DISS. ETH NO. 26344

EXPLOITING LOW-LEVEL FEATURES FOR HIGHER-LEVEL SCENE UNDERSTANDING

A dissertation submitted to

ETH ZÜRICH

for the degree of

DOCTOR OF SCIENCES OF ETH ZÜRICH

presented by

KEVIS-KOKITSI MANINIS
Diploma in Electrical and Computer Engineering
National Technical University of Athens
born 8 August 1990
citizen of Greece

accepted on the recommendation of

Prof. Dr. Luc Van Gool, examiner
Prof. Dr. Federico Tombari, co-examiner
Prof. Dr. Iasonas Kokkinos, co-examiner
Dr. Jordi Pont-Tuset, co-examiner

2019

To my family.

ABSTRACT

Scene understanding is one of the fastest growing areas in computer vision research. Such growth is mainly driven by the emergence of deep learning techniques that contributed to boosting performance on popular benchmarks for well-studied tasks, and to approaching tasks that have been very difficult to solve with traditional techniques. This dissertation examines how traditional low-level features such as boundaries and points help in tackling higher-level scene understanding tasks such as detection, segmentation, and 3D reconstruction.

First, we propose a hierarchical grouping algorithm that uses deeply learned boundaries and their orientation. We examine how grouping from predicted boundaries can help object detection and semantic segmentation when plugged into the corresponding pipelines.

Second, we use human-generated points for guided object segmentation. We show how to obtain segmented masks by using extreme points provided by humans, and how to speed up the time-consuming process of annotating for segmentation by using this technique.

Third, we show how automatically detected keypoints help 3D reconstruction in a complicated environment for robot-assisted retinal surgery. The task is to provide visual guidance during surgery by using two stereo cameras mounted on the surgical microscope. We propose a method for calibration, 3D registration, and 3D reconstruction from a single pipeline, by detecting specific robot keypoints, and by obtaining 3D to 2D correspondences just by moving the robot.

Last, we examine the interplay of low-level and high-level tasks when trained jointly in a single neural network. We propose ways to overcome problems such as task interference and limited capacity as a result of jointly training for many different, unrelated tasks. We propose a universal network that can tackle all tasks, but only one task at a time.

All in all, we show how to predict low-level features and how they contribute to different pipelines a) in combination with deep networks trained for scene understanding b) as human-generated input, c) in combination with 3D reconstruction, and d) by jointly training them with higher-level tasks.

ZUSAMMENFASSUNG

Szenenverständnis (Scene Understanding) ist einer der am schnellsten wachsenden Bereiche in der Bildverarbeitung. Diese Entwicklung wird hauptsächlich durch "Deep LearningTechnologien angetrieben, die dazu beigetragen haben, die Genauigkeit in Benchmarks vieler klassischer Problemstellungen der Computer Vision zu steigern und Aufgaben anzugehen, die mit traditionellen Techniken nur sehr schwer zu lösen waren. Diese Dissertation untersucht, auf welche Weise klassische low-level Features, wie Objektränder oder Keypoints, der Bewältigung von abstrakten SScene UnderstandingAufgaben, wie Objekterkennung, Segmentierung und 3D-Rekonstruktion, dienen können.

Zuerst entwickeln wir einen hierarchischen Gruppierungsalgorithmus, basierend auf Objekträndern und deren Orientierung. Wir untersuchen, wie die Gruppierung aus prognostizierten Objekträndern die Objekterkennung und semantische Segmentierung unterstützen kann, wenn sie in die entsprechenden Pipelines eingesetzt wird.

Zweitens verwenden wir von Menschen annotierte Punkte, um die automatisierte Segmentierung von Objekten zu führen. Wir zeigen, wie man vollständige Segmentierungsmasken anhand von wenigen manuell definierten Ëxtreme Pointsërhält, und wie man mit dieser Technik den zeitaufwendigen Prozess der Annotierung für Segmentierungsaufgaben beschleunigt.

Drittens zeigen wir, wie automatisch erkannte Keypoints die 3D-Rekonstruktion in der komplexen Umgebung robotergestützter Netzhautchirurgie ermöglichen kann. Die Aufgabe besteht darin, anhand eines am Operationsmikroskop montierten Stereokamerasystems, eine visuelle Führung während der Operation zu gewährleisten. Wir schlagen dazu ein Verfahren zur Kalibrierung, 3D-Registrierung und 3D-Rekonstruktion basierend auf einer einzigen Pipeline vor, wobei wir Roboter-spezifische Keypoints erkennen und 3D-zu-2D-Korrespondenzen durch kontrollierte Bewegungen des Roboterarms erhalten.

Schliesslich untersuchen wir das Zusammenspiel von low-level- und high-Level-Aufgaben, wenn sie gemeinsam in einem einzigen neuronalen Netzwerk trainiert werden. Durch das gemeinsame Training für viele verschiedene, voneinander unabhängige Aufgaben, können In-

terferenzen zwischen Aufgaben und die begrenzte Netzwerkkapazität problematisch werden. Wir zeigen, wie diese Probleme überwunden werden können anhand eines universellen Netzwerks, das alle Aufgaben (sequentiell) bewältigt.

Zusammenfassend zeigen wir vier Wege auf, wie low-level Features zum abstrakten, high-level Bildverstehen beitragen können: a) direkt, durch deren robuste Schätzung, b) durch das Ermöglichen effizienter manueller Annotierung, c) durch das Ermöglichen von 3D Rekonstruktion, und d) durch gemeinsames Training mit high-level Aufgaben.

PUBLICATIONS

The following publications are included as a whole or in parts in this dissertation:

- K.-K. Maninis et al. „Convolutional Oriented Boundaries.“ In: *European Conference on Computer Vision (ECCV)*. 2016.
- K.-K. Maninis et al. „Deep Extreme Cut: From Extreme Points to Object Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- K.-K. Maninis et al. „Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40.4 (2018), pp. 819–833.
- T. Probst et al. „Automatic Tool Landmark Detection for Stereo Vision in Robot-Assisted Retinal Surgery.“ In: *IEEE Robotics and Automation Letters (RA-L)* 3.1 (2018), pp. 612–619.
- K.-K. Maninis, I. Radosavovic, and I. Kokkinos. „Attentive Single-Tasking of Multiple Tasks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.

Furthermore, the following publications for which I was major contributor were part of my PhD research, but are nevertheless not covered in this thesis. The topics of these publications are outside of the scope of the material covered here:

- K.-K. Maninis et al. „Deep Retinal Image Understanding.“ In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2016.
- S. Caelles et al. „One-Shot Video Object Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017.
- K.-K. Maninis et al. „Video Object Segmentation Without Temporal Information.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2019).

ACKNOWLEDGMENTS

First of all, I would like to thank my advisor Prof. Luc Van Gool for our discussions, for trusting me when I wanted to do things *my way*, for all freedom in research that he provided, and for the wonderful environment he has created at CVL. I can not imagine a better place for someone to do their PhD.

This thesis would have never been in its current state without the guidance of Dr. Jordi Pont-Tuset, who was a true leader and a role model for me since the very beginning of my studies. The Acknowledgments' section would need to become a separate chapter if I listed all things I learned from him (from tiny details such as writing 'state of the art' vs. writing 'state-of-the-art results' to more fundamental things such as the importance of ablation studies and benchmarking). Jordi was my first contact with research in segmentation, and transferred his passion for the topic to me. From the bottom of my heart, thank you.

My deepest gratitude to Prof./*entrepreneur* Iasonas Kokkinos, with whom I wanted to collaborate since my undergrad. I had that opportunity during my internship, in one of the most productive periods for me. Thank you for all the advice, for your trust, and your passion for multi-tasking (both in research and in real-life).

I would like to thank Prof. Federico Tombari for promptly agreeing to be part of my jury, and for showing me how professional one can be for their work and for building their team.

This PhD would certainly be incomplete without my extremely talented collaborators: Sergi Caelles - good luck with your PhD, almost there! -, Thomas Probst - one way or another, we always worked on eyes -, Ajad Chhatkuli - deep learning $>$ non-rigird 3D -, Michal Havlena - let's have some good beers in Leuven again -, Prof. Pablo Arbeláez - I hope to see you soon at some conference -, Ilija Radosavovic - all the best with your new PhD adventure -, Yuhua Chen - you are sitting at your desk behind mine while I am writing this, it's so uncomfortable -, Simon Hecker - let's submit something to CRAS again.

I am very thankful for my office mates at D 115: Eirikur Agustsson, Fabian Mentzer, David Brüggemann, Yuhua and Sergi for the incredible work environment. Actually, there were many distractions in the

office, but we always had our noise-canceling headphones. These were completely useless when Dr. Nikolay Kobyshev was talking loud on the phone, 2 offices next to ours, and made it really feel as if he was in the office somehow. I will certainly miss all of this.

I would like to thank my students Miriam Bellver, Alberto Montes, and Apostolis Krystallidis for our collaboration, I hope you enjoyed it as much as I did.

To all the CVL members and the alumni for all sorts of activities that we did together, from everyday lunches to biking to Italy from Zurich during midnight and swimming with dolphins in Hawaii, these are memories I will never forget. Thank you.

To all the 'Greeks in computer vision' with whom we shared experiences, thoughts, problems, advice, long-term plans: Georgios (not to be confused with the name 'George'), Petros, Christos, Stam, Despoina, Menelaos, Kostas. Thank you.

In retrospect, one of the most life-changing moments in this ongoing journey was my first exposure to Computer Vision in Prof. Petros Maragos' classes, and our work together in human action recognition. Without his passion in teaching, and his guidance in my first steps, this thesis would have definitely been impossible. Thank you Prof. Maragos.

Any of these would be both impossible and meaningless without the unconditional love and support from my parents: Kazuyo Kawamura and Ioannis Maninis. The most special thanks goes to Evangelia, who has made my life so enjoyable. Thank you for always being there, and for all the life-changing decisions we have made together.

Finally, I thank Eureyecase and Specta.AI for funding my research, providing with data, equipment, infrastructure, and supporting me throughout my PhD.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1

INTRODUCTION

Recognizing and analyzing visual cues in images and videos is a fundamental goal of computer vision research. Methods and algorithms that have been developed throughout decades constitute contributions towards the unified goal of understanding the environment from visual cues, exactly as *humans can effortlessly do*.

The very early approaches in computer vision consisted in purely hand-crafted features that were intelligently designed. The community invested a lot of efforts in finding *invariant features* that were robust to changes that naturally occur in images and should not affect recognition results, such as illumination changes, rotation, scaling, etc. One of the most important findings was that gradient-based features [129, 14, 26, 46, 70] are robust to such changes, and thus suitable for recognition. Parts of these pipelines were gradually substituted by learning-based algorithms that automatically learned how to make decisions. For example, machine learning techniques such as support vector machines [43] or random forests [23] were used in conjunction with hand-crafted features to enhance recognition performance. But still the features remained a product of intelligent, but not automated, human design, which make them sub-optimal when the assumptions made during design do not hold.

Recently, a new learning paradigm that long existed but was limited by hardware and availability of data received most-deserved attention. Deep learning, in the form of Convolutional Neural Networks (CNN) [106] , applied to image classification, boosted performance by a very large margin back in 2012 [103]. The main idea is to learn the features and the classifier end-to-end, in a bottom-up fashion that primarily depends on the availability of data. This was made possible by modeling composition of functions by stacking together a sequence of convolutions, non-linear activations, and pooling functions. The large capacity of CNNs that can handle - and in fact, need - very large amounts of data shifted the community's attention from hand-crafted features to data engineering and smart data acquisition techniques. Ever since, the field is primarily dominated by modifications of CNN

architectures that have become deeper [199, 202] and deeper [74, 133], wider [229], or designed in a more principled way [218, 80, 40, 204].

CNNs are ever since revolutionizing the field of computer vision and machine learning. Popular benchmarks that were very competitive test-beds for various challenging tasks such as boundary detection and grouping [144], optical flow [24], semantic segmentation [54, 122, 42] or image classification [188] are today dominated by deep learning approaches that often surpass human performance for the respective tasks (in the controlled environments of the corresponding datasets). Even though the community has come a long way in terms of what is possible today compared to the past, there is still a lot to be explored.

The goal of this dissertation is to show that the use of *low-level fetures*, *i.e* boundaries and points, can help many different tasks for *higher-level scene understanding*, *i.e* semantic segmentation, guided segmentation, object detection, 3D reconstruction; when coupled with the aforementioned modern deep learning pipelines. We show that low-level features can be beneficial for higher-level tasks a) in combination with deep networks trained for scene understanding (Chapter 2), b) as human-generated input (Chapter 3), c) in combination with traditional pipelines for 3D reconstruction 4), and d) by jointly training them with higher-level tasks in a multi-task learning setup (Chapter 5). In particular, we make the following four contributions:

- A deep-learning based approach to use multi-scale boundary detection and region hierarchies for improving semantic segmentation or object detection pipelines (Chapter 2). We present Convolutional Oriented Boundaries (COB), which produces multi-scale oriented contours and region hierarchies starting from generic image classification CNNs. COB is computationally efficient, because it requires a single CNN forward pass for multi-scale contour detection and it uses a novel sparse boundary representation for hierarchical segmentation; it gives a significant leap in performance over the state of the art, and it generalizes very well to unseen categories and datasets. Particularly, we show that learning to estimate not only contour strength but also orientation provides more accurate results. We perform extensive experiments for low-level applications on BSDS, PASCAL Context, PASCAL Segmentation, and NYUD to evaluate boundary detection performance, showing that COB provides state-of-the-art

contours and region hierarchies in all datasets. We also evaluate COB on high-level tasks when coupled with multiple pipelines for object proposals, semantic contours, semantic segmentation, and object detection on MS-COCO, SBD, and PASCAL, showing that COB also improves the results for all tasks.

- An efficient scheme to obtain segmented object masks from extreme points using an end-to-end approach (Chapter 3). This chapter explores the use of extreme points in an object (left-most, right-most, top, bottom pixels) as input to obtain precise object segmentation for images and videos. We do so by adding an extra channel to the image in the input of a CNN, which contains a heatmap with Gaussians centered in each of the extreme points. The CNN learns to transform this information into a segmentation of an object that matches those extreme points. We demonstrate the usefulness of this approach for guided segmentation, interactive segmentation, video object segmentation, and dense segmentation annotation. We show that we obtain the most precise results to date, also with less user input, in an extensive and varied selection of benchmarks and datasets.

- A novel approach to predicting keypoints of surgical tools on images and using them to jointly solve calibration, 3D registration, and 3D reconstruction in a complicated, uncalibrated setup for robot-assisted retinal surgery (Chapter 4). In recent works, such operations are conducted under a stereo-microscope, and with a robot-controlled surgical tool. The complementarity of computer vision and robotics has however not yet been fully exploited. In order to improve the robot control we are interested in 3D reconstruction of the anatomy and in automatic tool localization using a stereo microscope. We solve this problem for the first time using a single pipeline, starting from uncalibrated cameras to reach metric 3D reconstruction and registration, in retinal microsurgery. The key ingredients of our method are: (a) surgical tool landmark detection, and (b) 3D reconstruction with the stereo microscope, using the detected landmarks. To address the former, we propose a novel deep learning method that detects and recognizes keypoints in high definition images faster than real time. We use the detected 2D keypoints along with their corresponding 3D coordinates obtained from the robot sensors to

calibrate the stereo microscope using an affine projection model. We design an online 3D reconstruction pipeline that makes use of smoothness constraints and performs robot-to-camera registration. The entire pipeline is extensively validated on open-sky porcine eye sequences. Quantitative and qualitative results are presented for all steps.

- A solution to task interference and network capacity issues that arise when trying to jointly train a universal CNN for multiple, potentially unrelated low-level and higher-level tasks (Chapter 5). We address these issues by considering that a network is trained on multiple tasks, but performs one task at a time, an approach we refer to as "single-tasking multiple tasks". The network thus modifies its behavior through task-dependent feature adaptation, or task attention. This gives the network the ability to accentuate the features that are adapted to a task, while shunning irrelevant ones. We further reduce task interference by forcing the task gradients to be statistically indistinguishable through adversarial training, ensuring that the common backbone architecture serving all tasks is not dominated by any of the task-specific gradients. Results in three multi-task dense labeling problems consistently show: (i) a large reduction in the number of parameters while preserving, or even improving performance and (ii) a smooth trade-off between computation and multi-task accuracy.

2

CONVOLUTIONAL ORIENTED BOUNDARIES: FROM IMAGE SEGMENTATION TO HIGHER-LEVEL TASKS

We present Convolutional Oriented Boundaries (COB), which produces multiscale oriented contours and region hierarchies starting from generic image classification Convolutional Neural Networks (CNNs). COB is computationally efficient, because it requires a single CNN forward pass for multi-scale contour detection and it uses a novel sparse boundary representation for hierarchical segmentation; it gives a significant leap in performance over the state of the art, and it generalizes very well to unseen categories and datasets. Particularly, we show that learning to estimate not only contour strength but also orientation provides more accurate results. We perform extensive experiments for low-level applications on BSDS, PASCAL Context, PASCAL Segmentation, and NYUD to evaluate boundary detection performance, showing that COB provides state-of-the-art contours and region hierarchies in all datasets. We also evaluate COB on high-level tasks when coupled with multiple pipelines for object proposals, semantic contours, semantic segmentation, and object detection on MS-COCO, SBD, and PASCAL; showing that COB also improves the results for all tasks.

2.1 INTRODUCTION

The adoption of Convolutional Neural Networks (CNNs) has caused a profound change and a large leap forward in performance throughout the majority of fields in computer vision. In the case of a traditionally category-agnostic field such as contour detection, it has recently fostered the appearance of systems [97, 219, 17, 18, 195, 58] that rely on large-scale category-specific information in the form of deep architectures pre-trained on ImageNet [188] for image classification [103, 202, 199, 74].

This chapter introduces Convolutional Oriented Boundaries (COB), a generic CNN architecture that allows end-to-end learning of multiscale oriented contours, and we show how it translates top performing base CNN networks into high-quality contours; allowing to bring future

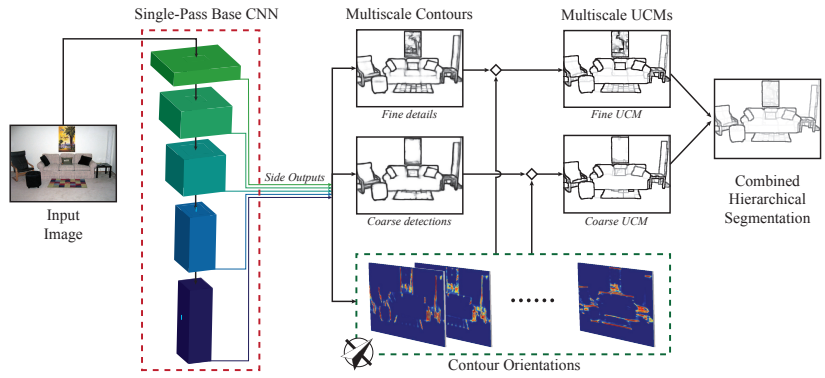


Figure 2.1: **Overview of COB**: From a single pass of a base CNN, we obtain multiscale oriented contours. We combine them to build Ultrametric Contour Maps (UCMs) at different scales and fuse them into a single hierarchical segmentation structure.

improvements in base CNN architectures into semantic grouping. We then propose a sparse boundary representation for efficient construction of hierarchical regions from the contour signal. Our overall approach is both efficient (it runs in 0.8 seconds per image) and highly accurate (it produces state-of-the-art contours and regions on PASCAL and on the BSDS). Figure 2.1 shows an overview of our system.

For the last fifteen years, the Berkeley Segmentation Dataset and Benchmark (BSDS) [144, 11] has been the experimental testbed of choice for the study of boundary detection and image segmentation. However, the current large-capacity and very accurate models have underlined the limitations of the BSDS as the primary benchmark for grouping. Its 300 train images are inadequate for training systems with tens of millions of parameters and, critically, current state-of-the-art techniques are reaching human performance for boundary detection on its 200 test images.

In terms of scale and difficulty, the next natural frontier for perceptual grouping is the PASCAL VOC dataset [54], an influential benchmark for image classification, object detection, and semantic segmentation which has a *trainval* set with more than 10 000 challenging and varied images. A first step in that direction was taken by Hariharan et al. [69], who annotated the VOC dataset for category-specific boundary detection on the foreground objects. More recently, the PASCAL Context

dataset [147] extended this annotation effort to all the background categories, providing fully-parsed images which are a direct VOC counterpart to the human ground truth of the BSDS. In this direction, this chapter investigates the transition from the BSDS to PASCAL Context in the evaluation of image segmentation.

We derive valuable insights from studying perceptual grouping in a larger and more challenging empirical framework. Among them, we observe that COB leverages increasingly deeper state-of-the-art architectures, such as the recent Residual Networks [74], to produce improved results. This indicates that our approach is generic and can directly benefit from future advances in CNNs. We also observe that, in PASCAL, the globalization strategy of contour strength by spectral graph partitioning proposed in [11] and used in state-of-the-art methods [166, 97] is unnecessary in the presence of the high-level knowledge conveyed by pre-trained CNNs and oriented contours, thus removing a significant computational bottleneck for high-quality contours.

We conduct two types of experiments, the first of which regards low-level vision applications, such as contour detection and generic segmentation on PASCAL Context and the BSDS500. We extend the evaluation to the NYUD RGB-D dataset, showing that the pipeline of COB can benefit from depth embeddings. We also include evaluation of object contour detection on the PASCAL VOC'12 database. In all cases, COB demonstrates state-of-the-art performance on contours and regions while being computationally efficient.

In a second set of experiments, we study the interplay of COB with various downstream recognition applications. We use our hierarchical regions as input to the combinatorial grouping algorithm of [166] and obtain state-of-the-art segmented object proposals on PASCAL VOC'12 Segmentation by a significant margin. Furthermore, we provide empirical evidence for the generalization power of COB by evaluating our object proposals without any retraining in the even larger and more challenging MS-COCO [122] dataset, where we also report competitive results compared to the state of the art. We have also studied the effects of COB when coupled with well-known pipelines, showing that injecting COB detections to them lead to improvements on Semantic Segmentation and Object Detection. Finally, we report a new state of the art on Semantic Boundary detection.

Our approach to segmentation has also found application in retinal image segmentation [140], obtaining state-of-the-art and super-human

performance in vessel and optic disc segmentation, which further highlights its generality. Furthermore, in a different line of our work we show that using the superpixels generated from COB enhances performance for video object segmentation [25, 136].

2.2 RELATED WORK

Contour Detection: Early approaches to contour detection relied on local gradient measurements in an image [184, 95, 170]. These simple edge detectors operate by applying local derivative filters on grayscale images. Gradient filtering was followed by detection of zero crossings [142], or by non-maximum suppression [26].

Such simple gradient techniques are unable to handle information captured by richer features such as color and texture [143], or Statistical Edges [99]. Martin et al. [143] define rich gradient operators out of color, brightness and texture, and use them as input to a logistic regression classifier. Their approach is extended by Arbeláez et al. [11], to combine contours at multiple scales.

Machine Learning techniques contributed to learnable features and classifiers that boosted contour detection performance, especially after the manual annotation of the BSDS database [143, 11]. The BEL algorithm [47] attempts to learn an edge classifier in the form of a probabilistic boosting tree. Kokkinos [96] trains an orientation-sensitive boundary detector using Multiple-Instance Learning. Ren and Bo [180] use patch representations automatically learned through sparse coding. Sketch Tokens [118] and Structured Edges [48] tackle both accuracy and speed, by using random forests to classify the central pixel of patches.

The latest wave of contour detectors takes advantage of deep learning to obtain state-of-the-art results [97, 219, 17, 18, 195, 58, 19]. Ganin and Lempitsky [58] use a deep architecture to extract features of image patches. They approach contour detection as a multi-class classification task, by matching the extracted features to predefined ground-truth features. The authors of [17, 18] make use of features generated by pre-trained CNNs to regress contours. They prove that object-level information provides powerful cues for the prediction of contours. Shen et al. [195] learn deep features using shape information. Xie and Tu [219] provide an end-to-end deep framework to boost the efficiency and accuracy of contour detection, using convolutional feature maps and a novel loss function. An extended version of their work, with many

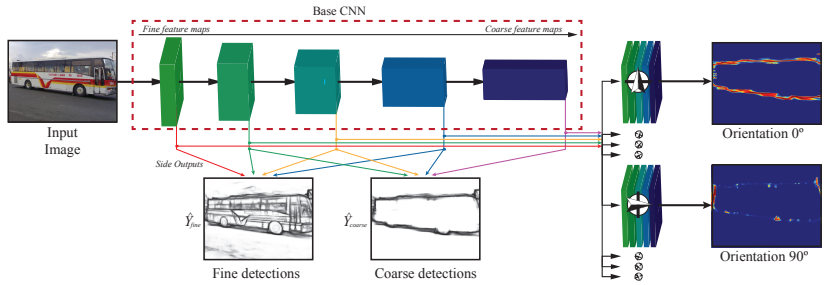


Figure 2.2: **The deep learning architecture of COB:** The connections show the different stages that are used to generate the multiscale contours. Orientations further require additional convolutional layers in multiple stages of the network (best viewed in color).

additional experiments can be found in [220]. Kokkinos [97] builds upon [219] and improves the results by tuning the loss function, running the detector at multiple scales, adding GPU-optimized globalization, and exploiting multi-instance learning in end-to-end training.

What many of the aforementioned methods have in common is that several simple components contribute to increased performance: (i) information at multiple scales [179, 11, 166, 97], (ii) contour orientation [67, 96, 11, 118], and (iii) end-to-end deep learning [219, 97]. COB is able to combine all of the above in a single pass of a CNN, producing an output that is richer than a linear combination of cues at different scales.

At the core of all these deep learning approaches lies a *base CNN*, starting from the seminal AlexNet [103] (8 layers), through the more complex VGGNet [199] (16 layers) and inception architecture of GoogLeNet [202] (22 layers), to the very recent and very deep ResNets [74] (up to 1001 layers). Image classification results, which originally motivated these architectures, have been continuously improved by exploring deeper and more complex networks. In this chapter, we present results both using VGGNet and ResNet, showing that COB is modular and can incorporate and benefit from future improvements in the base CNN.

Recent work has also explored weakly supervised or unsupervised deep learning of contours: Khoreva et al. [90] learn from the results of generic contour detectors coupled with object detectors; and Li et al. [115] train contour detectors from motion boundaries acquired from video sequences. Yang et al. [225] use Conditional Random Fields

(CRFs) to refine the inaccurately localized boundary annotations of PASCAL. Some works shift the domain of contours detection from abstract perceptual grouping to better defined tasks such as semantic or object contour detection [69, 90, 225]. Some methods also combine RGB-D cues for contour detection [65, 66, 48]. Extensive experiments on such benchmarks show that COB has an excellent performance even when shifting domains, showing state-of-the-art performance also in these new situations.

Hierarchical Image Segmentation and Grouping: One of the most studied category of methods for image segmentation are spectral methods, that rely on the generalized eigenvalue problem to solve a low-level pixel grouping problem. Notable approaches that fall into this category are Normalized Cuts [196], PMI [84], gPb [11], MCG [166]. Arbeláez et al. [11] showed the usefulness for jointly optimizing contours and regions (The duality between contours and regions was first studied by Najman and Schmitt [150]). Pont-Tuset et al. [166] leveraged multi-resolution contour detection and proved its interest for generating object proposals. COB also exploits the duality between contour detection and segmentation hierarchies. We differentiate from previous approaches mainly in two aspects. First, our sparse boundary representation translates into a clean and highly efficient implementation of hierarchical segmentation. Second, by leveraging high-level knowledge from the CNNs in the estimation of contour strength and orientation, our method benefits naturally from global information, which allows bypassing the globalization step (output of normalized cuts), a bottleneck in terms of computational cost, but a cornerstone of previous approaches.

Current lines of work: After the conference [138]/journal [139] versions of COB, several interesting works were published. Liu et al. [125] study how to make better use of intermediate feature-maps for edge prediction. Kong and Fawlkes learn pixel-level grouping by using a differentiable version of mean-shift, an idea closely related to [134] who learn affinities end-to-end. Other works focus on learning semantic edge detection in an end-to-end manner [228, 1] and outperform the results that we obtain in this chapter. The authors of [93] approach semantic instance segmentation from semantic segmentation by using region hierarchies. Related to learning of boundary orientation, [212] learn boundary orientations together with their strength, in order to assign foreground-background identities to the objects the boundaries belong to. In our contribution for video object segmentation [25, 136]

we show how performance can be further enhanced by snapping preliminary estimations to COB superpixels, exactly as shown in this chapter.

2.3 DEEP MULTISCALE ORIENTED CONTOURS

CNNs are by construction multi-scale feature extractors. If one examines the standard architecture of a CNN consisting of convolutional and spatial pooling layers, it becomes clear that as we move deeper, feature maps capture more global information due to the decrease in resolution. For contour detection, this architecture implies local and fine-scale contours at shallow levels, coarser spatial resolution and larger receptive fields for the units when going deeper and, consequently, more global information for predicting boundary strength and orientation. CNNs have therefore a built-in globalization strategy for contour detection, analogous to the hand-engineered globalization of contour strength through spectral graph partitioning in [11, 166].

Figure 2.2 depicts how we make use of information provided by the intermediate layers of a CNN to detect contours and their orientations at multiple scales. Different groups of feature maps contain different, scale-specific information, which we combine to build a multiscale oriented contour detector. The remainder of this section is devoted to introducing the recent approaches to contour detection using deep learning, to presenting our CNN architecture to produce contour detection at different scales, and to explaining how we estimate the orientation of the edges; all in a single CNN forward pass at the image level.

Training deep contour detectors: The recent success of [219] is based on a CNN to accurately regress the contours of an image. Within this framework, the idea of employing a CNN in an image-to-image fashion without any post-processing has proven successful, and lead to a big leap in performance for the task of contour detection. Their network, HED, produces scale-specific contour images (side outputs) for different scales of a network, and combines their activations linearly to produce a contour probability map. Using the notation of the authors, we denote the training dataset by $S = \{(X_n, Y_n), n = 1, \ldots, N\}$, with X_n being the input image and $Y_n = \{y_j^{(n)}, j = 1, \ldots, |X_n|\}, y_j^{(n)} \in \{0, 1\}$ the predicted

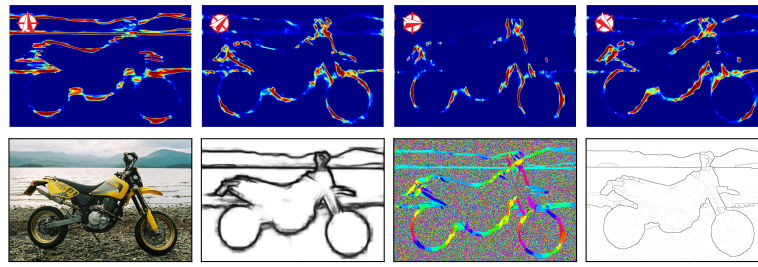


Figure 2.3: **Illustration of contour orientation learning:** Row 1 shows the responses B_k for 4 out of the 8 orientation bins. Row 2, from left to right: original image, contour strength, learned orientation map into 8 orientations, and hierarchical boundaries.

pixelwise labels. For simplicity, we drop the subscript n. Each of the M side outputs minimizes the objective function:

$$\ell_{side}^{(m)}\left(\mathbf{W},\mathbf{w}^{(m)}\right) = -\beta \sum_{j\in Y_+} \log P\left(y_j = 1|X;\mathbf{W},\mathbf{w}^{(m)}\right) \\ -(1-\beta)\sum_{j\in Y_-}\log P\left(y_j=0|X;\mathbf{W},\mathbf{w}^{(m)}\right) \tag{2.1}$$

where $\ell_{side}^{(m)}$ is the loss function for scale $m \in \{1,\ldots,M\}$, $\mathbf{W}$ denotes the standard set of parameters of the CNN, and $\{\mathbf{w}^{(m)}, m = 1,\ldots,M\}$ the corresponding weights of the the m-th side output. The multiplier β is used to handle the imbalance of the substantially greater number of background compared to contour pixels. Y_+ and Y_- denote the contour and background sets of the ground-truth Y, respectively. The probability $P(\cdot)$ is obtained by applying a sigmoid $\sigma(\cdot)$ to the activations of the side outputs $\hat{A}_{side}^{(m)} = \{a_j^{(m)}, j = 1,\ldots,|Y|\}$. In HED, the activations are finally fused linearly, as: $\hat{Y}_{fuse} = \sigma\left(\Sigma_{m=1}^{M} h_m \hat{A}_{side}^{(m)}\right)$ where $\mathbf{h} = \{h_m, m = 1,\ldots,M\}$ are the fusion weights. The fusion output is also trained to resemble the ground-truth applying the same loss function of Equation 2.1, by optimizing the complete set of parameters, including the fusion weights $\mathbf{h_m}$. We instead take advantage of the common CNN architectures to regress both the strength of the coarse and detailed (fine) contours, as well as the contour orientations. COB combines these output channels non-linearly to a single hierarchical segmentation. Inside this segmentation, the placement of each region in the hierarchy is determined by the strength of the boundaries to the neighbouring

regions. All in all, COB efficiently combines contour strengths and orientations into a segmentation hierarchy which can further facilitate high-level vision tasks related to segmented object proposals. In the rest of the chapter we use the class-balancing cross-entropy loss function of Equation 2.1.

Multiscale contours: We start from a deep network pre-trained on ImageNet [188], such as VGG [199] or ResNet [74]. The fully connected layers used for classification are removed, and so are the batch normalization layers, since we operate on one image per iteration. Therefore, the network consists mainly of convolutional layers coupled with ReLU activations, divided into 5 stages. We will refer to this architecture as the *base CNN* of our implementation. Each stage is handled as a different scale, since it contains feature maps of a similar size. At the end of a stage, there is a max pooling layer, which reduces the spatial dimensions of the produced feature maps to a half. As discussed before, the CNN naturally contains multiscale information, which we exploit to build a multiscale contour regressor.

We separately supervise the output of the last layer of each stage (side activation), comparing it to the ground truth using the loss function of Equation 2.1. This way, we enforce each side activation to produce an intermediate contour map at different resolution. The idea of supervising intermediate parts of a CNN has successfully been used in previous approaches, for a variety of tasks [202, 108, 219]. In the 5-scale base CNN illustrated in Figure 2.2, we linearly combine the side activations of the 4 finest and 4 coarsest scales to a fine-scale and a coarse-scale output ($\hat{Y}_{fine}$ and $\hat{Y}_{coarse}$, respectively) with trainable weights. The finer scale contains better localized contours, whereas the coarse scale leads to less noisy detections. To train the two sets of weights of the linear combinations, we freeze the pre-trained weights of the base CNN.

Estimation of Contour Orientations: In order to predict accurate contour orientations, we propose an extension of the CNN that we use to predict contour strength. We define the task as pixel-wise image-to-image multiscale classification into K bins. We connect K different branches (sub-networks) to the base network, each of which is associated with one orientation bin, and has access to feature maps that are generated from the intermediate convolutional layers at M different scales. We assign the parts of the CNN associated with each orientation a different task from the base network: classify the pixels of the contours that match a specific orientation. In order to design

these orientation-specific subtasks, we classify each pixel of the human contour annotations into K different orientations. The orientation of each contour pixel is obtained by approximating the ground-truth boundaries with polygons, and assigning each pixel the orientation of the closest polygonal segment, as shown in Figure 2.5. As in the case of multiscale contours, the weights of the base network remain frozen when training these sub-networks.

Each sub-network consists of M convolutional layers, each of them appended on different scales of the base network. Thus we need $M * K$ additional layers. In our setup, we use $K = 8$ and $M = 5$. All K orientations are regressed in parallel, and since they are associated with a certain angle, we post-process them to obtain the orientation map. Specifically, the orientation map is obtained as:

$$O(x,y) = \mathcal{T}\left(\arg\max_k B_k(x,y)\right), k = 1,\ldots,K \tag{2.2}$$

where $B_k(x,y)$ denotes the response of the k-th orientation bin of the CNN at the pixels with coordinates (x,y) and $\mathcal{T}(\cdot)$ is the transformation function which associates each bin with its central angle. For the cases where two neighboring bins lead to strong responses, we compute the angle as their weighted average. At pixels where there is no response for any of the orientations, we assign random values between 0 and π, not to bias the orientations. The different orientations as well as the resulting orientation map (color-coded) are illustrated in Figure 2.3.

In [11, 48, 166] the orientations are computed by means of local gradient filters. In Section 2.5 we show that our learned orientations are significantly more accurate and lead to better region segmentations.

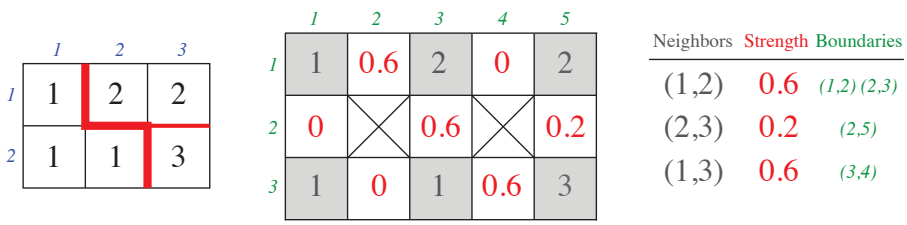
2.4 FAST HIERARCHICAL REGIONS

This section is devoted to building an efficient hierarchical image segmentation algorithm from the multiscale contours and the orientations extracted in the previous section. We build on the concept of Ultrametric Contour Map (UCM) [11], which transforms a contour detection probability map into a hierarchical boundary map, which gets partitions at different granularities when thresholding at various contour strength values. Despite the success of UCMs, their low speed limits

their applicability. We address this issue by using an alternative representation of an image partition which reduces the computation time of UCMs by an order of magnitude.

Sparse Boundary Representation of Hierarchies of Regions: An image partition is a clustering of the set of pixels into different sets, which we call regions. The most straightforward way of representing it in a computer is by a matrix of labels, as in the example in Figure 2.4(a), with three regions on an image of size 2×3. The boundaries of this partition are the edge elements, or *edgels*, between the pixels with different labels (highlighted in red). We can assign different *strengths* to these boundaries (thicknesses of the red lines), which indicate the *confidence* of that piece of being a boundary. By iteratively *erasing* these boundaries in order of increasing strength we obtain different partitions, which we call *hierarchy of regions*, or Ultrametric Contour Maps.

These boundaries are usually stored in the *boundary grid* (Figure 2.4(b)), a matrix of double the size of the image (minus one), in which the odd coordinates represent pixels (gray areas), and the positions in between represent boundaries (red numbers) and junctions (crossed positions). UCMs use this representation to store their boundary *strength* values, that is, each boundary position stores the threshold value beyond which that edgel *disappears* and the two neighboring regions merge. This way, by simply *binarizing* a UCM we have a partition represented as a boundary grid. Continuing with the example in Figure 2.4, binarizing the UCM at 0.5 the edge between region 2 and 3 would disappear, that is, 2 and 3 would merge and create a new region.



(a) Pixel labeling (b) Boundary grid (c) Sparse boundaries

Figure 2.4: **Image Partition Representation:** (a) Pixel labeling, each pixel gets assigned a region label. (b) Boundary grid, markers of the boundary positions. (c) Sparse boundaries, lists of boundary coordinates between neighboring regions.

This representation becomes very inefficient at run time, where the percentage of *activated* boundaries is very sparse. Not only are we wasting memory by storing those *empty* boundaries, but it also makes operating on them very inefficient by having to *sweep* over the entire matrix to perform a modification on a single boundary piece.

Inspired by how sparse matrices are handled, we designed the *sparse boundaries* representation (Figure 2.4(c)). It stores a look-up table for pairs of neigboring regions, their boundary strength, and the list of coordinates the boundary occupies. Apart from being more compact in terms of memory, this representation enables efficient operations on specific pieces of a boundary, since one only needs to perform a search in the look-up table and scan the activated coordinates; instead of sweeping the whole boundary grid.

Fast Hierarchies from Multiscale Oriented Contours: We are inspired by the framework proposed in [166], in which a UCM is obtained from contours computed at different image scales and then combined into a single hierarchy. The motivation behind this work is that the UCMs obtained from downscaled images will focus on the coarse structures and ignore textures, so their localization accuracy will decrease. On the other hand, upscaled images will bring very good localization in the boundaries, but it will be harder to distinguish between the high- and low-level contents. To bring the best of the two worlds, [166] progressively *projects* the coarse hierarchies into the finer ones by adapting the high-level contours into the better localized ones. The final hierarchy keeps the high-level information while being snapped to the correctly localized low-level boundaries.

The deep CNN presented in Section 2.3 provides different levels of detail for the image contours, so instead of processing the image at multiple resolutions we use the different outputs that are computed in a single pass of the CNN to obtain different hierarchies that focus on high- and low-level features.

A drawback of the original framework [166], however, is that the manipulation of the hierarchies and their projection to different scales is very slow (in the order of seconds), so the operations on the UCMs had to be performed at a small subset of the contour strengths (from thousands to a few dozens). By using the fast sparse boundary representation, we can operate on all thousands of contour strengths, yielding better results at a fraction of the original cost. Moreover, we use the

learned contour orientations for the computation of the Oriented Watershed Transform (OWT) [11], further boosting performance.

2.5 EXPERIMENTS ON LOW-LEVEL APPLICATIONS

This section presents the empirical evidence that supports our approach for low-level applications (image segmentation and contour detection). First, Section 2.5.1 explores ablated and baseline techniques in order to isolate and quantify the improvements due to different components of our system. Section 2.5.2 further analyzes and evaluates the proposed contour orientations. In Section 2.5.3, Section 2.5.4, and Section 2.5.5 we compare our results against the state of the art in generic RGB image segmentation, RGB object boundary detection, and RGB-D image segmentation, respectively. In all three cases, we obtain the best results to date by a significant margin. Finally, Section 2.5.6 analyzes the effect of the various components in terms of speed on COB.

In terms of datasets, we extend the main BSDS benchmark [143] to the PASCAL Context dataset [147], which contains carefully localized pixel-wise semantic annotations for the entire image on the PASCAL VOC 2010 detection train-val set. This results in 459 semantic categories across 10 103 images, which is an order of magnitude (20×) larger than the BSDS. In order to allow training and optimization of large capacity models, we split the data into train, validation, and test sets as follows: *VOC train* corresponds to the official PASCAL Context train with 4 998 images, *VOC val* corresponds to half the official PASCAL Context validation set with 2 607 images and *VOC test* corresponds to the second half with 2 498 images. In the remainder of the chapter, we refer to this dataset division. Note that, in contrast to the BSDS, in this setting boundaries are defined between different semantic categories and not between their parts.

In all our experiments for boundary detection and image segmentation, we used the standard evaluation benchmark evaluating boundaries (F_b [143]) and regions (F_{op} [167]). Through the literature, the tolerance in the boundary localization metric F_b is altered (the `maxDist` parameter), depending on the database and the quality of the annotations. To avoid confusion, we list the value of this parameter for all our experiments in Table 2.1. Please also note that methods that produce open contours instead of regions can not be evaluated using the region measure F_{op}. In all the produced curves, markers indicate the optimal operating point

that maximizes F_b and F_{op}. We used the publicly available *Caffe* [86] framework for training and testing CNNs, and all the state-of-the-art results are computed using the publicly-available code provided by the respective authors.

Training details: In our two-step training approach, we first train the base networks for the task of contour detection (coarse and fine). We use stochastic gradient descent with a momentum of 0.9 and weight decay of 0.0002 for 40k iterations. The base learning rate is set to 10^{-6}, and is divided by 10 after 30k iterations. After the first step is finished, the weights of the base network are frozen, and the layers of the orientation sub-network are connected and trained for an additional 10k iterations. Depending to the size of dataset we use different data augmentation strategies: flipping and rotation into 4 angles for PASCAL and NYUD; flipping, rotation into 16 angles, and scaling into 3 scales [219] for BSDS500. In all cases, we initialize the network from ImageNet pre-trained weights. The same ground-truth boundaries are used for training both the fine and the coarse contours.

Database	Task	train	test	`maxDist`
BSDS500	Generic Segmentation	300	200	0.0075
VOC Context	Generic Segmentation	7 605	2 498	0.0075
VOC'12 Segm.	Object Contours	1 464	1 449	0.01
NYUD	RGB-D Segmentation	795	654	0.011
SBD	Semantic Contours	8 498	2 857	0.02
VOC'12 Segm.	Semantic Segmentation	1 464	1 449	-
COCO	Object Proposals	-	40 504	-
VOC'07	Object Detection	5 011	4 952	-

Table 2.1: **Datasets and parameters for boundary detection:** The list of databases used to evaluate our approach on various low-level and high-level tasks. We report the number of images used for training and testing our algorithm, along with the tolerance for contour localization used in the literature, when applicable. In all our experiments, we keep those numbers unchanged.

2.5.1 *Control Experiments/Ablation Analysis*

This section presents the control experiments and ablation analysis to assess the performance of all subsystems of our method. We train on *VOC train*, and evaluate on *VOC val* set. We report the standard F measure at Optimal Dataset Scale (ODS) and Optimal Image Scale (OIS), as well as the Average Precision (AP), both evaluating boundaries (F_b [143]) and regions (F_{op} [167]).

Table 2.2 shows the evaluation results of the different variants, highlighting whether we include globalization and/or trained orientations. As a first baseline, we test the performance of MCG [166], which uses Structured Edges [48] as input contour signal. We then substitute SE by the newer HED [219], trained on *VOC train* as input contours and denote it MCG-HED. Note that the aforementioned baselines require multiple passes of the contour detector (3 scales).

In the direction of using the side outputs of the base CNN architecture as multiscale contour detections in one pass, we tested the baseline of naively taking the 5 side outputs directly as the contour detections. We trained both VGGNet [199] and ResNet50 [74] on *VOC train* and combined the 5 side outputs with our fast hierarchical regions of Section 2.4 (VGGNet-Side and ResNet50-Side).

We finally evaluate different variants of our system, as presented in Section 2.3. We first compare our system with two different base architectures: Ours (VGGNet) and Ours (ResNet50). We observe that the deeper architecture of ResNet translates into better boundaries and regions. Using the even deeper counterparts of ResNet lead to negligible gain in accuracy while significantly sacrificing speed.

We then evaluate the influence of our trained orientations and globalization, by testing the four possible combinations (the orientations are further evaluated in the next section). Our method using ResNet50 together with trained orientations leads to the best results both for boundaries and for regions. The experiments also show that, when coupled with trained orientations, globalization even decreases performance, so we can safely remove it and get a significant speed up. This behaviour arises from the fact that the image-to-image architecture of the base CNN already captures global information, addressing issues that could not be handled by local approaches, e.g., deleting internal contours of objects. Our technique with trained orientations and with-

out globalization is therefore selected as our final system and will be referred to in the sequel as Convolutional Oriented Boundaries (COB).

2.5.2 *Contour Orientation*

We evaluate contour orientation results by the classification accuracy into 8 different orientations, to isolate their performance from the global system. We compute the ground-truth orientations as depicted in Figure 2.5 by means of the sparse boundaries representation. We then sweep all ground-truth boundary pixels and compare the estimated orientation with the ground-truth one. Since the orientations are not well-balanced classes (much more horizontal and vertical contours), we compute the classification accuracy per each of the 8 classes and then compute the mean.

Figure 2.6 shows the classification accuracy with respect to the confidence of the estimation. We compare our proposed technique against the local gradient estimation used in previous literature [11, 48, 166]. As a baseline, we plot the result a random guess of the orientations would get. We observe that our estimation is significantly better than the previous approach. As a summary measure, we compute the area under the curve of the accuracy (ours 58.6%, local gradients 41.2%, random 12.5%), which corroborates the superior results from our technique.

			Boundaries - F_b			Regions - F_{op}		
Method	Global.	Orient.	ODS	OIS	AP	ODS	OIS	AP
MCG [166]	✓	✗	0.548	0.594	0.519	0.355	0.419	0.263
MCG-HED	✓	✗	0.691	0.727	0.693	0.459	0.520	0.374
VGGNet-Side	✓	✗	0.644	0.683	0.664	0.439	0.505	0.351
ResNet50-Side	✓	✗	0.676	0.711	0.681	0.456	0.521	0.374
Ours (VGGNet)	✗	✓	0.705	0.735	0.741	0.466	0.533	0.384
Ours (ResNet50)	✗	✗	0.734	0.767	0.757	0.475	0.545	0.405
Ours (ResNet50)	✓	✗	0.726	0.759	0.725	0.461	0.531	0.395
Ours (ResNet50)	✓	✓	0.732	0.763	0.731	0.481	**0.554**	**0.418**
Ours (ResNet50)	✗	✓	**0.737**	**0.768**	**0.758**	**0.483**	0.553	0.417

Table 2.2: **Ablation analysis on *VOC val*:** Comparison of different ablated and baseline versions of our system.

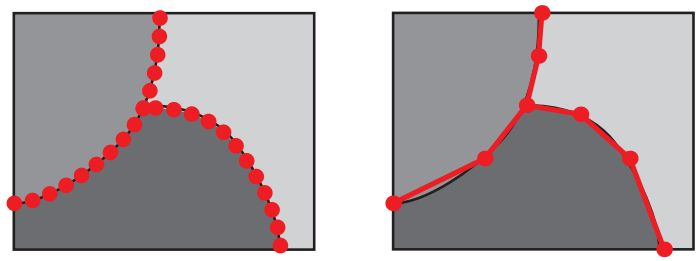


Figure 2.5: **Polygon simplification:** From all boundary points (left) to simplified polygons (right), which are used to compute the ground-truth orientation robustly.

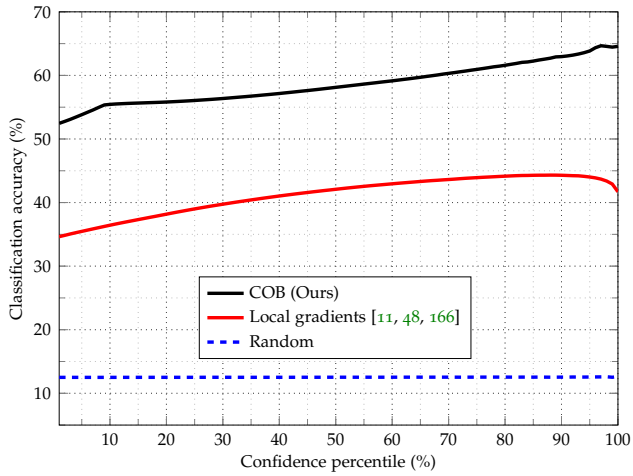
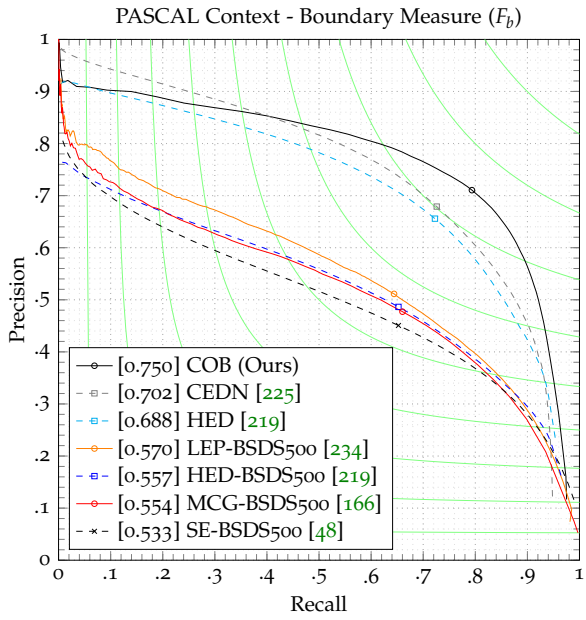


Figure 2.6: **Contour orientation:** Classification accuracy into 8 bins.

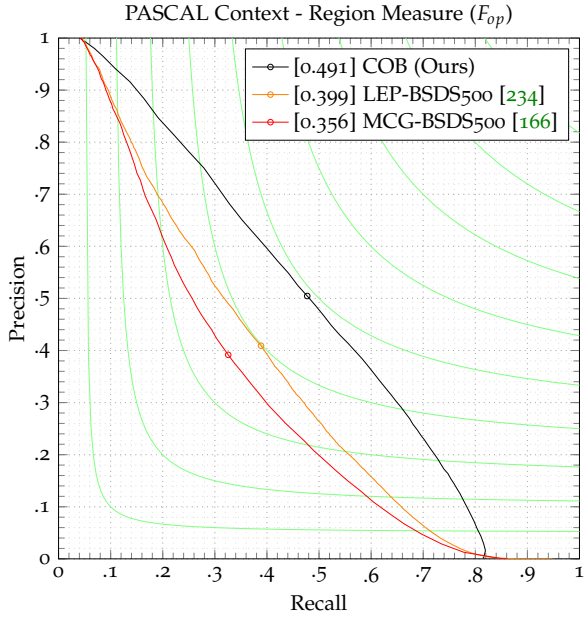
2.5.3 *Generic Image Segmentation*

We present our results for contour detection and generic image segmentation on PASCAL Context [147] as well as on the BSDS500 [144], which is the most established benchmark for perceptual grouping.

PASCAL Context: We train COB in the *VOC train*, and perform hyper-parameter selection on *VOC val*. We report the final results on the unseen *VOC test* when trained on *VOC trainval*, using the previously tuned hyper-parameters. We compare our approach to several methods trained on the BSDS [48, 166, 234, 219] and we also retrain the current state-of-the-art contour detection methods HED [219] and the recent

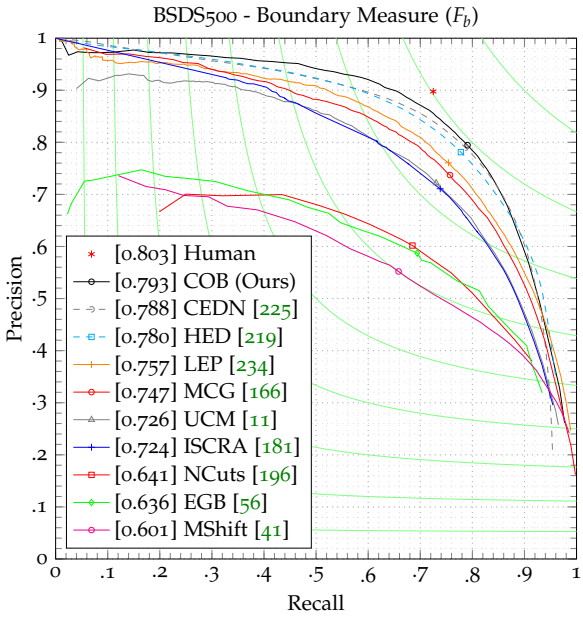


Method	Boundaries - F_b ODS	OIS	AP
COB	**0.750**	**0.781**	**0.773**
CEDN [225]	0.702	0.718	0.744
HED [219]	0.688	0.707	0.704
LEP-B [234]	0.570	0.636	0.547
HED-B [219]	0.557	0.588	0.518
MCG-B [166]	0.554	0.609	0.528
SE-B [48]	0.533	0.568	0.496

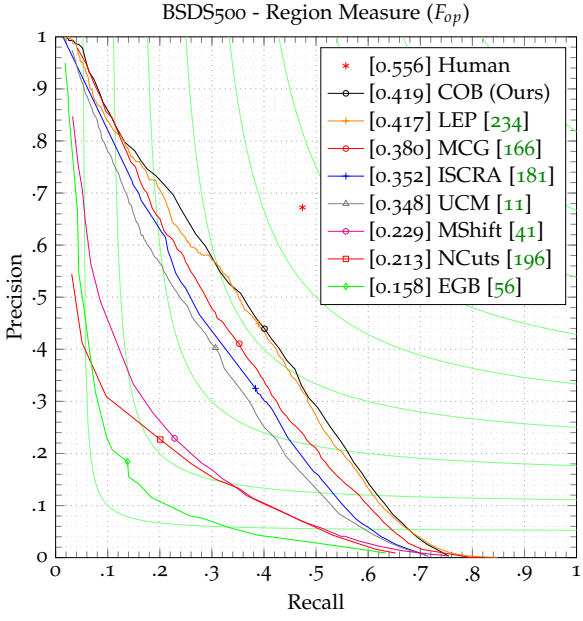


Method	Regions - F_{op} ODS	OIS	AP
COB	**0.491**	**0.565**	**0.439**
LEP-B [234]	0.399	0.461	0.304
MCG-B [166]	0.356	0.419	0.275

Figure 2.7: **PASCAL Context *VOC test* Evaluation:** Precision-recall curves for evaluation of boundaries (F_b [143]), and regions (F_{op} [167]). Open contour methods in dashed lines and closed boundaries (from segmentation) in solid lines. ODS, OIS, and AP summary measures. Markers indicate the optimal operating point, where F_b and F_{op} are maximized.



Method	Boundaries - F_b ODS	OIS	AP
COB (Ours)	**0.793**	**0.820**	**0.859**
CEDN [225]	0.788	0.804	0.834
HED [219]	0.780	0.796	0.834
LEP [234]	0.757	0.793	0.828
MCG [166]	0.747	0.779	0.759
UCM [11]	0.726	0.760	0.727
ISCRA [181]	0.724	0.752	0.783
NCuts [196]	0.641	0.674	0.447
EGB [56]	0.636	0.674	0.581
MShift [41]	0.601	0.644	0.493



Method	Regions - F_{op} ODS	OIS	AP
COB (Ours)	**0.419**	**0.478**	**0.343**
LEP [234]	0.417	0.468	0.334
MCG [166]	0.380	0.433	0.271
ISCRA [181]	0.352	0.418	0.275
UCM [11]	0.348	0.385	0.235
MShift [41]	0.229	0.292	0.122
NCuts [196]	0.213	0.270	0.096
EGB [56]	0.158	0.240	0.080

Figure 2.8: **BSDS500 Test Evaluation:** Precision-recall curves for evaluation of boundaries (F_b [143]), and regions (F_{op} [167]).

CEDN [225] on *VOC trainval* using the code provided by the respective authors.

Figure 2.7 presents the evaluation results of COB compared to the state of the art, showing that it outperforms all others by a considerable margin both in terms of boundaries and in terms of regions. The lower performance of the methods trained on the BSDS quantifies the difficulty of the task when moving to a larger and more challenging dataset.

BSDS500: We retrain COB using only the 300 *trainval* images of the BSDS, after data augmentation as suggested in [219], keeping the architecture decided in Section 2.5.1. For comparison to HED [219], we used the model that the authors provide online. We also compare with CEDN [225], by evaluating the results provided by the authors.

Figure 2.8 presents the evaluation results, which show that we also obtain state-of-the-art results in this dataset. The smaller margins are in all likelihood due to the fact that we almost reach human performance for the task of contour detection on the BSDS, which motivates the shift to PASCAL Context to achieve further progress in the field.

Qualitative Results: Figure 2.9 shows some qualitative results of our hierarchical contours. Please note that COB is capable of correctly distinguishing between internal contours and external, semantically meaningful boundaries.

2.5.4 *Object boundary detection*

Concurrent works presented results on object boundary detection [225, 90] on the PASCAL VOC'12 Segmentation database. The database consists of 1464 training and 1449 validation images, including pixel-wise annotations of the instances and the semantic classes of the objects. The goal is to detect the boundaries of the objects that belong to the 20 classes of PASCAL, without distinguishing the semantics. Different from generic image segmentation, boundaries that do not belong to an object are treated as background.

We retrain COB on VOC'12 train set and report the results on the validation set. We use the instance level annotation of the database, and extract contours from the semantic segmentation annotations of the database. The uncertain areas (annotated with value of 255) are treated as background. We compare to several baselines, together with recent state-of-the-art results. Specifically, Khoreva et al. [90] retrained

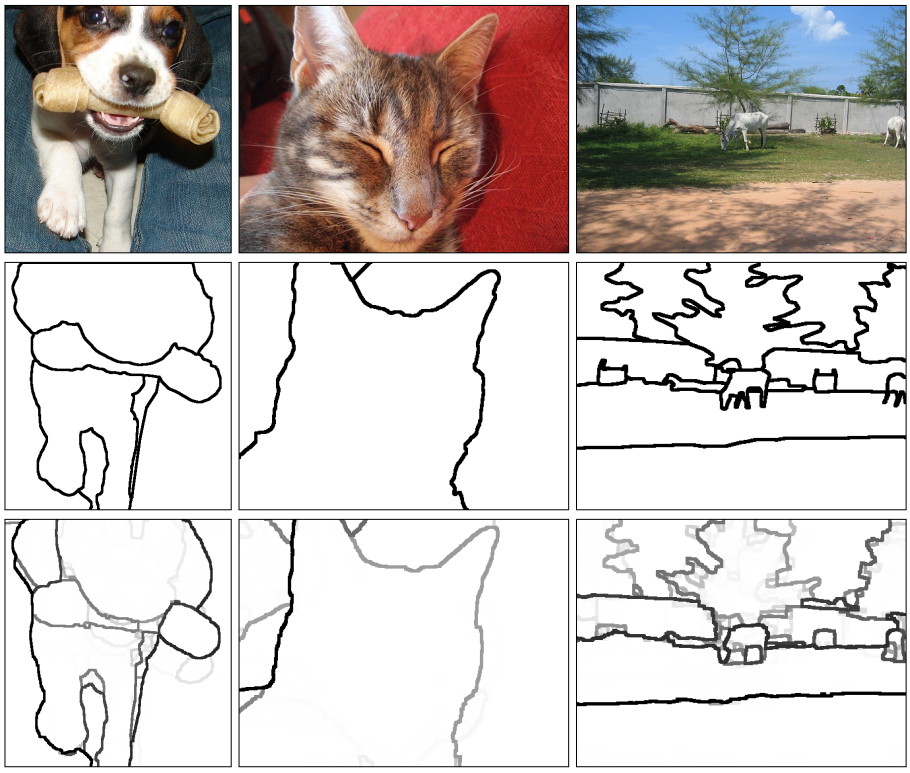


Figure 2.9: **Qualitative results on PASCAL - Hierarchical Regions:** Row 1: original images, Row 2: ground-truth boundaries, Row 3: hierarchical regions with COB.

HED [219] on object contours, and Yang et al. [225] proposed a novel encoder-decoder architecture to tackle the same task. We evaluate the best pre-computed results provided by the authors in both cases. The results are quantified in Figure 2.11. We observe that COB obtains state-of-the-art results in all metrics. CEDN [225] performs better in the high precision regime. However, the authors used extra images from the SBD dataset [69] for training their detector. Also, CEDN is trained on an improved version of the ground truth, aligning the uncertain areas of VOC'12 with the the true image boundaries by using a CRF. We report results of COB trained only on VOC'12 train set, to be consistent with the results of Khoreva et al. [90]. In this experiment, we use `maxDist` of 0.01, as is adopted by the literature [210, 90].

Figure 2.10 illustrates some qualitative results, as well as the differences of generic segmentation and object boundary detection. We show

our results on images of the VOC'12 val set using the model trained on PASCAL Context for generic image segmentation, and we compare qualitatively to the model retrained on the 20 classes of VOC'12 for object boundary detection. In the latter case, the detections are focused on the 20 object classes, disregarding strong contour cues of the background that are detected by the generic segmentation model.

2.5.5 *RGB-D boundary detection on NYUD dataset*

The NYUD (v2) dataset [151] consists of 1449 RGB-D indoor images, divided into splits of 795 training and 654 testing images, with the

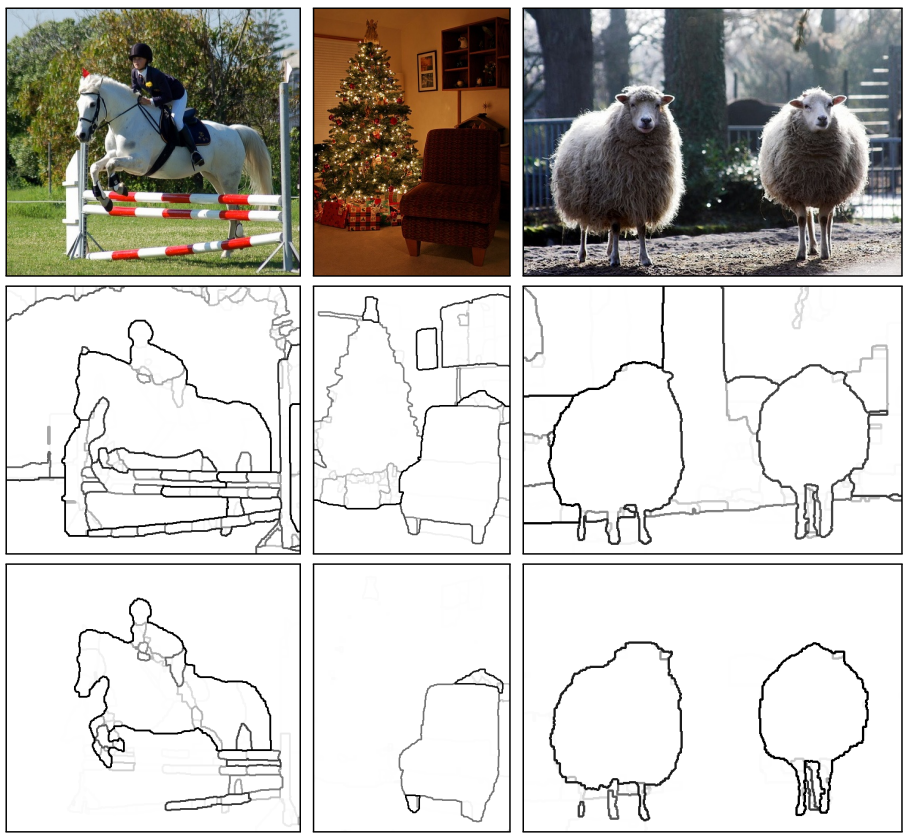
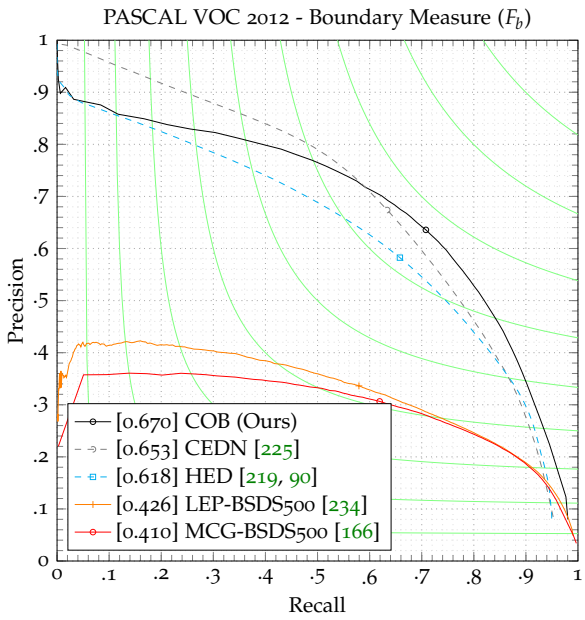
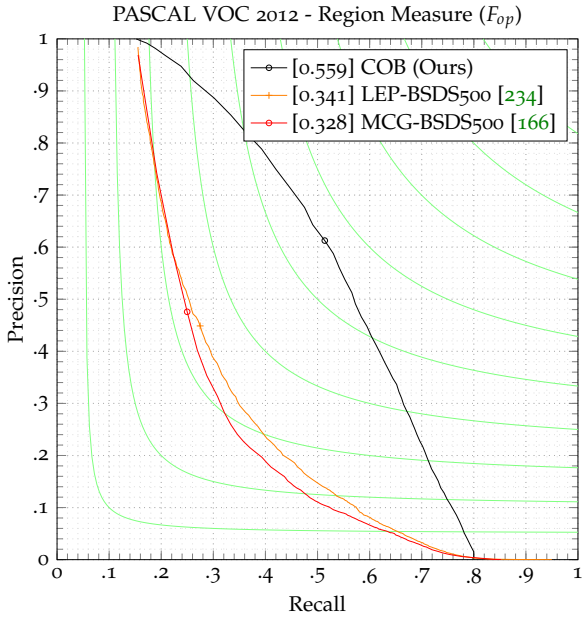


Figure 2.10: **Qualitative results for Object Boundaries:** Row 1: original images, Row 2: Generic Image Segmentation results, Row 3: Object Boundary results.

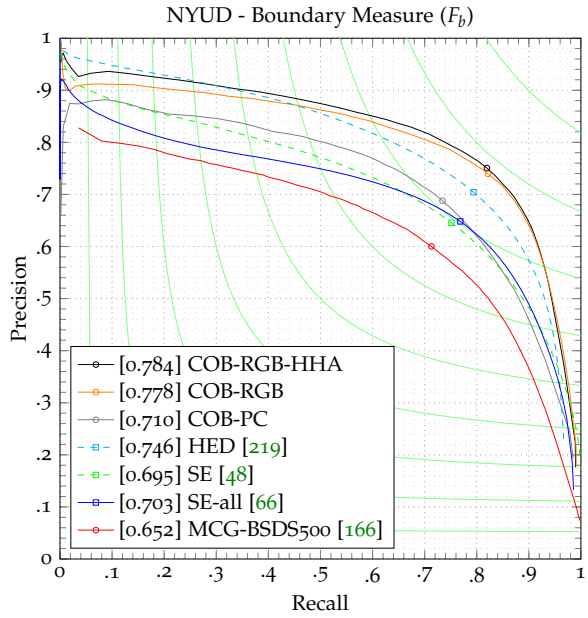


Method	Regions - F_{op} ODS	OIS	AP
COB (Ours)	**0.559**	**0.644**	**0.391**
LEP-BSDS500 [234]	0.341	0.456	0.156
MCG-BSDS500 [166]	0.328	0.431	0.140



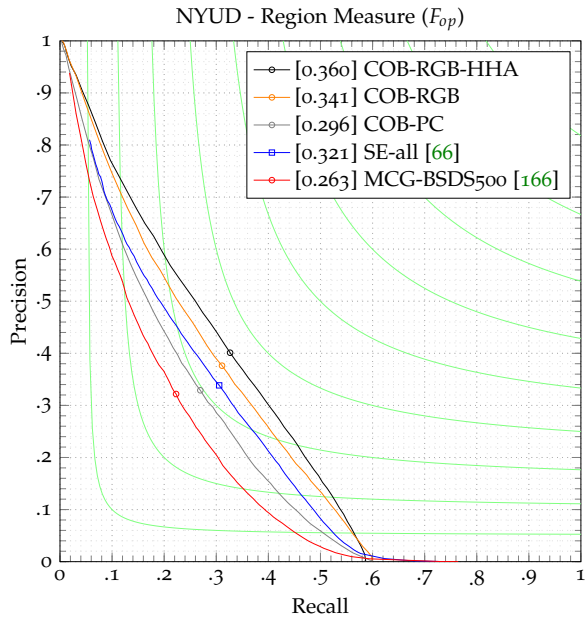
Method	Boundaries - F_b ODS	OIS	AP
COB (Ours)	**0.670**	**0.728**	**0.687**
CEDN [225]	0.653	0.683	0.686
HED [219, 90]	0.618	0.650	0.628
LEP-B [234]	0.426	0.517	0.325
MCG-B [166]	0.410	0.486	0.294

Figure 2.11: **Object Boundaries in PASCAL VOC 2012:** Precision-recall curves for boundaries (F_b [143]), and regions (F_{op} [167]). ODS, OIS, and AP summary measures.



Method	Boundaries - F_b ODS	OIS	AP
COB-RGB-HHA	**0.784**	**0.805**	**0.825**
COB-RGB	0.778	0.799	0.814
COB-PC	0.710	0.735	0.734
HED [219]	0.746	0.761	0.786
SE [48]	0.695	0.708	0.719
SE-all [66]	0.703	0.716	0.707
MCG-B [166]	0.652	0.681	0.613

Ablation: Method	Boundaries - F_b ODS	OIS	AP
ResNet50-RGB-HHA	0.745	**0.762**	**0.792**
ResNet50-RGB	**0.746**	0.761	0.789
ResNet50-RGBD	0.683	0.699	0.681



Method	Regions - F_{op} ODS	OIS	AP
COB-RGB-HHA	**0.360**	**0.398**	**0.270**
COB-RGB	0.341	0.376	0.252
COB-PC	0.296	0.332	0.203
SE-all [66]	0.321	0.354	0.179
MCG-BSDS500 [166]	0.263	0.296	0.159

Figure 2.12: **RGB-D Boundaries in NYUD test:** Precision-recall curves for evaluation of boundaries (F_b [143]), and regions (F_{op} [167]). ODS, OIS, and AP summary measures.

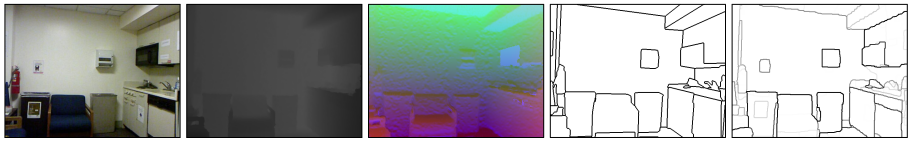


Figure 2.13: **Data and results on NYUD:** From left to right: RGB image, depth, HHA features [66], ground truth, and COB detections.

corresponding semantic and instance level segmentations. Gupta et al. [65] adopted this dataset for contour detection. In their experiments, they obtained the respective boundary annotations from the instance-level segmentations of the dataset. We evaluated the performance by using the standard benchmarks of BSDS. Following [48, 219, 66], we increased the tolerance for incorrect localizations from 0.0075 of the image diagonal to 0.11, to compensate for inaccurate annotations of boundaries.

We use the extra information of depth to train different variants of COB on the NYUD dataset. Gupta et al. [66] used the camera parameters of the images to encode the depth information in three channels: horizontal disparity, height above ground, and the angle of the local surface normal with the inferred gravity direction at each pixel (HHA). We retrain three different variants of the CNN: (a) Only using RGB data (ResNet50-RGB), (b) Incorporating depth information into a fourth channel (ResNet50-RGBD), and (c) Concatenating RGB and HHA channels and operate on 6 channels directly (ResNet50-RGB-HHA). Figure 2.13 illustrates an overview of the data, along with depth and HHA features that we used, as well as the results obtained by COB. In Figure 2.12 we show the ablation analysis by directly evaluating the CNN output, without any post-processing. We observe that the CNNs retrained on RGB and RGB-HHA channels obtain significantly better results than the one trained on RGB-D data, showing that HHA features provide an appropriate encoding for depth information. We retrain the full pipeline of COB (including orientations) on NYUD and we report the precision-recall curves. We compare with various state-of-the-art methods, showing significant improvements. Specifically, we compare with the SE [48] detector retrained on the RGB-D data of NYUD, the detector proposed by [66] trained on RGB and depth normal gradients, and the best result reported on NYUD

by HED [219], where the authors trained two different variants of the detector on RGB and HHA modalities respectively and averaged the obtained results. For completeness, we report results obtained by the original MCG [166] without any retraining on NYUD. The best result is obtained by the variant of COB trained on both RGB and HHA modalities. Compared to its RGB-only counterpart, the particular model achieves higher accuracy, suggesting that the depth embeddings are useful cues to discern contours when RGB modality alone is unable to do so. It is noteworthy that the post-processing step (orientations and UCMs) further boosts the performance of COB. For example, performance increases from 0.745 (ResNet50-RGB-HHA) to 0.784 (COB-RGB-HHA) by plugging in the orientations and the UCM pipeline to the trained ResNet50 architecture. We also report the results of the model trained on PASCAL Context (COB-PC) and operating only on RGB data, showing that it performs fairly well without any retraining on the NYUD dataset.

2.5.6 *Efficiency Analysis*

Contour detection and image segmentation, as a preprocessing step towards high-level applications, need to be computationally efficient. The previous state-of-the-art in hierarchical image segmentation [166, 11] was of limited use in practice due to its computational load.

As a core in our system, the forward pass of our network to compute the contour strength and 8 orientations takes 0.28 seconds on a NVidia Titan X GPU. Table 2.3 shows the timing comparison between the full system COB (Ours) and some related baselines on PASCAL Context. We divide the timing into different relevant parts, namely, the contour detection step, the Oriented Watershed Transform (OWT) and Ultrametric Contour Map (UCM) computation, and the globalization (normalized cuts) step.

Column (1) shows the timing for the original MCG [166], which uses Structured Edges (SE) [48]. As a first baseline, Column (2) displays the timing of MCG if we naively substitute SE by HED [219] at the three scales (running on a GPU). By applying the sparse boundaries representation we reduce the UCM and OWT time from 11.58 to 1.63 seconds (Column (3)). Our final technique COB, in which we remove the globalization step, computes the three scales in one pass and add contour orientations, takes 0.79 seconds in mean. Overall, comparing to

Steps	(1) MCG [166]	(2) MCG-HED	(3) Fast UCMs	(4) COB (Ours)
Contours	3.08	0.39*	0.39*	0.28*
OWT, UCM	11.33	11.58	1.63	0.51
Globalize	9.96	9.97	9.92	0.00
Total Time	24.37	21.94	11.94	**0.79**

Table 2.3: **Timing experiments for COB:** Comparing our approach to different baselines. Times computed using a GPU are marked with an asterisk. Numbers are in seconds.

previous state-of-the-art, we get a significant improvement at a fraction of the computation time (24.37 to 0.79 seconds).

2.6 EXPERIMENTS ON HIGH-LEVEL APPLICATIONS

This section is dedicated to present the interaction of COB boundaries and segments with higher vision tasks. In Section 2.6.1 we evaluate COB as object proposals by plugging in the detected UCMs into the combinatorial goruping pipeline of MCG [166]. In Section 2.6.2 we study the interplay of our boundary detector with semantic contours and semantic segmentation by combining COB with Dilated Network [227] and PSPNet [233], and in Section 2.6.3 we couple the COB proposals with the Fast-RCNN [61] pipeline for object detection. In all cases, we show that COB co-operates well with existing approaches by improving their performance.

2.6.1 *Object Proposals*

Object proposals are an integral part of current object detection and semantic segmentation pipelines [62, 61, 178], as they provide a reduced search space on locations, scales, and shapes over the image. This section evaluates COB as a segmented and bounding box proposal technique, when using our high-quality region hierarchies in conjunction with the combinatorial grouping framework of MCG [166]. In terms of segmented object proposals, we compare against the most recent techniques SharpMask [165], DeepMask [164], POISE [81], MCG and SCG [166], LPO [102], GOP [101], SeSe [209], GLS [173], and RIGOR [82]. In terms of bounding box proposals, we compare also against Sharp-

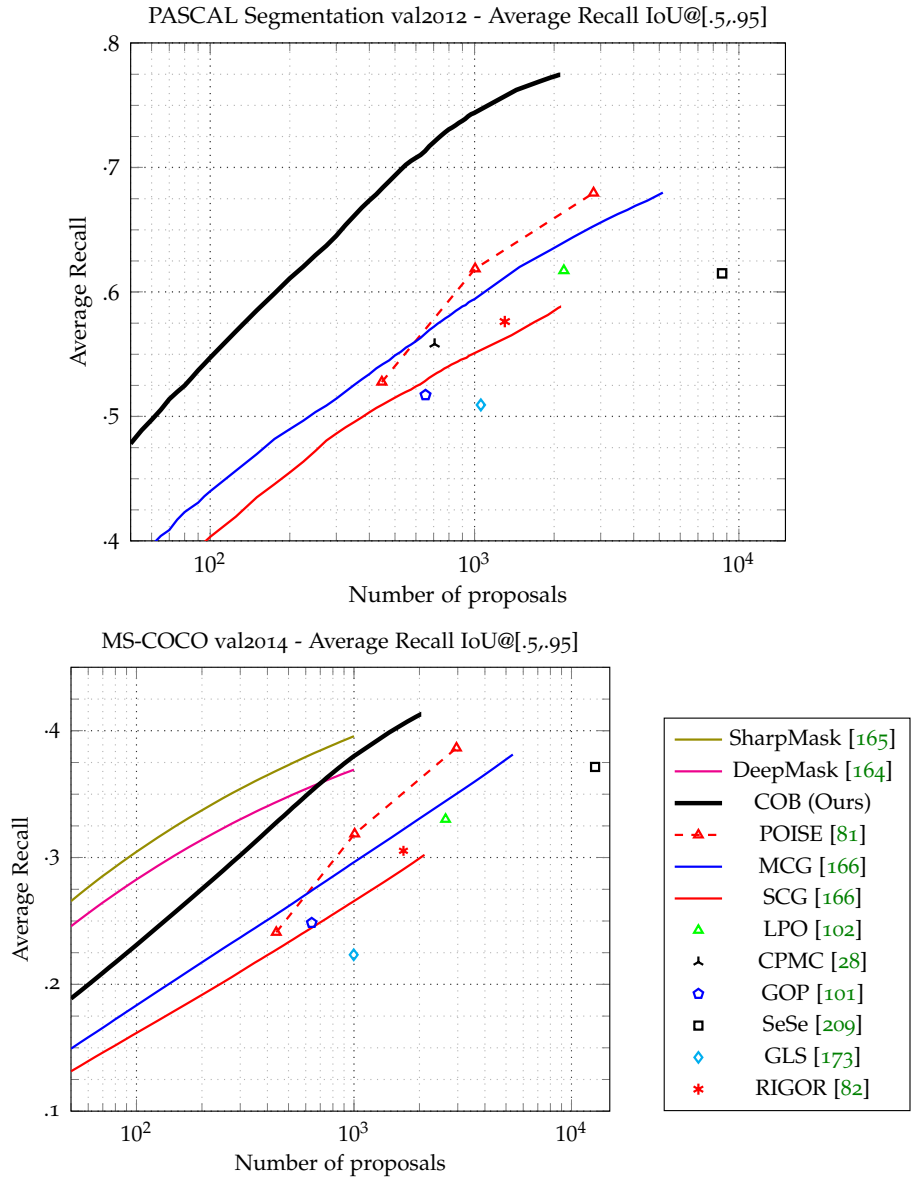


Figure 2.14: **Segmented object proposals evaluation in PASCAL Segmentation val and MS-COCO val:** Dashed lines refer to methods that do not provide a ranked set of proposals, but they need to be re-parameterized.

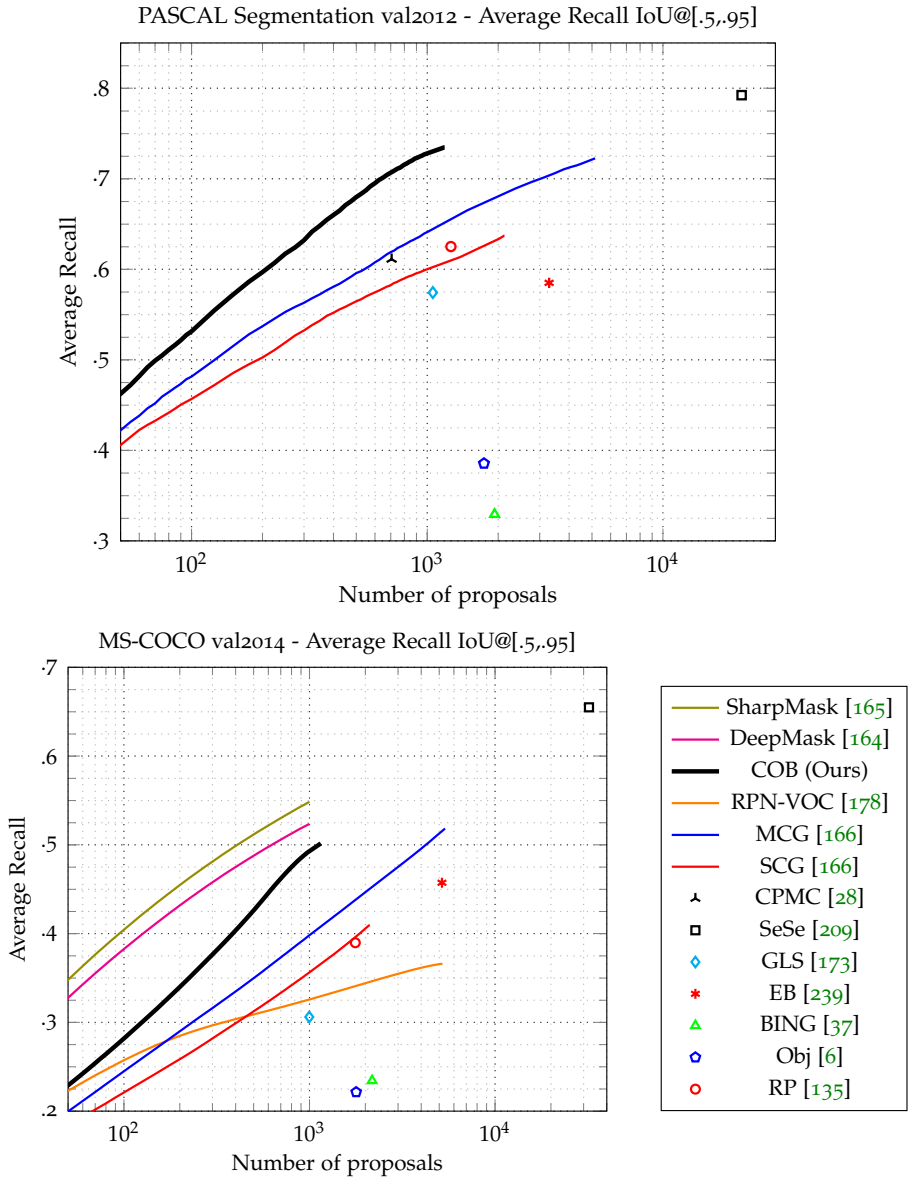


Figure 2.15: **Bounding-box object proposals evaluation on PASCAL Segmentation val and MS-COCO val:** Note that COB is designed to detect segmented object proposals and not bounding-box proposals.

Mask [165], DeepMask [164], EB [239], RPN [178] MCG and SCG [166], LPO [102], BING [37], SeSe [209], GLS [173], RIGOR [82], Obj [6], and RP [135]. Recent thorough comparisons of object proposal generation methods can be found in [169, 76].

We perform experiments on the PASCAL 2012 Segmentation dataset [54] and on the bigger and more challenging MS-COCO [122] (val2014 set). The hierarchies and combinatorial grouping are trained on PASCAL Context. To assess the generalization capability, we evaluate on MS-COCO, which contains a large number of previously unseen categories, without further retraining.

Figure 2.14 shows the average recall [76] with respect to the number of object proposals. In PASCAL VOC'12 Segmentation, the absolute gap of improvement of COB is at least of +13% with the second-best technique, and consistent in all the range of number of proposals. In MS-COCO, even though we did not train on any MS-COCO image, COB reaches competitive results for the task, with only very recent techniques [165, 164] reaching higher Average Recall when evaluating a low number of proposals. This shows that our contours, regions, and proposals are properly learning a generic concept of object rather than some specific categories.

Figure 2.15 shows the evaluation in terms of bounding box object proposals. COB is less competitive in terms of box proposals, however the algorithm was not specifically designed for detecting bounding boxes. We also show the comparison to RPN [178], which is trained on VOC'07, and thus does not generalize well in the classes of COCO.

2.6.2 *Semantic Boundaries and Semantic Segmentation*

The task of Semantic Boundaries, introduced by [69], requires not only detecting the boundaries, but also associating a semantic class to them. It can be thought as a combination of Boundary Detection and Semantic Segmentation, where except for the binary information of boundaries, one needs to label each of the detected pixels with the corresponding semantic class. The common approach to this task is to separately approach semantic segmentation and contour detection, and fuse the results of the two tasks [69, 18, 19]. Hariharan et al [69] tackled the task with generic object detectors and bottom up contours. Bertasius et al. [18, 19] show that results can be significantly improved when using deep-learning based semantic segmenters and contour detectors.

Technique	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Mean maxF
COB-dil	**84.2**	**72.3**	**81.0**	64.2	**68.8**	81.7	71.5	**79.4**	**55.2**	**79.1**	40.8	**79.9**	**80.4**	**75.6**	**77.3**	**54.4**	**82.8**	51.7	72.1	62.4	**70.7**
DilatedConv [227]	83.7	71.8	78.8	**65.5**	66.3	**82.6**	**73.0**	77.3	47.3	76.8	37.2	78.4	79.4	75.2	73.8	46.2	79.5	46.6	**76.4**	**63.8**	69.0
BNF [19]	76.7	60.5	75.9	60.7	63.1	68.4	62.0	74.3	54.1	76.0	**42.9**	71.9	76.1	68.3	70.5	53.7	79.6	**51.9**	60.7	60.9	65.4
HFL [18]	73.6	61.1	74.2	57.0	58.7	70.2	60.8	71.8	46.3	72.1	36.0	70.9	72.9	67.5	69.9	44.1	73.1	42.2	62.2	60.4	62.2
[90]	65.9	54.1	63.6	47.9	47.0	60.4	50.9	56.5	40.4	56.0	30.0	57.5	58.0	57.4	59.5	39.0	64.2	35.4	51.0	42.4	51.9
[69]	41.5	46.7	15.6	17.1	36.5	42.7	40.3	22.6	18.8	27.0	12.5	18.2	35.4	29.4	48.1	13.8	26.9	11.0	22.0	31.3	27.9

Table 2.4: **SBD val evaluation:** Semantic contours results: maximal F_b per class and mean maximal F_b is reported for all methods.

Technique	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Mean AP
COB-dil	**85.7**	**69.3**	**77.6**	**59.7**	**64.1**	**82.9**	**69.7**	**80.5**	**41.8**	**79.4**	**26.0**	**78.9**	**81.5**	**74.7**	**77.3**	**43.8**	**82.8**	**39.3**	**73.3**	**56.4**	**67.2**
BNF [19]	75.9	46.0	70.5	48.9	48.6	65.3	53.5	65.2	38.2	69.7	20.9	62.3	72.2	56.6	63.3	38.5	75.7	31.4	45.6	48.1	54.8
HFL [18]	71.3	54.9	68.8	45.6	48.3	70.9	56.5	65.6	29.0	65.8	17.6	64.3	68.3	64.0	65.6	28.8	66.5	25.8	59.5	49.8	54.3
[90]	67.1	50.5	62.2	42.1	38.9	57.8	47.7	53.7	32.1	52.3	17.5	53.1	56.0	53.2	57.7	29.4	62.2	24.0	46.2	32.8	46.8
[69]	38.4	38.9	8.6	9.3	23.0	37.1	33.6	18.4	11.5	16.0	5.1	12.2	29.0	21.3	46.9	7.2	15.8	5.6	14.4	21.4	20.7

Table 2.5: **SBD val evaluation:** Semantic contours results: Average Precision (AP) per class and mean AP (mAP) is reported for all methods.

Technique	BG	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Mean
COB-dil	**93.5**	**90.3**	**39.7**	**83.2**	**66.2**	**68.9**	**92.6**	**84.6**	**89.2**	**36.9**	**84.7**	**53.1**	**82.9**	**87.0**	**83.1**	**86.3**	**54.7**	**84.8**	**45.7**	**84.6**	**68.9**	**74.3**
DilatedConv [227]	92.8	87.1	39.2	79.6	65.9	66.3	90.0	82.5	85.3	36.2	81.7	51.7	78.1	83.8	80.2	83.4	50.5	82.6	43.1	83.8	65.3	71.9
COB-PSP	**95.4**	**90.9**	**44.8**	90.2	**76.1**	**84.1**	96.1	**92.1**	**95.3**	45.6	**95.4**	**59.9**	**92.0**	**93.2**	90.8	**90.1**	**68.0**	**93.4**	**50.2**	**93.3**	79.8	**81.7**
PSPNet [233]	95.3	90.7	44.4	90.2	74.8	83.4	**96.3**	92.0	95.0	**46.4**	94.6	59.1	91.9	92.5	**91.0**	89.9	66.0	91.6	50.2	93.0	**80.0**	81.3

Table 2.6: **PASCAL VOC Segmentation val evaluation:** Effect of COB on Semantic Segmentation. Per-class IoU and mean IoU are reported.

Kokkinos [97] approaches the task with fully-convolutional networks trained end to end, although the results do not reach the current state of the art.

We also follow the most common approach of mixing the two tasks. We couple the COB boundaries with Semantic Segmentation results by dilated convolutions [227]. Specifically, we mask the boundaries with Semantic Segmentation results, with a tolerance of 0.02 of the image diagonal.

Technique	Plane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	MBike	Person	Plant	Sheep	Sofa	Train	TV	Mean
COB	69.5	76.8	**69.7**	53.3	**44.6**	**80.5**	**81.3**	**83.1**	**45.3**	**74.2**	**69.4**	**80.1**	**84.2**	**76.7**	**72.8**	**35.9**	**67.1**	**68.4**	**75.1**	65.4	**68.7**
SeSe	**76.0**	76.8	65.3	**54.6**	38.0	76.5	78.2	81.6	40.1	74.1	66.5	78.9	81.8	74.5	66.2	32.9	65.6	67.7	73.4	**66.8**	66.8

Table 2.7: **VOC 2007 test evaluation:** Object Detection performance (mAP) of Fast-RCNN [61], using object proposals from [209] (original) or COB.

We report results on the SBD [69] database, for semantic boundary detection, by using the standard benchmark. Tables 2.4 and 2.5 compare the results among various methods, in both metrics used in the benchmark (mean maximal F-measure and Average Precision) for all classes. The combination of COB with [227], denoted with COB-dil, achieves state-of-the-art results in both metrics. For fair comparison, we also include the results obtained by evaluating the semantic segmentation results obtained by [227] directly as contours. We show that COB fairly improves the result.

Having explored the performance of COB combined with the Dilated Convolution network on Semantic Boundaries, it is interesting to investigate the dual task: the effects of COB in semantic segmentation. We treat the COB UCMs as superpixels, by applying a low value threshold (0.1) to the hierarchy, which results in high recall. We then snap the semantic segmentation results to the superpixels by majority voting of the regions, i.e superpixels that overlap more than 50% with the semantic class, are assigned the corresponding label. Table 2.6 reports the effects of such snapping on Semantic Segmentation, on the validation split of PASCAL VOC Segmentation dataset. In addition to the Dilated network, we also explored the most recent PSPNet [233] as the base semantic segmenter. Results improve consistently almost for all the classes in both cases, indicating that COB superpixels are further refining the semantic segmentation results on boundary locations. We observe a more moderate improvement in the PSPNet results, mainly because of the reduced false detections. We have excluded all images of VOC Segmentation val set for training the COB model.

In Figure 2.16 we present some qualitative results. Snapping to COB superpixels improves mainly on boundary locations, as well as on noisy semantic segmentation detections in places where COB superpixels are not present.

2.6.3 *COB Object Proposals for Object Detection*

Object Proposals have been extensively used to facilitate object detection [62, 61, 178]. Most common pipelines use object proposals in the form of a bounding box to regress a class score and a refined prediction of the bounding box locations. Even though our approach provides segmented object proposals from a hierarchy of regions, it is possible

Figure 2.16: **Qualitative results for Semantic Segmentation:** Row 1: original images, Row 2: Dilated Convolution Network, Row 3: Dilated Network with COB superpixels.

to study their effect on common object detection pipelines by simply extracting the bounding box around them.

We evaluate the bounding box proposals generated by COB by feeding them into the Fast-RCNN [61] pipeline for Object Detection. The original approach uses the VGG network [199] together with the box proposals generated by the Selective Search [209] algorithm to predict class probability and refine the localization for each of them. The final detection performance is evaluated by performing non-maximum suppression on the detections.

Experiments are performed on the VOC'07 detection database. The database consists of 5011 training, and we report the performance on its 4952 testing images. In our experiments, we change the box proposals of Selective Search, to the ones generated by COB. We keep all the hyper-parameters of the original approach unchanged, both at training and test times. Table 2.7 quantitatively evaluates the effects of COB proposals in performance. We observe improvements in object detection performance (mean Average Precision - mAP), which further proves

the high quality of the proposals generated by COB. We would like to emphasize that the latest developments on Object Detection use joint training of bounding box proposals and object class scores [178, 124, 176, 45], which together with training on external data achieves much higher results. Instead, we focus on proving the high quality of COB proposals compared to other object proposal techniques.

2.7 CONCLUSIONS

In this chapter, we have developed an approach to detect contours at multiple scales, together with their orientations, in a single forward pass of a convolutional neural network. We provide a fast framework for generating region hierarchies by efficiently combining multiscale oriented contour detections, thanks to a new sparse boundary representation. We shift from the BSDS to PASCAL to unwind all the potential of data-hungry methods such as CNNs and by observing that BSDS is close to saturation.

Our technique achieves state-of-the-art performance by a significant margin for contour detection, the estimation of their orientation, and generic (RGB and RGB-D) image segmentation. We show that our architecture is modular by using two different CNN base architectures, which suggests that it will be able to transfer further improvements in CNN base architectures to perceptual grouping. We also show that our method does not require globalization, which was a speed bottleneck in previous approaches. The generalization of COBwas further demonstrated when applied to high-level vision tasks (object proposals, object detection, and semantic contours and segmentation) in combination with recent pipelines, where the results are improved in all cases.

3

DEEP EXTREME CUT: FROM EXTREME POINTS TO OBJECT SEGMENTATION

This chapter explores the use of extreme points in an object (left-most, right-most, top, bottom pixels) as input to obtain precise object segmentation for images and videos. We do so by adding an extra channel to the image in the input of a convolutional neural network (CNN), which contains a Gaussian centered in each of the extreme points. The CNN learns to transform this information into a segmentation of an object that matches those extreme points.

We demonstrate the usefulness of this approach for guided segmentation (grabcut-style), interactive segmentation, video object segmentation, and dense segmentation annotation. We show that we obtain the most precise results to date, also with less user input, in an extensive and varied selection of benchmarks and datasets.

3.1 INTRODUCTION

Deep learning techniques have revolutionized the field of computer vision since their explosive appearance in the ImageNet competition [188], where the task is to classify images into predefined categories, that is, algorithms produce one label for each input image. Image and video segmentation, on the other hand, generate dense predictions where each pixel receives a (potentially different) output classification. Deep learning algorithms, especially Convolutional Neural Networks (CNNs), were adapted to this scenario by removing the final fully connected layers to produce dense predictions, among other modifications.

Supervised techniques, those that train from manually-annotated results, are currently the best performing in many public benchmarks and challenges [233, 33, 73]. In the case of image and video segmentation, the supervision is in the form of dense annotations, *i.e*each pixel has to be annotated in an expensive and cumbersome process. Weakly-supervised techniques, which train from incomplete but easier-to-obtain annotations, are still significantly behind the state of the art. Semi-automatic techniques, which need a human in the loop to pro-

Figure 3.1: **Example results of DEXTR:** The user provides the extreme clicks for an object, and the CNN produces the segmented masks.

duce results, are another way of circumventing the expensive training annotations but need interaction at test time, which usually comes in the form of a bounding box [44, 89] or scribbles [119] around the object of interest. How to incorporate this information at test time without introducing unacceptable lag, is also a challenge.

This paper tackles all these scenarios in a unified way and shows state-of-the-art results in all of them in a variety of benchmarks and setups. We present Deep Extreme Cut (DEXTR), that obtains an object segmentation from its four extreme points [156]: the left-most, right-most, top, and bottom pixels. Figure 3.1 shows an example result of our technique along with the input points provided.

In the context of semi-automatic object segmentation, we show that information from extreme clicking results in more accurate segmentations than the ones obtained from bounding-boxes (PASCAL, COCO, Grabcut) in a Grabcut-like formulation. DEXTR outperforms other methods using extreme points or object proposals (PASCAL), and provides a better input to video object segmentation (DAVIS 2016, DAVIS 2017). DEXTR can also incorporate more points beyond the extreme ones, which further refines the quality (PASCAL).

DEXTR can also be used to obtain dense annotations to train supervised techniques. We show that we obtain very accurate annotations with respect to the ground truth, but more importantly, that algorithms trained on the annotations obtained by our algorithm perform as good as when trained from the ground-truth ones. If we add the cost to obtain such annotations into the equation, then training using DEXTR is significantly more efficient than training from the ground truth for a given target quality.

We perform an extensive and comprehensive set of experiments on COCO, PASCAL, Grabcut, DAVIS 2016, and DAVIS 2017, to demonstrate the effectiveness of our approach.

3.2 RELATED WORK

Weakly Supervised Signals for Segmentation: Numerous alternatives to expensive pixel-level segmentation have been proposed and used in the literature. Image-level labels [159], noisy web labels [4, 87] and scribble-level labels [119] are some of the supervisory signal that have been used to guide segmentation methods. Closer to our approach, [15] employs point-level supervision in the form of a single click to train a CNN for semantic segmentation and [157] uses central points of an imaginary bounding box to weakly supervise object detection. Also related to our approach, [44, 89] train semantic segmentation methods from box supervision. Recently, Papadopoulos et al. proposed a novel method for annotating objects by extreme clicks [156]. They show that extreme clicks provide additional information to a bounding box, which they use to enhance GrabCut-like object segmentation from bounding boxes. Different than these approaches, we use extreme clicking as a form of guidance for deep architectures, and show how this additional information can be used to further boost accuracy of segmentation networks, and help various applications.

Instance Segmentation: Several works have tackled the task of grouping pixels by object instances. Popular grouping methods provide instance segmentation in the form of automatically segmented object proposals [77, 166]. Other variants provide instance-level segmentation from a weak guiding signal in the form of a bounding box [187]. Accuracy for both groups of methods has increased by recent approaches that employ deep architectures trained on large datasets with strong supervisory signals, to learn how to produce class-agnostic masks from patches [164, 165], or from bounding boxes [223]. Our approach relates to the second group, since we utilize information from extreme clicks to group pixels of the same instance, with higher accuracy.

Interactive Segmentation from points: Interactive segmentation methods have been proposed in order to reduce annotation time. In this context, the user is asked to gradually refine a method by providing additional labels to the data. Grabcut [187] is one of the pioneering works for the task, segmenting from bounding boxes by gradually updating an appearance model. Our method relates with interactive segmentation using points as the supervisory signal. Click Carving [85] interactively updates the result of video object segmentation by user-defined clicks. Recent methods use these ideas in the pipeline of deep architectures. iFCN [224] guides a CNN from positive and negative points acquired from the ground-truth masks. RIS-Net [68] build on iFCN to improve the result by adding local context. Our method significantly improves the results by using just 4 class-agnostic points as the supervisory signal: the extreme points.

Current lines of work: After the published version of DEXTR, there have been numerous works in the related fields. Mahadevan et al. [132] iteratively train on the erroneous areas for interactive object segmentation. Benenson et al [16] investigate interactive methods for large-scale object segmentation. Among the methods that directly propose improvements on DEXTR, [213] iteratively refine the output of DEXTR by using ideas from level set evolution, trainable by back-propagation. Agustsson et al [3] build on top of DEXTR and propose a method for interactively annotate for panoptic segmentation, *i.e* simultaneously tackle semantic segmentation of background classes and semantic instance segmentation of foreground objects [92]. In a separate line of work, [29] and [2] predict the polygon vertices of an object given its bounding box. Oh et al. [155] use DEXTR to initialize their video object segmentation pipeline, and propose a method that is fast and very

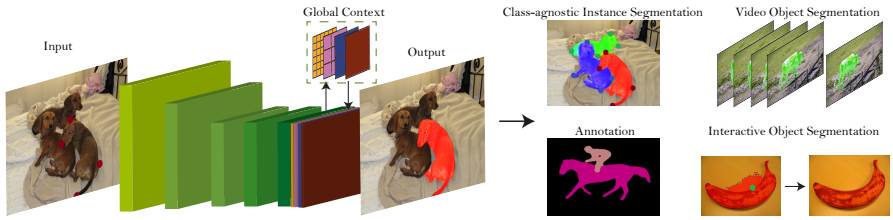


Figure 3.2: **Architecture of DEXTR:** Both the RGB image and the labeled extreme points are processed by the CNN to produce the segmented mask. The applicability of this method is illustrated for various tasks: Instance, Semantic, Video, and Interactive segmentation.

accurate towards practical video object segmentation for the first time. Zhou et al. [238] group extreme points for object detection, and proceed to semantic instance segmentation using DEXTR.

3.3 METHOD

3.3.1 *Extreme points*

One of the most common ways to perform weakly supervised segmentation is drawing a bounding box around the object of interest [22, 216, 187, 110]. However, in order to draw the corners of a bounding box, the user has to click points outside the object, drag the box diagonally, and adjust it several times to obtain a tight, accurate bounding box. This process is cognitively demanding, with increased error rates and labeling times [156].

Recently, Papadopoulos et al. [156] have shown a much more efficient way of obtaining a bounding box using extreme clicks, spending on average 7.2 seconds instead of 34.5 seconds required for drawing a bounding box around an object [201]. They show that extreme clicking leads to high quality bounding boxes that are on par with the ones obtained by traditional methods. These extreme points belong to the top, bottom, left-most and right-most parts of the object. Extreme-clicking annotations by definition provide more information than a bounding box; they contain four points that are on the boundary of the object, from which one can easily obtain the bounding-box. We

use extreme points for object segmentation leveraging their two main outcomes: the points and their inferred bounding box.

3.3.2 *Segmentation from Extreme Points*

The overview of our method is shown in Figure 3.2. The annotated extreme points are given as a guiding signal to the input of the network. To this end, we create a heatmap with activations in the regions of extreme points. We center a 2D Gaussian around each of the points, in order to create a single heatmap. The heatmap is concatenated with the RGB channels of the input image, to form a 4-channel input for the CNN. In order to focus on the object of interest, the input is cropped by the bounding box, formed from the extreme point annotations. To include context on the resulting crop, we relax the tight bounding box by several pixels. After the pre-processing step that comes exclusively from the extreme clicks, the input consists of an RGB crop including an object, plus its extreme points.

We choose *ResNet-101* [74] as the backbone of our architecture, as it has been proven successful in a variety of segmentation methods [31, 73]. We remove the fully connected layers as well as the max pooling layers in the last two stages to preserve acceptable output resolution for dense prediction, and we introduce atrous convolutions in the last two stages to maintain the same receptive field. After the last *ResNet-101* stage, we introduce a pyramid scene parsing module [233] to aggregate global context to the final feature map. Initializing the weights of the network from pre-training on ImageNet has been proven beneficial for various tasks [126, 220, 73]. For most experiments, we use the provided Deeplab-v2 model pre-trained on ImageNet, and fine-tuned on PASCAL for semantic segmentation.

The output of the CNN is a probability map representing whether a pixel belongs to the object that we want to segment or not. The CNN is trained to minimize the standard cross entropy loss, which takes into account that different classes occur with different frequency in a dataset:

$$\mathcal{L} = \sum_{j \in Y} w_{y_j} C\left(y_j, \hat{y}_j\right), \qquad j \in 1, ..., |Y| \tag{3.1}$$

where w_{y_j} depends on the label y_j of pixel j. In our case we define w_{y_j} with $y_j \in \{0, 1\}$ as the inverse normalized frequency of labels inside the minibatch. $C(.)$ indicates the standard cross-entropy loss between the

label and the prediction $\hat{y}_j$. The balanced loss has proven to perform very well in boundary detection [220, 139], where the majority of the samples belong to the background class. The same class-balancing was used in Eq. 2.1 of Chapter 2 for COB. We note that our method is trained from strong mask-level supervision, on publicly available datasets, using the extreme points as a guiding signal to the network.

In order to segment an object, our method uses a object-centered crop, therefore there is a much higher number of samples belonging to the foreground than to the background and the use of a balanced loss proves to be beneficial.

Alternatives for each of the components used in our final model have been studied in an ablation analysis, and a detailed comparison can be found in Section 3.4.2.

3.3.3 *Use cases for DEXTR*

Class-agnostic Instance Segmentation: One application of DEXTR is class-agnostic instance segmentation. In this task, we click on the extreme points of an object in an image, and we obtain a mask prediction for it. The selected object can be of any class, as our method is class agnostic.

In Section 3.4.3, we compare our method with the state of the art in two different datasets, PASCAL and Grabcut, where we improve current results. We also analyze the generalization of our method to other datasets and to unseen categories. We conclude positive results in both experiments: the performance drop for testing on a different dataset than the one used for training is very small and the result achieved is the same whether the class has been seen during training or not.

Annotation: The common annotation pipeline for segmentation can also be assisted by DEXTR. In this framework, instead of detailed polygon labels, the workload of the annotator is reduced to only providing the extreme points of an object, and DEXTR produces the desired segmentation. In this pipeline, the labeling cost is reduced by a factor of 10 (from 79 seconds needed for a mask [122], to 7.2 seconds needed for the extreme clicks [156]).

In Section 3.4.4, the quality of the produced masks are validated when used to train a semantic segmentation algorithm. We show that our method produces very accurate masks and the results trained on

them are on par with those trained on the ground-truth annotations in terms of quality, with much less annotation budget.

Video Object Segmentation: DEXTR can also improve the pipeline of video object segmentation. We focus on the semi-supervised setting where methods use one or more masks as inputs to produce the segmentation of the whole video. Our aim is to replace the costly per pixel annotation masks by the masks produced by our algorithm after the user has selected the extreme points of a certain object, and re-train strongly supervised state-of-the-art video segmentation architectures.

In Section 3.4.5, we provide results on two different dataset: DAVIS-2016 and DAVIS-2017. We conclude that state-of-the-art results can be achieved reducing the annotation time by a factor of 5. Moreover, for almost any specific annotation budget, better results can be obtained using a higher number of masks produced by our algorithm rather than expensive per-pixel annotated masks. Out efforts have been useful to the work of [155], where the authors initialize their video segmentation pipeline from predictions obtained by DEXTR.

Interactive Object Segmentation: The pipeline of DEXTR can also be used in the frame of interactive segmentation from points [224, 223]. We work on the case where the user labels the extreme points of an object, but is nevertheless not satisfied with the obtained results. The natural thing to do in such case is to annotate an extra point (not extreme) in the region that segmentation fails, and expect for a refined result. Given the nature of extreme points, we expect that the extra point also lies in the boundary of the object.

To simulate such behavior, we first train DEXTR on a first split of a training set of images, using the 4 extreme points as input. For the extra point, we infer on an image of the second split of the training set, and compute the accuracy of its segmentation. If the segmentation is accurate (eg. $IoU \geq 0.8$), the image is excluded from further processing. In the opposite case ($IoU < 0.8$), we select a fifth point in the erroneous area. To simulate human behavior, we perturbate its location and we train the network with 5 points as input. Results presented in Section 3.4.6 indicate that it is possible to recover performance on the difficult examples, by using such interactive user input.

Our method is extensively validated on five publicly available databases: PASCAL [54], COCO [122], DAVIS 2016 [162], DAVIS 2017 [168], and Grabcut [187], for various experimental setups that show its applicability and generalization capabilities. We use DEXTR trained on PASCAL (augmented by the labels of SBD [69] following the common practice - 10582 images), unless indicated differently. Some implementation details are given in Section 3.4.1. We then perform an ablation study to separately validate all components of our method in Section 3.4.2. Class-agnostic instance segmentation experiments from extreme points are presented in Section 3.4.3, whereas Sections 3.4.4 and 3.4.5 are dedicated to how DEXTR contributes to segmentation annotation and video object segmentation pipelines, respectively. Section 3.4.6 presents our method as an interactive segmenter from points.

3.4.1 *Implementation Details*

Simulated Extreme Points: In [156], extreme points in PASCAL were obtained by crowd-sourcing. We used their collected extreme points when experimenting on the same dataset, and collected new extreme points by humans in DAVIS 2016. To experiment on COCO, on which it was not feasible to collect extreme points by human annotators, we simulate them by taking the extreme points of the ground-truth masks jittered randomly by up to 10 pixels.

Training and testing details: DEXTR is trained on PASCAL 2012 Segmentation for 100 epochs or on COCO 2014 training set for 10 epochs. The learning rate is set to 10^{-8}, with momentum of 0.9 and weight decay of $5 * 10^{-4}$. A mini-batch of 5 objects is used for PASCAL, whereas for COCO, due to the large size of the database, we train on 4 GPUs with an effective batch size of 20. Training on PASCAL takes approximately 20 hours on a Nvidia Titan-X GPU, and 5 days on COCO. Testing the network is fast, requiring only 80 milliseconds. More details are provided in our open-source repository.

3.4.2 *Ablation Study*

The following sections show a number of ablation experiments in the context of class-agnostic instance segmentation to quantify the importance of each of the components of our algorithm and to justify various design choices. Table 3.2 summarizes these results. We use PASCAL VOC 2012 val set for the evaluation.

Architecture: We use *ResNet-101* as the backbone architecture, and compare two different alternatives. The first one is a straightforward fully convolutional architecture (Deeplab-v2 [31]) where the fully connected and the last two max pooling layers are removed, and the last two stages are substituted with dilated (or atrous) convolutions. This keeps the size of the prediction in reasonable limits ($8\times$ lower than the input). We also tested a region-based architecture, similar to Mask R-CNN [73], with a re-implementation of the ResNet-101-C_4 variant [73], which uses the fifth stage (C_5) for regressing a mask from the Region of Interest (RoI), together with the re-implementation of the RoI-Align layer. For more details please refer to [73]. In the first architecture, the input is a patch around the object of interest, whereas in the latter the input is the full image, and cropping is applied at the RoI-Align stage. Deeplab-v2 performs +3.9% better. We conclude that the output resolution of ResNet-101-C_4 (28×28) is inadequate for the level of detail that we target.

Bounding boxes vs. extreme points: We study the performance of Deeplab-v2 as a foreground-background classifier given a bounding box compared to the extreme points. In the first case, the input of the network is the cropped image around the bounding box plus a margin of 50 pixels to include some context. In the second case, the extreme points are fed together in a fourth channel of the input to guide the segmentation. Including extreme points to the input increases performance by +3.1%, which suggest that they are a source of very valuable information that the network uses additionally to guide its output.

Loss: For the task of class-agnostic instance segmentation, we compare two binary losses, i.e. the standard cross-entropy and a class-balanced version of it, where the loss for each class in the batch is weighted by its inverse frequency. Class-balancing the loss gives more importance to the less frequent classes, and has been successful in

various tasks [220, 139, 136]. DEXTR also performs better when the loss is balanced, leading to a performance boost of +3.3%.

Full image vs. crops: Having the extreme points annotated allows for focusing on specific regions in an image, cropped by the limits specified by them. In this experiment, we compare how beneficial it is to focus on the region of interest, rather than processing the entire image. To this end, we crop the region surrounded by the extreme points, relaxing it by 50 pixel for increased context and compare it against the full image case. We notice that cropping increases performance by +7.9%, and is especially beneficial for the small objects of the database. This could be explained by the fact that cropping eliminates the scale variation on the input. Similar findings have been reported for video object segmentation by [112].

Atrous spatial pyramid (ASPP) vs. pyramid scene parsing (PSP) module: Pyramid Scene Parsing Network [233] steps on the Deeplab-v2 [31] architecture to further improve results on semantic segmentation. Their main contribution was a global context module (PSP) that employs global features together with the local features for dense prediction. We compare the two network heads, the original ASPP [31], and the recent PSP module [233] for our task. The increased results of the PSP module (+2.3%) indicate that the PSP module builds a global context that is also useful in our case.

Manual vs. simulated extreme points: In this section we analyze the differences between the results obtained by DEXTR when we input either human-provided extreme points or our simulated ones, to check that the conclusions we draw from the simulations will still be valid in a realistic use case with human annotators. We do so in the two datasets where we have *real* extreme points from humans. The first one is a certain subset of PASCAL 2012 Segmentation and SBD (5623 objects) with extreme points from [156], which we refer to as PASCAL$_{EXT}$ and DAVIS 2016, for which we crowdsourced the extreme point annotations. The annotation time for the latter (average of all 1376 frames of the validation set) was 7.5 seconds per frame, in line with [156] (7.2 s. per image). Table 3.1 shows that the results are indeed comparable when using both type of inputs. The remainder of the chapter uses the simulated extreme points except when otherwise specified.

Distance-map vs. fixed points: Recent works [224, 223, 68] that focus on segmentation from (not-extreme) points use the distance transform of positive and negative annotations as an input to the network, in

Method	PASCAL$_{EXT}$	DAVIS 2016
Manual extreme points	80.1	80.9
Simulated extreme points	85.1	79.5

Table 3.1: **Manual vs. simulated extreme points:** Intersection over Union (IoU) of the DEXTR results when using manual or simulated extreme points as input.

order to guide the segmentation. We compare with their approach by substituting the fixed Gaussians to the distance transform of the extreme points. We notice a performance drop of -1.3%, suggesting that using fixed Gaussians centered on the points is a better representation when coupled with extreme points. In Section 3.4.3 we compare to such approaches, showing that extreme points provide a much richer guidance than arbitrary points on the foreground and the background of an object.

Summary: Table 3.2 summarizes the main ablated results that have been discussed above, analyzing all components.

Component #1	Component #2	Gain in IoU
Region-based	**Deeplab-v2**	+3.9%
Bounding Boxes	**Extreme Points**	+3.1%
Cross Entropy	**Balanced BCE**	+3.3%
Full Image	**Crop on Object**	+7.9%
ASPP	**PSP**	+2.3%
Fixed Points	Distance Map	−1.3%

Table 3.2: **Ablation study for DEXTR:** Comparative evaluation between different choices in various components of our system. Mean IoU over all objets in PASCAL VOC 2012 val set.

Table 3.3 illustrates the building blocks that lead to the best performing variant for our method. All in all, we start by a Deeplab-v2 base model working on bounding boxes. We add the PSP module (+2.3%), the extreme points in the input of the network (+3.1%), and more annotated data from SBD (+1%) to reach maximum accuracy. The improvement comes mostly because of the guidance from extreme points, which highlights their importance for the task.

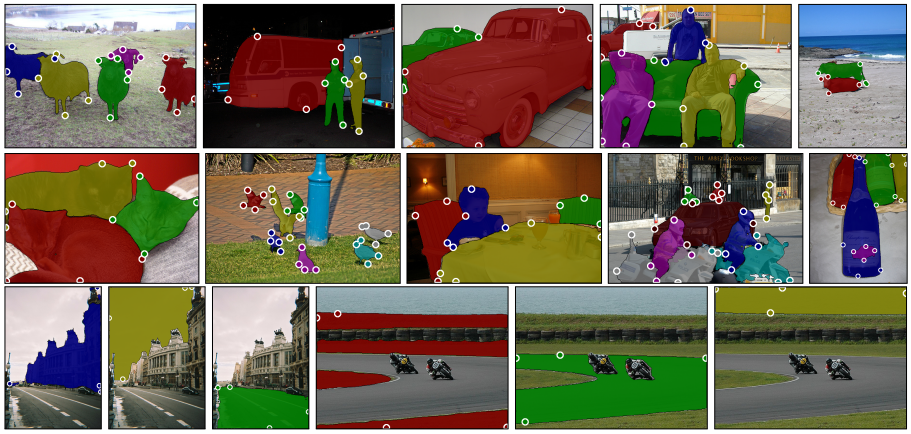


Figure 3.3: **Qualitative results by DEXTR in PASCAL:** Each instance with the simulated extreme points used as input and the resulting mask overlayed. The bottom row shows results on PASCAL Context stuff categories.

3.4.3 *Class-agnostic Instance Segmentation*

Comparison to the State of the Art in PASCAL: We compare our method against state-of-the-art class-agnostic instance segmentation methods in Table 3.4. DEXTR gets a boost of +6.5% with respect to using the *grabcut-based* method of [156] from extreme points.

We then compare to two other *baselines* using SharpMask [165], the state-of-the-art object proposal technique. In the first row, we evaluate the proposal (out of 1000) whose bounding box best overlaps with the ground-truth bounding box, mimicking a naive algorithm to segment boxes from proposals. The second row shows the upper bound of

Variant	IoU (%)	Gain
Full Image (Deeplab-v2 + PSP + Extreme Points)	82.6	
Crop on Object (Deeplab-v2)	85.1	+2.5%
+ PSP	87.4	+2.3%
+ Extreme Points	90.5	+3.1%
+ SBD data (Ours)	**91.5**	+1.0%

Table 3.3: **Best components for DEXTR:** Building performance in PASCAL VOC 2012 val set.

Method	IoU
Sharpmask [165] from bounding box	69.3%
Sharpmask [165] upper bound	78.0%
[156] from extreme points	73.6%
Ours from extreme points	**80.1%**

Table 3.4: **Comparison in PASCAL$_{EXT}$:** IoU of our results against class-agnostic instance segmentation methods, on the objects annotated by [156] to be able to compare to them.

SharpMask, that is, the best proposal against the ground truth, selected by an oracle. Both approaches are well below our result (-10.8% and -2.1%). Figure 3.3 illustrates some results obtained by our method on PASCAL.

Comparison to the State of the Art on the Grabcut dataset: We use our best PASCAL model and we test it in the Grabcut dataset [187]. This dataset contains 50 images, each with one annotated object from various categories, some of them not belonging to any of the PASCAL ones (banana, scissors, kangaroo, etc.). The evaluation metric is the error rate: the percentage of misclassified pixels within the bounding boxes provided by [110]. Table 3.5 shows the results, where DEXTR achieves 2.3% error rate, 1.1% below the runner up (or a 32% relative improvement).

Method	Error Rate (%)
GrabCut [187]	8.1
KernelCut [205]	7.1
OneCut [206]	6.7
[156] from extreme points	5.5
BoxPrior [110]	3.7
MILCut [216]	3.6
DeepGC [223]	3.4
Ours from extreme points	**2.3**

Table 3.5: **Comparison in the Grabcut dataset:** Error rates compared to the state-of-the-art techniques.

Generalization to unseen categories and across datasets: Table 3.6 shows our results when trained on a certain dataset (first column),

and tested in another one or certain categories (second column). In order to make a fair comparison, all the models are pre-trained only on ImageNet [188] for image labeling and trained on the specified dataset for category-agnostic instance segmentation. The first two rows show that our technique is indeed class agnostic, since the model trained on PASCAL achieves roughly the same performance in COCO mini-val (MVal) regardless of the categories tested. The remaining rows shows that DEXTR also generalizes very well across datasets, since differences are around only 2% of performance drop.

	Train	Test	IoU
Unseen categories	PASCAL	COCO MVal w/o PASCAL classes	80.3%
	PASCAL	COCO MVal only PASCAL classes	79.9%
Dataset generalization	PASCAL	COCO MVal	80.1%
	COCO	COCO MVal	82.1%
	COCO	PASCAL	87.8%
	PASCAL	PASCAL	89.8%

Table 3.6: **Generalization to unseen classes and across datasets:** Intersection over union results of training in one setup and testing on another one. MVal stands for mini-val.

Generalization to background (stuff) categories: In order to verify the performance of DEXTR in "background" classes, we trained a model using the background labels of PASCAL Context [147] (road, sky, sidewalk, building, wall, fence, grass, ground, water, floor, ceiling, mountain, and tree). Qualitative results (Figure 3.3 last row) suggest that our method generalizes to background classes as well. Quantitatively, we achieve a mIoU of 81.75% in PASCAL-Context validation set, for the aforementioned classes.

3.4.4 *Annotation*

As seen in the previous section, DEXTR is able to generate high-quality class-agnostic masks given only extreme points as input. The resulting masks can in turn be used to train other deep architectures for other tasks or datasets, that is, we use extreme points as a way to annotate a new dataset with object segmentations. In this experiment we compare

the results of a semantic segmentation algorithm trained on either the ground-truth masks or those generated by DEXTR (we combine all per-instance segmentations into a per-pixel semantic classification result).

Specifically, we train DEXTR on COCO and use it to generate the object masks of PASCAL train set, on which we train Deeplab-v2 [31], and the PSP [233] head as the semantic segmentation network. To keep training time manageable, we do not use multi-scale training/testing. We evaluate the results on the PASCAL 2012 Segmentation val set, and measure performance by the standard mIoU measure (IoU per-category and averaged over categories).

Figure 3.4 shows the results with respect to the annotation budget (left) and the number of images (right). For completeness, we also report the results of PSPNet [233] (•) by evaluating the model provided by the authors (pre-trained on COCO, with multi-scale training and testing). The results trained on DEXTR's masks are significantly better than those trained from the ground truth on the same budget (e.g. 70% IoU at 7-minute annotation time vs. 46% with the same budget, or 1h10 instead of 7 minutes to reach the same 70% accuracy). DEXTR's annotations reach practically the same performance than ground truth when given the same number of annotated images.

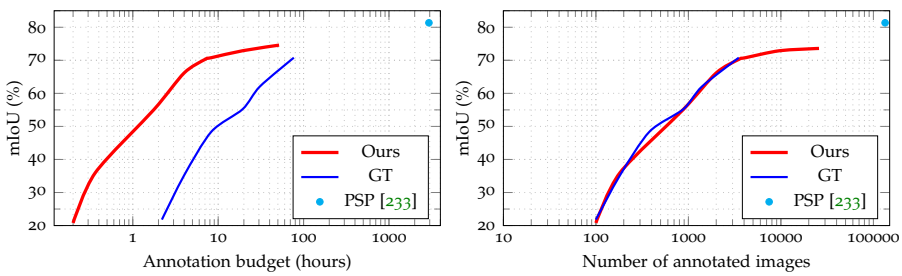


Figure 3.4: **Quality vs. annotation budget:** mIoU for semantic segmentation on PASCAL val set trained on our masks or the input, as a function of annotation budget (left) and the number of annotated images (right).

3.4.5 *Video Object Segmentation*

We test DEXTR also for Video Object Segmentation on the DAVIS datasets [162, 168]. We focus on the semi-supervised setting *i.e*the mask in one or more frames of the object that we want to segment is given as input to the algorithm, and as before we will compare the results obtained from the masks obtained by DEXTR or the ground truth having a certain annotation budget. We assume that the annotation time of the DAVIS masks is the same than that of COCO [122] (79 seconds per instance), despite the former are significantly more accurate.

We use OSVOS [25], as a state-of-the-art semi-supervised video object segmentation technique, which heavily relies on the appearance of the annotated frame, and their code is publicly available. Figure 3.5 (left) shows the performance of OSVOS in DAVIS 2016 [162] trained on the ground truth mask (—) or the masks generated by DEXTR from extreme points (—). We reach the same performance as using one ground-truth annotated mask with an annotation budget 5 times smaller. Once we train with more than one ground-truth annotated mask, however, even though we can generate roughly ten times more masks, we cannot achieve the same accuracy. We believe this is so because DAVIS 2016 sequences have more than one semantic instance per mask while we only annotate a global set of extreme points, which confuses DEXTR.

To corroborate this intuition, we perform the same experiment in DAVIS 2017 [168], where almost every mask contains only one instance. Figure 3.5 (right) shows that the performance gap with respect to using the full ground-truth mask is much smaller than in DAVIS 2016. Overall, we conclude that DEXTR is also very efficient to reduce annotation time in video object segmentation.

3.4.6 *Interactive Object Segmentation*

DEXTR for Interactive Segmentation: We experiment on PASCAL VOC 2012 segmentation for interactive object segmentation. We split the training dataset into two equal splits. Initially, we train DEXTR on the first split and test on the second. We then focus on the objects with inaccurate segmentations, i.e. IoU$<$0.8, to simulate the ones on which a human - unsatisfied with the result - would mark a fifth point. The extra point would lie on the boundary of the erroneous area (false positive

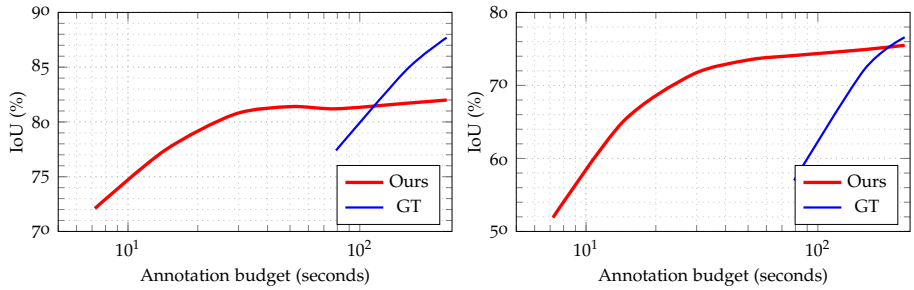


Figure 3.5: **Quality vs. annotation budget in video object segmentation:** OSVOS' performance when trained from the masks of DEXTR or the ground truth, on DAVIS 2016 (left) and on DAVIS 2017 (right).

or false negative), which we simulate as the boundary point closest to the highest error. From the perspective of network training, this can be interpreted as Online Hard Example Mining (OHEM) [197], where one only needs to back-propagate gradients for the training examples that lead to the highest losses. Results are presented in Table 3.7.

Trained on	4 points	4 points-all	5 points	5 points + OHEM
IoU	59.6%	69.0%	69.2%	**73.2%**

Table 3.7: **Interactive Object Segmentation Evaluation:** Average IoU on difficult cases of PASCAL VOC 2012 validation dataset.

We first select the objects that lead to poor performance (IoU$<$0.8) when applying the network trained on the first split. We report the average IoU on them (338 objects - 59.6%). Using the network trained further on the hard examples, with a fifth boundary point, performance increases to 73.2% ("5 points + OHEM").

Since the increased performance is partially due to the increased amount of training data (first split + hard examples of the second split), we need to disentangle the two sources of performance gain. To this end, we train DEXTR on 4 points, by appending the hard examples of the second split to the first split of our training set ("4 points-all").

Results suggest that DEXTR learns to handle more input information given interactively in the form of boundary clicks, to improve results of poorly segmented difficult examples (+4.2%). Interestingly, OHEM is a

crucial component for improving performance: without it the network does not focus on the difficult examples (only 11% of objects of the second training split are hard examples), and fails to improve on the erroneous region indicated by the fifth boundary point ("5 points").

Comparison to the State of the Art: We compare against the state-of-the-art in interactive segmentation by considering extreme points as 4 clicks. Table 3.8 shows the number of clicks that each method needs to reach a certain performance, as well as their performance when the input is 4 clicks, in PASCAL and the Grabcut dataset. DEXTR reaches about 10% higher performance at 4 clicks than the best competing method, and reaches 85% or 90% quality with fewer clicks. This further demonstrates the enhanced performance of the CNN, when guided by extreme points.

	Number of Clicks		IoU (%) @ 4 clicks	
Method	PASCAL@85%	Grabcut@90%	PASCAL	Grabcut
GraphCut [22]	> 20	> 20	41.1	59.3
Geodesic matting [12]	> 20	> 20	45.9	55.6
Random walker [63]	16.1	15	55.1	56 .9
iFCN [224]	8.7	7.5	75.2	84.0
RIS-Net [68]	5.7	6.0	80.7	85.0
Ours	**4.0**	**4.0**	**91.5**	**94.4**

Table 3.8: **PASCAL and Grabcut Dataset evaluation:** Comparison to interactive segmentation methods in terms of number of clicks to reach a certain quality and in terms of quality at 4 clicks.

To the meticulous reader, please note that the difference in performance of DEXTR between Table 3.8 (91.5%) and Table 3.1 (85.1%) comes from the fact that the former is on PASCAL VOC 2012 segmentation validation, so DEXTR is trained on SBD + PASCAL train, whereas the latter is on a subset of PASCAL that overlaps with train (PASCAL_{EXT}), so DEXTR is only trained on COCO.

3.5 CONCLUSIONS

We have presented DEXTR, a CNN architecture for semi-automatic segmentation that turns extreme clicking annotations into accurate object masks; by having the four extreme locations represented as a

heatmap extra input channel to the network. The applicability of our method is illustrated in a series of experiments regarding semantic, instance, video, and interactive segmentation in five different datasets; obtaining state-of-the-art results in all scenarios. DEXTR can also be used as an accurate and efficient mask annotation tool, reducing labeling costs by a factor of 10.

4

AUTOMATIC TOOL LANDMARK DETECTION FOR STEREO VISION IN ROBOT-ASSISTED RETINAL SURGERY

Computer vision and robotics are being increasingly applied in medical interventions. Especially in interventions where extreme precision is required they could make a difference. One such application is robot-assisted retinal microsurgery. In recent works, such interventions are conducted under a stereo-microscope, and with a robot-controlled surgical tool. The complementarity of computer vision and robotics has however not yet been fully exploited. In order to improve the robot control we are interested in 3D reconstruction of the anatomy and in automatic tool localization using a stereo microscope. In this chapter, we solve this problem for the first time using a single pipeline, starting from uncalibrated cameras to reach metric 3D reconstruction and registration, in retinal microsurgery. The key ingredients of our method are: (a) surgical tool landmark detection and (b) 3D reconstruction with the stereo microscope, using the detected landmarks. To address the former, we propose a novel deep learning method that detects and recognizes keypoints in high definition images at higher than real-time speed. We use the detected 2D keypoints along with their corresponding 3D coordinates obtained from the robot sensors to calibrate the stereo microscope using an affine projection model. We design an online 3D reconstruction pipeline that makes use of smoothness constraints and performs robot-to-camera registration. The entire pipeline is extensively validated on open-sky porcine eye sequences. Quantitative and qualitative results are presented for all steps.

4.1 INTRODUCTION

Robot and computer vision-assisted surgical procedures are becoming more and more popular due to their ability to attain high precision. One such procedure in ophthalmology involves the peeling of a retinal membrane to improve human vision. In this setup, the surgeon observes the retina and the tool under a stereo microscope while using a robotic

arm to control the surgical tool with high precision. This work also builds on such setup, consisting of a surgical tool which is positioned by a robot, and a stereo camera pair that is directly mounted on the surgical microscope. Generally, in such a setup the position of the surgical tool is known with respect to the robot's reference frame, but its position relative to the retinal surface and the cameras is unknown. As a result, for the robot to safely operate in an allowed region inside the eye, additional distance sensors are used to measure and maintain a safe distance to the retina. Visual guidance, however, still remains infeasible due to the different camera and robot coordinate systems. This means that information that comes from the processed images, e.g., the outcome of a retinal segmentation algorithm [140], cannot be effectively used. Due to limitations of the microscope acquisition it is further difficult to recover the actual 3D retinal surface. Therefore, accurate localization of the tool with respect to the retinal surface at every instant during surgery remains a very challenging problem.

In this chapter we tackle the problem of stereo microscope calibration, 3D reconstruction of the retina, and the registration of the landmark points on the tool with respect to the retinal surface. This is the first time all these problems are tackled together. In order to localize both the tool and the retinal surface in 3D, we exploit the robot kinematics which can be measured very accurately with current robotic systems. In this context we solve two important vision problems online: detecting the tool points accurately in the images and reconstructing the retina and the tool points in scale using the stereo microscope camera. Both are challenging problems on their own [203, 183, 30]. Detecting tool landmark points requires to take into account changes in viewpoint, de-focused images, specularities and fast movements of the tool. In addition, the surface reconstruction problem is hindered by the difficulty of calibrating the microscope cameras and the specularities in the images. Unlike consumer cameras, microscope cameras used in retinal microsurgery pose additional challenges: *a)* the narrow field of view and very long effective focal length, *b)* a small depth of field, *c)* rolling shutter and *d)* varying rotation and unknown baseline. These challenges make calibration very difficult in practice. Additionally, it is not obvious which camera model and reconstruction strategy best fits the problem of stereo reconstruction from microscopes used in retinal microsurgery. We show that automatically detecting tool landmarks in images, together with their respective 3D positions as they are di-

rectly obtained from the robot kinematics provides a reliable solution for microscope calibration and for the retinal reconstruction and tool registration. Figure 4.1 gives an overview of our method.

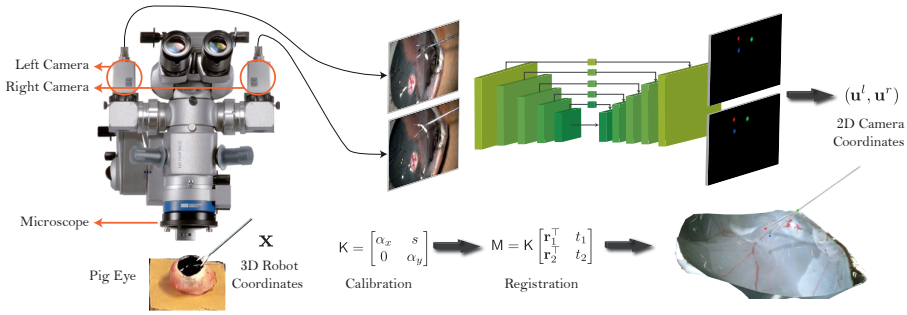


Figure 4.1: **Overview of our method:** Proposed pipeline for stereo calibration, retina reconstruction and tool registration. The microscope and the object of interest are not shown at the correct, relative scale. In the actual setup, the object is orders of times smaller and farther from the microscope.

The first task we tackle is markerless surgical tool keypoint detection in images. Several methods [20] have been developed to detect tools in images for various types of surgery. Most are based on hand-crafted features such as those obtained by color image transforms, image gradients and/or RGB pixel intensities [107, 7, 38]. Some work looked into tool detection in the specific case of retinal microsurgery [203, 183]. Such previous contributions were either restricted to generation of rather inaccurate bounding boxes, or their computational cost precluded real-time applications. We cast surgical tool detection as a landmark localization problem. We draw inspiration from deep learning algorithms initially used for human pose estimation, in order to detect the tool landmark points in images. We obtain automatically the 3D to 2D correspondences of these landmark keypoints from the robot kinematics. Our detection method runs on full-HD resolution (1920×1080 pixels) without the use of markers, at a frame rate of 35 frames per second with a GPU, and requires very few examples of annotated images for training. Experiments show the effect of image resolution on the detection of the tool landmarks and how detection noise affects the camera calibration.

We tackle the problem of stereo microscope calibration using the detected tool landmark points, by assuming a full affine camera model [71]

for each microscope camera. Previous methods [30, 75] propose calibration of the affine camera by first reconstructing the object with affine Structure from Motion (SfM) [207] and then computing the suitable upgrade for calibration. This may not be reliable due to the inherent problems of affine factorization-based SfM with respect to noise, the occurrence of missing data, and reconstruction ambiguities. In contrast, we formulate the calibration independent of the reconstruction, and base it solely on the measured robot motion. This frees the calibration from potential errors in factorization based reconstruction. In our pipeline, the tool is first moved around under the fixed cameras such that a few depths are covered, while the 3D positions from the robot encoder and the observed landmark 2D detections are recorded. We then use the full affine camera model to calibrate the intrinsics as well as the extrinsics using a Gaussian noise prior on the measurements and affine bundle adjustment. In order to initialize the bundle adjustment we use the Direct Linear Transform (DLT) [71]. The projection matrices obtained from the DLT calibration can be directly used to triangulate any stereo correspondence to a 3D point in the robot reference coordinates at the correct scale. We reconstruct the retinal surface by fitting a single smooth surface to the triangulated points. We use Bicubic B-Splines (BBS) to estimate the surface, using the point cloud while catering for its outliers and noise. To the best of our knowledge, this is the first work to employ the calibrated affine camera model for triangulating stereo pair image correspondences with tens of μm accuracy. This is an important result as calibration based on checkerboard patterns [232] and DLT with the perspective camera model fails. In summary, we present a method to obtain accurate camera as well as hand-eye calibration of the robot-camera system, localization of the tool, and reconstruction of the retina, all within the same pipeline. We use ex-vivo pig-eyes to validate our method. We provide detailed evaluation for each part, separately and in combination, showing several quantitative and qualitative results.

4.2 RELATED WORK

CNNs for Landmark Localization: Convolutional Neural Networks (CNNs) have recently revolutionized many computer vision tasks. Image recognition on very large datasets such as ImageNet [103, 199, 74] is one of the most representative examples. Models initially trained on ImageNet can often be fine-tuned for a variety of tasks, thus pro-

ducing state-of-the-art results, such as for object detection [178, 120] and segmentation [233, 31]. Related to this chapter are the CNN-based keypoint prediction methods, applied for Human Pose Estimation [154, 161, 208]. Drawing inspiration from such methods, we use a CNN to directly regress the keypoints, and thus the 2D pose of the surgical tool. Pavlakos et al. [160] use semantic keypoints to obtain the 6 degrees of freedom (DoF) pose of objects. Their pipeline is limited by the GPU memory, which enforces the authors to downsample the input images. In contrast, our method uses full-HD stereo images (1080×1920), and we argue that keeping the input resolution is crucial for achieving accurate localization. Concurrent work [104] uses tool landmark detection for assisting segmentation. Different from that approach, we focus on instrument landmark detection to assist in 3D vision tasks, such as microscopic camera calibration, and robot-to-camera registration. Our proposed method is also trained from scratch, meaning that we do not rely on pre-trained ImageNet weights that are difficult to acquire, and thus we are flexible in the network design. Our aim is to achieve real-time performance, which is usually not possible using very deep architectures [104].

Stereo Calibration and Reconstruction: There is an extensive literature on camera calibration for both stereo and monocular cameras [71, 232]. Yet, the problem is different for microscope cameras. For the task of modeling the projection geometry, it is not clear which camera models and calibration methods provide the best results. For example, [9] considers a perspective camera model to calibrate a standard microscope while [30] considers an affine camera model for a fundus camera. Due to the special optical arrangement of the camera, the small size of the viewed object and its relatively large distance to the camera, we use the affine camera model. In [30], the authors propose to reconstruct the retinal surface using classical affine Structure from Motion (SfM) [207], with a fundus camera. Such reconstruction is known only up to an unknown affine transform however, and the authors propose an upgrade to metric reconstruction by solving a highly non-linear cost function that requires a suitable initialization. The final retinal reconstruction is obtained only after fitting a spherical surface to the reconstructed points. In [75], the non-linear cost is avoided by using controlled robot motions so that the affine shape from factorization [207] can be used to formulate Linear Matrix Inequalities (LMI) for full affine calibration. Both works [30, 75] rely on having an accurate affine reconstruction

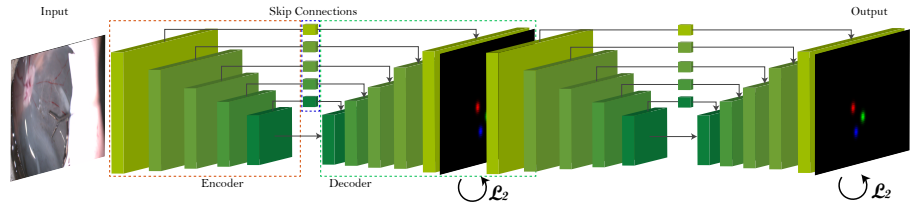


Figure 4.2: **Stacked Hourglass Network (SHN) architecture overview:** SHN is a fully convolutional architecture, which consists of multiple hourglass modules (two in the figure). Each hourglass is built by an encoder-decoder with skip connections. Each box represents a residual module [74]. Each hourglass is supervised by the l_2 loss.

for calibration. In practice, affine factorization is challenging due to outliers, missing data bound to be present in long view sequences as required here, and hence may not give an accurate result. We therefore propose to calibrate the affine cameras independently, using bundle adjustment accurately initialized by DLT before moving on to the reconstruction. This allows us to reconstruct surfaces online. We also do not require an *a priori* geometric model of the surface.

Current lines of work: Despite the high complexity of the presented setup, our work has inspired further research for Calibration, 3D reconstruction and registration for robot-assisted surgery. Specifically, in [236] the authors propose a hand-eye calibration technique for the Optical Coherence Tomography (OCT) device and the robot, by detecting the tip of the operating robot tool. In a follow-up work [237] they propose to estimate 6-DoF pose of the tool from the point cloud generated by the OCT. We note that information from stereo cameras and OCT are complementary, they can both be used in parallel during surgery, and as shown in these works, 3D processing can be tackled by the pipeline suggested in this chapter in both modalities.

4.3 AUTOMATIC SURGICAL INSTRUMENT LANDMARK LOCALIZATION

Number of keypoints: Before designing the localization algorithm, a careful consideration about the number of the landmark points required by subsequent parts of the pipeline is needed. In general, we require at least 3 non-collinear point correspondences in order to register two

coordinate frames (from a single image). We therefore design our CNN architecture to detect 3 keypoints in separate output channels. Note that the number of keypoints has a diminishing impact on the computational cost. In case of the surgical tool used for the retinal membrane peeling, we select the base of the tool, as well as the two tool tips as the landmark points of interest.

CNN architecture: For keypoint localization, we re-implemented the Stacked Hourglass Network architecture (SHN) [154], which has been proven very effective for Human Pose Estimation. Human pose estimation is dominated by keypoint localization approaches, focusing on various joints and landmarks of the human body (eg. head, right shoulder, etc.). Inspired by this approach, we substitute the body landmarks by the instrument landmarks, which makes SHN suitable for our purpose, although the original task is substantially different. SHN is a fully convolutional architecture, that consists of convolutional, ReLU, and pooling layers. Its core component is an encoder-decoder network enriched with skip connections. SHN is created by stacking together multiple such components, in a way such that the output of a previous component is the input to the next. Like this, coarse and fine features are gradually interchanged by pooling and upsampling operations on the feature maps, which builds a powerful representation for dense prediction tasks. SHNs also make extensive use of residual blocks [74] and intermediate supervision [108] which further enhance their performance. Figure 4.2 provides an overview of the SHN architecture. For a more detailed description, we refer the reader to the original paper [154]. We conducted a set of ablation experiments, with multiple architecture designs, where we concluded that the SHN architecture works best for the task of tool keypoint localization (Section 4.5.2).

We formulate tool keypoint localization as a heatmap regression problem. Specifically, for each keypoint, we regress a heatmap with its predicted location, as a separate channel of the CNN. We work with 3 keypoints and consequently 3 heatmaps. Our supervisory signal consists of the ground-truth locations, on top of which 2D Gaussians with standard deviation σ are centered. Centering Gaussians around the keypoints improves stability during training, since they ensure a softer loss over slight mis-localized detections. We train to minimize the l_2 loss. During inference, the peak activations in the final layer are

considered the locations of the keypoints. Specifically, we obtain the location of the kth keypoint as:

$$\hat{\mathbf{u}}_{det,k} = \frac{1}{|\Delta|} \sum_{\Delta} (\hat{\mathbf{u}}_{max,k} + \Delta)\, p_k\, (\hat{\mathbf{u}}_{max,k} + \Delta) \tag{4.1}$$

where $p_k(.)$ is the probabilistic activation of the k-th heatmap, $\hat{\mathbf{u}}_{max,k} =$ argmax p_k, and Δ is a small neighborhood. In our case, we define Δ as a circular neighborhood with radius 3σ. An example for the detected heatmaps is shown in Fig. 4.4. The detected 2D keypoints, together with the corresponding 3D locations acquired by the robot kinematics are fed to the next stages of the pipeline: camera calibration, registration, and 3D reconstruction.

4.4 AUTOMATIC CALIBRATION AND 3D RECONSTRUCTION

4.4.1 *Stereo Camera Calibration Using Robot Kinematics*

The problem of stereo camera calibration refers to that of obtaining the intrinsics and pose (extrinsics) of the cameras. The stereo camera used in retinal microsurgery, such as the one in Fig. 4.1, allow for a continuous adjustment of zoom and independent rotation of the cameras in a plane. Consequently, both extrinsics and intrinsics may change during the surgery. The standard way to calibrate a perspective camera is to use [232] on several images of a planar checkerboard pattern.

However, for the microscope cameras used in retinal microsurgery the projections are affine. This is because the distance from camera to object is orders of magnitudes larger than the object's size and the depth of field. Consequently, rays arrive almost parallel at the camera plane and perspective effects vanish. In such cases the equations of [232] are not well-conditioned and cannot be solved reliably. Furthermore, the use of a checkerboard is limited in practice, since in a realistic scenario a checkerboard can not be inserted into a human eye, which is the last thing that changes the optics for retinal surgery.

Automatic 3D-2D Correspondence Acquisition: In the case of robot-assisted surgery, we can exploit the fact that we are able to manipulate any surgical tool in 3D space while having instant position feedback computed using the robot kinematics. We therefore propose to tackle

the problem of affine stereo calibration by relying on automatic detection of distinct keypoints on the robot tool in the image. Having access to synchronized real-time kinematics, we can automatically accumulate any desired number of 3D-2D correspondences $\mathcal{C}$. Note that, given the robot-assisted surgical procedure, we obtain the correspondences for free once we have the 2D tool landmark predictions.

$$\mathcal{C} = \left\{ (\mathbf{x}, \mathbf{u}^l, \mathbf{u}^r) \in \mathbb{R}^3 \times \mathbb{R}^2 \times \mathbb{R}^2 \right\}_{t,k} \quad \begin{matrix} t \in [1, n_t] \\ k \in [1, n_k] \end{matrix} \tag{4.2}$$

While observing a sequence of n_t frames with a static camera pair, we detect n_k keypoints of the moving tool in each frame t, resulting in a set of $|\mathcal{C}| = n_k n_t$ correspondences. For each correspondence in $(\mathbf{x}, \mathbf{u}^l, \mathbf{u}^r)_i \in \mathcal{C}$, $\mathbf{x}$ is the tool 3D landmark expressed in the robot coordinate system while we refer to the corresponding 2D keypoints on the images as $\mathbf{u}^l$ and $\mathbf{u}^r$ for the left and the right camera. We use the subscript i as $\mathbf{x}_i$, $\mathbf{u}_i^l$ or $\mathbf{u}_i^r$ to denote the i-th 3D point in $\mathbf{C}$ and its projection on the left and right image, respectively.

Joint Affine Stereo Pair Calibration: We now formulate camera calibration as a problem of fitting an affine camera to model the image projections from given 3D points in the robot reference frame. Since we can control the robot, we make sure that a sufficient 3D volume is covered with point correspondences, to maximize the calibration accuracy. We are interested in an online stereo system that triangulates and reconstructs surfaces close to real-time from a pair of stereo images. Thus we deviate from the standard calibration methods based on affine reconstruction [30, 75] and triangulate Euclidean shapes directly using calibrated cameras. The affine camera projection is modeled by the projection matrix $\mathsf{M} \in \mathbb{P}_{\text{Affine}} \subset \mathbb{R}^{2\times 4}$ as $\mathbf{u} = \mathsf{M}\begin{bmatrix}\mathbf{x}^\top & 1\end{bmatrix}^\top$.

In order to jointly calibrate the stereo pair M^l, M^r, while accommodating for noise in the 2D detections and 3D measurements, we write the

following energy to robustly minimize reprojection errors in a bundle adjustment fashion:

$$\begin{aligned}
\min_{\mathsf{M}^l,\mathsf{M}^r,\tilde{\mathbf{x}},\tilde{\mathbf{u}}^l,\tilde{\mathbf{u}}^r} \quad & \mathcal{E}_\Pi(\mathsf{M}^c,\tilde{\mathbf{x}},\tilde{\mathbf{u}}^c)+\sigma_u^{-1}\mathcal{E}_\Theta(\tilde{\mathbf{u}}^c)+\sigma_x^{-1}\mathcal{E}_\Phi(\tilde{\mathbf{x}}) \\
\text{subject to,} \quad & \mathsf{M}^c \in \mathbb{P}_{\text{Affine}}, \quad c\in\{l,r\} \\
& \mathcal{E}_\Pi(\mathsf{M}^c,\tilde{\mathbf{x}},\tilde{\mathbf{u}}^c)=\frac{1}{2}\sum_c\sum_i(\tilde{\mathbf{u}}_i^c-\mathsf{M}^c\tilde{\mathbf{x}}_i)^2 \\
& \mathcal{E}_\Theta(\tilde{\mathbf{u}}^c)=\frac{1}{2}\sum_c\sum_i(\tilde{\mathbf{u}}_i^c-\mathbf{u}_i^c)^2 \\
& \mathcal{E}_\Phi(\tilde{\mathbf{x}})=\frac{1}{2}\sum_i(\tilde{\mathbf{x}}_i-\mathbf{x}_i)^2. \qquad (4.3)
\end{aligned}$$

The minimization problem in Eq. 4.3 is essentially the bundle adjustment for an affine camera. The first term describes the reprojection error. The last two terms model the uncertainty in the measurements as Gaussians with standard deviations σ_u, σ_x. We assume different intrinsics for each of the stereo camera pair and jointly optimize for the camera parameters $\mathsf{M}^{l,r}$, the 3D point positions $\tilde{\mathbf{x}}_i$, as well as for the 2D projections $\tilde{\mathbf{u}}_i^{l,r}$. Eq. 4.3 is optimized using a gradient-based interior-point technique.

Robust DLT for affine camera projection: In order to initialize the non-linear problem in Eq. 4.3 with a feasible configuration, we perform an affine Direct Linear Transform (DLT) on each camera separately. Writing down the affine projection for each point gives us the following system of equations for each camera:

$$\begin{bmatrix}\mathbf{u}_1 & \dots & \mathbf{u}_n\end{bmatrix} = \mathsf{M}\begin{bmatrix}\mathbf{x}_1 & \dots & \mathbf{x}_n \\ 1 & \dots & 1\end{bmatrix}. \qquad (4.4)$$

We solve for M in Eq. 4.4 by using the DLT algorithm [71] modified for affine projections. The problem becomes that of a linear least squares (LLS) that requires a minimum of $n = 4$ non-coplanar points. To tackle outliers and noise in 2D detections, we use Random Sample and Concensus (RANSAC) to estimate the projection matrix with Eq. 4.4. Although camera distortion could easily be included, this is not used for the sake of stability and better robustness against noise. Including distortion parameters also increases the number of minimum points needed by RANSAC, as well as the number of parameters in Eq. 4.3, outweighing the advantages of a more complex model.

Affine camera resection: For the perspective camera, resection refers to the decomposition of the projection matrix into the intrinsic calibration matrix and the 6 DoF pose of the camera. In the affine model, the problem is similar but only two rows of rotation and two translation components exist in the affine projection matrix. Consequently, the decomposition of the affine projection matrix is expressed as:

$$\mathsf{M} = \mathsf{K} \begin{bmatrix} \mathbf{r}_1^\top & t_1 \\ \mathbf{r}_2^\top & t_2 \end{bmatrix}, \quad \mathsf{K} = \begin{bmatrix} \alpha_x & s \\ 0 & \alpha_y \end{bmatrix} \tag{4.5}$$

where $\mathbf{r}_1^\top$ and $\mathbf{r}_2^\top$ are the first and second rows of a rotation matrix; t_1 and t_2 are the translation components, and K is the intrinsic affine calibration matrix. K is found by QR factorization of the projection matrix M. This gives the intrinsic calibration as well as the pose of each camera with respect to the robot reference frame except for the translation along the optical axis. The optimal camera parameters follow from Eq. 4.3 while enforcing $s = 0$ for stability. The intrinsics and pose parameters estimated from the bundle adjustment are used to recompute the camera projection matrices.

4.4.2 *Stereo Matching and Reconstruction*

The standard pipeline for stereo reconstruction with calibrated perspective cameras consists of dense disparity computation and depth map estimation by triangulation. The lighting used in retinal microsurgery often contaminates the images with specularities as well as other reflections, however. We therefore opt for a semi-dense matching method such as Deep Matching [214]. We filter out outliers based on the epipolar geometry derived from the affine fundamental matrix obtained during calibration. We then triangulate the matched points using the two affine projection matrices $\hat{\mathsf{M}}^l$ and $\hat{\mathsf{M}}^r$ for the stereo pair. Triangulation is possible because our estimated projection matrices are accurate. This directly gives us the 3D points of the observed surface in the Euclidean robot reference frame.

Robust surface estimation: The 3D points obtained from the triangulation contain outliers and noise due to two reasons. First, outlier removal using the epipolar geometry cannot reject all outliers in the stereo matches. Second, the affine triangulation is sensitive to noise naturally present in the 2D correspondences. In such case using a surface prior model such as a sphere for the retina [30], can make the

reconstruction better, but such a surface constraint may be too limiting. In the case of the open-sky pig eyes used for our tests, the retinas are far from spherical and can be of any smooth shape. We therefore propose to fit a single surface using Bicubic B-Splines (BBS) [211]. We use the image as the parametrization space for representing the surface. The surface $\Psi : \mathbf{u} \rightarrow \hat{\mathbf{x}}$ is thus a function of the image points $\mathbf{u} \in \Omega$ and the spline coefficients $\mathbf{c} \in \mathbb{R}^{2n_c}$, where n_c is the number of spline coefficients used to represent the surface. Consider there are n_r 3D points with the same number of 2D image correspondences. We express the surface reconstruction problem as:

$$\hat{\mathbf{c}} = \underset{\mathbf{c}}{\operatorname{argmin}} \sum_{i=1}^{n_r} \|\Psi(\mathbf{u}_i; \mathbf{c}) - \mathbf{x}_i\|_1 + \mu \int_{\Omega} \left\| \frac{\partial^2}{\partial \mathbf{u}^2} \Psi(\mathbf{u}; \mathbf{c}) \right\|_2^2 \tag{4.6}$$

Eq. 4.6 consists of a data term under the l_1-norm as well as a regularizer which penalizes very high frequency changes over the surface. The two terms are balanced by a hyperparameter $\mu \in \mathbb{R}^+$. We choose the l_1-norm to obtain a more robust surface fitting [39]. We also reject points where the l_1-norm of the data term exceeds a certain threshold ϵ. We then re-estimate the surface by solving Eq. 4.6 with the remaining points. The single iteration of point rejection and surface re-estimation gives a surface that is smooth and largely free of the reconstruction noise.

4.4.3 *Registration*

We define registration as the transformation of the camera pose and reconstructions to the robot coordinate frame. This is necessary because the stereo microscope (and the mounted cameras) may be moved during its use. Such motion can be measured from the images by the Perspective n-Point (PnP) method [111]. However, PnP cannot be used with the affine camera and we therefore compute registration using the reconstruction of the tool landmark positions and their positions measured by the robot kinematics.

Consider $(\mathsf{R}^c, \mathbf{t}^c)$, $c \in \{l, r\}$ to be the 6 DoF pose of the camera c with respect to its initial position, where $\mathsf{R}^c \in SO_3$ is the rotation and

$\mathbf{t}^c \in \mathbb{R}^3$ is the translation undergone by the microscope cameras. We then express the registration problem as:

$$\min_{\mathrm{R}^c,\ \mathbf{t}^c} \sum_{i=1}^{n_k n_f} \left\| \mathrm{R}^c \hat{\mathbf{x}}_i + \mathbf{t} - \mathbf{x}_i^{\mathrm{gt}} \right\|_2 \tag{4.7}$$

where $\hat{\mathbf{x}}_i$ is the ith triangulated tool landmark 3D point and $\mathbf{x}_i^{\mathrm{gt}}$ is the 3D ith tool landmark point as measured by the robot's measurement system. Eq. 4.7 is a well-studied problem and can be solved linearly using only three non-collinear points. In practice, a more accurate pose estimate is obtained by using multiple frames and accumulating 3D-3D correspondences, assuming a static camera within this time window.

4.5 EXPERIMENTS

4.5.1 *Dataset*

In order to train the deep network for keypoint localization, we manually annotated sequences of stereo images for the tool of interest with 3 keypoint locations. Such annotations are acquired with minimal effort, since for each stereo pair only 6 mouse-clicks are necessary. Apart from the manual 2D annotations, we acquired the 3D locations of the keypoints, from the kinematics of the robot. The dataset consists of 10 sequences, acquired from different pig eyes, with both artificial movements that help calibration and realistic movements performed by a surgeon. It includes more than 1500 full HD images and their labels. We limit acquisition to one type of tool, since our aim is accuracy rather than generalizing to different ones. The method itself is easily adaptable to other types of surgery, and tools with different landmarks. The dataset is publicly released to ease further research.

4.5.2 *Evaluation of Keypoint Localization*

Training details: For keypoint localization, we split the data into training and testing sets, and train the SHN model for 150 epochs. We use 7 sequences for training, and 3 for testing. Results for localization are reported for all images of the testing set. We use RMSProp [154] with $\alpha = 0.99$ and zero momentum. The initial learning rate is set to $5 \cdot 10^{-5}$, and is adapted by RMSProp for each of the layers. We use a standard

deviation of $\sigma = 5$ for the 2D gaussians centered on the keypoints. To avoid overfitting, we use extensive data augmentation that consists of random rotations $[-30°, 30°]$, and zooming $[0.75\times, 1.25\times]$. The images of the training set are randomly permuted, and a single model is trained for both the left and the right camera. For all our experiments, we train the models from scratch, in less than 4 hours with an NVidia Titan-X GPU. During testing, our batch contains both left and right images of the stereo camera. We note that during inference the CNN processes the images in higher than real time speed. Real time performance is especially important for 3D registration when the camera moves, and thus we keep all our experiments above the threshold of $30Hz$. We found that in practice, a stack of 2 Hourglasses is a fair compromise of speed and accuracy, when processing full-HD (1920×1080) images. Common models pre-trained on ImageNet [199, 74, 104] are much more memory and computation intensive, not allowing to process images at such resolution, let alone in real-time.

Evaluation metric: For evaluation of the 2D keypoint localization, we use the Percentage of Correct Keypoints (PCK) measure (also referred to as KBB [104]). In PCK, a detection is considered correct, if it falls 'near enough' to the label. The threshold is computed as a percentage of the distance of the tool-tip from its base.

Network architecture ablation: In order to decide on the final CNN architecture, we conduct an ablation experiment to show the importance of each of the used components. Starting from an encoder architecture like the ones used for image classification (without the fully connected layers), we observe poor performance (8.4% PCK) due to the heavily downsampled output. Adding the decoder architecture immediately solves this problem (75.1%). Skip connections and a second hourglass boost the overall performance further (95.2% and 99.6%). Substituting the convolutional modules by residual ones gives diminishing returns.

Architecture	Encoder	+Decoder	+Skip Connections	+Stacked	+Residual
PCK@0.05	8.4	75.1	95.2	**99.6**	**99.7**

Table 4.1: *CNN architecture ablation:* Various CNN architectures tested for keypoint localization and their quantitative contributions to the result.

Input image resolution: Fig. 4.3 illustrates the PCK measure as a function of the threshold for the accepted mis-localization, for various input image resolutions. For full-HD images, above the threshold of 1%, almost all detections are correct. The same accuracy is obtained for images of 480×640, for a threshold 6 times larger. Small errors in 2D lead to larger errors in 3D, so we argue that accurate 2D localization is crucial for the next steps of the pipeline, such as calibration (Fig. 4.5). Fig. 4.4 shows some qualitative examples of keypoint localization, obtained for high resolution images.

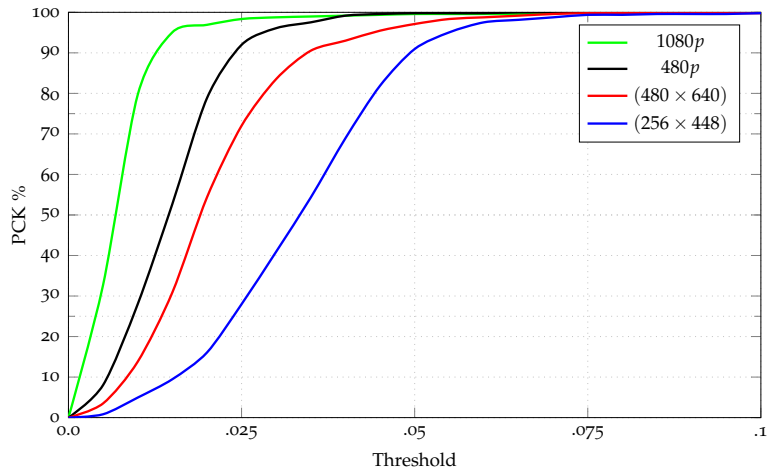


Figure 4.3: **Tool localization accuracy:** The PCK accuracy measure as a function of the maximum tolerance, for different input image resolutions. Tolerance is normalized by the size of the instrument tip.

Timing: Table 4.2 shows the execution rate of the CNN when the input resolution is varied. The timing regards the forward pass and the post-processing to obtain the locations of the landmarks from the heatmaps, for a batch of 2 images (left and right). Although we sacrifice execution speed for accuracy by using full HD images, the landmark localization remains faster than real-time (30Hz) at all resolutions. All experiments were conducted on a NVidia Titan-X GPU.

Table 4.2: *Execution Times*: Performance as a function of the input image resolution. All models achieve better than real-time performance.

Resolution	$1080p$	$480p$	480×640	256×448
Frequency (Hz)	35	77	95	**140**

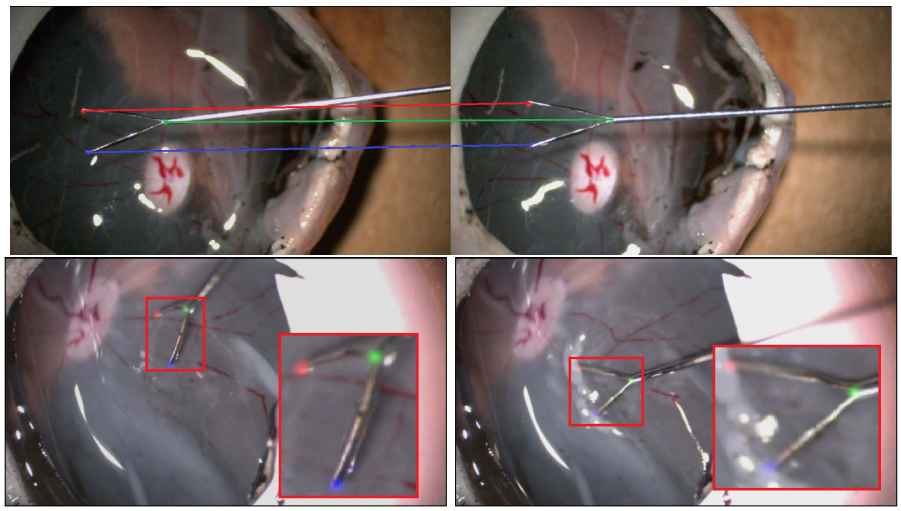


Figure 4.4: **Qualitative results for keypoint localization:** Keypoint localization on an example stereo image pair (top) and more qualitative example in another scenario (bottom).

4.5.3 *Evaluation of Calibration*

As to the calibration, we first investigate the influence of the image resolution used for keypoint detection. Fig. 4.5 shows the 3D reconstruction error and the 2D reprojection error for calibrations based on ground truth (GT) annotations and based on detections from four different image resolutions. Note that for 1080p we obtain virtually the same calibration quality as by using the manual annotation. As expected, the triangulation error as well as the reprojection error increase with lower resolutions. To validate the fitness of the affine camera model, we compare results with a perspective camera model calibrated with DLT from the full resolution annotations. This yields a much more unstable result compared to the affine model on the same data. Additionally,

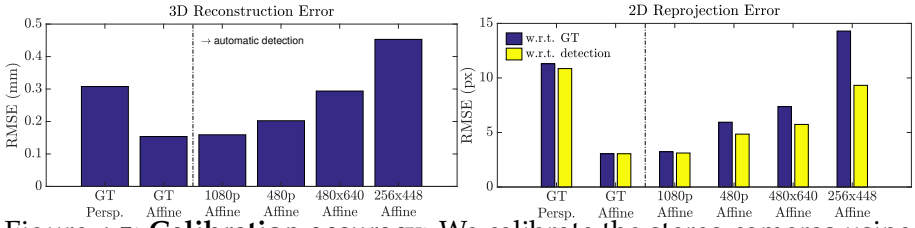


Figure 4.5: **Calibration accuracy:** We calibrate the stereo cameras using annotated tool keypoints (GT), and tool keypoint detections at four different image resolutions. The figures illustrate the 3D triangulation error (left) and 2D reprojection errors (right) for perspective and affine calibration. Clearly, the affine model performs better.

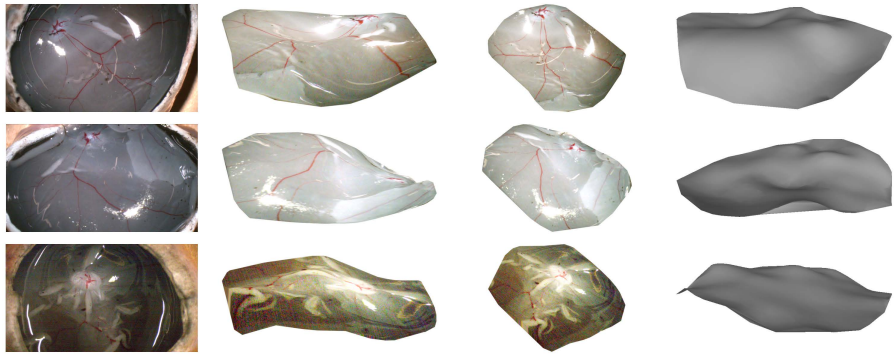


Figure 4.6: **Pig Eye Reconstruction:** Each row shows the result of our reconstruction method for a different pig eye.

decomposition of the perspective projection matrix is not possible due to the influence of the large focal length on the matrix conditioning.

Using our automatic affine calibration, we performed a second experiment to gauge reconstruction accuracy for a known planar calibration object. Instead of relying on the robot kinematics, we analyzed calibration accuracy by reconstructing points on a checkerboard with 0.5 mm squares. To minimize mismatching and correspondence noise we use manual correspondences refined by a corner detector. In this optimal setup, we observe a Root Mean Square Error (RMSE) of 25.479 µm in the reconstruction.

Figure 4.7: **Generic Object Reconstruction:** Each row shows the result of our reconstruction method for one object.

4.5.4 *Retinal Reconstruction and Tool Registration*

We reconstruct three open-sky pig eye sequences, each one for a different eye. The left camera image and the corresponding reconstructions are shown in Fig. 4.6. To qualitatively evaluate the reconstructions, we show some of everyday objects in Fig. 4.7. Like the pig-eye the objects are roughly 1cm in size. The reconstructions of the screw and leaf are particularly interesting because this shows that we can get high and low frequency surface aspects. Finally, we evaluate the registration using reconstructed tool points. Since there is no ground-truth label regarding the relative positions of the cameras, we synthetically move them by changing their projection matrices and measure the new pose using Eq. 4.7. We use one to several frames of the moving tool to measure the pose accuracy. Using $n_k = 3$ keypoints, we achieve an error below 150 µm after about 3 frames as shown in Fig. 4.8. This shows that we are able to quickly recover from 3D tracking failure in case the camera undergoes a change in pose by monitoring the consistency of the transformation over time. Note that the kinematics of our robot achieve accuracy of approximately 10 µm, whereas the diameter of a targeted vessel can range between 50 and 300 µm [53]. The online reconstruction pipeline in our proof-of-concept implementation runs at about 5Hz speed, the main bottleneck being the DeepMatching [214] method. Note that, although the speed may be increased further in the running system, currently envisioned applications do not necessarily require real-time reconstruction speed.

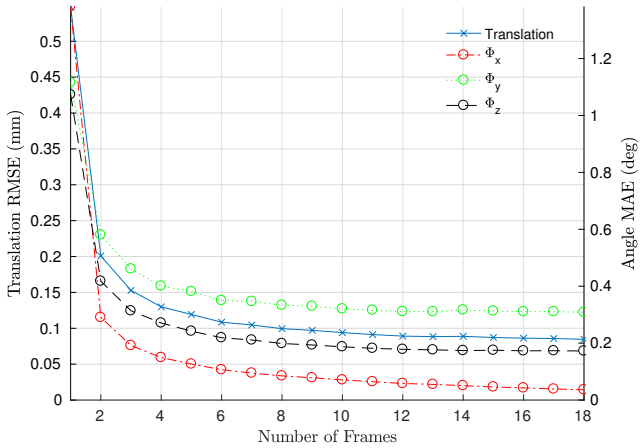


Figure 4.8: **Tool Registration:** Translational and rotational errors of the estimated alignment of the tool w.r.t. the camera frame. We express rotational errors with Euler angles (Φ_X, Φ_Y, and Φ_Z).

4.6 CONCLUSIONS

This chapter presented a new method for camera calibration, 3D reconstruction and registration, from automatically detected keypoints on a robotic end effector. Specifically, we use the 3D locations of the tool and the corresponding 2D locations on the acquired stereo images to establish correspondences, and initialize an affine bundle adjustment with the DLT method. We proposed a Stacked Hourglass CNN to detect the keypoints, which results in a very accurate and fast localization of the landmarks. We applied our method to robot-assisted eye surgery, where 3D processing is complicated due to various issues with microscope camera imaging and the quality of the acquired data. We validated each component of our pipeline independently and in combination. We created and released a database that can facilitate training CNNs for the task, and we show quantitative and qualitative results for all the steps of our algorithm. Results show high quality keypoint localization, 3D reconstruction, and registration, all in the context of a single pipeline.

5 ATTENTIVE SINGLE-TASKING OF MULTIPLE TASKS

In this chapter we address task interference in universal networks by considering that a network is trained on multiple tasks, but performs one task at a time, an approach we refer to as "Attentive Single-Tasking of Multiple Tasks" (ASTMT). The network thus modifies its behavior through task-dependent feature adaptation, or task attention. This gives the network the ability to accentuate the features that are adapted to a task, while shunning irrelevant ones. We further reduce task interference by forcing the task gradients to be statistically indistinguishable through adversarial training, ensuring that the common backbone architecture serving all tasks is not dominated by any of the task-specific gradients.

Results in three multi-task dense labeling problems consistently show: (i) a large reduction in the number of parameters while preserving, or even improving performance and (ii) a smooth trade-off between computation and multi-task accuracy.

5.1 INTRODUCTION

Real-world problems involve a multitude of visual tasks that call for multi-tasking, universal vision systems. For instance autonomous driving requires detecting pedestrians, estimating velocities and reading traffic signs, while identity recognition, pose, face and hand tracking are required for human-computer interaction.

A thread of works have introduced multi-task networks [194, 52, 73, 98] handling an increasingly large number of tasks. Still, it is common practice to train devoted networks for individual tasks when single-task performance is critical. This is supported by negative results from recent works that have aimed at addressing multiple problems with a single network [73, 98] - these have shown that there is a limit on performance imposed by the capacity of the network, manifested as a drop in performance when loading a single network with more tasks. Stronger backbones can uniformly improve multi-task performance,

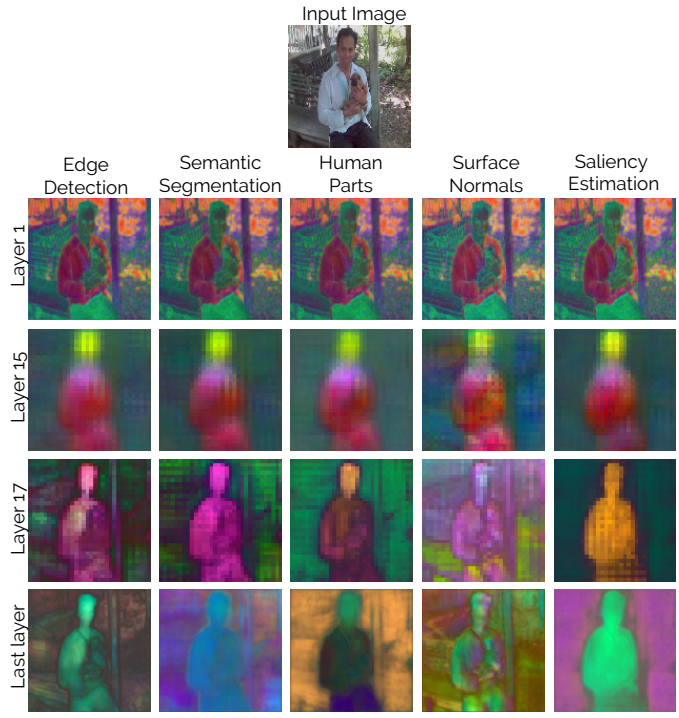


Figure 5.1: **Learned representations across tasks and layers:** We visualize how features change spatially in different depths of our multi-task network. For each layer (row) we compute a common PCA basis across tasks (column) and show the first three principal components as RGB values at each spatial location. We observe that the features are more similar in early layers and get more adapted to specific tasks as depth increases, leading to disentangled, task-specific representations in the later layers. We see for instance that the normal task features co-vary with surface properties, while the part features remain constant in each human part.

but still the per-task performance can be lower than the single-task performance with the same backbone.

This problem, known as task interference, can be understood as facing a the dilemma of invariance versus sensitivity: the most crucial information for one task can be a nuisance parameter for another, which leads to potentially conflicting objectives when training multi-task networks. An example of such a task pair is pose estimation

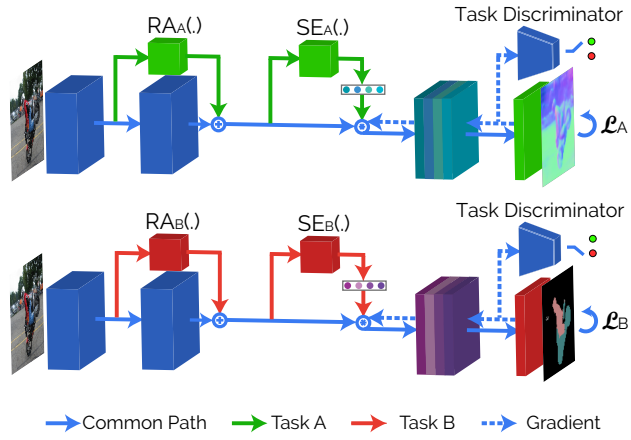


Figure 5.2: **Overview of ASTMT:** While using a shared backbone network, every task adapts its behavior in a separate, flexible, and lightweight manner, allowing us to customize computation for the task at hand. We refine features with a task-specific residual adapter branch (RA), and attend to particular channels with task-specific Squeeze-and-Excitation (SE) modulation. We also enforce the task gradients (dashed lines) to be statistically indistinguishable through adversarial training, further promoting the separation between task-specific and generic layers.

and object detection: when detecting or counting people the detailed pose information is a nuisance parameter that should be eliminated at some point from the representation of a network aiming at pose invariance [73]. At the same time, when watching a dance performance, one needs the detailed pose of the dancers, while ignoring the large majority of spectators. More generally this is observed when combining a task that is detail-oriented and requires high spatial acuity with a task that requires abstraction from spatial details, e.g. when one wants to jointly do low- and high-level vision. In other words, one task's noise is another one's signal.

We argue that this dilemma can be addressed by single-tasking, namely executing task a time, rather than getting all task responses in a single forward pass through the network. This reflects many practical setups, for instance when one sees the results of a single computational photography task at a time on the screen of a mobile phone, rather

than all of them jointly. Operating in this setup allows us to implement an "attention to task" mechanism that changes the network's behavior in a task-adapted manner, as shown in Fig. 5.1. We use the exact same network backbone in all cases, but we modify the network's behavior according to the executed task by relying on the most task-appropriate features. For instance when performing a low-level task such as boundary detection or normal estimation, the network can retain and elaborate on fine image structures, while shunning them for a high-level task that requires spatial abstraction.

We explore two different *task attention mechanisms*, as shown in Fig. 5.2. Firstly, we use data-dependent modulation signals [163] that enhance or suppress neuronal activity in a task-specific manner. Secondly, we use task-specific Residual Adapter [174] blocks that latch on to a larger architecture in order to extract task-specific information which is fused with the representations extracted by a generic backbone. This allows us to learn a shared backbone representation that serves all tasks but collaborates with task-specific processing to build more elaborate task-specific features.

These two extensions can be understood as promoting a disentanglement between the shared representation learned across all tasks and the task-specific parts of the network. Still, if the loss of a single task is substantially larger, its gradients will overwhelm those of others and disrupt the training of the shared representation. In order to make sure that no task abuses the shared resources we impose a *task-adversarial loss* to the network gradients, requiring that these are statistically indistinguishable across tasks. This loss is minimized during training through double back-propagation [49], and leads to an automatic balancing of loss terms, while promoting compartmentalization between task-specific and shared blocks.

5.2 RELATED WORK

This chapter draws ideas from several research threads.

Multiple Task Learning (MTL): Several works have shown that jointly learning pairs of tasks yields fruitful results in computer vision. Successful pairs include detection and classification [61, 178], detection and segmentation [73, 51], or monocular depth and segmentation [52, 221]. Joint learning is beneficial for unsupervised learning [172], when tasks provide complementary information (eg. depth boundaries and

motion boundaries [240]), in cases where task A acts as regularizer for task B due to limited data [123], or in order to learn more generic representations from synthetic data [182]. Xiao et al. [217] unify inhomogeneous datasets in order to train for multiple tasks, while [230] explore relationships among a large amount of tasks for transfer learning, reporting improvements when transferring across particular task pairs.

Despite these positive results, joint learning can be harmful in the absence of a direct relationship between task pairs. This was reported clearly in [98] where the joint learning of low-, mid- and high-level tasks was explored, reporting that the improvement of one task (e.g. normal detection) was to the detriment of another (e.g. object detection). Similarly, when jointly training for human pose estimation on top of detection and segmentation, Mask R-CNN performs worse than its two-task counterpart [73].

This negative result first requires carefully calibrating the relative losses of the different tasks, so that none of them deteriorates excessively. To address this problem, GradNorm [35] provides a method to adapt the weights such that each task contributes in a balanced way to the loss, by normalizing the gradients of their losses; a more recent work [200] extends this approach to homogenize the task gradients based on adversarial training. Following a probabilistic treatment [88] re-weigh the losses according to each task's uncertainty, while Sener and Koltun [193] estimate an adaptive weighting of the different task losses based on a pareto-optimal formulation of MTL. Similarly, [64] provide a MTL framework where tasks are dynamically sorted by difficulty and the hardest are learned first.

A second approach to mitigate task interference consists in avoiding the 'spillover' of gradients from one task's loss to the common features serving all tasks. One way of doing this is explicitly constructing complementary task-specific feature representations [189, 186], but results in an increase of complexity that is linear in the number of tasks. An alternative, adopted in the related problem of lifelong learning consists in removing from the gradient of a task's loss those components that would incur an increase in the loss of previous tasks [94, 128]. For domain adaptation [21] disentangle the representations learned by shared/task-specific parts of networks by enforcing similarity/orthogonality constraints. Adversarial Training has been used in the context of domain

adaptation [59, 123] to the feature space in order to fool the discriminator about the source domain of the features.

In our understanding these losses promote a compartmental operation of a network, achieved for instance when a block-structured weight matrix prevents the interference of features for tasks that should not be connected. A deep single-task implementation of this would be the gating mechanism of [5]. For multi-tasking, Cross Stitch Networks [146] automatically learn to split/fuse two independent networks in different depths according to their learned tasks, while [149] estimate a block-structured weight matrix during CNN training, and [191] search for the best combination of layers for different tasks, starting from a gigantic pre-specified network.

Attention mechanisms: Attention has often been used in deep learning to visualize and interpret the inner workings of CNNs [198, 231, 192], but has also been employed to improve the learned representations of convolutional networks for scale-aware semantic segmentation [32], fine-grained image recognition [57] or caption generation [222, 130, 10]. Squeeze and Excitation Networks [79] and their variants [215, 78] modulate the information of intermediate spatial features according to a global representation and be understood as implementing attention to different channels. Deep Residual Adapters [175, 174] modulate learned representations depending on their source domain. Several works study modulation for image retrieval [235] or classification tasks [163, 148], and embeddings for different artistic styles [50]. [226] learns object-specific modulation signals for video object segmentation, and [185] modulates features according to given priors for detection and segmentation. In our case we learn task-specific modulation functions that allow us to drastically change the network's behavior while using identical backbone weights.

5.3 ATTENTIVE SINGLE-TASKING MECHANISMS

Having a shared representation for multiple tasks can be efficient from the standpoint of memory- and sample- complexity, but can result in worse performance if the same resources are serving tasks with unrelated, or even conflicting objectives, as described above. Our proposed remedy to this problem consists in learning a shared representation for all tasks, while allowing each task to use this shared representation

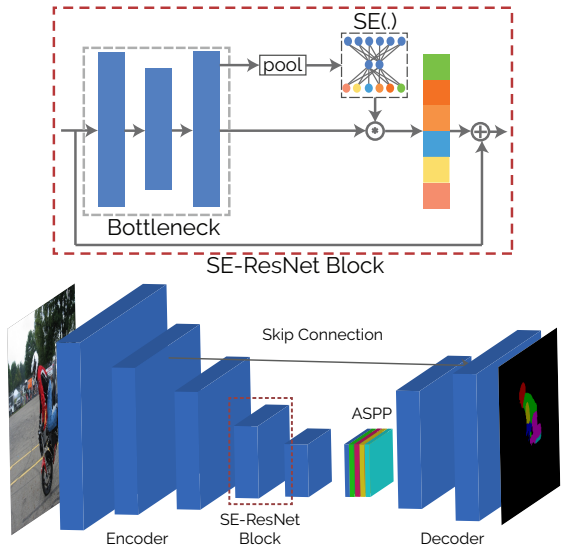


Figure 5.3: **Single-task network architecture:** We use Deeplab-v3+ with a Squeeze-and-Excitation (SE)-ResNet backbone. SE modules are present in all bottleneck blocks of the encoder and the decoder. Attentive multi-tasking uses different SE layers per task to modulate the network features in a task-specific manner.

differently for the construction of its own features.

5.3.1 *Task-specific feature modulation*

In order to justify our approach we start with a minimal example. We consider that we have two tasks A and B that share a common feature tensor $\mathrm{F}(x, y, c)$ at a given network layer, where x, y are spatial coordinates and $c = 1, \ldots, C$ are the tensor channels. We further assume that a subset $\mathcal{S}_A$ of the channels is better suited for task A, while $\mathcal{S}_B$ is better for B. For instance if A is invariant to deformations (detection) while B is sensitive (pose estimation), $\mathcal{S}_A$ could be features obtained by taking more context into account, while $\mathcal{S}_B$ would be more localized.

One simple way of ensuring that tasks A and B do not interfere while using a shared feature tensor is to hide the features of task B when training for task A:

$$\mathrm{F}_A(x, y, c) = m_A[c] \cdot \mathrm{F}(x, y, c) \tag{5.1}$$

where $m_A[c] \in \{0, 1\}$ is the indicator function of set $\mathcal{S}_A$. If $c \notin \mathcal{S}_A$ then $\mathrm{F}_A(x, y, c) = 0$, which means that the gradient $\frac{\partial \mathcal{L}_\mathrm{A}}{\partial \mathrm{F}(x,y,c)}$ sent by the loss $\mathcal{L}_A$ of task A to $c \in \mathcal{S}_A$ will be zero. We thereby avoid task interference since Task A does not influence nor use features that it does not need.

Instead of this hard choice of features per task we opt for a soft, differentiable membership function that is learned in tandem with the network and allows the different tasks to discover during training which features to use. Instead of a constant membership function per channel we opt for an image-adaptive term that allows one to exploit the power of the squeeze-and-excitation block [79].

In particular we adopt the squeeze-and-excitation (SE) block (also shown in Fig. 5.2), combining a global average pooling operation of the previous layer with a fully-connected layer that feeds into a sigmoid function, yielding a differentiable, image-dependent channel gating function. We set the parameters of this layer to be task-dependent, allowing every task to modulate the available channels differently. As shown in Section 5.5, this can result in substantial improvements when compared to a baseline that uses the same SE block for all tasks.

5.3.2 *Residual Adapters*

The feature modulation described above can be understood as shunning those features that do not contribute to the task while focusing on the more relevant ones. Intuitively, this does not add capacity to the network but rather cleans the signal that flows through it from information that the task should be invariant to. We propose to complement this by appending task-specific sub-networks that adapt and refine the shared features in terms of residual operations of the following form:

$$\mathrm{L}_A(x) = x + \mathrm{L}(x) + \mathrm{RA}_A(x), \tag{5.2}$$

where $\mathrm{L}(x)$ denotes the default behaviour of a residual layer, RA_A is the task-specific residual adapter of task A, and $\mathrm{L}_A(x)$ is the modified layer. We note that if $\mathrm{L}(x)$ and $\mathrm{RA}_A(x)$ were linear layers this would amount to the classical regularized multi-task learning of [55].

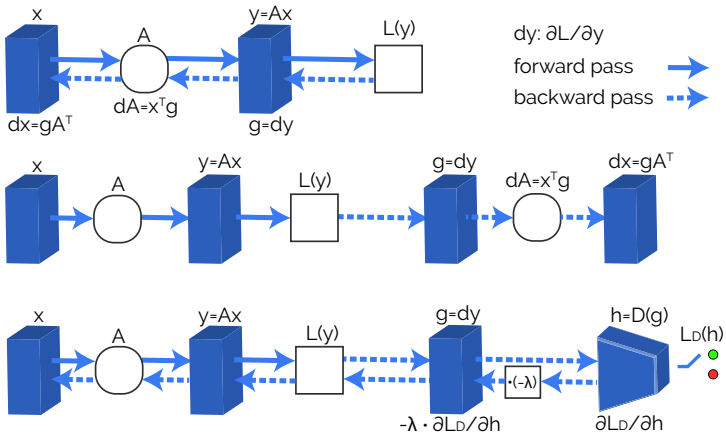


Figure 5.4: **Illustration of double back-propagation:** Double backprop [49] exposes the gradients computed during backprop (row 1) by unfolding the computation graph of gradient computation (row 2). Exposing the gradients allows us to train them in an adversarial setting by using a discriminator, forcing them to be statistically indistinguishable across tasks (row 3). The shared network features x then receive gradients that have the same distribution irrespective of the task, ensuring that no task abuses the shared network, e.g. due to higher loss magnitude. The gradient of the discriminator is reversed (negated) during adversarial training, and the parameter $\lambda \in [0, 1]$ controls the amount of negative gradient that flows back to the network [59].

These adapters introduce a task-specific parameter and computation budget that is used in tandem with that of the shared backbone network. We show in Section 5.5 that this is typically a small fraction of the budget used for the shared network, but improves accuracy substantially.

When employing disentangled computation graphs with feature modulation through SE modules and/or residual adapters, we also use task-specific batch-normalization layers, that come with a trivial increase in parameters (while the computational cost remains the same).

The idea behind the task-specific adaptation mechanisms described above is that even though a shared representation has better memory/computation complexity, every task can profit by having its own 'space', i.e. separate modelling capacity to make the best use of the representation - by modulating the features or adapting them with residual blocks.

Pushing this idea further we enforce a strict separation of the shared and task-specific processing, by requiring that the gradients used to train the shared parameters are statistically indistinguishable across tasks. This ensures that the shared backbone serves all tasks equally well, and is not disrupted e.g. by one task that has larger gradients.

We enforce this constraint through adversarial learning. Several methods, starting from Adversarial Domain Adaptation [60], use adversarial learning to remove any trace of a given domain from the learned mid-level features in a network; a technique called Adversarial multi-task training [123] falls in the same category.

Instead of removing domain-specific information from the features of a network (which serves domain adaptation), we remove any task-specific trace from the gradients sent from different tasks to the shared backbone (which serves a division between shared and task-specific processing). A concurrent work [200] has independently proposed this idea.

As shown in Fig. 5.4 we use double back-propagation [49] to 'expose' the gradient sent from a task t to a shared layer l, say $g_t(l)$. By exposing the variable we mean that we unfold its computation graph, which in turn allows us to back-propagate through its computation. By back-propagating on the gradients we can force them to be statistically indistinguishable across tasks through adversarial training.

In particular we train a task classifier on top of the gradients lying at the interface of the task-specific and shared networks and use sign negation to make the task classifier fail [59]. This amounts to solving the following optimization problem in terms of the discriminator weights, w_D and the network weights, w_N:

$$\min_{w_D} \max_{w_N} L(D(g_t(w_N), w_D), t), \tag{5.3}$$

where $g_t(w_N)$ is the gradient of task t computed with w_N, $D(\cdot, w_D)$ is the discriminator's output for input $\cdot$, and $L(\cdot, t)$ is the cross-entropy loss for label t that indicates the source task of the gradient.

Intuitively this forces every task to do its own processing within its own blocks, so that it does not need from the shared network anything different from the other tasks. This results in a separation of the network into disentangled task-specific and shared compartments.

5.5 EXPERIMENTAL EVALUATION

Datasets: We validate our approach on different datasets and tasks. We focus on dense prediction tasks that can be approached with fully convolutional architectures. Most of the experiments are carried out on the PASCAL [54] benchmark, which is popular for dense prediction tasks. We also conduct experiments on the smaller NYUD [151] dataset of indoor scenes, and the recent, large scale FSV [100] dataset of synthetic images. Statistics, as well as the different tasks used for each dataset are presented in Table 5.2.

Base architecture: We use our re-implementation of Deeplab-v3+ [33] as the base architecture of our method, due to its success on dense semantic tasks. Its architecture is based on a strong ResNet encoder, with a-trous convolutions to preserve reasonable spatial dimensions for dense prediction. We use the latest version that is enhanced with a parallel a-trous pyramid classifier (ASPP) and a powerful decoder. We refer the reader to [33] for more details. The ResNet-101 backbone used in the original work is replaced with its Squeeze-and-Excitation counterpart (Fig. 5.3), pre-trained on ImageNet [188]. The pre-trained SE modules serve as an initialization point for the task-specific modulators for multi-tasking experiments.

The architecture is tested for a single task in various competitive benchmarks for dense prediction: edge detection, semantic segmentation, human part segmentation, surface normal estimation, saliency, and monocular depth estimation. We compare the results obtained with various competitive architectures. For edge detection we use the BSDS500 [144, 11] benchmark and its optimal dataset F-measure (`odsF`) [143]. We use the fast SEISM [167] library for boundary detection evaluation. For semantic segmentation we train on PASCAL VOC `trainaug` [54, 69] (10582 images), and evaluate on the validation set of PASCAL using mean intersection over union (`mIoU`). For human part

Task	Dataset	Metric	R-101	strong baseline
Edge	BSDS500	odsF ↑	82.5	81.3 [97]
S.Seg	VOC	mIoU ↑	78.9	79.4 [33]
H. Parts	P. Context	mIoU ↑	64.3	64.9* [31]
Normals	NYUD	mErr ↓	20.1	19.0 [13]
Saliency	PASCAL-S	maxF ↑	84.0	83.5 [98]
Depth	NYUD	RMSE ↓	0.56	0.58 [221]

Table 5.1: **Architecture capacity**: We report upper-bounds of performance that can be reached on various competitive (but inhomogeneous) datasets by our architecture, and compare to strong task-specific baselines. All experiments are initialized from ImageNet pre-trained weights (* means that COCO pretraining is included). The arrow indicates better performance for each metric.

segmentation we use PASCAL-Context [34] and `mIoU`. For surface normals we train on the raw data of NYUD [151] and evaluate on the test set using mean error (`mErr`) in the predicted angles as the evaluation metric. For saliency we follow [98] by training on MSRA-10K [36], testing on PASCAL-S [114] and using the maximal F-measure (`maxF`) metric. Finally, for depth estimation we train and test on the fully annotated training set of NYUD using root mean squared error (`RMSE`) as the evaluation metric. For implementation details, and hyper-parameters, please refer to the Appendix.

Table 5.1 benchmarks our architecture against popular state-of-the-art methods. We obtain competitive results, for all tasks. We emphasize that these benchmarks are inhomogeneous, i.e. their images are not annotated with all tasks, while including domain shifts when training for multi-tasking (eg. NYUD contains only indoor images). In order to isolate performance gains/drops as a result of multi-task learning (and not domain adaptation, or catastrophic forgetting), in the experiments that follow, we use homogeneous datasets.

Multi-task learning setup: We proceed to multi-tasking experiments on PASCAL. We keep the splits of PASCAL-Context, which provides labels for edge detection, semantic segmentation, and human part segmentation. In order to keep the dataset homogeneous and the architecture identical for all tasks, we did not use instance level tasks (detection,

pose estimation) that are provided with the dataset. To increase the number of tasks we automatically obtained ground-truth for surface normals and saliency through label-distillation using pre-trained state-of-the-art models ([13] and [33], respectively), since PASCAL is not annotated with those tasks. For surface normals, we masked out predictions from unknown and/or invalid classes (eg. sky) during both training and testing. In short, our benchmark consists of 5 diverse tasks, ranging from low-level (edge detection, surface normals), to mid-level (saliency) and high-level (semantic segmentation, human part segmentation) tasks.

Evaluation metric: We compute multi-tasking performance of method m as the average per-task drop with respect to the single-tasking baseline b (i.e different networks trained for a single task each):

$$\Delta_m = \frac{1}{T}\sum_{i=1}^{T}(-1)^{l_i}\left(M_{m,i} - M_{b,i}\right)/M_{b,i} \tag{5.4}$$

where $l_i = 1$ if a lower value means better for measure M_i of task i, and 0 otherwise. Average relative drop is computed against the baseline that uses *the same backbone*.

To better understand the effect of different aspects of our method, we conduct a number of ablation studies and present the results in Tables (5.3-5.7).

We construct a second baseline, which tries to learn all tasks simultaneously with a single network, by connecting T task-specific convolutional classifiers (1×1 conv layers) at the end of the network. As also reported by [98], a non-negligible average performance drop can be observed (-6.6% per task for R-26 with SE). We argue that this drop is mainly triggered by conflicting gradients during learning.

Effects of modulation and adversarial training: Next, we introduce the modulation layers described in Section 5.3. We compare parallel residual adapters to SE (Table 5.4) when used for task modulation. Performance per task recovers immediately by separating the computation used by each task during learning (-1.4 and -0.6 vs. -6.6 for adapters and SE, respectively). SE modulation results in better performance, while using slightly fewer parameters per task. We train a second variant where we keep the computation graph identical for all tasks in the encoder, while using SE modulation only in the decoder (Table 5.5). Interestingly, this variant reaches the performance of residual adapters (-1.4), while being much more efficient in terms of number of

Database	Type	# Train Im.	# Test Im.	Edge	S.Seg	Parts	Normals	Sal	Depth	Albedo
PASCAL	Real	4,998	5,105	✓	✓	✓	✓*	✓*		
NYUD	Real	795	654	✓	✓		✓		✓	
FSV	Synth	223,197	50,080		✓				✓	✓

Table 5.2: **Multi-task benchmark statistics**: We conduct the main volume of experiments on PASCAL for 5 tasks (* labels obtained via distillation). We also use the fully labelled subsets of NYUD, and the synthetic FSV dataset.

SE-bb	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑
	1	70.3	63.98	55.85	15.11	63.92
✓	1	**71.3**	**64.93**	**57.12**	**14.90**	**64.17**
	5	68.0	58.59	53.80	**16.68**	60.71
✓	5	**69.2**	**60.20**	**54.10**	17.04	**62.10**

Table 5.3: **Baselines in PASCAL:** Using SE blocks in ResNet backbones (SE-bb) improves results. In all our experiments we use SE-bb baselines for fair comparison.

parameters and computation, as only one forward pass of the encoder is necessary for all tasks.

In a separate experiment, we study the effects of adversarial training described in Section 5.4. We use a simple, fully convolutional discriminator to classify the source of the gradients. Results in Table 5.6 show that adversarial training is beneficial for multi-tasking, increasing performance compared to standard multi-tasking (-4.4 vs -6.6). Even though the improvements are less significant compared to modulation, they come without extra parameters or computational cost, since the discriminator is used only during training.

SE	RA	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
		1	71.3	64.93	57.12	14.90	64.17	
		5	69.2	60.20	54.10	17.04	62.10	6.62
	✓	5	70.5	62.80	56.41	15.27	64.84	**1.42**
✓		5	71.1	64.00	56.84	15.05	64.35	**0.59**

Table 5.4: **Type of Modulation:** Both SE and RA are effective modulation methods. Results on PASCAL.

enc	dec	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
		1	71.3	64.93	57.12	14.90	64.17	
		5	69.2	60.20	54.10	17.04	62.1	6.62
	✓	5	70.6	63.33	56.73	15.14	63.23	**1.44**
✓	✓	5	71.1	64.00	56.84	15.05	64.35	**0.59**

Table 5.5: **Location of SE modulation:** Modulating varying portions of the network (e.g. encoder or decoder) allows trading off performance and computation. Results in PASCAL.

mod	A	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
		1	71.3	64.93	57.12	14.90	64.17	
		5	69.2	60.20	54.10	17.04	62.10	6.62
	✓	5	69.7	62.20	55.04	16.17	62.19	**4.34**
✓		5	71.1	64.00	56.84	15.05	64.35	**0.59**
✓	✓	5	71.0	64.61	57.25	15.00	64.70	**0.11**

Table 5.6: **Adversarial training:** Experiments on PASCAL show that adversarial training is beneficial both w/ and w/o SE modulation.

backbone	SEA	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
R-26		1	71.3	64.93	57.12	14.90	64.17	
R-26		5	69.2	60.20	54.10	17.04	62.10	6.62
R-26	✓	5	71.0	64.61	57.25	15.00	64.70	**0.11**
R-50		1	72.7	68.30	60.70	14.61	65.40	
R-50		5	69.2	63.20	55.10	16.04	63.60	6.81
R-50	✓	5	72.4	68.00	61.12	14.68	65.71	**0.04**
R-101		1	73.5	69.76	63.48	14.15	67.41	
R-101		5	70.5	66.45	61.54	15.44	66.39	4.50
R-101	✓	5	73.5	68.51	63.41	14.37	67.72	**0.60**

Table 5.7: **Backbones:** Improvements on PASCAL from SE modulation with adversarial training (SEA) are observed regardless of the capacity/depth of the backbones.

SEA	#T	Edge ↑	Seg ↑	Norm ↓	Depth ↓	$\Delta_m\%$ ↓
	1	74.4	32.82	23.30	0.61	
	4	73.2	30.95	23.34	0.70	5.44
✓	4	74.5	32.16	23.18	0.57	**-1.22**

Table 5.8: **Improvements from SE with modulation (SEA) transfer to NYUD dataset:** We report average performance drop with respect to single task baselines. We use R-50 backbone.

SEA	#T	Seg ↑	Albedo ↓	Disp ↓	$\Delta_m\%$ ↓
	1	71.2	0.086	0.063	
	3	66.9	0.093	0.078	7.04
✓	3	70.7	0.085	0.063	**-0.02**

Table 5.9: **Improvements from SE with modulation (SEA) transfer to FSV dataset:** We report average performance drop with respect to single task baselines. We use R-50 backbone.

The combination of SE modulation with adversarial training (Table 5.6) leads to additional improvements (-0.1% worse than the single-task baseline), while further adding residual adapters surpasses single-tasking (+0.45%), at the cost of 12.3% more parameters per task (Fig. 5.5).

Deeper Architectures: Table 5.7 shows how modulation and adversarial training perform when coupled with deeper architectures (R-50 and R-101). The results show that our method is invariant to the depth of the backbone, consistently improving the standard multi-tasking results.

Resource Analysis: Figure 5.5 illustrates the performance of each variant as a function of the number of parameters, as well as the FLOPS (multiply-adds) used during inference. We plot the relative average per-task performance compared to the single-tasking R-101 variant (blue cross), for the 5 tasks of PASCAL. Different colors indicate different backbone architectures. We see a clear drop in performance by standard multi-tasking (crosses vs. circles), but with fewer parameters and FLOPS. Improvements due to adversarial training come free of cost (triangles) with only a small overhead for the discriminator during training.

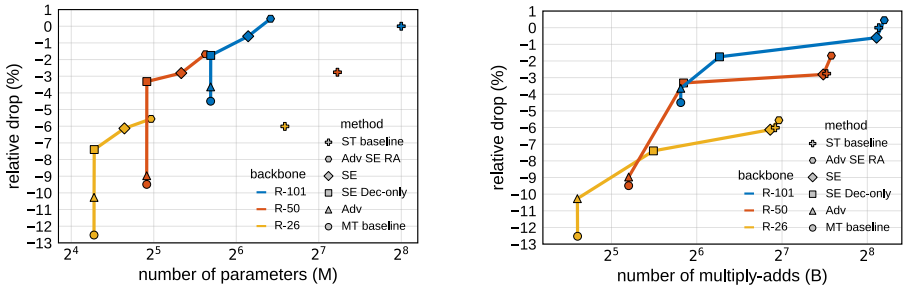


Figure 5.5: **Performance vs. Resources:** Average relative drop (Δ_m%) as a function of the number of parameters (left), and multiply-adds (right), for various points of operation of our method. We compare 3 different backbone architectures, indicated with different colors. We compare against single-tasking baseline (ST baseline), and multi-tasking baseline (MT baseline). Performance is measured relative to the best single-tasking model (R-101 backbone). An increase in performance comes for free with adversarial training (Adv). Modulation per task (SE) results in large improvements in performance, thanks to the disentangled graph representations, albeit with an increase in computational cost if used throughout the network, instead of only on the decoder (SE Dec-only vs. SE). We observe a drastic drop in number of parameters needed for our model in order to reach the performance of the baseline (SE, Adv). By using both modulation and adversarial training (Adv SE RA), we are able to reach single-task performance, with far fewer parameters.

Including modulation comes with significant improvements, but also with a very slight increase of parameters and a slight increase of computational cost when including the modules on the decoder (rectangles). The increase becomes more apparent when including those modules in the encoder as well (diamonds). Our most accurate variant using all of the above (hexagons) outperforms the single-tasking baselines by using only a fraction of their parameters.

We note that the memory and computational complexities of the SE blocks and the adapters are negligible, but since it affects the outputs of the layer it means that we cannot share the computation of the ensuing layers across all tasks, and thus the increased number of multiply-adds.

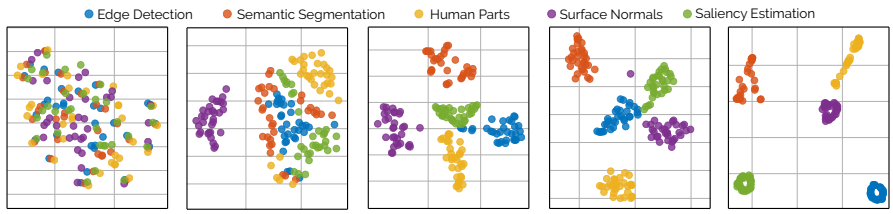


Figure 5.6: **t-SNE visualization of task-dependent feature activations of a single image:** Illustration at increasing depths of the network (from left to right). Features in early layers are more similar across tasks and progressively get more adapted to specific tasks in later layers.

Learned Disentangled Representations: In order to highlight the importance of task modulation, we plot the learned representations for different tasks in various depths of our network. Figure 5.6 shows the t-SNE representations [131] of the SE activations in equally spaced levels of the network. The activations are averaged for the first 32 samples of the validation set, following [79], and they are sorted per task. The resulting plots show that in the early stages of the network the learned representations are almost identical. They gradually become increasingly different as depth grows, until they are completely different for different tasks at the level of the classifier. We argue that this disentanglement of learned representations also translates to performance gains, as shown in Tables (5.3-5.7).

Validation on additional datasets: We validate our approach in two additional datasets, NYUD [151] and FSV [100]. NYUD is an indoor dataset, annotated with labels for instance edge detection, semantic segmentation into 41 classes, surface normals, and depth. FSV is a large-scale synthetic dataset, labelled with semantic segmentation (3 classes), albedo, and depth (disparity).

Table 5.9 presents our findings for both datasets. As in PASCAL, when we try to learn all tasks together, we observe a non-negligible drop compared to the single-tasking baseline. Performance recovers when we plug in modulation and adversarial training. Interestingly, in NYUD and FSV we observe larger improvements compared to PASCAL. Our findings are consistent with related works [221, 52] which report improved results for multi-tasking when using depth and semantics.

Figures 5.7 and 5.8 illustrate some qualitative examples, obtained by our method in PASCAL and NYUD, respectively. Results in each row are obtained with a single network. We compare our best model to the baseline architecture for multi-tasking (without per-task modulation, or adversarial training). We observe a quality degradation in the results of the baseline. Interestingly, some errors are obtained clearly as a result of standard multi-tasking. Edge features appear during saliency estimation in Fig 5.7, and predicted semantic labels change on the pillows, in areas where the surface normals change, in Fig 5.8. In contrast, our method provides disentangled predictions that are able to recover from such issues, reach, and even surpass the single-tasking baselines.

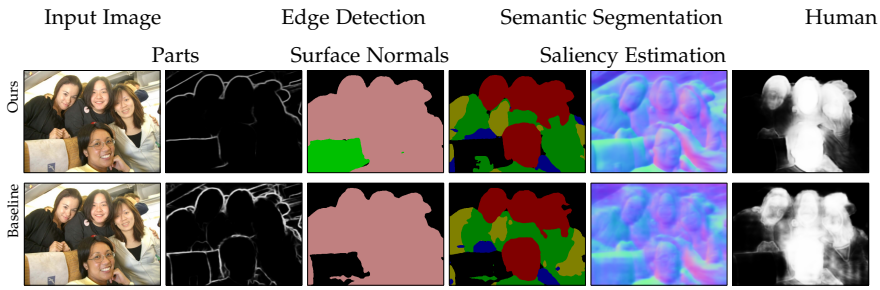


Figure 5.7: **Qualitative Results in PASCAL:** We compare our model against standard multi-tasking. For the baseline, features from edge detection appear in saliency estimation results, indicating the need to disentangle the learned representations.

5.6 Additional details and experimental evaluation

5.6.1 *More results on NYUD and FSV*

Table 5.10 illustrates the quantitative results obtained by our method on NYUD [151] and FSV [100], by changing the backbone architecture. Results are consistent among backbones, and by including modulation and adversarial training to the pipeline, we get improved results with respect to the multi-task and single-task baselines, irrespective of the network depth.

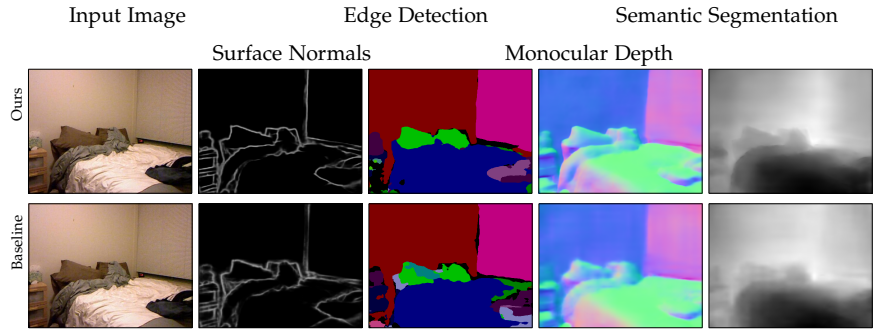


Figure 5.8: **Qualitative Results in NYUD**: We compare our model against standard multi-tasking. The baseline predicts blurry edges and depth, as well as inconsistent labels on the pillow (where surface normals change). Our method is able to recover from these issues.

5.6.2 *Connection of ASTMT to UberNet*

The architecture of [98] learns multiple tasks by using a common backbone and a light-weight decoder (skip connections and 1×1 convolutions) per task. We re-implement the UberNet architecture, and we substitute the VGG [199] backbone of the original work with the SE-ResNet [79] used in this chapter. The main difference with our architecture are the skip connections and 1×1 convolutions that comprise each task-specific head, instead of the powerful Deeplab-v3+ ASPP decoder [33] used for ASTMT. Similarly to when training ASTMT, and similarly to the observation of the original author, we observe a non-trivial drop in performance when learning to multi-task with a common, entirely shared backbone (Table 5.11). We plug-in SE and adversarial training and we recover most of the drop. We provide results for all 3 datasets that have been used throughout this chapter. Results obtained by our architecture are presented in the last row of each table. The relatively lower performance especially for semantic tasks compared to our method is due to the absence of a strong decoder. The observations regarding modulation and adversarial training are, nevertheless, consistent.

backbone	SEA	#T	Edge ↑	Seg ↑	Norm ↓	Depth ↓	Δ_m% ↓
R-26		1	72.9	29.87	24.34	0.650	0
R-26		4	72.4	27.74	24.83	0.729	5.50
R-26	✓	4	73.5	30.07	24.316	0.625	**-1.36**
R-50		1	74.4	32.82	23.3	0.610	
R-50		4	73.2	30.95	23.34	0.700	5.44
R-50	✓	4	74.5	32.16	23.18	0.570	**-1.22**
R-101		1	74.9	35.90	22.90	0.580	
R-101		4	73.8	31.20	23.07	0.650	6.63
R-101	✓	4	75.6	35.60	22.73	0.560	**-1.07**

backbone	SEA	#T	Seg ↑	Albedo ↓	Depth ↓	Δ_m% ↓
R-26		1	69.77	0.087	0.065	0
R-26		3	66.71	0.090	0.073	6.41
R-26	✓	3	71.36	0.085	0.065	**-1.80**
R-50		1	71.14	0.086	0.063	0
R-50		3	66.90	0.093	0.078	7.04
R-50	✓	3	70.69	0.085	0.063	**-0.02**
R-101		1	72.10	0.086	0.063	0
R-101		3	68.12	0.091	0.072	8.75
R-101	✓	3	72.24	0.083	0.062	**-1.57**

Table 5.10: **ASTMT for NYUD (top), and FSV (bottom)**: Results with different backbones: R-26, R-50, and R-101. Negative drop indicates improved performance with respect to the single-tasking baseline. Arrows indicate desired behaviour of each metric.

5.6.3 *ASTMT with MobileNet-v2 backbone*

Our multi-tasking framework could find application in light-weight CNNs designed for mobile phone applications. For example, by using our framework, many different tasks can be executed with only a single and small set of parameters being shipped to the end user. To test this idea, we change our backbone to the light-weight MobileNet-v2 [190], an architecture specifically designed for mobile phones. We change the decoder accordingly: convolutions are changed to depth-wise convolutions, and ReLU activations are changed to ReLU6. We pre-train a variant that uses Squeeze and Excitation modules on ImageNet

backbone	SEA	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
R-50-Uber		1	71.7	66.90	59.80	15.00	64.56	
R-50-Uber		5	70.3	60.90	57.00	16.65	62.15	7.10
R-50-Uber	✓	5	70.5	65.50	60.15	14.94	64.98	**0.43**
R-50	✓	5	72.4	68.00	61.10	14.80	65.70	n.a

backbone	SEA	#T	Edge ↑	Seg ↑	Norm ↓	Depth ↓	Δ_m% ↓
R-50-Uber		1	73.9	32.50	22.90	0.669	0
R-50-Uber		4	72.3	29.48	24.16	0.716	6.00
R-50-Uber	✓	4	73.7	31.19	23.46	0.632	**0.30**
R-50	✓	4	74.5	32.20	23.20	0.570	n.a

backbone	SEA	#T	Seg ↑	Albedo ↓	Depth ↓	Δ_m% ↓
R-50-Uber		1	70.26	0.092	0.101	0
R-50-Uber		3	67.02	0.093	0.124	9.50
R-50-Uber	✓	3	69.45	0.091	0.111	**3.32**
R-50	✓	3	70.70	0.085	0.063	n.a

Table 5.11: **UberNet for PASCAL (top), NYUD (mid), and FSV (bottom)**: Standard multi-task learning results in a significant drop in performance, that is recovered with modulation and adversarial training. Last rows of each table present results obtained by our architecture.

(SE-MobileNet), and fine-tune for multi-task learning. We test standard multi-tasking against the variants of our method that use SE modulation. Table 5.12 summarizes our findings. Similarly to the experiments using SE-ResNet, disentangling the representations for each task also helps for MobileNet. By using SE per task both on the encoder and decoder, our method outperforms the single-tasking baseline. Figure 5.9 puts these results in perspective, comparing them to results obtained by SE-ResNet architecture. It is remarkable that by using modulation, MobileNet is no worse than the R-50 standard multi-tasking baseline using much less computational cost, and only 8% of its parameters.

5.6.4 *Implementation Details*

In this section we provide the technical details for our implementation.

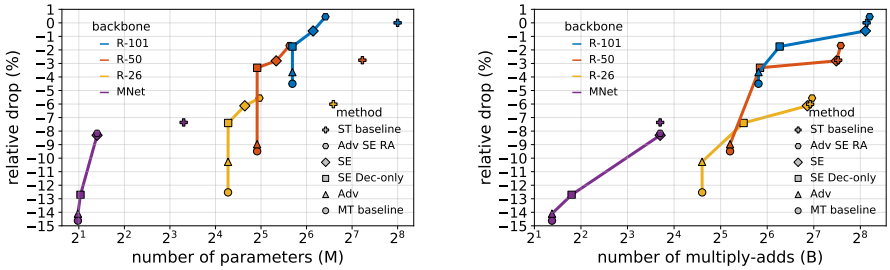


Figure 5.9: **Performance vs. Resources for MobileNet:** Average relative drop as a function of the number of parameters (left), and multiply-adds (right), for various points of operation of our method. Different backbones are indicated with different colors. MobileNet with our modulation is able to reach R-50 results for standard multi-tasking, by using much less parameters and computation.

backbone	enc	dec	#T	Edge ↑	Seg ↑	Parts ↑	Norm ↓	Sal ↑	Δ_m% ↓
MNet			1	69.5	62.10	54.88	14.88	66.30	0
MNet			5	67.2	54.10	53.00	16.76	62.70	7.57
MNet		✓	5	67.5	57.40	54.50	16.55	63.00	5.47
MNet	✓	✓	5	69.2	61.60	55.17	15.21	65.60	**0.97**

Table 5.12: **Results using MobileNet in PASCAL**: Modulation with SE is able to recover the performance that is lost using standard multi-task learning.

Generic hyper-parameters: The entire hyper-parameter search was performed on the single-task baselines. For all tasks, we use synchronized SGD with momentum of 0.9 and weight decay of 1e-4. We set the initial learning rate to 0.005 and use the poly learning rate [31]. All our models are trained on a single GPU with batch size 8, and spatial input of 512×512. We used multi-GPU training with batch size 16 and synchronous batchnorm layers only for the sanity check experiments (Table 5.1), for a fair comparison with competing methods. Standard flipping, rotations, and scaling was used for data augmentation. The number of total epochs is set to 60 for PASCAL [54], 200 for NYUD [151], and 3 for the large-scale FSV [100].

Weighting of the losses: Related work deals with automatically weighting of the losses for multi-task learning [35, 193]. We compared

these methods to selecting the optimal weights by grid-search. In our setup, we found out that grid-search works better, probably because of the very imbalanced optimal parameters (optimal weighting of loss for edge detection is 50 times higher than semantic segmentation, for example). In particular, we re-implemented [193] that uses multi-objective optimization to re-weight the gradients from each task, in order to optimize the shared parts of the network towards a direction useful for all tasks. This lead to better results than uniform weights, however we obtained better results by simple grid-search.

For all our experiments, when training for multiple tasks, we divide the learning rate of the shared layers by the number of tasks (T), since T updates are happening for the same mini-batch on the shared part of the network.

During training, we used the following formula for the weights of the losses:

$$L = (1 - w_d) \cdot \sum_{t=1}^{T} w_t \cdot L_t + w_d \cdot L_d, \tag{5.5}$$

where w_t weights the loss L_t of task t, and w_d the loss L_d of the discriminator. All losses are averaged to the number of samples that the prediction contains ($W \times H \times C \times N$).

Edge Detection: For edge detection, we use $w_t = 50$ and binary cross-entropy loss. As is common practice [220, 97, 139], the positive pixels are weighted more (0.95) than the negative ones (0.05), to account for the class imbalance. When training for a single task in BSDS (Table 5.1), where there are more than a single annotators, we use the multi-instance learning (MIL) loss of [97]. No MIL is used when training in PASCAL or NYUD. We follow the common evaluation on those two datasets [139] by setting the maximum allowed mis-localization of edges (`maxDist` parameter) to 0.0075 and 0.011 for PASCAL and NYUD, respectively.

Semantic Segmentation: For semantic segmentation, we used $w_t = 1$, and cross-entropy loss. When training on VOC trainaug [33] (Table 5.1), we did not finetune separately on VOC val.

Human Part Segmentation: For human part segmentation, we used $w_t = 2$. and cross-entropy loss. Samples that do not contain any humans did not contribute to the loss.

Surface Normals: For surface normal estimation, we used $w_t = 10$ and $\mathcal{L}_1$ loss with unit-vector normalization. During rotation augmentation, we carefully rotate the unit vector of the surface normals accordingly, to point to a consistent direction.

Saliency: For saliency, we used $w_t = 5$ and the balanced binary cross-entropy loss, as in the case of edge detection.

Albedo: For albedo, we use $w_t = 10$, and standard $\mathcal{L}_1$ loss. We emphasize that our architecture is not optimal for this task, since the output is 4 times smaller than the input, and tiny details are not captured. Using an architecture suited for albedo is out of the scope of this chapter.

Monocular Depth: For monocular depth estimation we use $w_t = 1$, and the loss of [52], which is a combination of $\mathcal{L}_1$ loss and a smoothness term that enforces the spatial gradients of the prediction to be consistent with the ones of the ground truth. Our experiments showed that including the smoothness term to the loss leads to better results, quantitatively and qualitatively.

Discriminator: We use a fully-convolutional discriminator, which consists of two 1×1 conv layers and a ReLU activation. We did not observe improvements when using discriminators of larger depth. We normalize the gradient of the losses of the tasks by their norm before passing them through the discriminator. This practice makes training more stable because the norm of the gradients becomes smaller as training progresses. We use $w_d = 0.1$.

Further Technical Details: ASTMT is implemented in PyTorch [158]. During weight update PyTorch applies momentum and weight decay to all modules in the definition of a network. This behaviour is not desired When using generic and task-specific weights, since the task-specific ones are only used in the forward pass of the particular task, which leads to $T - 1$ unwanted updates. This behaviour is avoided by tracing the graph of computation and updating only the weights that were used, which also translated into quantitative improvements.

5.7 CONCLUSIONS

In this chapter we have shown that we can attain, and even surpass single-task performance through multi-task networks, provided we execute one task at a time. We have achieved this by introducing Attentive Single-Tasking of Multiple Tasks (ASTMT), a method that allows a network to 'focus' on the task at hand in terms of task-specific feature modulation and adaptation.

In a general vision architecture one can think of task attention as being determined based on the operation currently being performed - e.g.

using object detection to find an object, normal estimation and segmentation to grasp it. Tasks can also be executed in an interleaved manner, with low-level tasks interacting with high-level ones in a bottom-up/top-down cascade [109]. We intend to explore these directions in future research.

6 ADDITIONAL RESEARCH

This chapter presents the abstracts and the respective publications of the research that has been conducted during my PhD research, but are not part of this dissertation. Connections of these works with the contributions of this dissertations have been made in the respective chapters.

6.1 DEEP RETINAL IMAGE UNDERSTANDING

This section presents the abstract of Deep Retinal Image Understanding (DRIU) [140], a unified framework of retinal image analysis that provides both retinal vessel and optic disc segmentation. We make use of deep Convolutional Neural Networks (CNNs), which have proven revolutionary in other fields of computer vision such as object detection and image classification, and we bring their power to the study of eye fundus images. DRIU uses a base network architecture on which two set of specialized layers are trained to solve both the retinal vessel and optic disc segmentation. We present experimental validation, both qualitative and quantitative, in four public datasets for these tasks. In all of them, DRIU presents super-human performance, that is, it shows results more consistent with a gold standard than a second human annotator used as control.

6.2 ONE-SHOT VIDEO OBJECT SEGMENTATION

This section presents the abstract of One-Shot Video Object Segmentation (OSVOS) [25], a method that tackles the task of semi-supervised video object segmentation, *i.e*, the separation of an object from the background in a video, given the mask of the first frame. Our method is based on a fully-convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object of the test sequence

(hence one-shot). Although all frames are processed independently, the results are temporally coherent and stable. We perform experiments on three annotated video segmentation databases, which show that OSVOS is fast and improves the state of the art by a significant margin (79.8% vs 68.0%).

6.3 Video object segmentation without temporal information

Video Object Segmentation, and video processing in general, has been historically dominated by methods that rely on the temporal consistency and redundancy in consecutive video frames. When the temporal smoothness is suddenly broken, such as when an object is occluded, or some frames are missing in a sequence, the result of these methods can deteriorate significantly. This sections presents the abstract of a method [136] that explores the orthogonal approach of processing each frame independently, *i.e*disregarding the temporal information. In particular, it tackles the task of semi-supervised video object segmentation: the separation of an object from the background in a video, given its mask in the first frame. We present Semantic One-Shot Video Object Segmentation (OSVOS^S), based on a fully-convolutional neural network architecture that is able to successively transfer generic semantic information, learned on ImageNet, to the task of foreground segmentation, and finally to learning the appearance of a single annotated object of the test sequence (hence one shot). We show that instance-level semantic information, when combined effectively, can dramatically improve the results of our previous method, OSVOS. We perform experiments on two recent single-object video segmentation databases, which show that OSVOS^S is both the fastest and most accurate method in the state of the art. Experiments on multi-object video segmentation show that OSVOS^S obtains competitive results.

7 DISCUSSION

7.1 SUMMARY OF CONTRIBUTIONS

In this dissertation we have presented four different ways to use low-level features, *i.e* edges and boundaries, for higher level scene understanding tasks. In particular, we showed (a) how to use boundary detection and hierarchical image segmentation to enhance semantic segmentation and object detection, (b) how to use extreme clicks provided by humans for object segmentation, (c) how to establish 3D-2D correspondences for calibration, registration, and 3D reconstruction by detecting points, and (d) how to learn multiple low-level and high-level tasks jointly without compromising performance.

In Chapter 2 we presented Convolutional Oriented Boundaries (COB), a method that uses deeply learned boundaries in multiple scales for hierarchical image segmentation in a single forward pass of a CNN. We showed that learning the orientation of boundaries on top of their strength leads to more accurate region hierarchies, that improve the state of the art in various boundary detection benchmarks (BSDS, PASCAL Context, PASCAL Segmentation, NYUD) as well as in benchmarks for segmented object proposals. We illustrated that snapping semantic segmentation results of popular algorithms to COB regions improves their performance, indicating that boundary detection and semantic segmentation provide complementary information. COB bounding box object proposals also help object detection by replacing popular object proposal algorithms. Our method is also fast compared to the existing region hierarchy methods, by using a sparse matrix representation of the boundaries, running at 1 second per image.

In Chapter 3 we presented Deep Extreme Cut (DEXTR), that turns extreme clicks provided by the user into accurate segmentation masks of objects. We do so by using a heatmap representation of the extreme points, with Gaussians centered on each of them. The heatmap representation along with the cropped RGB image are fed to a dense prediction network to recover the mask of the object of interest, *i.e* the one that the extreme points belong to. We show that DEXTR compares favorably

to the state of the art for class-agnostic instance segmentation, and interactive object segmentation. We illustrate the practical applicability of DEXTR by using it to initialize the masks of semi-supervised video object segmentation algorithms, and by using it for fast and accurate annotation. Performance is evaluated on various benchmarks (PASCAL, COCO, Grabcut, DAVIS 2016/2017).

In Chapter 4 we presented how to provide visual guidance for robot-assisted retinal surgery in a complicated setup that comprises of an accurate co-manipulated robotic arm, a stereo-microscope and two RGB cameras in an uncalibrated stereo setup. We show that by detecting specific keypoints of the robotic arm on the images we can establish 3D (from the accurate robot kinematics) to 2D (from our pixel coordinates) correspondences. Once detection of keypoints is accurate, we are able to collect correspondences just by moving the robot, which allows us to cover regions on demand. We observe that conventional calibration techniques using checkerboards are neither practical nor suitable for the affine cameras of the microscope. With our technique we achieve 100μm accuracy for 3D reconstruction, in real-time.

In Chapter 5 we presented Attentive Single-Tasking of Multiple Tasks (ASTMT) for jointly training a CNN architecture for multiple low-, mid-, and high-level dense prediction tasks. We discuss problems such as network architecture capacity and task interference when jointly training for potentially unrelated or conflicting tasks. We provide two different solutions by (a) building a CNN that is able to perform all tasks but only a single task at a time, and (b) using adversarial training on the gradients of each of the tasks, in order to make them indistinguishable. The benefit from statistically indistinguishable gradients is that no task dominates the other during joint training. We propose different modulation of the shared network depending on the task by dynamically changing the computational graph and using residual adapters and squeeze and excitation as per-task modulators. Experiments on 3 different benchmarks (PASCAL, NYUD, and FSV) show that ASTMT, built on top of a powerful network, such as Deeplab-v3+, is able to reach, or even surpass single-tasking performance when jointly training for 3, 4, or even 5 different tasks.

7.2 DISCUSSION, LIMITATIONS, AND FUTURE RESEARCH

This section is dedicated to discussing limitations of the various contributions presented in this thesis, and trigger potential directions for future research.

7.2.1 *Convolutional Oriented Boundaries*

Even though COB is a powerful tool that has been proven useful for various tasks outside the scope of this thesis, such as enhancing the output of video object segmentation [25, 136], extracting boundaries for semantic segmentation from limited data [145] etc., there are certain drawbacks that should be discussed thoroughly.

- *It is not an end-to-end solution:* The part of the pipeline that uses the predicted contours in multiple scales and turns them into region hierarchies is - while being orders of magnitude faster - very similar to [166], which is hand-crafted. It is composed of various components that are not differentiable (eg. oriented watershed transform) and cannot follow the modern paradigm of end-to-end training with backpropagation.

- *Non-practical for object detection pipelines:* In Section 2.6.3 we showed that bounding box proposals generated by recovering a bounding box from segmented proposals of COB can improve object detection when plugged into the Fast R-CNN [61] pipeline. However, modern ROI-based object detectors [178, 73, 45] have substituted external algorithms that generate object proposals by end-to-end generation of bounding-box proposals, in the same CNN architecture used for detection, achieving much better results than when using COB. In addition, proposal-free approaches for object detection that gain more and more attention [176, 124, 177, 121] can not benefit from COB proposals, since they do not require them. The practical applicability of COB for object detection is thus shadowed by recent, more practical developments.

- *In need of yet another CNN:* The improvements that COB results in for higher-level tasks come with the overhead of including COB into their respective pipelines, *i.e* an additional network for region hierarchy prediction. The overhead is not trivial when discussing

about real-time methods (COB is not a real-time method; it takes 1 second to generate results).

All of the above limit the practical applicability of COB for fast applications. Even though using sparse matrix representations for the boundaries significantly reduced its speed compared to popular region hierarchy methods [166, 209], the computational overhead is still large. In the future, interesting research directions include end-to-end training for region hierarchies which can achieve real-time performance with light-weight networks.

7.2.2 *Deep Extreme Cut*

DEXTR has been successful in terms of impact and triggered further research [3, 155, 238] due to its practical applicability. Here we discuss some limitations of this work:

- *DEXTR is not very interactive:* Although we benchmarked DEXTR against methods for interactive segmentation and achieved state-of-the-art results, in reality our method is not very interactive. DEXTR works with 4, 5, or 6 points at most for interactive object segmentation, without further improvements when adding more points. Its strength lies on the fact that the initial 4 extreme points are very well defined, we can recover a bounding box, and the initial segmentation is very accurate. The ideal interactive object segmentation method should improve on initial predictions based on user input as the number of provided cues increases. Our method does not perform well in cases that the initial prediction is very bad, and the correct outcome mainly relies on interaction with the user.

- *Weaker performance in stuff classes:* DEXTR is primarily designed for foreground classes, for which we obtain a mIoU of 91.5% in PASCAL (Table 3.3). However, when we re-train for stuff classes performance drops to mIoU of 81.75% in PASCAL Context. Even if the results are fairly satisfying, better performance can be achieved by directly applying modern semantic segmentation networks (around 85-89% [233, 33]). We attribute this to the fact that stuff (or background) classes do not have a notion of instances (for example an image with two connected components

of 'sky' class does not have 2 skies). Thus, the task of segmenting background classes given their extreme points requires further considerations.

7.2.3 *Automatic Tool Landmark Detection for Stereo Vision in Robot-Assisted Retinal Surgery*

Although our pipeline for calibration, reconstruction, and registration from detected keypoints works well in open-sky eyes, and is easily adjustable to different types of robotic surgery, in-vivo retinal surgery is a special case which comes with many challenges:

- *Distortions from the lens of the examined eye:* During in-vivo surgery, the setup that needs to be calibrated includes the lens of the examined eye. For this reason using the tool that is inside the eye for calibration is a good idea, but the distortions introduced by the lens need to be modeled as well, which is not done in our work.

- *Strong illumination changes, bleeding:* In our simulations on porcine eyes we did not take into account strong illumination changes and other sources of smooth texture-less areas such as bleeding, which is very common in patients with retinal disorders. These factors play an important role for finding sufficient amount of correspondences for 3D reconstruction.

- *The surgeon's movements are limited by the incision point of the tool:* Even though in theory we acquire correspondences from just moving the tool in order to calibrate, in practice these movements are limited by the incision point of the tool. The co-manipulated robot is specifically designed to be stable at the incision point, thus the possible movements are limited to rotation around it, and to translation along the axis of the tool. This constraint together with the physical constraints of the eye (can not go past the retina or the boundaries of the eye) limit the movements of the robotic tool.

- *Data are difficult to acquire:* In-vivo data of a not medically approved setup are very difficult to acquire. Even after approval, such data are acquired by conducting animal trials that need

coordination of surgeons, hospitals, and engineers. In order to study in-vivo retinal surgery of humans, realistic data need to be acquired. No such data are available yet for robot-assisted retinal surgery, to the best of our knowledge.

- *In need of reliable robot kinematics:* Our assumption for developing our method is that 3D kinematics of the robot are accurate. When this assumption does not hold, for example if the robotic tool bends because the incision point is not correctly calibrated, our entire pipeline collapses. The Achilles heel of our method, and thus a bottleneck in safety, is that it relies too much in accurate robot kinematics, and it will stop working if acquisition of such information is not possible.

Of course, when it comes to real surgery, other sensors are to be used as well (distance sensors on the tool, Optical Coherence Tomography devices, etc.). Nevertheless, our method is the first to tackle 3D reconstruction and registration by using only RGB affine microscopic cameras, and motivates future work in this direction. As future work, it would be very interesting to fuse the information that all these sensors provide. Combining less accurate 3D reconstruction from RGB cameras, with more accurate information from OCT devices (that are however slower and with limited field of view), and with distance sensors that are very fast and accurate (but their information is limited to a single point) is a very challenging topic. Towards 3D semantic understanding of the retina, the approach that we took in this project was combining our two methods: [140] for 2D semantic segmentation of vessels and optic nerve, and [171] for 3D reconstruction. By projecting 2D segmentation on top of 3D reconstruction results, we were able to obtain a 3D map of the retina. Future research could investigate ways to combine the two components with an end-to-end approach. Together with in-vivo data processing, this area of research has many exciting challenges to solve. Solving the task convincingly is the only step towards clinical validation, approvals, and trials for real applications.

7.2.4 *Attentive Single-Tasking of Multiple Tasks*

Interpretations of multi-task learning: Multi-task learning (MTL) has various interpretations across the different contributions in literature.

The different meanings are usually attributed to very close relationships of MTL with the following topics:

- *Domain Adaptation* is the task of bridging the statistics of two or more datasets. Domain adaptation is necessary when we need to use the (usually cheaply acquired) data of the source dataset to train a model that performs well on the (usually expensive to acquire) target dataset. If the statistics of the two datasets are not similar, this is a difficult problem due to domain shifts. In connection to MTL that predicts different outputs given a single input, the goal of domain adaptation is to predict the same output given inputs from different domains. The usage of those two terms is often confusing and subjective [127, 91]. Domain shifts in MTL can arise when using two partially labeled datasets. In connection to Chapter 5, if we want to jointly predict monocular depth and human parts, we need to train from both the NYUD dataset of indoor scenes and PASCAL images that are annotated with the respective tasks. Which brings us to the next bullet point.

- *Catastrophic Forgetting* [94, 116] happens when one task does not recur for long time intervals in a MTL or domain adaptation scenario. For example, after fine-tuning the weights of an ImageNet-pretrained network for a different task, the network loses its ability to perform ImageNet classification without retraining. For MTL forgetting can happen when a task is not executed for a long time, and the weights of the shared network update towards optimizing other tasks. This can happen when training from different partially annotated datasets. In connection to our previous example, forgetting happens when jointly training for human parts and monocular depth by using the 1500 images of PASCAL and the hundreds of thousands images of NYUD. In practical terms, the signal for human part segmentation is lost in the proportionally huge amount of updates for monocular depth prediction. And this brings us to the following problem.

- *Training from additional, imbalanced datasets* is observed for MTL [217, 98] when there is additional, partially annotated data that can be used as additional source of supervision. How to effectively handle the separate data sources is an active topic of research. In fact, it has been shown that tasks can benefit from MTL when

data are not sufficient for a task, but there are available data for a different, closely related task [123, 72].

So how can we ablate performance when training from different imbalanced datasets, that are partially annotated? Are our results improving because of 'universal representations' created by using MTL? Are improvements attributed to using more data from a different data source? On the opposite side, did our results get worse because of task interference and limited network capacity? Or due to catastrophic forgetting and domain shifts because we used an additional large dataset? These are questions that MTL alone is not able to answer, because they result from a composition of all the above topics.

In ASTMT we included many different, potentially unrelated tasks, with a diversity in losses and objectives. However, performance gains and drops are attributed to pure multi-tasking, because we isolated all the other sources of performance changes. For each experiment we used one single dataset (as opposed to experiencing domain shifts and using additional data), and all images of that dataset were annotated with labels from all tasks (as opposed to catastrophic forgetting). Even though the study of MTL isolated from all other topics is very interesting from the point of view of research, in the real world we need systems that effectively handle multiple datasets, forgetting, domain adaptation, and multi-task learning, so that we exploit all possible sources of data to feed data-hungry CNN methods, while being able to ablate the sources of performance change at the same time. Building systems that can handle all of the above is a very interesting line of future research.

Instance-level tasks: Tasks for instance-level recognition such as object detection [61, 178], semantic instance segmentation [73], multi-person pose estimation [27], and dense pose prediction [8] were not included in ASTMT despite the partial availability of their labels. The reason is that the most well-performing architectures for those tasks operate on Region of Interests (ROI) [178]. Combining ROI-based architectures with the fully convolutional architecture used in ASTMT requires compromises (deeper task-specific heads, input/output resolution, treatment of batchnorm layers [83] along with batch size) that would be difficult to ablate. However, these are important tasks that are being tackled more and more with fully convolutional architectures [117, 113, 152, 153, 105]. In future work it would be very interesting how

instance-level and category-level tasks interact with each other [98] with a unified, consistent architecture.

7.3 Open-sourced contributions

Reproducible research and open-sourced repositories are one of the main reasons that the fields of computer vision and machine learning are evolving with high speed. Small details that are not well-explained in a paper, hyper-parameter values and also well-structured code that one can learn a lot from, are available in publicly released repositories. The results obtained in this dissertation are a product of inspiration from such open-sourced research. With a feeling of gratitude to our community, all components of this dissertation, including the additional research conducted during my PhD studies are open-sourced to help future researchers:

- Code, pre-computed results, pre-trained models, and benchmarks for Convolutional Oriented Boundaries (Chapter 2) are publicly available at `http://people.ee.ethz.ch/~cvlsegmentation/cob/`.
- All resources of Deep Extreme Cut (Chapter 3) are available at `http://people.ee.ethz.ch/~cvlsegmentation/dextr/`.
- The dataset of correspondences used for calibration and 3D reconstruction, and to train automatic detection of keypoints (Chapter 4) is available at `http://people.ee.ethz.ch/~kmaninis/keypoints2stereo/`.
- All resources for Attentive Single-Tasking of Multiple Tasks (Chapter 5) are available at `http://people.ee.ethz.ch/~kmaninis/astmt/`.
- All resources for Deep Retinal Image Understanding and One-Shot Video Object Segmentation (Chapter 6) are available in `http://people.ee.ethz.ch/~cvlsegmentation/`.

BIBLIOGRAPHY

[1] D. Acuna, A. Kar, and S. Fidler. „Devil is in the edges: Learning semantic boundaries from noisy annotations." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 10).

[2] D. Acuna, H. Ling, A. Kar, and S. Fidler. „Efficient interactive annotation of segmentation datasets with Polygon-RNN++." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 42).

[3] E. Agustsson, J. R. Uijlings, and V. Ferrari. „Interactive Full Image Segmentation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on pp. 42, 110).

[4] E. Ahmed, S. Cohen, and B. Price. „Semantic object selection." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 41).

[5] K. Ahmed and L. Torresani. „MaskConnect: Connectivity Learning by Gradient Descent." In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 84).

[6] B. Alexe, T. Deselaers, and V. Ferrari. „Measuring the objectness of image windows." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 34 (2012), pp. 2189–2202 (cit. on pp. 33, 34).

[7] M. Allan, S. Ourselin, S. Thompson, D. J. Hawkes, J. Kelly, and D. Stoyanov. „Toward Detection and Localization of Instruments in Minimally Invasive Surgery." In: *IEEE Transactions on Biomedical Engineering* (2013) (cit. on p. 61).

[8] R. Alp Güler, N. Neverova, and I. Kokkinos. „Densepose: Dense human pose estimation in the wild." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 114).

[9] M. Ammi, V. Fremont, and A. Ferreira. „Automatic Camera-Based Microscope Calibration for a Telemicromanipulation System Using a Virtual Pattern." In: *IEEE Transactions on Robotics* 25.1 (2009), pp. 184–191 (cit. on p. 63).

[10] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. „Bottom-up and top-down attention for image captioning and visual question answering." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 84).

[11] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. „Contour detection and hierarchical image segmentation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 33.5 (2011), pp. 898–916 (cit. on pp. 6–11, 14, 17, 20, 21, 23, 30, 89).

[12] X. Bai and G. Sapiro. „Geodesic matting: A framework for fast interactive image and video segmentation and matting." In: *International Journal of Computer Vision (IJCV)* 82.2 (2009), pp. 113–132 (cit. on p. 57).

[13] A. Bansal, X. Chen, B. Russell, A. Gupta, and D. Ramanan. „Pixelnet: Representation of the pixels, by the pixels, and for the pixels." In: *arXiv:1702.06506* (2017) (cit. on pp. 90, 91).

[14] H. Bay, T. Tuytelaars, and L. Van Gool. „SURF: Speeded up robust features." In: *European conference on computer vision (ECCV)*. 2006, pp. 404–417 (cit. on p. 1).

[15] A. Bearman, O. Russakovsky, V. Ferrari, and L. Fei-Fei. „What's the Point: Semantic Segmentation with Point Supervision." In: *European Conference on Computer Vision (ECCV)* (2016) (cit. on p. 41).

[16] R. Benenson, S. Popov, and V. Ferrari. „Large-scale interactive object segmentation with human annotators." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 42).

[17] G. Bertasius, J. Shi, and L. Torresani. „DeepEdge: A Multi-Scale Bifurcated Deep Network for Top-Down Contour Detection." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on pp. 5, 8).

[18] G. Bertasius, J. Shi, and L. Torresani. „High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 5, 8, 34, 35).

[19] G. Bertasius, J. Shi, and L. Torresani. „Semantic Segmentation with Boundary Neural Fields.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 8, 34, 35).

[20] D. Bouget, M. Allan, D. Stoyanov, and P. Jannin. „Vision-based and marker-less surgical tool detection and tracking: a review of the literature.“ In: *Medical Image Analysis* (2017) (cit. on p. 61).

[21] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan. „Domain separation networks.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2016 (cit. on p. 83).

[22] Y. Y. Boykov and M.-P. Jolly. „Interactive graph cuts for optimal boundary & region segmentation of objects in ND images.“ In: *International Conference on Computer Vision (ICCV)*. 2001 (cit. on pp. 43, 57).

[23] L. Breiman. „Random forests.“ In: *Machine Learning* 45.1 (2001), pp. 5–32 (cit. on p. 1).

[24] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. „A naturalistic open source movie for optical flow evaluation.“ In: *European Conference on Computer Vision (ECCV)*. 2012 (cit. on p. 2).

[25] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. „One-Shot Video Object Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. ix, 8, 10, 55, 105, 109).

[26] J. Canny. „A computational approach to edge detection.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 6 (1986), pp. 679–698 (cit. on pp. 1, 8).

[27] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. „Realtime multi-person 2d pose estimation using part affinity fields.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 114).

[28] J. Carreira and C. Sminchisescu. „CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 34.7 (2012), pp. 1312–1328 (cit. on pp. 32, 33).

[29] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. „Annotating object instances with a polygon-rnn.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 42).

[30] T. Chanwimaluang and G. Fan. „Affine Camera for 3-D Retinal Surface Reconstruction.“ In: *International Symposium on Visual Computing (ISVC)*. 2006 (cit. on pp. 60, 62, 63, 67, 69).

[31] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. „Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2017) (cit. on pp. 44, 48, 49, 54, 63, 90, 101).

[32] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. L. Yuille. „Attention to scale: Scale-aware semantic image segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 84).

[33] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. „Encoder-decoder with atrous separable convolution for semantic image segmentation.“ In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on pp. 39, 89–91, 98, 102, 110).

[34] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. „Detect what you can: Detecting and representing objects using holistic models and body parts.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 90).

[35] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. „GradNorm: Gradient Normalization for Adaptive Loss Balancing in Deep Multitask Networks.“ In: *arXiv:1711.02257* (2018) (cit. on pp. 83, 101).

[36] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu. „Global contrast based salient region detection.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 37.3 (2015), pp. 569–582 (cit. on p. 90).

[37] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. „BING: Binarized Normed Gradients for Objectness Estimation at 300fps.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 33, 34).

[38] A. Chhatkuli, A. Bartoli, A. Malti, and T. Collins. „Live image parsing in uterine laparoscopy.“ In: *IEEE International Symposium on Biomedical Imaging (ISBI)*. 2014 (cit. on p. 61).

[39] A. Chhatkuli, D. Pizarro, A. Bartoli, and T. Collins. „A Stable Analytical Framework for Isometric Shape-from-Template by Surface Integration.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.5 (2017), pp. 833–850 (cit. on p. 70).

[40] F. Chollet. „Xception: Deep learning with depthwise separable convolutions.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 2).

[41] D. Comaniciu and P. Meer. „Mean shift: a robust approach toward feature space analysis.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 24.5 (2002), pp. 603–619 (cit. on p. 23).

[42] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. „The Cityscapes Dataset for Semantic Urban Scene Understanding.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 2).

[43] C. Cortes and V. Vapnik. „Support-vector networks.“ In: *Machine Learning* 20.3 (1995), pp. 273–297 (cit. on p. 1).

[44] J. Dai, K. He, and J. Sun. „Boxsup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 40, 41).

[45] J. Dai, Y. Li, K. He, and J. Sun. „R-FCN: Object Detection via Region-based Fully Convolutional Networks.“ In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. 38, 109).

[46] N. Dalal and B. Triggs. „Histograms of oriented gradients for human detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2005 (cit. on p. 1).

[47] P. Dollar, Z. Tu, and S. Belongie. „Supervised learning of edges and object boundaries.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2006 (cit. on p. 8).

[48] P. Dollár and C. L. Zitnick. „Fast edge detection using structured forests.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 37.8 (2015), pp. 1558–1570 (cit. on pp. 8, 10, 14, 19–22, 28–30).

[49] H. Drucker and Y. Le Cun. „Double backpropagation increasing generalization performance.“ In: *International Joint Conference on Neural Networks (IJCNN)*. 1991 (cit. on pp. 82, 87, 88).

[50] V. Dumoulin, J. Shlens, and M. Kudlur. „A learned representation for artistic style.“ In: *International Conference on Learning Representations (ICLR)*. 2017 (cit. on p. 84).

[51] N. Dvornik, K. Shmelkov, J. Mairal, and C. Schmid. „BlitzNet: A real-time deep network for scene understanding.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on p. 82).

[52] D. Eigen and R. Fergus. „Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 79, 82, 96, 103).

[53] L. Esteveny, L. Schoevaerdts, A. Gijbels, D. Reynaerts, and E. V. Poorten. „Experimental validation of instrument insertion precision in robot-assisted eye-surgery.“ In: *CRAS*. 2015 (cit. on p. 76).

[54] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results* (cit. on pp. 2, 6, 34, 47, 89, 101).

[55] T. Evgeniou and M. Pontil. „Regularized multi–task learning.“ In: *KDD*. 2004 (cit. on p. 86).

[56] P. F. Felzenszwalb and D. P. Huttenlocher. „Efficient graph-based image segmentation.“ In: *International Journal of Computer Vision (IJCV)* 59 (2004) (cit. on p. 23).

[57] J. Fu, H. Zheng, and T. Mei. „Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 84).

[58] Y. Ganin and V. Lempitsky. „N^4-Fields: Neural Network Nearest Neighbor Fields for Image Transforms.“ In: *Asian Conference on Computer Vision (ACCV)*. 2014 (cit. on pp. 5, 8).

[59] Y. Ganin and V. Lempitsky. „Unsupervised domain adaptation by backpropagation.“ In: *International Conference on Machine Learning (ICML)*. 2015 (cit. on pp. 84, 87, 88).

[60] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky. „Domain-adversarial training of neural networks.“ In: *Journal of Machine Learning Research (JMLR)* (2016) (cit. on p. 88).

[61] R. Girshick. „Fast R-CNN.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 31, 35–37, 82, 109, 114).

[62] R. Girshick, J. Donahue, T. Darrell, and J. Malik. „Rich feature hierarchies for accurate object detection and semantic segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 31, 36).

[63] L. Grady. „Random walks for image segmentation.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 28.11 (2006), pp. 1768–1783 (cit. on p. 57).

[64] M. Guo, A. Haque, D.-A. Huang, S. Yeung, and L. Fei-Fei. „Dynamic Task Prioritization for Multitask Learning.“ In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 83).

[65] S. Gupta, P. Arbeláez, and J. Malik. „Perceptual organization and recognition of indoor scenes from RGB-D images.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013 (cit. on pp. 10, 29).

[66] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. „Learning rich features from RGB-D images for object detection and segmentation.“ In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on pp. 10, 28, 29).

[67] S. Hallman and C. C. Fowlkes. „Oriented Edge Forests for Boundary Detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 9).

[68] J. Hao Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng. „Regional Interactive Image Segmentation Networks.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on pp. 42, 49, 57).

[69] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik. „Semantic contours from inverse detectors.“ In: *International Conference on Computer Vision (ICCV)*. 2011 (cit. on pp. 6, 10, 25, 34–36, 47, 89).

[70] C. G. Harris and M. Stephens. „A combined corner and edge detector.“ In: *Alvey vision conference*. Vol. 15. 50. 1988, pp. 10–5244 (cit. on p. 1).

[71] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521623049, 2003 (cit. on pp. 61–63, 68).

[72] K. He, R. Girshick, and P. Dollár. „Rethinking imagenet pre-training.“ In: *International Conference on Computer Vision (ICCV)*. 2019 (cit. on p. 114).

[73] K. He, G. Gkioxari, P. Dollár, and R. Girshick. „Mask R-CNN.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on pp. 39, 44, 48, 79, 81–83, 109, 114).

[74] K. He, X. Zhang, S. Ren, and J. Sun. „Deep Residual Learning for Image Recognition.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 2, 5, 7, 9, 13, 19, 44, 62, 64, 65, 72).

[75] R. Horaud, S. Christy, and R. Mohr. „Euclidean reconstruction and affine camera calibration using controlled robot motions.“ In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 1997 (cit. on pp. 62, 63, 67).

[76] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. „What makes for effective detection proposals?“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.4 (2016), pp. 814–830 (cit. on p. 34).

[77] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. „What makes for effective detection proposals?“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.4 (2016), pp. 814–830 (cit. on p. 42).

[78] J. Hu, L. Shen, S. Albanie, G. Sun, and A. Vedaldi. „Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks.“ In: *arXiv:1810.12348* (2018) (cit. on p. 84).

[79] J. Hu, L. Shen, and G. Sun. „Squeeze-and-excitation networks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on pp. 84, 86, 96, 98).

[80] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. „Densely connected convolutional networks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 2).

[81] A. Humayun, F. Li, and J. M. Rehg. „The Middle Child Problem: Revisiting Parametric Min-cut and Seeds for Object Proposals.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 31, 32).

[82] A. Humayun, F. Li, and J. M. Rehg. „RIGOR: Recycling Inference in Graph Cuts for generating Object Regions.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 31, 32, 34).

[83] S. Ioffe and C. Szegedy. „Batch normalization: Accelerating deep network training by reducing internal covariate shift.“ In: *arXiv:1502.03167* (2015) (cit. on p. 114).

[84] P. Isola, D. Zoran, D. Krishnan, and E. H. Adelson. „Crisp boundary detection using pointwise mutual information.“ In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on p. 10).

[85] S. D. Jain and K. Grauman. „Click Carving: Segmenting Objects in Video with Point Clicks.“ In: *Conference on Human Computation and Crowdsourcing (HCOMP)*. 2016 (cit. on p. 42).

[86] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. „Caffe: Convolutional Architecture for Fast Feature Embedding.“ In: *arXiv:1408.5093* (2014) (cit. on p. 18).

[87] B. Jin, M. V. Ortiz Segovia, and S. Susstrunk. „Webly Supervised Semantic Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 41).

[88] A. Kendall, Y. Gal, and R. Cipolla. „Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 83).

[89] A. Khoreva, R. Benenson, J. Hosang, M. Hein, and B. Schiele. „Simple Does It: Weakly Supervised Instance and Semantic Segmentation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. 40, 41).

[90] A. Khoreva, R. Benenson, M. Omran, M. Hein, and B. Schiele. „Weakly Supervised Object Boundaries." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 9, 10, 24, 25, 27, 35).

[91] E. Kim, C. Ahn, P. H. Torr, and S. Oh. „Deep Virtual Networks for Memory Efficient Inference of Multiple Tasks." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 113).

[92] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. „Panoptic segmentation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 42).

[93] A. Kirillov, E. Levinkov, B. Andres, B. Savchynskyy, and C. Rother. „Instancecut: from edges to instances with multicut." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 10).

[94] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. „Overcoming catastrophic forgetting in neural networks." In: *national academy of sciences* 114.13 (2017), pp. 3521–3526 (cit. on pp. 83, 113).

[95] J. Kittler. „On the accuracy of the Sobel edge detector." In: *Image and Vision Computing* 1.1 (1983), pp. 37–42 (cit. on p. 8).

[96] I. Kokkinos. „Boundary detection using F-measure-, filter-and feature-(f3) boost." In: *European Conference on Computer Vision (ECCV)*. 2010 (cit. on pp. 8, 9).

[97] I. Kokkinos. „Pushing the boundaries of boundary detection using deep learning." In: *International Conference on Learning Representations (ICLR)*. 2016 (cit. on pp. 5, 7–9, 35, 90, 102).

[98] I. Kokkinos. „UberNet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. 79, 83, 90, 91, 98, 113, 115).

[99] S. Konishi, A. L. Yuille, J. M. Coughlan, and S. C. Zhu. „Statistical edge detection: Learning and evaluating edge cues.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 25.1 (2003), pp. 57–74 (cit. on p. 8).

[100] P. Krähenbühl. „Free supervision from video games.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on pp. 89, 96, 97, 101).

[101] P. Krähenbühl and V. Koltun. „Geodesic Object Proposals.“ In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on pp. 31, 32).

[102] P. Krähenbühl and V. Koltun. „Learning to propose objects.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on pp. 31, 32, 34).

[103] A. Krizhevsky, I. Sutskever, and G. E. Hinton. „ImageNet Classification with Deep Convolutional Neural Networks.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2012 (cit. on pp. 1, 5, 9, 62).

[104] I. Laina, N. Rieke, C. Rupprecht, J. P. Vizcaino, A. Eslami, F. Tombari, and N. Navab. „Concurrent Segmentation and Localization for Tracking of Surgical Instruments.“ In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2017 (cit. on pp. 63, 72).

[105] H. Law and J. Deng. „Cornernet: Detecting objects as paired keypoints.“ In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 114).

[106] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. „Gradient-based learning applied to document recognition.“ In: *IEEE* 86.11 (1998), pp. 2278–2324 (cit. on p. 1).

[107] C. Lee, Y.-F. Wang, D. R. Uecker, and Y. Wang. „Image analysis for automated tracking in robot-assisted endoscopic surgery.“ In: *International Conference on Pattern Recognition (ICPR)*. 1994 (cit. on p. 61).

[108] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. „Deeply-supervised nets.“ In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2015 (cit. on pp. 13, 65).

[109] T. S. Lee and D. Mumford. „Hierarchical Bayesian inference in the visual cortex.“ In: *Journal of the Optical Society of America (JOSA)* 20.7 (2003), pp. 1434–1448 (cit. on p. 104).

[110] V. Lempitsky, P. Kohli, C. Rother, and T. Sharp. „Image segmentation with a bounding box prior.“ In: *International Conference on Computer Vision (ICCV)*. 2009, pp. 277–284 (cit. on pp. 43, 52).

[111] V. Lepetit, F. Moreno-Noguer, and P. Fua. „EPnP: An Accurate O(n) Solution to the PnP Problem.“ In: *International Journal of Computer Vision (IJCV)* 81.2 (2009) (cit. on p. 70).

[112] X. Li, Y. Qi, Z. Wang, K. Chen, Z. Liu, J. Shi, P. Luo, C. C. Loy, and X. Tang. „Video Object Segmentation with Re-identification.“ In: *The 2017 DAVIS Challenge on Video Object Segmentation - CVPR Workshops* (2017) (cit. on p. 49).

[113] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. „Fully convolutional instance-aware semantic segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 114).

[114] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille. „The secrets of salient object segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 90).

[115] Y. Li, M. Paluri, J. M. Rehg, and P. Dollár. „Unsupervised learning of edges.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 9).

[116] Z. Li and D. Hoiem. „Learning without forgetting.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40.12 (2017), pp. 2935–2947 (cit. on p. 113).

[117] X. Liang, L. Lin, Y. Wei, X. Shen, J. Yang, and S. Yan. „Proposal-free network for instance-level object segmentation.“ In: *IEEE transactions on pattern analysis and machine intelligence (TPAMI)* 40.12 (2017), pp. 2978–2991 (cit. on p. 114).

[118] J. J. Lim, C. L. Zitnick, and P. Dollár. „Sketch tokens: A learned mid-level representation for contour and object detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013 (cit. on pp. 8, 9).

[119] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. „Scribblesup: Scribble-supervised convolutional networks for semantic segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 40, 41).

[120] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. „Feature pyramid networks for object detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 63).

[121] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. „Focal loss for dense object detection.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on p. 109).

[122] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. „Microsoft COCO: Common objects in context.“ In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on pp. 2, 7, 34, 45, 47, 55).

[123] P. Liu, X. Qiu, and X. Huang. „Adversarial multi-task learning for text classification.“ In: *Association for Computational Linguistics (ACL)*. 2017 (cit. on pp. 83, 84, 88, 114).

[124] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. „SSD: Single Shot MultiBox Detector.“ In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. 38, 109).

[125] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai. „Richer convolutional features for edge detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 10).

[126] J. Long, E. Shelhamer, and T. Darrell. „Fully Convolutional Networks for Semantic Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 44).

[127] M. Long, Z. Cao, J. Wang, and S. Y. Philip. „Learning multiple tasks with multilinear relationship networks.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2017 (cit. on p. 113).

[128] D. Lopez-Paz and M. Ranzato. „Gradient episodic memory for continual learning.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2017 (cit. on p. 83).

[129] D. G. Lowe. „Distinctive image features from scale-invariant keypoints.“ In: *International journal of computer vision (IJCV)* 60.2 (2004), pp. 91–110 (cit. on p. 1).

[130] J. Lu, J. Yang, D. Batra, and D. Parikh. „Hierarchical question-image co-attention for visual question answering.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2016 (cit. on p. 84).

[131] L. v. d. Maaten and G. Hinton. „Visualizing data using t-SNE.“ In: *Journal of Machine Learning Research (JMLR)* 9.Nov (2008), pp. 2579–2605 (cit. on p. 96).

[132] S. Mahadevan, P. Voigtlaender, and B. Leibe. „Iteratively trained interactive segmentation.“ In: *British Machine Vision Conference (BMVC)*. 2018 (cit. on p. 42).

[133] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, and L. van der Maaten. „Exploring the limits of weakly supervised pretraining.“ In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 2).

[134] M. Maire, T. Narihira, and S. X. Yu. „Affinity CNN: Learning pixel-centric pairwise relations for figure/ground embedding.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 10).

[135] S. Manén, M. Guillaumin, and L. Van Gool. „Prime Object Proposals with Randomized Prim's Algorithm.“ In: *International Conference on Computer Vision (ICCV)*. 2013 (cit. on pp. 33, 34).

[136] K.-K. Maninis, S. Caelles, Y. Chen, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. V. Gool. „Video Object Segmentation Without Temporal Information.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2019) (cit. on pp. ix, 8, 10, 49, 106, 109).

[137] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. V. Gool. „Deep Extreme Cut: From Extreme Points to Object Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. ix).

[138] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. „Convolutional Oriented Boundaries.“ In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. ix, 10).

[139] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. „Convolutional Oriented Boundaries: From Image Segmentation to High-Level Tasks.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 40.4 (2018), pp. 819–833 (cit. on pp. ix, 10, 45, 49, 102).

[140] K.-K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool. „Deep Retinal Image Understanding.“ In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2016 (cit. on pp. ix, 7, 60, 105, 112).

[141] K.-K. Maninis, I. Radosavovic, and I. Kokkinos. „Attentive Single-Tasking of Multiple Tasks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. ix).

[142] D. Marr and E. Hildreth. „Theory of edge detection.“ In: *Proc. Royal Soc. of London* 207.1167 (1980), pp. 187–217 (cit. on p. 8).

[143] D. Martin, C. Fowlkes, and J. Malik. „Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 26.5 (2004), pp. 530–549 (cit. on pp. 8, 17, 19, 22, 23, 27, 28, 89).

[144] D. Martin, C. Fowlkes, D. Tal, and J. Malik. „A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.“ In: *International Conference on Computer Vision (ICCV)*. 2001 (cit. on pp. 2, 6, 21, 89).

[145] A. Milan, T. Pham, K. Vijay, D. Morrison, A. W. Tow, L. Liu, J. Erskine, R. Grinover, A. Gurman, T. Hunn, et al. „Semantic segmentation from limited training data.“ In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018 (cit. on p. 109).

[146] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. „Cross-stitch networks for multi-task learning.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 84).

[147] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. „The Role of Context for Object Detection and Semantic Segmentation in the Wild.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 7, 17, 21, 53).

[148] P. K. Mudrakarta, M. Sandler, A. Zhmoginov, and A. Howard. „K For The Price Of 1: Parameter Efficient Multi-task And Transfer Learning.“ In: *International Conference on Learning Representations (ICLR)*. 2019 (cit. on p. 84).

[149] C. Murdock, Z. Li, H. Zhou, and T. Duerig. „Blockout: Dynamic model selection for hierarchical deep networks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 84).

[150] L. Najman and M. Schmitt. „Geodesic saliency of watershed contours and hierarchical segmentation.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 18.12 (1996), pp. 1163–1173 (cit. on p. 10).

[151] P. K. Nathan Silberman Derek Hoiem and R. Fergus. „Indoor Segmentation and Support Inference from RGBD Images.“ In: *European Conference on Computer Vision (ECCV)*. 2012 (cit. on pp. 26, 89, 90, 96, 97, 101).

[152] D. Neven, B. D. Brabandere, M. Proesmans, and L. V. Gool. „Instance Segmentation by Jointly Optimizing Spatial Embeddings and Clustering Bandwidth.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 114).

[153] A. Newell, Z. Huang, and J. Deng. „Associative embedding: End-to-end learning for joint detection and grouping.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2017 (cit. on p. 114).

[154] A. Newell, K. Yang, and J. Deng. „Stacked hourglass networks for human pose estimation.“ In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. 63, 65, 71).

[155] S. W. Oh, J.-Y. Lee, N. Xu, and S. J. Kim. „Video object segmentation using space-time memory networks.“ In: *International Conference on Computer Vision (ICCV)*. 2019 (cit. on pp. 42, 46, 110).

[156] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. „Extreme clicking for efficient object annotation.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on pp. 40, 41, 43, 45, 47, 49, 51, 52).

[157] D. P. Papadopoulos, J. R. Uijlings, F. Keller, and V. Ferrari. „Training object class detectors with click supervision.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 41).

[158] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. „Automatic differentiation in PyTorch.“ In: *Advances in Neural Information Processing Systems - Workshop (NIPS-W)*. 2017 (cit. on p. 103).

[159] D. Pathak, E. Shelhamer, J. Long, and T. Darrell. „Fully convolutional multi-class multiple instance learning.“ In: *International Conference on Learning Representations (ICLR)*. 2015 (cit. on p. 41).

[160] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. „6-DoF Object Pose from Semantic Keypoints.“ In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017 (cit. on p. 63).

[161] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis. „Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 63).

[162] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung. „A Benchmark Dataset and Evaluation Methodology for Video Object Segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 47, 55).

[163] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville. „FiLM: Visual reasoning with a general conditioning layer.“ In: *arXiv:1709.07871* (2017) (cit. on pp. 82, 84).

[164] P. O. Pinheiro, R. Collobert, and P. Dollár. „Learning to segment object candidates.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2015 (cit. on pp. 31–34, 42).

[165] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. „Learning to refine object segments.“ In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on pp. 31–34, 42, 51, 52).

[166] J. Pont-Tuset, P. Arbeláez, J. Barron, F. Marques, and J. Malik. „Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 39.1 (2017), pp. 128–140 (cit. on pp. 7, 9–11, 14, 16, 19–23, 27, 28, 30–34, 42, 109, 110).

[167] J. Pont-Tuset and F. Marques. „Supervised Evaluation of Image Segmentation and Object Proposal Techniques." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.7 (2016), pp. 1465–1478 (cit. on pp. 17, 19, 22, 23, 27, 28, 89).

[168] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool. „The 2017 DAVIS challenge on video object segmentation." In: *arXiv:1704.00675* (2017) (cit. on pp. 47, 55).

[169] J. Pont-Tuset and L. Van Gool. „Boosting Object Proposals: From Pascal to COCO." In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on p. 34).

[170] J. M. Prewitt. „Object enhancement and extraction." In: *Picture processing and Psychopictorics* 10.1 (1970), pp. 15–19 (cit. on p. 8).

[171] T. Probst, K.-K. Maninis, A. Chhatkuli, M. Ourak, E. V. Poorten, and L. V. Gool. „Automatic Tool Landmark Detection for Stereo Vision in Robot-Assisted Retinal Surgery." In: *IEEE Robotics and Automation Letters (RA-L)* 3.1 (2018), pp. 612–619 (cit. on pp. ix, 112).

[172] A. Ranjan, V. Jampani, K. Kim, D. Sun, J. Wulff, and M. J. Black. „Adversarial Collaboration: Joint Unsupervised Learning of Depth, Camera Motion, Optical Flow and Motion Segmentation." In: *arXiv:1805.09806* (2018) (cit. on p. 82).

[173] P. Rantalankila, J. Kannala, and E. Rahtu. „Generating object segmentation proposals using global and local search." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 31–34).

[174] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. „Efficient parametrization of multi-domain deep neural networks." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on pp. 82, 84).

[175] S.-A. Rebuffi, H. Bilen, and A. Vedaldi. „Learning multiple visual domains with residual adapters.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2017 (cit. on p. 84).

[176] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. „You only look once: Unified, real-time object detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 38, 109).

[177] J. Redmon and A. Farhadi. „YOLO9000: better, faster, stronger.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 109).

[178] S. Ren, K. He, R. Girshick, and J. Sun. „Faster R-CNN: Towards real-time object detection with region proposal networks.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2015 (cit. on pp. 31, 33, 34, 36, 38, 63, 82, 109, 114).

[179] X. Ren. „Multi-scale improves boundary detection in natural images.“ In: *European Conference on Computer Vision (ECCV)* (2008) (cit. on p. 9).

[180] X. Ren and L. Bo. „Discriminatively Trained Sparse Code Gradients for Contour Detection.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2012 (cit. on p. 8).

[181] Z. Ren and G. Shakhnarovich. „Image Segmentation by Cascaded Region Agglomeration.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2013 (cit. on p. 23).

[182] Z. Ren and Y. J. Lee. „Cross-domain self-supervised multi-task feature learning using synthetic imagery.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 83).

[183] N. Rieke, D. J. Tan, M. Alsheakhali, F. Tombari, C. A. di San Filippo, V. Belagiannis, A. Eslami, and N. Navab. „Surgical Tool Tracking and Pose Estimation in Retinal Microsurgery.“ In: *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI)*. 2015 (cit. on pp. 60, 61).

[184] L. G. Roberts. „Machine perception of three-dimensional solids.“ PhD thesis. MIT, 1963 (cit. on p. 8).

[185] A. Rosenfeld, M. Biparva, and J. K. Tsotsos. „Priming Neural Networks.“ In: *arXiv:1711.05918* (2018) (cit. on p. 84).

[186] A. Rosenfeld and J. K. Tsotsos. „Incremental learning through deep adaptation.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2018) (cit. on p. 83).

[187] C. Rother, V. Kolmogorov, and A. Blake. „Grabcut: Interactive foreground extraction using iterated graph cuts.“ In: *ACM Transactions on Graphics (TOG)*. 2004 (cit. on pp. 42, 43, 47, 52).

[188] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. „ImageNet Large Scale Visual Recognition Challenge.“ In: *International Journal of Computer Vision (IJCV)* (2015) (cit. on pp. 2, 5, 13, 39, 53, 89).

[189] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. „Progressive neural networks.“ In: *arXiv:1606.04671* (2016) (cit. on p. 83).

[190] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen. „MobileNetV2: Inverted Residuals and Linear Bottlenecks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 99).

[191] S. Saxena and J. Verbeek. „Convolutional neural fabrics.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2016 (cit. on p. 84).

[192] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, et al. „Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization.“ In: *International Conference on Computer Vision (ICCV)*. 2017 (cit. on p. 84).

[193] O. Sener and V. Koltun. „Multi-Task Learning as Multi-Objective Optimization.“ In: *Advances in Neural Information Processing Systems (NIPS)*. 2018 (cit. on pp. 83, 101, 102).

[194] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. „Overfeat: Integrated recognition, localization and detection using convolutional networks.“ In: *International Conference on Learning Representations (ICLR)*. 2014 (cit. on p. 79).

[195] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang. „DeepContour: A Deep Convolutional Feature Learned by Positive-sharing Loss for Contour Detection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on pp. 5, 8).

[196] J. Shi and J. Malik. „Normalized cuts and image segmentation.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 22.8 (2000) (cit. on pp. 10, 23).

[197] A. Shrivastava, A. Gupta, and R. Girshick. „Training region-based object detectors with online hard example mining.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on p. 56).

[198] K. Simonyan, A. Vedaldi, and A. Zisserman. „Deep inside convolutional networks: Visualising image classification models and saliency maps.“ In: *arXiv:1312.6034* (2013) (cit. on p. 84).

[199] K. Simonyan and A. Zisserman. „Very deep convolutional networks for large-scale image recognition.“ In: *International Conference on Learning Representations (ICLR)*. 2015 (cit. on pp. 2, 5, 9, 13, 19, 37, 62, 72, 98).

[200] A. Sinha, Z. Chen, V. Badrinarayanan, and A. Rabinovich. „Gradient Adversarial Training of Neural Networks.“ In: *arXiv:1806.08028* (2018) (cit. on pp. 83, 88).

[201] H. Su, J. Deng, and L. Fei-Fei. „Crowdsourcing annotations for visual object detection.“ In: *International Conference on Artificial Intelligence and Statistics Workshops (AAAI-W)*. 2012 (cit. on p. 43).

[202] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. „Going deeper with convolutions.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on pp. 2, 5, 9, 13).

[203] R. Sznitman, R. Richa, R. H. Taylor, B. Jedynak, and G. D. Hager. „Unified Detection and Tracking of Instruments during Retinal Microsurgery.“ In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 35.5 (2013), pp. 1263–1273 (cit. on pp. 60, 61).

[204] M. Tan and Q. V. Le. „EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.“ In: *International Conference on Machine Learning (ICML)*. 2019 (cit. on p. 2).

[205] M. Tang, I. Ben Ayed, D. Marin, and Y. Boykov. „Secrets of GrabCut and Kernel K-means.“ In: *International Conference on Computer Vision (ICCV)*. 2015, pp. 1555–1563 (cit. on p. 52).

[206] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov. „Grabcut in one cut." In: *International Conference on Computer Vision (ICCV)*. 2013, pp. 1769–1776 (cit. on p. 52).

[207] C. Tomasi and T. Kanade. „Shape and motion from image streams under orthography: a factorization method." In: *International Journal of Computer Vision (IJCV)* 9.2 (1992), pp. 137–154 (cit. on pp. 62, 63).

[208] A. Toshev and C. Szegedy. „Deeppose: Human pose estimation via deep neural networks." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on p. 63).

[209] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. „Selective Search for Object Recognition." In: *International Journal of Computer Vision (IJCV)* 104.2 (2013), pp. 154–171 (cit. on pp. 31–35, 37, 110).

[210] J. Uijlings and V. Ferrari. „Situational Object Boundary Detection." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015 (cit. on p. 25).

[211] G. Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990 (cit. on p. 70).

[212] P. Wang and A. Yuille. „Doc: Deep occlusion estimation from a single image." In: *European Conference on Computer Vision (ECCV)*. 2016 (cit. on p. 10).

[213] Z. Wang, D. Acuna, H. Ling, A. Kar, and S. Fidler. „Object instance annotation with deep extreme level set evolution." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on p. 42).

[214] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid. „DeepFlow: Large displacement optical flow with deep matching." In: *International Conference on Computer Vision (ICCV)*. 2013 (cit. on pp. 69, 76).

[215] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. „CBAM: Convolutional block attention module." In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 84).

[216] J. Wu, Y. Zhao, J.-Y. Zhu, S. Luo, and Z. Tu. „MILcut: A sweeping line multiple instance learning paradigm for interactive image segmentation.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2014 (cit. on pp. 43, 52).

[217] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. „Unified Perceptual Parsing for Scene Understanding.“ In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on pp. 83, 113).

[218] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He. „Aggregated residual transformations for deep neural networks.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 2).

[219] S. Xie and Z. Tu. „Holistically-Nested Edge Detection.“ In: *International Conference on Computer Vision (ICCV)*. 2015 (cit. on pp. 5, 8, 9, 11, 13, 18, 19, 21–25, 27–30).

[220] S. Xie and Z. Tu. „Holistically-Nested Edge Detection.“ In: *International Journal of Computer Vision (IJCV)* (2017), pp. 1–16 (cit. on pp. 9, 44, 45, 49, 102).

[221] D. Xu, W. Ouyang, X. Wang, and N. Sebe. „PAD-Net: Multi-Tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on pp. 82, 90, 96).

[222] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. „Show, attend and tell: Neural image caption generation with visual attention.“ In: *International Conference on Machine Learning (ICML)*. 2015 (cit. on p. 84).

[223] N. Xu, B. Price, S. Cohen, J. Yang, and T. Huang. „Deep GrabCut for Object Selection.“ In: *British Machine Vision Conference (BMVC)*. 2017 (cit. on pp. 42, 46, 49, 52).

[224] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang. „Deep interactive object selection.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 42, 46, 49, 57).

[225] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang. „Object Contour Detection with a Fully Convolutional Encoder-Decoder Network.“ In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 (cit. on pp. 9, 10, 22–25, 27).

[226] L. Yang, Y. Wang, X. Xiong, J. Yang, and A. K. Katsaggelos. „Efficient video object segmentation via network modulation." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 84).

[227] F. Yu and V. Koltun. „Multi-scale context aggregation by dilated convolutions." In: *International Conference on Learning Representations (ICLR)*. 2016 (cit. on pp. 31, 35, 36).

[228] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam. „Casenet: Deep category-aware semantic edge detection." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on p. 10).

[229] S. Zagoruyko and N. Komodakis. „Wide residual networks." In: *British Machine Vision Conference (BMVC)*. 2016 (cit. on p. 2).

[230] A. R. Zamir, A. Sax, W. Shen, L. Guibas, J. Malik, and S. Savarese. „Taskonomy: Disentangling Task Transfer Learning." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018 (cit. on p. 83).

[231] M. D. Zeiler and R. Fergus. „Visualizing and understanding convolutional networks." In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on p. 84).

[232] Z. Zhang. „A Flexible New Technique for Camera Calibration." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 22.11 (2000), pp. 1330–1334 (cit. on pp. 62, 63, 66).

[233] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. „Pyramid scene parsing network." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 (cit. on pp. 31, 35, 36, 39, 44, 49, 54, 63, 110).

[234] Q. Zhao. „Segmenting Natural Images with the Least Effort as Humans." In: *British Machine Vision Conference (BMVC)*. 2015 (cit. on pp. 21–23, 27).

[235] X. Zhao, H. Li, X. Shen, X. Liang, and Y. Wu. „A Modulation Module for Multi-task Learning with Applications in Image Retrieval." In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 84).

[236] M. Zhou, M. Hamad, J. Weiss, A. Eslami, K. Huang, M. Maier, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasseri. „Towards Robotic Eye Surgery: Marker-Free, Online Hand-Eye Calibration Using Optical Coherence Tomography Images." In: *IEEE Robotics and Automation Letters (RA-L)* 3.4 (2018), pp. 3944–3951 (cit. on p. 64).

[237] M. Zhou, X. Hao, A. Eslami, K. Huang, C. Cai, C. P. Lohmann, N. Navab, A. Knoll, and M. A. Nasseri. „6DOF Needle Pose Estimation for Robot-Assisted Vitreoretinal Surgery." In: *IEEE Access* 7 (2019), pp. 63113–63122 (cit. on p. 64).

[238] X. Zhou, J. Zhuo, and P. Krahenbuhl. „Bottom-up object detection by grouping extreme and center points." In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 (cit. on pp. 43, 110).

[239] C. L. Zitnick and P. Dollár. „Edge Boxes: Locating Object Proposals from Edges." In: *European Conference on Computer Vision (ECCV)*. 2014 (cit. on pp. 33, 34).

[240] Y. Zou, Z. Luo, and J.-B. Huang. „DF-Net: Unsupervised Joint Learning of Depth and Flow using Cross-Task Consistency." In: *European Conference on Computer Vision (ECCV)*. 2018 (cit. on p. 83).

CURRICULUM VITAE

PERSONAL DATA

Name	Kevis Kokitsi Maninis
Date of Birth	August 08, 1990
Place of Birth	Tochigi, Japan
Citizen of	Greece

EDUCATION

2015 – 2019	Computer Vision Lab, ETH Zürich Doctoral Studies
2009 – 2014	National Technical University of Athens Diploma (M.Eng) in Electrical and Computer Engineering
2005 – 2008	1st Lyceum of Rafina High-School

PROFESSIONAL EXPERIENCE

2015 – 2019	Research Assistant *ETH Zürich*
2018	Research Intern *Facebook AI Research*
2013 – 2014	Research Assistant *National Technical University of Athens*