektedir.

Bu deneyimizde hesaplama açısından pahalı olmasından dolayı düşük çözünürlüklü bir video dizisi seçilmiştir, ör. 320x240, ve ön çalışma olarak 50-çerçeve deneyleri kullanılmaktadır. Patch-wise eğitimi için CDnet2014 veri kümesinden *highway* video dizisi seçilmiştir. Bu bölümde tartışacağımız iki eğitim stratejisi vardır: bunlar rastgele alt set eğitimi (RST) ve tüm set eğitimi (EST). RST tüm çıkarılan görüntü yamalarından rastgele alt kümeleri kullanarak ağı eğitmeyi ifade etmektedir. EST ise tüm çıkarılan yamaları kullanarak ağı eğitmeyi ifade etmektedir. Üstelik, RST az sayıdaki örnek nedeniyle eğitim süreleri boyunca tüm set eğitiminden daha verimlidir.

Rastgele Alt Set Eğitimi (RST): Bu deneyde tüm çıkarılan yamalarından yamaların rastgele alt setleri rastgele olarak seçilmiştir. İkili sınıflandırmada dengesiz veri probleminin üstesinden gelmek için eğitim verisi ön plan örneklerin sayısı, arka plan örneklerinin sayısına eşit olacak şekilde dengelenmiştir. Bu durumda her bir antrenman çerçevesinin tüm yamaları değiştirilmeden 4,200 ön plan yaması ve 4.200 arka plan yaması rastgele seçilmiştir. Her bir çerçeve için 76.800 yamadan (320x240) rastgele seçilen 8,400 yama vardır. 50-çerçeve deneyleri kullanılarak toplamda 420K yamalar vardır, burada toplam yamaların % 20'si *validasyon* seti için bölünmüştür.

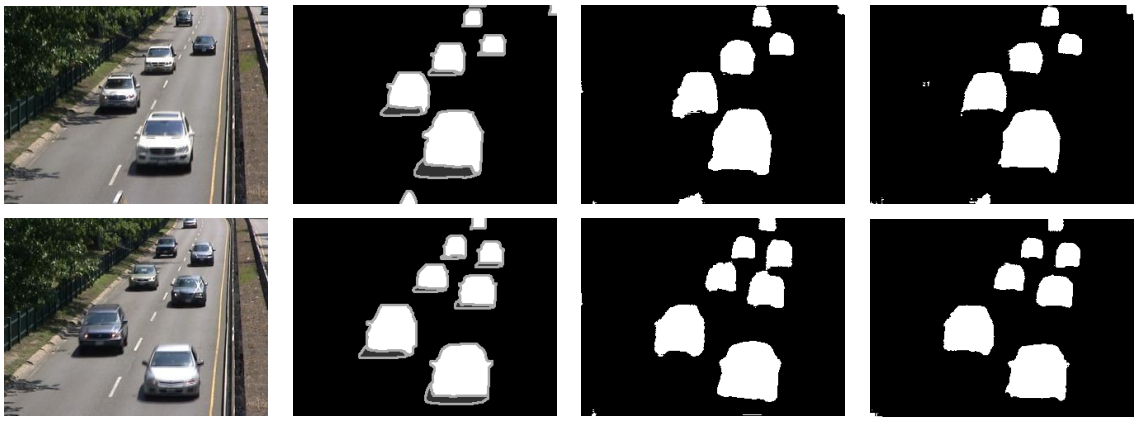
Bu deneyimiz Tensorflow altyapısı (Chollet vd. 2015) ve İntel core i5 CPU ile Mac platformunda Keras framework kullanılarak gerçekleştirilmiştir. Ağımız 1e-3 öğrenme oranı, 24 batch boyutu, 24 epoch'u ve 29 yama boyutuna ayarlanarak eğitilmiştir. Eğitimde 420K yamaları kullanarak yaklaşık 34 dakika sürmüş ve Binary Cross Entropy

kayıp fonksiyonu kullanılmıştır. Not olarak; eğitim sırasında nesnelerin sınırları veya ilgi olmayan bölgeler (veya non-region of interest veya non-ROI) dikkate alınmamıştır.

Tüm Set Eğitimi (EST): Bu deneyde tüm konfigürasyonlar, rastgele alt set eğitimi ile aynı tutulmuştur. Bunun dışında yamaların rastgele alt setlerinin yerine tüm görüntü yamaları eğitilmiştir. Bu durumda non-ROI bırakılarak yaklaşık 3M eğitim yaması vardır ve validasyon seti için yaklaşık 686K yama seçilmiştir. Ağ 22 epoch için eğitilmiştir ve yaklaşık 4.8 saat sürmektedir.

2.5.3 Sonuçlar

Önceki bölümde açıklandığı gibi iki eğitim stratejisi gerçekleştirilmiştir. Şekil 2.14'te her iki stratejinin bazı sonuçlarını göstermektedir. Not olarak; uzun segmentasyon süreleri nedeniyle iki deney için test performansı hakkında herhangi bir istatistik bizim tarafından sağlanmamaktadır (seçilen video dizisi için tek bir CPU kullanarak yaklaşık 4 gün sürmektedir).



Şekil 2.14 Patch-wise eğitiminin bazı sonuçları
a. video çerçeveleri; b. ground-truths; c. *rastgele alt set eğitimi* sonuçları; d. *tüm set eğitim*i sonuçları

Şekil 2.14'te görülebileceği gibi rastgele alt küme yamalarıyla eğitilen model (kolon c) uygun sonuçlar vermiştir. Ancak, nesne sınırları tüm set eğitimine göre pürüzsüz değildir (kolon d). Bu problem, tüm set eğitiminden 8.7x daha az olan rastgele eğitim setinin sayısından kaynaklı olabilir.

Bu bölümde ön çalışma olarak patch-wise öğrenmesi tartışılmıştır. Ancak, hafıza, segmentasyon süreleri ve hesaplama açısından pahalıdır. Bir sonraki bölümde yama-tabanlı (patch-wise) öğrenmeden daha doğru ve verimli olan imge-tabanlı (image-wise) öğrenme yaklaşımı tartışılacaktır.

2.6 Imge Tabanlı (Image-wise) Öğrenme

Bu çalışmada, hareketli nesnelerin segmentasyon problemine üç çözüm önermekteyiz: (1) üçlü CNN (çoklu-girişler) yapısının sonuna eklenmiş transpoze edilen konvolüsyonel sinir ağı (TCNN) içeren enkoder-dekoder yapısının kullanılması, (2) tek girişli CNN'nin ardından Feature Pooling Modülü (FPM)'nün ve TCNN'nin kullanılması, (3) tek girişli bir CNN'nin arkasından değiştirilen FPM (M-FPM)'nün ve yeni bir dekoder ağınının kullanılması. Kurgulanan Foreground Segmentation Netwok, kısaca FgSegNet olarak adlandırılmaktadır.

İlk yaklaşımda, Önceden eğitilmiş VGG-16 Net'in (Simonyan ve Zisserman 2014) ilk dört bloğunu CNN'lerimizin başlangıcında çok ölçekli öznitelik enkoderimiz olarak üçlü bir iskelet altında uyarlanmaktadır ve bu enkoderden çıkarılan özniteliklerden piksel seviyesinde ön plan olasılık haritasına dönüştürmek için enkoderin sonundaki yeni bir dekoder ağını entegre etmektedir. Bu yöntem FgSegNet_M (Multi-inputs) olarak adlandırılmıştır.

İkinci yaklaşımda, aynı enkoder-dekoder yapısını ilk yaklaşımda olduğu gibi kullanarak FPM modülünün de etkili olduğu ve üçlü ağ ile karşılaştırılabilir sonuçlar ürettiği gösterilmektedir. Bu yönteme FgSegNet_S (Single-input) adı verilmektedir.

Üçüncü yaklaşımda, aynı enkoder yapısını ikinci çözümdeki gibi kullanarak FPM modülü içindeki çok ölçekli öznitelikleri birleştirerek bu modül daha da geliştirilmektedir. Bu değiştirilen FPM (M-FPM), çok-ölçekli girişlerin ağa olan ihtiyacını azaltabilen, kamera hareketine karşı sağlam bir öznitelik ayıklaması sağlamaktadır. Daha fazla performans geliştirmesi için M-FPM'nin üzerine yeni bir dekoder de önerilmektedir. Bu yönteme FgSegNet_v2 (versiyon 2) adı verilmektedir.

Önerilen yöntemler önceki yaklaşımlara göre basittir ve çok daha az sayıda eğitim örneğini, yani 200, 50 veya daha azını gerektirmekte, etkileyici segmentasyon sonuçları üretmektedir.

Bizim yöntemlerimiz piksel seviyesinde ground-truth'lar içeren halka açık en büyük CDnet2014 (Wang vd. 2014), SBI2015 (Maddalena ve Petrosino 2015) ve UCSD Background Subtraction (Mahadevan ve Vasconceloset 2010) veri kümesi ile değerlendirilmiştir. Test sonuçları metodumuzun ortalama F-Measure ve ortalama MCC açısından önceki en iyi yöntemi önemli ölçüde geliştirdiğini ortaya koymaktadır.

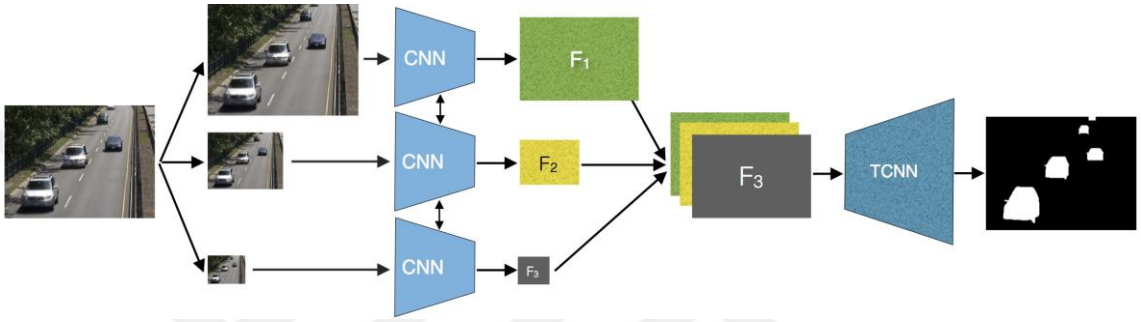
Bizim uygulamamız bir NVIDIA GTX 970 GPU ile Tensorflow tabanlı Keras iskeleti (Chollet vd. 2015) kullanılarak gerçekleştirilmiştir. Bu bölümde bu tezin ana konusu olan FgSegNet ailesinin önemli verisetleri üzerindeki performansı hakkında ayrıntılı bilgi verilecektir.

2.6.1 FgSegNet_M ve FgSegNet_S

2.6.1.1 FgSegNet_M ağ mimarisi

Öznitelik kodlaması için üç farklı ölçekte çalışan üçlü CNN ve kod çözmesi için dönüştürülmüş bir konvolüsyonel ağ içeren uçtan uca ağ mimarimiz şekil 2.15'te gösterilmektedir. Bu ağ ham piksel değerlerin bir setini $\boldsymbol{P^R}$ olasılık değerlerin bir setine dönüştüren bir $\boldsymbol{f}$ fonksiyonunu öğrenmektedir, yani ön plan olasılık haritasını $\boldsymbol{P^M}$ temsil eden 0 ile 1 arasındaki değerler. Bu fonksiyon şu şekilde tanımlanmaktadır: $\boldsymbol{f : P^R \rightarrow P^M}$. Bu dönüştürülen fonksiyonunu ($\boldsymbol{f}$) doğru bir şekilde öğrenmek için her pikselin

komşusu etrafındaki bağlamsal bilgiler önemlidir. Bir pikselin üzerinde ortalanmış küçük bir sabit pencereden sınıflandırılmasının öğrenilmesi zordur. Etrafındaki benzer yoğunlukları paylaşan düz, büyük bir bölgenin kediye ait olduğunu ve çok küçük bir bölgesi sınıflandırmak istediğimizi düşünelim: küresel içerik olmadan bu yerel içeriği analiz ederek kedinin bir parçası olup olmadığını söylemek zordur. Bu yerel bölgenin çevresiyle olan bağlamsal ilişkisini anlamak için ağın küresel bilgileriyle çoklu ölçeklerde bağlanması gerekmektedir.



Şekil 2.15 FgSegNet_M Mimarisi

Bu fikirler, ağ eğitiminde tam boyutlu ve çok ölçekli görüntüleri kullanmamıza ilham vermiştir. Bu ölçeklerde sabit bir alıcı alan (veya receptive field) çalıştırılmaktadır. Şekil 2.15'te gösterildiği gibi, RGB renk uzayında temsil edilen giriş görüntüsünün çözünürlüğü, aşağıda gösterilen Gauss dağılımının % 99'undan fazlasını kapsayan bir sigma ile bir Gauss piramidi kullanılarak iki faktör azaltılmaktadır:

$$sigma = \frac{downscale}{3}$$

burada ***downscale*** görüntünün çözünürlüğü azaltılan faktördür, bunu uygulamamızda 2'ye ayarlanmıştır. Bir giriş görüntüsü $\boldsymbol{I}$ verildiğinde, bu görüntünün çözünürlüğü $\boldsymbol{I_i} : i \in [0, 1, 2]$'ye düşürülmektedir; burada $\boldsymbol{I_0}$ görüntünün orijinal boyutudur. Bu üç görüntü eşzamanlı olarak üçlü CNN'imize paralel olarak beslenmektelerdir. Not olarak; üçlüdeki CNN'lerin mimarisi tam olarak aynıdır ve ağırlıklarını paylaşmaktadırlar (mimarın detayları için bölüm 2.6.1.1.1'e bakınız). Her giriş görüntüsünden elde edilen katıştırmalar $\boldsymbol{F_i} : i \in [1, 2, 3]$ ile belirtilmektedir, burada $\boldsymbol{F_1}$, $\boldsymbol{F_2}$ ve $\boldsymbol{F_3}$ sırasıyla $\boldsymbol{I_0}$, $\boldsymbol{I_1}$ ve $\boldsymbol{I_2}$ girişlerinin katıştırmalarıdır. Daha sonra bu öznitelik katıştırmaları, kod çözme ağının

birleşik öznitelik sunumunu oluşturmak için yeniden düzenlenmektedir. Bu bağlamda F_2 ve F_3, F_1'in ölçeğiyle eşleşecek şekilde en yakın komşu enterpolasyonunu kullanarak büyütülmek ve birleşik öznitelik haritasını (F) oluşturmak için bunlar derinlik ekseni boyunca birleştirilmektedir. En sonunda; F, kod çözme ağırlıklarını öğrenmek için tek bir TCNN'yi beslenmektedir. Son çıktı sonucu (M_{out}), orijinal girişle (I_0) aynı boyuta sahip bir segmentasyon maskesidir. Farklı giriş ölçeklerini denenmiş ancak 1 veya 2 giriş ölçeği biraz zayıf performansa yol açmıştır (muhtemelen daha az bağlamsal bilgiyi kapsadığı için), daha fazla girdi ölçeği ise gerekli hesaplamaların artmasına yol açmıştır. Kodlama ve kod çözme ağ konfigürasyonumuzun detayları aşağıda verilmektedir.

Çizelge 2.2 FgSegNet_M ve FgSegNet_S encoder-dekoder ağ konfigürasyonumuz

	CNNs (VGG-16)			TCNN	
0	WxHx3	rgb image	9	WxHx1	F=1x1,S=1, seg.mask
1	WxHx64 WxHx64 max-pool.	F=3x3,S=1 F=3x3,S=1 F=2x2,S=2	8	WxHx64	F=5x5,S=2
2	$\frac{W}{2}$x$\frac{H}{2}$x128 $\frac{W}{2}$x$\frac{H}{2}$x128 max-pool.	F=3x3,S=1 F=3x3,S=1 F=2x2,S=2	7	$\frac{W}{2}$x$\frac{H}{2}$x128 $\frac{W}{2}$x$\frac{H}{2}$x64 $\frac{W}{2}$x$\frac{H}{2}$x64	F=1x1,S=1 F=3x3,S=1 F=1x1,S=1
3	$\frac{W}{4}$x$\frac{H}{4}$x256 $\frac{W}{4}$x$\frac{H}{4}$x256 $\frac{W}{4}$x$\frac{H}{4}$x256	F=3x3,S=1 F=3x3,S=1 F=3x3,S=1	6	$\frac{W}{2}$x$\frac{H}{2}$x256 $\frac{W}{2}$x$\frac{H}{2}$x64 $\frac{W}{4}$x$\frac{H}{4}$x64	F=1x1,S=1 F=5x5,S=2 F=1x1,S=1
4	$\frac{W}{4}$x$\frac{H}{4}$x512 dropout $\frac{W}{4}$x$\frac{H}{4}$x512 dropout $\frac{W}{4}$x$\frac{H}{4}$x512 dropout	F=3x3,S=1 rate=0.5 F=3x3,S=1 rate=0.5 F=3x3,S=1 rate=0.5	5	$\frac{W}{4}$x$\frac{H}{4}$x512 $\frac{W}{4}$x$\frac{H}{4}$x64 $\frac{W}{4}$x$\frac{H}{4}$x64	F=1x1,S=1 F=3x3,S=1 F=1x1,S=1

Değiştirilmiş bir CNN'ler (VGG-16) blok 1'den 4'e kadardır, burada blok 0 RGB giriş görüntüsüdür. TCNN blok 5'den 9'a kadardır, burada blok 9 ağdan çıkış olasılığı maskesidir

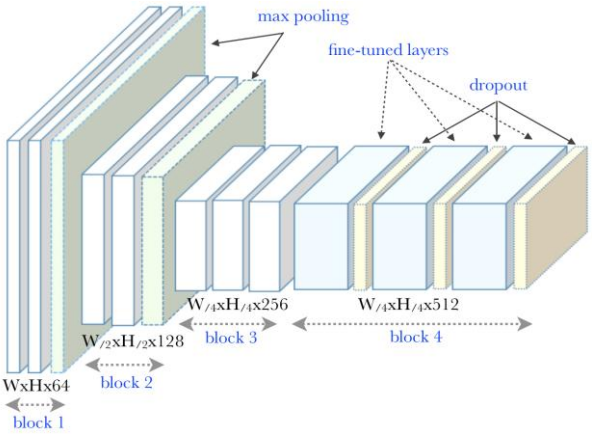
2.6.1.1.1 Üçlü CNN konfigürasyonu

CNN'ler farklı alanlardaki çeşitli problemlerde insan performansından daha iyi performans göstermiştir. CNN'lerin öğrendikleri hakkında daha derin bir fikir edinmek için her katmanda öğrenilen filtrelerin görselleştirmeleri denetlenebilmektedir. Bu görselleştirmeler, alt katmanların öznitelik temsilleri olarak kullanıldığında birçok görevde yararlı olan renk lekeleri, çeşitli yönlerdeki kenarlar ve dokular gibi bazı düşük düzeyli öznitelikleri öğrenmiş olduğunu göstermiştir (Zeiler ve Fergus 2014). CNN'lerin bu genel özellik kodlama özelliklerinin yararlılığı nedeniyle, üç farklı ölçekte aynı girdiye paralel olarak çalışan bir CNN'nin üç kopyasını içeren bir üçlü CNN kullanılmaktadır. Bu ağların ilk dört bloğu, önceden eğitilmiş VGG-16 Net'in değiştirilmiş kopyalarıdır; yani üçüncü ve dördüncü maksimum pooling katmanları çıkarılmış ve şekil 2.16'da gösterildiği gibi dördüncü konvolüsyon bloğunun her bir katmanı arasında dropout'lar eklenmiştir (VGG-16 Net'in tam ağ mimarisi için Simonyan ve Zisserman (2014) orijinal makalesine bakılabilir).

Her CNN girişi farklı boyutlarda ham RGB görüntülerdir. Giriş görüntü boyutunun WxHx3 olduğunu varsayalım: burada W görüntü genişliğidir, H görüntü yüksekliğidir ve 3 RGB renk kanallarıdır. Bu giriş görüntüsü ilk konvolüsyon bloğunun sonunda WxH boyutunda 64 öznitelik haritasına dönüştürülmektedir, daha sonra bu öznitelik haritaları 2 adımlı bir 2x2 maksimum pooling katmanı ile çözünürlüğü azaltılmakta ve ikinci blok sonunda $W_{/2}xH_{/2}$ boyutunda 128 öznitelik haritasına dönüştürülmektedir. Yine, bu öznitelik haritaları 2 adımlı bir 2x2 maksimum pooling katmanı ile çözünürlük azaltılmakta ve üçüncü blok sonunda $W_{/4}xH_{/4}$ boyutunda 256 öznitelik haritasına dönüştürülmektedir. Son olarak, bu öznitelik haritaları dördüncü bloğun sonunda $W_{/4}xH_{/4}$ boyutunda 512 öznitelik haritasına dönüştürülmektedir.

Segmentasyon yaklaşımımızda model oluşturması için sadece birkaç eğitim örneği kullanılmaktadır. Bu yüzden, ezberin (overfitting) önlenmesi için dördüncü konvolüsyon bloğunda her bir konvolüsyon katmanından sonra dropout regularization'u uygulanmaktadır. Not olarak; çıkışlarımızdaki girişlerin uzamsal boyutlarını korumak için ağımızdaki tüm konvolüsyon katmanlara zero-padding uygulanmaktadır. Kodlama

ağı konfigürasyonunun detayları çizelge 2.2'de sunulmaktadır.



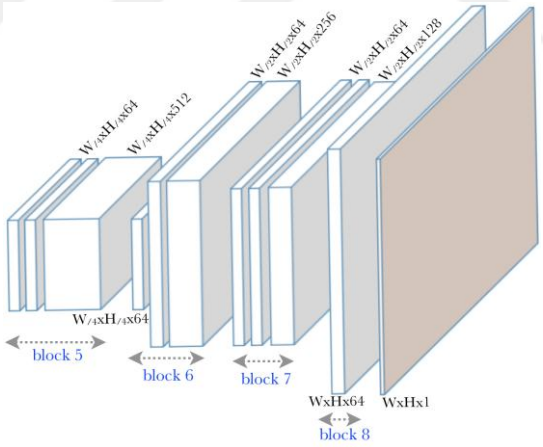
Şekil 2.16 Üçlü ağdaki her CNN'nin mimarisi

2.6.1.1.2 TCNN konfigürasyonu

Kodlama ağının çıktısı (***F***) üç farklı ölçekte öznitelik haritalarının birleştirilmiş formudur. Bu birleştirilmiş öznitelik haritası, öznitelik haritaları kod çözmek için ağırlıkları öğrenmek üzere TCNN'e beslenmektedir. Bu kod çözme işleminin çıktısı yoğun bir olasılık maskesi olacaktır (Şekil 2.17). Ağımızda ***F*** üç farklı ölçekte özniteliklerin birleştirilmesi nedeniyle büyük bir derinliğe sahiptir (1536 öznitelik). Hesaplama verimliliği için ve ağımızdaki karar fonksiyonunun lineer olmayanlığını arttırmak için her bir blokta yüksek boyutlu bir öznitelik haritası derinliğini daha düşük bir boyuta yansıtmak için 1x1 dönüştürülmüş katmanlar kullanılmaktadır.

Çizelge 2.2'nin sağ alt satırında ayrıntılı olarak belirtilen TCNN'deki blok 5'i dikkate alırsak $W_{/4}xH_{/4}x1536$ biçimin birleştirilmiş ***F*** özelliği, 1 adımlı bir 1x1 dönüştürülmüş konvolüsyon kullanılarak $W_{/4}xH_{/4}x64$'e yansıtılmaktadır. Yansıtılan öznitelikler 1 adımlı bir 3x3 dönüştürülmüş konvolüsyon ile çalıştırılmış ve WxHx64'e yansıtılmaktadır. En sonunda, bu öznitelikler derinlik ekseni boyunca öznitelik haritalarının sayısını büyütmek için $W_{/4}xH_{/4}x512$'ye yansıtılmaktadır. Katmanların benzer yapıları blok 6 ve 7'de çalıştırılmaktadır, ancak blok 6'daki iki faktörle öznitelik

haritalarının çözünürlüğünü arttırmak için 2 adımlı bir 5x5 dönüştürülmüş konvolüsyon uygulanmaktadır. Üstelik, öznitelik haritalarının sayısı sırasıyla 6 ve 7 blokları için 256 ve 128'e azaltılmaktadır. Blok 8'de giriş görüntüsünün orijinal boyutuna uyacak şekilde öznitelik haritalarını büyütmek için 2 adımlı bir 5x5 dönüştürülmüş konvolüsyon kullanılmaktadır. Block 9'da 1 adımlı bir 1x1 dönüştürülmüş konvolüsyon çalıştırarak blok 8'in 64 öznitelik haritası 1 öznitelik haritasına yansıtılmaktadır. En sonunda, her bir piksel için bir ön plan pikseli olma olasılığını 0 ve 1 arasında bir değerle kodlamak için bir olasılık maskesi oluşturmak üzere son katmana bir sigmoid fonksiyonu uygulanmaktadır. Not olarak; bir olasılık maskesinin tahmin edilmesi için bir sigmoid aktivasyonunun kullanıldığı son dönüştürülmüş konvolüsyonel katman dışında ReLU non-linearty, her iki modifiye VGG-16 Net ve TCNN'deki her (transpoze) konvolüsyonel katmana uygulanmaktadır. Dropout dışında ağımızdaki overfitting'i azaltmak için 5, 6, 7 ve 8 numaralı bloklarda ilk dönüştürülmüş konvolüsyonel katmanlarda ağırlıklara L2 regularization'u uygulanmaktadır. Sonraki bölümde ikinci ağ mimarisi hakkında tartışılacaktır.



Şekil 2.17 TCNN mimarisi
TCNN, öznitelik uzayından görüntü uzayına dönüştürmektedir. Sayısal karmaşıklığı azaltmak için bu dekoder bölümünde point-wise konvolüsyonu (veya 1x1 konvolüsyonu) kullanılmaktadır

2.6.1.2 FgSegNet_S ağ mimarisi

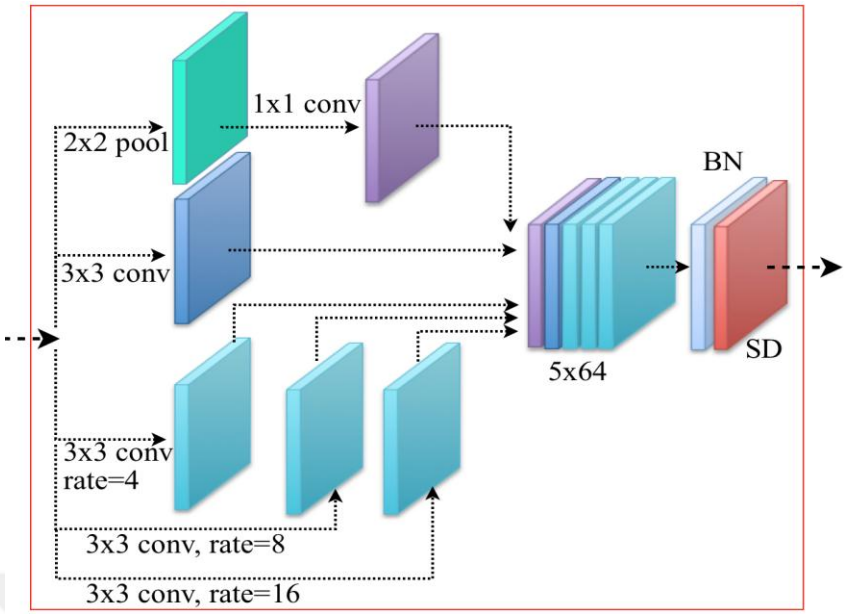
Bu bölümde FgSegNet_S adlı ikinci metodu hakkında tartıştılmaktadır.

2.6.1.2.1 Enkoder ve dekoder ağı

Bu yöntemde, dekoder aynı kalırken enkoder olarak üçlü bir ağ (veya çoklu girişler) kullanmak yerine tek bir CNN (veya tek giriş) kullanılmaktadır. Çoklu ölçek özniteliklerini çıkartmak için tek bir enkoderın üstüne takılabilen Feature Pooling Modülü (FPM) önerilmiştir. Bu fikir bir sonraki bölümde tartışılacaktır.

2.6.1.2.2 Feature Pooling Modülü

Çoklu ölçeklerdeki öznitelikleri ayıklamak için son CNN çıktısının üstünde çalışan bir Feature Pooling Module (FPM) (Şekil 2.18) önermekteyiz. Not olarak FgSegNet_S için çok ölçekli girişler kullanmak yerine tek bir ölçek girdisi kullanılmaktadır. Genişlemiş konvolüsyon (veya Atrous konvolüsyonu) başarıyla uygulanmış ve semantik segmentasyon alanında umut verici sonuçlar elde edilmiştir (Yu ve Koltun 2015, Chen vd. 2017, Chen vd. 2018). Atrous konvolüsyonun fikri, ek parametre öğrenmeden ağın görüş alanını arttırmaktır. Aynı fikri uyarlayarak maksimum pooling katmanı ve farklı genişleme oranlarına sahip birkaç paralel genişlemiş konvolüsyon uygulayan bir öznitelik piramit pooling modülü tasarlanmaktadır. Şekil 2.18'de verilen öznitelik haritaları F; 1x1 konvolüsyon takip edilen 2x2 maksimum pooling, normal 3x3 konvolüsyon ve 4, 8 ve 16'lık genişleme oranlarına sahip üç 3x3-genişlemiş konvolüsyonlar, aynı öznitelikler F üzerinde çalışmaktadır. Elde edilen öznitelikler, derinlik ekseni boyunca birleştirilmektedir ve ardından BatchNormalization (Ioffe ve Szegedy 2015) ve 0.25 oranıyla SpatialDropout (Tompson vd. 2015) takip etmekte ve daha sonra TCNN'den geçirilmektedir. Not olarak maksimum pooling ve tüm konvolüsyon katmanlarınada adım 1 olarak seçilmiş ve BatchNormalization'dan hemen sonra ReLU uygulanmaktadır.



Şekil 2.18 FPM modülü
BN (BatchNormalization), SD (SpatialDropout). Tüm konvolüsyon katmanlarda 64 öznitelik vardır

2.6.1.3 Eğitim detayları

Şekil 2.16 ve çizelge 2.2'de gösterildiği gibi eğitim sırasında ilk VGG-16 Net modelinde olduğu gibi, 1, 2 ve 3 numaralı konvolüsyonel blokların ağırlıkları tutulmakta ve 4 numaralı konvolüsyon bloğunun ağırlıkları uyarlanmaktadır. Ön plan segmentasyonunda Binary Cross Entropi kayıp fonksiyonu kullanılmaktadır. Not olarak; eğitim sırasında kayıp hesaplarımızda hiçbir etiketi, ROI olmayan ve nesnelerin sınırı gibi bilinmeyen bölgelerle ilişkilendirilmemektedir. Bu kasıtlı kaçmanın ağımızı piksel tahmininde daha emin hale getirdiği gözlemlenmiştir.

Ağımız, 1 batch-size ile *RMSProp* kullanılarak sırasıyla 0.9 ve 1e-08 olan *rho* ve *epsilon* değerleri ile eğitilmiştir. İnce ayarda halihazırda iyi olan öznitelik temsillerinin mevcut ağırlıkları değiştirmek istemediğimizden küçük bir öğrenme oranı (1e-4) kullanılmaktadır. Dikkat edilmesi gereken önemli bir nokta, video dizilerinin doğası nedeniyle ağa sıralı bir sırada okuma ve besleme yaparak eğitim çerçevelerinin öğrenilen ağırlıklarda bir yanlılığa yol açabileceğidir, çünkü bir satırdaki birçok çerçeve çok benzer içerik barındırmaktadır. Pratikte bu sorunu önlemek için iki aşamada rastgele karıştırma yapılmaktadır: bunlar eğitim/validasyon setine bölmeden önce

eğitim çerçevelerinin rastgele karıştırılması ve eğitim sürecinde her epoch'dan önce rastgele karıştırılmasıdır. Ağımızın bu işlemlerden yarar sağladığını ve daha hızlı bir şekilde yakınsadığını gözlemlenmektedir. Karıştırdıktan sonra % 20'lik bir validasyon bölmesi (veya validation split) gerçekleştirilmektedir; bu nedenle, eğitim örneklerinin % 80'i modeli eğitmek için kullanılmaktadır.

Bölüm 2.4'te açıklandığı gibi eğitim sırasında bir ön plan pikselinin arka plan piksel olarak sınıflandırılması durumunda kayıp cezalandırılmaktadır. Bu teknik ağ performansının bazı marjlarla geliştirilmesine yardımcı olmaktadır. Not olarak; eğitim sırasında ortalama çıkarmalar dahil herhangi bir giriş normalleştirmesi gerçekleştirilmemektedir. Deneylerimizde modelimiz, N sabit ve 25 (50 veya 200) olan N eğitim örneği kullanılarak eğitilmektedir.

Minimum validasyon kaybı 6 epoch içinde geliştirmeyi durdurduğunda öğrenme oranı 10 kat azaltılmaktadır. Ağımız 50 eğitim örneği için 60 epoch ve 200 eğitim örneği içinse 50 epoch kullanılarak eğitilmektedir. Validasyon kaybı azaldığında model kontrol noktası (veya model checkpoint) en iyi modeli kaydetmek için kullanılmaktadır; yani uygulamada model kontrol noktasının bizim için en iyi modeli seçmesine izin verilmektedir. Deneylerimizde L2 regularization'unun gücünü 5e-4 olarak ayarlanmaktadır.

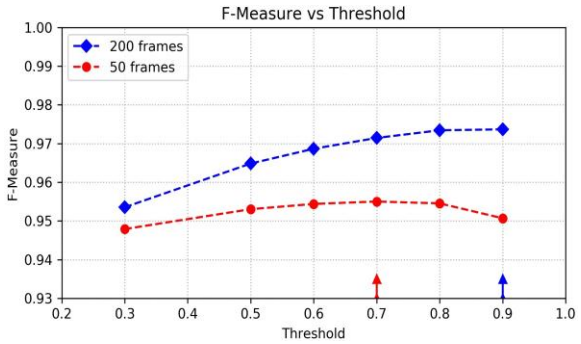
Ağımızda FgSegNet_M için 6.5M eğitilebilir parametre ve yaklaşık 1.7M eğitilmeyen parametre içeren toplam 8.2M parametre bulunmaktadır. Toplam parametrelerin yaklaşık % 93'ü VGG-16 Net'ten gelmektedir, ve diğer % 7'si TCNN kısmından gelmektedir. TCNN'deki parametreler, 1x1 transpoze edilmiş konvolüsyon katmanlarının boyut indirgemesinden dolayı önemli ölçüde daha azdır. FgSegNet_S ise 7.6M eğitilebilir parametre ve yaklaşık 1.7M eğitilmeyen parametre ile toplamda 9.4M parametre içermektedir.

Not olarak FgSegNet_S için tek bir girişin kullanılması dışında, aynı eğitim prosedürü hem FgSegNet_M hem de FgSegNet_S için kullanılmaktadır (ayrıntılar için bölüm 2.6.1.2'ye bakınız).

2.6.1.4 Eşikleme (Thresholding)

Ağlerimizin çıktısı her piksel için 0 ile 1 arasında değerler içeren bir olasılık maskesi nedeniyle bu olasılıkları ikili maskelere dönüştürmek için bir eşik kullanılmaktadır. Olasılık belirli bir eşiği aşarsa bu piksele karşılık gelen maske değeri 1 olarak şeçilmektedir. Not olarak; uygulamamızda son segmentasyon maskesinin tutarlılığını sağlamak için koşullu rastgele alan (conditional random field veya CRF) veya başka herhangi bir grafik modelinin kullanımı gibi eşiklemeden sonra post-processing uygulanmamaktadır.

Şekil 2.19'da ağlerımızın (FgSegNet_M) farklı eşik değerlerin bir seti için sınıflandırma performansını göstermektedir. Görülebileceği gibi 200 çerçeve kullandığımız deneyler için 0.9'luk bir eşiğin 11 kategoride en iyi ortalama F-Measure değerini verdiğini göstermektedir. Bu yüksek olasılık, yöntemimizin genel olarak tahminleriyle son derece emin olduğunu göstermektedir. 50 çerçeve durumu ise güven seviyesi biraz azaltmaktaır, yani 0.7'lik bir eşik en iyi skoru vermektedir. Her iki deney için eşiği belirli bir değere sabitlemek için tüm deneylerimizde 0.8 sabit bir eşik değeri seçilmektedir.



Şekil 2.19 Farklı eşiklere karşı 11 kategoride test setindeki ortalama F-Measure'nin bir ilüstrasyonudur
Ok işaretleri en yüksek F-Measure değerini göstermektedir

2.6.1.5 Sonuçlar ve tartışmalar

2.6.1.5.1 CDnet2014 veri setinde

Wang vd. (2017)'de açıklandığı gibi aynı eğitim çerçeve seçim stratejisi (rastgele manuel seçimi) bizim tarafından takip edilmektedir. İlk olarak, el ile seçtiğimiz bir çerçevelerin setini kullanarak deneyler gerçekleştirilmektedir, ve yalnızca ground-truth etiketlerini içeren çerçevelerin aralığını dikkate alarak test sonuçları rapor edilmektedir. Bu deneylerin sonuçları çizelge 2.3'te gösterilmektedir. Not olarak; bu değerler sadece test çerçeveleri kullanılarak hesaplanmaktadır, yani performans değerlendirmesinde eğitim çerçeveleri hariç tutulmaktadır. Bu ayarda FgSegNet_M için 50 çerçeve deneyleri ile 0.9545 ve 200 çerçeve deneyleri ile 0.9734 bir genel ortalama F-Measure elde edilmiştir. FgSegNet_S ise 50 çerçeve deneyleri ile 0.9633 ve 200 çerçeve deneyleri ile 0.9775 bir genel ortalama F-Measure elde edilmiştir. Karşılaştırıldığında FgSegNet_M'e göre FgSegNet_S, 50-kare deneyi için % 0.88 ve 200-kare deneyi için % 0.41 puan ilerletmektedir.

Çizelge 2.3'te görüldüğü gibi 200-çerçeve deneylerinde ağımız ön plan segmentasyonunda yüksek doğruluk sağlamaktadır. *Baseline* kategorisinde diğer kategorilerle karşılaştırma en yüksek ortalama F-Measure elde edilmektedir. Eğitim örneklerinin sayısını 50 kareye indirerek F-Measure bazı puanlar ile azalmaktadır. Özellikle *lowFrameRate* kategorisinde 200 eğitim örneğine sahip olan modelle karşılaştırma F-Measure FgSegNet_M için % 6.5 ve FgSegNet_S için % 1.11 azalmaktadır. Halbuki şu anki teknoloji harikası yöntemlerinden daha iyi çalışan 11 kategoride toplamda FgSegNet_M için 0.9545'lük ve FgSegNet_S için 0.9633'lük ortalama F-Measure ile kabul edilebilir sonuçlar elde edilmektedir. Sonuç olarak, yöntemimizin birçok zorlu ön plan nesnelerinin segmentasyonu alanlarında sağlam bir şekilde çalıştığını göstermektedir.

Çizelge 2.3 CDnet2014 veri kümesinden el ile ve rasgele olarak 50 ve 200 kare seçerek elde edilen sonuçlar

Category	FgSegNeg_M								FgSegNeg_S							
	Recall		Precision		PWC		**F-Measure**		Recall		Precision		PWC		**F-Measure**	
	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f	50f	200f
baseline	0.9887	0.9951	0.9964	0.9986	0.0405	0.0152	**0.9926**	**0.9968**	0.9922	0.9966	0.9962	0.9983	0.0333	0.0126	**0.9942**	**0.9975**
cam. jitter	0.9702	0.9878	0.9906	0.9950	0.1696	0.0605	**0.9801**	**0.9914**	0.988	0.9919	0.9856	0.9923	0.1084	0.0565	**0.9872**	**0.9921**
bad weath.	0.9484	0.9759	0.9805	0.9755	0.1180	0.0494	**0.9636**	**0.9757**	0.9787	0.9825	0.9798	0.9876	0.0546	0.0286	**0.9792**	**0.9850**
dyna. bg.	0.9826	0.9906	0.9883	0.9826	0.0208	0.0071	**0.9854**	**0.9865**	0.9895	0.9940	0.9761	0.9845	0.0166	0.0056	**0.9825**	**0.9892**
inter. obj.	0.9670	0.9889	0.9833	0.9956	0.1295	0.0823	**0.9749**	**0.9922**	0.9890	0.9897	0.9332	0.9969	0.1883	0.0730	**0.9546**	**0.9932**
low f.rate	0.8374	0.8990	0.8049	0.8687	0.1000	0.0336	**0.8164**	**0.8816**	0.8645	0.9126	0.8736	0.8584	0.0736	0.0282	**0.8690**	**0.8801**
night vid.	0.8817	0.9606	0.9671	0.9788	0.2740	0.0992	**0.9216**	**0.9696**	0.9506	0.9768	0.9515	0.9751	0.1872	0.0747	**0.9509**	**0.9759**
PTZ	0.9350	0.9755	0.9779	0.9417	0.0523	0.0164	**0.9557**	**0.9567**	0.9789	0.9889	0.9186	0.9646	0.0528	0.0125	**0.9439**	**0.9760**
shadow	0.9839	0.9922	0.9768	0.9966	0.1211	0.0374	**0.9800**	**0.9944**	0.9926	0.9953	0.9838	0.9967	0.0836	0.0271	**0.9881**	**0.9960**
thermal	0.9598	0.9871	0.9859	0.9944	0.2042	0.0683	**0.9725**	**0.9907**	0.9819	0.9906	0.9796	0.9953	0.1337	0.0464	**0.9807**	**0.9929**
turbulence	0.9443	0.9675	0.9704	0.9772	0.0426	0.0264	**0.9571**	**0.9722**	0.9722	0.9792	0.9606	0.9696	0.0362	0.0239	**0.9663**	**0.9743**
Overall	**0.9454**	**0.9746**	**0.9656**	**0.9732**	**0.1156**	**0.0451**	**0.9545**	**0.9734**	0.9708	**0.9817**	**0.9581**	**0.9745**	**0.0880**	**0.0354**	**0.9633**	**0.9775**

Her satır, her bir kategorinin ortalama sonuçlarını göstermektedir. Son satır, 11 kategoride ortalama sonuçları göstermektedir. Not olarak; rapor edilen performanslara sadece test çerçeveleri dahil edilmektedir

Önceki yöntemler tüm çerçeveleri içermesi nedeniyle sonuçlarımızı önceki yöntemlerle karşılaştırmak için performans değerlendirmelerinde tüm çerçeveleri dikkate almamız gerekmektedir. Her bir kategori için farklı metotların F-Measure ve MCC performansları sırasıyla çizelge 2.4-2.5'te verilmektedir. 11 kategorideki genel performanslar çizelge 2.6'da gösterilmektedir. Derin öğrenme tabanlı yöntemler, *nightVideo* ve *PTZ* gibi çok zorlu kategorilerde daha iyi performans göstermektedir; ancak diğer kategorilere göre *lowFrameRate* kategorisinde düşük performans göstermektedir. *lowFrameRate* kategorisinde iyi performans göstermemektedir. Derin öğrenme yöntemlerinin çoğu hala geniş marjlarla geleneksel yaklaşımlardan daha iyi performans göstermektedir.

Eğitim örneklerini hariç tutarak skorların karşılaştırması çizelge 2.8'te verilmektedir. Burada metodumuzla karşılaştırmak için sadece CDnet2014'teki mevcut en iyi yöntem seçilmiştir. Bu çalışma için çerçeve seviyesi ön plan maskeleri ve eğitim çerçeveleri halka açık olması nedeniyle adil karşılaştırmalar yapabilmektedir.

Bu deneylere ek olarak Wang vd. (2016)'teki yazarlar tarafından sağlanan eğitim çerçevelerini kullanarak modelimizi eğittiğimiz (yalnızca dört sahneyi ayarlayarak) ve Change Detection 2014 yarışmasından elde ettiğimiz sonuçları çizelge 2.7'de rapor edilmiş ek deneyler de gerçekleştirilmektedir. Bu sonuçlar, yöntemlerin bütün olarak veri seti ile performansını göstermektedir, yani ground-truth değerlerinin açık veri setiyle paylaşılmadığı ek kareleri de içermektedir. Yöntemimiz bugünkü teknoloji harikası yöntemlerinden daha iyi çalışmaktadır, dolayısıyla Change Detection 2014 yarışması web framework performans değerlendirmelerinde birinci sırada yer almaktadır.

Çizelge 2.4 F-Measure'nin 7 yöntem arasında 11 kategoride bir kategori-wise karşılaştırılması

Methods	**F-Measure (category-wise)**										
	baseline	cam.jitter	bad.weat	dyna.bg	int.obj.m.	low f.rate	night vid.	PTZ	shadow	thermal	turbul.
FgSegNet_S	**0.9980**	**0.9951**	**0.9902**	**0.9952**	**0.9942**	**0.9511**	**0.9837**	**0.9880**	**0.9967**	**0.9945**	**0.9796**
FgSegNet_M	**0.9975**	**0.9945**	**0.9838**	**0.9939**	**0.9933**	**0.9558**	**0.9779**	**0.9893**	**0.9954**	**0.9923**	**0.9776**
Cascade	0.9786	0.9758	0.9451	0.9658	0.8505	0.8804	0.8926	0.9344	0.9593	0.8958	0.9215
DeepBS	0.9580	0.8990	0.8647	0.8761	0.6097	0.5900	0.6359	0.3306	0.9304	0.7583	0.8993
IUTIS-5	0.9567	0.8332	0.8289	0.8902	0.7296	0.7911	0.5132	0.4703	0.9084	0.8303	0.8507
PAWCS	0.9397	0.8137	0.8059	0.8938	0.7764	0.6433	0.4171	0.4450	0.8934	0.8324	0.7667
SuBSENSE	0.9503	0.8152	0.8594	0.8177	0.6569	0.6594	0.4918	0.3894	0.8986	0.8171	0.8423

Her satır her yöntemin sonuçlarını göstermektedir. Her sütun her kategorideki ortalama sonuçları göstermektedir. Not olarak; CDnet2014 veri kümesindeki ground-truth'ların tüm kareleri dikkate alınmaktadır

Çizelge 2.5 MCC'nın 7 yöntem arasında 11 kategoride bir kategori-wise karşılaştırılması

Methods	MCC (category-wise)										
	baseline	cam.jitter	bad.weat	dyna.bg	int.obj.m.	low f.rate	night vid.	PTZ	shadow	thermal	turbul.
FgSegNet_S	**0.9979**	**0.9948**	**0.9900**	**0.9951**	**0.9938**	**0.9515**	**0.9834**	**0.9880**	**0.9966**	**0.9942**	**0.9796**
FgSegNet_M	**0.9975**	**0.9942**	**0.9836**	**0.9938**	**0.9929**	**0.9557**	**0.9774**	**0.9892**	**0.9952**	**0.9920**	**0.9775**
Cascade	0.9780	0.9748	0.9443	0.9658	0.8591	0.8798	0.8911	0.9349	0.9577	0.8932	0.9230
DeepBS	0.9571	0.8976	0.8718	0.8777	0.6371	0.6061	0.6617	0.3701	0.9284	0.7609	0.9024
IUTIS-5	0.9553	0.8274	0.8333	0.8932	0.7406	0.7943	0.5182	0.5002	0.9060	0.8328	0.8584
PAWCS	0.9375	0.8121	0.8120	0.8936	0.7737	0.6573	0.4327	0.4911	0.8875	0.8282	0.7857
SuBSENSE	0.9487	0.8080	0.8596	0.8240	0.6738	0.6860	0.4998	0.4442	0.8958	0.8098	0.8448

Not olarak; CDnet2014 veri kümesindeki ground-truth'ların tüm kareleri dikkate alınmaktadır

Çizelge 2.6 Her bir yöntem için 11 kategoride ortalama sonuçlar

Methods	Overall				
	Precision	Recall	PWC	F-Measure	MCC
FgSegNet_S	**0.9864**	**0.9895**	**0.0327**	**0.9878**	**0.9877**
FgSegNet_M	**0.9889**	**0.9841**	**0.0426**	**0.9865**	**0.9863**
Cascade	0.9048	0.9584	0.3882	0.9272	0.9274
DeepBS	0.8401	0.7650	1.8699	0.7593	0.7701
IUTIS-5	0.8105	0.7972	1.0863	0.7820	0.7872
PAWCS	0.7841	0.7724	1.1196	0.7477	0.7556
SuBSENSE	0.7522	0.8144	1.5869	0.7453	0.7540

Not olarak; CDnet2014 veri kümesindeki ground-truth'ların tüm kareleri dikkate alınmaktadır

Şekil 2.20 - 2.22'de, 7 farklı yöntemle tahmin edilen ön plan maskelerini gösteren bazı örnek sonuçlar sunulmaktadır. Yer sınırlamaları nedeniyle her bir kategoriden rastgele bir video sahnesi seçilmektedir. Bu şekillerden görülebileceği gibi modelimiz, ön plandaki nesneler çok küçük ve belirsiz olsa bile daha doğru nesne sınırlarını tahmin edebilmektedir. Dinamik arka plan ve gölgeleri de tamamen ortadan kaldırabilmektedir. Bundan başka, Modelimiz aynı zamanda çeşitli kamera hareketlerini içeren *cameraJitter* ve *PTZ* kategorilerinde görülebileceği gibi büyük kamera hareketlerine karşı da dayanıklıdır; bu durumda her iki kategoride de F-Measure 0.95'den fazla elde edilmektedir.

Çizelge 2.7 Yöntemiz ve CDnet2014 veri setinde mevcut teknoloji harikası yöntemleri arasında bir karşılaştırma

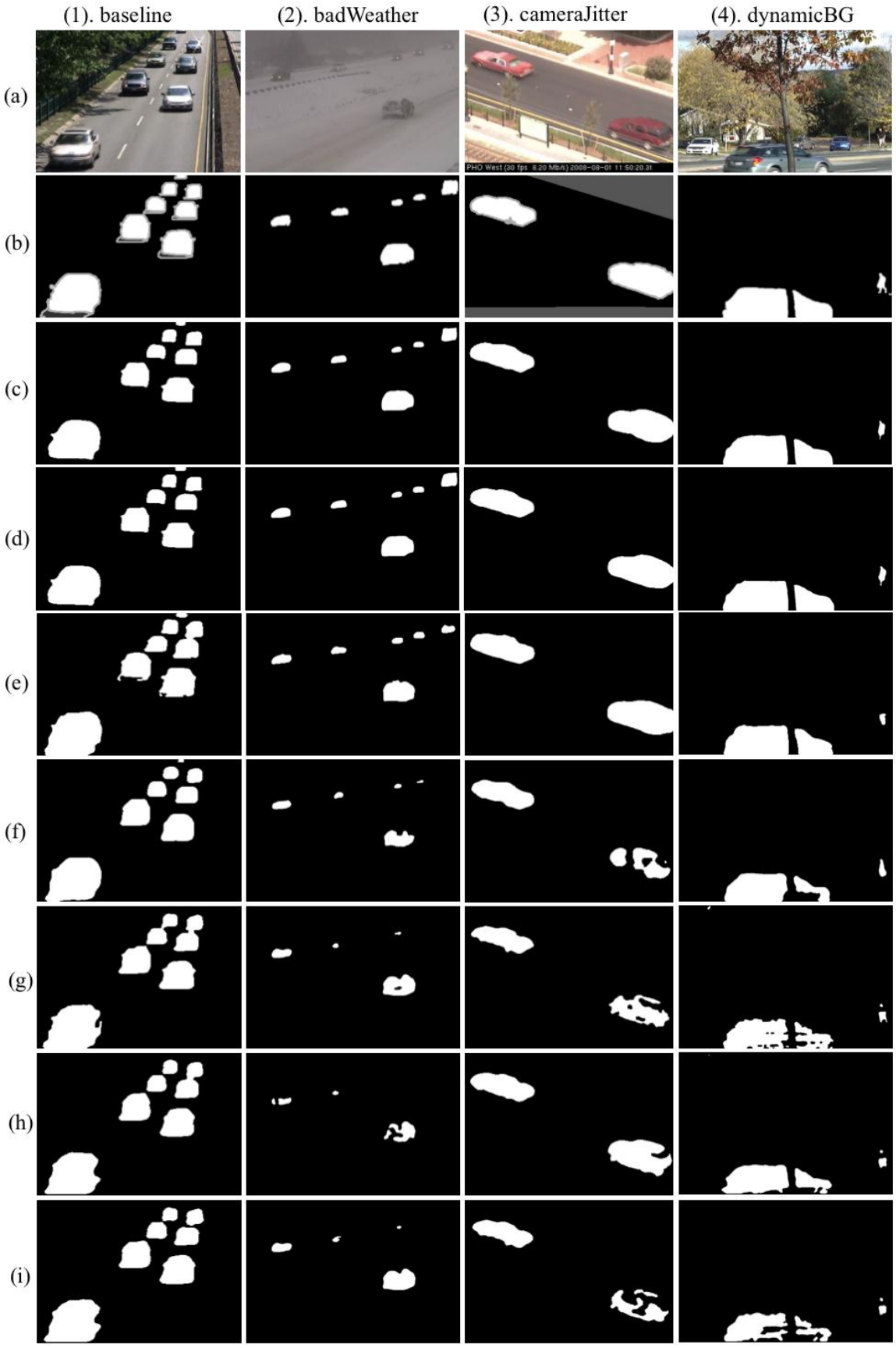
Methods	Overall			
	avg. Precision	avg. Recall	avg. PWC	avg. F-Measure
FgSegNet_S	**0.9751**	**0.9896**	**0.0461**	**0.9804**
FgSegNet_M	**0.9758**	**0.9836**	**0.0559**	**0.9770**
Cascade	0.8997	0.9506	0.4052	0.9209
DeepBS	0.8332	0.7545	1.9920	0.7458
IUTIS-5	0.8087	0.7849	1.1986	0.7717
PAWCS	0.7857	0.7718	1.1992	0.7403
SuBSENSE	0.7509	0.8124	1.6780	0.7408

Not olarak; bu sonuçlar Change Detection 2014 yarışma web sitesinden elde edilmektedir

Çizelge 2.8 Yöntemiz ve CDnet2014 veri setinde mevcut en iyi yöntem arasında bir karşılaştırma

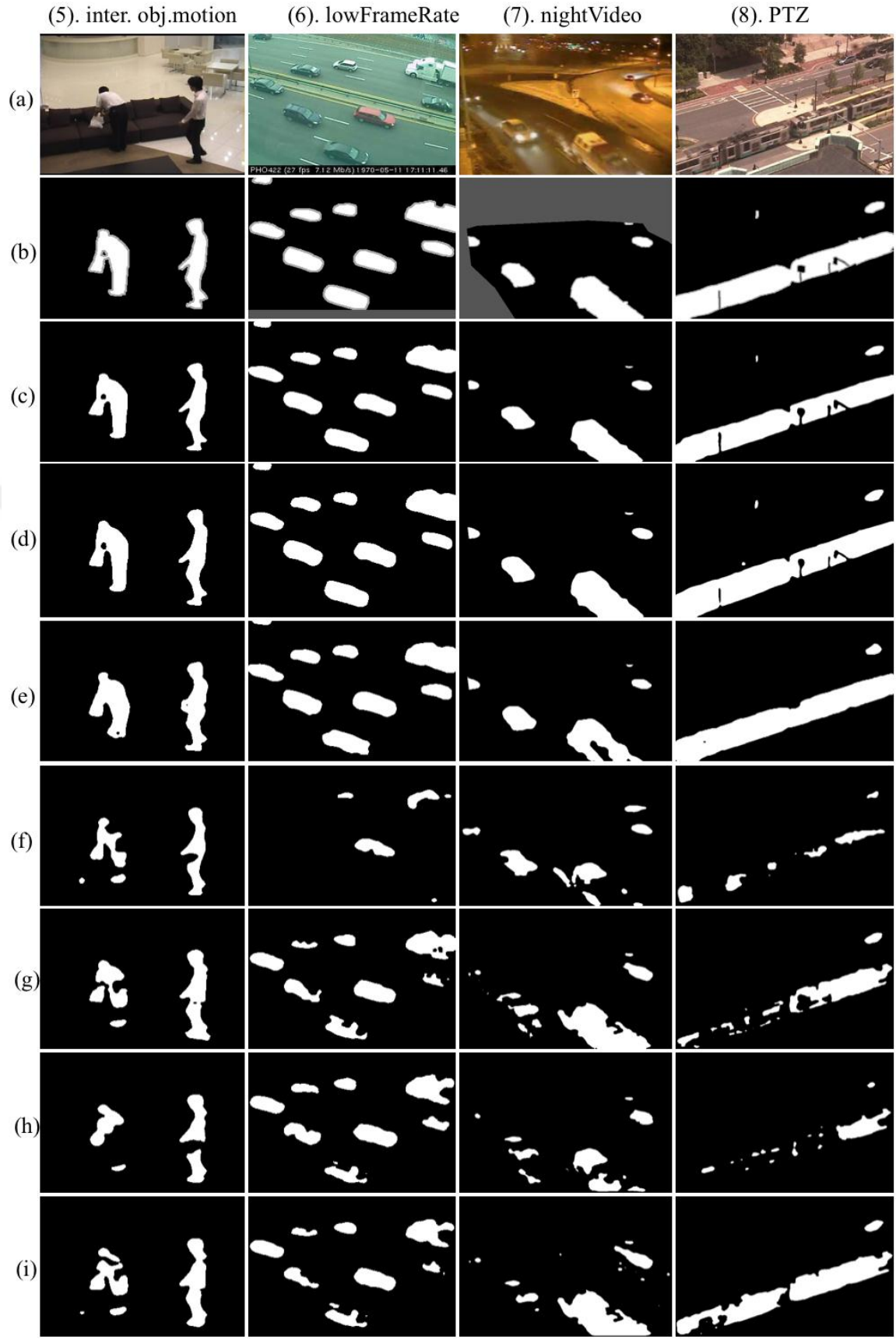
Methods	**F-Measure**												**MCC**	**Seg./Train. Speed**
	baseline	cam.Jitter	badWea.	dyna.bg	int.obj	lowF.rate	nightVid.	PTZ	shadow	thermal	turbul.	**Overall**	**Overall**	
FgSegNet_S	**0.9975**	**0.9921**	**0.9850**	**0.9892**	**0.9932**	**0.8801**	**0.9759**	**0.9760**	**0.9960**	**0.9929**	**0.9743**	**0.9775**	**0.9776**	**∼21fps/-**
FgSegNet_M	**0.9968**	**0.9914**	**0.9757**	**0.9865**	**0.9922**	**0.8816**	**0.9696**	**0.9567**	**0.9944**	**0.9907**	**0.9722**	**0.9734**	**0.9734**	**∼18fps/23.7min**
Cascade	0.9779	0.9687	0.9421	0.6515	0.8225	0.7373	0.8882	0.7052	0.9548	0.8785	0.9190	**0.8587**	**0.8600**	∼13fps/35min

Not olarak; bu puanlar eğitim kareleri hariç tutularak değerlendirilmektedir



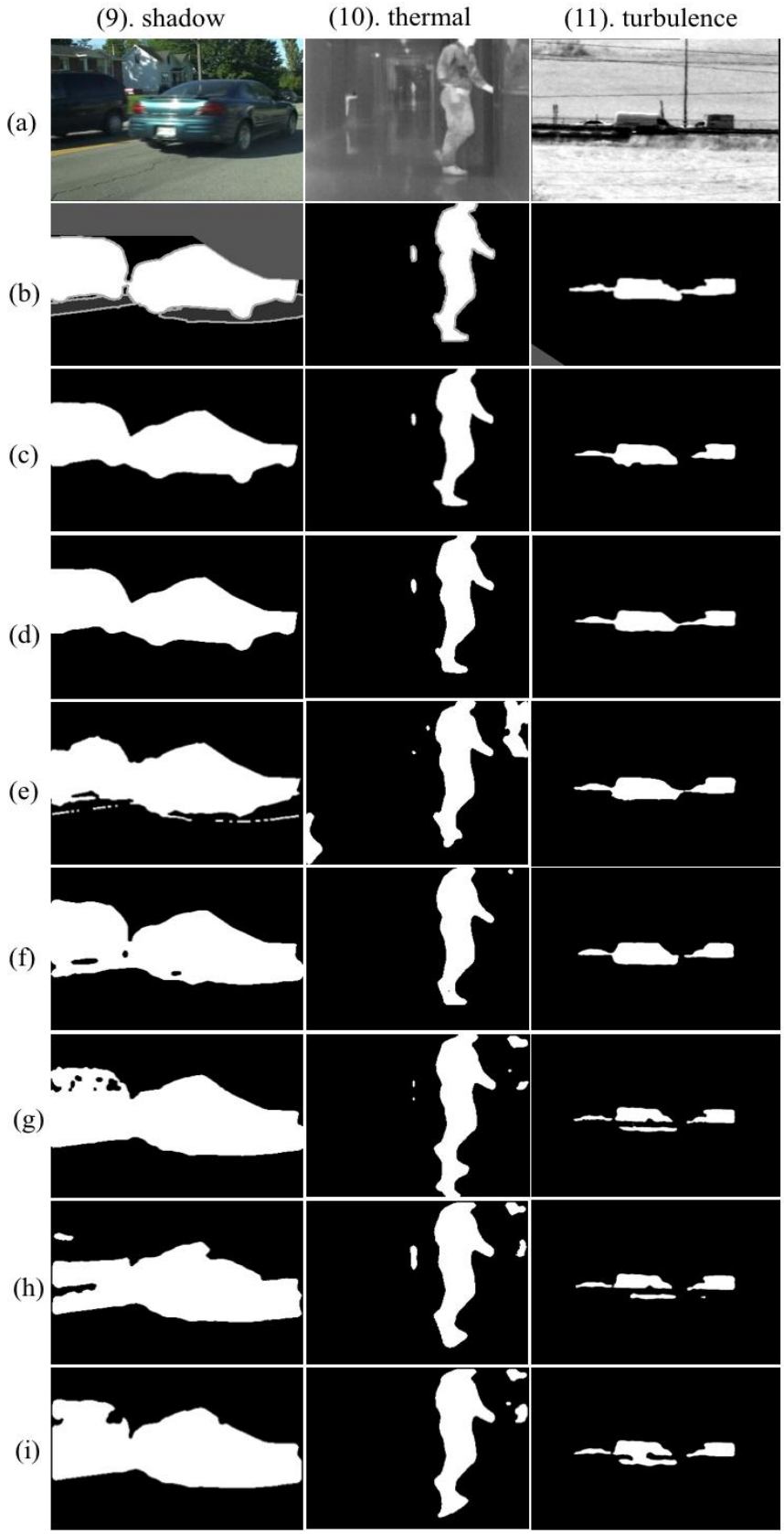
Şekil 2.20 Her kategorideki seçilen bir sahneden elde edilen sonuçlar

a. ham görüntüler; b. ground-truth; c. FgSegNet_M sonuçları; d. FgSegNet_S sonuçları; e., f., g., h. ve i. sırasıyla Cascade, DeepBS, IUTIS-5, PAWCS ve SuBSENSE'den elde edilen sonuçları göstermektedir



Şekil 2.21 Her kategorideki seçilen bir sahneden elde edilen sonuçlar

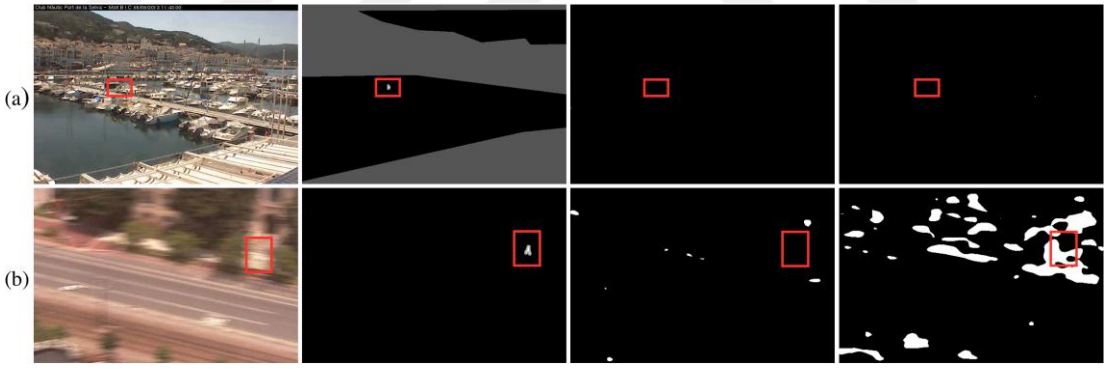
a. ham görüntüler; b. ground-truth; c. FgSegNet_M sonuçları; d. FgSegNet_S sonuçları; e., f., g., h. ve i. sırasıyla Cascade, DeepBS, IUTIS-5, PAWCS ve SuBSENSE'den elde edilen sonuçları göstermektedir



Şekil 2.22 Her kategorideki seçilen bir sahneden elde edilen sonuçlar

a. ham görüntüler; b. ground-truth; c. FgSegNet_M sonuçları; d. FgSegNet_S sonuçları; e., f., g., h. ve i. sırasıyla Cascade, DeepBS, IUTIS-5, PAWCS ve SuBSENSE'den elde edilen sonuçları göstermektedir

F-Measure'yi diğer kategorilerle karşılaştırdığımızda metodumuz *lowFrameRate* kategorisi için yetersiz performans göstermektedir. Bu düşük performans, temel olarak düşük kare oranında görüntülenen son derece küçük ön plan nesnelerinin bulunduğu bir sahnenin zorlu içeriğinden kaynaklanebilmektedir. Bir insan gözlemcinin nesnelerin mekansal konumlarını tespit etmesi bile zordur. Şekil 2.23.a'da bir örnek verilmektedir. İkinci düşük performans *PTZ* kategorisinde gözlenmektedir (Şekil 2.23.b). Burada kamera sürekli olarak sahnenin etrafında kaydırmak, eğmek ve yakınlaştırmaktadır. Bu hareket tarafından bulanık sahneler oluşturulmaktadır. Sahneden açıkça görülebileceği gibi kamera hareket etmeye başladığında bir ön plan nesnesi (bu durumda kaldırım boyunca yürüyen beyaz bir gömlek giymiş bir insan) arka planla tamamen karıştırılmaktadır. Bir insan gözlemcisi için bile bu bölgenin ön plan nesnesini içerip içermediğini ayırt etmek zordur. Bu kategoride modelimiz çok yanlış pozitifler (FP) üretmektedir. FgSegNet_M'nin FgSegNet_S ile karşılaştırıldığında kamera hareketlerine karşı daha gürbüz olduğunu gözlemlenmiştir. Fakat, PTZ kategorisinin ortalama puanı diğer kategorilere kıyasla hala kabul edilebilmektedir.



Şekil 2.23 Modelimizin yetersiz performans gösterdiği örnek sahneleri
a. LowFrameRate kategorisi; b. PTZ kategorisi. Her sütun sırasıyla ham görüntüleri, ground-truth, FgSegNet_M ve FgSegNet_S sonuçlarını göstermektedir

2.6.1.5.2 SBI2015 veri setinde

Wang vd. (2017) tarafından sağlanan groundtruth etiketli 14 video sekansı içeren Scene Background Initialization 2015 (SBI2015) veri kümesi (Maddalena ve Petrosino 2015) üzerinde daha fazla deney yapılmaktadır. Wang vd. (2017) ile aynı eğitim protokolünü

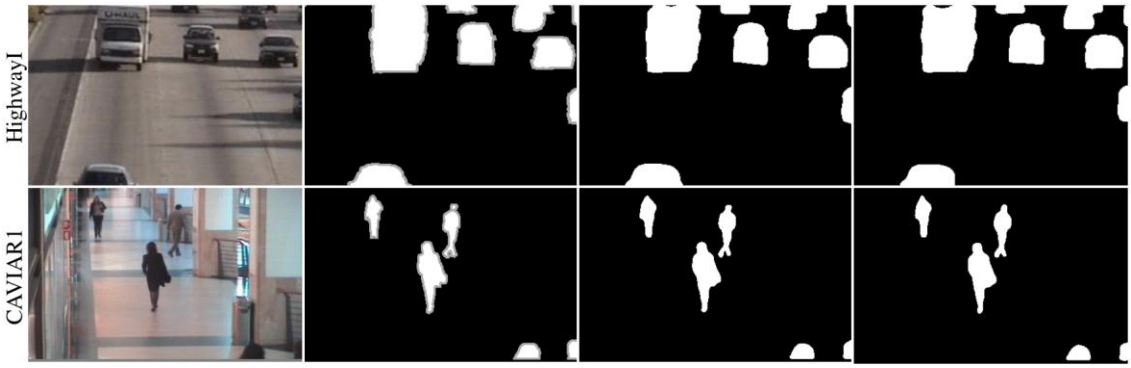
takip ederek % 20'si eğitim için (n kare, n ∈ [2−148]) ve % 80'i test için bölünmüştür.

Test sonuçlarımız çizelge 2.9'da ve şekil 2.24'te verilmektedir. Görülebileceği gibi yöntemlerimiz sert gölgeleri tamamen ortadan kaldırabilmekte (Şekil 2.24, ilk satır) ve sadece 6 kare (*eğitim+validasyon* için yalnızca n = 2 kare kullanılmıştır) içeren *Toscana* video dizisi dışında tüm video dizilerinde yüksek F-Measure elde edilebilmektedir. Yine de FPM modülü ile FgSegNet_S, FgSegNet_M'den % 0.37 puan daha iyi performans göstermektedir. FgSegNet_S ve FgSegNet_M metotlarımız, Cascade (Wang vd. 2017)'i sırasıyla % 8.99 ve % 8.62 puan geliştirmektedir. Sonuçlar, yöntemlerimizin sağlam olduğunu ve nesnelere son derece az sayıda eğitim çerçevesinden doğru şekilde bölütlenmeyi öğrenebildiğini göstermektedir.

Çizelge 2.9 SBI2015 test setindeki sonuçlar

Video	**FgSegNeg_S**		**FgSegNeg_M**	
	F-Measure	PWC	F-Measure	PWC
Board	0.9977	0.1364	0.9978	0.1291
Candela_m1.10	0.9936	0.0507	0.9954	0.0364
CAVIAR1	0.9987	0.0097	0.9989	0.0079
CAVIAR2	0.9826	0.0136	0.9834	0.0130
CaVignal	0.9864	0.3158	0.9881	0.2768
Foliage	0.9726	3.8221	0.9724	3.8377
HallAndMonitor	0.9918	0.0394	0.9923	0.0371
HighwayI	0.9926	0.1457	0.9928	0.1414
HighwayII	0.9950	0.0299	0.9947	0.0314
HumanBody2	0.9919	0.1653	0.9918	0.1663
IBMtest2	0.9850	0.1372	0.9845	0.1416
PeopleAndFoliage	0.9910	0.9382	0.9912	0.9153
Snellen	0.9790	2.3244	0.9781	2.4090
Toscana	0.9060	3.8057	0.8496	5.0605
Overall	**0.9831**	**0.8524**	**0.9794**	**0.9431**
Cascade	**0.8932**	**5.5800**	-	-

Her satır, her video dizisinin sonuçlarını göstermektedir. Son satırda bir önceki yöntemle bir karşılaştırma sunulmaktadır



Şekil 2.24 SBI2015 veri setinde test sonuçları
Her sütun sırasıyla ham görüntüleri, ground-truth'lar, FgSegNet_M ve FgSegNet_S sonuçlarını göstermektedir

2.6.1.5.3 UCSD Background Subtraction setinde

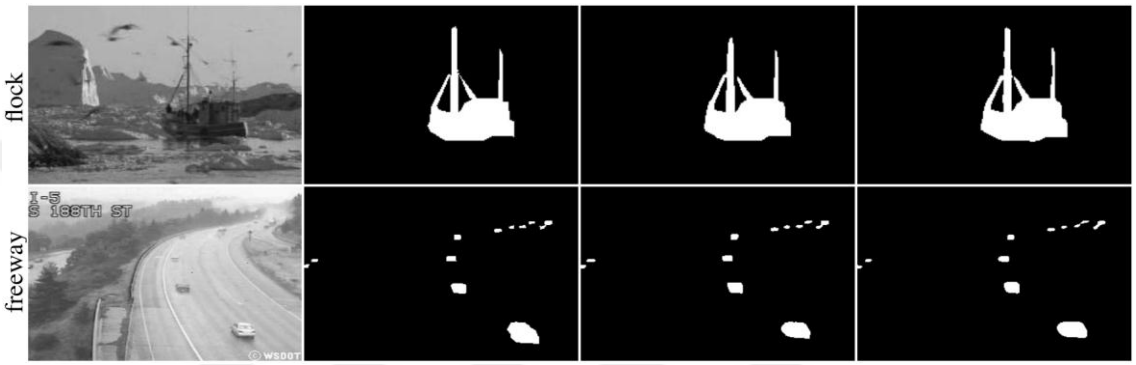
Bir başka deney ise ground-truth etiketlerine sahip 18 video dizisi içeren UCSD Background Subtraction veri kümesinde yapılmaktadır. Bu veri kümesi, arka plan çıkarma alanında son derece zorlayıcı dinamik arka planlar içermektedir. İki deney seti gerçekleştirilmektedir: ilkinde, çerçevelerin % 20'si eğitim için (n çerçeve, $n \in [3-23]$) ve % 80'ı test için bölünmüş, ikincisinde çerçevelerin % 50'si eğitim için (n çerçeve, $n \in [7-56]$) ve % 50'si test için ayrılmıştır. UCSD veri kümesinin eğitim karelerinin sayısı, CDnet2014 ve SBI2015 veri kümelerine kıyasla küçüktür.

Alan sınırlamaları nedeniyle, sadece çizelge 2.10'da ve şekil 2.25'te genel sonuçlar sağlanmaktadır. Görülebileceği gibi FgSegNet_M modeli, bu veri kümesinde bazı noktalara göre tüm bölünmelerde ve eşiklerde FgSegNet_S'den daha iyi performans göstermektedir. % 20'lik bölünme için *eğitim+validasyon* kümesi küçük olduğu için (min 3 kare, max 23 kare) 0.90'ın altında F-Measure elde edilmektedir. Halbuki, *eğitim+validasyon* için % 50 oranında bölüştüğümüzde (min 7 kare, max 56 kare) F-Measure 0.90'ı geçmektedir. Bu sonuçlar son derece dinamik arka planları ağın öğrenmesi için 2 veya 3 çerçevenin yeterli olmadığını göstermektedir. Halbuki, önerilen ağ oldukça dinamik arka plan ile nispeten küçük nesneleri yine de tahmin edebilmekte ve $n \in [3-23]$ eğitim karelerine rağmen % 0.89'un üzerinde F-Measure üretmektedir.

Çizelge 2.10 UCSD testinden elde edilen toplam sonuçlar

Train. split/threshold	FgSegNet_S		FgSegNet_M	
	F-Measure	PWC	F-Measure	PWC
20%_th0.4	0.8822	0.7052	0.8948	0.6260
20%_th0.7	0.8905	0.6273	0.8912	0.6381
50%_th0.4	0.9139	0.5024	0.9203	0.4637
50%_th0.7	0.9149	0.4676	0.9151	0.4878

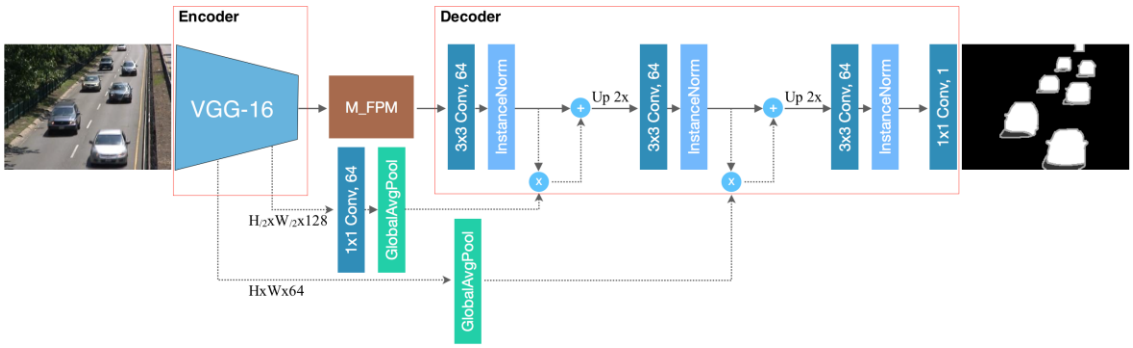
Farklı eşikler ile 18 video dizisinde belirlenmektedir



Şekil 2.25 UCSD Background Subtraction veri setinde test sonuçları
Her sütun sırasıyla ham görüntüleri, ground-truth'lar, FgSegNet_M ve FgSegNet_S sonuçlarını göstermektedir

2.6.2 FgSegNet_v2 ağ mimarisi

Bu bölümde önceki çalışmamız olan FgSegNet'i hem enkoder hem de FPM modülünde tekrar gözden geçirmekteyiz.



Şekil 2.26 FgSegNet_v2 mimarisi

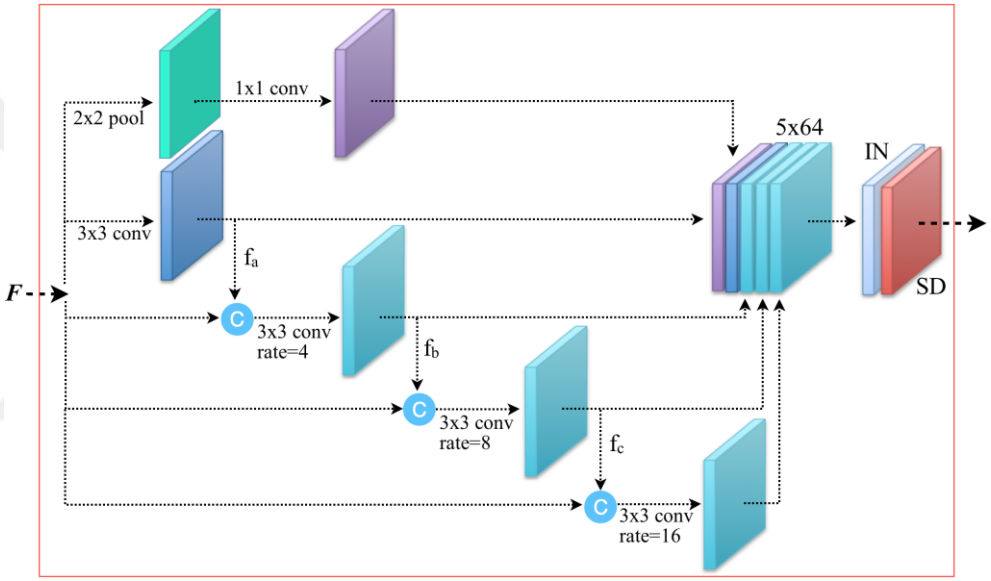
2.6.2.1 Enkoder ağı

Önceden eğitilmiş VGG-16 ağının düşük seviyeli özniteliklerin motive ettiği FgSegNet'te, son beş blok ve üçüncü maksimum pooling katmanı kaldırarak VGG-16 ağının ilk dört bloğu kullanılmıştır. Sonuç olarak daha yüksek çözünürlüklü özellik haritaları elde etmiştir. Dropout (Srivastava vd. 2014) katmanları, değiştirilen ağın her bir konvolüsyon katmanından sonra yerleştirilmiş ve daha sonra bu bloğa ince ayar yapılmıştır. Bu çalışmada FgSegNet uygulamasında olduğu gibi aynı enkoder mimarisi de kullanılmaktadır. Bu değiştirilen ağın, ResNet (He vd. 2016) gibi önceden eğitilmiş diğer ağlarla karşılaştırıldığında performansı artırdığı gözlemlenmiştir.

2.6.2.2 Değiştirilen Feature Pooling Modülü

Enkoderin çıktısından elde edilen F öznitelik haritalarından orijinal FPM modülü birden fazla ölçekli öznitelikler ayıklamaktadır, bu işlemi farklı genişleme oranlarıyla birkaç konvolüsyon katmanları, aynı öznitelik haritasıları F'e 1x1 konvolüsyonu uygulaması ve ardından maksimum pooling katmanıdan geçirilerek ayıklanan öznitelikler derinlik boyutu boyunca birleştirilerek gerçekleştirmektedir. Son olarak birleştirilen öznitelikler BatchNormalization ve SpatialDropout katmanlarından geçirilmektedir. Bu çalışmada orijinal FPM modülünün geliştirilmesi için bazı değişiklik önerileri iki kısımdan oluşmaktadır (Şekil 2.27): (1) Normal 3x3-conv'dan sonuçlanan f_a öznitelikleri, F özniteliği ile birleştirilmekte ve bu birleştirilen öznitelikten 4'lük genişleme oranı ile 3x3-conv tarafından kademeli olarak başka f_b öznitelikleri ayıklanmaktadır. Sonra, F ve f_b birleştirilip 8'lik genişleme oranıyla 3x3-conv'a beslenmekte ve f_c öznitelikleri elde edilmektedir. Tekrar, F ve f_c birleştirilip 16'lık genişleme oranıyla 3x3-conv'u beslenmektedir. Son olarak, ayıklanan tüm özellikler, F' olarak adlandırılan 5x64 derinlik özniteliklerini oluşturmak için birleştirilmektedir; yani F', FgSegNet_S'ten daha geniş alıcı alanları olan çok ölçekli öznitelikler içermektedir. (2) InstanceNormalization (Ulyanov vd. 2016)'ın küçük toplu boyutta biraz daha iyi performans verdiğini deneysel olarak gözlemlediğimizden, BatchNormalization, InstanceNormalization ile değiştirilmektedir. Birçok pooling katmanı aynı öznitelikler F üzerinde çalıştırıldığı için birleştirilen F' özniteliklerinin ilişkili olması muhtemeldir.

Bağımsız öznitelik haritalarını yükseltmek için, öznitelik haritalarındaki komşu piksellerin güçlü bir şekilde ilişkilendirilmesi durumunda, 2D öznitelik haritalarının tümünü belirli oranlarda (örn. 0.25) kaldırmak için SpatialDropout kullanılmaktadır. Bir öznitelik düzlemindeki bireysel elemanların (nöronların) bırakıldığı normal Dropout'un aksine SpatialDropout'un performansı iyileştirmeye yardımcı olduğunu ve ağımızdaki overfitting'i önlediğini gözlemlemekteyiz. Not olarak M-FPM modülünde InstanceNormalization'dan hemen sonra bir kez ReLU uygulanmaktadır. Bundan sonra değiştirilen FPM'yi M-FPM olarak adlandırılmaktadır.



Şekil 2.27 Değiştirilen FPM modülü
IN (InstanceNormalization), SD (SpatialDropout)

2.6.2.3 Dekoder ağı ile GAP modülü

İki GAP'lı dekoder ağımız şekil 2.26'da gösterilmektedir. Dekoder kısmında, üç 3x3-conv katmanlarının yığını ve 1x1-conv katmanını içermektedir. Burada InstanceNormalization'u takip edilen 3x3-conv katmanları ve 1x1-conv katmanı, öznitelik uzayından görüntü uzayına projeksiyondur. Bir öznitelik dilimi içeren 1x1-conv katmanı dışında tüm 3x3-conv katmanları 64'lük öznitelik haritalarına sahiptir. Not olarak dekoder kısmında InstanceNorm'dan sonra lineer olmayan ReLU fonksiyonu uygulanmakta ve 1x1-conv'dan sonra sigmoid aktivasyon fonksiyonu uygulanmaktadır.

Global Average Pooling (GAP): Enkoderin düşük seviyeli özniteliklerinden ve dekoderin yüksek seviyeli özniteliklerinden gelen bilgileri birleştiren iki katsayı vektörü vardır: (1) Birinci katsayı vektörü, maksimum pooling katmanından hemen önce ikinci konvolüsyon katmanından ayıklanmaktadır. (2) İkinci katsayılar vektörü dördüncü katmandan ayıklanmaktadır. Dekoder kısımda 3x3-conv katmanları 64 özniteliğe sahip olduğu için enkoderin dördüncü konvolüsyon katmanı ilk olarak 128 öznitelikten 64 özniteliğine yansıtılmaktadır.

Her iki katsayı vektörü (α_i), dekoder kısımdaki birinci ve ikinci konvolüsyon katmanlarının çıktı öznitelikleri (f^i_j) ile çarpılmaktadır (Şekil 2.26). Ölçeklenen öznitelikler, f'^i_j özniteliklerini oluşturmak için orijinal özniteliklere eklenmektedir, burada $f'^i_j:\alpha_i*f^i_j+f^i_j$, i□[0, 63] her bir öznitelik derinliğinin indeksi ve j ise her bir öznitelik dilimindeki bir elemanın indeksidir. Son olarak, birleştirilen f'^i_j, çift-doğrusal enterpolasyon kullanılarak iki katına büyütülüp bir sonraki katmanları beslenmektedir. GAP modülüne sahip ağın çok az hesaplama maliyeti eklediğini, buna ek olarak genel olarak performansı artırdığını gözlemlemekteyiz.

2.6.2.4 Eğitim detayları

Bu yöntem için FgSegNet_S ile aynı eğitim prosedürü takip edilmiş, ancak minimum validasyon hatası 5 epoch içinde gelişme göstermeyip durdurduğunda öğrenme oranı 10 kat azaltılmaktadır. Maksimum 100 epoch belirlenmiş ve validasyon hatası 10 epoch içinde iyileşmeyi durdurduğunda eğitim erken durdurulmaktadır. Aynı eğitim seti (yani 200 kare) FgSegNet_S (veya FgSegNet_M)'de olduğu gibi kullanılmakta ve başka bir deneyi gerçekleştirmek için bu 200 kareden rasgele 25 kare seçilerek az sayıda eğitim karesi kullanmaya çalışılmaktadır.

2.6.2.5 Eşikleme (Thresholding)

FgSegNet_M ve FgSegNet_S'de olduğu gibi eşikleme kullanılmaktadır. Bu şekilde 200-kare deneyi için yüksek 0,9'luk bir eşik ve 25-kare deneyi için 0,7'lik bir eşik seçilmektedir.

2.6.2.6 Sonuçlar ve tartışmalar

2.6.2.6.1 CDnet2014 veri setinde

Global Average Pooling için Deneyler: Önerilen ağda GAP katmanlarının etkinliğini değerlendirmek için CDNet2014 veri kümesinden 30 video dizisi içeren en zorlu 6 kategoriyi (örn. *cameraJitter, badWeather, dynamicBackground, intermittent-ObjectMotion, shadow, turbulence*) seçilerek iki faklı deney gerçekleştirilmiştir. Seçilen video sekansları 1150 ila 7999 arasında değişen bir dizi çerçeve içermektedir. Önceki bölümde belirtildiği gibi, *eğitim+validasyon* için sadece 25 kare kullanılmış ve *test* için kalan kareler seçilmiştir.

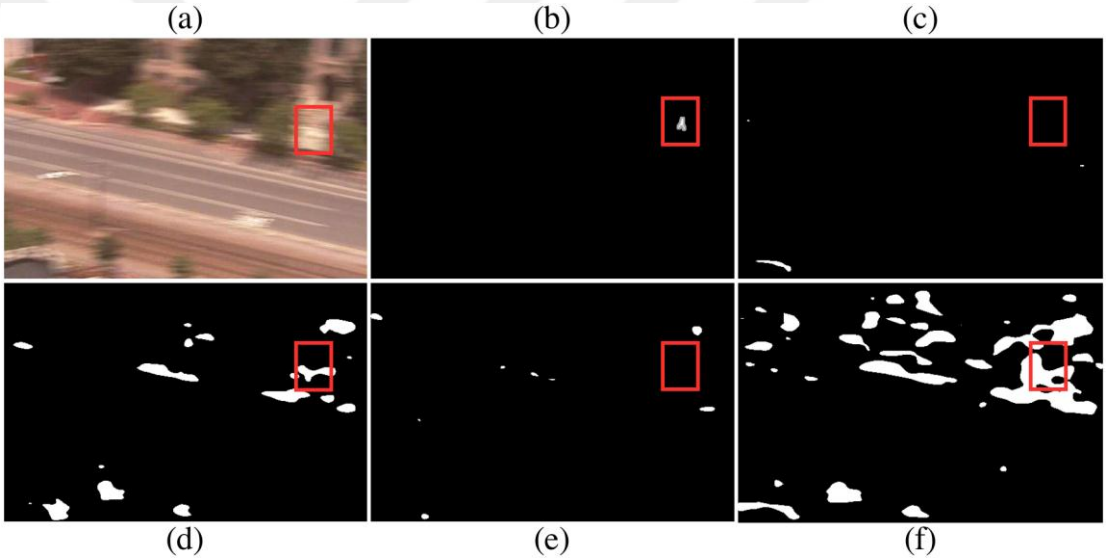
İlk deneyde, GAP katmanları ağdan tamamen kaldırılmakta ve bu değiştirilen konfigürasyon *no_GAP* olarak adlandırılmaktadır. İkinci deneyde, GAP katmanları tutulmaktadır. Çizelge 2.11'de görülebileceği gibi, GAP'lı ağ bazı kategorilerde ve bazı puanlarda *no_GAP*'a göre öndedir. Özellikle GAP, cameraJitter kategorisinde no_GAPC'ın % 2.34 puan geçmiştir.

Çizelge 2.11 GAP ve no_GAP ile sonuçlar

Category	no_GAPC		GAPC	
	F-Measure	PWC	F-Measure	PWC
cameraJit	0.9506	0.3026	0.9740	0.2271
badWeather	0.9781	0.0657	0.9783	0.0639
dynamicBg	0.9636	0.0311	0.9665	0.0325
intermitt	0.9597	0.2997	0.9735	0.3268
shadow	0.9840	0.1265	0.9853	0.1159
turbulence	0.9600	0.0439	0.9587	0.0438

Modified FPM (M-FPM) Deneyler: Bu çalışmada M-FPM modülünün orijinal FPM'ye kıyasla etkinliği gösterilmektedir. Yine iki farklı deney gerçekleştirilmektedir: İlk deneyde önerilen decoder ile M-FPM modülü kullanılmaktadır. İkinci deneyde ise önerilen decoder ile orijinal FPM modülü kullanılmaktadır. Deney sonuçları şekil 2.28'de gösterilmektedir. Görülebileceği gibi: **(1)** önerilen M-FPM modülü (Şekil 2.28.c), orijinal FPM modülüne (Şekil 2.28.d) kıyasla daha az yanlış pozitif üretmiştir,

(2) Önerilen decoder (Şekil 2.28.d), FgSegNet_S dekoderine (Şekil 2.28.f) kıyasla daha etkilidir, **(3)** FgSegNet_M, çoklu giriş ağ öznitelikleri birleştirildiği ve ortaklaşa çalıştığı için kamera hareketi için sağlamdır (Şekil 2.28.e). Bir karşılaştırma olarak M-FPM modülünün çok-ölçekli öznitelik füzyonunu bu modülün içine tanıtarak hesaplama açısından daha pahalı olan çok-girişli ağ ihtiyacını azaltabileceği deneysel olarak gözlemlenmektedir. Şekil 2.28'den görülebileceği gibi, önerilen yöntem (Şekil 2.28, (c)), FgSegNet ailesi (Şekil 2.28.e,f) ile karşılaştırıldığında çok daha az yanlış pozitif sonuç vermekte ve PTZ kategorisinde FgSegNet_M, FgSegNet_S ve Cascade'yi sırasıyla % 0.43, % 0.56 ve % 5.92 puanlarla geçmiştir (Çizelge 2.12'de bakınız). Şekil 2.29'da da önerilen M-FPM'nin etkinliği gösterilmektedir.



Şekil 2.28 Orijinal FPM ile karşılaştırıldığında geliştirilen M_FPM modülü
a. ham görüntü; b. ground-truth; c. *M_FPM+önerilen dekoder* sonucu; d. *FPM+önerilen dekoder* sonucu; e. FgSegNet_M sonucu; f. FgSegNet_S sonucu

Şekil 2.29 Önerilen yöntem ve kamera hareket kategorisinde (cameraJitter) mevcut son teknoloji yöntemler arasında bir karşılaştırma

Eklemelerin etkinliğinin CDnet2014 veri kümesinin zorlu alt kümesiyle değerlendirdikten sonra önerilen mimari konfigürasyonu kullanarak daha fazla deney gerçekleştirilmektedir. Yoğun ground-truth'ları etiketlemek daha fazla insan çabası gerektirmektedir ve etiketleme yükünü azaltmak için önceki yöntemlerimizde ve Wang vd. (2016) sadece birkaç eğitim örneği kullanmıştır. Aynı fikir 200-kare deney için de uyarlamaktayız; halbuki, bu çalışmada eğitim örneklerini 8x daha azaltılıp 25 kareye indirilmektedir. Özellikle, 25 kare ve 200 kare kullanarak iki set deney yapılmakta ve çizelge 5.8'de test sonuçları gösterilmektedir. Görülebileceği gibi 25-kare deneyler için 11 kategoride genel F-Measure 0.9473 elde edilmektedir. Çerçevelerin sayısını 200'e artırmak 25-frame deney sonuçlarına kıyasla F-Measure % 3.16 puanını artırmaktadır. Benzer şekilde 25 ila 200 arasındaki karelerin sayısını artırdığımızda PWC % 0.1149 oranında azalmaktadır. Not olarak eğitim çerçeveleri bu değerlendirmelere dahil edilmeyip sadece test çerçeveleri kullanılmaktadır. deneyimizde en iyi performansları sağladığı için 25-kare deneyler için 0.7 eşiği ve 200-kare deneyler için 0.9 eşiği kullanılmaktadır.

Çizelge 2.13'te önerilen yöntem ile son teknoloji yöntemler arasındaki sonuçları karşılaştırmaktayız. Not olarak çerçeve sayısı açısından bir karşılaştırma yapmak için CDnet2014 veri kümesindeki tüm sağlanan ground-truth'ları kullanarak metrikler hesaplanmaktadır. Görülebileceği gibi, yöntemimiz (FgSegNet_v2) bazı puanlarda mevcut son teknoloji yöntemleri geçmektedir. Özellikle, önerilen yöntem, kamera hareket kategorilerinde önemli ölçüde geliştirmektedir (PTZ ve cameraJitter kategorisinde sahnelerin etrafında kamera titremesi, kamera çevrinmesi, kamera eğmesi veya kamera zoom yapmasının hareketi vardır). FgSegNet_v2, FgSegNet_M'deki çoklu giriş öznitelik füzyonunun kullanımını kaldırabilmektedir.

Wang vd. (2016) tarafından sağlanan eğitim çerçevelerini kullanarak başka bir deney daha yapılmakta ve modelimiz Change Detection 2014 (changedetection.net) yarışmasında değerlendirilmektedir. Karşılaştırma çizelge 2.14'te verilmektedir. Görülebileceği gibi yöntemimiz, bazı puanlarla mevcut son teknoloji yöntemlerden daha iyi performans göstermiştir. Özellikle, derin öğrenme yöntemi için önerilen yöntem, FgSegNet_S, FgSegNet_M, Cascade ve DeepBS yöntemlerini sırasıyla % 0.43,

% 0.77, % 6.38 ve % 23.89 puanlarla geçmektedir. Önerilen yöntem de tüm geleneksel yöntemleri % 21.3 puan ile geçmektedir. Yöntemimiz, başvuru sırasında 1 numara olarak sıralanmıştır.

Çizelge 2.12 Test sonuçları 11 kategoride CDnet2014 veri kümesinden 25 ve 200 kare manuel ve rasgele seçerek elde edilmektedir

Category	FPR		FNR		Recall		Precision		PWC		**F-Measure**	
	25f	200f	25f	200f	25f	200f	25f	200f	25f	200f	**25f**	**200f**
baseline	0.0002	0.00004	0.0100	0.0038	0.9900	0.9962	0.9942	0.9985	0.0480	0.0117	**0.9921**	**0.9974**
cameraJit	0.0004	0.00012	0.0419	0.0093	0.9581	0.9907	0.9907	0.9965	0.2271	0.0438	**0.9740**	**0.9936**
badWeather	0.0003	0.00009	0.0257	0.0215	0.9743	0.9785	0.9825	0.9911	0.0639	0.0295	**0.9783**	**0.9848**
dynamicBg	0.0001	0.00002	0.0315	0.0075	0.9685	0.9925	0.9655	0.9840	0.0325	0.0054	**0.9665**	**0.9881**
intermitt	0.0017	0.00015	0.0243	0.0104	0.9757	0.9896	0.9720	0.9976	0.3268	0.0707	**0.9735**	**0.9935**
lowFrameR.	0.0003	0.00008	0.2496	0.0956	0.7504	0.9044	0.7860	0.8782	0.1581	0.0299	**0.7670**	**0.8897**
nightVid.	0.0008	0.00022	0.1197	0.0363	0.8803	0.9637	0.9540	0.9861	0.3048	0.0802	**0.9148**	**0.9747**
PTZ	0.0002	0.00004	0.0870	0.0215	0.9130	0.9785	0.9776	0.9834	0.0892	0.0128	**0.9423**	**0.9809**
shadow	0.0003	0.0001	0.0203	0.0056	0.9797	0.9944	0.9911	0.9974	0.1159	0.0290	**0.9853**	**0.9959**
thermal	0.0009	0.00024	0.0456	0.0089	0.9544	0.9911	0.9815	0.9947	0.2471	0.0575	**0.9677**	**0.9929**
turbulence	0.0002	0.00011	0.0369	0.0221	0.9631	0.9779	0.9546	0.9747	0.0438	0.0232	**0.9587**	**0.9762**
Overall	**0.0005**	**0.0001**	**0.0630**	**0.0220**	**0.9370**	**0.9780**	**0.9591**	**0.9802**	**0.1507**	**0.0358**	**0.9473**	**0.9789**

Her sıra, her bir kategorinin ortalama sonuçlarını göstermektedir

Çizelge 2.13 11 kategoride 8 yöntem arasında bir karşılaştırma

Methods	F-Measure											Overall
	baseline	cameraJit	badWeater	dynamicBg	intermitt	lowFrameRate	nightVid.	PTZ	shadow	thermal	turbul.	
FgSegNet_v2	**0.9980**	**0.9961**	0.9900	0.9950	0.9939	**0.9579**	0.9816	**0.9936**	0.9966	0.9942	**0.9815**	**0.9890**
FgSegNet_S	**0.9980**	0.9951	**0.9902**	**0.9952**	**0.9942**	0.9511	**0.9837**	0.9880	**0.9967**	**0.9945**	0.9796	**0.9878**
FgSegNet_M	0.9975	0.9945	0.9838	0.9939	0.9933	0.9558	0.9779	0.9893	0.9954	0.9923	0.9776	**0.9865**
Cascade	0.9786	0.9758	0.9451	0.9658	0.8505	0.8804	0.8926	0.9344	0.9593	0.8958	0.9215	**0.9272**
DeepBS	0.9580	0.8990	0.8647	0.8761	0.6097	0.5900	0.6359	0.3306	0.9304	0.7583	0.8993	**0.7593**
IUTIS-5	0.9567	0.8332	0.8289	0.8902	0.7296	0.7911	0.5132	0.4703	0.9084	0.8303	0.8507	**0.7820**
PAWCS	0.9397	0.8137	0.8059	0.8938	0.7764	0.6433	0.4171	0.4450	0.8934	0.8324	0.7667	**0.7477**
SuBSENSE	0.9503	0.8152	0.8594	0.8177	0.6569	0.6594	0.4918	0.3894	0.8986	0.8171	0.8423	**0.7453**

Her satır, her yöntemin sonuçlarını göstermektedir. Her sütun, her kategorideki ortalama sonuçları göstermektedir. Not olarak bu değerlendirmede CDnet2014 veri kümesinin ground-truth'ları tüm çerçeveler kullanılmıştır

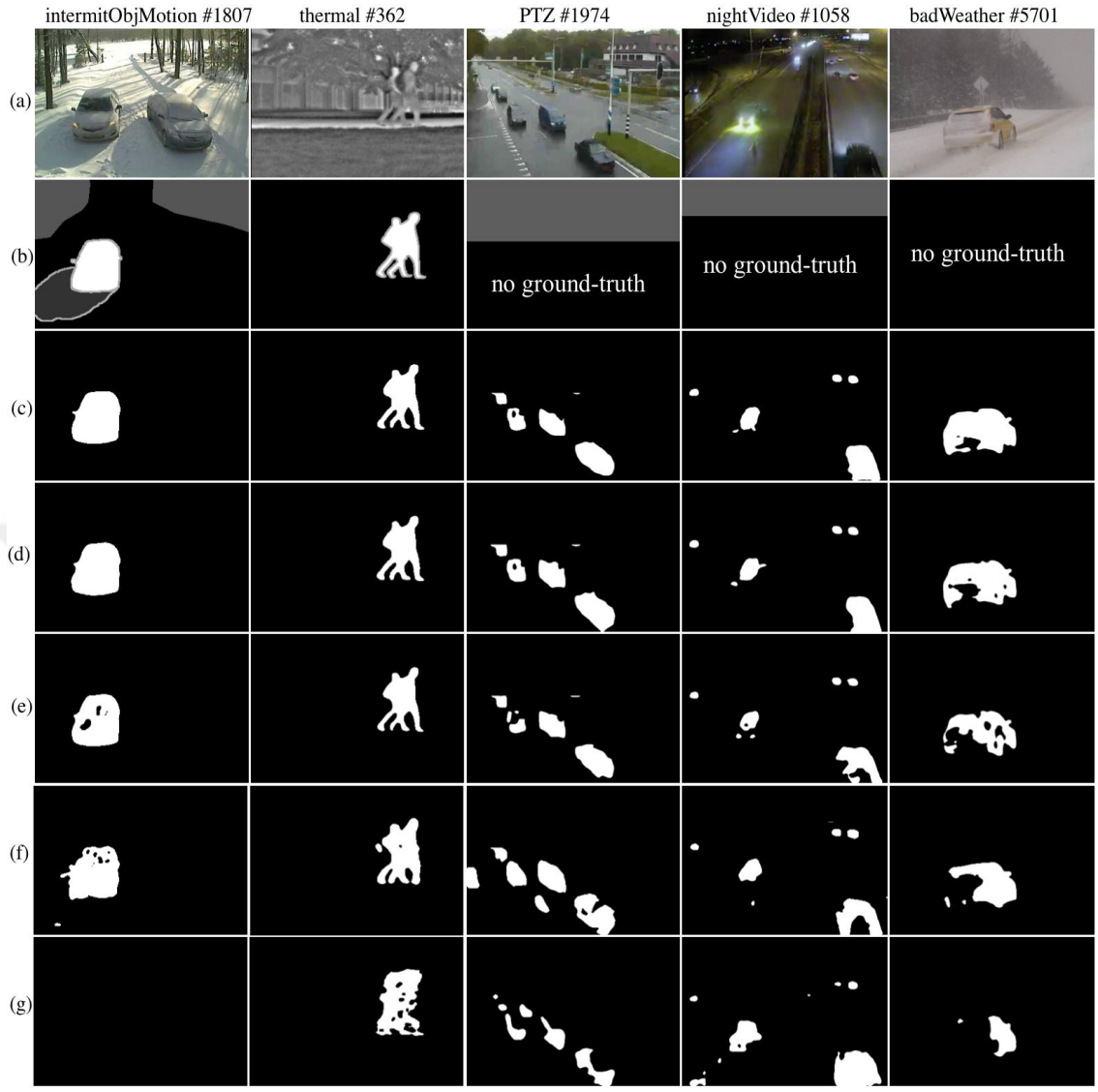
Çizelge 2.14 Son teknoloji yöntemler ile bir karşılaştırma

Methods	Overall			
	Precision	Recall	PWC	F-Measure
FgSegNet_v2	**0.9823**	**0.9891**	**0.0402**	**0.9847**
FgSegNet_S	0.9751	0.9896	0.0461	0.9804
FgSegNet_M	0.9758	0.9836	0.0559	0.9770
Cascade	0.8997	0.9506	0.4052	0.9209
DeepBS	0.8332	0.7545	1.9920	0.7458
IUTIS-5	0.8087	0.7849	1.1986	0.7717
PAWCS	0.7857	0.7718	1.1992	0.7403
SuBSENSE	0.7509	0.8124	1.6780	0.7408

Bu ortalama sonuçlar Change Detection 2014 yarışmasından elde edilmektedir

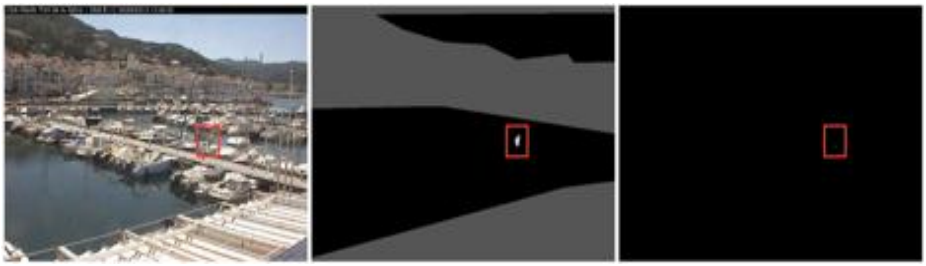
Tüm yöntemlerin genelleme yeteneğini veri setinden bazı örnek çerçevelerle görüntülemek için segmentasyon sonuçları iki şekilde sağlanmaktadır; ilk olarak sağlanan ground-truth'lar aralığında rastgele bazı segmentasyon sonuçları örnek olarak seçilmektedir (örn. *intermittenObj.motion* ve *thermal kategori*), ikinci olarak ground-truth'ların halka açık olmadığı test setinden bazı örnek kareleri rastgele olarak seçilmektedir (örn. *PTZ, nightVideo* ve *badWeather kategori*). Şekil 2.30'dan görülebileceği gibi, yöntemimiz iyi segmentasyon sonuçları vermektedir; özellikle, bir arabanın uzun süre durduğu ve hemen hareket etmeye başladığı *intermittenObjmotion* kategorisinde. Bu senaryo çoğu yöntemin önemli ölçüde başarısız olduğu durumdur (örn. (g) metodu). Ek olarak, şekil 2.30 (*PTZ, nightVideo, badWeather categories*)'te görülebileceği gibi, yöntemimiz, sahnelerin zorlu içeriğinin zamanda değiştiği ve tamamen görülmeyen verilere iyi bir şekilde genelemektedir. Yöntemimiz diğer yöntemlere göre iyi segmentasyon maskeleri üretmektedir.

Önerdiğimiz yöntemin diğer kategorilerle karşılaştırıldığında (F-Measure ~ 0.9579) zayıf performans gösterdiği LowFrameRate kategorisi dışında neredeyse tüm kategorilerde daha iyi performans göstermektedir (Çizelge 2.13'e bakınız). Bu düşük performans, temel olarak kademeli aydınlatma değişikliklerine sahip dinamik sahnelerde çok küçük ön plan objelerinin bulunduğu zorlu bir video dizisinden kaynaklanmaktadır (Şekil 2.31). Bu durumda ağ, ana sınıfa daha fazla ilgi gösterebilmektedir (bg) ancak nadir sınıfa (fg) daha az dikkat edebilmektedir. Sonuç olarak çok küçük ön plan nesnelerini yanlış sınıflandırmaktadır.



Şekil 2.30 5 yöntem arasında bazı karşılaştırmalar
a. input images; b. ground-truth'lar; c. önerilen segmentasyon sonuçları; d. *FgSegNet_S* sonuçları; e. *FgSegNet_M* sonuçları; f. Cascade sonuçları; g. DeepBS sonuçları

Buna rağmen, önerilen yöntem, bu kategoride bazı puanlarda hala en iyi yöntemi geliştirmektedir. Ancak yöntemimiz sahnedeki harmanlanan nesneleri tespit etmekte başarısız olmaktadır (Şekil 2.28.c). Bu durumda bir yaya, arka plana tamamen karışmıştır ve insanın ön plan ile arka plan arasında ayrım yapması bile zordur.



Şekil 2.31 Yöntemimizin zayıf performans gösterdiği *lowFrameRate* kategorisindeki video dizisi

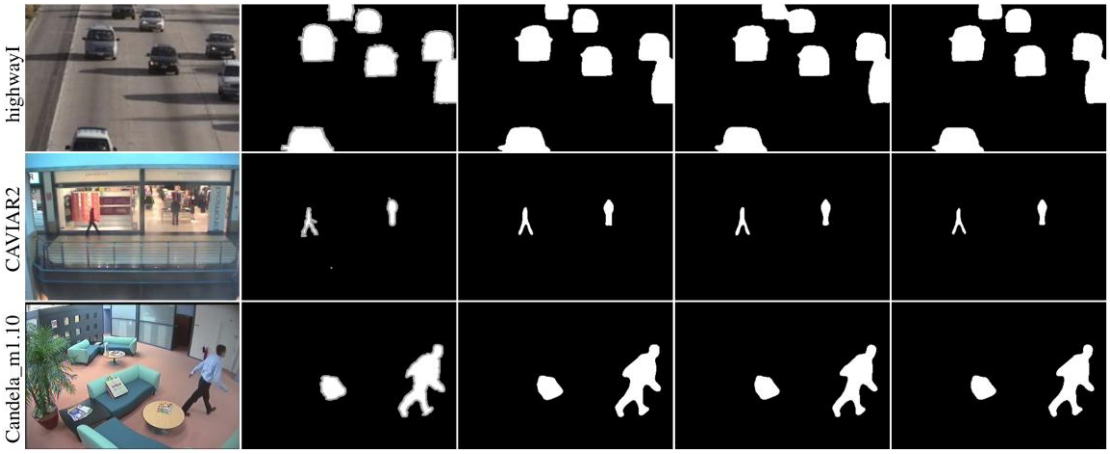
Birincisi, ikinci ve üçüncü sütun, giriş imajını, ground-truth'larını ve segmentasyon sonucumuzu göstermektedir

2.6.2.6.2 SBI2015 veri setinde

Test sonuçları çizelge 2.15'de gösterilmektedir. Görülebileceği gibi, yöntemimiz bazı puanlarda önceki yöntemlerden daha iyi performans göstermektedir. FgSegNet_v2, FgSegNet ailesini % 0.22 ve % 0.59 puan artırırken, Cascade (Wang vd. 2016)'i % 9.21 puan artırmaktadır. Bezer şekilde metodumuzun PWC'si diğer yöntemlere göre önemli oranda daha azdır. *Toscana* dizisinde en düşük performansı 0.9291 F-Measure ile elde edilmektedir. Bu öncelikle çok az sayıda eğitim çerçevesinin kullanılmasından kaynaklanmaktadır (*eğitim+validasyon* için sadece 2 kare). Şekil 2.32'de bazı örnek sonuçlar gösterilmektedir. Görülebileceği gibi, yöntemimiz iyi segmentasyon maskeleri üretmektedir; özellikle, *highwayI* video dizisinde gölgeleri tamamen ortadan kaldırılmaktadır.

Çizelge 2.15 SBI2015 veri setinde 0.3 eşk değeriyle test sonuçları ve en son teknoloji yöntemlerle bazı karşılaştırmalar

Video Seq.	FPR	FNR	F-Measure	PWC
Board	0.0009	0.0019	0.9979	0.1213
Candela_m1.10	0.0003	0.0037	0.9950	0.0399
CAVIAR1	0.0001	0.0007	0.9988	0.0086
CAVIAR2	0.0001	0.0092	0.9834	0.0131
CaVignal	0.0027	0.0076	0.9859	0.3310
Foliage	0.0771	0.0207	0.9732	3.7675
HallAndMonitor	0.0002	0.0051	0.9926	0.0357
HighwayI	0.0008	0.0068	0.9924	0.1358
HighwayII	0.0001	0.0051	0.9952	0.0289
HumanBody2	0.0009	0.0079	0.9920	0.1636
IBMtest2	0.0007	0.0220	0.9817	0.1680
PeopleAndFoliage	0.0066	0.0102	0.9919	0.8468
Snellen	0.0211	0.0147	0.9644	1.7573
Toscana	0.0046	0.1155	0.9291	2.5901
Proposed	**0.0083**	**0.0165**	**0.9853**	**0.7148**
FgSegNet_S	**0.0090**	**0.0146**	**0.9831**	**0.8524**
FgSegNet_M	**0.0059**	**0.0310**	**0.9794**	**0.9431**
Cascade	–	–	**0.8932**	**5.5800**



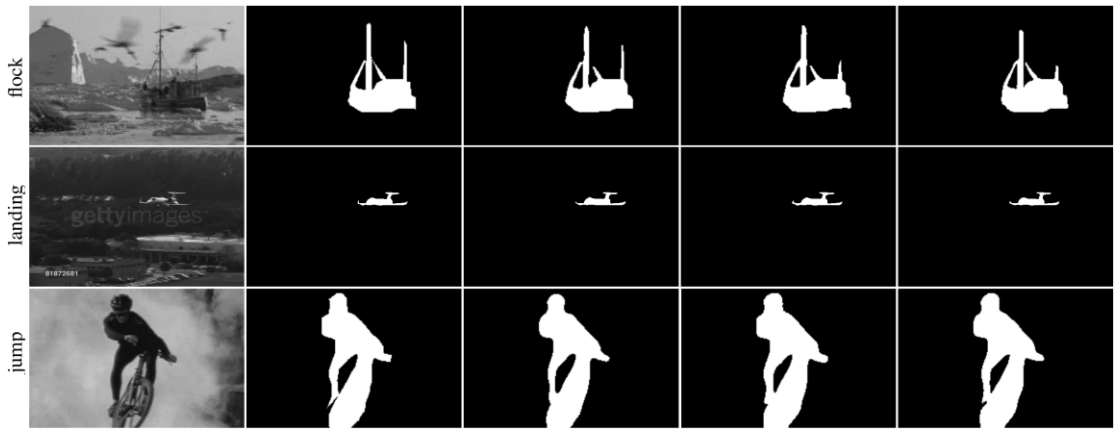
Şekil 2.32 SBI2015 veri setinde bir karşılaştırma
Her sütun ham görüntüleri, ground-truth'ları, segmentasyon sonuçlarımızı, FgSegNet_S ve FgSegNet_M sonuçlarını sırasıyla göstermektedir

2.6.2.6.3 UCSD Background Subtraction veri setinde

Test sonuçları çizelge 2.16'da gösterilmektedir. Çok az sayıda *eğitim+validasyon* çerçevesi olmasına rağmen, % 20 bölünme durumunda ortalama 0.8945 ve % 50 bölünme durumunda da ortalama 0.9203 F-Measure'ları elde edilmektedir. Yöntemimiz önceki yöntemlerle yarışabilir sonuçlar vermekle birlikte PWC önceki yöntemlere göre dikkate değer bir şekilde azalmaktadır. Şekil 2.33'te bazı segmentasyon sonuçları gösterilmektedir.

Çizelge 2.16 UCSD veri setinde 0.6 eşk değeriyle test sonuçları ve en son teknoloji yöntemlerle bazı karşılaştırmalar

Video Seq.	20% split				50% split			
	FPR	FNR	F-Measure	PWC	FPR	FNR	F-Measure	PWC
birds	0.0025	0.1423	0.8649	0.5205	0.0021	0.1162	0.8884	0.4315
boats	0.0009	0.0729	0.9213	0.1678	0.0008	0.0382	0.9437	0.1213
bottle	0.0014	0.0406	0.9550	0.2462	0.0009	0.0476	0.9605	0.2134
chopper	0.0023	0.0760	0.9140	0.3991	0.0017	0.0810	0.9232	0.3544
cyclists	0.0030	0.0738	0.9213	0.5382	0.0019	0.0488	0.9492	0.3457
flock	0.0048	0.0641	0.9383	0.9270	0.0036	0.0375	0.9591	0.6179
freeway	0.0013	0.3012	0.7787	0.5480	0.0028	0.1349	0.8394	0.4635
hockey	0.0197	0.0716	0.9165	2.8430	0.0138	0.0527	0.9400	2.0359
jump	0.0066	0.0746	0.9358	1.4309	0.0041	0.0464	0.9603	0.8892
landing	0.0007	0.0653	0.9245	0.1278	0.0006	0.0559	0.9388	0.1031
ocean	0.0012	0.1014	0.8931	0.2253	0.0010	0.0607	0.9243	0.1615
peds	0.0048	0.0955	0.8776	0.7499	0.0038	0.0907	0.8942	0.6402
rain	0.0029	0.1180	0.9174	1.0490	0.0025	0.0558	0.9534	0.5881
skiing	0.0022	0.0846	0.9171	0.4338	0.0018	0.0586	0.9385	0.3251
surfers	0.0015	0.1147	0.8887	0.2990	0.0012	0.0764	0.9173	0.2232
surf	0.0008	0.2941	0.7307	0.1778	0.0005	0.2321	0.7968	0.1343
traffic	0.0017	0.0899	0.9070	0.3186	0.0015	0.0566	0.9301	0.2427
zodiac	0.0003	0.0735	0.8988	0.0429	0.0002	0.0815	0.9086	0.0383
Proposed	**0.0033**	**0.1086**	**0.8945**	**0.6136**	**0.0025**	**0.0762**	**0.9203**	**0.4405**
FgSegNet_S	**0.0058**	**0.0559**	**0.8822**	**0.7052**	**0.0039**	**0.0544**	**0.9139**	**0.5024**
FgSegNet_M	**0.0037**	**0.0904**	**0.8948**	**0.6260**	**0.0027**	**0.0714**	**0.9203**	**0.4637**



Şekil 2.33 UCSD veri setinde bir karşılaştırma
Her sütun sırasıyla ham görüntüleri, ground-truth'lar, FgSegNet_M ve FgSegNet_S sonuçlarını göstermektedir

2.6.3 İşleme hızı

Daha önce de açıklandığı gibi Tensorflow tabanlı Keras ve NVIDIA GTX 970 GPU kullanılarak uygulama sürecimiz hızlandırılmaktadır. FgSegNeg_M ve 200-çerçeve deneyleriyle 320x240 boyutunda 1700 kareye sahip bir video dizisi düşünüldüğünde 50 epoch için yaklaşık 23.7 dakika sürmektedir. Test sırasında kalan 1500 kareyi bölütlemek için yaklaşık 1.39 dakika sürmektedir, yani FgSegNet_M'nın saniyede yaklaşık 18 kare bölütleme yeteneğine sahip olduğu anlamına gelmektedir. FgSegNet_S ise saniyede yaklaşık 21 kare bölütleme yeteneğine sahiptir. FgSegNet_v2 için saniyede yaklaşık 23 kere segmentasyon yeteneğine sahiptir. Bunun sonucu olarak en iyi önceki yönteme kıyasla, yöntemimiz eğitim ve segmentasyon hızı açısından daha hızlıdır (Çizelge 2.7, en sağdaki sütun).

3. SEMANTİK SEGMENTASYON

Tezin bu bölümünde, geliştirdiğimiz iki ayrı yöntem üzerinden semantik segmentasyon alanında yürüttğümüz çalışmalar sunulmaktadır. İlk yöntemde orijinal FgSegNet ağ mimarisi kullanılmak yerine mevcut mimaride bazı değişiklikler önerilmektedir. İkinci yöntemde doğruluğu önemli ölçüde artırdığını bulduğumuz yeni bir mimari önerilmektedir. Not olarak; ilk yöntem için yaptığımız deneylerde için herhangi bir parametre araştırması veya en uygun mimari uygulanmamaktadır. Çalışmamız Tensorflow tabanlı Keras çerçevesi kullanılarak gerçekleştirilmektedir. Her iki yaklaşımda tahmin edilen segmentasyon maskelerinde conditional random field (CRF) veya diğer grafik modeller gibi herhangi bir post-processing tekniği uygulanmamaktadır.

3.1 Literatür İncelemesi

Derin öğrenme yöntemleri, özellikle konvolüsyel sinir ağları (CNNs), çeşitli tanıma problemlerinde büyük başarılar göstermişlerdir. Bu başarılara motive ettiğinde araştırmacıların çoğu, semantik segmentasyon alanı için bu tür ağın yeteneklerini keşfetmektedir ve bu alanı daha iyi hale getirmektedir. Son zamanlarda semantik segmentasyon alanında Long vd. (2015) tarafından tam konvolüsyonel ağlar veya Fully Convolutional Networks (FCN) önerilmiştir. Araştırmacılar tarafından AlexNet (Krizhevsky vd. 2012), VGG-Net (Simonyan ve Zisserman 2014) ve GoogLeNet (Szegedy vd. 2015) gibi ILSVRC (ImageNet Large Scale Visual Recognition Challenge) yarışmasında önceden eğitilmiş ağın tam bağlantılı katmanları (veya Fully-Connected Layers) tam konvolüsyonel formlara dönüştürülmüştür. Ağda öznitelik çözünürlüklerini arttırma (veya in-network upsampling) ve atlama mimarisi (veya skip architecture), semantik çıktıları düzeltmek için tanıtılmıştır. Long vd. (2015)'in yöntemi, Pascal VOC 2011-12 ve SIFT Flow gibi çeşitli veri setlerinde performansı artırmıştır. Bu ağ, keyfi boyutların giriş görüntülerini alarak ve kompakt tahminler üreterek uçtan uca eğitilebildiği için bu çalışma bir dönüm noktası olarak kabul edilmiştir.

Bunun dışında FCN'nin sınırlamasını azaltmak için Noh vd. (2015) tarafından bir dekoder yaklaşımı önerilmiştir. Bu çalışmada dekonvolüsyon katmanları ve unpooling katmanları, önceden eğitilmiş sınıflandırma modelinin (VGG-16 Net) üstüne simetrik bir şekilde yerleştirilmiştir. Noh vd. (2015)'in ağı, bir giriş görüntüsünde her nesne teklifine uygulanmış ve daha sonra final segmentasyon çıktısını oluşturmak için tüm tekliflerden elde edilen semantik çıktılar birleştirilmiştir. Bu teknik, FCN'nin sabit alıcı alanından (fixed receptive field) kaynaklanan sınırlamayı hafifletmiştir.

SegNet, Badrinarayanan vd. (2017) tarafından önerilen teknoloji harikası bir yöntemdir ve bu ağ, CamVid veri seti (Brostow vd. 2009) ile eğitilmiştir. Ağ, önceden eğitilmiş VGG-Net'in üstüne yerleştirilmiş bir dekoder ağından oluşturulmuştur. Dekoder ağı, konvolüsyon katmanları ve upsampling katmanlarından oluşmuştur, burada enkoder ağının maksimum pooling (veya max-pooling) indeksleri dekoder kısmında lineer olmayan öznitelik çözünürlüğünü artırmasını (veya non-linear upsampling) gerçekleştirmek için kullanılmıştır. SegNet, FCN (Long vd. 2015) ve DeconvNet (Noh vd. 2015) gibi mevcut yöntemlerden daha iyi performans göstermiştir.

Video semantik segmentasyonda yeni bir başka yöntem Jegou vd. (2017) tarafından önerilmiştir. Bu çalışmada yazar tarafından herhangi bir önceden eğitilmiş ağ kullanmadan ağları sıfırdan eğitilmiştir. Fakat, Jegou vd. (2017)'nin ağ mimarisi, Huang vd. (2017) tarafından tanıtılan görüntü sınıflandırma ağı, DenseNets'e dayanılarak tasarlanmıştır. Önerilen yaklaşım, CamVid ve Gatech (Raza vd. 2013) veri kümesinde teknoloji harikası doğruluğunu geliştirmiştir.

ENet adlı başka bir etkili semantik segmentasyon yöntemi Paszke vd. (2016) tarafından önerilmiştir. Bu çalışmada ResNet (He vd. 2016) ağının fikirleri yazarlar tarafından benimsenmiştir. Üstelik, özellikle hızlı çıkarımlar için modeli tasarlanmış ve düşük hesaplama maliyetlerine sahiptir; yine de önerilen yöntemler mevcut yöntemlere benzer sonuçlar sağlamıştır.

Video semantik segmentasyon problemi için artık birleşik bir konvolüsyon ağı veya residual coalesced convolutional network (RCC-Net) Ardiyanto ve Adji (2017)

tarafından önerilmiştir. Bu enkoder-dekoder tipi ağ mimarisi, ResNet modelinin ve Inception (Szegedy vd. 2015) ağının esinlenmesine dayanılarak tasarlanmıştır. Hesaplama verimliliği için enkoder ağına geçmeden önce giriş görüntülerinin boyutlarını küçük öznitelik haritalarına indirgemek için bir başlangıç modül yazar tarafından kullanılmıştır. Deney sonuçunda bu ağ teknoloji harikası yöntemlerinden daha iyi performans göstermiştir.

Ek olarak, genişlemiş konvolüsyon veya atrous konvolüsyonu semantik segmentasyon görevinde başarıyla kullanılmıştır (Yu ve Koltun 2015, Chen vd. 2017, Chen vd. 2018a, Chen vd. 2018b). Genişlemiş konvolüsyonunun fikri, ek parametreler öğrenmeden ağdaki görüş alanlarını genişletmektir. Atrous konvolüsyonu kullanarak önerilen yöntemler, çeşitli kıyaslamalarda mevcut olan en ileri teknoloji yöntemlerinden daha iyi performansla çalışmaktadır.

3.2 Materyal

Bu problem alanında kullanılan materyallar için bölüm 2.2.1 ve 2.2.2'ye bakınız.

3.2.1 Veri kümesi

Bu deneyimizde 360x480 çözünürlükte 367 eğitim, 101 validasyon ve 233 test görüntüsünden oluşan CamVid veri seti kullanılmaktadır. CamVid yarışması video sahnelerini yol, bina, araba, yaya, kaldırım, trafik işareti gibi 11 sınıfa ayırmaktır. Ağımız, Badrinarayanan vd. (2017)'de açıklandığı gibi aynı eğitim çerçeveleri kullanılarak eğitilmektedir.

3.2.2 Değerlendirme metrikleri

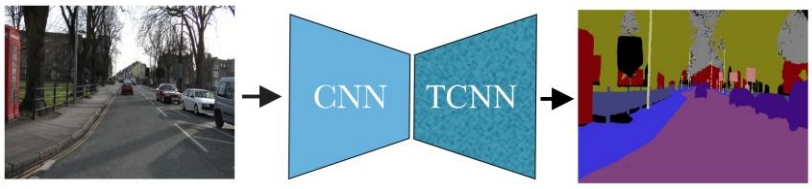
Ağımızın performansı, Badrinarayanan vd. (2017)'yi takip eden üç metrik kullanılarak ölçülmektedir: bunlar doğru şekilde sınıflandırılan piksellerin yüzdesi (G), tüm sınıflara

göre tahmin edilen doğruluk veya sınıf ortalama doğruluk (C), ve tüm sınıflar üzerinde ortalama kesişim veya mean intersection over union (mIoU).

3.3 Yöntem 1

3.3.1 Ağ konfigürasyonu ve eğitim detayları

Bu ön çalışma için FgSegNet mimarisine çok ölçekli görüntüler yerine tek bir ölçek kullanarak adapte edilmektedir (Şekil 3.1). Tam FgSegNet ağ konfigürasyonu için okuyucu yayınlanan makaleye veya bu tezin ilk bölümüne bakabilir.



Şekil 3.1 Değiştirilen FgSegNet mimarimiz

Bu deneyde FgSegNet yöntemimizde açıklandığı gibi aynı ağ konfigürasyonu kullanılmaktadır. 367 görüntüden 1468 görüntüye kadar eğitim boyutlarını arttırmak için yatay olarak çevirerek, +10/-10 derece görüntü döndürerek veri büyütmesi gerçekleştirilmektedir. Piksel sınıflandırmasında büyük bir problem, dengesiz sınıf problemidir, baskın sınıflardaki eğitim örneklerinin sayısı, nadir sınıflardaki eğitim örneklerinin sayısından daha ağır basmaktadır. Not olarak; bu deneyimizde sınıf dengelemesi yapılmamaktadır. Yöntemlerimizin, sınıfları dengeleyerek daha da geliştirilebileceğini ummaktayız. İkinci deneyde semantik segmentasyonda bu dengesiz sınıf problemi ele alınacaktır.

3.3.2 Sonuçlar ve tartışmalar

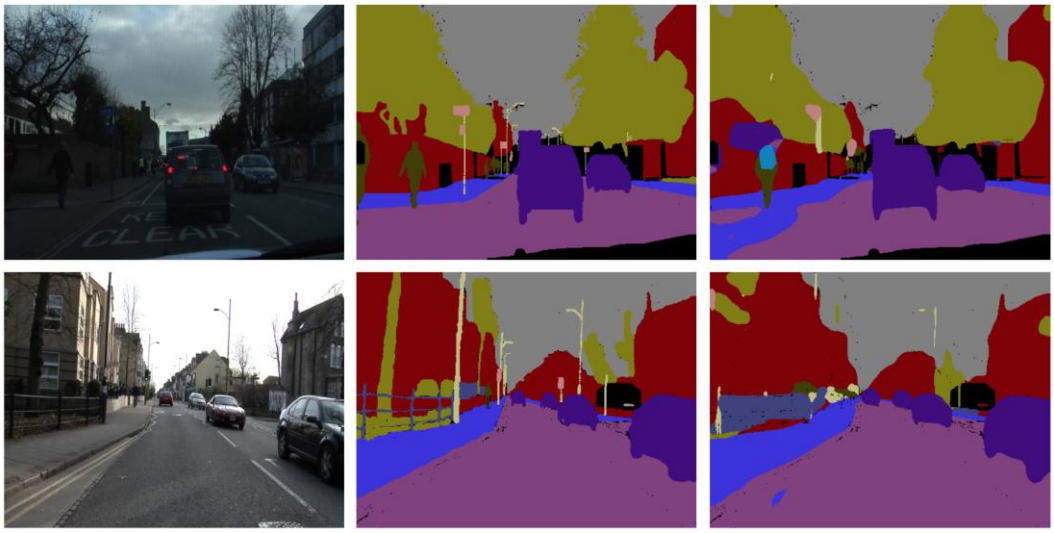
CamVid test setinde teknoloji harikası yöntemleri ile yöntemlerimiz arasında bir karşılaştırma çizelge 3.1'de gösterilmektedir.

Çizelge 3.1 Yöntemlerimiz ile CamVid test setindeki en ileri teknoloji yöntemleri arasında bir karşılaştırma

Metotlar	**G**	**C**	**mIoU**
Ours (exp1)	**88.69**	**66.66**	**58.03**
Ours (exp2)	**90.15**	**74.08**	**62.98**
ReSeg (Visin vd. 2016)	88.7	68.1	58.8
RCC-Net (Ardiyanto ve Adji 2017)	~	71.5	53.3
ENet (Paszke vd. 2016)	~	68.3	51.3
SegNet-Basic (Badrinarayanan vd. 2017)	84.20	56.50	47.70
SegNet (Badrinarayanan vd. 2017)	88.6	65.9	50.2
FCN (Long vd. (2015)	83.50	57.30	47.00
DeconvNet (Noh vd. 2015)	85.6	~	48.9

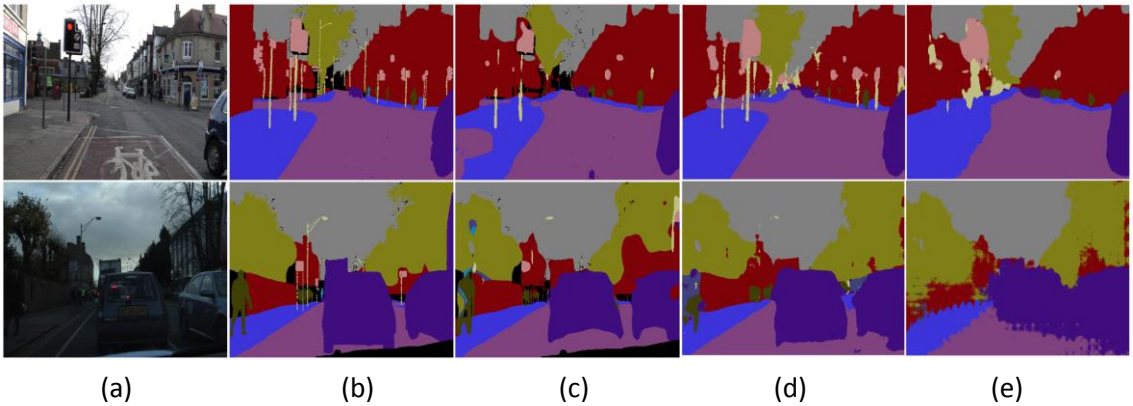
G (Global accuracy), C (Class average acuracy), mIoU (mean Intersection over Union)

Yöntemimiz (exp1) çoğu metrikte listelenen yöntemlerden daha iyi çalışmaktadır. Şekil 3.2'de segmentasyon sonuçlarımız gösterilmektedir. Görüldüğü gibi yöntemimiz karanlık bir sahnede olmasına rağmen düzgün segmentasyon maskeleri üretebilmektedir. Sahnelerin çoğu bölümleri doğru sınıflandırılmaktadır; öznitelikle yol, bina, ağaç veya araba gibi baskın sınıflar. Ancak, sahnelerin bazı parçaları ağımız tarafından sınıflandırılmamaktadır; yol kaldırım olarak sınıflandırılmakta veya binanın bir kısmı araba olarak sınıflandırılmakta (ilk satır, son sütun), ve ağaç bina olarak sınıflandırılmaktadır (ikinci sıra, son sütun). Üstelik, direkler veya bisikletçiler gibi nadir sınıflar ağımız tarafından yanlış sınıflandırılmaktadır. Bu problem dengesiz sınıf problemindenlerine sebep olabilmektedir, burada ağımız sadece majör sınıflara daha fazla dikkat etmektedir.



Şekil 3.2 Yöntem 1'nin sonuçları
İlk sütünde ham görüntülerdir. İkinci sütünde ground-truth etiketleridir. Üçüncü sütünde segmentasyon sonuçlarımız

Yukarıdaki çizelgedan iki teknoloji yöntemi (SegNet, FCN) seçilmiştir ve şekil 3.3'te bir karşılaştırma sunulmaktadır. Görüldüğü gibi yöntemimiz yayalar gibi küçük nesneleri karanlık mekanda bölütleyebilmektedir, burada insan gözlemcinin bu nesneleri tespit etmesi bile zordur. Fakat, ağımız trafik ışıkları veya direkler gibi bazı nadir sınıfları yanlış sınıflandırmaktadır.



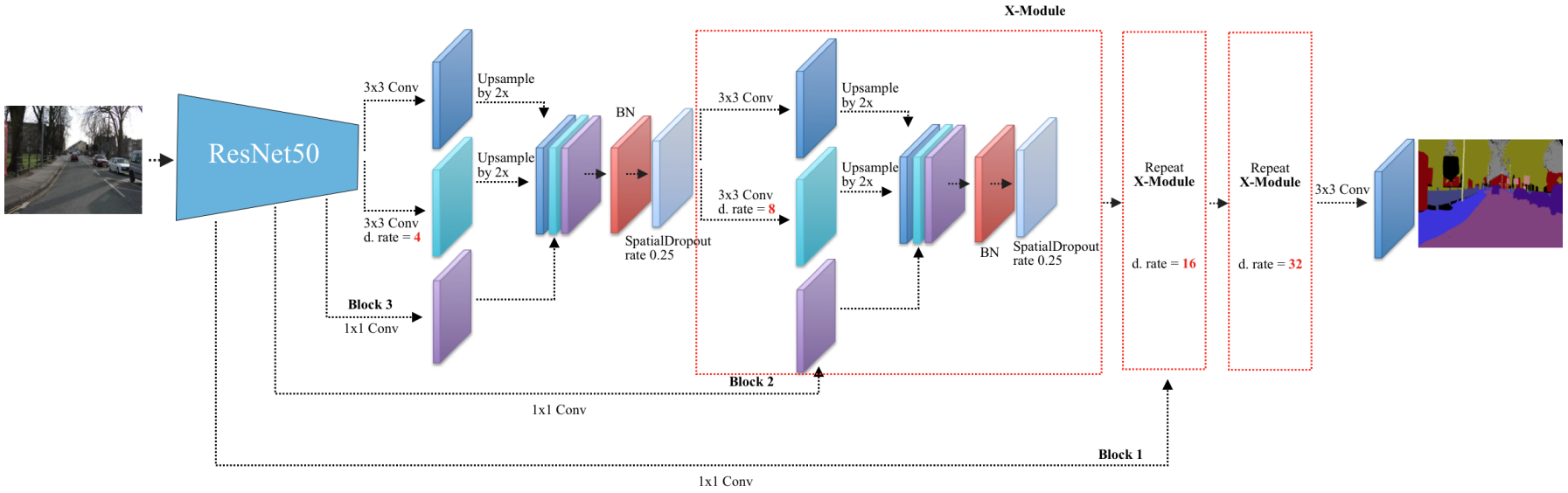
Şekil 3.3 CamVid test setinde metodumuz ile teknoloji harikası yöntemleri arasında bir karşılaştırma
a. ham görüntülerdir; b. ground-truth etiketleridir; c. segmentasyon sonuçlarımız; d. ve e. sırasıyla SegNet'in ve FCN'nin sonuçları gösterilmektedir

3.4 Yöntem 2

3.4.1 Ağ konfigürasyonu

Bu deneyimizde VGG Net (Simonyan ve Zisserman 2014) kullanmak yerine, ResNet 50 katmanları olarak da bilinen önceden eğitilmiş ResNet-50 (He vd. 2016)'yı uyarlanmak ve şekil 3.4'te gösterildiği gibi bir ağ mimarisi tasarlanmaktadır. Detaylı ResNet ağı mimarisi için orijinal makaleye bakılabilir. ResNet-50'nin ilk dört bloğu (conv1, conv2_x, conv3_x, conv4_x) kullanılmakta, ince ayar için tutulan conv4_x'in son özdeşlik bloğunun dışında ağın tüm katmanlarını dondurulmaktadır. Atlama bağlantısı (skip connection) tekniği, ağ üzerinden gradyan akışını kolaylaştırmak için görüntü tanıma görevinde (He vd. 2016) ve enkoder kısmında pooling yapıldığında kaybolan ilgili özniteliklerden öğrenmesini sağlamak için semantik segmentasyon görevinde (Ronneberger vd. 2015) başarıyla uygulanmıştır. Bu fikirlerden hareketle uygulamamızda atlama katmanları (skip layers) uygulandığında ağın daha hızlı yakınsadığını ve doğruluğunu geliştirdiğini keşfetmekteyiz.

Dekoder ağımız, birbirinin üzerine eklenmiş dört modülden oluşmaktadır (Şekil 3.4), her modül tam olarak aynı konfigürasyonu içermektedir, ancak dilatasyon oranı ilk modülden son modüle 2x artırılmaktadır. Her modül için bir öznitelik haritasının bir seti $\boldsymbol{F}$ verildiğinde paralel olarak iki tane konvolüsyon işlemi çalıştırılmaktadır.



Şekil 3.4 Önerilen ağ mimarisimiz

İlk olarak, HxWx64 boyutlu bir öznitelik haritalarının bir seti *(M)* üretmek için belirli bir öznitelik haritası *(F)* üzerinde 1 adım ile sabit bir alıcı alan 3x3 konvolüsyonu çalıştırılmaktadır. Sonrasında öznitelik haritasının *(M)*, H'xW'x64 boyutlu bir öznitelik haritaların bir seti *(M')* üretmek için çözünürlüğünü 2 kat artırılmaktadır.

- İkinci olarak, HxWx64 boyutlu bir öznitelik haritalarının bir seti *(N)* üretmek için genişleme oranı *(R)* ve 1 adım ile bir genişlemiş 3x3 konvolüsyonu aynı öznitelik haritası *(F)* üzerinde çalıştırılmaktadır. Öznitelik haritasının *(N)*, H'xW'x64 boyutlu bir öznitelik haritalarının bir setini *(N')* üretmek için çözünürlüğünü 2 kat artırılmaktadır.
- Üçüncü olarak, birinci, ikinci ve üçüncü modülde 1x1 konvolüsyonu, enkoder kısmında yüksek boyutlu öznitelik harita derinlikleri düşük boyutlu *(P)* H'xW'x64 boyutuna yansıtılmaktadır. Not olarak, enkoder kısmının blok 2 ve blok 3'teki atlama katmanları, son özdeşlik bloğundan alınmaktadır (BatchNormalization (Ioffe ve Szegedy 2015)'den hemen önce ve bu özdeşlik bloğundan ekleme işleminden önce bu altlama katmanları seçilmektedir).
- Son olarak, *H'xW'x192* öznitelik haritaları üretmek için derinlik ekseni boyunca öznitelik haritası *M'*, *N'* ve *P* birleştirilmiştir, dördüncü modül ise atlama katmanı kullanmamaktadır. Not olarak; birleştirilmiş öznitelikler sırasıyla BatchNormalization, ReLU ve SpatialDropout (Tompson vd. 2015) katmanları tarafından takip edilmektedir. Bunların ardından, bileşke öznitelik haritaları bir sonraki katmandan geçirilmektedir.

Ek Analiz: 3x3-konvolüsyonun sabit alıcı alanı, global bilgileri dahil etmeden lokal bölgeleri öğrenme olarak yorumlanabilir, genişlemiş konvolüsyon ise ağın bağlamsal bilgileri dikkate alarak yoğun tahmin için önemli olan geniş bir alıcı alana sahip olmasını sağlamaktadır. Bizim uygulamamızda bu iki türden konvolüsyon sonucu önceki atlama katmanlarında ilgili bilgilerle birleştirilmekte ve daha sonra yoğun bir tahmin öğrenmek için bir sonraki katmandan geçirilmektedir. Bitişiklik pikselleri aşırı eğitim ve overfitting neden olan güçlü bir korelasyon olduğu için SpatialDropout (Tompson vd. 2015)'in uygulamamızda etkili olduğu bulunmuştur. Her modülde (Şekil 3.4), birleşik öznitelik haritası aynı girişin üzerinde çalıştırılan iki tane konvolüsyon işleminin sonucu olduğu için bu birleşik öznitelik haritasının güçlü bir şekilde ilişkili

olması beklenmektedir. Bu durumda, ilişkili öznitelik haritalarını % 25'lik bir dropout oranıyla düşürmek için birleşik öznitelikten hemen sonra SpatialDropout uygulanmaktadır.

3.4.2 Eğitim detayları

Bu deneyde eğitim boyutları yatay olarak çevirerek, +10/-10 derece döndürerek genişletmek için veri büyütmesi gerçekleştirilmektedir. Ağın kutup veya trafik ışığı gibi nadir sınıflara dikkat etmesi için başka bir teknik vardır. Bu teknikte bu nadir sınıflara odaklanarak giriş görüntüsünde yakınlaştırma ve kırpma (sol ve sağ) gerçekleştirebilmektedir. Not olarak; bu deneyimizde görüntüler % 100 ve % 200 oranlarında sağa ve sola yakınlaştırma yapılmakta ve daha sonra yakınlaştırma yapılan görüntüler kırpılmaktadır (Şekil 3.5). Eğitim için 367 yakınlaştırılmış görüntüden yapay olarak genişletilen toplam 5138 görüntü vardır.

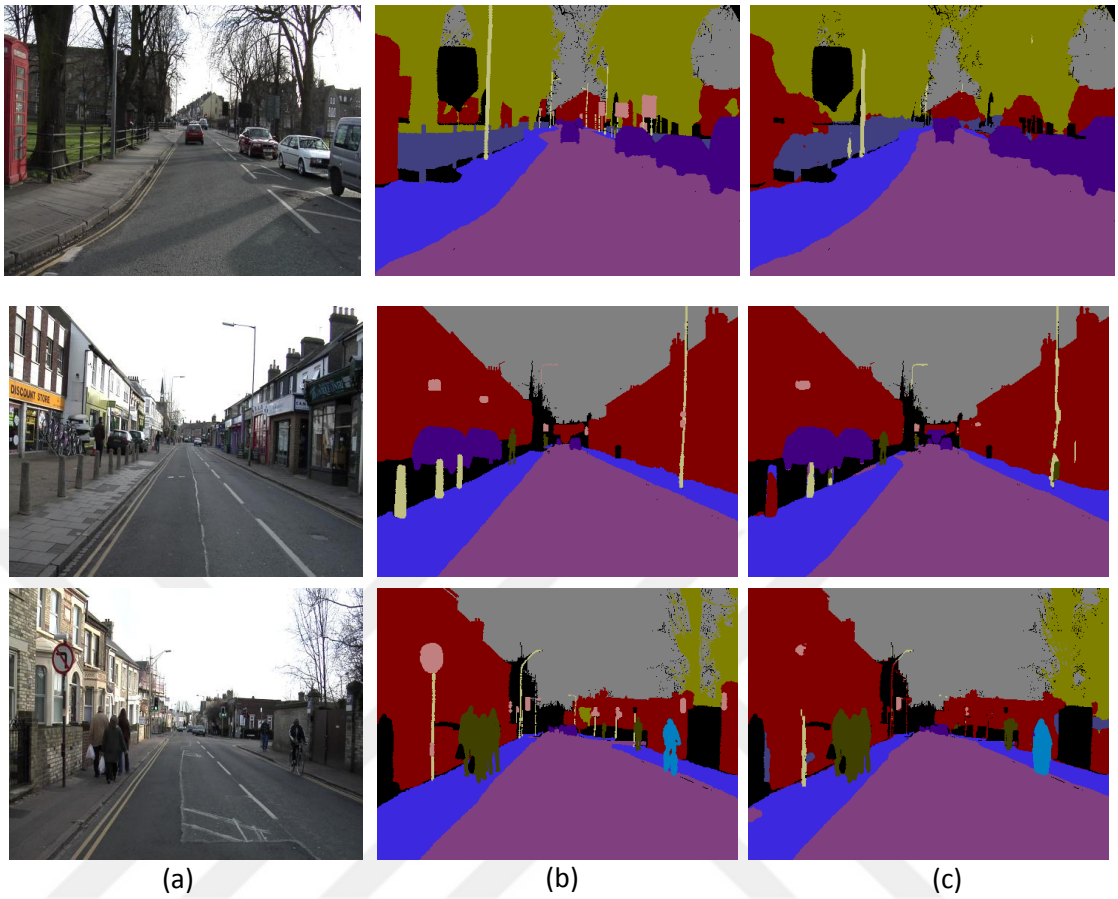
Şekil 3.5 Yakınlaştırma ve kırpma bir örnek

Not olarak; FgSegNet bölümünde açıklandığı gibi ağdaki her bir sınıf için dikkat seviyesi ayarlanmaktadır, burada sınıf ağırlıkları tüm eğitim örneklerinden hesaplanmıştır. Bu teknik ayrıca ağın nadir sınıfa dikkat çekmesine ve doğruluğunu artırmasına yardımcı olabilmektedir.

Ağımız, 1 batch-size ile *Adam* optimize edicisini kullanıp öğrenme oranını 1e-3'e ayarlanarak, 15 epok eğitilmektedir. Minimum validasyon kaybı 5 epok boyunca iyileşmeyi durdurduğunda öğrenme oranını 5 kat azaltılmaktadır. Minimum validasyon kaybı 10 epok boyunca gelişmeyi durdurduğunda erken durdurma (min_delta=1e-4) uygulanmaktadır. Eğitimimizde softmax çapraz entropi kaybı kullanılmaktadır. Not olarak; bu deneyde eğitim görüntüleri 352x480 çözünürlüğe yeniden boyutlandırılmaktadır.

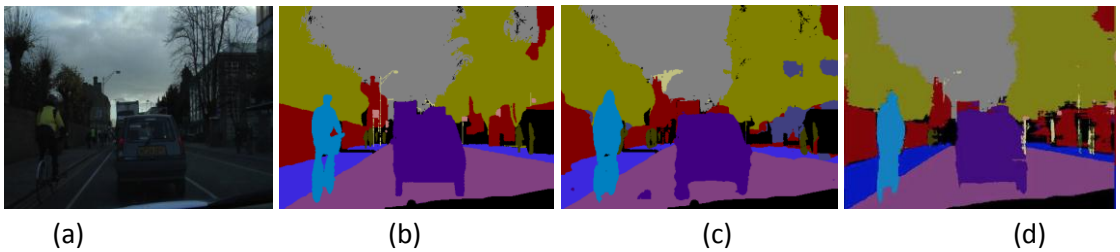
3.4.3 Sonuçlar ve tartışma

Bu deneyde yöntemimiz (Çizelge 3.1, *exp2*) tüm metriklerde listelenen yöntemlerden daha iyi performansla çalışmaktadır ve mIoU ilk yöntemimiz (*exp1*)'e kıyasla yaklaşık 5 puan geliştirmektedir. Nesne sınırlarının kaybolduğu *exp1*'in aksine, *exp2* keskin nesne sınırları oluşturmaktadır (Şekil 3.6). Sahnelerin çoğu parçaları özellikle yol, bina, ağaç veya araba gibi büyük sınıflara göre doğru bir şekilde sınıflandırılmaktadır. Dikkat değer bir şey ise *exp2*'nin *exp1*'in başarısız olduğu bisikletçiler ve yayalar arasındaki farkı ayırt edebilmesidir (Şekil 3.6.c). Bu gelişim, genişlemiş konvolüsyon kullanarak bağlamsal bilginin toplanmasından kaynaklanabilmektedir. Halbuki, sahne ile benzer renk yoğunluklarını paylaştıkları trafik ışıklarını tespit edememektedir.



Şekil 3.6 CamVid test setinde *exp2*'nin bazı sonuçları
a. ham görüntüler; b. ground-truth etiketleri; c. segmentasyon sonuçlarımız

Şekil 3.7'de modelimiz (*exp2*) ve ReSeg (Visin vd. 2016) modeli arasında bir karşılaştırma sağlanmaktadır.



Şekil 3.7 CamVid test setinde metodumuz (*exp2*) ve ReSeg (Visin vd. 2016) yöntemi arasında bir karşılaştırma
a. ham bir görüntü; b. ground-truth etiket; c. segmentasyon sonucumuz; d. ReSeg sonucu

4. SONUÇLAR VE ÖNERİLER

Bu tezin birinci kısmında (Bölüm 2) ön plan nesneleri segmentasyonu hakkında iki öğrenme stratejileri dayanan kapsamlı bir çalışma sunulmuştur: bunlar yama-tabanlı (patch-wise) öğrenme ve imge-tabanlı (image-wise) öğrenme stratejileridir.

Yama-tabanlı öğrenme stratejisi için iki eğitim stratejisi tartışılmıştır. Bu tartışmada Random Subset Training stratejisinin Entire Set Training stratejisine kıyasla etkili ve kabul edilebilir segmentasyon sonuçları sağladığı görülmüştür. Fakat, bu iki yöntem gerçek zamanlı uygulamalar için hesaplama açısından pahalı olduğundan pratik değildir.

İmge-tabanlı öğrenme stratejisi için sadece birkaç eğitim örneği kullanarak uçtan uca eğitilebilen üç gürbüz enkoder-dekoder tipi ağ modeli gösterilmiştir ve önerilen metotlar çeşitli zorlu sahnelerde yüksek doğrulukta segmentasyon maskeleri üretmiştir. Çok ölçekli bilgileri birleştiren üç tür ağ konfigürasyonu önerilmiştir. Bunlardan ilki VGG-16 Net'in daha yüksek katmanlarının bazıları değiştirilip, üçlü ağ konfigürasyonu altında alt katmanları değişmeden tutularak uyarlanmıştır. Yeni bir dekoder, geri görüntü uzayına öznitelik haritalarının çözünürlüğü arttırmak için enkoder ağının üstüne yerleştirilmiştir. İkinci olarak, çok ölçekli öznitelikleri ayıklamak için tek giriş enkoderının üstüne takılabilen bir Feature Pooling Module (FPM) önerilmiş ve aynı dekoder, görüntü uzayına projeksiyonu öğrenmek için çıkarılan özniteliklerin üzerine yerleştirilmiştir. Üçüncü olarak, FPM modülü çok ölçekli öznitelikleri birleştirerek değiştirilmiş ve sonuç olarak ağı çok-ölçekli girişlerle eğitme ihtiyacını azaltabilen kameranın hareketlerine karşı güçlü bir modül oluşturulmuştur. Daha fazla performans elde edilmesi için değiştirilen FPM'nin üzerine yeni bir dekoder ağı önerilmiştir.

Modellerimiz farklı veri kümeleri (Örn. CDnet2014, SBI2015 ve UCSD Background Subtraction) üzerinde değerlendirilmiş ve yöntemlerimizin aydınlatma değişiklikleri, arka plan veya kamera hareketi, kamuflaj etkisi, gölge gibi çeşitli zor durumlara karşı dayanıklı olduğunu gösterilmiştir. Önerilen metotlar hem iç hem de dış mekanlarda kullanılabilmektedir. Metotlarımızın önceki en iyi derin öğrenme tabanlı yöntem de dahil olmak üzere mevcut tüm yöntemlerden daha iyi çalıştığı gösterilmiştir.

Yöntemlerimiz, segmentasyon sonuçlarını düzeltmek için herhangi bir post-processing tekniği veya zamansal verileri dikkate almayı gerektirmemiştir.

Modellerimiz ön plan nesnelerini izole çerçevelerle öğrenmektedir, yani zaman dizisi, eğitim sırasında dikkate alınmamaktadır ve ağın öğrenmesi için birkaç eğitim çerçevesi yeterlidir. Gelecekteki bir çalışma olarak zamansal verileri bir araya getirmeyi ve çok az sayıda örnekle öğrenebilen bir yöntemi yeniden tasarlamayı planlamaktayız.

Tezin ikinci kısmında (Bölüm 3) semantik segmentasyon alanı ile ilgili kapsamlı bir çalışma sunulmuştur, burada ön plan nesneleri segmentasyonu alanındaki mevcut ağ (FgSegNet) ilk yöntemde uyarlanmıştır. İkinci yöntemde, yeni bir ağ mimarisi önerilmiş ve segmentasyon sonuçlarını önemli ölçüde artırmak için bazı faydalı teknikler uygulanmıştır. Bu kapsamda geliştirilen yöntemler oldukça iyi sonuçlar üretmekte ve tüm metriklerde en ileri teknoloji yöntemlerinden daha iyi performansla çalışmaktadır. Ağın, pooling işlemleri uygularken ilgili bilgileri kaybetme sorununu hafifletmek için tasarlandığından, önceden eğitilmiş mimari (residual networks or ResNet-50)'yi genişlemiş residuel ağ (dilated residula network (Yu vd. 2017)) ile değiştirilerek yöntemimizin daha da geliştirilebileceğine inanmaktayız. İleri araştırma olarak bahsetmiş olduğumuz problemlerin çözümleri araştırılacaktır.

KAYNAKLAR

Ardiyanto, I., and Adji, T. B. 2017. Deep residual coalesced convolutional network for efficient semantic road segmentation. IPSJ Transactions on Computer Vision and Applications, 9(1), 6.

Babaee, M., Dinh, D. T., and Rigoll, G. 2017. A deep convolutional neural network for background subtraction. arXiv preprint arXiv:1702.01731.

Badrinarayanan, V., Kendall, A., and Cipolla, R. 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12), 2481-2495.

Barnich, O., and Van Droogenbroeck, M. 2011. ViBe: A universal background subtraction algorithm for video sequences. IEEE Transactions on Image processing, 20(6), 1709-1724.

Benezeth, Y., Jodoin, P. M., Emile, B., Laurent, H., and Rosenberger, C. 2008. Review and evaluation of commonly-implemented background subtraction algorithms. In Pattern Recognition, 2008. ICPR 2008. 19th International Conference on December 2008 (pp. 1-4). IEEE.

Bianco, S., Ciocca, G., and Schettini, R. 2017. How far can you get by combining change detection algorithms?. In International Conference on Image Analysis and Processing on September 2017 (pp. 96-107). Springer, Cham.

Boughorbel, S., Jarray, F., and El-Anbari, M. 2017. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. PloS one, 12(6), e0177678.

Bouwmans, T., El Baf, F., and Vachon, B. 2008. Background modeling using mixture of gaussians for foreground detection-a survey. Recent Patents on Computer Science, 1(3), 219-237.

Braham, M., and Van Droogenbroeck, M. 2016. Deep background subtraction with scene-specific convolutional neural networks. In Systems, Signals and Image Processing (IWSSIP), 2016 International Conference on May 2016 (pp. 1-4). IEEE.

Brostow, G. J., Fauqueur, J., and Cipolla, R. 2009. Semantic object classes in video: A high-definition ground truth database. Pattern Recognition Letters, 30(2), 88-97.

Chawla, N. V., Japkowicz, N., and Kotcz, A. 2004. Special issue on learning from imbalanced data sets. ACM Sigkdd Explorations Newsletter, 6(1), 1-6.

Chen, L. C., Papandreou, G., Schroff, F., and Adam, H. 2017. Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.

Chen, L. C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. 2018a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous

convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40(4), 834-848.

Chen, L. C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. arXiv preprint arXiv:1802.02611.

Chollet, F., and Others 2015. Keras. GitHub, website: https://keras.io.

El Baf, F., Bouwmans, T., and Vachon, B. 2008. A fuzzy approach for background subtraction. In Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on October 2008 (pp. 2648-2651). IEEE.

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., and Zisserman, A. 2015. The pascal visual object classes challenge: A retrospective. International journal of computer vision, 111(1), 98-136.

Farabet, C., Couprie, C., Najman, L., and LeCun, Y. 2013. Learning hierarchical features for scene labeling. IEEE transactions on pattern analysis and machine intelligence, 35(8), 1915-1929.

Goodfellow, I., Bengio, and Y., Courville, A. 2016. Deep learning. Cambridge: MIT press.

Goyette, N., Jodoin, P. M., Porikli, F., Konrad, J., and Ishwar, P. 2012. Changedetection. net: A new change detection benchmark dataset. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on June 2012 (pp. 1-8). IEEE.

Graves, A., Mohamed, A. R., and Hinton, G. 2013, May. Speech recognition with deep recurrent neural networks. In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on (pp. 6645-6649). IEEE.

He, K., Zhang, X., Ren, S., and Sun, J. 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

He, H., and Garcia, E. A. 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering, 21(9), 1263-1284.

Heikkila, M., and Pietikainen, M. 2006. A texture-based method for modeling the background and detecting moving objects. IEEE transactions on pattern analysis and machine intelligence, 28(4), 657-662.

Hofmann, M., Tiefenbacher, P., and Rigoll, G. 2012. Background segmentation with feedback: The pixel-based adaptive segmenter. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on June 2012 (pp. 38-43). IEEE.

Huang, G., Liu, Z., Weinberger, K. Q., and van der Maaten, L. 2017. Densely connected convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition on July 2017 (Vol. 1, No. 2, p. 3).

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. 2017. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on July 2017 (pp. 1175-1183). IEEE.

Kaewtrakulpong, P., and Bowden, R. 2001. An improved adaptive background mixture model for realtime tracking with shadow detection.

Karpathy, A., and Fei-Fei, L. 2015. Deep visual-semantic alignments for generating image descriptions. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3128-3137).

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Lai, A. H., and Yung, N. H. 1998. A fast and accurate scoreboard algorithm for estimating stationary backgrounds in an image sequence. In Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on May 1998 (Vol. 4, pp. 241-244). IEEE.

Lai, S., Xu, L., Liu, K., and Zhao, J. 2015. Recurrent Convolutional Neural Networks for Text Classification. In AAAI on January 2015 (Vol. 333, pp. 2267-2273).

LeCun, Y., and Bengio, Y. 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), 1995.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lin, M., Chen, Q., and Yan, S. 2013. Network in network. arXiv preprint arXiv:1312.4400.

Lipton, A. J., Fujiyoshi, H., and Patil, R. S. 1998. Moving target classification and tracking from real-time video. In Applications of Computer Vision, 1998. WACV'98. Proceedings., Fourth IEEE Workshop on October 1998 (pp. 8-14). IEEE.

Long, J., Shelhamer, E., and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 3431-3440).

Noh, H., Hong, S., and Han, B. 2015. Learning deconvolution network for semantic segmentation. In Proceedings of the IEEE International Conference on Computer Vision (pp. 1520-1528).

Matthews, B. W. 1975. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta (BBA)-Protein Structure, 405(2), 442-451.

Oliver, N. M., Rosario, B., and Pentland, A. P. 2000. A Bayesian computer vision system for modeling human interactions. IEEE transactions on pattern analysis and machine intelligence, 22(8), 831-843.

Paszke, A., Chaurasia, A., Kim, S., and Culurciello, E. 2016. Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147.

Piccardi, M. 2004. Background subtraction techniques: a review. In Systems, man and cybernetics, 2004 IEEE international conference on October 2004 (Vol. 4, pp. 3099-3104). IEEE.

Pinheiro, P., and Collobert, R. 2014. Recurrent convolutional neural networks for scene labeling. In International conference on machine learning on January 2014 (pp. 82-90).

Rao, C. S., and Darwin, P. 2012. Frame Difference And Kalman Filter Techniques For Detection Of Moving Vehicles In Video Surveillance. Int. J. Eng. Res. Appl. IJERA, 2(6), 1168-1170.

Raza, S. H., Grundmann, M., and Essa, I. 2013. Geometric context from videos. In Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on June 2013 (pp. 3081-3088). IEEE.

Ronneberger, O., Fischer, P., and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention on October 2015 (pp. 234-241). Springer, Cham.

Sakkos, D., Liu, H., Han, J., and Shao, L. 2017. End-to-end video background subtraction with 3d convolutional neural networks. Multimedia Tools and Applications, 1-19.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Stauffer, C., and Grimson, W. E. L. 1999. Adaptive background mixture models for real-time tracking. In Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on. (Vol. 2, pp. 246-252). IEEE.

St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. 2015. Subsense: A universal change detection method with local adaptive sensitivity. IEEE Transactions on Image Processing, 24(1), 359-373.

St-Charles, P. L., Bilodeau, G. A., and Bergevin, R. 2015. A self-adjusting approach to change detection based on background word consensus. In Applications of Computer Vision (WACV), 2015 IEEE Winter Conference on January 2015 (pp. 990-997). IEEE.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D.,

Vanhouche, V., and Rabinovich, A. 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition on June 2015, pp. 1–9.

Tompson, J., Goroshin, R., Jain, A., LeCun, Y., and Bregler, C. 2015. Efficient object localization using convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 648-656).

Tuzel, O., Porikli, F., and Meer, P. 2005. A bayesian approach to background modeling. In Computer Vision and Pattern Recognition-Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference on June 2005 (pp. 58-58). IEEE.

Ulyanov, D., Vedaldi, A., and Lempitsky, V.S. 2016. Instance Normalization: The Missing Ingredient for Fast Stylization. CoRR, abs/1607.08022.

Van Droogenbroeck, M., and Paquot, O. 2012. Background subtraction: Experiments and improvements for ViBe. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2012 IEEE Computer Society Conference on June 2012 (pp. 32-37). IEEE.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. 2014. Translating videos to natural language using deep recurrent neural networks. arXiv preprint arXiv:1412.4729.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. 2015. Show and tell: A neural image caption generator. In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on June 2015 (pp. 3156-3164). IEEE.

Visin, F., Romero, A., Cho, K., Matteucci, M., Ciccone, M., Kastner, K., ... and Courville, A. 2016. Reseg: A recurrent neural network-based model for semantic segmentation. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2016 IEEE Conference on June 2016 (pp. 426-433). IEEE.

Wang, Y., Jodoin, P. M., Porikli, F., Konrad, J., Benezeth, Y., and Ishwar, P. 2014. CDnet 2014: An expanded change detection benchmark dataset. In Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on June 2014 (pp. 393-400). IEEE.

Wang, Y., Luo, Z., and Jodoin, P. M. 2017. Interactive deep learning method for segmenting moving objects. Pattern Recognition Letters, 96, 66-75.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. 2015. Show, attend and tell: Neural image caption generation with visual attention. In International Conference on Machine Learning on June 2015 (pp. 2048-2057).

Yao, J., and Odobez, J. M. 2007. Multi-layer background subtraction based on color and texture. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on June 2007 (pp. 1-8). IEEE.

Yu, F., and Koltun, V. 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Yu, F., Koltun, V., and Funkhouser, T. 2017. Dilated residual networks. In Computer Vision and Pattern Recognition on May 2017 (Vol. 1).

Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In European conference on computer vision on September 2014 (pp. 818-833). Springer, Cham.

Zhang, H. X., and Xu, D. 2006. Fusing color and gradient features for background model. In Signal Processing, 2006 8th International Conference on (Vol. 2). IEEE.

Zivkovic, Z. 2004. Improved adaptive Gaussian mixture model for background subtraction. In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on August 2004 (Vol. 2, pp. 28-31). IEEE.

Zivkovic, Z., and Van Der Heijden, F. 2006. Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters, 27(7), 773-780.

ÖZGEÇMİŞ

Adı Soyadı	:	Long Ang LİM
Doğum Yeri	:	Kampong Thom, Kamboçya
Doğum Tarihi	:	20.06.1991
Medeni Hali	:	Bekar
Yabancı Dili	:	İngilizce, Türkçe

Eğitim Durumu (Kurum ve Yıl)

Lise	: Toul Kbel Lisesi (2009)
Lisans	: Royal University of Phnom Penh Üniversitesi Bilim Fakültesi Bilgisayar Bilimi Bölüm (2013)
Yüksek Lisans	: Ankara Üniversitesi Fen Bilimleri Enstitüsü Bilgisayar Mühendisliği Anabilim Dalı (Eylül 2015 – Eylül 2018)

İş Tecrübeleri (Kurum ve Yıl)

Yazılım Mühendisi	: JC Group Co., Ltd. Şirketi (Ağustos 2012 – Kasım 2013)
Robot Yazılım Mühendisi	: SoftBank Group Corp. Şirketi (Kasım 2013 – Kasım 2014)
Yazılım Mühendisi	: JC Group CO., Ltd. Şirketi (Kasım 2014 – Eylül 2015)

(SCI) Yayınlar

Lim, Long Ang, and Hacer Yalim Keles. "Foreground Segmentation Using Convolutional Neural Networks for Multiscale Feature Encoding." Pattern Recognition Letters 112 (2018): 256-261.

Lim, Long Ang, and Hacer Yalim Keles. "Learning Multi-scale Features for Foreground Segmentation.", arXiv preprint arXiv:1808.01477 (2018). Hazır durumda, başka dergisine gönderilmesi planlanmaktadır.