UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Improving Object Detection and Segmentation by Utilizing Context

Permalink

https://escholarship.org/uc/item/5955t4nq

Author

Tripathi, Subarna

Publication Date

2018

Peer reviewed|Thesis/dissertation

eScholarship.org Powered by the California Digital Library
University of California

UNIVERSITY OF CALIFORNIA, SAN DIEGO

Improving Object Detection and Segmentation by Utilizing Context

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Electrical Engineering (Signal and Image Processing)

by

Subarna Tripathi

Committee in charge:

Professor Truong Nguyen, Chair
Professor Serge Belongie, Co-Chair
Professor Pamela Cosman
Professor Zhuowen Tu
Professor Nuno Vasconcelos

2018

Copyright
Subarna Tripathi, 2018
All rights reserved.

The dissertation of Subarna Tripathi is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Chair

University of California, San Diego

2018

DEDICATION

To my parents, my husband and my little daughter.

TABLE OF CONTENTS

Signature Page iii

Dedication iv

Table of Contents v

List of Figures vii

List of Tables xi

Acknowledgements xii

Vita xv

Abstract of the Dissertation xvii

Chapter 1 Introduction 1

Chapter 2 Segmenting Videos into Supervoxels 5
2.1 Background 5
2.2 Related Work 6
2.3 Methodology 7
2.4 Results and Discussions 8
2.5 Summary 12

Chapter 3 Object Detection in Videos by Class Label Propagation 14
3.1 Background 14
3.2 Related work 16
3.3 Video Object Proposals 18
3.4 Learning Video Object Detector Model 20
3.4.1 Training 21
3.4.2 Inference 21
3.5 OVERLAP Algorithm 22
3.6 Results and Discussions 30
3.7 Summary 36

Chapter 4 Object Detection in Videos using Recurrent Neural Network . . . 37
4.1 Background 37
4.2 Related Work 39
4.3 Methodology 40
4.3.1 Training 42
4.3.2 Inference 45

4.4 Results and Discussions . . . 46
4.5 Summary . . . 51

Chapter 5 Semantic Segmentation in Videos . . . 52
5.1 Background . . . 52
5.2 Related Work . . . 54
5.3 Methodology . . . 55
5.4 Results and Discussions . . . 58
5.5 Summary . . . 59

Chapter 6 Person Instance Segmentation using Human-Pose . . . 63
6.1 Background . . . 63
6.2 Related Work . . . 66
6.3 Methodology . . . 68
6.3.1 Pose2Instance Inference . . . 69
6.3.2 Pose2Instance Learning . . . 72
6.4 Results and Discussions . . . 74
6.4.1 Pose2Instance in a Constrained Setup . . . 74
6.4.2 Pose2Instance in Realistic Environments . . . 76
6.5 Summary . . . 79

Chapter 7 Object Detection for Embedded Systems . . . 81
7.1 Background . . . 81
7.2 Related Work . . . 83
7.3 Methodology . . . 85
7.3.1 Network Architecture . . . 85
7.3.2 Training Methodology . . . 86
7.3.3 Detection-Specific Layers . . . 87
7.3.4 Quantized Model . . . 88
7.4 Results and Discussions . . . 89
7.4.1 Dataset . . . 90
7.4.2 Detection Accuracy . . . 90
7.4.3 Complexity and Memory-BandWidth Analysis . . . 94
7.4.4 Visual Results . . . 97
7.5 Summary . . . 99

Bibliography . . . 103

LIST OF FIGURES

Figure 2.1: Number of supervoxels vs. 3D Boundary Recall for streamGBH, streamGBH with bilateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Stefan" (right). 9

Figure 2.2: Number of supervoxels vs. Explained Variation metric values for streamGBH, streamGBH with bi-lateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Soccer" (right) 10

Figure 2.3: Number of supervoxels vs. 3D segmentation accuracy for streamGBH, streamGBH with bilateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Soccer"(right) 11

Figure 2.4: Different details of objects in the layers of hierarchy. Video frame, supervoxel segmented frame at hierarchy level 1, 10 and 15 11

Figure 2.5: Qualitative time consistent performance of streamGBH+. Left to right: pairs of video frame and supervoxel segmented frame from video clip "Pirate. 12

Figure 2.6: Qualitative performance of streamGBH+. Top: input video Ice, frame 1, frame 15 and frame 26. Middle: segmentation result of state-of-the-art streamGBH at hierarchy level 5. Bottom: segmentation result of our method streamGBH+ at hierarchy level 5. 13

Figure 3.1: OVERLAP: Object class labels get propagated by streaming clustering of temporally consistent proposals. The system classifies only those VOPs which belong to a new cluster. 15

Figure 3.2: Spatio-Temporal EdgeBoxes. Clock-wise from top-left - (i) A frame from Youtube-Objects , (ii) Temporal edge $\mathbf{E}_t$ (iii) Spatial edge $\mathbf{E}_s$ (iv) $\mathbf{E}$, linear combination of $\mathbf{E}_s$ and $\mathbf{E}_t$. Green and Red circles highlight strong responses and missing responses respectively. . . . 19

Figure 3.3: Synthetic experiment using perturbed ground truth boxes in PASCAL VOC showing affinity-based clustering of candidate windows. The clustering technique works efficiently with some exceptions where overlapping object instances share very similar color. 24

Figure 3.4: Segmentation masks on two different images from clustered object proposals. 25

Figure 3.5: Segmentation masks generation is not successful where proposal boxes do not tightly enclose the actual object. This happens in cluttered background cases. 26

Figure 3.6: Two segmentation masks from clustered video object proposals from Youtube video "Bird and Cat" for frames #5, #20 and #45. 27

Figure 3.7: Sample results of Video Object Detection with VOP. First 6 rows show successful detection cases and the last row shows false detection cases. 34

Figure 3.8: Temporally consistent VOPs on "Horse riding", "Bird-cat" and "Alaskan bear". Windows drawn in same color belong to same cluster. Top row denotes clustered proposals on sample frames #2, #10, #25 ; bottom row shows the results of streaming clustering. 35

Figure 4.1: Results from the final eight frames of five different test-set sequences. Top and bottom row of each pair show the *pseudo-labeler* and RNN respectively. The RNN predicts correct categories, multiple instances, detects missing objects by increasing confidence score. . 49

Figure 4.2: Failure cases for the proposed model. Left: the RNN cannot recover from incorrect *pseudo-labels*. Right: RNN localization performs worse than *pseudo-labels* possibly owing to multiple instances of the same object category. 50

Figure 5.1: Video Semantic Segmentation involves semantic segmentation of multiple frames together. 53

Figure 5.2: Framework for intra and inter-frame dense connectivity. Every pixel in every frame is connected to each other. 56

Figure 5.3: Qualitative results on Camvid. Input frames, TextonBoost classifier scores; Dense-CRF [KK13]; Video-Level Dense-CRF [ASB14]; Dense-CRF with P^n-Potts [VWT12]; proposed video-level Dense-CRF with P^n-Potts; graph-cut inference; and Ground truth. 61

Figure 6.1: Pose2Instance segmentation model incorporates a *learnable* component conditioned on the human "pose". The model generates keypoint heatmaps and segmentations at instance level and uses the keypoints heatmaps output as an additional input to the segmentation. 64

Figure 6.2: Instance segmentation with oracle skeleton. Top: input image; Sobel edges; RAG with edge strength from Sobel responses. Bottom: DeepLab *person* segmentation; Oracle skeleton, distance transform on RAG, *pose-maps* and instance segmentation output. 69

Figure 6.3: Pose2Instance inference with oracle skeleton. Top row: An image from Inria dataset containing 9 persons; Instance classification; Bottom row: DeepLab-people score, and the *Pose2Instance* inference output generated by fusing DeepLab-people and *pose-instance map*. 71

Figure 6.4: Architectures for joint pose - segmentation learning. Left: Multitask model where pose estimation and segmentation are two parallel output paths. Right: Cascaded Model where pose dedicated parameters are *learned* for mapping pose to segmentation. 73

Figure 6.5: From left to right: Ground truth instance segmentations; Corresponding image from COCO keypoints dataset; Pose2Instance inference with oracle keypoints. Colored boxes show errors in segmentation ground truth that are corrected using our keypoint conditioned model. 75

Figure 6.6: Pose2Instance with oracle bounding boxes vs oracle keypoints. From left to right: A frame from COCO keypoints dataset; Ground truth instance segmentations; Baseline instance segmentation from oracle bounding boxes and Pose2Instance inference from oracle keypoints. 75
Figure 6.7: Pose2Instance in a constrained setup. Top: DeepLab person segmentation. Bottom:Pose2Instance inference from oracle keypoints on COCO evaluation dataset. (best viewed in color) 76
Figure 6.8: Visualization of shape likelihood from pose estimation. Instances from COCO validation set; intermediate latent shape likelihood for (i) *Pose Only model*; (ii) *Multitask model* and (iii) *Pose2Instance model* respectively. joint models learn to capture the overall contour. 77
Figure 6.9: Pose2Instance without oracle keypoints. Top row: Instance bounding boxes of COCO validation images. Middle row: Ground truth segmentation. Bottom row: predicted segmentation masks for the instance bounding boxes. 78

Figure 7.1: YOLO vs proposed network. Output channel's value 16 corresponds to 1 class-conditional and $1+4$ confidence and coordinates for each of the $B = 3$ boxes. 85
Figure 7.2: Performance of LCDet on FDDB for LeakyReLU vs ReLU model with discrete core metric. 91
Figure 7.3: Performance comparison between floating and fixed point models (ReLU) at different IoU thresholds on FDDB. 92
Figure 7.4: Performance with varying IOU criteria. The fixed point model achieves 5% improvement in true positives when IOU going down from 50% to 40%. Quantization loss for box coordinate regression is high. Floating point model is relatively resilient to IOU. 93
Figure 7.5: Precision-Recall performance on Widerface Validation set with relaxed IoU criteria. 94
Figure 7.6: Frame Rate Improvement for Fixed Point Model. Fixed-Point LCDet also outperforms fixed-point SSD300. 96
Figure 7.7: Layer-wise Bandwidth Requirement for LCDet-8bit-fixed Point Implementation 97
Figure 7.8: Effect of available Dynamic Random Access (DDR) memory Bandwidth (BW) on Frame Rate 98
Figure 7.9: Face detection results for faces with different scales on some images from FDDB. 98
Figure 7.10: Face detection results on black and white images from FDDB dataset. 99
Figure 7.11: Detection results for multiple faces of frontal and side profiles on sample images from FDDB. 99
Figure 7.12: Successful face detection results of LCDet on challenging Widerface Validation images containing difficult examples with pose variation, illumination changes, photograph styles, and different scales. 100

Figure 7.13: Localization challenges of LCDet on Widerface images. Faces marked as the *yellow* regions are considered false positives as per 50% IoU criteria, but true positive for more relaxed IoU criteria. The regions marked in *red* show missed detections. 101

LIST OF TABLES

Table 2.1: Average comparative metric values for all videos in Chen database . 10

Table 3.1: Object Detection Results on Youtube-Objects test set. Pre-trained R-CNN detector is downloaded from [GDDM]. DPM Detection results are from [KFS15]. SS, EB and VOP correspond to Selective Search, EdgeBoxes and proposed VOP ($\lambda = 0.2$). 31

Table 3.2: Complexity and accuracy. Proposal generation (Prop time), per-frame (PF) baseline methods with Selective Search, EdgeBoxes, proposed VOPs and OVERLAP. OF denotes optical flow. mAP increases as #VOPs increases. $3\times$ speedup with 500 VOPs and 4% drop in mAP. 31

Table 4.1: Object detection with DPM, Video Object Proposal method, YOLO, domain-adapted YOLO. RNN-IOS regularizes on input-output similarity, to which RNN-WS adds category-level weak-supervision, to which RNN-PS adds a regularizer encouraging prediction smoothness. 47

Table 4.2: Overall detection results on Youtube-Objects dataset. Our best model (RNN-PS) provides 7% improvements over DA-YOLO baseline. . . 48

Table 5.1: Per-class segmentation accuracy on CamVid dataset for different methods. 59

Table 6.1: Oracle keypoints provides 10% to 12% relative improvement over oracle bounding box at various IOU thresholds atop DeepLab-people segmentation model. Results of FAIRCNN[ZLL$^+$16] and CUHK[QSL$^+$15] that also use VGG as the base network are from COCO Leaderboard. 76

Table 6.2: Segmentation accuracy on COCO validation instances. Pose2Instance model achieves higher accuracy over the *multitask* model that outperforms *segmentation only* model. Overall, relative improvement from *segmentation only* model is 3.8% to 10.5% at various IOU thresholds. 77

Table 7.1: Comparison between RPN, ConvDet and YLDet. RP stands for Region Proposals, *cls* denotes classification. 87

Table 7.2: Inference speed of LCDet vs others on NVidia Tesla K40. YOLO* is our TF-based face detector. Max TP is the highest true positive rate value achieved in FDDB. SSD and Faster R-CNN running time use TF-SSD [B.] and TF Faster RCNN [Cha] respectively. 95

Table 7.3: Performance analysis of Fixed-point LCDet in terms of OPs, Memory Activation Footprints 95

ACKNOWLEDGEMENTS

I came to graduate school from industry with hopes and apprehensions. Prof. Nguyen gave me the opportunity to volunteer in his lab even before I formally started my doctoral studies. He gradually but effectively steered me in the right direction and helped me in shaping my vision. He has given me the freedom to pursue my own goals as a researcher. He has always been very supportive and encouraging. His help on facing daily challenges as a researcher has been invaluable. Prof. Nguyen's guidance in the early phase of my studies kept me going. Specially his support during the last couple of months has truly been a blessing.

Prof. Serge Belongie has greatly influenced how I approach a research problem. I was fortunate that I started working with him in an academia-industry collaboration project. He has shared his wisdom and philosophical insights into computer vision and its practical real world issues. Throughout my doctoral studies Prof. Belongie has kept my passion for research alive. I could not ask more from my co-advisor. It's a magic how he made sure our interactions, mostly over video chats, became absolute delight and productive each and every time. Prof. Belongie and his research group in Cornell Tech have always been wonderful to interact with, and provided food for thought in most amazing ways possible.

I would also like to thank others who helped guide my research. All my interactions with Prof. Zhuowen Tu helped me be motivated towards meaningful research. His own research and all our interactions have always been inspiring. I thank Prof. Nuno Vasconcelos for our insightful discussions. I would like to thank Prof. Pamela Cosman for being supportive through out, and teaching how to present myself and my work.

Graduate school would have been much lonelier without other students whom I shared unlimited days and nights with. We shared unlimited discussions on technical and non-technical topics. Yeejin Lee played an important role as a co-worker and a friend

when we spent sleepless nights in the lab doing course projects or hacking the code. It has been a smooth sailing experience while collaborating with Byeongkeun Kang on multiple projects, and subsequent papers. Working and collaborating with Zachary Chase Lipton has been exciting and fun. We spent hours together from brainstorming ideas to implementation, from focusing into details to summarized presentation. It has been a great learning exercise for me. Igor Federov helped in revising my manuscripts even during his vacation. Many others have made the last couple of years enjoyable. The list includes Yuanyuan Wu, Shibin Parameswaran, Yonatan Vaizman, Srinjoy Das, Enming Luo among others. My heartfelt thanks to all.

I thank Dr. Youngbae Hwang from KETI for the collaboration project we worked together for over three years. Some of these projects significantly helped shape my research interests. I am also grateful to my internship hosts and mentors in Qualcomm Inc., Microsoft Research and Google Research for the exposure and outlook I was given during all three of my research internships. I express my gratitude to Dr. Gokce Dane and Dr. Brian Guenter specially for their honest discussions on important technical and non-technical issues.

One person that deserves my special thanks is Mrinal Paul. His love, support and encouragement make everything I do a little more worthwhile. He kept our family functioning by his patience and cooking skills during most crucial times, even though he has pretty demanding work schedule. Not only he brought food for me in the lab before the deadlines, he brought enough food for my project mates as well. His patience as a friend and husband has only increased in the last decade.

Finally, my parents' attention, love and guidance have made me the person I am today. From instilling inquisitiveness to being open to new ideas and ideologies, they made sure I can face challenges. I can only hope to make them proud.

Parts of this dissertation are based on papers I have co-authored with others.

Chapter 2 is in part based on the paper "Improving streaming video segmentation with early and mid-level visual processing", by S. Tripathi, Y.B. Hwang, S. Belongie and T. Nguyen [THBN14b]. The dissertation author was the primary investigator and author of this paper.

Chapter 3 is in part based on the paper "Detecting Temporally Consistent Objects in Videos through Object Class Label Propagation", by S. Tripathi, S. Belongie, Y.B. Hwang, and T. Nguyen [TBHN16]. The dissertation author was the primary investigator and author of this paper.

Chapter 4 is in part based on the paper "Context Matters : Refining Object Detection in Video with Recurrent Neural Networks", by S. Tripathi, Z.C. Lipton, S. Belongie, and T. Nguyen [TLBN16]. The dissertation author was the primary investigator and author of this paper.

Chapter 5 is in part based on the paper "Semantic Video Segmentation: Exploring Inference Efficiency", by S. Tripathi, Y.B. Hwang, S. Belongie, and T. Nguyen [THBN15]. The dissertation author was the primary investigator and author of this paper.

Chapter 6 is in part based on the paper "Pose2Instance: Harnessing Keypoints for Person Instance Segmentation", by S. Tripathi, M. Collins, M. Brown, and S. Belongie [TCBB17]. The dissertation author was the primary investigator and author of this paper.

Chapter 7 is in part based on the papers "LCDet: Low-complexity Fully-Convolutional Neural Networks for Object Detection in Embedded Systems", by S. Tripathi, G. Dane, B. Kang, V. Bhaskaran, and T. Nguyen [TDK+17] and and "Low-Complexity Object Detection with Deep Convolutional Neural Network for Embedded Systems", by S. Tripathi, B. Kang, G. Dane, and T. Nguyen [TKDN17]. The dissertation author was the primary investigator and author of these papers.

VITA

2005	B.Tech in Computer Science and Engineering, West Bengal University of Technology, India
2011	M.S.(Research) in Electrical Engineering, Indian Institute of Technology, Delhi, India
2013-2018	Graduate Research Assistant, University of California, San Diego
2018	Ph.D. in Electrical Engineering (Signal and Image Processing), University of California, San Diego

PUBLICATIONS

S. Tripathi, G. Dane, B. Kang, V. Bhaskaran, and T. Nguyen, "LCDet: Low-complexity fully convolutional Neural Networks for Object Detection in Embedded Systems", *CVPRW* 2017.

S. Tripathi, B. Kang, G. Dane, and T. Nguyen, "Low-Complexity Object Detection with Deep Convolutional Neural Network for Embedded System", *SPIE* 2017.

S. Tripathi, M. Collins, M. Brown, and S. Belongie, "Pose2Instance: Harnessing Keypoints for Person Instance Segmentation", *arXiv preprint* arXiv:1704.01152, 2017.

S. Tripathi, Z. C. Lipton, S. Belongie, and T. Nguyen, "Context Matters: Refining Object Detection in Video with Recurrent Neural Networks", *BMVC* 2016.

S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen, "Detecting Temporally Consistent Objects in Videos through Object Class Label Propagation", *WACV* 2016.

S. Tripathi, S. Belongie, Y. Hwang, and T. Nguyen, "Semantic Video Segmentation: Exploring Inference Efficiency", *ISOCC* 2015.

S. Tripathi, Y. Hwang, S. Belongie, and T. Nguyen, "Improving streaming video segmentation with early and mid-level visual processing", *WACV* 2014.

S. Tripathi, and B. Guenter, "A Statistical Approach to Continuous Self-Calibrating Eye Gaze Tracking for Head-Mounted Virtual Reality Systems", *WACV* 2017 **(Best Paper Award)**.

S. Tripathi, B. Kang, and T. Nguyen, "Harnessing Basis Functions of Image Transformation in Generative Adversarial Networks", currently under review.

S. Tripathi, Z.C. Lipton, and T.Q. Nguyen "Correction by Projection: Denoising Images with Generative Adversarial Networks", currently under review.

Z. C Lipton and **S. Tripathi**, "Precise Recovery of Latent Vectors from Generative Adversarial Network", *ICLR 2017 workshop track*.

B. Kang, **S. Tripathi**, and T. Nguyen, "Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth map", *ACPR* 2015.

K. Chen, **S. Tripathi**, Y. Hwang, and T. Nguyen, "Moving objects detection using classifying object proposals for driver assistance system", *ISOCC*, 2016.

S. Tripathi, Y. Hwang, Sung-Joon Jang, S. Belongie, and T. Nguyen, "Hardware Feasibility Analysis for Motion Segmentation Initialization", *ISOCC*, 2014

ABSTRACT OF THE DISSERTATION

Improving Object Detection and Segmentation by Utilizing Context

by

Subarna Tripathi

Doctor of Philosophy in Electrical Engineering (Signal and Image Processing)

University of California, San Diego, 2018

Professor Truong Nguyen, Chair
Professor Serge Belongie, Co-Chair

Object detection and segmentation are important computer vision problems that have applications in several domains such as autonomous driving, virtual and augmented reality systems, human-computer interaction etc. In this dissertation, we study how to improve object detection and segmentation by utilizing different contexts. Context refers to one of many application scenarios such as (i) video frames for consistent prediction over time, (ii) specific domain knowledge such as human keypoints for person segmentation, and (iii) implementation context aiming for efficiency in embedded systems.

Temporal Context of Videos: Video data understanding has drawn considerable

interest in recent times as a result of access to huge amount of video data and success in image-based models for visual tasks. However, motion blur, compression artifacts cause apparently consistent video signals to produce high temporal variation on frame-level output for vision tasks such as object detection or semantic segmentation. We study and propose efficient early, and high-level visual processing algorithms by leveraging video content in a streaming fashion. We show how to fuse motion and color to achieve improved streaming hierarchical supervoxels. As a high-level visual task, we propose consistent and efficient video object detection using Convolutional Neural Network (CNN) by clustering video object proposals and propagating object class labels through the videos. Next, we propose an end-to-end framework for learning video object detection through Recurrent Neural Network (RNN) by posing video as a time series. We also present a post-processing framework for improving semantic segmentation in videos.

Domain Knowledge Context for Segmentation: Person instance segmentation is a promising research frontier for a range of applications such as human-robot interaction, sports performance analysis, and action recognition. Human keypoints are a well-studied representation of people. We explore how to use keypoint models to improve instance-level person segmentation in constrained and unconstrained environments with or without training.

Efficiency Context for Embedded Implementation: To make an object detector system amenable for embedded implementation, we propose a low-complexity fully convolutional neural network. Additionally, we employ 8-bit quantization on the learned weights. As a mobile use case, we choose face detection. The results show that the proposed method achieves competitive accuracy comparing with the state-of-the-art CNN-based object detection methods while reducing the model size by $3\times$ and memory-BW by $3-4\times$ comparing with its strongest baselines.

Chapter 1

Introduction

Object detection and segmentation are most popular perspectives of object recognition in computer vision. Object detection addresses the problem of localizing each object instance with a bounding box in addition to identifying their corresponding category. Semantic segmentation aims for assigning category label to each pixel, thus providing pixel-precise localization but fails to resolve individual instances of a category. Moreover, instance segmentation problem deals with the pixel-wise delineation of multiple objects, combining segment-level localization and per-pixel object category classification. In this dissertation, we show how to improve object detection and segmentation by analyzing context. Context refers to three different application scenarios such as (i) temporal context in videos, (ii) domain knowledge context for a particular task like using human keypoints for person instance segmentation, and (iii) implementation context like efficiency for embedded systems.

With the availability and access to numerous videos through Youtube, Facebook and other sharing resources, there is a growing need of processing and understanding video. Significant interest is emerging for using video as an alternative rich source of data. Often, a single video shows multiple views of an object, contains multiple deformations.

Motion in video also has a potential for automatically segmenting objects from the background. However, motion blur, compression artifacts cause apparently consistent video signals to produce high temporal variation on frame-level output for vision tasks. To mitigate the high temporal output variability of frame-level processing, researchers intend to learn visual representations using video as the supervisory signal instead of single images. Enforcing the constraint of similar representation within neighboring frames, learning latent representations for successive frames are some of the research avenues currently being explored in the computer vision community. Processing of arbitrarily long duration of video which could otherwise be ideal for better model learning, becomes prohibitive in terms of memory and computational complexity. Thus, early, mid-level or high-level visual processing on video can most effectively be performed in a streaming framework. As one of the early-visual processing techniques, we investigate on improving streaming hierarchical supervoxel segmentations in Chapter 2. In order to impose spatio-temporal consistencies in several vision tasks, supervoxels play an important role. For high-level vision tasks, we explore on harnessing video content for object detection and semantic segmentation in Chapter 3 to Chapter 5. In Chapter 3, we propose to generate better proposals for moving as well as static objects in a video. Additionally we show how to detect objects through class label propagation within in a streaming framework. This framework reduces the complexity of convolutional neural network (CNN) based object detection baseline by more than $100\times$. In Chapter 4, we describe a new method for refining a video-based object detection by modeling a video as a time series. We utilize the recurrent neural network (RNN) framework, and propose an effective training strategy for the RNN utilizing a combination of strong supervision, weak supervision and regularization encouraging prediction smoothness. In Chapter 5, we demonstrate how to improve semantic segmentation in video by an efficient inference method in a co-segmentation framework.

Instance segmentation, in particular *person* instance segmentation is a promising research frontier for a range of applications such as human-robot interaction, sports performance analysis, and action recognition. Deep convolutional neural networks are the current state-of-the-art methods for the task of instance level segmentation. Although these methods work extremely well for *any* category of objects, there is a potential for human-specific domain knowledge to boost the person segmentation performance. Proximity of person instances each exhibiting extremely dynamic poses makes the person instance segmentation more challenging than *generic* object instance segmentation. Person instance segmentation is especially challenging because in real world, people tend to appear in groups. They exhibit highly complex interaction with other people and objects. In Chapter 6, first we develop a thorough understanding of whether person-specific domain knowledge is useful for person instance segmentation. Next, we quantify the importance of human keypoints as a useful domain knowledge for improving segmentation over the baseline of best performing deep learning models trained only for segmentation in absence of person domain knowledge. With the availability of image datasets that include both segmentation masks and keypoints annotations, we consider a methodical approach to quantify the importance of keypoints for people instance segmentation. We explore what happens if an oracle provides all the keypoints, or only bounding boxes, and how people instance segmentation can be improved respectively.

Current state-of-the-art CNN-based methods for object detection run on powerful GPUs that dissipate huge amount of power. However, embedded processors and Digital Signal Processors (DSPs) consume only a small amount of power by virtue of fixed-point operations. From implementation efficiency context, a CNN-based object detection method should ideally run on embedded processors, especially in portable devices such as mobiles and vehicles, with reduced power requirement, limited memory and fixed point operations without compromising the detection accuracy. In Chapter 7, we propose

and analyze a low-complexity fully-convolutional neural network based object detector to satisfy the above requirements in embedded systems. Additionally, we investigate the effects of $8-$bit quantization on the proposed trained model. We show that the $8-bit$ fixed point model leads to additional $4\times$ memory reduction with almost no loss in detection accuracy that also runs $20\times$ faster.

Chapter 2

Segmenting Videos into Supervoxels

2.1 Background

There are three different paradigms in video segmentation. First is frame processing in which each frame is independently segmented, but no temporal information is used. This method is fast but the temporal coherence is poor. Second is 3D volume processing that represents a model for the whole video. It is bi-directional and multipass processing. The results are best, but both the complexity and the memory requirement are too high to process long videos and does not cater to the need for streaming videos. Stream processing processes the current frame only based on a few previously processed frames. It is forward-only online processing, and the results are good and efficient in terms of time and space complexity. The state-of-the-art streaming segmentation, *streamGBH* [XXC12a],, outperforms other streaming methods and competitive with full-video hierarchical methods. In this framework, the streaming video is conceptualized as a set of non-overlapping subsequences $v = \{v_1, v_2, , v_m\}$ with k_i frames for subsequence v_i. The hierarchical segmentation result, s, is approximately decomposed into $s = \{s_1, s_2, , s_m\}$, where s_i is hierarchical segmentation of subsequence v_i. StreamGBH

adopts the graph-based grouping method into the streaming hierarchical segmentation framework. It constructs a graph which is similar to a graph over the spatial-temporal video volume with a 26-neighborhood in 3D space-time. However, this graph is only constructed for the current two subsequences in process, v_i and v_{i-1}. This graph is the first layer of the hierarchy and its edge weights are direct color dissimilarity of voxels. The color dissimilarity is measured by χ^2 distance of the normalized color histogram. Given the hierarchical segmentation result s_{i-1} of v_{i-1}, the layer by layer hierarchical segmentations for v_i are inferred. Being a streaming segmentation method, StreamGBH thus became a powerful early vision tool.

We propose a significant increment and improvement to the state-of-the-art streamGBH in terms of improving segmentation quality by using dense optical flow. We use optical flow as feature and at the same time as a guiding tool for the temporal connection in the initial graph and call our approach *StreamGBH+*. We perform thorough experimental analysis on a benchmark database [CC10] used in the evaluation of libsvx library [XC12] for StreamGBH.

2.2 Related Work

Video analysis literature has two predominant research directions in feature-based [Lap05] methods and segmentation-based methods [MD02]. Several driving applications such as object recognition [RW08], augmented reality [BCL15] or animation [BS09] require dense segmentation, and feature based method is not a natural fit in such cases. Thus video segmentation acquired a rich history in the last decade. Despite of several approaches proposed in the literature of video segmentation, more recently the idea of associating initial over-segmentations into supervoxels, followed by processes such as hierarchical grouping [GKHE10], long range tracking [LASL11], superpixel flow

[VRAPM10] have become popular. As an early vision tool, color-based supervoxel segmentation method *streamGBH* [XXC12a] reportedly has the best performance in terms of quality and complexity. We propose StreamGBH+, an improvement to StreamGBH, and demonstrate that the new approach outperforms the StreamGBH.

2.3 Methodology

As a pre-processing step, we apply bilateral filtering [PD09] as an edge-preserving smoothing to improve segmentation. Along with color similarity, we consider motion similarity of voxels and influence of motion direction on graph connectivity. Use of dense optical flow [Liu09] considerably improves segmentation results. First, instead of connecting a voxel (i, j, t) to its immediate 9 neighbors $(i+m, j+n, t-1), m, n \in \{-1, 0, +1\}$ in the previous frame, we connect it to its 9-neighbors $(i+u(i, j)+m, j+v(i, j)+n, t-1)$ along the backward flow vector (u, v), similar to the approach proposed in [GKHE10]. This is a generalization of prior grid-based volumetric approaches which can only be achieved using a graph representation. Next, we use optical flow as a feature for each region during hierarchical segmentation. As optical flow is only consistent within a frame, we use a per-frame discretized flow histogram. Unlike [GKHE10] which discusses SIFT-like (with respect to angle) motion feature representation, we propose a simple representation of two histograms - one for the horizontal component and another for the vertical component of optical flow field. The benefit of this simpler approach is to distinguish motions with same direction but different magnitude. Matching the flow-descriptors of two regions then involves averaging the χ^2 distance of their normalized per-frame flow-histograms over time. Finally, we combine the χ^2 distance of the normalized color histograms $d_c \in [0, 1]$ with the χ^2 distance of the normalized flow histograms $d_f \in [0, 1]$ by $(d_c, d_f) \rightarrow (1-(1-d_c)(1-d_f))^2$. This function is close to

zero if both distances are close to zero, and close to one if any one of them is close to one. We also tried exponential form of distance function which produces similar result. Throughout this paper we term our approach of supervoxel segmentation as streamGBH+.

2.4 Results and Discussions

Here, we demonstrate the results of applying streamGBH+ onto several videos.

Data: We use the published benchmark dataset [CC10] and video segmentation performance metrics [XC12] for the quantitative evaluations. This video dataset is a subset of the well-known *xiph.org* videos that have been supplemented with a 24-class-semantic-pixel labeling set. The classes are same as the classes in the MSRC object-segmentation dataset [CC10]. In the implementation, we use sub-sequence length of only 3 in all experiments and thus performance is not expected to be near to full-video processing.

The 8 videos *(Bus, Container, Garden, Ice, Paris, Salesman, Soccer , and Stefan)* in this set are densely labeled with semantic pixels and have duration of 85 frames each. This dataset has been used for evaluation for the state-of-the-art StreamGBH method. This dataset allowed us to evaluate these segmentation methods against human perception.

3D Boundary Recall: The 3D boundary is the shape boundary of a three-dimensional object, composed by surfaces. It measures the detection of spatio-temporal boundary. For each segment in the ground-truth and segmentations, we extract the within-frame and between-frame boundaries and measure recall using the standard formula [XC12]. Figure 2.1 shows the dependency of 3D Boundary Recall on the number of segments. StreamGBH+ performs better.

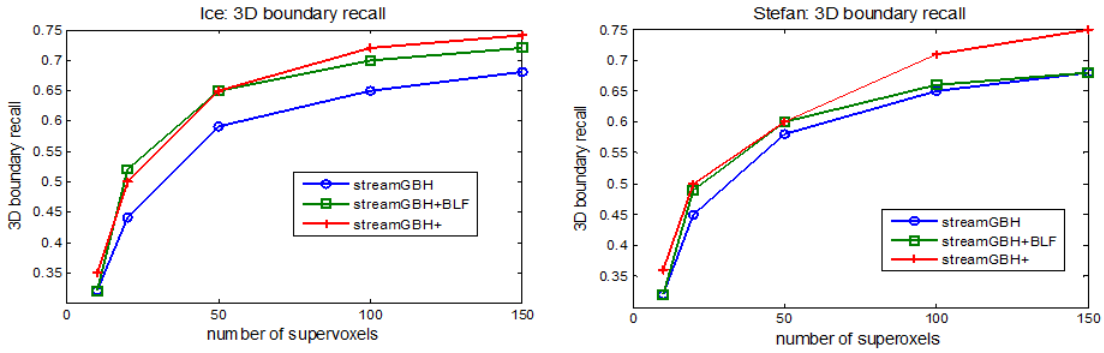


Figure 2.1: Number of supervoxels vs. 3D Boundary Recall for streamGBH, streamGBH with bilateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Stefan" (right).

Explained Variation: Explained Variation metric is proposed in [MPW+08] as a human-independent metric. It considers the supervoxel as a compression method of a video. The metric is defined as: $R^2 = \frac{\Sigma_i(\mu_i - \mu)^2}{\Sigma_i(x_i - \mu)^2}$. It is evaluated by summing over i voxels where x_i is the actual voxel value, μ is the global voxel mean and μ_i is the mean value of the voxels assigned to the supervoxel that contains x_i [XC12]. Figure 2.2 shows the dependency of explained variation metric on the number of supervoxels. StreamGBH+ again performs better than streamGBH.

3D Segmentation accuracy: This metric measures what fraction of a ground-truth segment is correctly classified by the supervoxels; each supervoxel should overlap with only one object/segment as a desired property [XC12] of video segmentation. To evaluate the overall segmentation quality, we also take the average of the fraction across all ground-truth segments in the video. Figure 2.3 shows the dependency of 3D segmentation accuracy on the number of supervoxels. StreamGBH+ performs better than streamGBH.

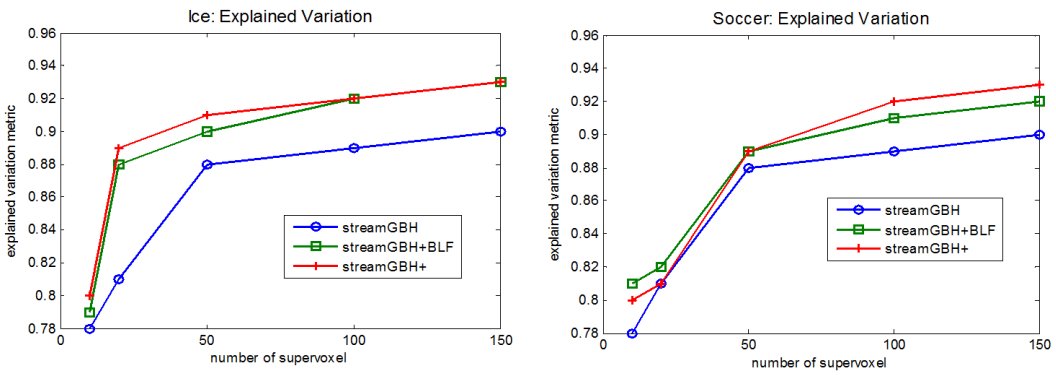


Figure 2.2: Number of supervoxels vs. Explained Variation metric values for streamGBH, streamGBH with bi-lateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Soccer" (right)

Table 2.1: Average comparative metric values for all videos in Chen database

Metric	StreamGBH	StreamGBH with Motion	streamGBH with BLF	streamGBH+
Boundary recall 2D	0.44	0.442	0.451	**0.452**
Boundary recall 3D	0.47	0.482	0.49	**0.49**
Explained variation	0.71	0.72	0.73	**0.75**
Accuracy 2D	0.58	0.58	0.58	**0.58**
AccuracyD	0.56	0.54	0.55	0.55
Undersegmentation 2D	8	9	9	10
Undersegmentation 3D	18	18	15	18

3D Under-segmentation Error: 3D under-segmentation error measures what fraction of voxels goes beyond the volume boundary of the ground-truth segment when mapping the segmentation onto it. The details for this metric can be found in the benchmark paper [XC12].

Proposed streamGBH+ performs better than the state-of-the-art streamGBH for videos with object motions. Overall, streamGBH+ outperforms state-of-the-art for metrics such as boundary recall 2D, boundary recall 3D, explained variation. As per Table 2.1 for the metrics such as *accuracy 2D* and *accuracy 3D* we see no difference for this dataset overall. Though, the performance of *StreamGBH+* for the under segmentation metrics is not better than the state-of-the-art, we care for other metrics more at this time

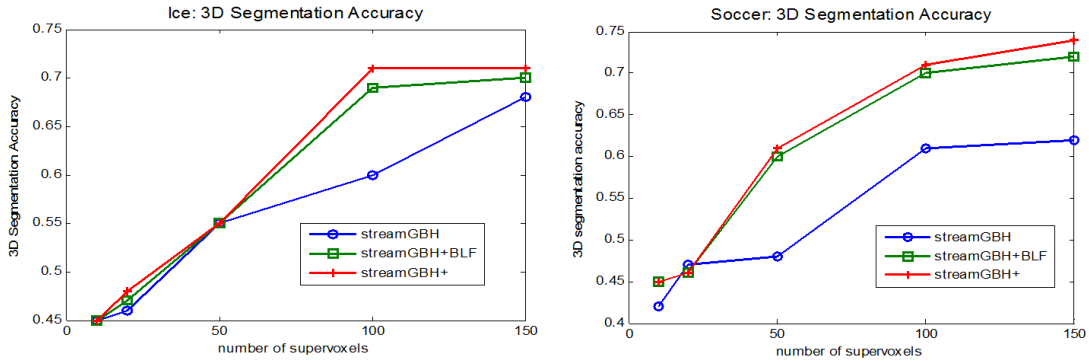


Figure 2.3: Number of supervoxels vs. 3D segmentation accuracy for streamGBH, streamGBH with bilateral filtering and streamGBH+. Evaluation results for "Ice" (left) and "Soccer"(right)

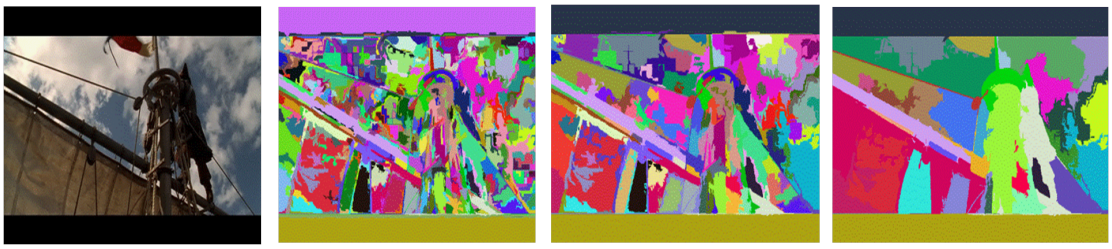


Figure 2.4: Different details of objects in the layers of hierarchy. Video frame, supervoxel segmented frame at hierarchy level 1, 10 and 15

because of our future target application on object recognition. We use the optical flow method [Liu09] based on constant memory; coarse to fine warping techniques which uses fixed point iterations. Thus the overhead of memory and computational complexity of streamGBH+ is not significant comparing with *StreamGBH*.

Qualitative Performance: Here we show some qualitative results on long videos, which necessitate a streaming method. We see different details of an object in the layers of hierarchy. For example, in Figure 2.4 one can see more than three parts in the object "Pirate" in 5-th layer, and locate a single human body pose in 15-th layer. Figure 2.5 shows a long term temporal coherence of streamGBH+.

Figure 2.6 corroborates that streamGBH+ is able to avoid unnecessary over-

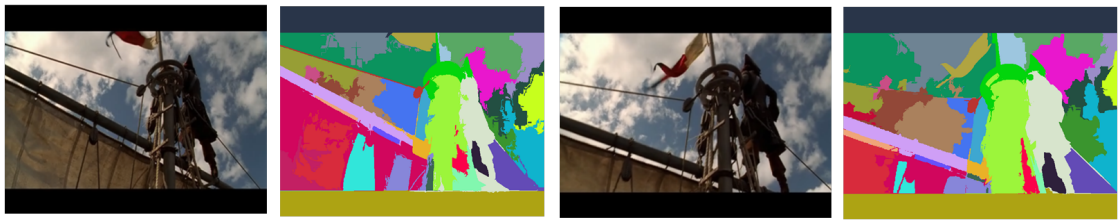


Figure 2.5: Qualitative time consistent performance of streamGBH+. Left to right: pairs of video frame and supervoxel segmented frame from video clip "Pirate.

segmentation compared to state-of-the-art streamGBH.

2.5 Summary

We have presented an improvement to the state-of-the-art color-based supervoxel segmentation methodology by incorporating motion information. We have evaluated the performance of our approach vis-à-vis the performance of *streamGBH* on different videos and have shown that our approach, *StreamGBH+*, produces improved supervoxels results comparing with the state-of-the-art.

Acknowledgments

Chapter 2 is in part based on the paper "Improving streaming video segmentation with early and mid-level visual processing", by S. Tripathi, Y.B. Hwang, S. Belongie and T. Nguyen [THBN14b]. The dissertation author was the primary investigator and author of this paper.

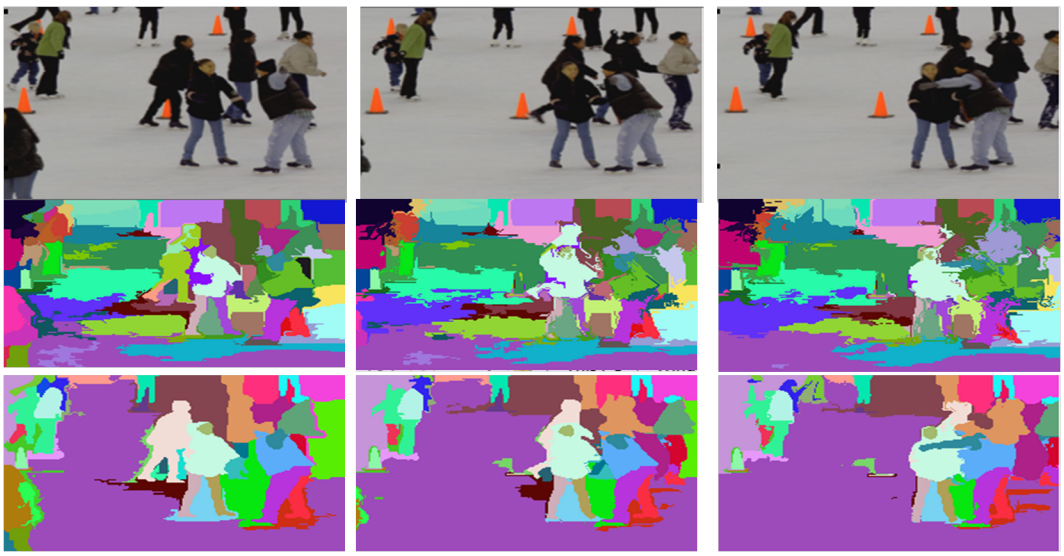


Figure 2.6: Qualitative performance of streamGBH+. Top: input video Ice, frame 1, frame 15 and frame 26. Middle: segmentation result of state-of-the-art streamGBH at hierarchy level 5. Bottom: segmentation result of our method streamGBH+ at hierarchy level 5.

Chapter 3

Object Detection in Videos by Class Label Propagation

3.1 Background

The paradigm of object detection in images has been shifted from sliding window based methods to object proposals in recent years. Object proposals dramatically decrease the number of detection hypotheses to be assessed. Thus, use of CNN-features [GDDM14], which is more effective but computationally expensive, have turned out to be feasible for accurate detection.

For the detection of video objects, proposals not only need to consider the space-time complexity, but also need to address the temporal consistency. We propose generating Video Object Proposals (VOP) by scoring candidate windows based on spatio-temporal edge content and show that these VOPs help in learning better video object detectors. Further, we propose an efficient online clustering of these proposals in order to process arbitrarily long videos. We show that the joint-analysis of all such windows provides a way towards multiple object segmentations and helps in reducing test time

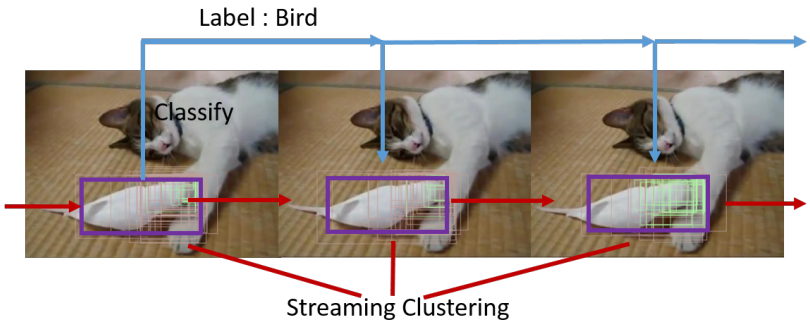


Figure 3.1: OVERLAP: Object class labels get propagated by streaming clustering of temporally consistent proposals. The system classifies only those VOPs which belong to a new cluster.

object detection complexity.

We divide a video into sub-sequences with one-frame overlap in a streaming fashion [XXC12b, THBN14a]. We analyze all candidate windows jointly within a sub-sequence followed by affinity-based clustering to produce temporally consistent clusters of object proposals at every video frame. The advantage of performing a streaming spatio-temporal clustering on the object proposals is that it enables an easy label propagation through the video in an online framework. Presumably all object proposals of a cluster have the same object class type. We propose deep-learning based video object detection through objects' class label propagation using online clustering of VOPs. As opposed to applying R-CNN [GDDM14]-like approaches, which essentially classify every window through evaluating expensive CNN features, at every video frame, the proposed method of class label propagation requires detection/classification only on video frames which has new clusters (Fig. 3.1).

Our contributions can be summarized as follows. First, we present a simple yet effective Video Object Proposal (VOP) method for detecting moving and static video objects by quantifying the spatio-temporal edge contents, and demonstrate VOP's efficacy in learning a better CNN-based video object detector model. Next, we present a

novel algorithm OVERLAP, "Objects in Video Enabler thRough LAbel Propagation". Through streaming clustering of VOPs, this framework enables efficient and temporally consistent object detection in videos by performing objects' class label propagation. Additionally, we show how object segmentation can be achieved as a by-product of this OVERLAP framework.

3.2 Related work

There are several approaches towards video object detection which broadly fall into three categories : (1) image object proposals for each frame (2) motion segmentation in video (3) supervoxel aggregation.

Object Proposals and detection: Unsupervised category-independent detection proposals are evidently shown to be effective for object detection in images. Some of these methods are Objectness [ADF10], category-independent object proposals [EH10], SelectiveSearch [UvdSGS13], MCG [APTB$^+$14], GOP [KK14], BING [CZLT14], EdgeBoxes [ZD14] among others. A comparative literature survey on object proposal methods and their evaluations can be found in [HBS14, HBDS15]. Although there is no "best" detection proposal method, EdgeBoxes, which scores windows based on edge content, achieve better balance between recall and repeatabilty.

Applying image object proposals directly for each frame in video may be problematic due to time complexity and temporal consistency. In addition, issues like motion blur and compression artifacts can pose significant obstacles to identifying spatial contours, which degrades the object proposal qualities. Recent advances like SPPnet [HZRS14], Fast R-CNN [Gir15], and Faster R-CNN [RHGS15] have dramatically reduced the running time by computing deep features for all image locations at the same time and snapping them on appropriate proposal boxes. Per-frame object detection still needs clas-

sification of proposal windows and temporal consistency still remains a challenge. The proposed framework dispenses with the need of classifying every candidate window of every video frame through spatio-temporal clustering. With minimum memory overhead it smartly analyzes the video instead of treating each frame as an independent image, thus addresses temporal consistency.

Motion Segmentation in Video: Motion based segmentation is the task of separating moving foreground objects from the background. Several popular methods of motion segmentation include the layered Directed Acyclic Graph (DAG) based framework [ZJS13], Maximal weight cliques [ML12], fast motion segmentation [PF13], tracking many segments [LKH$^+$13], identifying key segments [LKG11], and many more. Although video motion segmentation can detect moving foregrounds robustly, it is not easy to detect multiple objects or if objects suddenly stop or change motion abruptly.

Supervoxel aggregation: Spatio-temporal object proposals have been considered in the context of aggregating supervoxels with spatio-temporal connectivity between neighboring labels. Jain *et al* [JGJ$^+$14] developed an extension of the hierarchical clustering method of SelectiveSearch [UvdSGS13] to obtain object proposals in video. Even though their independent motion evidence effectively segment objects with motions from the background, static objects can not be recovered. Oneata *et al.* [ORV$^+$14] presented spatio-temporal object proposals by a randomized supervoxel merging process. Sharir *et al.* [ST12] proposed the extension of category-independent object proposals [EH10] from image to video by extracting object proposals at each frame and linking across frames into object hypotheses in a framework of graph-based segmentation using higher-order potentials leading to a high computational expense. All these supervoxels often cannot replace the object proposal step for object detection either due to its complexity or the associated over-segmentations.

We propose a novel method for efficiently detecting temporally consistent static or moving objects in videos through class label propagation. We also demonstrate that object segmentation can be achived as a by-product of this detection framework.

3.3 Video Object Proposals

We extend EdgeBoxes [ZD14] from generating image object proposals to video object proposals. In addition to the spatial edge responses, $\mathbf{E_s}$, at every pixel in EdgeBoxes [ZD14], we consider exploiting temporal edge responses, $\mathbf{E_t}$, at every pixel location using mid-range optical flow analysis. where $\mathbf{E_t} \in \mathbb{R}_{\geq 0}^{M \times N}$; M and N are the height and width of a video frame.

Spatio-Temporal Contours and VOP

Optical flow field between pairs of consecutive frames provide approximate motion contours. We perform 2-frame forward optical flow [BM11] for every consecutive frame-pair. At every pixel, the magnitude of the flow field's gradient and the difference in direction of motion from its neighbor [PF13] contribute to the measure of the motion contour. To address incompleteness and inaccuracies of two-frame optical flow estimation, we analyze mid-range optical flow over a subset of video frames. Within a sub-sequence, we approximate which pixels consistently reside inside a moving object using inside-outside maps [PF13]. In our experiments, the inside-outside maps, accumulated over $3to5$ frames, provide good estimates of time-consistent gross location priors for moving objects. A simple edge detector on this location prior is called temporal edge, $\mathbf{E_t}$.

EdgeBoxes [ZD14] employs efficient data structures to score millions of candidates based on the difference of number of spatial contours that exist in the box and those that straddle box's boundary. We use a similar scoring strategy, but on spatio-temporal

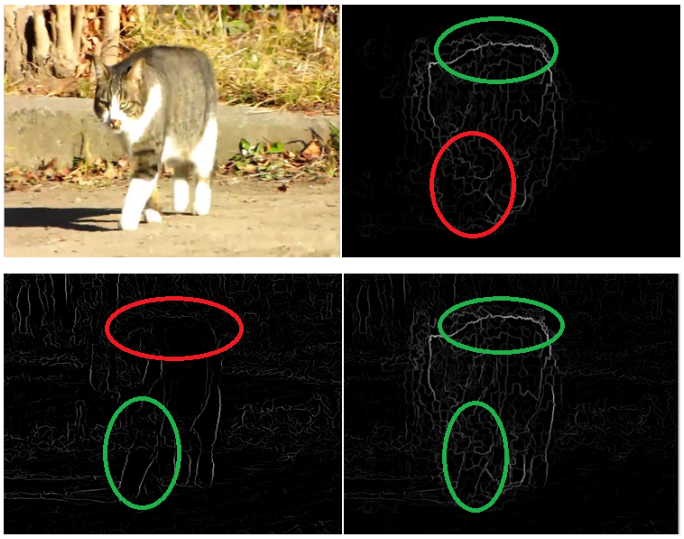


Figure 3.2: Spatio-Temporal EdgeBoxes. Clock-wise from top-left - (i) A frame from Youtube-Objects , (ii) Temporal edge $\mathbf{E}_t$ (iii) Spatial edge $\mathbf{E}_s$ (iv) $\mathbf{E}$, linear combination of $\mathbf{E}_s$ and $\mathbf{E}_t$. Green and Red circles highlight strong responses and missing responses respectively.

edge $\mathbf{E} \in \mathbb{R}_{\geq 0}^{M \times N}$ which is formed according to Eq 3.1.

$$\mathbf{E} = \lambda \mathbf{E_t} + (\mathbf{1} - \lambda)\mathbf{E_s}, \lambda \in [0, 1] \tag{3.1}$$

As the value of λ increases, the system favors detecting only moving objects. One example of spatio-temporal edge is demonstrated in figure 3.2. A linear combination of spatial and temporal edge responses represents spatio-temporal contours. This enables a simple yet efficient strategy for scoring based on spatio-temporal edge content through edge groups. We find intersecting edge groups along horizontal and vertical boundaries using two efficient data structures [ZD14]. Similar to EdgeBoxes, we also use the integral image-based implementation to speed up scoring of boxes in sliding window fashion. As demonstrated in the next section, object proposals based on these spatio-temporal contours outperforms those based on only spatial contours for video object detection. The presence of motion blur in spatial edges affects the performance of spatial contour

based proposal prediction in video frames. In practice, $\lambda = 0.2$ to 0.5 works well for the desired Youtube-Objects [PKL$^+$14] dataset.

3.4 Learning Video Object Detector Model

We aim for detecting objects in generic consumer videos. Due to the domain shift issues between images and video frames [KFS15], our 10-class video object detection uses supervised pre-training from ImageNet reference model for classification and fine-tuning on annotated frames from Youtube-Objects dataset v2.0 [PKL$^+$14, KFS15, PLC$^+$12a] for video objects detection.

Youtube-Objects dataset. The dataset is composed of videos collected from Youtube by querying for the names of 10 object classes of the PASCAL VOC Challenge. It contains 155 videos in total and between 9 and 24 videos for each class. The duration of each video varies between 30 seconds and 3 minutes. However, only 6087 frames are annotated with a bounding-box around an object instance. Hence, the number of annotated samples is approximately $4\times$ smaller than in PASCAL VOC.

The bottom-up region proposal methods play an important role. Motion blur and compression artifacts affect the quality of spatial edges in video frames. Thus generating good object proposals becomes more challenging. This is to be noted that R-CNN [GDDM14], or, Fast R-CNN [Gir15] are fine-tuned for image object detection task, especially for 20-class PASCAL VOC image object categories which is a superset of Youtube-objects categories.

Feature extraction. We extract a 4096-dimensional feature vector corresponding to each region proposal with GPU (GeForce GTX 680) using Caffe [JSD$^+$13] implementation of the CNN described by Krizhevsky *et al.* [KSH12]. Features are computed by forward propagating a mean-subtracted 227×227 R-G-B image through five convolu-

tional layers and two fully connected layers.

Region Proposals. We use approximately 2000 candidate proposals per video frame to be processed for learning detectors. We investigate the object detection model with different region proposal methods such as selective search[UvdSGS13] and EdgeBoxes[ZD14]. As the resolution of different videos varies from VGA to HD, we resize every video frame to 500×500 before performing proposal generation task.

3.4.1 Training

We discriminatively pre-train the CNN on a large auxiliary dataset (ILSVRC2012 classification) using image-level annotations, followed by domain specific fine-tuning by replacing the last layer of AlexNet [KSH12] model with $10 + 1$ softmax output layer. We use two-step initialization for fine-tuning as described in [BHPB14]. As per PASCAL detection criteria, we treat all region proposals with $\geq 50\%$ Intersection Over Union (IoU) overlap with a ground-truth box as positive examples for that class of the box and the rest as negatives. Once features are extracted and training labels are applied, we optimize one linear SVM per class.

3.4.2 Inference

During test time, forward propagation is performed through the CNN to compute features of approximately 500 to 2000 VOPs. Next, we perform scoring using per-class trained SVM similar to [GDDM14] followed by non-maximum suppression.

In the next section 3.5, we describe an efficient detection framework, *OVERLAP*. OVERLAP stands for *Objects in Video Enabler thRough LAbel Propagation*.

3.5 OVERLAP Algorithm

Classical approach for object localization has traditionally been image window classification, where each window is scored independent of other candidate windows. Recently, more success in object detection has been reported by considering spatial relations to all other windows and their appearance similarity [VF15] with examplar-based associative embedding. Our approach towards video object detection considers spatial relationship and appearance similarity with windows within and even in other nearby video frames, yet in much simpler way through spatio-temporal clustering, to avoid classifying every candidate windows.

Joint analysis of windows: We aim to detect dissimilarity between video Object Proposals (VOPs) generated within a sub-sequence based on a simple underlying principle: proposals corresponding to the same object exhibit higher statistical dependencies than proposals belonging to different objects.

As a motivation for our approach, we consider generating proposal boxes by perturbing the ground truth locations of PASCAL VOC 2007 objects' bounding boxes and observe the statistical association of those proposal boxes. Let, A and B denote generic features of neighboring proposal windows, where neighborhood is characterized by non-zero Intersection-Over-Union, denoted by u. We investigate the joint distribution over pairings $\{A,B\}$. Let, $p(A,B;u)$ be the joint probability of features A and B of windows with spatial overlap value, u. Then, $P(A,B)$ could ideally be computed as :

$$P(A,B) = \frac{1}{Z}\int_{u}^{1} w(u)p(A,B;u)du \tag{3.2}$$

where, $u \in [0,1]$, w is a weighting function, $w(0) = 0$, and Z is a normalization constant. To simplify the process, we use uniform weighting function and work in the discrete

(quantized) space of u and replace the integral with summation. We take the marginals of the distribution to get $P(A)$ and $P(B)$. Motivated by the analysis presented in crisp boundary detection work by Isola *et al.* [IZKA14], we model affinity with point-wise mutual information like function:

$$PMI_{\rho}(A,B) = log\frac{P(A,B)^{\rho}}{P(A)P(B)} \tag{3.3}$$

We choose the value of ρ to be 1.2 which produces best performance in PASCAL VOC dataset with perturbed ground truth proposals. In order to identify the boundary between two features, the model needs to be able to capture the low probability regions of P(A,B). We use a non-parametric kernel (Epanechnikov) density estimator. The number of sample points are the number of overlapping candidate windows. We perform affinity-based clustering afterwards. The affinity matrix, $\mathbf{W}$, for a sub-sequence is created from the affinity function, PMI_{ρ}, as follows:

$$\mathbf{W}_{i,j} = e^{PMI_{\rho}(\mathbf{f}_i,\mathbf{f}_j)} \tag{3.4}$$

Where i and j are the indices of proposal windows and $\mathbf{f}$ is the feature vector defined for a proposal window. Figure 3.3 shows an example of spatial clustering of proposal windows. The boxes drawn in same colors correspond to same clusters. In the streaming VOP clustering framework for Youtube videos, we perform the joint analysis on all proposals within a sub-sequence. Intuitively this is an easy yet effective clustering technique which works on actual test proposals (see Figure 3.8) which are not simply perturbed ground truth bounding boxes.

This is to be noted that the proposed method measures the affinity between different object proposal windows (of varying sizes) within a video sub-sequence unlike [IZKA14], where the affinity is between the neighboring pixels in an image.

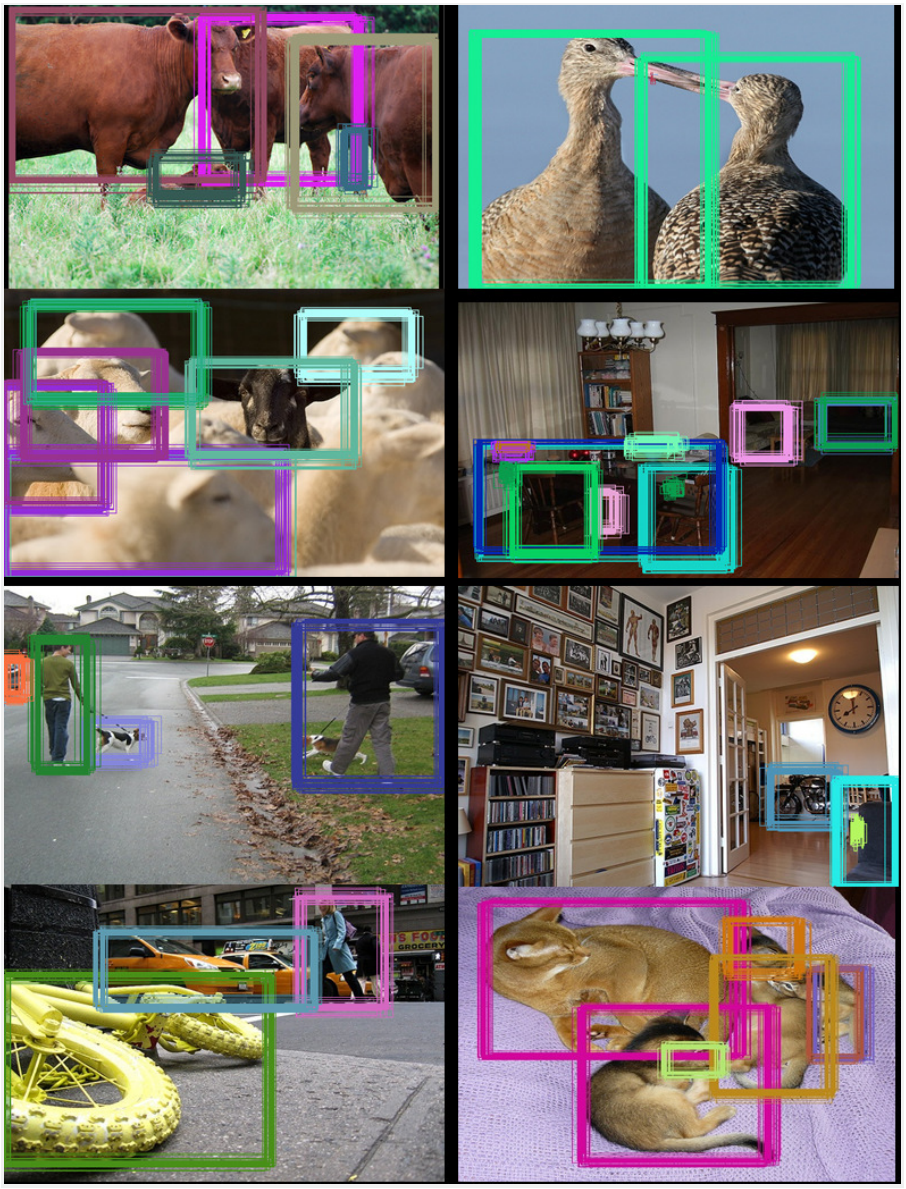


Figure 3.3: Synthetic experiment using perturbed ground truth boxes in PASCAL VOC showing affinity-based clustering of candidate windows. The clustering technique works efficiently with some exceptions where overlapping object instances share very similar color.

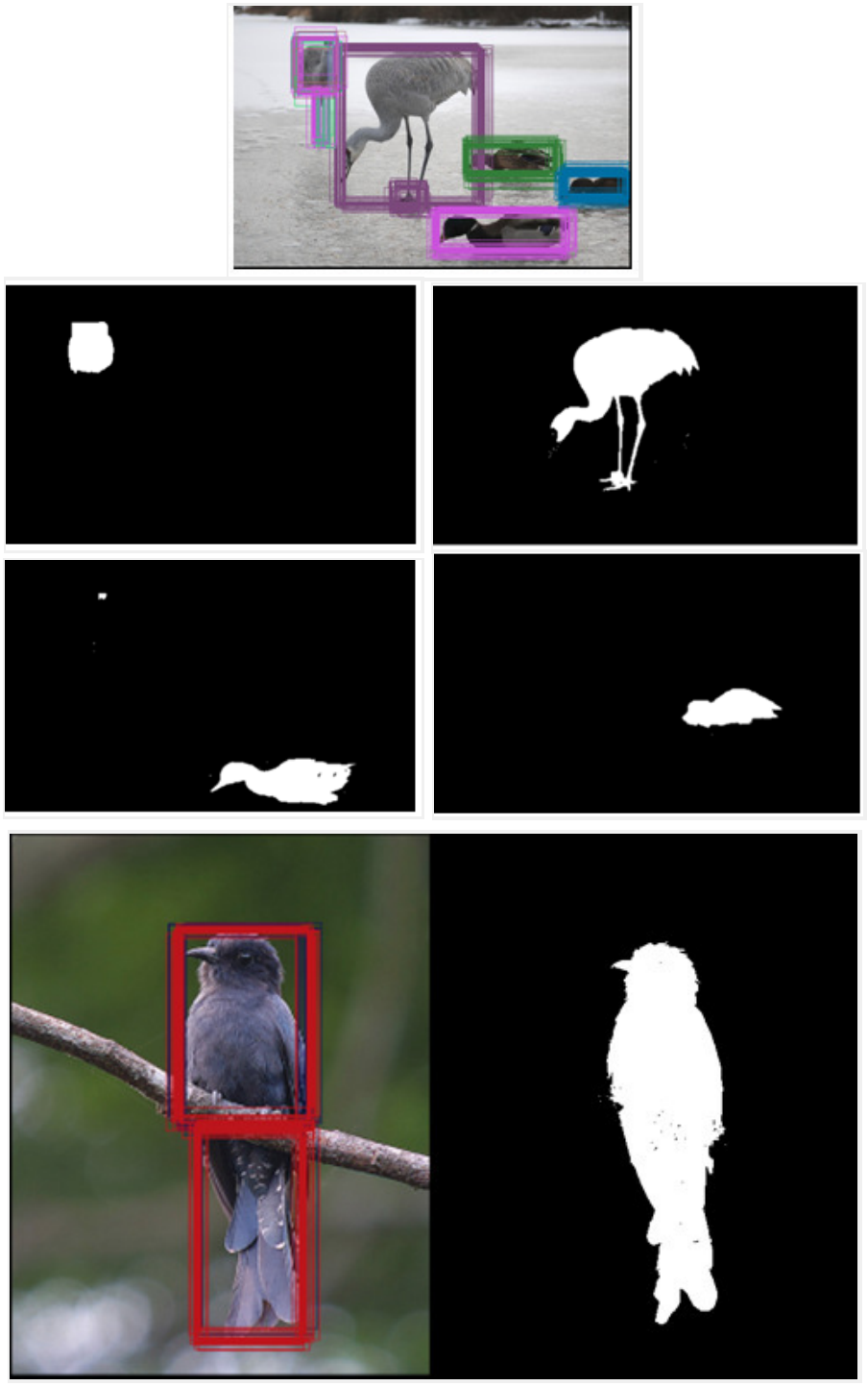


Figure 3.4: Segmentation masks on two different images from clustered object proposals.

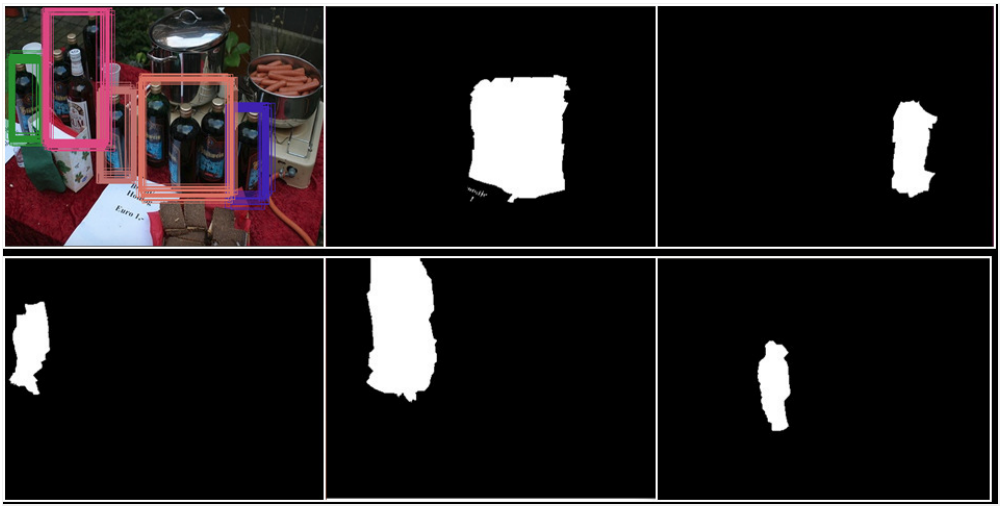


Figure 3.5: Segmentation masks generation is not successful where proposal boxes do not tightly enclose the actual object. This happens in cluttered background cases.

We demonstrate that object segmentation can be achieved as a by-product of this clustering algorithm. We cast the segmentation problem through random-field based background-foreground segmentation without manual labeling [KGF12]. Uniformly weighted sum of the location of every window corresponding to a unique cluster defines the foreground location prior for that cluster. However, unlike [KGF12], in our approach, the location prior is not coming from the global neighbors of the image but from within itself and the clustering allows multiple objects segmentations. The segmentation works well if the proposal boxes tightly enclose an object as shown in Figure 3.4 with some failure cases as shown in Figure 3.5 where the proposal boxes do not tightly enclose the object in a cluttered background.

Figure 3.6 shows two segmentation masks generated on individual video frames from the clustered VOPs generated by the proposed method on real videos.

Streaming clustering of proposals: One of the main contributions of this paper is a simple, principled, and unsupervised approach to spatio-temporal grouping of candi-

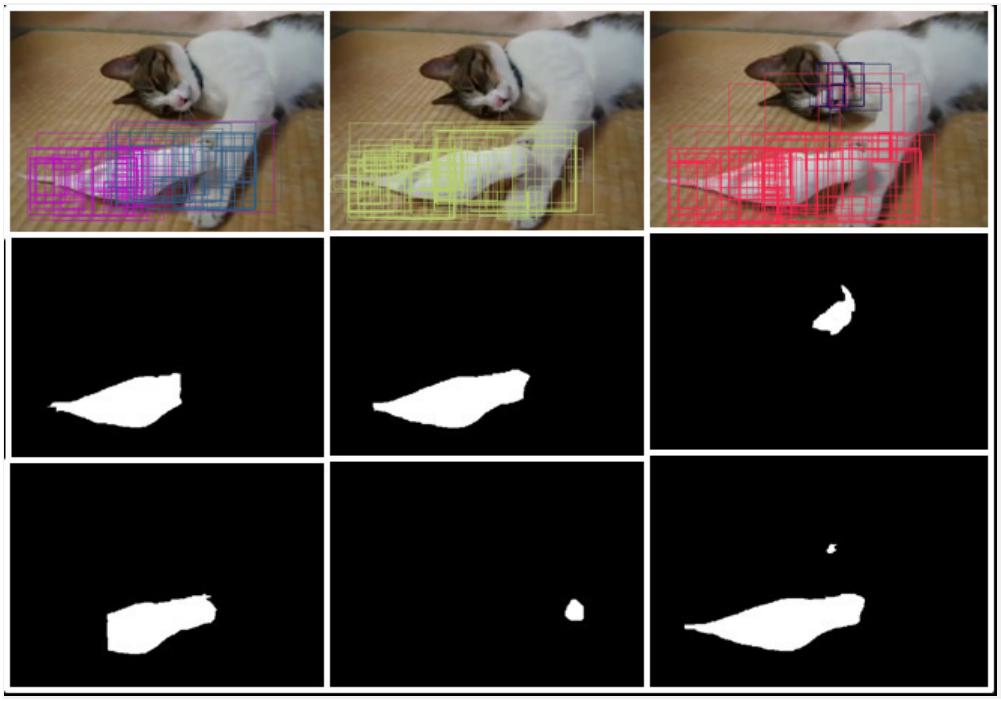


Figure 3.6: Two segmentation masks from clustered video object proposals from Youtube video "Bird and Cat" for frames #5, #20 and #45.

date regions in streaming fashion. We describe a clustering framework which enforces a Markovian assumption on the video stream to approximate a batch grouping of VOPs. A video is divided into a series of sub-sequences with one frame overlap with the previous sub-sequence as described in [XXC12b]. VOP clustering within the current sub-sequence depends on the results from only the previous sub-sequence. We consider sub-sequence length of 3 to 5 frames as a trade-off between quality and complexity. This is the same sub-sequence volume, where mid-range motion analysis is performed for detecting temporal edges (Section 3.3). The color-histogram features are used for estimating joint probability between any overlapping window-pair and affinity-based clustering is performed afterwards. There are two important aspects in this streaming clustering method. The first is generic to any clustering algorithm *i. e.* how to select the number of clusters and the second is specific to streaming method *i. e.* how to associate cluster number of the current sub-sequence with any of the clusters of previous and/or future sub-sequences?

Number of clusters: Common consumer videos or Youtube videos contain limited number of moving objects, often less than five. Youtube-Objects dataset contains maximum of 3 object instances and quite often a single moving object. We assume the presence of at most 5 objects to keep the computational complexity tractable and amenable to practical applications. We explore two modes of operations. The first method uses fixed number of clusters, $k = 5$, with careful initialization of cluster centers using k-mean++ [AV07] during spectral clustering. The second one is the spectral clustering with self-tuning [ZMP04]. We observe that while self-tuning outperforms the fixed cluster number case for hypothetically good object windows (such as the perturbed ground truth regions for PASCAL VOC), both modes perform similarly in case of real object proposals generated by some proposal method.

Cluster Label Association: In the streaming framework, any sub-sequence except for the first one, needs to address the problem of either associating a cluster with a cluster number in the previous sub-sequence or generating a new one. We perform density estimation using an Epanechnikov kernel using the KD-tree implementation from [IM] for every cluster using the 4-dimensional location (2D center, height and width) and 45-bin color histogram (15 for each color channel) of the regions of the proposals corresponding to each and every cluster. If the minimum KL-divergence between a distribution of the current cluster and a cluster from the previous sub-sequence is less than a threshold, we perform the cluster assignment. Otherwise, we create a new cluster.

This is to be noted that, for detecting primarily moving objects in videos, the weight of temporal edges could be as high as 0.7 or more as described in section 3.3. In such cases, considering as low as only 100 VOPs can potentially detect moving objects. Clusters may contain fewer number of proposal windows than the dimension of the original feature space which is 49-dimensional. Thus we perform PCA-based dimensionality reduction before estimating the distribution. Also, we perform scaling of features to ease the process of selecting kernel evaluation points.

Object Label Propagation: Time-consistent clustering enables object label propagation through the video. We perform CNN-based object detection *i. e.* classification of every window for a cluster in CNN feature space at a video frame only when we encounter a new cluster label. An assigned previously created cluster label means the object category is the same as what was already detected in the associated cluster of the previous sub-sequence. We still need to perform the localization, however. In order to address the localization, we experimented with two strategies. First, applying the same non-maximal suppression for all the proposals belonging to the same cluster. Second, we fit a 4-D Gaussian distribution on the location parameters *i. e.* center (x,y), height(h)

and width(w), of windows in a cluster. We simply keep track of the distance, **d**, of the detected final object location (after first-time detection using R-CNN like approach) from the mean of the fitted Gaussian for every cluster. Furthermore, we localize the object by adding **d** with the mean of the 4-D Gaussian location distribution of the cluster in current sub-sequence. Generally in videos, new objects do not appear in every video frame. Thus, we do not need to detect objects at every video frame. Even when a new object appears, we need to detect/classify only for the proposals assigned to the new cluster. Thus, OVERLAP framework requires to process CNN features for only a small fraction of the number of proposals generated.

In some sense, the spatio-temporal clustering for object detection is related to tracking. However, a set of windows is tracked instead of a single region/object. Ability to bypass critical tracker initialization step and the possibility for applying R-CNN like detection at a more frequent and configurable interval to increase the detection accuracy (if needed) are the major advantages comparing with object tracking based detection.

3.6 Results and Discussions

Video Object Detection using VOP: We observe that the proposed video Object Proposals (VOPs) help in learning a better object detector model. Table 3.1 shows the per-class detection accuracy and the mean Average Precision (mAP) for the 10-class Youtube-Objects [PKL$^+$14] test set.

Fine-tuning on Youtube-Objects training data with Selective Search proposals [UvdSGS13] improves the detection results by at least 9% compared with the model fine-tuned for the image dataset PASCAL VOC. The detector learned with EdgeBoxes performs better than the one learned with Selective Search proposals. However, the detector learned with VOP even outperforms the detection rate by another 5.5% and

Table 3.1: Object Detection Results on Youtube-Objects test set. Pre-trained R-CNN detector is downloaded from [GDDM]. DPM Detection results are from [KFS15]. SS, EB and VOP correspond to Selective Search, EdgeBoxes and proposed VOP ($\lambda = 0.2$).

classes	R-CNN	DPM	Fine-tune SS	Fine-tune EB	Fine-tune VOP
plane	14.1	28.42	25.57	26.52	**29.77**
bird	24.2	**48.14**	27.27	27.27	28.82
boat	16.9	25.50	27.52	33.69	**35.34**
car	27.9	**48.99**	35.18	36	41
cat	17.9	1.69	25.02	27.05	**33.7**
cow	28.6	19.24	43.01	44.76	**57.56**
dog	12.2	15.84	24.05	27.07	**34.42**
horse	29.4	35.10	41.84	44.82	**54.52**
mbike	21.3	**31.61**	26.70	27.07	29.77
train	13.2	**39.58**	20.48	24.93	29.23
mAP	20.57	29.41	29.67	31.92	**37.413**

Table 3.2: Complexity and accuracy. Proposal generation (Prop time), per-frame (PF) baseline methods with Selective Search, EdgeBoxes, proposed VOPs and OVERLAP. OF denotes optical flow. mAP increases as #VOPs increases. $3\times$ speedup with 500 VOPs and 4% drop in mAP.

	PF SS	PF EB	PF VOP		OVERLAP							
			CPU OF	GPU OF	CPU Optical Flow				GPU Optical Flow			
					200 V	500 V	1K V	2K V	200 V	500 V	1K V	2K V
Prop time	10	0.3	3.8	1.3	3.8				1.3			
Overall time	30	20.3	23.8	21.3	6.2	9.3	14.6	28.0	3.7	6.8	12.1	26.5
mAP	29.62	31.95	37.72	37.72	28.59	33.59	35.82	36.63	28.59	33.59	35.82	36.63

achieves state-of-the-art detection accuracy (37.4%) on this dataset. Although detection accuracy for "cat", "cow", "dog" and "horse" have improved by a huge margin after using CNN features, categories like "train" and "bird" are still best detected using DPM [FGMR10] detector.

Streaming Clustering of VOP : Figure 3.8 shows the results of frame-level clustering vs streaming clustering at sub-sequence levels on arbitrary videos downloaded from Youtube. In these experiments, a sub-sequence contains 3 video frames, with an overlap of one frame with the previous sub-sequence. We use high λ value (0.8)

to identify only the moving objects with very few number of proposals. For clear visualization, we use only 50 proposal windows at every frame and that makes less than 200 proposals per sub-sequence. We aim to take the advantage of fast approximate spectral clustering algorithm which scales linearly with the problem size. In our current implementation, clustering takes less than 0.1 second for 3 frames streaming-volume with 50 VOPs per frame.

Video Object Detection: To investigate the relative detection rate with OVERLAP comparing with frame-wise R-CNN like approach, we create a subset (955 frames) of Youtube-Objects test-set (1783 frames) where the video frames form a valid video play. We find that for OVERLAP, CNN feature extraction and classification is needed only for 10% to 30% windows among all of them. Table 3.2 corroborates the fact that the detection accuracy improves as we increase the number of VOPs from 200 to 2000 in OVERLAP at the cost of increased complexity needed for spectral clustering. Difference in mAP is between 1-9% . As an example, compared to per-frame detection, OVERLAP achieves about $3\times$ speedup at the cost of only 4% mean Average Precision (mAP) with 500 VOPs per frame. The non-GPU based spectral clustering implementation in MATLAB makes cases for more than 2000 VOPs even slower than per-frame RCNN.

Per-frame proposal generation with Selective Search and EdgeBoxes takes about 10 seconds [HBDS15] and 0.3 seconds [HBDS15] respectively. Overall time for object detection per-frame in R-CNN per-frame becomes about 30 seconds and 20.3 seconds with the above corresponding methods. Generation of VOPs requires optical flow which takes 3.5 sec per frame in CPU-implementation and 1 sec per frame for GPU-implementation for about 500×500 resolution frame-pairs. "CPU" and "GPU" in Table 3.2 denote optical flow implementation in CPU [BM11] and GPU [SBK10] respectively. The Youtube-Objects dataset mostly contains moving objects. In addition, the test dataset does not contain significant number of test cases where multiple instances of objects with

similar appearances are spatially overlapped. Thus, accuracy of OVERLAP successfully approaches the baseline per-frame detection accuracy as we increase the number of VOPs. For the fastest detection (with 200 VOPs only), we use more weight (lambda = 0.6) for temporal edge and still manage to get acceptable detection accuracy with over 5$\times$ speedup as shown in the column corresponding to 200 VOP in Table 3.2. GPU-based spectral clustering can potentially lead to further speed up.

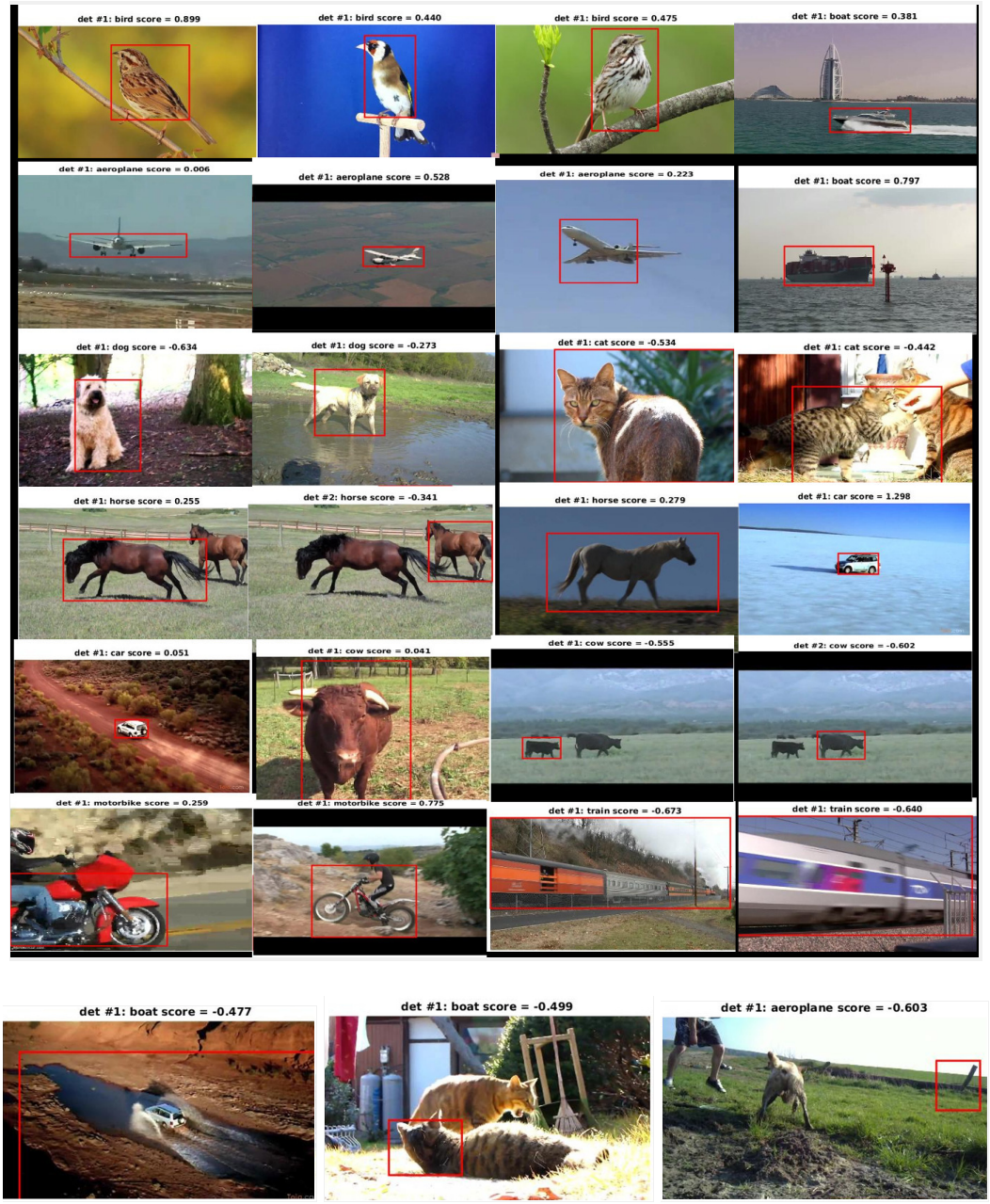


Figure 3.7: Sample results of Video Object Detection with VOP. First 6 rows show successful detection cases and the last row shows false detection cases.

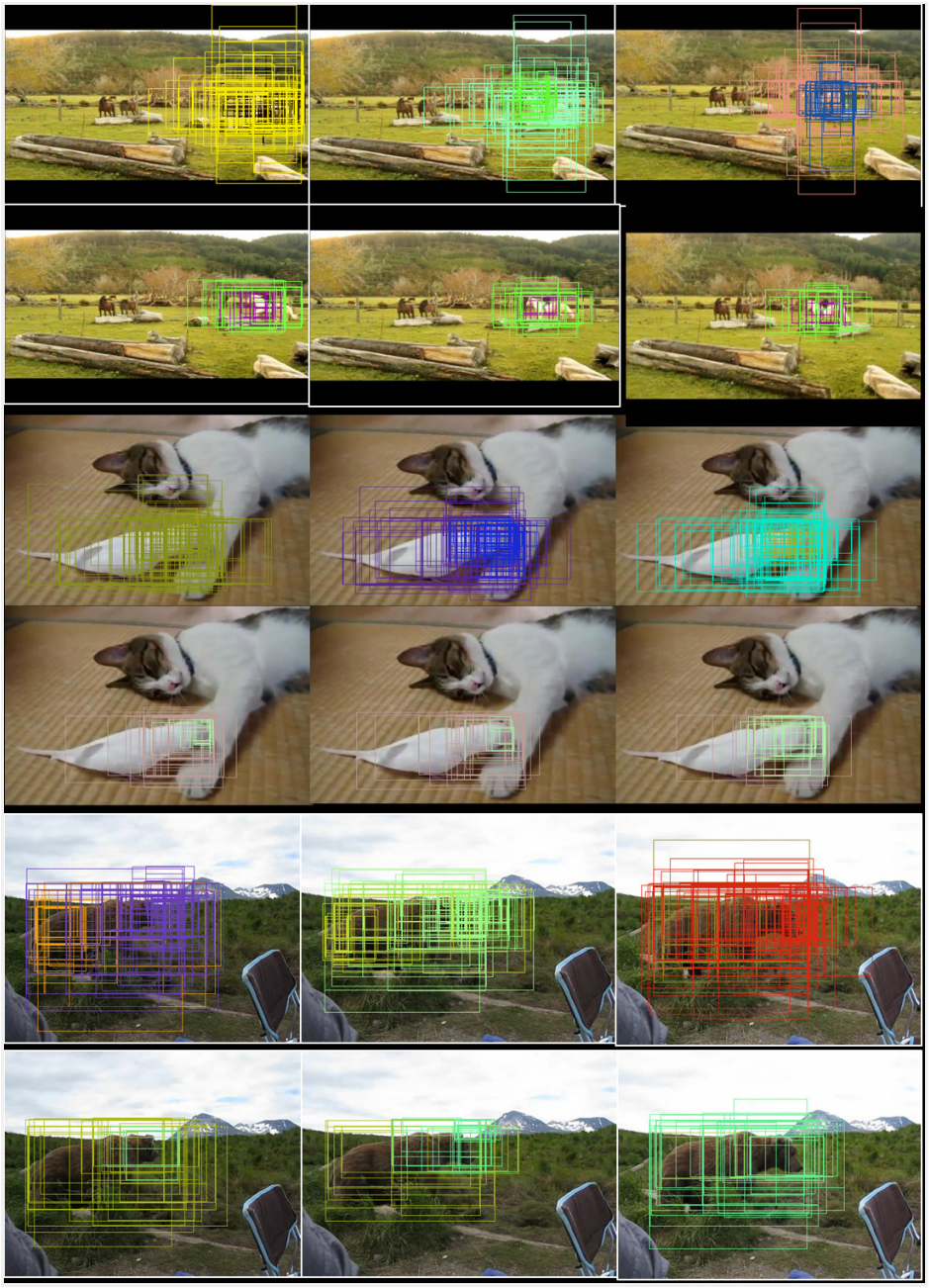


Figure 3.8: Temporally consistent VOPs on “Horse riding”, “Bird-cat” and “Alaskan bear”. Windows drawn in same color belong to same cluster. Top row denotes clustered proposals on sample frames #2, #10, #25 ; bottom row shows the results of streaming clustering.

3.7 Summary

Experimental results show that VOP helps in learning a better moving or static video object detector model and achieves state-of-the-art detection accuracy on Youtube-Video dataset. We show that the proposed OVERLAP framework can detect temporally consistent objects in videos through object class label propagation using streaming clustering of VOPs with significant speedup compared with naive per-frame detection with acceptable loss of accuracy. We also show that multiple objects segmentation can also be achieved as a by-product of OVERLAP.

Acknowledgments

Chapter 3 is in part based on the paper "Detecting Temporally Consistent Objects in Videos through Object Class Label Propagation", by S. Tripathi, S. Belongie, Y.B. Hwang, and T. Nguyen [TBHN16]. The dissertation author was the primary investigator and author of this paper.

Chapter 4

Object Detection in Videos using Recurrent Neural Network

4.1 Background

Online Video content via outlets such as Youtube and Facebook are attaining popularity as huge source of supervisory signals for vision. Given the breakthroughs of deep convolutional neural networks for detecting objects in static images, the application of these methods to video might seem straightforward. However, motion blur and compression artifacts cause substantial frame-to-frame variability, even in videos that appear smooth to the eye. Reiterating video object detection problem from the last chapter, these attributes complicate prediction tasks like classification and localization. Object-detection models trained on images tend not to perform competitively on videos owing to domain shift factors [KFS15]. Moreover, object-level annotations in popular video data-sets can be extremely sparse, impeding the development of better video-based object detection models.

Girshik *et al.* [GDDM14] demonstrate that even given scarce labeled training

data, high-capacity convolutional neural networks can achieve state of the art detection performance if first pre-trained on a related task with abundant training data, such as 1000-way ImageNet classification. Followed the pretraining, the networks can be fine-tuned to a related but distinct domain. Also relevant to our work, the recently introduced models Faster R-CNN [RHGS15] and You Look Only Once (YOLO) [RDGF16] unify the tasks of classification and localization. These methods, which are accurate and efficient, propose to solve both tasks through a single model, bypassing the separate object proposal methods used by R-CNN [GDDM14].

In this chapter, we introduce a method to extend unified object recognition and localization to the video domain. Our approach applies transfer learning from the image domain to video frames. Additionally, we present a novel recurrent neural network (RNN) method that refines predictions by exploiting contextual information in neighboring frames. to summarize, we propose a method for refining a video-based object detection consisting of two parts: (i) a *pseudo-labeler*, which assigns provisional labels to all available video frame; (ii) (ii) A recurrent neural network, which reads in a sequence of provisionally labeled frames, using the contextual information to output refined predictions. Next, we demonstrate an effective training strategy utilizing (i) category-level weak-supervision at every time-step, (ii) localization-level strong supervision at final time-step (iii) a penalty encouraging prediction smoothness at consecutive time-steps, and (iv) similarity constraints between *pseudo-labels* and prediction output at every time-step. Finally, we perform an extensive empirical investigation demonstrating that on the YouTube Objects [PKL$^+$14] dataset, our framework achieves mean average precision (mAP) of 68.73 on test data, compared to a best published result of 37.41 [TBHN16] and 61.66 for a domain adapted YOLO network [RDGF16].

4.2 Related Work

Our work builds upon a rich literature in both image-level object detection,video analysis, and recurrent neural networks. Several papers propose ways of using deep convolutional networks for detecting objects [GDDM14, Gir15, RHGS15, RDGF16, SREA14, BZBG15, OWZ+15, SEZ+13, YYLL16, GK15]. Some approaches classify the proposal regions [GDDM14, Gir15] into object categories and some other recent methods [RHGS15, RDGF16] unify the localization and classification stages. Kalogeiton *et al.* [KFS15] identifies domain shift factors between still images and videos, necessitating video-specific object detectors. To deal with shift factors and sparse object-level annotations in video, researchers have proposed several strategies. Recently, [TBHN16] proposed both transfer learning from the image domain to video frames and optimizing for temporally consistent object proposals. Their approach is capable of detecting both moving and static objects. However, the object proposal generation step that precedes classification is slow.

Prest *et al.* [PLC+12b], utilize weak supervision for object detection in videos via category-level annotations of frames, absent localization ground truth. This method assumes that the target object is moving, outputting a spatio-temporal tube that captures this most salient moving object. This paper, however, does not consider context within video for detecting multiple objects.

A few recent papers [OWZ+15, BZBG15] identify the important role of context in visual recognition. For object detection in images, Bell *et al.* [BZBG15] use spatial RNNs to harness contextual information, showing large improvements on PASCAL VOC [EGW+10] and Microsoft COCO [LMB+14b] object detection datasets. Their approach adopts proposal generation followed by classification framework. This paper exploits spatial, but not temporal context.

Recently, Kang *et al.* [KOLW16] introduced tubelets with convolutional neural networks (T-CNN) for detecting objects in video. T-CNN uses spatio-temporal tubelet proposal generation followed by the classification and re-scoring, incorporating temporal and contextual information from tubelets obtained in videos. T-CNN won the recently introduced ImageNet object-detection-from-video (VID) task with provided densely annotated video clips. Although the method is effective for densely annotated training data, it's behavior for sparsely labeled data is not evaluated.

By modeling video as a time series, especially via GRU [CvMBB14] or LSTM RNNs[HS97], several papers demonstrate improvement on visual tasks including video classification [YHNHV$^+$15], activity recognition [DHG$^+$14], and human dynamics [FLFM15]. These models generally aggregate CNN features over tens of seconds, which forms the input to an RNN. They perform well for global description tasks such as classification [YHNHV$^+$15, DHG$^+$14] but require large annotated datasets. Yet, detecting multiple generic objects by explicitly modeling video as an ordered sequence remains less explored.

Our work differs from the prior art in a few distinct ways. First, this work is the first, to our knowledge, to demonstrate the capacity of RNNs to improve localized object detection in videos. The approach may also be the first to refine the object predictions of frame-level models. Notably, our model produces significant improvements even on a small dataset with sparse annotations.

4.3 Methodology

In this work, we aim to refine object detection in video by utilizing contextual information from neighboring video frames. We accomplish this through a two-stage process. First, we train a *pseudo-labeler*, that is, a domain-adapted convolutional neural

network for object detection, trained individually on the labeled video frames. Specifically, we fine-tune the YOLO object detection network [RDGF16], which was originally trained for the 20-class PASCAL VOC [EGW$^+$10] dataset, to the Youtube-Video [PKL$^+$14] dataset.

When fine-tuning to the 10 sub-categories present in the video dataset, our objective is to minimize the weighted squared detection loss (equation 7.1) as specified in YOLO [RDGF16]. While fine-tuning, we learn only the parameters of the top-most fully-connected layers, keeping the 24 convolutional layers and 4 max-pooling layers unchanged. The training takes roughly 50 epochs to converge, using the RMSProp [TH12] optimizer with momentum of 0.9 and a mini-batch size of 128.

As with YOLO [RDGF16], our fine-tuned $pseudo-labeler$ takes 448×448 frames as input and regresses on category types and locations of possible objects at each one of $S \times S$ non-overlapping grid cells. For each grid cell, the model outputs class conditional probabilities as well as B bounding boxes and their associated confidence scores. As in YOLO, we consider a *responsible* bounding box for a grid cell to be the one among the B boxes for which the predicted area and the ground truth area shares the maximum Intersection Over Union. During training, we simultaneously optimize classification and localization error (equation 7.1). For each grid cell, we minimize the localization error for the *responsible* bounding box with respect to the ground truth only when an object appears in that cell.

Next, we train a Recurrent Neural Network (RNN), with Gated Recurrent Units (GRUs) [CvMBB14]. This net takes as input sequences of *pseudo-labels*, optimizing an objective that encourages both accuracy on the target frame and consistency across consecutive frames. Given a series of *pseudo-labels* $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(T)}$, we train the RNN to generate improved predictions $\hat{\mathbf{y}}^{(1)}, ..., \hat{\mathbf{y}}^{(T)}$ with respect to the ground truth $\mathbf{y}^{(T)}$ available only at the final step in each sequence. Here, t indexes sequence steps and T denotes the

length of the sequence. As output, we use a fully-connected layer with a linear activation function, as our problem is regression. In our final experiments, we use a 2-layer GRU with 150 nodes per layer, hyper-parameters determined on validation data.

The following equations define the forward pass through a GRU layer, where $\mathbf{h}_l^{(t)}$ denotes the layer's output at the current time step, and $\mathbf{h}_{l-1}^{(t)}$ denotes the previous layer's output at the same sequence step:

$$\begin{aligned}
\mathbf{r}_l^{(t)} &= \sigma(\mathbf{h}_{l-1}^{(t)} W_l^{xr} + \mathbf{h}_l^{(t-1)} W_l^{hr} + \mathbf{b}_l^r) \\
\mathbf{u}_l^{(t)} &= \sigma(\mathbf{h}_{l-1}^{(t)} W_l^{xu} + \mathbf{h}_l^{(t-1)} W_l^{hu} + \mathbf{b}_l^u) \\
\mathbf{c}_l^{(t)} &= \sigma(\mathbf{h}_{l-1}^{(t)} W_l^{xc} + r_t \odot (\mathbf{h}_l^{(t-1)} W_l^{hc}) + \mathbf{b}_l^c) \\
\mathbf{h}_l^{(t)} &= (1 - \mathbf{u}_l^{(t)}) \odot \mathbf{h}_l^{(t-1)} + \mathbf{u}_l^{(t)} \odot \mathbf{c}_l^{(t)}
\end{aligned} \tag{4.1}$$

Here, σ denotes an element-wise logistic function and $\odot$ is the (element-wise) Hadamard product. The reset gate, update gate, and candidate hidden state are denoted by $\mathbf{r}$, $\mathbf{u}$, and $\mathbf{c}$ respectively. For $S = 7$ and $B = 2$, the pseudo-labels $\mathbf{x}^{(t)}$ and prediction $\hat{\mathbf{y}}^{(t)}$ both lie in $\mathbb{R}^{1470}$.

4.3.1 Training

We design an objective function (Equation 4.2) that accounts for both accuracy at the target frame and consistency of predictions across adjacent time steps in the following ways:

$$\text{loss} = \text{d_loss} + \alpha \cdot \text{s_loss} + \beta \cdot \text{c_loss} + \gamma \cdot \text{pc_loss} \tag{4.2}$$

Here, d_loss, s_loss, c_loss and pc_loss stand for detection_loss, similarity_loss, category_loss and prediction_consistency_loss described in the following sections. The values of the hyper-parameters $\alpha = 0.2$, $\beta = 0.2$ and $\gamma = 0.1$ are chosen based on the detection performance on the validation set. The training converges in 80 epochs for parameter up-

dates using RMSProp [TH12] and momentum 0.9. During training we use a mini-batch size of 128 and sequences of length 30.

Strong Supervision at Target Frame

On the final output, for which the ground truth classification and localization is available, we apply a multi-part object detection loss as described in YOLO [RDGF16].

$$\begin{aligned}\text{detection_loss} = {} & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(x_i^{(T)} - \hat{x}_i^{(T)}\right)^2 + \left(y_i^{(T)} - \hat{y}_i^{(T)}\right)^2 \\ & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left(\sqrt{w_i}^{(T)} - \sqrt{\hat{w}_i^{(T)}}\right)^2 + \left(\sqrt{h_i}^{(T)} - \sqrt{\hat{h}_i^{(T)}}\right)^2 \\ & + \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i^{(T)} - \hat{C}_i{}^{(T)})^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} \left(C_i^{(T)} - \hat{C}_i^{(T)}\right)^2 \\ & + \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} \left(p_i^{(T)}(c) - \hat{p}_i{}^{(T)}(c)\right)^2 \end{aligned} \tag{4.3}$$

where $\mathbb{1}_i^{obj}$ denotes if the object appears in cell i and $\mathbb{1}_{ij}^{obj}$ denotes that jth bounding box predictor in cell i is *responsible* for that prediction. The loss function penalizes classification and localization error differently based on presence or absence of an object in that grid cell. x_i, y_i, w_i, h_i corresponds to the ground truth bounding box center coordinates, width and height for objects in grid cell (if it exists) and $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ stand for the corresponding predictions. C_i and $\hat{C}_i$ denote confidence score of *objectness* at grid cell i for ground truth and prediction. $p_i(c)$ and $\hat{p}_i(c)$ stand for conditional probability for object class c at cell index i for ground truth and prediction respectively. We use similar settings for YOLO's object detection loss minimization and use values of $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$.

Similarity between *Pseudo-labels* and Predictions

Our objective function also includes a regularizer that penalizes the dissimilarity between *pseudo-labels* and the prediction at each time frame t.

$$\text{similarity_loss} = \sum_{t=0}^{T} \sum_{i=0}^{S^2} \hat{C}_i^{(t)} \left(\mathbf{x}_i^{(t)} - \hat{\mathbf{y}}_i{}^{(t)} \right)^2 \tag{4.4}$$

Here, $\mathbf{x}_i^{(t)}$ and $\hat{\mathbf{y}}_i{}^{(t)}$ denote the *pseudo-labels* and predictions corresponding to the i-th grid cell at t-th time step respectively. We perform minimization of the square loss weighted by the predicted confidence score at the corresponding cell.

Object Category-level Weak-Supervision

Replication of the static target at each sequential step has been shown to be effective in [LKEW16, YHNHV$^+$15, DL15]. Of course, with video data, different objects may move in different directions and speeds. Yet, within a short time duration, we could expect all objects to be present. Thus we employ target replication for classification but not localization objectives.

We minimize the square loss between the categories aggregated over all grid cells in the ground truth $\mathbf{y}^{(T)}$ at final time step T and predictions $\hat{\mathbf{y}}^{(t)}$ at all time steps t. Aggregated category from the ground truth considers only the cell indices where an object is present. For predictions, contribution of cell i is weighted by its predicted confidence score $\hat{C}_i^{(t)}$. Note that cell indices with positive detection are sparse. Thus, we consider the confidence score of each cell while minimizing the aggregated category loss.

$$\text{category_loss} = \sum_{t=0}^{T} \left(\sum_{c \in classes} \left(\sum_{i=0}^{S^2} \hat{C}_i^{(t)} \left(\hat{p}_i^{(t)}(c) \right) - \sum_{i=0}^{S^2} \mathbb{1}_i^{obj^{(T)}} \left(p_i^{(T)}(c) \right) \right) \right)^2 \tag{4.5}$$

Consecutive Prediction Smoothness

Additionally, we regularize the model by encouraging smoothness of predictions across consecutive time-steps. This makes sense intuitively because we assume that objects rarely move rapidly from one frame to another.

$$\text{prediction_consistency_loss} = \sum_{t=0}^{T-1} \left(\hat{\mathbf{y}}_i^{(t)} - \hat{\mathbf{y}}_i^{(t+1)} \right)^2 \tag{4.6}$$

4.3.2 Inference

The recurrent neural network predicts output at every time-step. The network predicts 98 bounding boxes per video frame and class probabilities for each of the 49 grid cells. We note that for every cell, the net predicts class conditional probabilities for each one of the C categories and B bounding boxes. Each one of the B predicted bounding boxes per cell has an associated *objectness* confidence score. The predicted confidence score at that grid is the maximum among the boxes. The bounding box with the highest score becomes the *responsible* prediction for that grid cell i.

The product of class conditional probability $\hat{p}_i^{(t)}(c)$ for category type c and *objectness* confidence score $\hat{C}_i^{(t)}$ at grid cell i, if above a threshold, infers a detection. In order for an object of category type c to be detected for i-th cell at time-step t, both the class conditional probability $\hat{p}_i^{(t)}(c)$ and *objectness score* $\hat{C}_i^{(t)}$ must be reasonably high.

Additionally, we employ Non-Maximum Suppression (NMS) to winnow multiple high scoring bounding boxes around an object instance and produce a single detection for an instance. By virtue of YOLO-style prediction, NMS is not critical.

4.4 Results and Discussions

In this section, we empirically evaluate our model on the popular **Youtube-Objects** dataset, providing both quantitative results (as measured by mean Average Precision) and subjective evaluations of the model's performance, considering both successful predictions and failure cases.

The **Youtube-Objects** dataset[PKL+14] is composed of videos collected from Youtube by querying for the names of 10 object classes of the PASCAL VOC Challenge. It contains 155 videos in total and between 9 and 24 videos for each class. The duration of each video varies between 30 seconds and 3 minutes. However, only 6087 frames are annotated with 6975 bounding-box instances. The training and test split is provided.

Experimental Setup

We implement the domain-adaption of YOLO and the proposed RNN model using Theano [The16]. Our best performing RNN model uses two GRU layers of 150 hidden units each and dropout of probability 0.5 between layers, significantly outperforming domain-adapted YOLO alone. While we can only objectively evaluate prediction quality on the labeled frames, we present subjective evaluations on sequences.

Objective Evaluation

We compare our approach with other methods evaluated on the Youtube-Objects dataset. As shown in Table 4.1 and Table 4.2, Deformable Parts Model (DPM) [FMR08])-based detector reports [KFS15] mean average precision below 30, with especially poor performance in some categories such as *cat*. The method of Tripathi *et al.* (VPO) [TBHN16] uses consistent video object proposals followed by a domain-adapted AlexNet classifier (5 convolutional layer, 3 fully connected) [KSH12] in an R-CNN [GDDM14]-like

Table 4.1: Object detection with DPM, Video Object Proposal method, YOLO, domain-adapted YOLO. RNN-IOS regularizes on input-output similarity, to which RNN-WS adds category-level weak-supervision, to which RNN-PS adds a regularizer encouraging prediction smoothness.

	Average Precision on 10-categories									
Methods	airplane	bird	boat	car	cat	cow	dog	horse	mbike	train
DPM	28.42	48.14	25.50	48.99	1.69	19.24	15.84	35.10	31.61	39.58
VOP	29.77	28.82	35.34	41.00	33.7	57.56	34.42	54.52	29.77	29.23
YOLO	76.67	89.51	57.66	65.52	43.03	53.48	55.81	36.96	24.62	62.03
DA YOLO	**83.89**	**91.98**	59.91	81.95	46.67	56.78	53.49	42.53	32.31	67.09
RNN-IOS	82.78	89.51	68.02	**82.67**	47.88	70.33	52.33	61.52	27.69	**67.72**
RNN-WS	77.78	89.51	**69.40**	78.16	51.52	**78.39**	47.09	81.52	36.92	62.03
RNN-PS	76.11	87.65	62.16	80.69	**62.42**	78.02	**58.72**	**81.77**	**41.54**	58.23

framework, achieving mAP of 37.41. We also compare against YOLO (24 convolutional layers, 2 fully connected layers), which unifies the classification and localization tasks, and achieves mean Average Precision over 55.

In our method, we adapt YOLO to generate *pseudo-labels* for all video frames, feeding them as inputs to the refinement RNN. We choose YOLO as the *pseudo-labeler* because it is the most accurate among feasibly fast image-level detectors. The domain-adaptation improves YOLO's performance, achieving mAP of 61.66.

Our model with RNN-based prediction refinement, achieves superior aggregate mAP to all baselines. The RNN refinement model using both input-output similarity, category-level weak-supervision, and prediction smoothness performs best, achieving 68.73 mAP. This amounts to a relative improvement of 11.5% over the best baselines. Additionally, the RNN improves detection accuracy on most individual categories (Table 4.1).

While we don't compare against conditional random fields (CRFs) or hidden Markov models (HMMs), we point to the growing body of recent work demonstrating the superior predictive accuracy achieved by RNNs. On tasks ranging from video analysis to speech recognition, when ample data is available, RNNs yield state of the

art performance [AAB+15]. Notably [ZJRP+15] shows RNNs outperform stand-alone CRF-based approaches [KK13] for structured prediction on the task of semantic image segmentation.

Table 4.2: Overall detection results on Youtube-Objects dataset. Our best model (RNN-PS) provides 7% improvements over DA-YOLO baseline.

mean Average Precision on all categories							
Methods	DPM	VOP	YOLO	DA YOLO	RNN-IOS	RNN-WS	RNN-PS
mAP	29.41	37.41	56.53	**61.66**	65.04	67.23	**68.73**

Subjective Evaluation

We provide a subjective evaluation of the proposed RNN model in Figure 4.1. Top and bottom rows in every pair of sequences correspond to *pseudo-labels* and results from our approach respectively. While only the last frame in each sequence has associated ground truth, we can observe that the RNN produces more accurate and more consistent predictions across time frames. The predictions are consistent with respect to classification, localization and confidence scores.

In the first example, the RNN consistently detects the *dog* throughout the sequence, even though the *pseudo-labels* for the first two frames were wrong (*bird*). In the second example, *pseudo-labels* were *motorbike*, *person*, *bicycle* and even *none* at different time-steps. However, our approach consistently predicted *motorbike*. The third example shows that the RNN consistently predicts both of the cars while the *pseudo-labeler* detects only the smaller car in two frames within the sequence. The last two examples show how the RNN increases its confidence scores, bringing out the positive detection for *cat* and *car* respectively both of which fell below the detection threshold of the *pseudo-labeler*.

Areas For Improvement

The YOLO scheme for unifying classification and localization [RDGF16] imposes strong

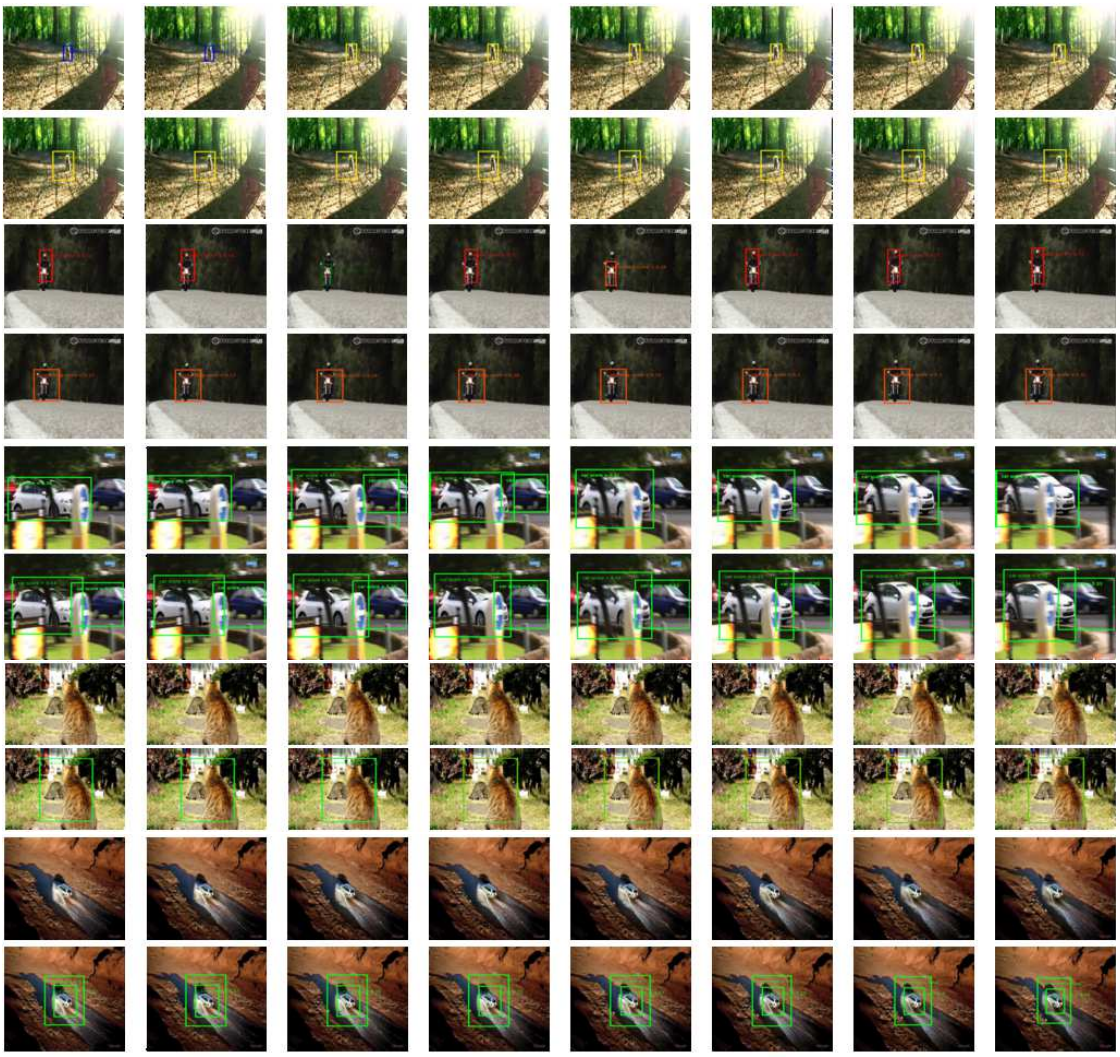


Figure 4.1: Results from the final eight frames of five different test-set sequences. Top and bottom row of each pair show the *pseudo-labeler* and RNN respectively. The RNN predicts correct categories, multiple instances, detects missing objects by increasing confidence score.

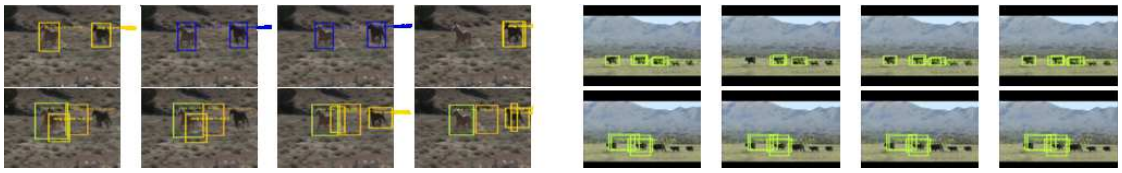


Figure 4.2: Failure cases for the proposed model. Left: the RNN cannot recover from incorrect *pseudo-labels*. Right: RNN localization performs worse than *pseudo-labels* possibly owing to multiple instances of the same object category.

spatial constraints on bounding box predictions since each grid cell can have only one class. This restricts the set of possible predictions, which may be undesirable in the case where many objects are in close proximity. Additionally, the rigidity of the YOLO model may present problems for the refinement RNN, which encourages smoothness of predictions across the sequence of frames. Consider, for example, an object which moves slightly but transits from one grid cell to another. Here smoothness of predictions seems undesirable.

Figure 4.2 shows some failure cases. In the first case, the *pseudo-labeler* classifies the instances as *dogs* and even as *birds* in two frames whereas the ground truth instances are *horses*. The RNN cannot recover from the incorrect pseudo-labels. Strangely, the model increases the confidence score marginally for a different wrong category *cow*. In the second case, possibly owing to motion and close proximity of multiple instances of the same object category, the RNN predicts the correct category but fails on localization. These point to future work to make the framework robust to motion.

The category-level weak supervision in the current scheme assumes the presence of all objects in nearby frames. While for short snippets of video this assumption generally holds, it may be violated in case of occlusions, or sudden arrival or departure of objects. In addition, our assumptions regarding the desirability of prediction smoothness can be violated in the case of rapidly moving objects.

4.5 Summary

We introduce a framework for refining object detection in video. Our approach extracts contextual information from neighboring frames, generating predictions with state of the art accuracy that are also temporally consistent. Importantly, our model benefits from context frames even when they lack ground truth annotations.

For the recurrent model, we demonstrate an efficient and effective training strategy that simultaneously employs localization-level strong supervision, category-level weak-supervision, and a penalty encouraging smoothness of predictions across adjacent frames. On a video dataset with sparse object-level annotation, our framework proves effective. A subjective analysis of failure cases suggests that the current approach may struggle most on cases when multiple rapidly moving objects are in close proximity. Likely, the sequential smoothness penalty is not optimal for such complex dynamics.

Our results point to several promising directions for future work. First, recent state of the art results for video classification show that longer sequences help in global inference. However, the use of longer sequences for localization remains unexplored. We also plan to explore methods to better model local motion information with the goal of improving localization of multiple objects in close proximity. In another promising direction, we would like to experiment with loss functions that incorporate specialized handling of classification and localization objectives.

Acknowledgments

Chapter 4 is in part based on the paper "Context Matters : Refining Object Detection in Video with Recurrent Neural Networks", by S. Tripathi, Z.C. Lipton, S. Belongie, and T. Nguyen [TLBN16]. The dissertation author was the primary investigator and author of this paper.

Chapter 5

Semantic Segmentation in Videos

5.1 Background

Deep convolutional neural networks (DCNNs) trained on a large number of images with pixel-level annotations or a combination of strongly labeled and weakly-labeled images have recently been the state-of-the-art in semantic image segmentation with significant performance improvement. However, due to the very invariance properties that make DCNNs good for high level tasks such as classification, visual delineation capacities for deep learning techniques are limited. Recent approaches address this problem with Conditional Random Field (CRF) based graphical model in two ways: either by (1) adding a post-processing step [CPK+14, PCMY15] of CRF-based probabilistic graphical model for the pixel-level classification or, (2) integrating the graphical model as a part of the CNN to make the end-to-end learning [ZJRP+15] with the usual back-propagation possible without the need of post-processing. In either case, the final pixel-level classification accuracy and efficiency remain highly dependent on the inference step of the image-based CRF [KK13] involved where fast approximate MPM inference is performed using cross bilateral filtering techniques within a mean-field approximation framework.

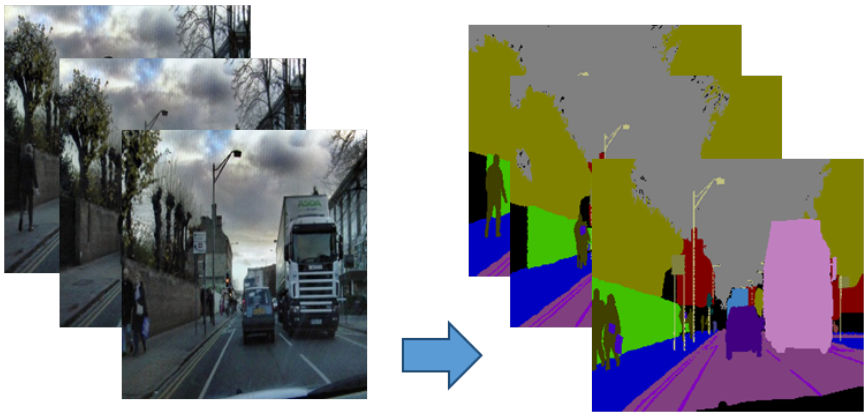


Figure 5.1: Video Semantic Segmentation involves semantic segmentation of multiple frames together.

Figure 5.1 shows the task of video semantic segmentation where inference is performed on a set of video frames together instead of taking inference at individual frames one at a time, thus improving temporal consistency.

Alvarez *et al.* [ASB14] demonstrates that performing inference on all test images at once in a dense CRF yields better results than inferring one image at a time without additional computation cost compared to performing segmentation sequentially on individual images. It is to be noted that the dense CRF [KK13] achieves good results with only unary and pairwise terms. This fully-connected pair-wise model is more expressive than its 4 or 8 connected random field counter-parts. Yet, it lacks the ability to handle high-order terms. Models [KLT09, RTK09, LSA$^+$10] using higher-order terms such as label consistency over large regions (pattern-based potentials) and relations of global co-occurrence potentials, are shown to be more expressive and effective for object class segmentation task. Filter-based inference for those higher-order terms is formulated in [VWT12] which enables significant speed-up compared to those graph-based methods [KLT09, RTK09, LSA$^+$10]. Yet, it needs to consider temporal consistency when applied in co-segmentation or video semantic segmentation.

We explore the efficiency of the CRF inference module beyond image level

semantic segmentation. The key idea is to combine best of two worlds - semantic co-labeling and exploiting more expressive models. Similar to [ASB14] our formulation enables us perform inference over ten thousand images within seconds. On the other hand, it can handle higher-order clique potentials similar to [VWT12] in terms of region-level label consistency and context in terms of co-occurrences. We follow the mean-eld updates for higher order potentials and extend the spatial smoothness and appearance kernels to address co-segmentation of multiple frames of a video; thus making the system amenable to perform video semantic segmentation most effectively.

5.2 Related Work

Deep convolutional neural networks (DCNNs) trained on a large number of images with pixel-level annotations or a combination of strongly labeled and weakly-labeled images have recently been the state-of-the-art in semantic image segmentation with significant performance improvement. However, due to the very invariance properties that make DCNNs good for high level tasks such as classification, visual delineation capacities for deep learning techniques are limited. Recent approaches address this problem with Conditional Random Field (CRF) based graphical model in two ways: either by (1) adding a post-processing step [CPK$^+$14, PCMY15] of CRF-based probabilistic graphical model for the pixel-level classification or, (2) integrating the graphical model as a part of the CNN to make the end-to-end learning [ZJRP$^+$15] with the usual back-propagation possible without the need of post-processing. In either case, the final pixel-level classification accuracy and efficiency remain highly dependent on the inference step of the image-based CRF [KK13] involved where fast approximate MPM inference is performed using cross bilateral filtering techniques within a mean-field approximation framework.

Alvarez *et al.* [ASB14] demonstrates that performing inference on all test images

at once in a dense CRF yields better results than inferring one image at a time without additional computation cost compared to performing segmentation sequentially on individual images. It is to be noted that the dense CRF [KK13] achieves good results with only unary and pairwise terms. This fully-connected pair-wise model is more expressive than its 4 or 8 connected random field counter-parts. Yet, it lacks the ability to handle high-order terms. Models [KLT09, RTK09, LSA$^+$10] using higher-order terms such as label consistency over large regions (pattern-based potentials) and relations of global co-occurrence potentials, are shown to be more expressive and effective for object class segmentation task. Filter-based inference for those higher-order terms is formulated in [VWT12] which enables significant speed-up compared to those graph-based methods [KLT09, RTK09, LSA$^+$10]. Yet, it needs to consider temporal consistency when applied in co-segmentation or video semantic segmentation.

We propose to use more expressive models such as pattern-based potentials in a co-segmentation framework, and show that with almost no additional time overhead a boost in video semantic segmentation performance can be achieved.

5.3 Methodology

Standard CRF Formulation : Common multiclass image segmentation and labeling use conditional random fields (CRF) defined over pixels or image regions. Basic CRF models consist of unary potentials on individual pixels or image patches and pair-wise potentials on neighboring pixels or patches. In dense model, all pairs of variables are directly connected by pair-wise potentials. Krähenbühl *et al.* [KK13] proposed an efficient fully connected CRF models with the complete set of pixels in an image. The inference method is based on mean field approximation and efficient message passing using high-dimensional filtering. The dense conditional random field is defined over a set

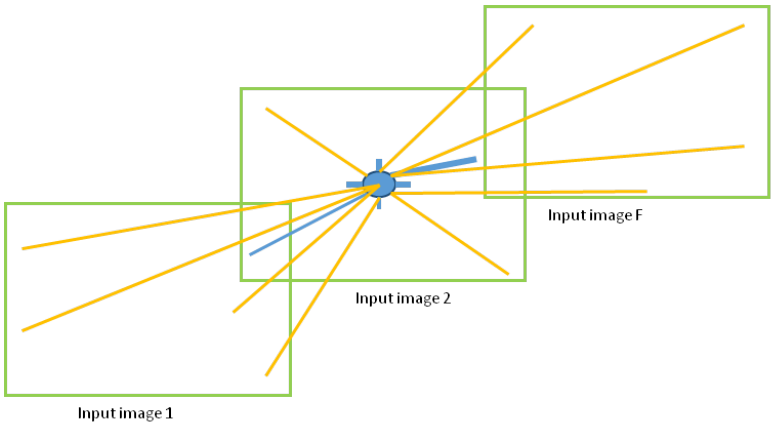


Figure 5.2: Framework for intra and inter-frame dense connectivity. Every pixel in every frame is connected to each other.

$\mathbf{X} = \{X_1, \ldots, X_N\}$ of variables conditioned on the image I and the model parameters Θ. The domain of each variable is a set $\mathbf{L} = \{l_1, \ldots, l_H\}$ of labels. Typically the inference problem solves the MAP of Gibbs energy of a label assignment.

Joint Labeling on Video Frames : In this work, we deal with multiple images $\mathbf{I}_1, \ldots, \mathbf{I}_M$ and the random field is defined over the set $Y = \{\mathbf{X}^1, \ldots, \mathbf{X}^M\}$, where superscript denotes index in image sequence.

MAP labeling can be found by computing the assignment that maximizes the distribution, P (Equation 5.1) or equivalently minimizing the energy function E (Equation 5.2).

$$P(\{X^1, \ldots, X^F\} | I^1, \ldots, I^F) \tag{5.1}$$

$$E(\{X^1, \ldots, X^F\} | I^1, \ldots, I^F) = \Sigma_{\substack{i \in F \\ j \in N}} \psi_u(x_j^i) + \Sigma_{\substack{i,i' \in F \\ j,j' \in N}} \psi_p(x_j^i, x_{j'}^{i'}) \tag{5.2}$$

where $F = \{1, ..., F\}$ and $N = \{1, ..., N\}$, $\psi_u(.)$ and $\psi_p(.,.)$ denote the unary and pairwise potentials respectively.

The unary potential function encodes the cost of assigning a specific label to a pixel. The intra and inter dense connectivity is graphically shown in Figure 5.2. At every iteration of the algorithm, the update for each variable involves summing over all the random variables. However, with a Gaussian kernel, the expensive summation becomes equivalent to performing high-dimensional Gaussian filtering [KK13] and thus remains completely tractable. Thus the pairwise potential functions take the form

$\psi_p(x_j^i, x_{j'}^{i'}) = \mu(x_j^i, x_{j'}^{i'}) \Sigma_{m=1}^{M} w_m k_m(\mathbf{v}_j^i, \mathbf{v}_{j'}^{i'})$, where $\mu(.,.)$ is the label compatibility function. This encodes a Potts model *i. e.* $\mu(x_j^i, x_{j'}^{i'}) = 1[x_j^i \neq x_{j'}^{i'}]$ The kernel $k_m(.,.)$ is a Gaussian Kernel computed over feature vector $\mathbf{v}_j^i$ that describes pixel j in image i. We utilize spatial smoothness and appearance kernels k_1 and k_2 as :

$$k_1(\mathbf{v}_j^i, \mathbf{v}_{j'}^{i'}) = \exp\left(-\frac{\|p_j^i - p^{i'}\|^2}{\sigma_p^2} - \frac{\|i - i'\|^2}{\sigma_f^2}\right) \tag{5.3}$$

$$k_2(\mathbf{v}_j^i, \mathbf{v}_{j'}^{i'}) = \exp\left(-\frac{\|p_j^i - p^{i'}\|^2}{\sigma_p^2} - \frac{\|i - i'\|^2}{\sigma_f^2} - \frac{\|I_j^i - I^{i'}\|^2}{\sigma_c^2}\right) \tag{5.4}$$

where p_j^i and I_j^i encode the image location and color vector of pixel j in image i. Setting σ_f to a low value will make the kernel k_1 vanish for any two random variables not belonging to the same image. Mean-field updates are calculated using ConCave-ConvexProcedure (CCCP) which is explained in [KK13]. Like [KK13], our approach approximates this operation using the permutohedral formulation. This performs convolution on a down-sampled version of the graph. Thus the overall cost becomes linear with the input size. The cost of performing inference in the CRF through our approach which exploits all images simultaneously is the same as the cost of performing inference sequentially on the individual images.

Higher-Order clique Potentials: The introduction of higher-order terms greatly expands the expressive power of such densely connected models. Vineet and Warrel

et al. [VWT12] show in their experiments that filter-based inference generally outperforms the best alternative methods in terms of speed and accuracy for some higher-order clique potentials. For high-order cliques potentials, the general energy takes the form $E(V|I) = \Sigma_{c \in C} \psi_c(v_c|I)$ A pattern-based potential is defined as:

$$\psi_c^{pat}(x_c) = \begin{cases} \gamma_{x_c} & x_c \in P_c \\ \gamma_{max} & \text{otherwise} \end{cases} \tag{5.5}$$

In general, $X \subseteq V$ when higher-order cliques (HOC) are taken into consideration, whereas without HOC, equation5.5 would have become similar to equation 5.2; where $X = V$. Among several higher-order potentials, the most useful ones for object detection task are pattern-based potential and co-occurrence potentials. For still images, patterns based potentials are coming from the different number of superpixels. Authors in [VWT12] demonstrates how the mean-field based updates works for these pattern-based potentials. We consider the use of patterns from either superpixels or the slices of supervoxels corresponding to different video frames, thus enforcing temporal consistency. Thus, inference in video can exploit intra and inter-frame connectivity and also respect patterns based potentials which are time consistent.

5.4 Results and Discussions

We evaluate our approach on standard benchmark dataset for multiclass segmentation: the Cambridge-driving Labeled Video dataset (CamVid) [BFC09]. CamVid consists of four image sequences with ground truth labels at 1fps that associate each pixel with one of 32 semantic classes. All the experiments were conducted using single threaded code on a standard Intel(R) Core(TM) i7-3770 CPU 3.40GHz desktop. In our current experiments, we used the TextonBoost [SWRC09] unary potentials for easy comparison with other

recent methods. Figure 5.3 shows some qualitative results of semantic segmentation in Camvid video dataset. Video-Level Dense-CRF [ASB14] shows improved temporal consistency over frame-level operation [KK13](previous row) without additional time overhead. For, pattern-based potentials, we use three different superpixel segmentations by varying parameters of the meanshift algorithm. Frame-level Dense-CRF with this P^n-Potts model [VWT12] almost achieves similar quality as of previous graph-cut based slow inference method [RTK09], but lacks temporal consistency.

Table 5.1: Per-class segmentation accuracy on CamVid dataset for different methods.

DenseCRF [KK13]	Co-segmentation [ASB14]	High-Order [VWT12]	Ours (High-order + co-segmentation)
47.9	48.8	55.3	56.8

The proposed video-level Dense-CRF with P^n-Potts model shows improved temporal consistency over the frame-level operation (previous row) without additional time-overhead. Video-Level dense CRF [ASB14] and the proposed method perform inference on 50 frames at once. On CamVid, with TextonBoost unaries our proposed method achieves 8% more accuracy than [ASB14] by virtue of P^n-Potts model and 1.5% more accuracy over [VWT12] without additional time overhead by virtue of co-labeling (Table 5.1). Average per-class accuracy is computed as the average over all classes of the ratio of correctly classified pixels in a class to the total number of pixels in that class.

5.5 Summary

We explore the efficiency of the CRF inference module beyond image level semantic segmentation. The key idea is to combine best of two worlds - semantic co-labeling and exploiting more expressive models. We follow the mean-eld updates for higher order potentials and extend the spatial smoothness and appearance kernels to

address co-segmentation of multiple frames of a video; thus making the system amenable to perform video semantic segmentation most effectively.

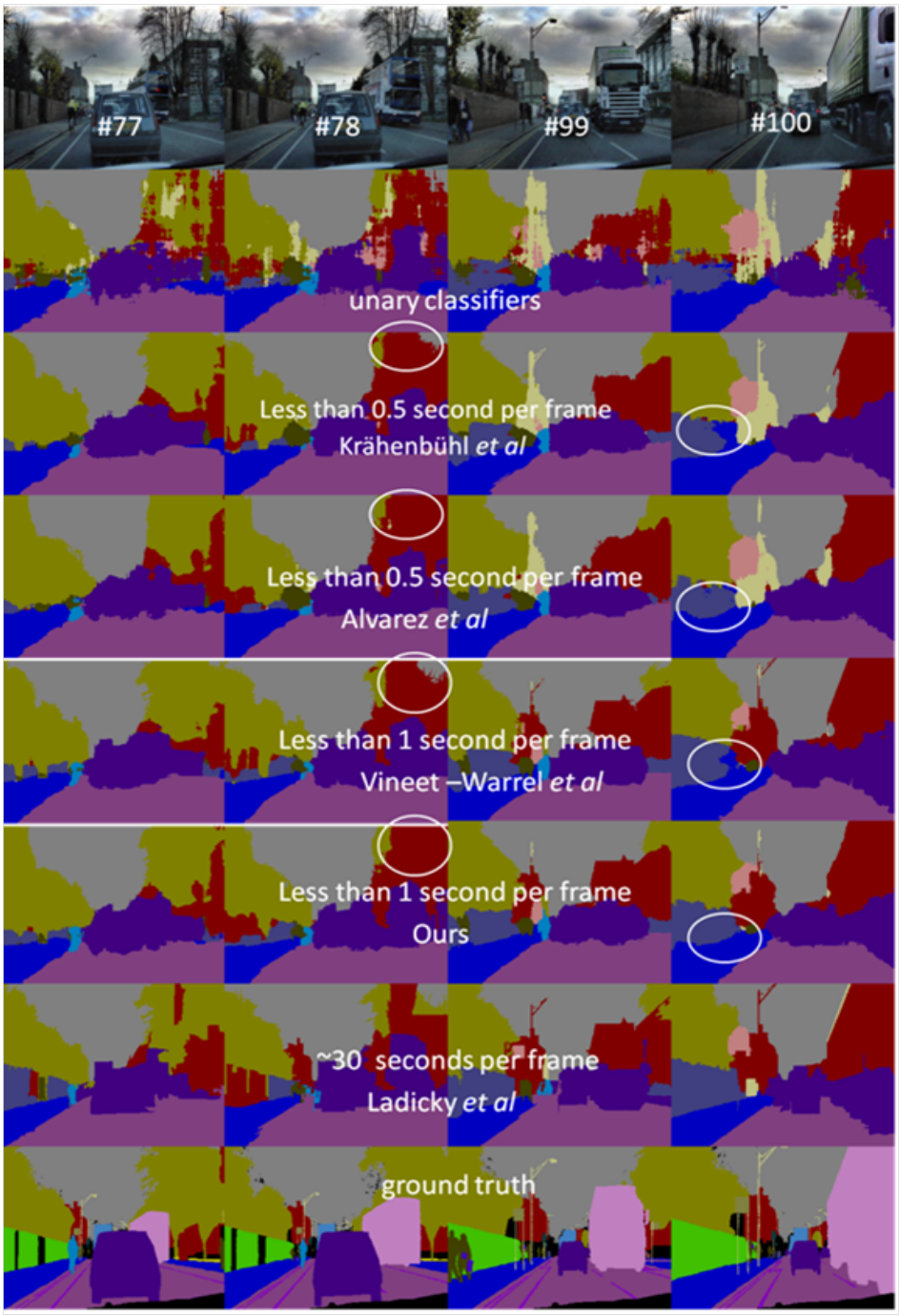


Figure 5.3: Qualitative results on Camvid. Input frames, TextonBoost classifier scores; Dense-CRF [KK13]; Video-Level Dense-CRF [ASB14]; Dense-CRF with P^n-Potts [VWT12]; proposed video-level Dense-CRF with P^n-Potts; graph-cut inference; and Ground truth.

Acknowledgments

Chapter 5 is in part based on the paper “Semantic Video Segmentation: Exploring Inference Efficiency”, by S. Tripathi, Y.B. Hwang, S. Belongie, and T. Nguyen [THBN15]. The dissertation author was the primary investigator and author of this paper.

Chapter 6

Person Instance Segmentation using Human-Pose

6.1 Background

Instance segmentation, in particular person instance segmentation, is a promising research frontier for a range of applications such as human-robot interaction, sports performance analysis, and action recognition. Separating instances that share similar local appearances is highly challenging. Deep convolutional neural networks are the current state-of-the-art methods for the task of instance level segmentation. For example, the entrants to the 2016 COCO segmentation challenge [WGM] achieve excellent performance on instance segmentation for the 80 object categories considered on the COCO dataset. Although these methods work extremely well for *any* category of objects, there is a potential for human-specific domain knowledge to boost person segmentation performance.

In this chapter, we investigate the importance of human keypoints as a prior for the task of instance-level person segmentation. With the availability of image datasets

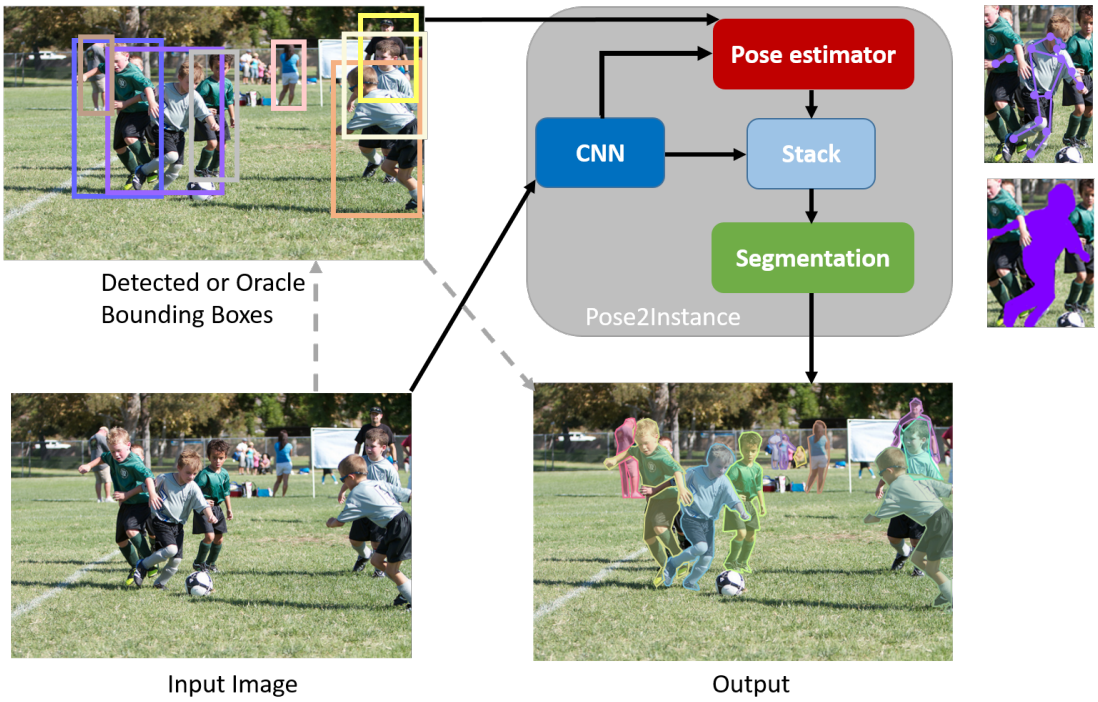


Figure 6.1: Pose2Instance segmentation model incorporates a *learnable* component conditioned on the human "pose". The model generates keypoint heatmaps and segmentations at instance level and uses the keypoints heatmaps output as an additional input to the segmentation.

that include both segmentation masks and keypoint annotations, we consider a methodical approach to quantify the importance of keypoints for people instance segmentation. We explore what happens if an oracle provides all the keypoints, or only bounding boxes, and how people instance segmentation can be improved respectively.

In order to evaluate the segmentation conditioned on human pose, we consider all instances of people from the COCO segmentation dataset [LMB+14a] where the instances also have keypoints ground truth. We do not include COCO person instances that are marked as *crowd*. By comparing the image and instance identifiers from the COCO segmentation and COCO keypoints dataset, we see there exists $45,174$ images in the training dataset and $21,634$ images in the validation dataset. This amounts to $185,316$ and $88,153$ ground truth person instances with both segmentation and keypoints annotations in the training and validation split respectively. This work refers to this intersection between COCO instance segmentation dataset and COCO person keypoints dataset as the COCO dataset throughout this chapter.

We first explore a human pose prior represented as the distance transform of a skeleton and show how this prior can directly yield instance-level human segmentation when combined with existing semantic segmentation model such as DeepLab [CPK+15] trained for human segmentation. This analysis also validates the idea of combining two existing different models, one for pixel-level person segmentation (non-instance) and another for detecting keypoints, for improving instance segmentation.

Next, we propose an approach to directly generate the per-pixel probability of person instances conditioned on human poses using a deep convolutional neural network (CNN). We call this pose-conditioned segmentation model *Pose2Instance*. Figure 6.1 outlines the approach. Person instance bounding boxes are either provided by an oracle or they come from a person detector. The model is trained for generating keypoint heatmaps and segmentation at instance level by sharing *CNN* parameters up to the penultimate

layer and using the keypoints heatmap as an additional input channel for the segmentation output.

To summarize, we first show that the human pose prior represented as the distance transform of the human skeleton yields significant performance gain for the person instance segmentation during inference without any training. Next, we show how the *learned* segmentation can be conditioned on the keypoints by learning additional parameters specifically for mapping shape to segmentation while training a DCNN *jointly* for keypoints and segmentation. Finally, we perform extensive empirical investigation of the proposed Pose2Instance method on the intersection of COCO instance segmentation and COCO keypoints dataset. We show the effectiveness of the pose conditioned deep instance segmentation model by qualitative and quantitative analysis.

6.2 Related Work

Our work builds upon a rich literature in both semantic segmentation using convolutional neural networks and joint pose-segmentation modeling.

Semantic and Instance Segmentation:
DeepLab [CPK+15] and FCN [LSD15] achieved significant breakthroughs for the challenging task of semantic segmentation using deep convolutional neural networks. Subsequently, a set of instance segmentation methods [LWS+16, ZFU16, LHM16, LWS+15, DHL+16, RT16, PB16, AT16] were proposed, which begin with pixel-wise semantic segmentation and generate instance-level segmentation from them. Recently, [LQD+] achieved the state-of-the-art performance on the 80-category instance segmentation using a fully convolutional end-to-end solution. Except [LHM16], none of these methods look into learning implicit or explicit shapes of different object categories.

Human Pose Estimation:

Human pose estimation from static images [RPZ13, KRBT08, EF10, LZBC15] or videos [HVRPE12] with hand-crafted features and explicit modeling gained considerable interest in the last decade. Human pose estimation using an articulated grammar model is proposed in [RPZ13]. Hernández-Vela *et al.* [HVRPE12] proposed Spatio-Temporal GrabCut-based human segmentation that combines tracking and segmentation with hand-crafted initialization. In [EF10], Eichner and Ferrari proposed a multi-person pose estimator framework that extends pictorial structures for explicitly modeling interaction between people. A detailed review on pose estimation literature survey is available in [LZBC15].

Recently, convolutional neural networks have been successfully applied for pose estimation from videos [LKR16], human body parts segmentation [OVB$^+$16], and multi-person pose estimation [PIT$^+$16, IPA$^+$16, CWSS, KPT$^+$]. Additionally, among the most accurate results are those shown by *chained prediction* [GTJ16].

Joint Pose Estimation and Segmentation:

The most closely related works to this one are those that also seek to jointly estimate human pose and segmentation in static images or videos [KRBT08, LHHHH13, ASSL13, SASL15]. Kohli *et al.* [KRBT08] proposed *PoseCut*, a conditional random field (CRF) framework to tackle segmentation and pose estimation together for *one* person. The CRF model explicitly combines hand crafted image features and a prior on shape and pose in a Bayesian framework. The prior is represented by the distance transform of a human *skeleton*. The *inference* in PoseCut finds the MAP solution of the energy of the pose-specific CRF by doing optimization over different configurations of the latent shape prior. With a good initialization the inference step requires 50 seconds per frame. Similar inference strategies for deep models are computationally prohibitive.

Among other significant efforts towards joint pixel-wise segmentation and pose

estimation of multiple people, Alahari and Seguin *et. al.* [ASSL13, SASL15] use additional motion and disparity cues from stereo videos. The appearance and disparity cues are generated using HOG features. The pose estimation model [ASSL13] is represented as a set of parts, where a part refers to a patch centered on a body-joint or on an interpolated point on a line connecting two joints. They learn up to eight mixture components for each part and an articulated pose mask for the mixture components.

Very recently, Mask R-CNN [HGDG17] reports person segmentation and pose-estimation as two separate branches. Their powerful back-end architecture and ROIAlign technique proves to be beneficial for *segmentation-only* model. But their multi-task model improves the pose-estimation only at the cost of segmentation precision.

We propose a different and effective framework for incorporating pose prior into deep segmentation models. The proposed DCNN model consists of additional parameters that are trained/optimized specifically for the mapping of shape to segmentation. The Pose2Instance *inference* does not require optimization such as finding the MAP solution. The prediction task involves only *one* forward pass through the trained network.

6.3 Methodology

Our Pose2Instance approach looks at the problem of incorporating a pose prior into segmentation in two ways. We begin with a constrained environment study where the keypoints are provided by an oracle, and we investigate a way to improve the instance segmentation inference given a state-of-the-art pixel-level person classifier [CPK$^+$15]. Next, we move to a more realistic case where oracle keypoints are not available and propose a framework to train segmentation model directly while benefiting from a pose estimator.

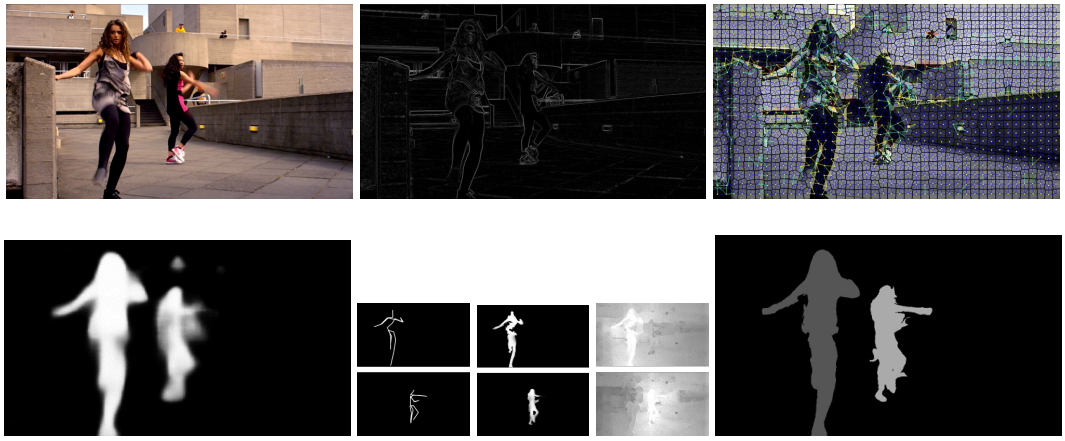


Figure 6.2: Instance segmentation with oracle skeleton. Top: input image; Sobel edges; RAG with edge strength from Sobel responses. Bottom: DeepLab *person* segmentation; Oracle skeleton, distance transform on RAG, *pose-maps* and instance segmentation output.

6.3.1 Pose2Instance Inference

We first present Pose2Instance within a constrained environment that assumes that the keypoints are provided by an oracle. This allows us to investigate the contribution of the pose prior independent of the other components of the whole system. In the COCO dataset, 17 person keypoints along with their corresponding visibility flags are annotated. We will handle these as part of a *skeleton* that links joint keypoints by the corresponding body parts.

In the investigation of the prior alone, with oracle keypoints, we address the inference stage of instance segmentation without any training. The sole task-specific training is done on the already existing DeepLab [CPK$^+$15] network. In section 6.3.1 below, we first fine-tuned this network for person-specific segmentation on COCO, with other labels discarded. This model directly predicts per-pixel probability of the *person* class label for the whole image. We call this model *DeepLab-people* in this paper.

Person Instances from Oracle Keypoints

We use the notion of a distance transform of the person skeleton [KRBT08], generated from the oracle keypoints, as a prior for the instance segmentation task. For this proof of concept, we follow the below steps.

We create a Region Adjacency Graph (RAG), $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ where the nodes $\mathbf{V}$ are superpixels and the weights of the edges $\mathbf{E}$ between the nodes depend on the strength of image edges. We obtain superpixels using SLIC [ASS+12], and the image edge responses using Sobel operator. We can define the pose prior as a distribution over the labels of this graph. Given a superpixel $p \in \mathbf{V}$, we can compute a conditional probability it belongs to a given instance. For each instance, we color those nodes where the corresponding superpixel contains a part of the human skeleton line that is generated from the oracle keypoints where visibility flags are marked as *valid*. The colored nodes in the RAG represent a foreground binary mask, and are assigned the highest probability of belonging to this instance. For each such binary mask corresponding to each person, we apply distance transform in the RAG using the Floyd-Warshall [Hou10] shortest paths algorithm. A point-wise softmax of this distance transform then represents the likelihood of each person's gross shape. We call this shape-likelihood the *pose-instance map*. For an image of height h and width w, with n oracle instances, the shape of pose-instance map is $h \times w \times n$. Figure 6.2 shows these intermediate steps of generating the RAG, its nodes and the weights of its edges, the oracle skeletons and the instance segmentations.

Instance-level to Image-level inference

Element-wise multiplication of this pose-instance map and the DeepLab-people score generates instance heatmap of size $h \times w \times n$. Here, h, w and n denote the height and width of the image, and the oracle-provided number of person instances respectively.



Figure 6.3: Pose2Instance inference with oracle skeleton. Top row: An image from Inria dataset containing 9 persons; Instance classification; Bottom row: DeepLab-people score, and the *Pose2Instance* inference output generated by fusing DeepLab-people and *pose-instance map*.

An argmax on instance heatmap produces the final instance segmentation on the image.

Figure 6.3 shows intermediate results for the inference step in this constrained setup. There are 9 persons in this image. Combining DeepLab-people score with pose-instance map improves the instance segmentation quality over the pose-instance map. Quantitative results in section 6.4.1 show that person keypoints represented as the distance transform can be an excellent source of additional domain knowledge for improving people instance segmentation.

Person Instances from Oracle Bounding Boxes

As a baseline, we take the approach of snipping the pixel-level DeepLab-people score at oracle bounding boxes for the COCO validation images. Though this bounding box approach does not comply with the relative depth ordering or visibility of one instance over another, the method still can be used as a reasonable baseline to compare the

performance of Pose2Instance inference.

We performed similar experiments with fast-sweeping [WDB$^+$08] based distance transform on pixel grid for reducing complexity using single-pixel width skeleton as the binary mask. However, the distance transform produces worse result compared with the specified non-grid RAG approach.

6.3.2 Pose2Instance Learning

After this proof of concept in the inference stage in a controlled setup with oracle keypoints, we move to a more realistic scenario where ground truth keypoints annotations are unavailable and we strive for learning a segmentation model by jointly optimizing for the segmentation and pose.

Our proposed network has a DeepLab-style architecture [CPK$^+$15]. This is a modified VGG network [SZ14] that uses atrous convolution with hole filling [CPK$^+$15] and replaces fully-connected layers by fully-convolutional layers. The baseline model is a 2-class *DeepLab-people* model. To construct this model, we start with the publicly available Deeplab model trained on the PASCAL VOC dataset, and fine-tune it for predicting only people on the COCO training instances. The second and third exploratory architectures involve two output layers each, a segmentation output and a 17-channel heatmap for pose estimation output. The first among them *Pose and Seg* is a multitask model, where the two parallel output layers share the parameters up to previous convolutional layers. The 2-class segmentation layer and the 17-class pose estimation output layers use cross-entropy loss after softmax and sigmoid activations respectively. The later one, *Pose2Seg*, is a cascaded model, where the 17-channel keypoints heatmap is followed by an 1×1 convolution to generate the shape likelihood. Segmentation feature maps from the last layer is combined with the above shape likelihood, and the softmax

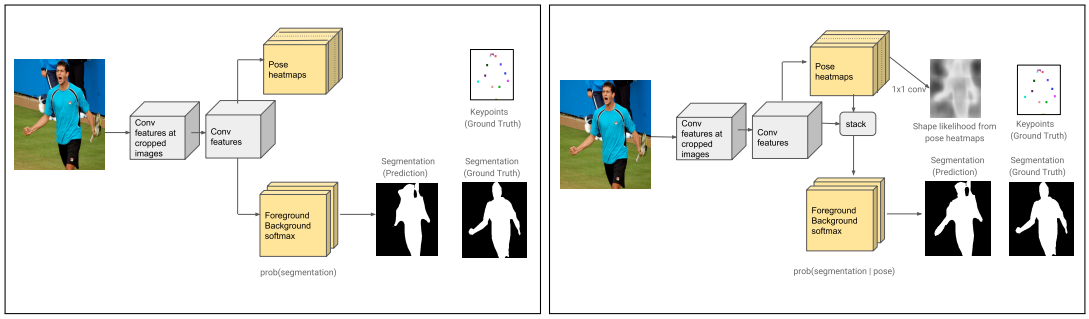


Figure 6.4: Architectures for joint pose - segmentation learning. Left: Multitask model where pose estimation and segmentation are two parallel output paths. Right: Cascaded Model where pose dedicated parameters are *learned* for mapping pose to segmentation.

segmentation is trained. Comparing with the segmentation only model, the cascaded model has only 18 extra parameters for learning the 1×1 convolutional kernels. 17 parameters are for the key-points heatmaps, and 1 for shape likelihood.

Figure 6.4 shows the two above mentioned architectures. The *stack* operation is a 1×1 convolution on the estimated 17-channel pose heatmap. Its output can be used as the gross shape-likelihood of a person based on the estimated keypoints. In the cascaded model, the segmentation output is directly conditioned on the pose heatmap. As we also see in Fig 6.9, the 1×1 convolution on the pose heatmap preserves the general notion of shape of a person from its keypoints, the segmentation model thus can be thought of conditioned on the latent shape of a person.

In the Pose2Instance framework, we try to improve the segmentation accuracy from both keypoints and segmentation supervision. In particular, a model learned with one supervisory signal *pose-estimation* acts as a prior to the model learned with another supervisory signal *segmentation*. Different sources of supervision have proven to be useful for learning segmentation. For example, ScribbleSup [LDJ+16] performs semantic segmentation from additional scribble based supervision broadly in grabcut [CR04] like framework. In [BRFFF16], Bearman *et al.* discussed various levels of supervisions such as pixel-level strong supervision, and sparse point-level supervision for semantic

segmentation. Our method is substantially different from these since none of the above specifically addresses (instance) segmentation problem with one as a prior to the other.

6.4 Results and Discussions

We implement the Pose2Instance model using TensorFlow-Slim. We train the model on specified COCO training instances. We initialize the model from DeepLab-people and continue training for $20,0000$ iterations using stochastic gradient descent with mini-batch size of 16 and momentum 0.9.

6.4.1 Pose2Instance in a Constrained Setup

In order to analyze the Pose2Instance inference with oracle keypoints, we use COCO validation images. While table 6.1 shows the performance of Pose2Instance inference with oracle keypoints, figure 6.6 shows some qualitative results comparing with the oracle bounding box baselines. The figures show that for overlapping person instances, the proposed pose prior significantly outperforms a baseline using the bounding box as an ad-hoc prior.

The inference stage which consists of combining the existing semantic segmentation model and the oracle keypoints outperforms the oracle bounding box case by 10% relative improvement. FAIRCNN[ZLL$^+$16] and CUHK[QSL$^+$15] are the instance segmentation models that also use VGG as the base network. We include their instance segmentation results only on *'person'* category from COCO detection challenge Leaderboard as references. Newer models from the Leaderboard use more powerful ResNet in their backend, so are not directly comparable.

Figure 6.7 shows qualitative results of instance segmentation on COCO validation

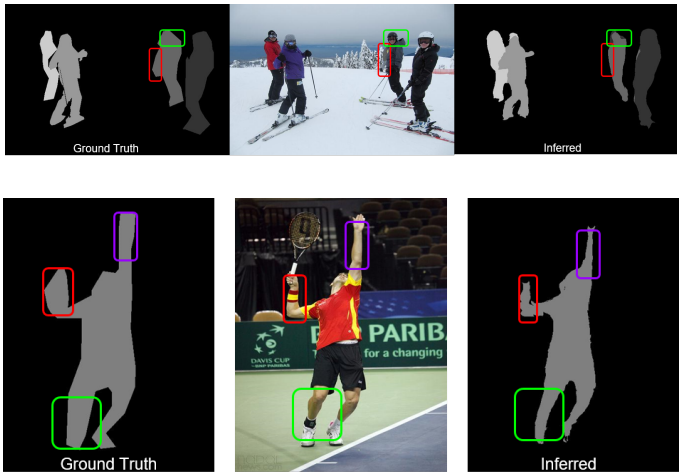


Figure 6.5: From left to right: Ground truth instance segmentations; Corresponding image from COCO keypoints dataset; Pose2Instance inference with oracle keypoints. Colored boxes show errors in segmentation ground truth that are corrected using our keypoint conditioned model.

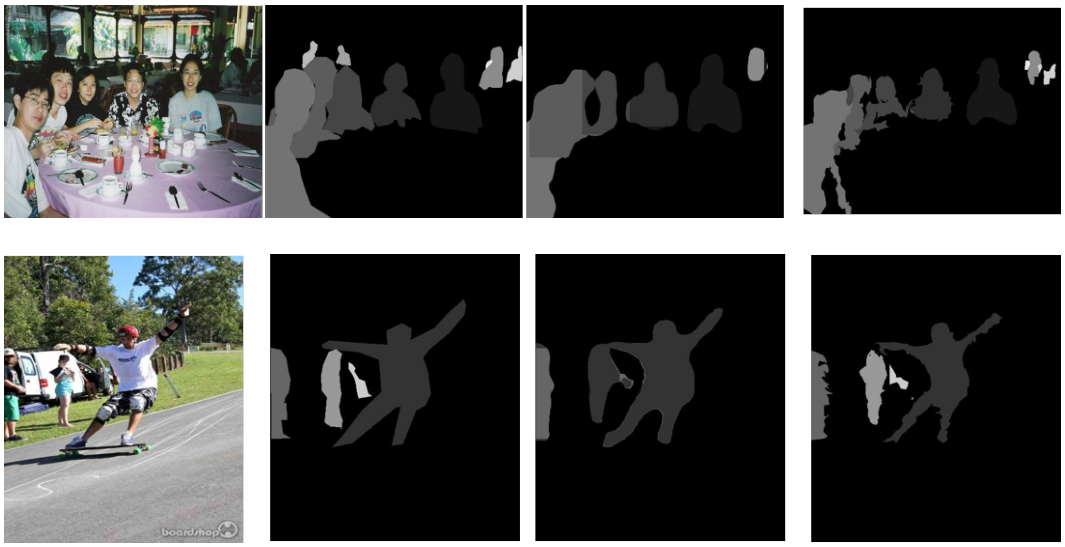


Figure 6.6: Pose2Instance with oracle bounding boxes vs oracle keypoints. From left to right: A frame from COCO keypoints dataset; Ground truth instance segmentations; Baseline instance segmentation from oracle bounding boxes and Pose2Instance inference from oracle keypoints.

Table 6.1: Oracle keypoints provides 10% to 12% relative improvement over oracle bounding box at various IOU thresholds atop DeepLab-people segmentation model. Results of FAIRCNN[ZLL$^+$16] and CUHK[QSL$^+$15] that also use VGG as the base network are from COCO Leaderboard.

Instance Segmentations on COCO validation Images		
Methods	AP^r0.5 IoU=0.5	AP^r IoU=[0.5 to 0.9]
DeepLab+Oracle BB	0.437	0.252
DeepLab+Oracle keypoints	**0.533**	**0.283**
FAIRCNN[ZLL$^+$16]	0.504	0.206
CUHK[QSL$^+$15]	0.478	0.214



Figure 6.7: Pose2Instance in a constrained setup. Top: DeepLab person segmentation. Bottom:Pose2Instance inference from oracle keypoints on COCO evaluation dataset. (best viewed in color)

dataset in such constrained environments. We note that this method only applies during the inference step with oracle keypoints and does not involve any training. Human keypoints ground truth is easier to collect than precise segmentation ground truth. Figure 6.5 shows how the errors in the segmentation ground truth can be corrected with our pose-conditioned segmentation model.

6.4.2 Pose2Instance in Realistic Environments

After validating the effectiveness of the inference with keypoint-specific distance transform, we evaluate the proposed Pose2Instance model on COCO validation instances

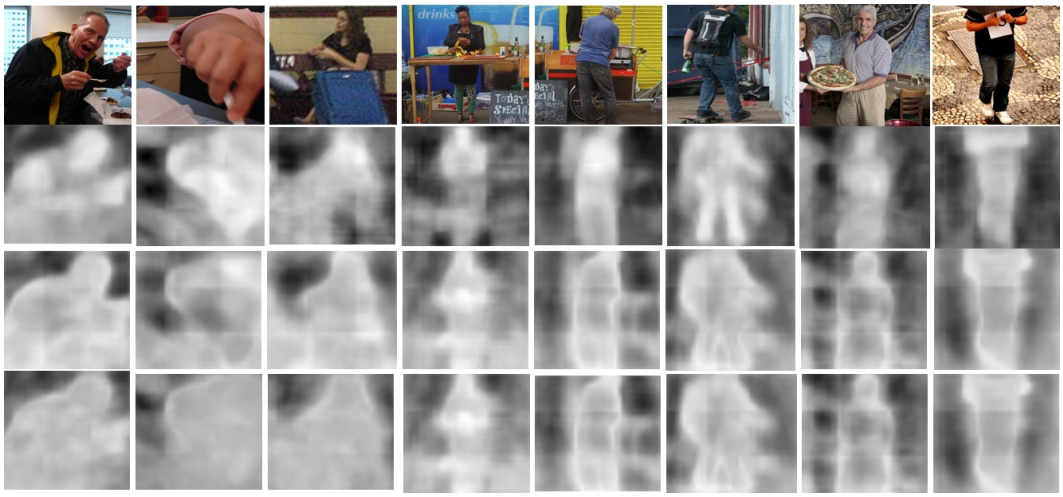


Figure 6.8: Visualization of shape likelihood from pose estimation. Instances from COCO validation set; intermediate latent shape likelihood for (i) *Pose Only model*; (ii) *Multitask model* and (iii) *Pose2Instance model* respectively. joint models learn to capture the overall contour.

in a more realistic environment where oracle keypoints are unavailable. We assume the availability of oracle bounding boxes. The model estimates the keypoints and segmentations at all instances.

Table 6.2: Segmentation accuracy on COCO validation instances. Pose2Instance model achieves higher accuracy over the *multitask* model that outperforms *segmentation only* model. Overall, relative improvement from *segmentation only* model is 3.8% to 10.5% at various IOU thresholds.

Instance Segmentations without Oracle Keypoints

Methods	AP^r IoU=0.5	AP^r IoU=[0.5 to 0.95]
DeepLab Seg only	0.79	0.38
Multitask: Pose and Seg	0.80	0.40
Cascaded: Pose2Seg	**0.82**	**0.42**

In Table 6.2, we show the comparative segmentation performance evaluation for the proposed Pose2Instance method without oracle keypoints. Average Precision at 0.5 IOU improves by 3% over the *segmentation only* model and 2% over the multitask model.



Figure 6.9: Pose2Instance without oracle keypoints. Top row: Instance bounding boxes of COCO validation images. Middle row: Ground truth segmentation. Bottom row: predicted segmentation masks for the instance bounding boxes.

The corresponding improvements for the $[0.5, 0.9]$ IOU are 4% and 2% respectively.

In terms of relative improvements, at 0.5 IoU and $[0.5, 0.9]$ IoU, the pose-conditioned segmentation model improves the AP^r by 3.8% and 10.5% over the *segmentation only* model [Table 6.2] respectively. This demonstrates the proof-of-concept of how to incorporate pose prior effectively into deep segmentation model.

Figure 6.9 shows qualitative results on some challenging examples. These rectangular regions contain one or more partial person instances in addition to the primary person instance. We see that the Pose2Instance model learns to produce instance segmentation only for the intended one. The last two figures are examples of most difficult cases with many people in close proximity, and the Pose2Instance predictions are far from being ideal due to the current limitation of the VGG-based pose-estimator output.

In this work, we assess the effectiveness of pose conditioned segmentation performance, and did not evaluate the parallel key-points estimation output. We performed

some qualitative analysis for the pose estimation output from the described multitask and cascaded models. Additionally, we implemented another vanilla pose estimator model with the same network except the segmentation output. We call this model a *PoseOnly* model which is optimized only for 17-class pose-estimation problem. Our subjective analysis of the latent shape likelihood of person include *PoseOnly*, *multitask* and *cascaded* models. Figure 6.8 shows some visualizations of the latent shape likelihood on some COCO validation images.

6.5 Summary

Our experiments suggest that human pose is a useful domain knowledge even atop state-of-the-art deep person segmentation models. We show that in a constrained environment with oracle keypoints, at various IOU thresholds, the instance segmentation accuracy achieves 10% to 12% relative improvement over a strong baseline with oracle bounding boxes without any training. In a more realistic environment, without the oracle keypoints, the proposed Pose2Instance deep model achieves relatively 3.8% to 10.5% higher segmentation accuracy than the strongest baseline of a deep network trained only for segmentation.

Our proposed method is applicable to *any* such architecture that shares the necessary properties of the Deeplab model. Models optimized for the segmentation task, including the one covered in our experiments and future better-performing segmentation models, could potentially incorporate the same methodology to utilize pose information.

While at present we show results on images, it is that likely similar dynamics are embedded in videos. Human keypoints ground truth is easier to collect than precise segmentation masks. Thus, a pose conditioned segmentation model can be more powerful for person instance segmentation for natural scenes where people tend to appear in

groups, have dynamic interactions, and partial occlusions. This work represents a first step towards embedding pose into segmentation in complex scenes. An exploratory follow-up work can include investigation on incorporating keypoints based dynamic person model into video segmentation.

Acknowledgments

This work has been done in part during an internship in Google Research and Machine Intelligence. Chapter 6 is in part based on the paper "Pose2Instance: Harnessing Keypoints for Person Instance Segmentation", by S. Tripathi, M. Collins, M. Brown, and S. Belongie [TCBB17]. The dissertation author was the primary investigator and author of this paper.

Chapter 7

Object Detection for Embedded Systems

7.1 Background

Deep Convolutional Neural Network(CNN) based models are the current state-of-the-art for object detection. The best methods for object detection aim to increase the accuracy on standard datasets. They run on powerful GPUs that dissipate a huge amount of power. On the other hand, embedded processors and DSPs are great low-power solutions where the instruction sets benefit from fixed-point operations. For practical deployment of object detector on mobile devices, we need low-complexity CNN models that can run on embedded processors. The algorithms need to leverage the fixed point operations without compromising accuracy.

In this chapter, we propose *LCDet*, a low-complexity object detector to address the above issues. We design and develop an end-to-end TensorFlow-based fully-convolutional deep neural network for object detection inspired by YOLO [RDGF16]. We replace the last two fully connected layers by fully-convolutional layers. For simplicity, we call

YOLO's last two fully-connected layers as YLDet and the last two convolutional layers of the proposed *LCDet* as ConvDet. The backend of the CNN architecture is similar to YOLO except *LCDet* can work on any input image resolution by virtue of being fully-convolutional.

We choose face detection as a use-case due to its many practical applications in mobile phones, although the algorithm is generic enough for any number of classes. The detection pipeline of our TensorFlow-Slim based network requires a single forward pass through the network. Evaluation results for the face detection performance on publicly available FDDB [JLM10] dataset show that the proposed method achieves comparative accuracy with respect to state-of-the-art CNN-based face detection methods while reducing the model size by $3\times$ and memory-BW by $3-4\times$ comparing with YOLO [RDGF16], one of the fastest DCN-based object detector. Additionally, we quantize the model by $8-$bit precision, which leads to additional $4\times$ memory reduction with almost no loss in detection accuracy. The $8-$bit quantization is one of the most important steps for a deployment in fixed-point architectures such as DSPs or dedicated convolution accelerators. We report $8-$bit quantization on TensorFlow models for object detection task that heavily uses regression. It is understood that 8-bit quantization of floating point models that are trained for regression is more prone to accuracy drop comparing with the models that are trained for classification. Experimental results show that the object detection accuracy with quantized model reports less than $1-2\%$ loss comparing with the floating point model and achieve $20\times$ performance gain in terms of frame rate per second comparing with the floating-point model.

7.2 Related Work

CNN-based Object Detection: Several papers propose ways of using deep convolutional networks for detecting objects [GDDM14, Gir15, RHGS15, SREA14, SREA14, SEZ+13, YYLL16, GK15, AWI+16]. Some approaches classify the proposal regions [GDDM14, Gir15] into object categories and some other recent methods [RHGS15, RDGF16, LAE+15] unify the localization and classification stages. For detailed prior-art on object detectors and their speed-accuracy trade-off can be found in [HRS+16].

Precisely, the single stage detection pipeline of YOLO [RDGF16] is extremely fast. YOLO is the first reported real-time CNN-based object detector model that runs with high-end GPUs. Its performance accuracy on PASCAL VOC [EGW+10] dataset is comparable with state-of-the-art methods. Unlike YOLO, our model is fully-convolutional. Thus it is highly memory-efficient, computationally more effective and not restricted by input image resolution.

CNN-based Object Detection for Embedded Systems: The most accurate and best performing CNN-based models require high-end GPUs. There is a growing interest for developing specific hardware design [Mod, CES16] including FPGAs, DSPs, custom vision chips and embedded GPUs for energy efficient CNN-based object-detection. Detailed algorithmic advancements and case-studies for CNN-algorithms for embedded systems can be found in [QWY+16, MYT+16].

The best-performing CNN-based object detection methods which run on real-time (on high-end GPUs), falls significantly short on embedded GPUs. For example, a supplier in surveillance camera market (protecting anonymity for the purpose of reviewing) found out that even after replacing the back-end of YOLO from GoogleNet to a simpler CNN

such as AlexNet, it's embedded implementation runs at most 5 frames per second. This motivates us to investigate on fully-convolutional low complexity object detector that can run real-time on embedded platforms.

TensorFlow is open source and used by many developers to create their own AI-applications. Recently [Sb17] announced Snapdragon 835 [Qua17] includes TensorFlow-optimized Hexagon 682 DSP. This DSP architecture and others in this family are designed to process certain features more quickly and at lower power than a CPU or GPU. Our proposed TensorFlow-based model exploits the advantages of similar architecture and is useful for real-time object detection tasks on these platforms.

CNN-based Face Detection: We choose face detection as an application and evaluate the proposed object detector *LCDet* for this task. As per the FDDB evaluation server [JLM10], the state-of-the-art face detection methods are based on convolutional neural networks [YLLT15, LSWW16, YJW$^+$16, WCZ$^+$16, JL16, SWH17, QYLH16]. Yang *et al.* presented a neural network which combines feature responses regarding facial parts [YLLT15]. Li *et al.* presented an integrated method of neural networks and 3D face model [LSWW16]. Yu *et al.* modified VGG-16 networks, and also proposed intersection over union (IoU) loss layer [YJW$^+$16]. Recently presented works are based on faster R-CNN [RHGS15] [WCZ$^+$16, JL16, SWH17]. Our model achieves comparable quality with Faster-RCNN based methods. Additionally, our method meet all the requirements for real-time embedded applications while the above other CNN-based face detection methods can not achieve real-time performance in embedded platforms.

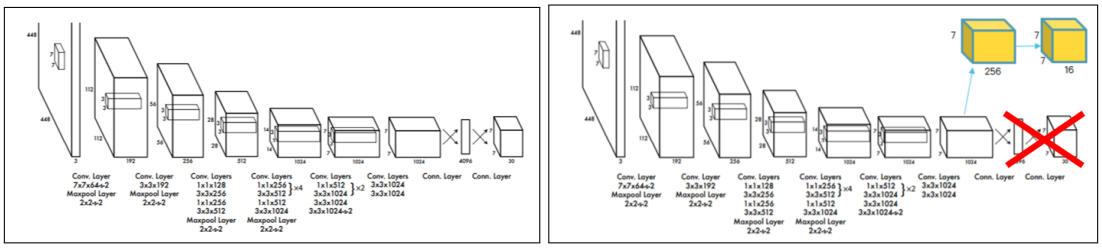


Figure 7.1: YOLO vs proposed network. Output channel's value 16 corresponds to 1 class-conditional and $1+4$ confidence and coordinates for each of the $B = 3$ boxes.

7.3 Methodology

7.3.1 Network Architecture

Our proposed model is inspired by YOLO [RDGF16] which adopts a single-pass detection pipeline combining bounding box localization and classification by a single network with the differences as outlined in Figure 7.1. The last two fully-connected layers of YOLO are replaced by fully-convolutional layers in the proposed model. Let's suppose, the convolutional layer right before the first fully-connected layer in YOLO is called the final feature map of spatial size $W \times H$. Here, W_f and H_f denote the number of grid centers along the horizontal and vertical axes.

From the same feature layer, the proposed model uses the final layer as a convolutional layer that outputs a grid ($W \times H \times Channels$) as shown in Figure 7.1. Each grid center is associated with C class probabilities, 1 confidence score, and 4 scalar values of coordinates for each of the $B = 3$ possible bounding boxes. Similar to YOLO, the confidence score is the predictor for Intersection-over-Union with the ground truth bounding box. Finally, we employ Non-Maximum suppression (NMS) for keeping top bounding boxes. During inference, the detection pipeline consists of a single forward pass through the network.

7.3.2 Training Methodology

Unlike Faster R-CNN [RHGS15], which deploys a 4-step alternating training strategy to train Region Proposal Network (RPN) and detector network, our detection network can be trained end-to-end, similarly to YOLO [RDGF16]. We apply a multi-part object detection loss as described in (equation 7.1) similar to YOLO.

$$\begin{aligned}\text{loss} = \lambda_{coord}(&\sum_{i=0}^{S^2}\sum_{j=0}^{K}\mathbb{1}_{ij}^{obj}\left(x_i-\hat{x}_i\right)^2+\left(y_i-\hat{y}_i\right)^2+\sum_{i=0}^{S^2}\sum_{j=0}^{K}\mathbb{1}_{ij}^{obj}\left(\sqrt{w_i}-\sqrt{\hat{w}_i}\right)^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2)+\\ &\sum_{i=0}^{S^2}\sum_{j=0}^{K}\mathbb{1}_{ij}^{obj}(C_i^{(T)}-\hat{C}_i{}^2+\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{K}\mathbb{1}_{ij}^{noobj}\left(C_i-\hat{C}_i\right)^2+\sum_{i=0}^{S^2}\mathbb{1}_{i}^{obj}\sum_{c\in classes}\left(p_i(c)-\hat{p}_i(c)\right)^2\end{aligned} \tag{7.1}$$

where $\mathbb{1}_i^{obj}$ denotes if the object appears in cell i and $\mathbb{1}_{ij}^{obj}$ denotes that jth bounding box predictor in cell i is *responsible* for that prediction. The loss function penalizes classification and localization error differently based on presence or absence of an object in that grid cell. x_i, y_i, w_i, h_i corresponds to the ground truth bounding box center coordinates, width and height for objects in grid cell (if it exists) and $\hat{x}_i, \hat{y}_i, \hat{w}_i, \hat{h}_i$ stand for the corresponding predictions. C_i and $\hat{C}_i$ denote confidence score of *objectness* at grid cell i for ground truth and prediction. $p_i(c)$ and $\hat{p}_i(c)$ stand for conditional probability for object class c at cell index i for ground truth and prediction respectively. We use similar settings for YOLO's object detection loss minimization and use values of $\lambda_{coord} = 5$ and $\lambda_{noobj} = 0.5$.

We additionally apply sigmoid activation on the prediction of confidence score. Confidence score should be in [0,1] as it is ideally the IOU predictor. We employ the softmax on class prediction. However, for the special case of single-class detection in YOLO-style, we employ sigmoid activation on 1 class prediction output from the network.

The proposed model uses 448×448 frames as input while training and regresses on category types and locations of possible objects at each one of $S \times S$ non-overlapping grid cells. For each grid cell, the model outputs class conditional probabilities as well as K bounding boxes and their associated confidence scores. As in YOLO, we consider a *responsible* bounding box for a grid cell to be the one among the K boxes for which the predicted area and the ground truth area shares the maximum Intersection Over Union. During training, we simultaneously optimize classification and localization error (equation 7.1). For each grid cell, we minimize the localization error for the *responsible* bounding box with respect to the ground truth only when an object appears in that cell.

7.3.3 Detection-Specific Layers

From the feature layer of size $W_f \times H_f \times Ch_f$, YOLO [RDGF16] employs two fully-connected layers. For simplicity, we denote these two layers together as YLDet. We denote the last two convolutional layers of the proposed model by ConvDet.

Table 7.1: Comparison between RPN, ConvDet and YLDet. RP stands for Region Proposals, *cls* denotes classification.

	RP	**cls**	**# Parameters**
RPN	✓	✗	$Ch_f K(5+C)$
YLDet	✓	✓	$F_{fc1}(W_f H_f Ch_f + W_o H_o(C+5K))$
ConvDet	✓	✓	$F_w F_h Ch_{d1}(Ch_f + (C+5K))$

In YOLO [RDGF16], input feature map size is $7 \times 7 \times 1024$. $F_{fc1} = 4096$, $C = 20$, $W_0 = H_0 = 7$. Thus the number of parameters in the last two connected layers is about 269×10^6. The first convolutional layer in the detection specific net in the ConvDet has $Ch_{d1} = 256$. If we use the same feature map sizes, number of output grid centers, ConvDet only requires 2.3×10^6 parameters, which is $115\times$ less than YLDet.

7.3.4 Quantized Model

Often times, DSPs or dedicated convolution accelerators operate on fixed point instruction set. There exists literature on fixed point models for embedded systems [WLW+15, Gys16] for classification task. It is well-known that 8−bit models [Tenb] perform as good as the floating point model specially for classification [VSM11]. The justification for the high accuracy in low-precision modes comes from the fact that the final activation is a probability *i. e.* in [0,1] intervals, can be represented with an unsigned number without any concern on scaling. However, we are not aware of many published reports on the study of quantization for detection (regression) task.

We convert the 32-bit floating point models to 8-bit fixed point model via [War, Tena]. Although literature exists for training with low precision mode [CBD14], we perform quantization only for the inference. Since training is performed off-line, it is reasonable and more practical to quantize the trained model for inference.

To quantize the 32-bit floating point to the 8-bit fixed point, we first store minimum and maximum value at a layer. Then, we quantize the relative value to the linearly distributed closest integer in [0,255].

$$w_q = \left[255 \frac{w_f - w_{min}}{w_{max} - w_{min}}\right] \tag{7.2}$$

where w_q, w_f, w_{min}, and w_{max} represents quantized variable, variable in floating point, minimum, and maximum. $[\cdot]$ represents rounding to the closest integer.

We see from our experimental results that regression models (such as our LCDet) also works well for low-precision inference. Quantized LCDet is as good as the floating-point model in terms of detection accuracy for the face detection application in general.

7.4 Results and Discussions

In this section, we present the performance of the proposed algorithm with floating point model as well as the associated 8-bit quantization on commonly used face databases such as FDDB benchmark [JLM10] and the Widerface [YLLT16] validation split. We also provide an analysis of the proposed method's performance in terms of speed, complexity and memory requirements. Although, face detection is studied as a use case here, the network does not use any face-specific processing such as facial parts or attributes. LCDet is a general purpose object detector.

We use transfer learning by first converting the weights for detecting 20-category PASCAL objects [EGW$^+$10] from DarkNet [Red17] library to a TensorFlow checkpoint, called *YOLO PASCAL TF*. For face detection with baseline YOLO [RDGF16], we first restore parameters from the *YOLO PASCAL TF* checkpoint for all except the last layer. We then fine-tune it for the 1-class (only face) object detection task. For LCDet, we restore parameters for all but last two layers from the *YOLO PASCAL TF*. We then initialize the last two layers with random weights and train the proposed LCDet for single-class object detection.

We adopt data augmentation techniques such as scaling and object-centric cropping for reducing overfitting. In our experiments, we use initial learning rate of 10^{-5}, the minibatch size of 32, and $8Kepochs$. We use Adam [KB14] optimizer for training. We used NVIDIA Tesla K40 GPUs for training and testing experiments. We also trained similar models with batch normalization for the convolutional layers. The training appeared to converge early, but that didn't yield any improvement in the detection accuracy. In order to minimize the number of parameters, we go with models without batch normalization.

7.4.1 Dataset

FDDB [JLM10] is a benchmark dataset for face detection in unconstrained settings. It contains $2,845$ images with a total of $5,171$ faces. The dataset provides fixed partitioning of 10 folds. WIDER FACE is a larger dataset [YLLT16] for face detection. It consists of $32,203$ images with $393,703$ faces. The dataset is organized in 61 event classes. For each event class, 40%, 10%, and 50% of data were selected for training, validation, and testing. The ground truth for test split is not disclosed. In our experiments, we scaled each image to a pre-determined size. For gray-scale images, we duplicate the single channel three times to make them images with three channels.

7.4.2 Detection Accuracy

We first evaluate LCDet for two different nonlinearities such as Leaky ReLU and ReLU at each convolutional layers. Although the model has been initialized from YOLO PASCAL TF which was trained with Leaky ReLU activations. After finetuning for face detection task with two different models with two different activations, we find ReLU activation performs better than leaky ReLU. The performance of LCDet with LeakyReLU vs ReLU has been shown in Figure 7.2. Next we evaluate the performances of LCDet and its 8-bit quantized model on FDDB dataset. The models are trained on FDDB images. This allows us to investigate the performance gap of floating and fixed point models independent of the other components in the whole system.

Figure 7.4 shows the performance of the floating point model and the 8-bit quantized model by the TP-FP curve with discrete score evaluation method per [JLM10]. Solid curve denotes the performance at standard detection crietria *i. e.* 50% Intersection over Union (IoU) with ground truth. Dotted lines denote less strict but practical detection criteria *i. e.* 40% IoU with ground truth boxes. Regressed coordinate locations from the

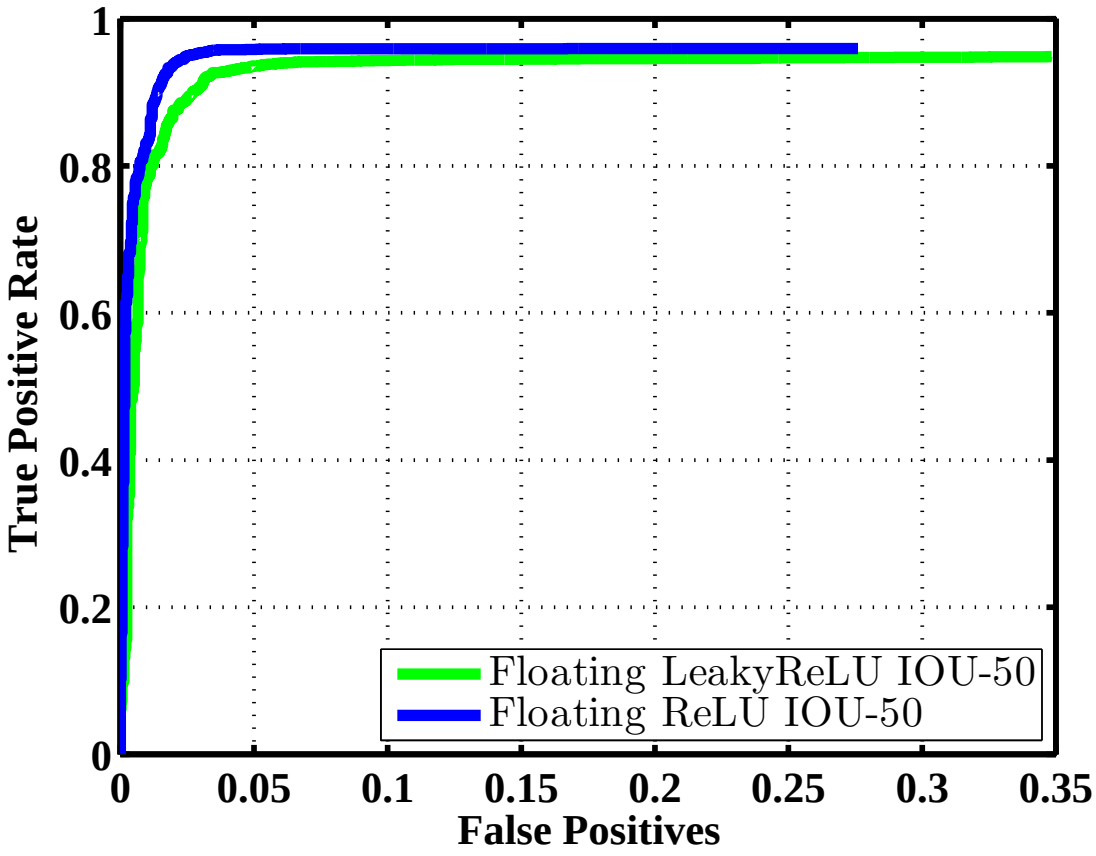


Figure 7.2: Performance of LCDet on FDDB for LeakyReLU vs ReLU model with discrete core metric.

fixed point model suffer from higher deviation from the floating point model prediction. This effect becomes more evident when we increase the IoU criteria for detection. As we see in Figure 7.3, for more relaxed IoU criteria such as 40% IoU, floating vs fixed point model exhibit similar detection performances. However, for stricter detection criteria such as 60% IoU or higher, the performance of the fixed-point model appears to drop significantly, especially for the lower false positive regions on the TP-FP curves.

Mobile devices use face detection for several face based quality enhancement processing such as auto-exposure or auto-focus. Any false detection should be highly penalized as their consequences are more expensive. On the other hand, if the detected box overlaps with the actual face by little less than 50%, certain end use cases that use

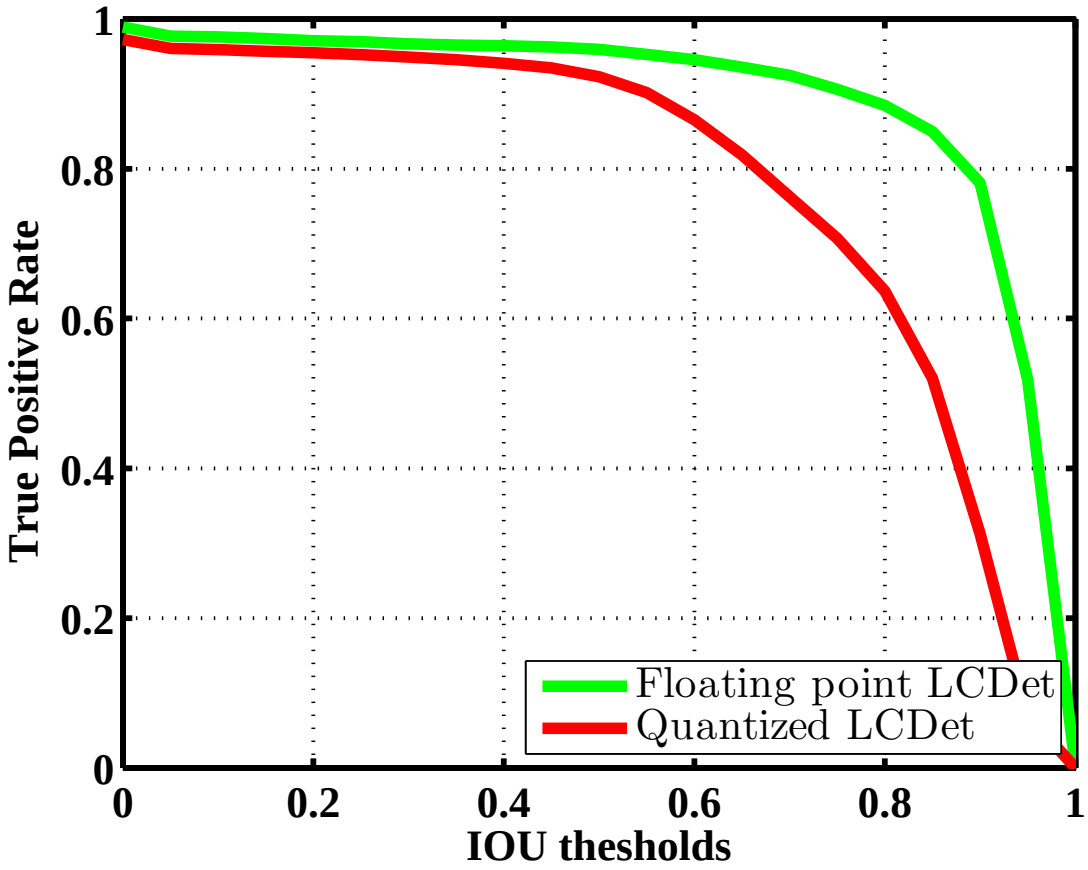


Figure 7.3: Performance comparison between floating and fixed point models (ReLU) at different IoU thresholds on FDDB.

face detection bounding box as input can still function with similar performance. In less strict IoU operating region, LCDet fixed point model is regarded as good as the the LCDet floating point model. Figure 7.3 shows that the detection rate in those operating points (upto 45% IoU) is similar for floating and fixed LCDet models.

Next, we train LCDet on the Widerface dataset. The Widerface has about 20× more faces than FDDB. some of the faces are extremely small. We evaluate the performance of LCDet on the Widerface validation dataset using the provided evaluation toolbox. One of the current limitation of the YOLO-style training is that it assumes at most one ground truth object at each grid location, although it can predict up to k objects per grid. For a training image of size 448×448 means 7×7 grids, thus can exploit only

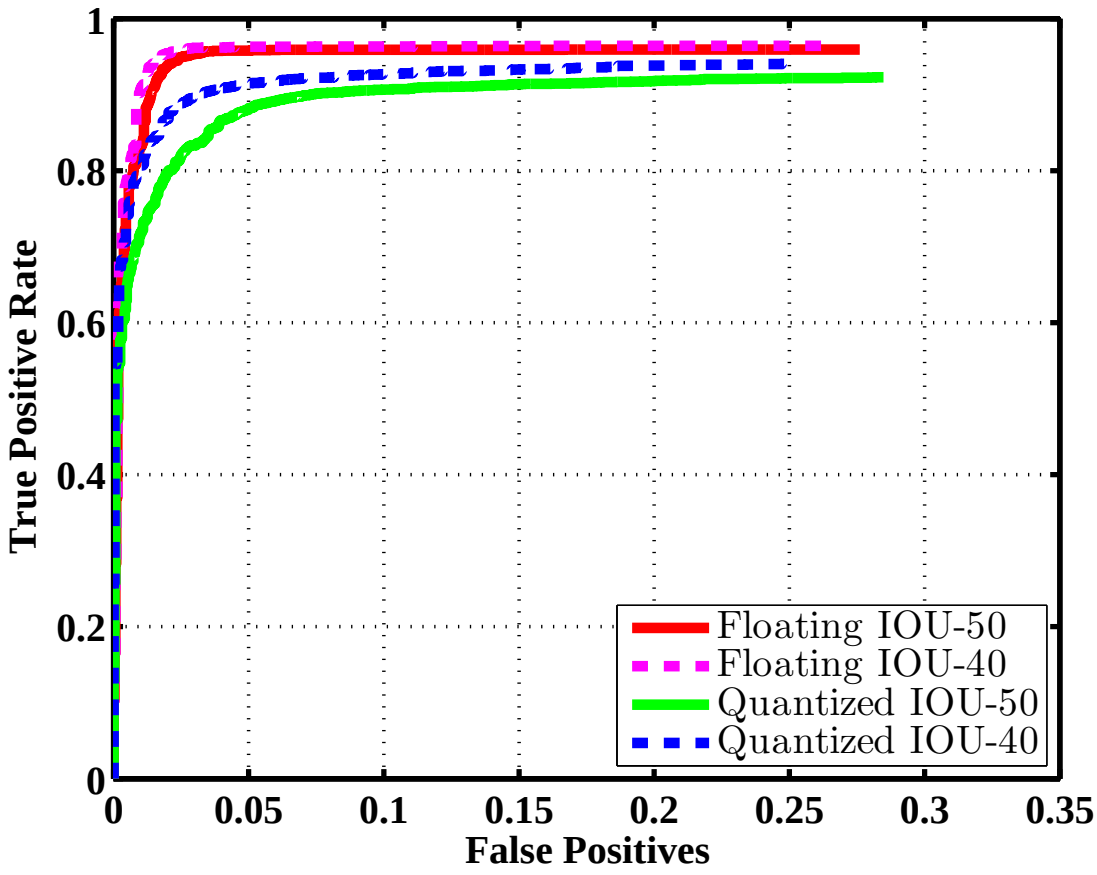


Figure 7.4: Performance with varying IOU criteria. The fixed point model achieves 5% improvement in true positives when IOU going down from 50% to 40%. Quantization loss for box coordinate regression is high. Floating point model is relatively resilient to IOU.

49 ground truth objects. On the other hand, the Widerface has more than 100 ground truth faces in at least 200 training images. YOLO-style training could not use all the ground truth available. whereas, faster-RCNN type network can use all possible ground truth objects. As shown in Figure 7.5, the model needs improvement for the localization accuracies specially for small objects mostly present in Widerface. As the IoU criteria is getting relaxed, the model approaches comparable or even better accuracy than other state-of-the-methods.

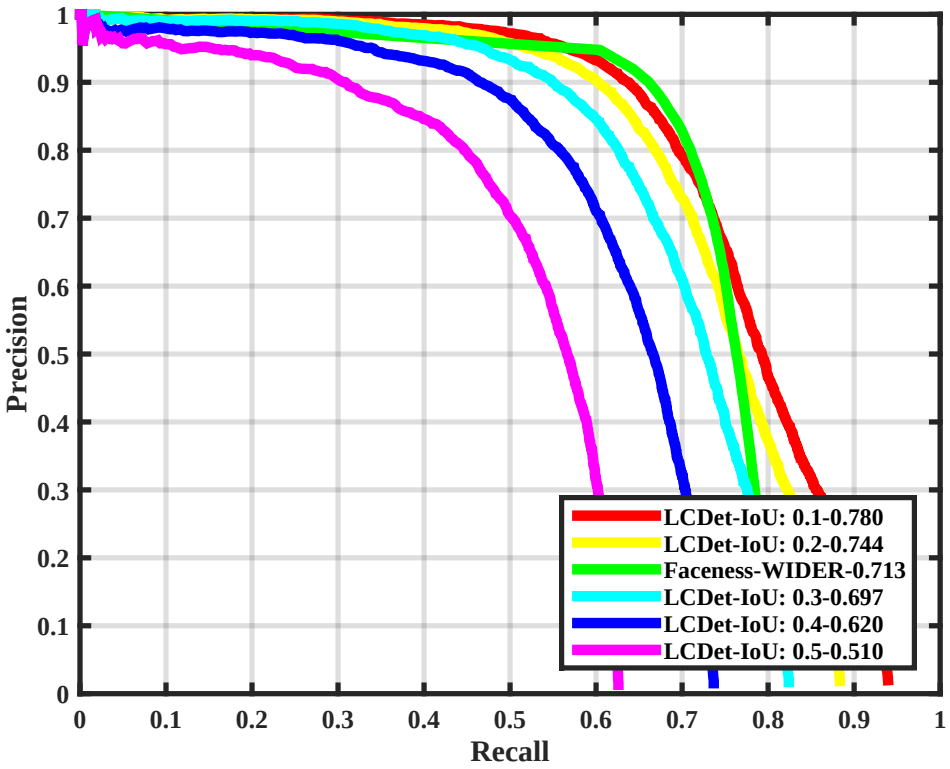


Figure 7.5: Precision-Recall performance on Widerface Validation set with relaxed IoU criteria.

7.4.3 Complexity and Memory-BandWidth Analysis

We first convert the darknet [Red17] YOLO implementation to TensorFLow-Slim based implementation. We leveraged the weights from the darknet [Red17] for all of the 24 convolutional layers and first fully connected layer. The number of output nodes in the final detection layer is $W_f \times H_f \times (C + K \times 5)$. For face detection task, $C = 1$, and we use $K = 3$ for all our experiments. Then using the mentioned training methodology, we fine-tune all layers for the face detection task. Table 7.2 demonstrates the performance and accuracy of these models along with some of the other recent models on powerful GPUs.

The detection-specific module of LCDet uses two convolutional layers. The first one has 4096 kernels of size 3×3 and the second one has 16 kernels of 1×1 size each.

Table 7.2: Inference speed of LCDet vs others on NVidia Tesla K40. YOLO* is our TF-based face detector. Max TP is the highest true positive rate value achieved in FDDB. SSD and Faster R-CNN running time use TF-SSD [B.] and TF Faster RCNN [Cha] respectively.

	Model Size (MB)	**FLOPs** $\times 10^9$	**Inference Speed (FPS)**	**Max TP**
LCDet	250	20	17.55	93.0
YOLO*	1126	20.1	15.44	85.0
Faster-RCNN+VGG16	485	98	4.61 [Cha]	96.1 [JL16]
SSD300+VGG16[LAE$^+$15, B.]	105	31.6	17.97[B.]	NA

Table 7.3: Performance analysis of Fixed-point LCDet in terms of OPs, Memory Activation Footprints

	Model Size (MB)	**OPs** $\times 10^9$	**Activation Memory Footprint (MB)**
Quantized YOLO	281	20.1	333
Quantized LCDet	62	20.0	88

Irrespective the backend feature-extractor network, LCDet has 115$\times$ fewer parameters for only the detection part comparing with YOLO as described in sec 7.3.3.

Next, we analyze and compare the performance of proposed method with respect to the state-of-the-art deep neural network-based object detector simulated on a commercially available Snapdragon platform such as in [Qua17]. Hexagon DSP includes fixed-point vector extensions which make it an attractive computing unit for computer vision applications and provides performance per power compared to CPU and GPUs on mobile platforms. We quantized our baseline YOLO model and also quantized the proposed LCDet model, and compare their relative model sizes and activation memory

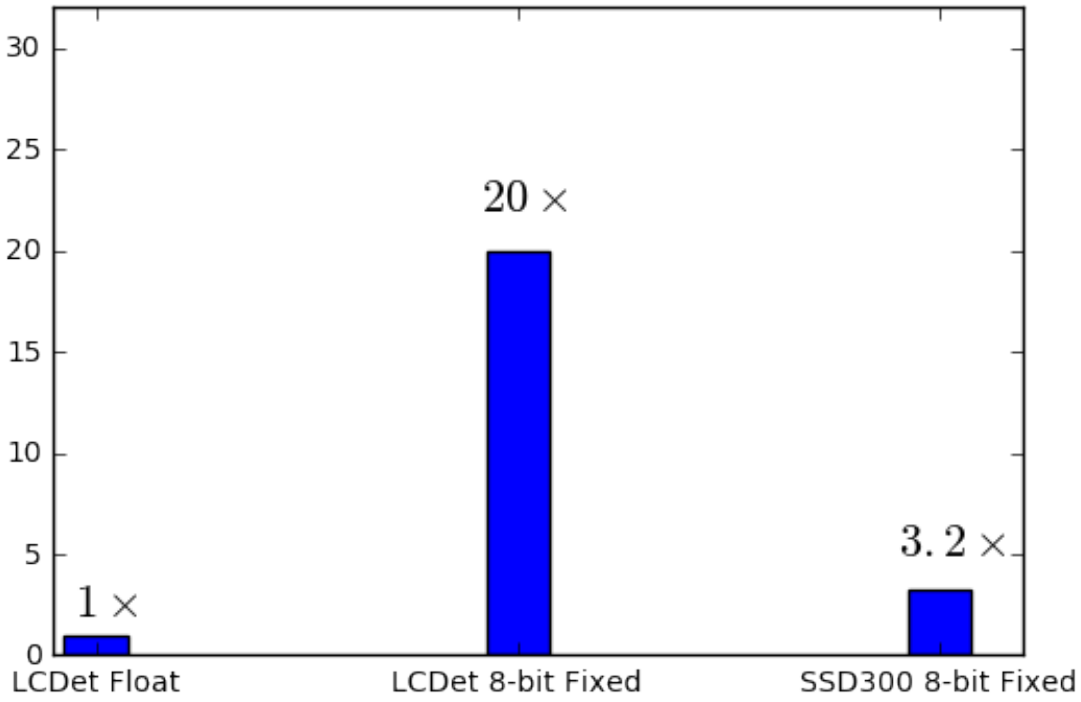


Figure 7.6: Frame Rate Improvement for Fixed Point Model. Fixed-Point LCDet also outperforms fixed-point SSD300.

footprints for the same input resolution size as shown in Table 7.3. In Figure 7.6, we compare the achievable frame rate of the following methods: LCDet-float and LCDet-8bit-fixed, SSD-300-8bit-fixed. In a fixed-point implementation, activations and weights are implemented in 8-bits and mapped to vector extension of DSP, whereas in float implementation vector extension is not utilized. By bringing down the model size and bandwidth (BW) per layer, we achieve close to the 20x increase in frame rate with respect to floating point implementation. The average DDR bandwidth that our quantized model requires is roughly 1 Gbps, whereas the instantaneous BW has a wider range reaching close to 20 Gbps for some layers as shown in Figure 7.7. Typically, the DDR BW is throttled when multiple applications are run on the embedded systems and frame rate degrades because of stringent BW constraints. This is depicted in Figure 7.8, where the frame rate drops as DDR BW decreases from 6 Gbps to 1 Gbps.

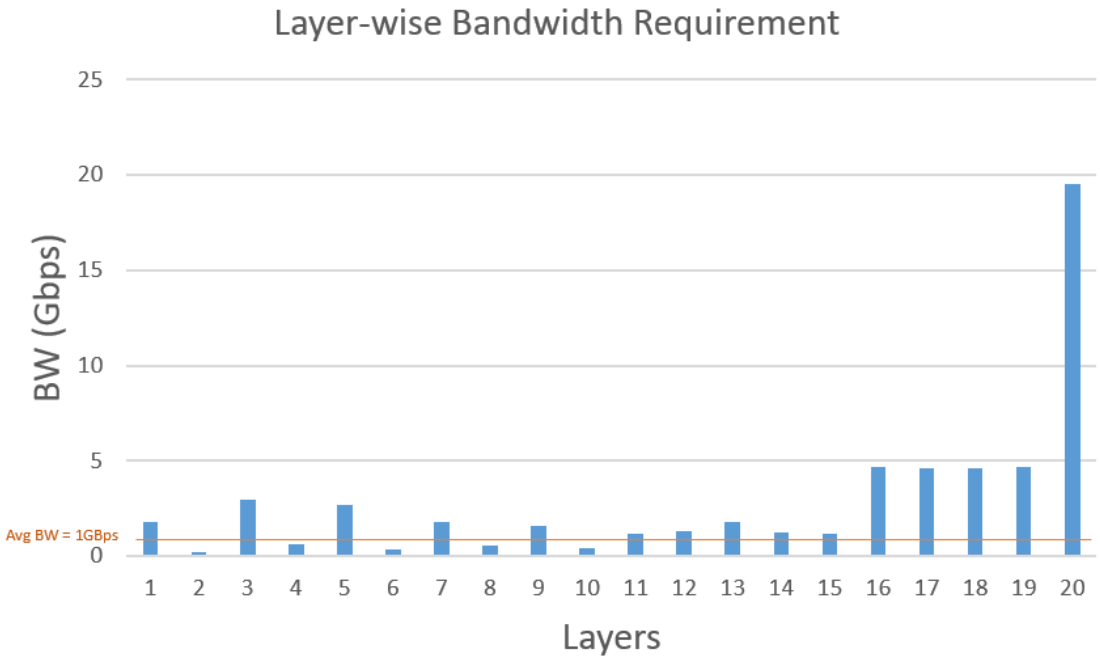


Figure 7.7: Layer-wise Bandwidth Requirement for LCDet-8bit-fixed Point Implementation

7.4.4 Visual Results

We show visual detection results of the face detection performed by the proposed LCDet. Detected faces are marked as *blue* rectangle. Figure 7.9 to Figure 7.11 demonstrate face detection results on FDDB dataset. These figures show that LCDet detects multiple faces of different sizes and poses accurately for a variety of illumination and scale changes. Some of the difficult examples from the Widerface validation dataset are shown in Figure 7.12. LCDet performs well in detecting faces.

Current limitation of the model is that it struggles in tightly localizing small objects in close proximity. Figure 7.13 demonstrates some of the examples where localization might have failed as per strict 50% IoU criteria (marked as *yellow* regions), however the detected faces are not false positives for further face-based processing pipeline. The regions marked in *red* shows missed detections.

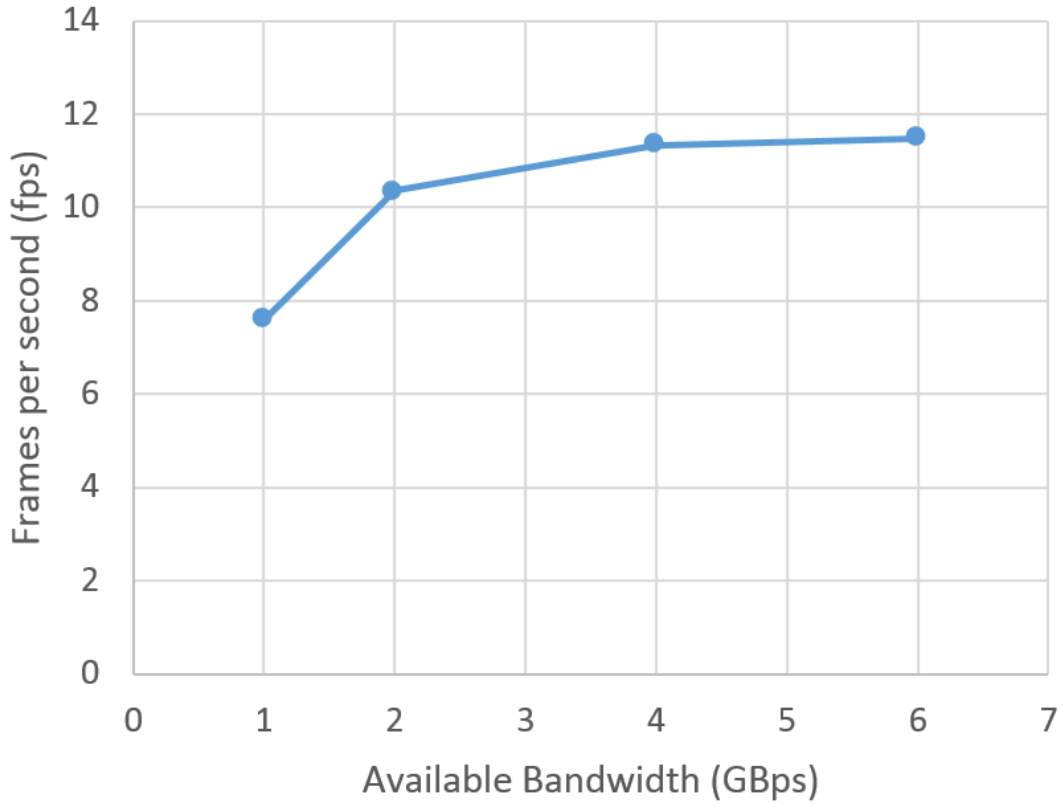


Figure 7.8: Effect of available Dynamic Random Access (DDR) memory Bandwidth (BW) on Frame Rate

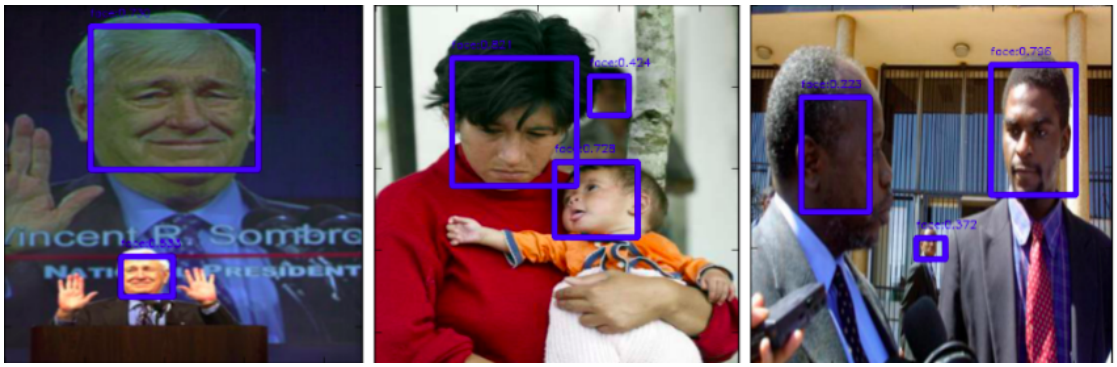


Figure 7.9: Face detection results for faces with different scales on some images from FDDB.



Figure 7.10: Face detection results on black and white images from FDDB dataset.



Figure 7.11: Detection results for multiple faces of frontal and side profiles on sample images from FDDB.

7.5 Summary

We propose LCDet, a low-complexity fully-convolutional neural network for object detection amenable for embedded deployment. This is a unified localization and classification model inspired by [RDGF16] that bypasses the object proposals bottleneck. LCDet performs comparably with state-of-the-art CNN-based face detection methods on FDDB, while being one of the most computationally effective method. We additionally perform 8-bit quantization on this TF-slim based LCDet model, and report one of the first analysis of quantized model for regression model. The quantized LCDet model performs as good as floating point model and reduces the memory footprint by $4\times$. Quantization makes this model apt for implementation in DSPs, or dedicated convolution accelerators.



Figure 7.12: Successful face detection results of LCDet on challenging Widerface Validation images containing difficult examples with pose variation, illumination changes, photograph styles, and different scales.

Although, face detection is studied as a use case here, the network is not optimized for face-specific detection. It is easily expendable to detect any other object categories.

We are aware of the very recent work YOLO9000 [RF16] that has become the state-of-the-art on standard object detection datasets. YOLO9000 that is an improved version of YOLO. Empirically, the performance of LCDet lies between YOLO [RDGF16] and YOLO9000 [RF16]. On the other hand, another recent work SqueezeDet [WIJK16] on low-complexity CNN achieves state-of-the-art performance on KITTI object detector. SqueezeDet appears to be the smallest object detector by virtue of powerful but small backend network of SqueezeNet [IMA$^+$16]. As a future work, we will evaluate the relative performance of our proposed model comparing with these methods. It is also interesting to explore SqueezeNet as the back-end and study performance of quantization.



Figure 7.13: Localization challenges of LCDet on Widerface images. Faces marked as the *yellow* regions are considered false positives as per 50% IoU criteria, but true positive for more relaxed IoU criteria. The regions marked in *red* show missed detections.

Acknowledgments

The work has been done in part during an internship in Qualcomm Multimedia R&D. Chapter 7 is in part based on the papers "LCDet: Low-complexity Fully-Convolutional Neural Networks for Object Detection in Embedded Systems", by S. Tripathi, G. Dane, B. Kang, V. Bhaskaran, and T. Nguyen [TDK+17] and "Low-complexity object detection with deep convolutional neural network for embedded systems", by S. Tripathi, B. Kang, G. Dane, and T. Nguyen [TKDN17]. The dissertation author was the primary investigator and author of these papers.

Bibliography

[AAB+15] Dario Amodei, Rishita Anubhai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Jingdong Chen, Mike Chrzanowski, Adam Coates, Greg Diamos, Erich Elsen, Jesse Engel, Linxi Fan, Christopher Fougner, Tony Han, Awni Y. Hannun, Billy Jun, Patrick LeGresley, Libby Lin, Sharan Narang, Andrew Y. Ng, Sherjil Ozair, Ryan Prenger, Jonathan Raiman, Sanjeev Satheesh, David Seetapun, Shubho Sengupta, Yi Wang, Zhiqian Wang, Chong Wang, Bo Xiao, Dani Yogatama, Jun Zhan, and Zhenyao Zhu. Deep speech 2: End-to-end speech recognition in english and mandarin. *CoRR*, abs/1512.02595, 2015.

[ADF10] Bogdan Alexe, Thomas Deselares, and Vittorio Ferrari. What is an object? *CVPR*, 2010.

[APTB+14] Pablo Arbeláez, Jordi Pont-Tuset, Jonathan Barron, Ferran Marques, and Jitendra Malik. Multiscale combinatorial grouping. *CVPR*, 2014.

[ASB14] Jose M. Alvarez, Mathieu Salzmann, and Nick Barnes. Large-scale semantic co-labeling of image sets. In *IEEE Winter Conference on Applications of Computer Vision*, pages 501–508, March 2014.

[ASS+12] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(11):2274–2282, November 2012.

[ASSL13] Karteek Alahari, Guillaume Seguin, Josef Sivic, and Ivan Laptev. Pose estimation and segmentation of people in 3d movies. In *Proc. IEEE International Conference on Computer Vision*, 2013.

[AT16] Anurag Arnab and Philip H.S. Torr. Bottom-up instance segmentation using deep higher-order crfs. *British Machine Vision Conference (BMVC) 2016*, 2016.

[AV07] David Arthur and Sergei Vassilvitskii. k-means++: the advantages of careful seeding. *ACM-SIAM symposium on Discrete algorithms*, 2007.

[AWI+16] Khalid Ashraf, Bichen Wu, Forrest N. Iandola, Matthew W. Moskewicz, and Kurt Keutzer. Shallow networks for high-accuracy road object-detection. *CoRR*, abs/1606.01561, 2016.

[B.] Paul B. SSD-Tensorflow. https://github.com/balancap/SSD-Tensorflow.

[BCL15] Mark Billinghurst, Adrian Clark, and Gun Lee. A survey of augmented reality. *Found. Trends Hum.-Comput. Interact.*, 8(2-3):73–272, March 2015.

[BFC09] Gabriel J. Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *Pattern Recogn. Lett.*, 30(2):88–97, January 2009.

[BHPB14] Steve Branson, Grant Van Horn, Pietro Perona, and Serge Belongie. Improved bird species recognition using pose normalized deep convolutional nets. *BMVC*, 2014.

[BM11] Thomas Brox and Jitendra Malik. Large displacement optical flow: Descriptor matching in variational motion estimation. *PAMI*, 2011.

[BRFFF16] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the Point: Semantic Segmentation with Point Supervision. *ECCV*, 2016.

[BS09] Xue Bai and Guillermo Sapiro. Geodesic matting: A framework for fast interactive image and video segmentation and matting. *Int. J. Comput. Vision*, 82(2):113–132, April 2009.

[BZBG15] Sean Bell, C. Lawrence Zitnick, Kavita Bala, and Ross B. Girshick. Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. *CVPR 2016*, abs/1512.04143, 2015.

[CBD14] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Low precision arithmetic for deep learning. *CoRR*, abs/1412.7024, 2014.

[CC10] Albert Y. C. Chen and Jason J. Corso. Propagating multi-class pixel labels throughout video frames. In *Image Processing Workshop (WNYIPW), 2010 Western New York*, pages 14–17, Nov 2010.

[CES16] Yu-Hsin Chen, Joel Emer, and Vivienne Sze. Eyeriss: A spatial architecture for energy-efficient dataflow for convolutional neural networks. In *Proceedings of the 43rd International Symposium on Computer Architecture*, ISCA '16, pages 367–379, Piscataway, NJ, USA, 2016. IEEE Press.

[Cha] Fu-Hsiang Chan. Faster rcnn tensorflow. https://github.com/smallcorgi/Faster-RCNN_TF.

[CPK+14] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *CoRR*, abs/1412.7062, 2014.

[CPK+15] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015.

[CR04] Andrew Blake Carsten Rother, Vladimir Kolmogorov. Grabcut - interactive foreground extraction using iterated graph cuts. In *ACM Transactions on Graphics (SIGGRAPH)*, August 2004.

[CvMBB14] KyungHyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proc. Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014.

[CWSS] Zhe Cao, Shih-En Wei, Tomas Simon, and Yaser Sheikh. Multiperson pose estimation using part affinity fields. http://image-net.org/challenges/talks/2016/Multi-person%20pose%20estimation-CMU.pdf.

[CZLT14] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. *IEEE CVPR*, 2014.

[DHG+14] Jeff Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. *CoRR*, abs/1411.4389, 2014.

[DHL+16] Jifeng Dai, Kaiming He, Yi Li, Shaoqing Ren, and Jian Sun. Instance-sensitive fully convolutional networks. *ECCV*, 2016.

[DL15] Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3061–3069, 2015.

[EF10] Marcin Eichner and Vittorio Ferrari. We are family: Joint pose estimation of multiple persons. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *ECCV*, volume 6311 of *Lecture Notes in Computer Science*, pages 228–242. Springer, 2010.

[EGW+10] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vision*, 88(2):303–338, June 2010.

[EH10] Ian Endres and Derek Hoiem. Category independent object proposals. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 575–588, Berlin, Heidelberg, 2010. Springer-Verlag.

[FGMR10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, Sept 2010.

[FLFM15] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[FMR08] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[GDDM] Ross. Girshick, Jeff Donahue, Trevor. Darrell, and Jitendra. Malik. http://www.cs.berkeley.edu/ rbg/r-cnn-release1-data.tgz.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[Gir15] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[GK15] Spyros Gidaris and Nikos Komodakis. Object detection via a multi-region and semantic segmentation-aware cnn model. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[GKHE10] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph-based video segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2141–2148, June 2010.

[GTJ16] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *ECCV*, 2016.

[Gys16] Philipp Gysel. Ristretto: Hardware-oriented approximation of convolutional neural networks. *CoRR*, abs/1605.06402, 2016.

[HBDS15] Jan Hosang, Rodrigo. Benenson, Piotr Dollár, and B. Schiele. What makes for effective detection proposals? *PAMI*, 2015.

[HBS14] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. How good are detection proposals, really? *BMVC*, 2014.

[HGDG17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[Hou10] Stefan Hougardy. The floyd–warshall algorithm on graphs with negative cycles. *Inf. Process. Lett.*, 110(8-9):279–281, April 2010.

[HRS+16] Jonathan Huang, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, and Kevin Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[HS97] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[HVRPE12] Antonio Hernańdez-Vela, Miguel Reyes, Victor Ponce, and Sergio Escalera. Grabcut-based human segmentation in video sequences. *Sensors*, (11), 2012.

[HZRS14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[IM] Alex Ihler and Mike Mandel. http://www.ics.uci.edu/̃ihler/code/kde.html.

[IMA+16] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[IPA+16] Eldar Insafutdinov, Leonid Pishchulin, Bjoern Andres, Mykhaylo Andriluka, and Bernt Schiele. Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In *ECCV*, 2016.

[IZKA14] Phillip Isola, Daniel Zoran, Dilip Krishnan, and Edward H. Adelson. Crisp boundary detection using pointwise mutual information. *ECCV*, 2014.

[JGJ+14] Mihir Jain, Jan Van Gemert, Herve Jegou, Patrick Bouthemy, and Cees G.M. Snoek. Action localization with tubelets from motion. In *CVPR*, pages 740–747, June 2014.

[JL16] Huaizu Jiang and Erik G. Learned-Miller. Face detection with the faster R-CNN. *CoRR*, abs/1606.03473, 2016.

[JLM10] Vidit Jain and Erik Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. Technical Report UM-CS-2010-009, University of Massachusetts, Amherst, 2010.

[JSD+13] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: An open source convolutional architecture for fast feature embedding, 2013. http://caffe.berkeleyvision.org.

[KB14] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[KFS15] Vicky Kalogeiton, Vittorio Ferrari, and Cordelia Schmid. Analysing domain shift factors between videos and images for object detection. *CoRR*, abs/1501.01186, 2015.

[KGF12] Daniel Kuettel, Matthieu Guillaumin, and Vittorio Ferrari. Segmentation propagation in imagenet. *ECCV*, 2012.

[KK13] Philipp Kraehenbuehl and Vladlen Koltun. Parameter learning and convergent inference for dense random fields. In Sanjoy Dasgupta and David Mcallester, editors, *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, volume 28, pages 513–521. JMLR Workshop and Conference Proceedings, May 2013.

[KK14] Philip Krähenbühl and Vladen Koltun. Geodesic object proposals. *ECCV*, 2014.

[KLT09] Pushmeet Kohli, L'Ubor Ladický, and Philip H. Torr. Robust higher order potentials for enforcing label consistency. *Int. J. Comput. Vision*, 82(3):302–324, May 2009.

[KOLW16] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. *CVPR*, 2016.

[KPT+] Nori Kanazawa, George Papandreou, Alex Toshev, Tyler Zhu, Hartwig Adam, Chris Bregler, Kevin Murphy, and Jonathan Tompson. G-RMI pose estimation. http://mscoco.org/dataset/#keypoints-leaderboard.

[KRBT08] Pushmeet Kohli, Jonathan Rihan, Matthieu Bray, and Philip H. S. Torr. Simultaneous segmentation and pose estimation of humans using dynamic graph cuts. *International Journal of Computer Vision*, 79(3):285–298, 2008.

[KSH12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 2012.

[LAE+15] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, and Scott E. Reed. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[Lap05] Ivan Laptev. On space-time interest points. *Int. J. Comput. Vision*, 64(2-3):107–123, September 2005.

[LASL11] Jose Lezama, Karteek Alahari, Josef Sivic, and Ivan Laptev. Track to the future: Spatio-temporal video segmentation with long-range motion cues. In *CVPR 2011*, June 2011.

[LDJ+16] Di Lin, Jifeng Dai, Jiaya Jia, Kaiming He, and Jian Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016.

[LHHHH13] Taegyu Lim, Seunghoon Hong, Bohyung Han, and Joon Hee Han. Joint segmentation and pose tracking of human in natural videos. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[LHM16] Ke Li, Bharath Hariharan, and Jitendra Malik. Iterative instance segmentation. *CVPR*, 2016.

[Liu09] Ce Liu. *Exploring new representations and applications for motion analysis*. PhD thesis, Massachusetts Institute of Technology, 2009.

[LKEW16] Zachary Chase Lipton, David C. Kale, Charles Elkan, and Randall Wetzell. Learning to diagnose with LSTM recurrent neural networks. *ICLR*, 2016.

[LKG11] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *ICCV*, 2011.

[LKH+13] Fuxin Li, Taeyoung Kim, Ahmad Humayun, David Tsai, and James M. Rehg. Video segmentation by tracking many figure-ground segments. In *ICCV*, December 2013.

[LKR16] Marko Linna, Juho Kannala, and Esa Rahtu. Real-time human pose estimation from video with convolutional neural networks. *CoRR*, abs/1609.07420, 2016.

[LMB+14a] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollr, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, Zrich, 2014.

[LMB+14b] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[LQD+] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Translation-aware Fully Convolutional Instance Segmentation. https://github.com/daijifeng001/TA-FCN.

[LSA+10] L'ubor Ladický, Paul Sturgess, Karteek Alahari, Chris Russell, and Philip H. S. Torr. What, where and how many? combining object detectors and crfs. In *Proceedings of the 11th European Conference on Computer Vision: Part IV*, ECCV'10, pages 424–437, Berlin, Heidelberg, 2010. Springer-Verlag.

[LSD15] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[LSWW16] Yunzhu Li, Benyuan Sun, Tianfu Wu, and Yizhou Wang. Face detection with end-to-end integration of a convnet and a 3d model. In *ECCV*, 2016.

[LWS+15] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Jianchao Yang, Liang Lin, and Shuicheng Yan. Proposal-free network for instance-level object segmentation. *CoRR*, abs/1509.02636, 2015.

[LWS+16] Xiaodan Liang, Yunchao Wei, Xiaohui Shen, Zequn Jie, Jiashi Feng, Liang Lin, and Shuicheng Yan. Reversible recursive instance-level object segmentation. In *CVPR*, 2016.

[LZBC15] Zhao Liu, Jianke Zhu, Jiajun Bu, and Chun Chen. A survey of human pose estimation. *J. Vis. Comun. Image Represent.*, 32(C):10–19, October 2015.

[MD02] Remi Megret and Daniel Dementhon. A survey of spatio-temporal grouping techniques. Technical report, University of Maryland, College Park, 2002.

[ML12] Tianyang Ma and L.J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677, June 2012.

[Mod] Modivious. Miriad vpu. https://www.movidius.com/?49b390.

[MPW+08] A. P. Moore, S. J. D. Prince, J. Warrell, U. Mohammed, and G. Jones. Superpixel lattices. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[MYT+16] Huizi Mao, Song Yao, Tianqi Tang, Boxun Li, Jun Yao, and Yu Wang. Towards real-time object detection on embedded systems. *IEEE Transactions on Emerging Topics in Computing*, 2016.

[ORV+14] Dan Oneata, Jerome Revaud, Jakob Verbeek, , and Cordelia Schmid. Spatio-temporal object detection proposals. *ECCV*, 2014.

[OVB+16] Gabriel L. Oliveira, Abhinav Valada, Claas Bollen, Wolfram Burgard, and Thomas Brox. Deep learning for human part discovery in images. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2016.

[OWZ+15] Wanli Ouyang, Xiaogang Wang, Xingyu Zeng, Shi Qiu, Ping Luo, Yonglong Tian, Hongsheng Li, Shuo Yang, Zhe Wang, Chen-Change Loy, and Xiaoou Tang. Deepid-net: Deformable deep convolutional neural networks for object detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[PB16] Eunbyung Park and Alexander C. Berg. Learning to decompose for object detection and instance segmentation. *ICLR*, 2016.

[PCMY15] George Papandreou, Liang-Chieh Chen, Kevin Murphy, and Alan L. Yuille. Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. *CoRR*, abs/1502.02734, 2015.

[PD09] Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *Int. J. Comput. Vision*, 81(1):24–52, January 2009.

[PF13] Anestis Papazoglou and Vittorio Ferrari. Fast object segmentation in unconstraint video. *ICCV*, 2013.

[PIT+16] Leonid Pishchulin, Eldar Insafutdinov, Siyu Tang, Bjoern Andres, Mykhaylo Andriluka, Peter V. Gehler, and Bernt Schiele. Deepcut: Joint subset partition and labeling for multi person pose estimation. In *CVPR*, 2016.

[PKL+14] Alessandro Prest, Vicky Kalogeiton, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Youtube-objects dataset v2.0, 2014. University of Edinburgh (CALVIN), INRIA Grenoble (LEAR), ETH Zurich (CALVIN).

[PLC+12a] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. *CVPR*, 2012.

[PLC+12b] Alessandro Prest, Christian Leistner, Javier Civera, Cordelia Schmid, and Vittorio Ferrari. Learning object class detectors from weakly annotated video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012*, pages 3282–3289, 2012.

[QSL+15] Xiaojuan Qi, Jianping Shi, Shu Liu, Renjie Liao, Jiaya Jia, undefined, undefined, undefined, and undefined. Semantic segmentation with object clique potential. *2015 IEEE International Conference on Computer Vision (ICCV)*, 00:2587–2595, 2015.

[Qua17] Qualcomm. Snapdragon 835 processor, 2017. https://www.qualcomm.com/products/snapdragon/processors/835.

[QWY+16] Jiantao Qiu, Jie Wang, Song Yao, Kaiyuan Guo, Boxun Li, Erjin Zhou, Jincheng Yu, Tianqi Tang, Ningyi Xu, Sen Song, Yu Wang, and Huazhong Yang. Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, FPGA '16, pages 26–35, New York, NY, USA, 2016. ACM.

[QYLH16] Hongwei Qin, Junjie Yan, Xiu Li, and Xiaolin Hu. Joint training of cascaded cnn for face detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[RDGF16] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[Red17] Joseph Chet Redmon. Darknet, 2017. http://pjreddie.com/darknet/.

[RF16] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[RHGS15] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[RPZ13] Brandon Rothrock, Seyoung Park, and Song-Chun Zhu. Integrating grammar and segmentation for human pose estimation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3214–3221, 2013.

[RT16] Bernardino Romera-Paredes and Philip Hilaire Sean Torr. Recurrent instance segmentation. In *ECCV*, pages 312–329, 2016.

[RTK09] Chris Russell, Philip H. S. Torr, and Pushmeet Kohli. Associative hierarchical crfs for object class image segmentation. In *Proceedings of the International Conference on Computer Vision*, 2009.

[RW08] Peter M. Roth and Martin Winter. Survey of appearance-based methods for object recognition. Technical report, Graz University of Technology, Austria, 2008.

[SASL15] Guillaume Seguin, Karteek Alahari, Josef Sivic, and Ivan Laptev. Pose estimation and segmentation of people in 3D Movies. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(8):1643–1655, 2015.

[Sb17] Snapdragon-blog. Tensorflow machine learning now optimized for the snapdragon 835 and hexagon 682 dsp, 2017. `https://www.qualcomm.com/news/snapdragon/2017/01/09/tensorflow-machine-learning-now-optimized-snapdragon-835-and-hexagon-682`.

[SBK10] Narayanan Sundaram, Thomas Brox, and Kurt Keutzer. Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. In *ECCV*, 2010.

[SEZ+13] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[SREA14] Christian Szegedy, Scott E. Reed, Dumitru Erhan, and Dragomir Anguelov. Scalable, high-quality object detection. *CoRR*, abs/1412.1441, 2014.

[ST12] Gilad Sharir and Tinne Tuytelaars. Video object proposals. *CVPR workshop on Large-Scale Video Search and Mining*, 2012.

[SWH17] Xudong Sun, Pengcheng Wu, and Steven C. H. Hoi. Face detection using deep learning: An improved faster RCNN approach. *CoRR*, abs/1701.08289, 2017.

[SWRC09] Jamie Shotton, John Winn, Carsten Rother, and Antonio Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *Int. J. Comput. Vision*, 81(1):2–23, January 2009.

[SZ14] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[TBHN16] Subarna Tripathi, Serge J. Belongie, Youngbae Hwang, and Truong Q. Nguyen. Detecting temporally consistent objects in videos through object class label propagation. In *IEEE Winter Conference on Applications of Computer Vision, Lake Placid, NY, USA, March 24-26, 2016*, 2016.

[TCBB17] Subarna Tripathi, Maxwell Collins, Matthew Brown, and Serge J. Belongie. Pose2instance: Harnessing keypoints for person instance segmentation. 2017.

[TDK+17] Subarna Tripathi, Gokce Dane, Byeongkeun Kang, Vasudev Bhaskaran, and Truong Nguyen. Lcdet: Low-complexity fully-convolutional neural networks for object detection in embedded systems. *CoRR*, 2017.

[Tena] TensorFlow. Quantization. https://www.tensorflow.org/performance/quantization.

[Tenb] TensorFlow. Tensorflow for poets. https://petewarden.com/2015/05/23/why-are-eight-bits-enough-for-deep-neural-networks/.

[TH12] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude, 2012.

[THBN14a] Subarna Tripathi, Youngbae Hwang, Serge Belongie, and Truong Q. Nguyen. Improving streaming video segmentation with early and mid-level visual processing. *WACV*, 2014.

[THBN14b] Subarna Tripathi, Youngbae Hwang, Serge J. Belongie, and Truong Q. Nguyen. Improving streaming video segmentation with early and mid-level visual processing, 2014.

[THBN15] Subarna Tripathi, Youngbae Hwang, Serge J. Belongie, and Truong Q. Nguyen. Semantic video segmentation: Exploring inference efficiency. In *International SoC Design Conference, Korea, November 2-5, 2015*, pages 155–158, 2015.

[The16] Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688, May 2016.

[TKDN17] Subarna Tripathi, Byeongkeun Kang, Gokce Dane, and Truong Nguyen. Low-complexity object detection with deep convolutional neural network for embedded system. In *SPIE 10396, Applications of Digital Image Processing XL*, 2017.

[TLBN16] Subarna Tripathi, Zachary Chase Lipton, Serge J. Belongie, and Truong Q. Nguyen. Context matters : Refining object detection in video

with recurrent neural networks. In *British Machine Vision Conference, York, UK, Sept 19-22, 2016*, 2016.

[UvdSGS13] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013.

[VF15] Alexander Vezhnevets and Vittorio Ferrari. Looking out of the window: object localization by joint analysis of all windows in the image. *CoRR*, abs/1501.01181, 2015.

[VRAPM10] Amelio Vazquez-Reina, Shai Avidan, Hanspeter Pfister, and Eric Miller. *Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part V*, chapter Multiple Hypothesis Video Segmentation from Superpixel Flows, pages 268–281. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[VSM11] Vincent Vanhoucke, Andrew Senior, and Mark Z. Mao. Improving the speed of neural networks on cpus. In *Deep Learning and Unsupervised Feature Learning Workshop, NIPS 2011*, 2011.

[VWT12] Vibhav Vineet, Jonathan Warrell, and Philip H. S. Torr. Filter-based mean-field inference for random fields with higher-order terms and product label-spaces. In *Proceedings of the European Conference on Computer Vision*, 2012.

[War] Pete Warden. Pete warden's blog. https://petewarden.com/2016/05/03/how-to-quantize-neural-networks-with-tensorflow/.

[WCZ+16] Shaohua Wan, Zhijun Chen, Tao Zhang, Bo Zhang, and Kong-kat Wong. Bootstrapping face detection with hard negative examples. *CoRR*, abs/1608.02236, 2016.

[WDB+08] Ofir Weber, Yohai S. Devir, Alexander M. Bronstein, Michael M. Bronstein, and Ron Kimmel. Parallel algorithms for approximation of distance maps on parametric surfaces. *ACM Trans. Graph.*, 27(4):104:1–104:16, November 2008.

[WGM] Peng Wang, Sergio Guadarrama, and Kevin Murphy. G-RMI Instance Segmentation. http://mscoco.org/dataset/#detections-leaderboard.

[WIJK16] Bichen Wu, Forrest N. Iandola, Peter H. Jin, and Kurt Keutzer. Squeezedet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving. *CoRR*, abs/1612.01051, 2016.

[WLW+15] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. *CoRR*, abs/1512.06473, 2015.

[XC12] Chenliang Xu and Jason J. Corso. Evaluation of super-voxel methods for early video processing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1202–1209, June 2012.

[XXC12a] Chenliang Xu, Caiming Xiong, and Jason J. Corso. Streaming hierarchical video segmentation. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 626–639, Berlin, Heidelberg, 2012. Springer-Verlag.

[XXC12b] Chenliang Xu, Caiming Xiong, and Jason J. Corso. Streaming hierarchical video segmentation. *ECCV*, 2012.

[YHNHV+15] Joe Yue-Hei Ng, Matthew Hausknecht, Sudheendra Vijayanarasimhan, Oriol Vinyals, Rajat Monga, and George Toderici. Beyond short snippets: Deep networks for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4694–4702, 2015.

[YJW+16] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao, and Thomas Huang. Unitbox: An advanced object detection network. In *Proceedings of the 2016 ACM on Multimedia Conference*, MM '16, pages 516–520, New York, NY, USA, 2016. ACM.

[YLLT15] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3676–3684, Dec 2015.

[YLLT16] Shuo Yang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Wider face: A face detection benchmark. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[YYLL16] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z. Li. Craft objects from images. *CVPR*, 2016.

[ZD14] C. Lawrence Zitnick and Piotr Dollar. Edge boxes: Locating object proposals from edges. *ECCV*, 2014.

[ZFU16] Ziyu Zhang, Sanja Fidler, and Raquel Urtasun. Instance-level segmentation with deep densely connected mrfs. In *CVPR*, 2016.

[ZJRP+15] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip Torr. Conditional random fields as recurrent neural networks. In *International Conference on Computer Vision (ICCV)*, 2015.

[ZJS13] Dong Zhang, O. Javed, and M. Shah. Video object segmentation through spatially accurate and temporally dense extraction of primary object regions. In *CVPR*, pages 628–635, June 2013.

[ZLL+16] Sergey Zagoruyko, Adam Lerer, Tsung-Yi Lin, Pedro H. O. Pinheiro, Sam Gross, Soumith Chintala, and Piotr Dollár. A multipath network for object detection. *CoRR*, abs/1604.02135, 2016.

[ZMP04] Lihi Zelnik-Manor and Pietro Perona. Self-tuning spectral clustering. *NIPS*, 2004.