

Towards Scalable Segmentation of 3D Image Stacks

THÈSE N° 6937 (2016)

PRÉSENTÉE LE 17 JUIN 2016

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS

LABORATOIRE DE VISION PAR ORDINATEUR

PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Carlos Joaquin BECKER

acceptée sur proposition du jury:

Prof. W. Gerstner, président du jury

Prof. P. Fua, directeur de thèse

Dr S. Saalfeld, rapporteur

Prof. J. Buhmann, rapporteur

Dr G. Knott, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2016

Abstract

Imaging modalities such as Electron Microscopy (EM) and Light Microscopy (LM) can now deliver high-quality, high-resolution image stacks of neural structures. Though these imaging modalities can be used to analyze a variety of components that are critical to understanding brain function, the amount of human annotation effort required to analyze them remains a major bottleneck. This has triggered great interest in automating the annotation process, with most state-of-the-art algorithms nowadays relying on machine learning. However, such methods still require significant amounts of labeled examples for training, which can be highly time consuming and arduous, stressing the need for new approaches that require less amount of human effort. In light of this, we present here two efficient machine learning algorithms that incorporate expert knowledge to maximize prediction performance and simultaneously speed up analysis by reducing the required amount of labeled data.

First, we present a new approach for the automated segmentation of synapses in image stacks acquired in EM that relies on image features specifically designed to take spatial context into account. These features are used to train a classifier that can effectively learn cues such as the presence of a nearby post-synaptic region. Our algorithm successfully distinguishes synapses from the numerous other organelles that appear within an EM volume, including those whose local textural properties are relatively similar. We evaluate our approach on three different datasets and demonstrate our ability to reliably collect shape, density, and orientation statistics over hundreds of synapses.

Second, we focus on reducing the required amount of annotation effort. Due to changing experimental conditions in the image acquisition process, successive stacks often exhibit differences that are severe enough to make

it difficult to use a classifier trained for a specific volume on another one. This means that the tedious annotation process has to be repeated for each new stack, resulting in a major bottleneck. We present a domain adaptation algorithm that addresses this issue by effectively leveraging labeled examples across different acquisitions and significantly reducing the annotation requirements. Our approach can handle complex, non-linear image feature transformations and scales to large microscopy datasets and high-dimensional feature spaces. We evaluate our approach on four EM and LM applications where annotation is very costly. We achieve a significant improvement over the state-of-the-art methods and demonstrate our ability to greatly reduce human annotation effort.

Third, we apply our synapse segmentation approach to analyze and compare the structure and shape of synaptic densities in adult and aged mice, such as their area and number of perforations. This detailed analysis requires labeling each voxel within every synapse, making manual annotation unfeasible for large volumes. We show that we can bridge this gap with our approach and demonstrate its effectiveness on six large FIB/SEM brain stacks. Our approach generates segmentations that agree with expert annotations, while requiring very little annotation effort. To our knowledge, we are the first ones to analyze synapse shape in such detail on large stacks, as previous work has strongly relied on manual annotations, restricting analysis to small volumes.

Keywords: Computer vision, segmentation, machine learning, electron microscopy, synapses, medical imaging, domain adaptation.

Résumé

La découverte de nouvelles technologies en imagerie ont été fondamentales aux récentes avancées en neurosciences. Des nouvelles techniques d'acquisition d'images comme la microscopie électronique (ME) et la microscopie optique (MO) sont maintenant capables de générer des images du cerveau de haute résolution et haute qualité. Même si ces techniques d'acquisition des images peuvent être utilisées pour étudier la fonctionnalité du cerveau, leur analyse demande, dans la majorité des cas, un travail humain d'annotation conséquent. Des systèmes d'automatisation d'annotations ont permis la mise en place de la plupart des algorithmes à la pointe qui sont dans leur majorité issus du domaine de l'apprentissage automatique. Cependant, ces méthodes demandent toujours un nombre de données important pour mettre en place leur apprentissage et donc un effort humain conséquent. Alors que l'un des objectifs est de minimiser cet effort, de nouvelles approches doivent donc être proposées. C'est dans cette optique que nous proposons deux algorithmes d'apprentissage automatique qui incorporent la connaissance d'experts pour maximiser la performance de la prédiction du système, ainsi qu'en accélérant simultanément l'analyse, cela en réduisant la quantité de données annotées nécessaire.

Premièrement, nous proposons une nouvelle stratégie pour la segmentation de synapses dans les images de ME qui utilise des caractéristiques spécialement définies pour prendre en compte le contexte. De telles représentations sont utilisées pour entraîner un algorithme de classification capable, par exemple, d'apprendre à reconnaître la présence d'une région postsynaptique dans son voisinage. Notre algorithme est capable de distinguer des synapses des autres structures présentes dans les images de ME, et ce, même si ces autres structures présentent une apparence locale très similaire. Nous validons notre technique sur trois ensembles de données de ME et démontrons

sa capacité à collecter des informations de forme, de densité et d'orientation sur des centaines de synapses.

Deuxièmement, nous proposons une nouvelle méthode d'apprentissage pour réduire la quantité d'annotations manuelles à fournir avec les images d'entraînement. À cause des conditions d'acquisition changeantes, des images successives montrent de larges différences. Ceci pose un problème pour les méthodes automatiques et rend difficile l'application d'une méthode de classification entraînée dans une image sur une autre. Par conséquent, il faut annoter manuellement de larges quantités d'exemples chaque fois qu'une image est acquise, cela demandant beaucoup de temps. Pour surmonter cette difficulté, nous développons un algorithme d'adaptation de domaine capable d'exploiter les annotations dans de multiples acquisitions et de réduire la quantité d'exemples nécessaire pour la segmentation de chaque nouvelle image. Notre modèle peut traiter des représentations complexes, avec des transformations non-linéaires et peut être appliqué à de larges ensembles de données d'images microscopiques qui impliquent souvent des représentations dans d'espaces de grande dimension et de grands volumes de donnée 3D. Nous validons notre algorithme dans quatre images de ME et MO où faire des annotations est extrêmement coûteux.

Troisièmement, nous utilisons notre algorithme de segmentation automatique pour analyser et comparer la structure et les formes des synapses dans des souris adultes et âgées. Cette analyse serait impossible sans un algorithme d'apprentissage automatique, car il faudrait annoter manuellement des images de grande taille. Nous démontrons que notre procédure résout ce problème et est capable de générer des résultats de haute qualité et cela efficacement, sans avoir besoin de grandes quantités d'exemples annotés. À notre connaissance, nous sommes les premiers à analyser les formes des synapses avec de tels détails sur de larges volumes, puisque les travaux précédents ont toujours nécessité l'annotation manuelle d'exemples, les contraignant à analyser de petits volumes.

Mots-clés: Vision par ordinateur, segmentation, apprentissage automatique, microscopie électronique

Acknowledgements

There are many people without whom this work would not have been possible. I want to first thank Pascal Fua for his guidance and for giving me the opportunity to be part of EPFL and his exceptional lab. I am very thankful to him for the knowledge and experience I acquired during these last four years at CVLab.

I thank the members of the committee, Stephan Saalfeld, Joachim Buhmann and Graham Knott for accepting to evaluate this thesis. I want to particularly thank Stephan for offering me the possibility to work with him and his lab at Janelia, which helped open my perspective to new aspects of microscopy and biological imaging. It was at Janelia where the chapter on Multiple Instance Learning of this thesis was born. I also want to greatly thank Graham for his time and patience at answering my naive questions about biology and neural functioning. He was a great source of motivation, especially for the last chapter of this thesis.

Four times around the sun is a long journey on Earth; at times it is easy to feel overwhelmed and lonely when life breaks. Fortunately, I was very lucky to be surrounded by not only incredibly smart people, but also caring and warm-hearted friends. I want to give special thanks to Emti, Radhakrishna and Karim for their kindness, motivation and unconditional support during the toughest times. I also want to thank Amaury for his friendship, discussions and guidance, and of course for his outstanding garlic.

Aboard the PhD cruise I was fortunate to share my office with Tomasz, Karim¹ and Ksenia, with whom I shared uncountable laughs, research discussions, and great muffins and cakes. I also want to thank Agata, Anne, Aurelien, Pablo, Pol, Ursula, Raphael, Gill and Parag for their friendship, coffee breaks and beers shared together. I am also very thankful to Emilie for the wonderful times spent together, her love, kindness and courage.

¹Also known as the *boosting messiah*.

I want to especially thank Karim Ali and Mario Christoudias, known as the “que te pasa man” team, for the many discussions and research we did together. They were both greatly inspirational and it is in great part because of them that I am successfully submitting this thesis today. I want to give special thanks to Josiane for her kindness and assistance whenever needed.

In machine learning we know well that, when appropriately chosen, noise can be a fundamental regularizer. Noise or not, music always played a big role in me, and I want to express my gratitude to Mario, Kenneth, Pol, Dimitri, Róger and Damien for helping me fulfill my insatiable quest for noise while at EPFL.

Finally, I want to give special thanks to my family for their love and support. To my parents for their unconditional help, patience, kindness and support to anything I aspired to. To my grandmother Nurie and all the “domingueros”, for the beautiful times spent together and for teaching me unconditional love. To Victoria and Pablo for being such great loving and playful siblings, with whom spending time is always relaxing and joyful. To Irene and Mario, for their warmth and for aiding me understand love and existence. To Maria, Marina, Jaume and Irene for their kindness, warmth and welcoming hospitality. To Nicole, Véronique, Gérard, Nicolas, and all the relatives in France, for being such beautiful people and embracing me from the very first time I met them.

CONTENTS

List of Figures	v
List of Tables	vii
1 Introduction	1
1.1 Motivation	1
1.2 Challenges	2
1.2.1 Data Size & Scalability	2
1.2.2 Annotation Effort & Interactivity	3
1.2.3 Acquisition Variability	3
1.3 Contributions	4
1.3.1 Context Cues for Synapse Segmentation	5
1.3.2 Domain Adaptation for Microscopy Imaging	6
1.3.3 Synaptic Structure in the Aging Mouse Cortex	6
1.4 Outline	7
I Synapse Segmentation and Detection	9
2 Context Cues for Synapse Segmentation	11
2.1 Introduction	11
2.2 Related Work	13
2.3 Proposed Approach	17
2.3.1 Context Cue Location	18
2.3.2 Context Cue Features	19

CONTENTS

2.3.3	Contextual Classifier	19
2.4	Implementation Details	20
2.4.1	Image Channels	20
2.4.2	Context Cue Parametrization	21
2.4.3	Estimating Synaptic Cleft Orientation	22
2.4.4	Anisotropic Stacks	23
2.4.5	Learning Method	23
2.4.6	Pose Indexing	24
2.4.7	Supervoxels	25
2.5	Experiments	26
2.5.1	Datasets and Evaluation Methodology	26
2.5.2	Baselines	30
2.5.3	Voxel-wise Accuracy	31
2.5.4	Selected Features	33
2.5.5	Detection Accuracy	34
2.5.6	Biological Statistics	36
2.5.7	Computational Complexity	37
2.6	Conclusion	38
3	Automatic Polarity Estimation with Clustered Multiple Instance Learning	41
3.1	Introduction	41
3.2	Related Work	43
3.3	Classic Multiple Instance Learning	44
3.3.1	Classic MIL Model and Formulation	44
3.3.2	Applying MIL to Synapse Segmentation	46
3.4	Proposed Approach	46
3.4.1	Clustering Instances and Orientations	47
3.4.2	CMIL Formulation	48
3.4.3	Numerical Stability	49
3.5	Evaluation	49
3.5.1	Baselines	49
3.5.2	Quantitative Results	50
3.5.3	Qualitative Results	52

3.6	Conclusion	52
II	Domain Adaptation	53
4	Domain Adaptation for Microscopy Imaging	55
4.1	Introduction	55
4.2	Related Work	60
4.3	Our Approach	62
4.3.1	Shared Latent Space Model	62
4.3.2	Weak Learners	64
4.4	Evaluation	67
4.4.1	Datasets	67
4.4.2	Experimental Setup	70
4.4.3	Baselines	71
4.4.4	Results: Mitochondria and Synapse Segmentation	71
4.4.5	Results: Path Classification	74
4.4.6	Partial Dependence Analysis	81
4.5	Conclusion	82
III	Synaptic Structure in the Aging Mouse Cortex	83
5	Synaptic Structure in the Aging Mouse Cortex	85
5.1	Introduction	85
5.2	Data Acquisition and EM Stacks	86
5.2.1	Manual Annotation with Spheres	86
5.3	Automatic Segmentation	87
5.3.1	Data Annotation	88
5.3.2	Training	88
5.3.3	Prediction	88
5.3.4	False Positive Removal	89
5.4	Comparison and Results	89
5.4.1	Synapse Counting & Density	89
5.4.2	Synaptic Cleft Size	90

CONTENTS

5.5	Exploting Segmentation Data: Synaptic Cleft Shape	95
5.5.1	Biological Motivation	95
5.5.2	Data Processing	95
5.5.3	Synaptic Cleft Shape Analysis	96
5.6	Conclusion	100
IV	Final Words	103
6	Concluding Remarks	105
6.1	Future Work	106
6.1.1	Automated Segmentation of Organelles	107
6.1.2	Domain Adaptation	108
	References	111

LIST OF FIGURES

1.1	Importance of context for synapse segmentation	5
2.1	Importance of context for synapse segmentation	12
2.2	Context cues representation.	17
2.3	Image channels.	18
2.4	Coordinate system versors in a simulated synapse.	22
2.5	Qualitative segmentation results.	27
2.6	Illustration of the ambiguities at the synaptic boundary and evaluation ignore zone.	28
2.7	Example of the exclusion zone.	29
2.8	Precision-recall curves for segmentation.	31
2.9	Highest Jaccard index as a function of exclusion zone size d	32
2.10	Our approach with and without orientation estimation.	33
2.11	Illustration of the features selected by AdaBoost for synapse segmentation.	34
2.12	Examples of detected synaptic voxels after thresholding.	35
2.13	Synapse detection performance.	35
2.14	Examples of polarity prediction.	37
2.15	Results of our approach on a large Somatosensory Cortex stack.	39
3.1	Illustration of classic MIL and our proposed CMIL approach.	42
3.2	Generation of consistent synapse polarities.	47
3.3	Segmentation performance of different MIL methods.	50
3.4	Polarity estimates computed by our MIL approach on the training data.	51

LIST OF FIGURES

4.1	Segmentation and path classification applications we consider for Transfer Learning.	56
4.2	Illustration of the difference between Multi-Task Learning and our Domain Adaptation approach.	58
4.3	Quantitative results for EM Segmentation.	72
4.4	Qualitative results for EM Segmentation.	73
4.5	Quantitative results for Path Classification, 3D as Source Domain. . . .	75
4.6	Quantitative results for Path Classification, 2D as Source Domain. . . .	76
4.7	Qualitative results for the OPF path classification dataset.	78
4.8	Qualitative results for the Brightfield path classification dataset. . . .	79
4.9	Analysis of the behavior of the trained classifiers through partial dependence plots.	80
5.1	Comparison of synapse density estimates.	90
5.2	Manual annotation: distribution of synaptic cleft size.	91
5.3	Our approach: distribution of synaptic cleft size.	93
5.4	Ours corrected: distribution of synaptic cleft size.	94
5.5	Example of surface fitting on a synapse with two perforations.	97
5.6	Feret diameter equivalent area and true synapse surface area	98
5.7	Distribution of synaptic cleft surface area.	99
5.8	Proportion of perforated synapses in each stack.	100
5.9	Distribution and relative quantity of perforated synapses.	101
6.1	Illustration of a T-Bar synapse.	107

LIST OF TABLES

2.1	Algorithm parameters and default values.	20
2.2	Dataset description.	26
2.3	Train and test times for our approach and baselines.	38
3.1	Examples of commonly-used soft-max functions in MIL.	45
4.1	Description of the segmentation and path classification experiments used for evaluation.	68
5.1	Details about the six C57 black 6 male mice involved in the experiments.	86

We don't want to conquer the cosmos, we simply want to extend the
boundaries of Earth to the frontiers of the cosmos.

Stanisław Lem, Solaris.

INTRODUCTION

1.1 Motivation

New imaging technologies have been a key driver of recent advances in neuroscience. Imaging modalities such as Electron (EM) and Light Microscopy (LM) can now deliver high-quality, high-resolution image stacks of neural structures. These imaging techniques can be used to analyze a variety of components that are critical to understanding brain function. For example, evidence obtained from EM images suggests that the size, shape and distribution of synapses vary during the course of normal life but also under specific pathological conditions [74]. Similarly, EM imaging has provided new insights into synaptic signaling [62], its relationship to mitochondrial activity [70], as well as to some neuro-degenerative diseases [55, 86].

Analysis is typically carried out by manually segmenting the various structures of interest using tools such as Fiji [93], or through a combination of manual and semi-automated tools, such as [77, 92, 95]. This is not only a tedious and time consuming process but also an error-prone one. Thus, while the growing number of EM datasets offers a unique opportunity to unlock new concepts and secrets of neuronal function, the required amount of human effort remains a major bottleneck. Therefore, there has been a great interest in automating the annotation process and most state-of-the-art algorithms nowadays rely on machine learning.

Though machine learning algorithms are typically flexible enough to be applied to many different fields and modalities, it is often necessary to adapt them to the task at hand to maximize the quality of their predictions. For example, EM data poses

1. INTRODUCTION

unique challenges for automatic segmentation algorithms, as the volumes are heavily cluttered with structures that exhibit similar textures and are therefore difficult to distinguish based solely on local image statistics. Consequently, directly applying a machine learning algorithm for such task may lead to poor results, particularly in the presence of limited annotations.

Therefore, we look forward to exploiting expert knowledge and incorporating it into machine learning algorithms, improving prediction performance, while simultaneously reducing the required amount of labeled data.

In this thesis we focus on the segmentation of organelles in EM stacks. We place special emphasis on synapse segmentation, as this represents a crucial step towards understanding the functioning mechanisms behind synaptic transmission and plasticity [42]. We also show that our algorithms are generic enough and can be adapted to tasks such as mitochondria segmentation and neuron tracing in Light Microscopy imaging, allowing us to efficiently process large stacks with high performance and little annotation effort.

In the remainder of this chapter we introduce the challenges of EM data, followed by the contributions and outline of this thesis.

1.2 Challenges

EM data poses a series of challenges which must be tackled simultaneously to guarantee a feasible and practical solution. We discuss these different aspects below.

1.2.1 Data Size & Scalability

EM can now deliver up to 4 nm voxel-size 3D stacks, revealing high details of fine structures in neural tissue. While this is highly desirable to explore deeper into the brain and its functioning, it also poses an important challenge for automated algorithms, as they need to be able to deal with large amounts of data for both training and prediction.

For example, the size of the brain of a drosophila fruit fly, an insect that has been a subject of great interest [46, 85], is around $600 \times 600 \times 200 \mu\text{m}$ [53]. Imaged with anisotropic EM with a voxel size of $5 \times 5 \times 20 \text{ nm}$, this translates to a stack of 144 terabytes. Therefore, to be of practical use, we need algorithms that are not only fast, but that scale linearly or sub-linearly with stack size.

Another important aspect, sometimes disregarded, is that image stacks are intrinsically three-dimensional. Though slices can be analyzed individually, this typically happens at the expense of information loss. For structures such as synapses that extend over small regions and are oriented, treating 2D slices independently may incur in a significant performance loss, as we show in the experiments of Chapter 2. Therefore, to exploit the full potential of the data, we aim at processing the volumes directly in 3D whenever possible.

1.2.2 Annotation Effort & Interactivity

Supervised machine learning algorithms require labeled data for training. Typically, the more annotated data are available to the algorithm, the more accurate it is at predicting unseen data. However, labeling 3D stacks can be very time consuming and arduous, in particular for tasks such as segmentation, where per-voxel labels are needed. Therefore, methods that require less training data to achieve certain performance level are preferred over others. This is particularly important when working with interactive environments, such as ilastik [95] or Espina [77], to deliver quality results with less effort and speed up analysis.

Simultaneously, an interactive scenario not only requires fast prediction, but it must also be fast to re-train, as new data is annotated. Therefore, we aim at developing approaches that are fast to train and predict, and require relatively little labeled data to achieve a desired performance.

1.2.3 Acquisition Variability

Machine learning normally relies on the fact that the training and run-time data samples are drawn from the same distribution. In microscopy, this may be a problem because the data preparation processes tend to be complicated and not easily repeatable, which means that a classifier trained on one acquisition is unlikely to perform very well on a new one, even when using the same modality.

For example, acquiring the EM images of brain structures normally requires tissue staining to increase contrast, followed by resin encasing before the acquisition. As a result, two samples of the same brain region acquired at different times may look significantly different due to differences in their preparation. This is even more true when the samples come from different parts of the brain, so that classifiers trained for

1. INTRODUCTION

one part of the brain perform poorly on another one. While it is theoretically possible to gather new training data after each new image acquisition, it is impractical if high throughput is desired.

Hence, it is necessary to take acquisition differences into account to make machine learning algorithms of practical use and to reduce annotation effort. At first it may look as if features can be easily normalized to compensate for different acquisitions. However, as we will see later, typical normalizations such as zero-mean-unit-variance or histogram equalization are not effective. Moreover, the staining and acquisition processes are very complex and difficult to model, which makes it hard to find a proper normalization function.

1.3 Contributions

In this thesis we aim at tackling the challenges mentioned above simultaneously, whenever possible. We first introduce a novel synapse segmentation approach that works directly in 3D, exploiting contextual information that is well known to experts. The result is an efficient segmentation method that outperforms state-of-the-art approaches. Part of this work appears in [12, 13].

We then propose a new domain adaptation method to compensate for acquisition variability in EM and LM imaging. We do so by allowing for a coordinate-wise non-linear transformation between domains. Our approach simultaneously learns the decision boundary and a non-parametric estimation of the transformation between domains. Our method requires very little labeled data in the target domain, outperforming existing domain adaptation approaches. Part of this work appears in [14, 15].

Finally, we evaluate the effectiveness and practicality of our synapse segmentation approach by applying it to six large mice stacks to analyze differences in synapse density distribution and geometry between aged and young adult mice. We show that our approach generates segmentations that agree with expert annotations, while requiring very little annotation effort.

We describe each contribution next.

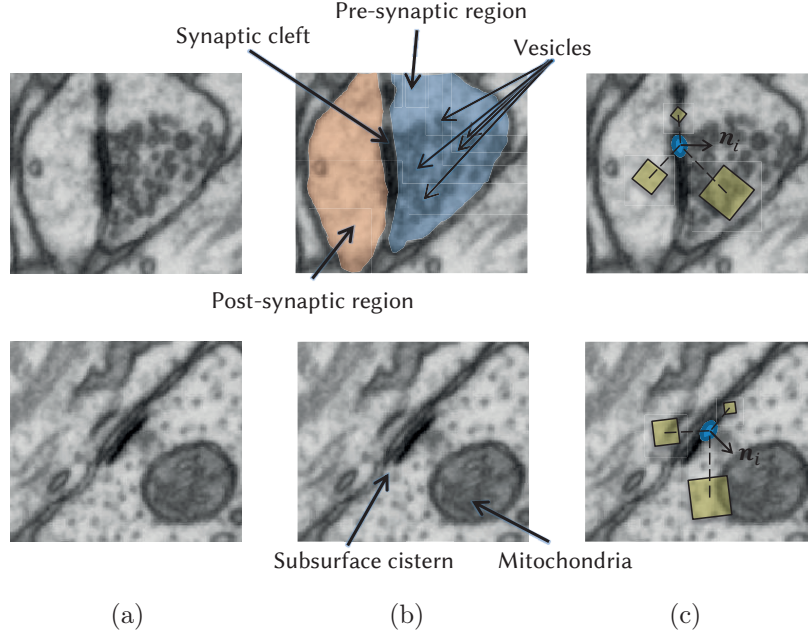


Figure 1.1: Importance of context for synapse segmentation. (a) Two close-ups of regions containing wide dark structures that could potentially be synaptic clefts. However, only the one at the top really is one, as evidenced by the small spheres known as vesicles on its right. These denote the pre-synaptic region and are missing from the bottom image. The diagram on top of (b) depicts the three elements that evidence the existence of a synapse, namely the synaptic cleft and the pre-synaptic and post-synaptic regions. The latter are labeled in blue and red, respectively. (c) The features we use are designed to capture this fact. To classify a voxel (blue), we consider sums over image cubes (shown as yellow squares) whose respective positions are defined relative to an estimated normal vector \mathbf{n}_i .

1.3.1 Context Cues for Synapse Segmentation

We introduce a new approach for automatically segmenting synapses in EM image stacks that relies on image features specifically designed to take spatial context into account. As shown in Fig. 1.1, a synapse can only be distinguished from other structures by relying on contextual clues such as the presence of a nearby cluster of vesicles.

We therefore design features that can capture relevant context around the voxel of interest, that are then used to train a classifier that can effectively learn cues such as the presence of a nearby post-synaptic region. As a result, our algorithm successfully distinguishes synapses from the numerous other organelles that appear within an EM volume, including those whose local textural properties are relatively similar.

Furthermore, as a by-product of the segmentation, our method flawlessly determines

1. INTRODUCTION

synaptic orientation, a crucial element in the interpretation of brain circuits. We evaluate our approach on three different datasets, compare it against the state-of-the-art in synapse segmentation and demonstrate our ability to reliably collect shape, density, and orientation statistics over hundreds of synapses.

1.3.2 Domain Adaptation for Microscopy Imaging

We then move on to tackling the problem of acquisition variability, to drop the need to extensively annotate newly-acquired stacks. We aim at compensating the shift between acquisitions by introducing a novel domain adaptation approach. We assume that features have undergone an unknown non-linear transformation, but that their usefulness for the task at hand (e.g. segmentation) has not changed. Our method effectively leverages upon labeled examples across different acquisitions, significantly reducing annotation requirements. Our approach can handle complex, non-linear image feature transformations and scales to large microscopy datasets that often involve high-dimensional feature spaces and large 3D data volumes.

We evaluate our approach on four challenging EM and LM applications that exhibit very different image modalities and where annotation is very costly. Across all applications we achieve a significant improvement over the state-of-the-art machine learning methods and demonstrate our ability to greatly reduce human annotation effort.

1.3.3 Synaptic Structure in the Aging Mouse Cortex

Finally, we apply our synapse segmentation approach to analyze and compare the structure and shape of synaptic densities between adult and aged mice. Such detailed analysis requires labeling each voxel within every synapse in a stack, rendering manual annotation unfeasible for large volumes. We show we can bridge this gap with our automated segmentation approach, and show its effectiveness on six large EM brain stacks acquired from six different mice. Three of them are young adults, and the other three are aged mice, offering an excellent opportunity to try our segmentation approach to analyze how aging affects the brain.

We demonstrate that our approach can efficiently generate full 3D segmentations that agree with expert annotations, while requiring very little annotation effort. To our knowledge, we are the first ones to analyze synapse shape in such detail on large

stacks, as previous work has strongly relied on manual annotations, restricting analysis to small volumes.

1.4 Outline

We begin with Part I, introducing in Chapter 2 our synapse segmentation approach for EM stacks. We use a classifier that relies on features designed to capture context around the voxel of interest, inspired by the information experts use to identify synapses. We show our approach outperforms the state-of-the-art on three datasets, while being efficient and fast to train and predict. Next, we combine our approach with Multiple Instance Learning (MIL) to drop the need for polarity annotations during training in Chapter 3. We show that our MIL-based method can flawlessly identify synapse polarity at training time, giving up the need for extra manual labeling effort.

Part II targets the problem of acquisition variability and introduces our boosting-based domain adaptation approach. We develop an algorithm that compensates for inter-domain feature transformations. These transformations are estimated at the same time as the decision boundary is learned, resulting in a compact and efficient algorithm.

In Part III we apply our segmentation approach to obtain biologically-relevant synapse measures on six large stacks from young adult and aged mice. We show the effectiveness of our automated method to reducing the annotation effort, while generating high quality segmentation outputs that would be unattainable with full manual annotation.

Finally, we close this thesis with Chapter 6 with the concluding remarks and a discussion on future work.

Part I

Synapse Segmentation and Detection

CONTEXT CUES FOR SYNAPSE SEGMENTATION

In this chapter we present a new approach for the automated segmentation of synapses in image stacks acquired by Electron Microscopy (EM) that relies on image features specifically designed to take spatial context into account. These features are used to train a classifier that can effectively learn cues such as the presence of a nearby post-synaptic region. As a result, our algorithm successfully distinguishes synapses from the numerous other organelles that appear within an EM volume, including those whose local textural properties are relatively similar.

Furthermore, as a by-product of the segmentation, our method flawlessly determines to which side of the synapse the pre- and post-synaptic regions are located, a crucial element in the interpretation of brain circuits. We evaluate our approach on three different datasets, compare it against the state-of-the-art in synapse segmentation and demonstrate our ability to reliably collect shape, density, and orientation statistics over hundreds of synapses.

2.1 Introduction

EM data poses unique challenges for automatic segmentation algorithms in part because the volumes are heavily cluttered with structures that exhibit similar textures and are therefore difficult to distinguish based solely on local image statistics. The synapse segmentation task is well illustrative of this difficulty. As shown in Fig. 2.1(b), a synapse can only be distinguished from other structures by relying on contextual clues such as the presence of a nearby cluster of vesicles. Well-established criteria enable human expert to

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

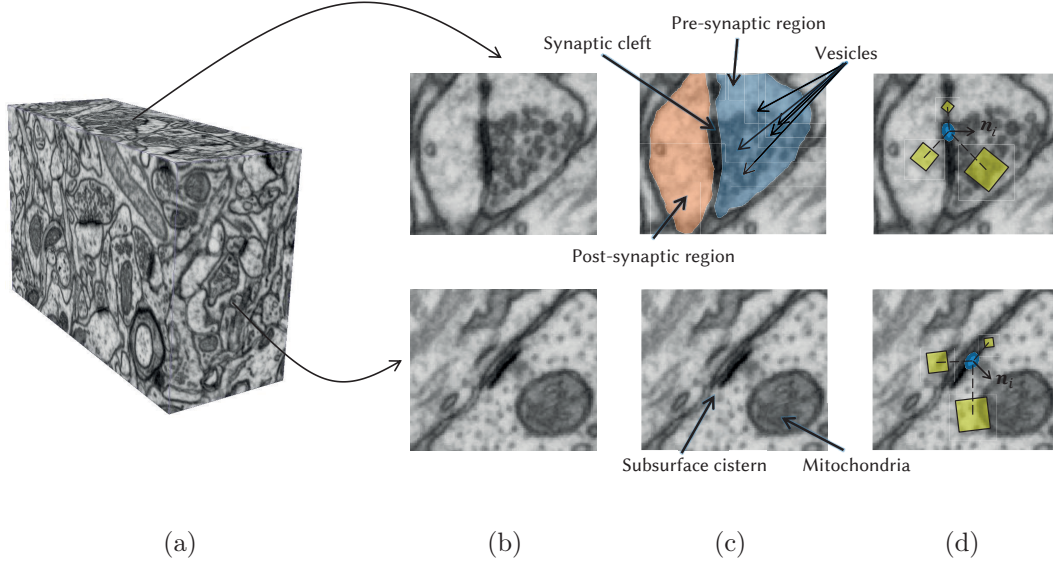


Figure 2.1: Importance of context for synapse segmentation. (a) A FIBSEM stack with $5nm$ resolution in all three directions. (b) Two close-ups on regions containing wide dark structures that could potentially be synaptic clefts. However, only the one at the top really is one, as evidenced by the small spheres known as vesicles on its right. These denote the presynaptic region and are missing from the bottom image. The diagram on top of (c) depicts the three elements that evidence the existence of a synapse, namely the synaptic cleft and the pre-synaptic and post-synaptic regions. The latter are labeled in blue and red, respectively. (d) The features we use are designed to capture this fact. To classify a voxel (blue), we consider sums over image cubes (shown as yellow squares) whose respective positions are defined relative to an estimated normal vector \mathbf{n}_i .

identify synapses: densities on the pre-synaptic and post-synaptic membranes, vesicles in the pre-synaptic axon terminal and finally a synaptic cleft, as shown in Fig. 2.1(c). It is therefore essential for an automatic segmentation method to proceed in a similar fashion.

Current methods for automated synapse detection either require first finding cell membranes [76] or operate on individual slices [45], thus failing to leverage the 3D structure of the data. By contrast, the recent method of [57, 58] operates entirely in 3D. However, the latter does not exploit the contextual clues that allow human experts to distinguish synaptic clefts from other membranes exhibiting similar textures such as myelin sheaths. More generally, though progress has been made towards the segmentation of various organelles from EM stacks, context has yet to be exploited in a meaningful way.

In this chapter, we propose an approach designed to take such contextual cues into account and emulate the human ability to distinguish synapses from regions that merely share a similar texture. Our method is fully automated, processes the data directly in 3D and is specifically designed to leverage context cues. We run various filters over the EM stack and compute our features over arbitrarily sized cubes placed at arbitrary locations inside an extended neighborhood of the voxel to be classified. As this generates a feature representation for each voxel in the order of a hundred thousand, we rely on Boosting to select the relevant filter channels as well as the relevant cube locations and sizes. The resulting classifier is thus highly flexible, able to utilize context from a high variety of regions in the neighborhood of the voxel of interest.

We apply our classifier to the synapse segmentation task and compare our results with the state-of-the-art synapse segmentation method of Kreshuk et al. [57], a fully automated 3D approach which does not utilize context, and with the Convolutional Neural Network (CNN) method of [88]. By working directly in 3D and honing in on the presence of pre-synaptic vesicles and post-synaptic regions, our method significantly outperforms the approaches of [57] and [88]. As an added benefit, our method also flawlessly identifies synaptic orientation, a key and hitherto unexplored task.

We validate our method on three datasets obtained from three different regions of the adult mammalian brain: the Somatosensory cortex, the Hippocampus, and the Cerebellum. We demonstrate our ability to automatically process large EM stacks, reliably collect density, shape and orientation statistics from hundreds of synapses.

Moreover, our approach is not limited to synapse segmentation, and in fact it has already been used as the input of a Conditional Random Field (CRF)-based approach to segment mitochondria in EM, in [67]. The latter shows that, compared to other manually-designed features, using our context features as an input in a CRF significantly improves segmentation performance.

2.2 Related Work

Prior work on segmenting neuronal structures from EM volumes has covered a range of approaches from early attempts at full manual tracing [33, 34, 72] to semi-automatic methods requiring user initialization [19, 64], and lately fully-automated methods [48, 66].

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

Manual segmentation has clear and well understood limitations for the analysis of EM stacks. One notable example can be found in [108], where the successful segmentation of the nervous system of a nematode worm, containing only 302 neurons, necessitated a sustained effort over a ten year period. The need for expert knowledge and the growing size of EM datasets render manual segmentation impractical and highlight the need for automation.

Semi-automated methods based on active contours and level sets [10, 19, 52, 64, 68, 81] as well as graphcuts [78] have achieved some measure of success on EM images. However, these methods require careful manual initialization of each object to be segmented, which is done by supplying seed points and tuning various parameters. Though active interactions and feedback may in the long term prove essential to the successful large-scale segmentation of EM stacks, the amount of user input required by these methods remains prohibitively high. Ultimately, when applied to large EM data sets containing millions or even billions of structures, these semi-automatic segmentation methods suffer from the same intractability issue as their manual counterparts.

Recent research has focused on methods relying on machine learning, requiring little to no user interaction. Among those, several follow the same methodology by performing a segmentation in individual 2D EM slices before linking the segmented regions across slices in 3D. For instance, in [76], a perceptron operating on Hessian ridge was shown to provide promising results in segmenting membranes. However, in addition to the post-processing required to link membranes across the various 2D slices, this method also suffers from the need to remove internal sub-cellular structures from the segmentation result. In [106] a Boosted classifier operating on Gabor filter based features is used to segment mitochondria in 2D slices while a connected component analysis generated the desired 3D segmentation. In [79], 2D mitochondria segmentation followed by simple 3D interpolation is obtained from a number of classifiers including Adaboost, Support Vector Machines and Nearest Neighbor trained on Texton features. Finally, in [54], a random forest classifier trained with Haar-like features is used to detect membranes in individual EM slices, while a graph cut optimization is used to enforce perceptual grouping and 3D continuity constraints.

While slice-by-slice methods have been shown to provide both reasonable segmentation results and computational savings, they fail to leverage the consistency of the structures in all three dimensions. This situation arises in part from the fact these

approaches were designed for anisotropic EM modalities, such as transmission electron microscopy (TEM). Though the reduced resolution in the z-direction makes slice-by-slice approaches a reasonable choice, recent works [4, 48, 80, 98] have demonstrated the benefits of processing the data directly in 3D even in highly anisotropic image stacks. More generally, the appearance of objects in 2D slices can be significantly altered depending on the 3D orientation of the object with respect to the stack axes. Given that such variability is far less pronounced when observing the objects in 3D, processing EM stacks slice by slice significantly complexifies the segmentation task and can prove exceedingly detrimental when compared to direct 3D processing. Such a strategy is clearly foolhardy in the case of 2D images where the analogue would consist of a column by column or a row by row processing.

For these reasons, a number of works have addressed the segmentation of various neuronal structures directly in 3D. For example, [48] uses a multilayer convolutional artificial neural network (ANN) to segment neuronal membranes. By employing a convolutional ANN, [48] removes the need to hand design features and instead learns the necessary filters directly from the data. Andres et al. [4] propose a bottom-up hierarchical segmentation framework that uses a Random Forest classifier and watersheds to segment neural tissue. Though both of these methods produce excellent membrane segmentation results, they are designed for datasets prepared with an extra-cellular stain which highlights cell membranes while suppressing the various intracellular structures. In [98], an affinity graph that can be paired with standard partitioning algorithms is generated using a convolutional ANN. Much as in [48], this method learns both the features as well as the decision function directly from the data.

Even though progress has been made towards the automatic segmentation of neural structures, none of the aforementioned methods, whether operating in 2D or in 3D, can reliably segment objects such as synapses, which are characterized by specific arrangements of structures in addition to local textural cues. Though current algorithms generally compute features in a neighborhood around the voxel of interest, they do not exploit context in a meaningful way: features are either pooled into global histograms [66, 79], are computed in regions centered around the voxel of interest [4, 48, 54, 106], or operate on a limited neighborhood around the voxel of interest [48, 98]. The resulting classifiers are therefore unable to hone in on arbitrary localized context cues.

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

The importance of context for the purposes of segmentation has been highlighted by a few attempts to leveraging ad-hoc and heuristic contextual cues to improve segmentation. For instance, [106] uses vesicle detection cues to suppress false alarms on vesicle clusters that can interfere with mitochondria segmentation, while [52, 101] propose to sample features in a 2D stencil neighborhood around the pixel of interest. By allowing the classifier to measure features computed at various locations in addition to the pixel of interest, [52, 101] are able to identify membranes at regions of minor discontinuities. However, by relying on a pre-determined set of locations from which features can be sampled, these approaches strongly restrict the use of context. By contrast, our approach learns the relevant context automatically, overcoming these limitations.

The recent re-emergence of Neural Networks has also inspired their use for EM segmentation. [46] applies a CNN for synapse detection in the fly brain, consisting of an unsupervised learning component and followed by supervised classification with a multilayer perceptron. Along similar lines, [87] first trains a CNN for membrane detection on a very large dataset, followed by a vesicle detector and a synapse detector that only operates over membranes and nearby vesicles. Another interesting example is the so-called U-Net [88]. In contrast to sliding-window CNNs, the U-Net consists of a contracting and an expanding path in its architecture, which allows to propagate context information from the input image to higher resolution layers, improving segmentation performance with respect to its sliding window counterpart. Although these methods are promising, they require large amounts of training data, and in some cases auxiliary annotations such as labeled membranes. Moreover, they are typically slow to train, limiting their application to interactive scenarios.

Closest to our work is the state-of-the-art method of Kreshuk et al. [57], specifically targeted to synapse segmentation in isotropic image stacks. This approach relies on voxel-wise classification, training a Random Forest classifier that employs a set of pre-defined features such as smoothed gradient magnitudes, Laplacian of Gaussians and Hessian and Structure Tensor eigenvalues, evaluated at the voxel of interest. Therefore, context can only be captured through the isotropic Gaussian filters applied to the image stack, ignoring the presence of the asymmetric and localized context information generated by the pre-synaptic and post-synaptic regions. This typically translates into a high number of false positive detections, which motivated a follow-up publication from

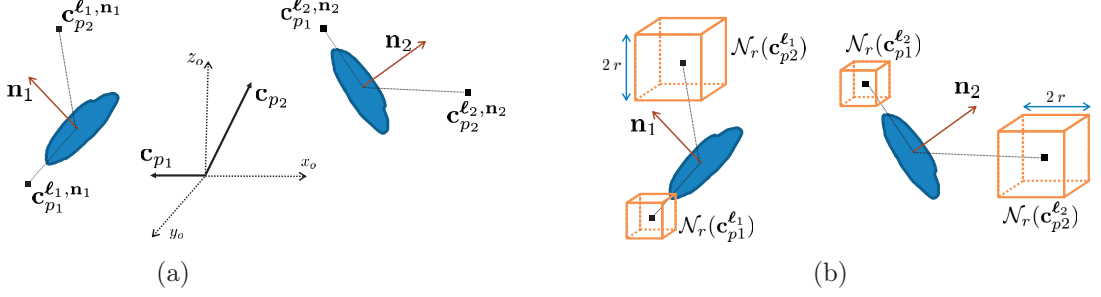


Figure 2.2: Context cues representation: (a) context cue locations \mathbf{c}_p in the global coordinate system x_o, y_o, z_o are rotated according to the orientation estimate of the voxel of interest \mathbf{n} to yield locations $\mathbf{c}_p^{\ell_i}$ that are consistent. (b) At each of these locations, image channels are summed over cubes of radius r around their center. Our approach employs AdaBoost to select the most discriminative features for synapse segmentation.

the same authors in [58]. The latter applies connected components to the segmentation output of the former, and the user is asked to label the false-positive detections. These object labels are then used to train a new classifier to refine the output of the first. Although this helps reduce false positive counts, it requires additional annotation effort from the user, which we believe can be avoided if context information is encoded directly in the first place.

2.3 Proposed Approach

Let $x \in \mathcal{X} = [0, 1]^{W \times H \times D}$ be an EM volume of width W , height H and depth D . Voxels are indexed by $i \in \{1, \dots, W \times H \times D\}$, and the location of each voxel is designated $\ell_i \in \mathbb{N}^3$. Our goal is to find a function $\varphi(x, \ell_i) \in \mathbb{R}$ that yields high scores at locations ℓ_i in the volume that are part of synaptic tissue, and lower score values at those that are not.

As shown in Fig. 2.1(b), it can be difficult to distinguish synapses from other structures based solely on local texture. Human experts confirm their presence by looking nearby for post-synaptic densities and vesicles. This protocol cannot be emulated simply by measuring filter responses at the target voxel [57], pooling features into a global histogram [66, 79] or relying on hand-determined locations for feature extraction [52, 101].

To emulate the human ability to identify synapses, we design features, termed context cues, that can be extracted in any cube contained within a large volume centered on the voxel to be classified at ℓ_i , as depicted in Fig. 2.2(b). They are computed in

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

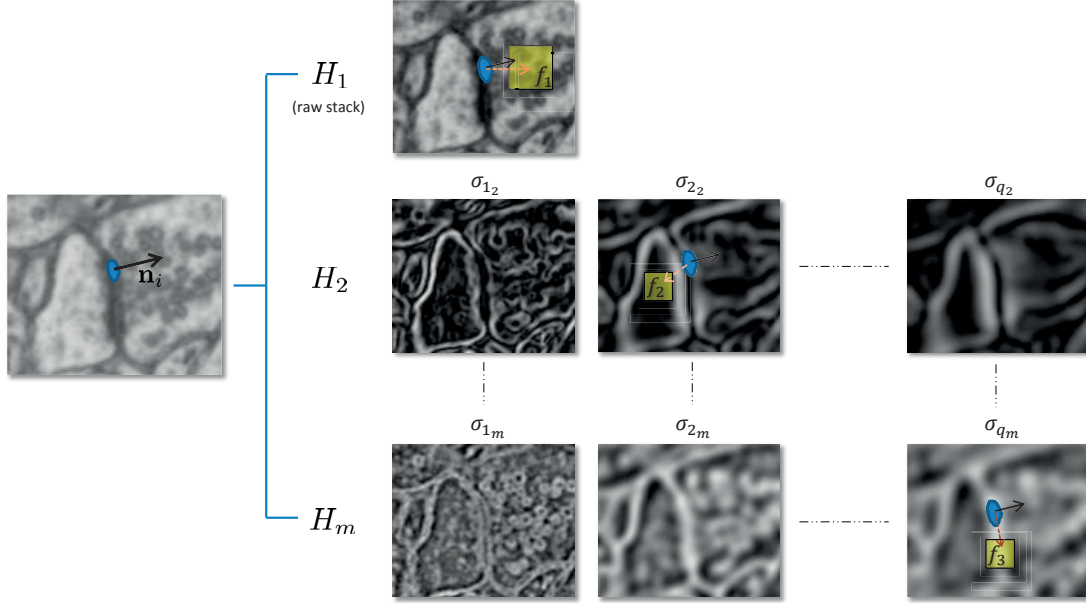


Figure 2.3: Image channels. The image is convolved with different filters and features are computed within the yellow rectangles whose coordinates are expressed with respect to the location of the voxel to be classified and the local orientation vector \mathbf{n} . Each H_i line depicts a specific channel designed to capture different statistical characteristics.

several image channels using a number of Gaussian kernels, as shown in Fig. 2.3. As will be discussed in §2.4.2 this yields more than 100,000 potential features. We therefore rely on AdaBoost [36] to select the most discriminative ones.

Given that synapses have arbitrary 3D orientations, we ensure that our context cues are computed at consistent locations across differently oriented synapses. We rely on the pose-indexing framework of [2, 35] to enforce this consistency.

In the remainder of this section, we describe briefly the main structure of our context features. Their implementation is discussed in more detail later in § 2.4.

2.3.1 Context Cue Location

Let us consider a voxel located at ℓ_i and an associated unit vector $\mathbf{n}_i \in \mathbb{R}^3$, as in Fig. 2.2(a). This unit vector is computed so that it is normal to the synaptic cleft. Let

$$\mathbf{c}_p \in \mathbb{R}^3, \quad p = 1, \dots, P \quad (2.1)$$

denote a set of P locations expressed in the common x_o, y_o, z_o reference frame shown at the center of Fig. 2.2(a). These locations are translated and rotated to occur at consistent locations relative to a target voxel by defining,

$$\mathbf{c}_p^{\ell_i} = \ell_i + \mathbf{R}(\ell_i)\mathbf{c}_p \quad (2.2)$$

where $\mathbf{R}(\ell_i)$ is a 3×3 rotation matrix such that $\mathbf{R}(\ell_i)(0, 0, 1)^T = \mathbf{n}_i$.

2.3.2 Context Cue Features

Given the $\mathbf{c}_p^{\ell_i}$ locations of Eq. 2.2, our goal now is to compute image statistics inside cubic neighborhoods $\mathcal{N}_r(\mathbf{c}_p^{\ell_i})$ of edge length $2r$ centered around these locations, such as those depicted in Fig. 2.2(b).

To this end, we process the original EM volume by convolving it with a number of different filters as depicted in Fig. 2.3. Each of the resulting data cubes, in addition to the original one, is treated as a data channel m , and is smoothed using several isotropic Gaussian kernels with variance σ_{n_m} . We denote the gray levels in the resulting data volumes as

$$H_{m, \sigma_{n_m}}(x, \mathbf{z}) \in \mathcal{X} , \quad (2.3)$$

where x is the original EM volume and \mathbf{z} represents the 3D location. We take context cue features to be

$$f_{\mathbf{c}_p, m, \sigma_{n_m}, r}(x, \ell_i) = \sum_{\mathbf{z} \in \mathcal{N}_r(\mathbf{c}_p^{\ell_i})} H_{m, \sigma_{n_m}}(x, \mathbf{z}) . \quad (2.4)$$

In other words, we sum the smoothed channel output over the cubic boxes centered at all $\mathbf{c}_p^{\ell_i}$ for all possible values of m , σ_n , and r . This yields a set of K features, which we will denote for simplicity

$$f_k(x, \ell_i), \quad k = 1, \dots, K , \quad (2.5)$$

and which we use for classification purposes as explained next.

2.3.3 Contextual Classifier

Given the context features f_k , we create decision stumps by simple thresholding and combine these stumps via a standard AdaBoost procedure [36] into a strong learner of the form

$$\varphi(x, \ell_i) = \sum_{t=1}^T \alpha_t \mathbf{1}_{\{f_t(x, \ell_i) > \rho_t\}} . \quad (2.6)$$

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

Table 2.1: Algorithm parameters and default values.

Parameter	Symbol	Default value
AdaBoost iterations	T	2000
Weighting-by-resampling ratio	$M = \frac{N_{\text{neg}}}{N_{\text{pos}}}$	2
Number of weak learners explored per iteration	Ω	4000
Context cue maximum distance	$\ \mathbf{c}_p\ _{\max}$	40 voxels
Context cue distance quantization steps	$Q_{\ \mathbf{c}\ }$	6
Context cue maximum box size	r_{\max}	20 voxels
Context cue box size quantization steps	Q_r	11
Context cue φ quantization steps	Q_{φ}	9
Context cue θ quantization steps	Q_{θ}	9
Supervoxel seed size	SV_n	2 voxels
Supervoxel cubeness	SV_m	16
Hessian scale for orientation estimation	$\sigma_{\text{Ho}} = \frac{w_s}{2\sqrt{2}}$	18 nm

AdaBoost [36] solves for Eq. (2.6) in a stage-wise manner, building it one term at a time by greedy minimization of an empirical exponential loss. Our resulting classifier is pose-indexed as its constituent features translate and rotate according to ℓ_i and $\mathbf{R}(\ell_i)$ respectively.

2.4 Implementation Details

In what follows, the specifics of our implementation are provided. We follow the same notation as in §2.3 and summarize all algorithm parameters in Table 2.1.

2.4.1 Image Channels

We broadly follow the methodology employed by [57] and process each EM volume with several different filters, resulting in different data channels such as those of Fig. 2.3. Channels can additionally be smoothed using a varying isotropic Gaussian kernels. More specifically, we use:

- The identity (original stack)
- Gradient Magnitude,
- Structure Tensor Eigenvalues

The Gradient Magnitude channels are computed by first smoothing the image with isotropic Gaussian filters of $\sigma_{\text{GM}} = \{1.0, 1.6, 3.5, 5.0\}$. On the other hand, structure tensor eigenvalues are computed at $\rho_{\text{ST}} = \{1.0, 1.6, 3.5, 5.0\}$ with $\sigma_{\text{ST}} = \frac{\rho_{\text{ST}}}{2}$. No smoothing is applied to the identity channel. Given that there are three eigenvalues per structure tensor, this results in a total of 17 different filtered versions of the original EM volume, which are available to construct the weak learners.

Note that, in contrast with our first approach [12] and with the work of [57], we opted for a reduced set of channels. An important observation regarding our framework lies in the fact that our features sum the response of a filter inside a box: In the case of *linear* filters and boxes larger than a single voxel, these sums can be directly computed in the original image channel, up to a scale factor and additional negligible filter border effects. We were therefore able to eliminate the Gaussian smoothing over the original image as well as the Laplacian of Gaussian channel. Several experiments confirmed this observation and further allowed us to eliminate the Hessian Eigenvalue channel which was found to be uninformative for our framework.

2.4.2 Context Cue Parametrization

Context cue locations $\mathbf{c}_p, p = 1, \dots, P$ in the common reference frame are parametrized in spherical coordinates as

$$\mathbf{c}_p = \left(\|c_p\| \cos \varphi_p \sin \theta_p, \|c_p\| \sin \varphi_p \sin \theta_p, \|c_p\| \cos \theta_p \right)$$

with $0 \leq \|c_p\| \leq \|c_p\|_{\text{max}}$, $0 \leq \theta_p \leq \pi$ and $0 \leq \varphi_p \leq 2\pi$. The parameter space is quantized uniformly in $Q_{\|c\|}$, Q_θ and Q_φ bins respectively. This is also applied to the cube edge length $2r$, which is quantized in Q_r steps with $\frac{1}{2} \leq r \leq r_{\text{max}}$.

To compute our context cue features of Eq. 2.4 efficiently, we employ 3D integral images for each channel H_{m,σ_n} . This allows us to compute the sum of any channel inside an arbitrary cube in constant time. Note that to allow for maximum consistency across the differently oriented synapses, the cubes over which sums of image channel

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

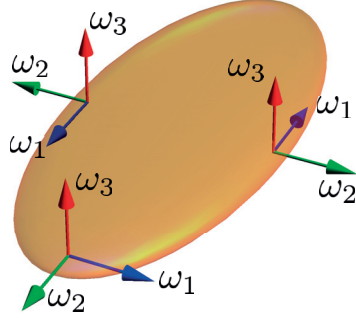


Figure 2.4: Coordinate system versors (unit vectors) computed for $\mathbf{R}(\ell_i)$ at different locations of a simulated synapse. $\omega_1(\ell_i)$, $\omega_2(\ell_i)$, $\omega_3(\ell_i)$ shown in blue, green and red respectively. Note that $\omega_2(\ell_i)$ (green) always points towards the outside of the synapse, which comes as a consequence of Eq. 2.10.

values are computed should also be pose-indexed and hence rotate according to $\mathbf{R}(\ell_i)$. However, this would either impose a heavy memory burden if rotated integral volumes were used or a large computational cost otherwise. For this reason, we do not pose-index the cubes and restrict the boxes over which channel voxels are summed to be axis-aligned as shown in Fig. 2.2(b). Note that, in practice, the axis-aligned cubes overlap significantly with their rotated counterparts and therefore provide a fairly good approximation.

2.4.3 Estimating Synaptic Cleft Orientation

The context cues defined above are located relative to a normal estimate \mathbf{n}_i , which induces the rotation matrix $\mathbf{R}(\ell_i)$ of Eq. 2.2, such that

$$\mathbf{R}(\ell_i)(0, 0, 1)^T = \mathbf{n}_i \quad (2.7)$$

Let $\{\xi_1(\ell_i), \xi_2(\ell_i), \xi_3(\ell_i)\}$ be the Eigenvectors of the Hessian matrix at the voxel of interest, ordered by increasing magnitude of their respective eigenvalues. Hence, $\xi_3(\ell_i) = \mathbf{n}_i$ corresponds to the eigenvector with the highest-magnitude eigenvalue, and is perpendicular to the synaptic cleft, assuming there is one at ℓ_i . We write

$$\mathbf{R}(\ell_i) = \begin{pmatrix} \omega_1(\ell_i) & \omega_2(\ell_i) & \omega_3(\ell_i) \end{pmatrix} \quad (2.8)$$

with

$$\omega_3(\ell_i) = \xi_3(\ell_i) \quad (2.9)$$

$$\omega_2(\ell_i) = \text{sign} [D_{\xi_2(\ell_i)}(\ell_i)] \xi_2(\ell_i) \quad (2.10)$$

$$\omega_1(\ell_i) = \omega_3(\ell_i) \times \omega_2(\ell_i) \quad (2.11)$$

where $D_{\mathbf{u}}(\mathbf{x})$ is the directional derivative of the image along \mathbf{u} at location \mathbf{x} . As shown in Fig. 2.4, introducing Eq. 2.10 makes $\omega_2(\ell_i)$ point towards the outside of the synaptic cleft when ℓ_i is close to the border of the synapse. The motivation behind the latter is that $\mathbf{R}(\ell_i)$ in Eq. 2.7 is only defined up to a rotation in the $\omega_1(\ell_i)/\omega_2(\ell_i)$ plane. However, it is preferable to define a rotation matrix that is consistent, particularly at ℓ_i close to the edge of the synapses. Towards the center of the synaptic cleft, there exists a rotational ambiguity in the $\omega_1(\ell_i)/\omega_2(\ell_i)$ plane which can in fact be ignored due to the symmetry of synaptic structures there. Both the Hessian and the derivatives $D_{\mathbf{u}}(\mathbf{x})$ are computed at scale $\sigma_{\text{Ho}} = \frac{w_s}{2\sqrt{2}}$ where w_s is the average synaptic cleft width, estimated once per EM volume from a single slice.

2.4.4 Anisotropic Stacks

Even though we have so far assumed the stacks be isotropic, anisotropic stacks can be easily handled with two minor modifications. First, the kernels used for generating the image channels and orientation estimates must be scaled accordingly. Finally, context cue locations $\mathbf{c}_p^{\ell_i}$ and the edge length of the cubic neighborhoods $\mathcal{N}_r(\mathbf{c}_p^{\ell_i})$ must be scaled according to the anisotropy of the stack. This allows us to process isotropic and anisotropic stacks indistinguishably.

2.4.5 Learning Method

To make training computationally tractable, we specialize the AdaBoost learning procedure [36] as follows:

2.4.5.1 Weighting-by-sampling

Denote $W = \{w_1 \dots w_N\}$ and $Y = \{y_1 \dots y_N\}$ the weights and labels for each training sample, with $1 \leq i \leq N$. Assume that the weights have been normalized such that $\sum_{i=1}^N w_i = 1$. At each AdaBoost iteration, instead of searching for the weak learner that

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

minimizes the weighted error over all N training samples, we employ the weighting-by-sampling scheme [2, 35] to approximate the distribution W . This is done by finding the weak learner that minimizes a weighted error computed on a subset S of the training data, formally

$$S = \{S_P, S_N\} \quad (2.12)$$

where S_P comprises all the positive samples and S_N is a subset of the negative samples, obtained by weighted sampling with replacement according to the weights W . The weight of each sample in S_P is its respective weight in W , while the samples in S_N are assigned a constant weight $\frac{\sum_{y_i=-1} w_i}{\|S_N\|}$.

In situations where the amount of negative samples outnumbers the number of positives, approaches such as weighting-by-sampling can reduce training time significantly. We call $M = \frac{\|S_N\|}{\|S_P\|}$ the ratio between the number of negative and positive samples selected by weighting-by-sampling, which is a parameter for our algorithm. In our experiments we have observed that segmentation performance is robust against the value of M . We set $M = 2$, which yields faster training without decreasing the performance of the final classifier.

2.4.5.2 Random Weak Learner Search

Due to the large number of possible weak learners $f_k(x, \ell_i)$, $k = 1, \dots, K$, it is impractical to explore them all at each AdaBoost iteration. Instead, we only explore a subset of size Ω , obtained by randomly sampling, at each Boosting iteration, from the pool of K possible weak learners. This also speeds up training significantly. We have experimented with different values of Ω and observed that segmentation performance is fairly independent of its value. For all the results presented here we used $\Omega = 4000$.

2.4.6 Pose Indexing

2.4.6.1 Pose Annotations

Learning our pose-indexed classifier requires annotated training data. Since our contextual features are computed both for a given location and orientation, our training data must include both. While the location of synaptic voxels is manually specified by user annotation, synaptic orientation $\omega_3(\ell_i)$ is automatically extracted as explained in §2.4.3, using eigen analysis on the Hessian matrix computed at ℓ_i . The obtained $\omega_3(\ell_i)$

vector is only defined up to a polarity, which is insufficient given that the pre-synaptic and post-synaptic regions are starkly different in appearance. We therefore follow the standard pose-indexing methodology [35] by labeling the polarity of the $\omega_3(\ell_i)$ vector during training. Note that the orientation labeling procedure in training only requires single user click per synapse to consistently direct all $\omega_3(\ell_i)$ vectors to the pre-synaptic region. Nonetheless, this is an extra burden for the annotator, which we alleviate later in Chapter 3, where we show that we can automatically infer the polarity of all synapses in the training data automatically with a technique based on Multiple Instance Learning.

During testing, our learned contextual classifier $\varphi(\cdot)$ is evaluated for both polarities of the extracted $\omega_3(\ell_i)$ vector and the maximum response retained.

2.4.6.2 Sampling Negatives

Under the pose-indexing framework [2, 35], samples that do not exhibit the same pose (location and orientation) as positive samples should be considered as negative during training. However, in practice, samples that are too close in pose-space to the positives should be excluded from training, as their appearance can be similar to that of positives and their inclusion can therefore deteriorate performance.

For example, following [35], any voxel lying on the synaptic cleft with an incorrect orientation should be treated as a negative sample. Likewise, a voxel that is immediately next to a synaptic cleft should also be considered a negative sample. However, given the overlap in appearance, it is difficult for the learning method to disambiguate such voxels from the positive set. Thus, as is commonly done in object detection [2], we setup conservative training exclusion zones in pose-space around our positive examples and sample negative examples outside these exclusion zones. In particular, we do not use positive voxels with the wrong orientation as negatives, as discussed above. Moreover, we also exclude voxels that are outside the synaptic cleft and less than 10 voxels away from a positive-labeled voxel.

2.4.7 Supervoxels

Our entire algorithm including feature extraction, training and testing is designed and implemented to operate on individual voxels of the EM volumes. However, significant computational savings can be achieved by grouping voxels into supervoxels [1] for specific operations for training and testing. Thus, during training, instead of using every voxel

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

Table 2.2: Dataset description.

Dataset	Voxel Size	Train		Test	
		Size (voxels)	Labeled synapses	Size (voxels)	Labeled synapses
(A) Som. cortex	6.8 nm	$750 \times 564 \times 750$	9 (some)	$655 \times 429 \times 250$	28 (all)
(B) Hippocampus	5 nm	$1024 \times 653 \times 165$	20 (all)	$1024 \times 883 \times 165$ $1024 \times 1536 \times 200$	79 (all)
(C) Cerebellum	5 nm	$1398 \times 1343 \times 299$	7 (some)	$1966 \times 1343 \times 200$	56 (all)

as a positive or negative data sample, we restrict our method to training only on voxels corresponding to centers of super-voxels. In effect, this amounts to a spatially-driven sampling of the training data which significantly speeds up training while maintaining performance. Likewise, during testing, instead of evaluating our learned contextual classifier $\varphi(\cdot)$ on every voxel in the EM test volume, we only evaluate $\varphi(\cdot)$ on voxels corresponding to super-voxel centers while off-center voxels are assigned a response equal to that of the center.

2.5 Experiments

We evaluated our method on three different EM stacks acquired from different regions of the adult rat brain¹. We assessed performance both in terms of voxel-wise segmentation and synapse detection.

In this section, we first describe these datasets and our training and evaluation methodology. We then use our datasets to evaluate both the voxel-wise precision of our method and its accuracy in terms of how many entire synapses are correctly detected. We use the Random Forest method of [57] and the CNN approach of [88] as baselines against which we compare our results. Finally, we show that our method can be used to compute biologically relevant statistics and discuss computational complexity issues.

2.5.1 Datasets and Evaluation Methodology

We used three different datasets from (A) the Somatosensory Cortex, (B) Hippocampus, and (C) Cerebellum of an adult rat. Example slice cuts of each dataset are shown in Fig. 2.5.

¹Source code available at <http://cvlab.epfl.ch/software/synapse>

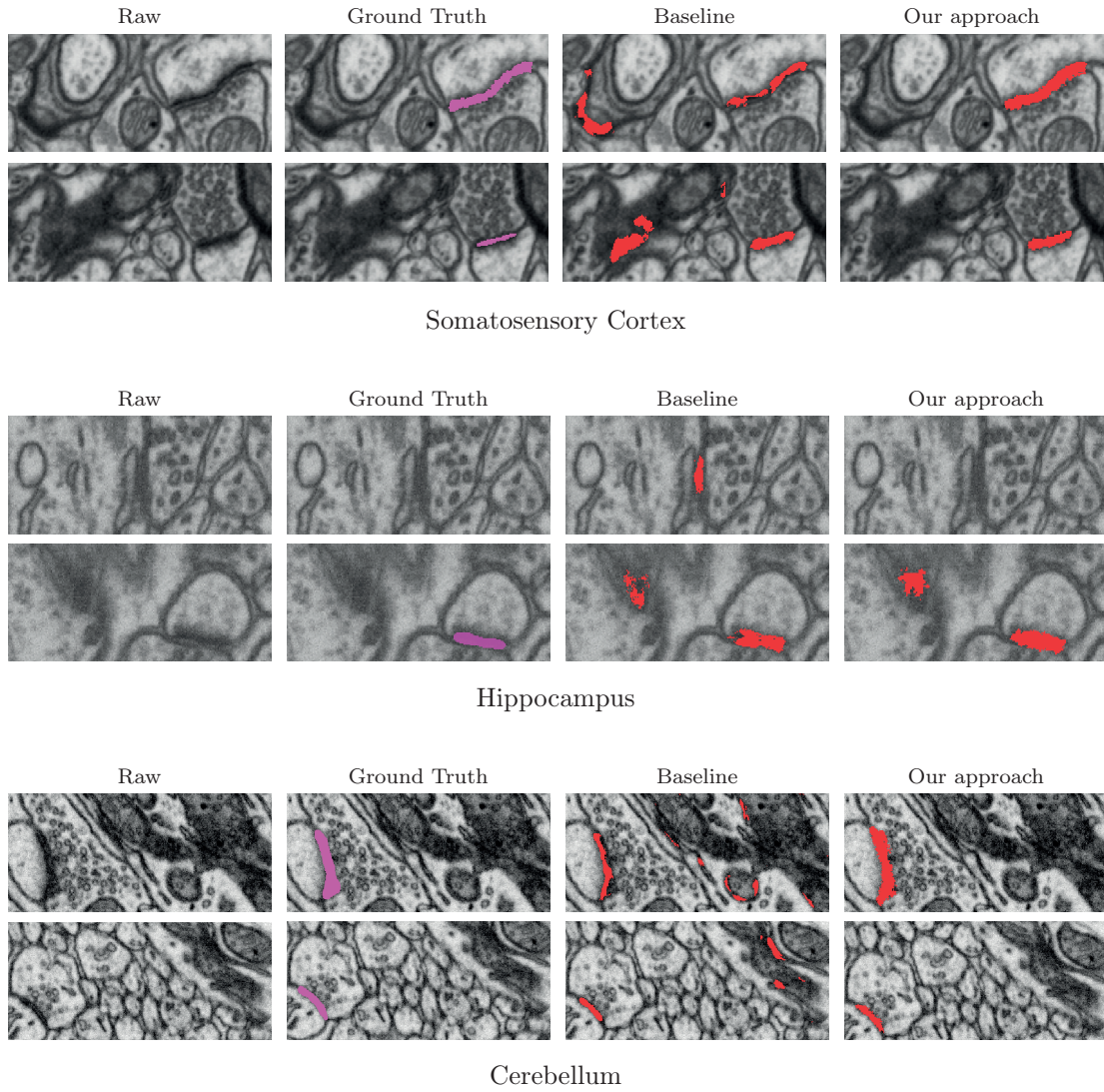


Figure 2.5: Qualitative results (slice cuts) for the three different datasets after thresholding. Threshold set at best VOC. Note that our approach yields more accurate segmentation results as well as reducing the amount of false positives.

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

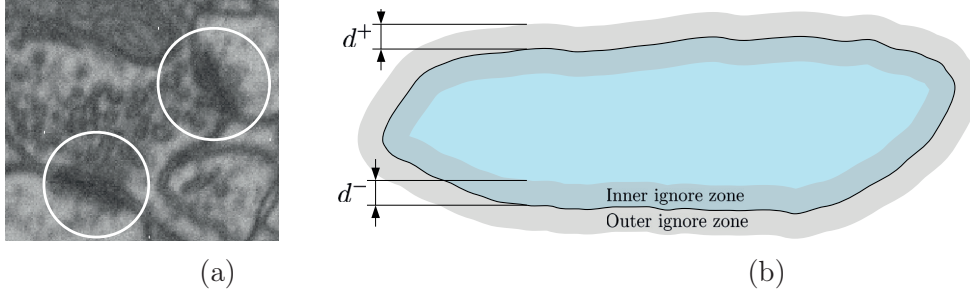


Figure 2.6: Illustration of the ambiguities at the synaptic boundary and evaluation ignore zone. (a) Slice cut of a region from the hippocampus dataset. Synapses indicated with circles. Labeling voxels close to the synaptic boundary is an ill-posed problem. (b) Diagram showing inner (d^-) and outer (d^+) exclusion zones on synthetic synapse ground truth (blue). Voxels within the exclusion zone are ignored during evaluation.

The amounts of training and test data for each dataset are summarized in Table 2.2. The volumes were annotated in a voxel-wise fashion using Fiji [93]. Note that testing volumes were fully annotated, each voxel being assigned a synapse or a non synapse label, in order to generate as large as possible a test set and report meaningful results. Training volumes on the other hand, in particular for the large datasets A and C, were only partially annotated in order to reduce labeling cost. In those cases, an approach similar to what was used in [57] was followed, labeling a fraction of the voxels inside the volume as positive or negative, leaving most of the voxels un-annotated and therefore not used for training. In the case of Dataset B, two test subvolumes were extracted from different regions of the Hippocampus.

2.5.1.1 Ground Truth and Gold Standard Annotations

It is important to highlight that annotating synapses is a difficult task and it is not uncommon to miss some of them. For example, [85] has shown that different proofreaders have an agreement of around 90% precision and recall when labeling T-bar synapses in fruit fly brain EM stacks.

Though in this thesis we use the term *Ground Truth* for consistency when referring to the expert annotations, *Gold Standard* may be a more appropriate term as it is likely to be imperfect.

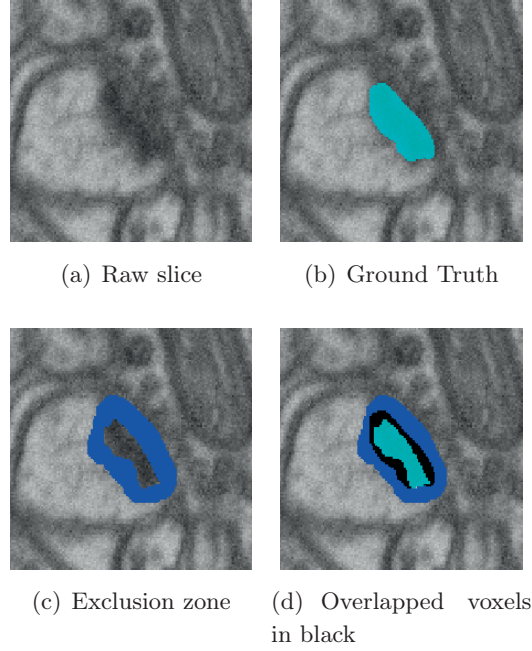


Figure 2.7: Example of exclusion zone with $d^+ = 4$ and $d^- = 1.6$ voxels. Overlap between ground truth and exclusion zone shown in black in (d). Note the labeling ambiguity in (b) for the voxels close to the vesicles.

2.5.1.2 Data Annotation

Synapse labeling is an ill-posed problem because boundaries are generally blurry, as shown in Fig. 2.6(a). For training purposes, we adopted a conservative labeling policy whereby voxels are labeled positive only if experts are highly confident. This yields a ground truth volume whose positive samples are mainly located at the center of synapses.

For testing purposes, the behavior of a method on ambiguous voxels may be of particular interest to the practitioner. Thus, we adopted a different procedure for the annotation of test volumes. Experts were given more freedom in deciding whether or not a specific voxel lies on a synaptic cleft while our evaluation procedure, explained next, was designed to study the behavior of our detector at synaptic boundaries.

2.5.1.3 Evaluation Methodology

Voxel-wise evaluation is essential to assess and compare the performance of different segmentation methods. However, such an evaluation must take into account the afore-

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

mentioned boundary issue to provide meaningful performance measures.

Much as we defined a training exclusion zone, we define a testing exclusion zone about the labeled border of the synapse with an exterior radius of d^+ and an interior radius of d^- , as depicted in Fig. 2.6(b). Voxels within the exclusion zone, shown in blue in Fig. 2.7(c), are ignored during evaluation.

Rather than arbitrarily fixing the values of d^+ and d^- , we propose to assess performance as a function of their values. We plot precision-recall (PR) curves at different exclusion zone sizes, and the value of the Jaccard Index [31, 66], also known as the VOC Score, as a function of d^+ and d^- . The VOC score measures the segmentation quality when ground-truth data is available. It is computed as the ratio of the area of the intersection between what has been segmented and the ground truth, and of the area of the union of these two regions.

To facilitate interpretation, we fix the ratio $\eta = \frac{d^+}{d^-}$ and plot the performance measure as a function of $d = d^+$. In our experiments we considered $0 \leq d \leq 5$ voxels and fixed $\eta = 2.5$ so that the maximum d^- is 2 voxels. Limiting d^- is essential to confine the exclusion zone to boundary voxels only, thus preserving most of the labeled synaptic voxels such that the evaluation remains meaningful. In practice, the exclusion zone is found by pre-computing a chamfer distance volume w.r.t. the synaptic boundaries in the ground truth.

2.5.2 Baselines

We evaluate our approach and compare it against the following baselines:

- **Kreshuk et al.:** the Random Forest method of [57] designed for synapse segmentation in isotropic stacks, that operates directly in 3D. This approach relies on voxel-wise classification, training a classifier that employs a set of pre-defined features evaluated at the voxel of interest.
- **U-Net CNN:** the 2D Convolutional Neural Network method of [88] designed to exploit large context information. To give this approach its best chance on our datasets, we choose the number of training iterations that maximize the Jaccard Index on the test set. We also explored extending the U-Net to 3D to leverage the data available in the EM stacks, but our results so far were unsatisfactory

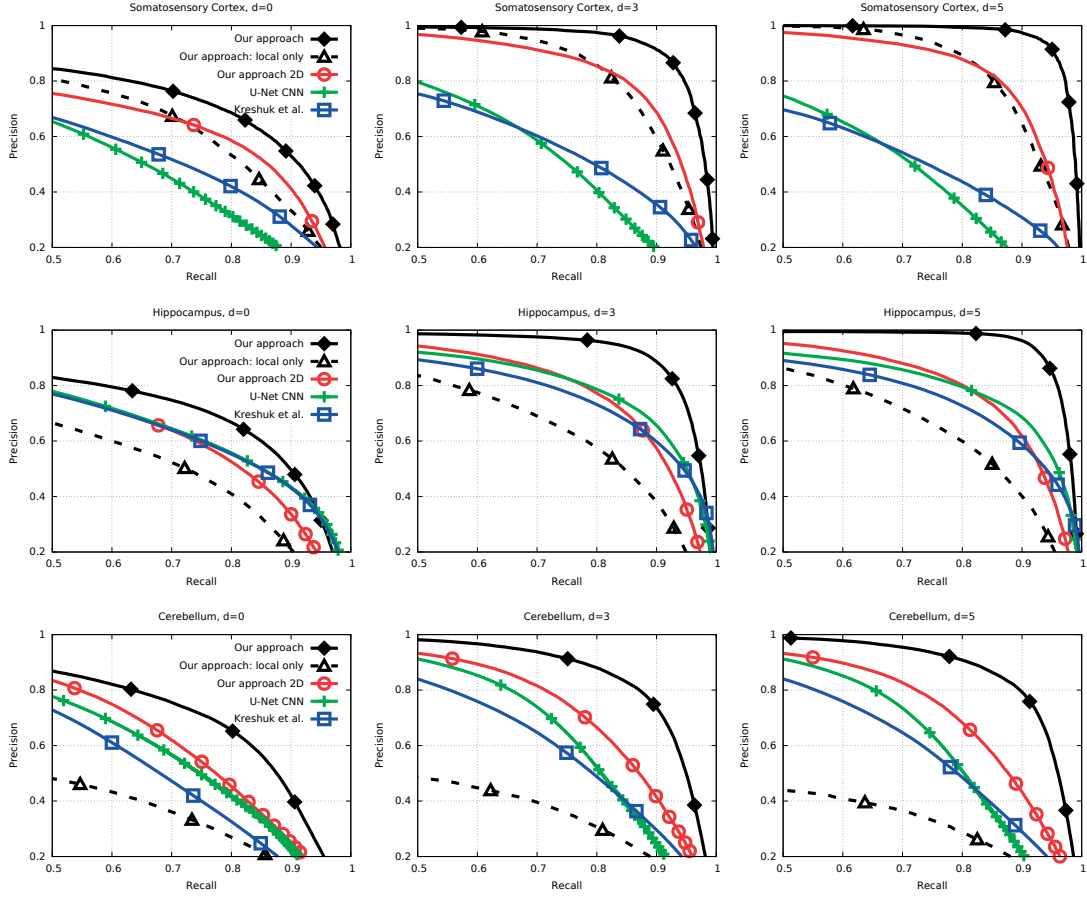


Figure 2.8: Precision-recall curves for each dataset for different values of d . Our approach always yields better performance than the baseline of [57].

and led to severe overfitting, obtaining significantly lower performance than the 2D counterpart.

For both methods we implemented our own version that runs on CPU for the first method, and GPU for the second.

2.5.3 Voxel-wise Accuracy

We evaluate the voxel-wise segmentation performance of our approach and compare it against the baselines in terms of precision-recall (PR) curves and Jaccard index values, also known as VOC scores [66]. For comparison, we also report the performance of our approach applied on individual 2D slices, called *Our Approach 2D*.

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

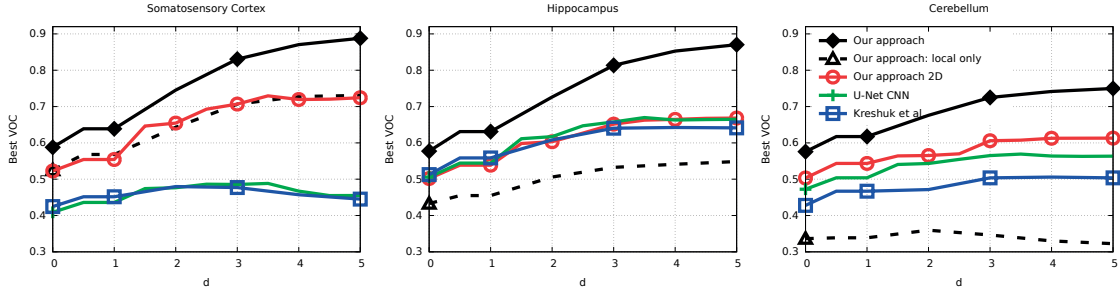


Figure 2.9: Highest Jaccard index (VOC score) as a function of exclusion zone size d for the different datasets. Our approach outperforms [57] for all values of d .

Precision-recall curves at different exclusion zone sizes d are shown in Fig. 2.8. Our approach clearly outperforms the baselines for all recall values as well as exclusion zone sizes d . We also show the value of the highest Jaccard index obtained at different values of d in Fig. 2.9, which manifests the same behavior. Overall, our results indicate a significant improvement in segmentation performance for both border and center voxels.

There is a clear significant performance drop when 3D information is discarded with *Our Approach 2D*, demonstrating the advantage of operating directly in 3D. Within the baselines that use 2D information exclusively, we observe that the *U-Net CNN* outperforms the 2D version of our approach only for the Hippocampus dataset, which corresponds to the only training stack that has been fully labeled. This corroborates the well-known fact that ANN need large amounts of training data to perform well, when training them from scratch.

Importance of contextual information To demonstrate that improved performance comes from using context, we also evaluate the performance of a degenerate version of our approach that relies only on local information and ignores context. To this end, we set the parameter $\|\mathbf{c}_p\|_{\max}$ of §2.4.2 to 0. This means that post-synaptic and pre-synaptic context is not exploited and only local statistics are leveraged. We plot the corresponding curves in Figs. 2.8 and 2.9, labeled as *Our approach: local only*. It can be seen that our full approach utilizing context does systematically and significantly better than the degenerate version which ignores context, highlighting the importance of contextual information for synapse segmentation.

We note that the relative performance between [57] and the degenerate version of our algorithm is somewhat variable. Both these methods ignore context and rely solely

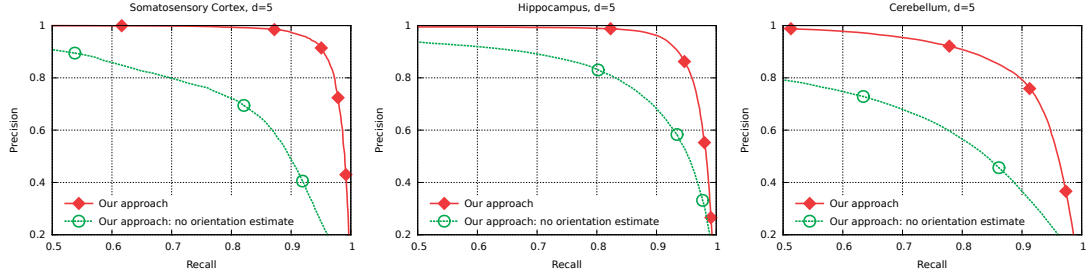


Figure 2.10: Comparison of our approach (diamonds) with a degenerate version of our method (circles) that uses a fixed orientation estimate for all voxels, where $\mathbf{R}(\ell_i)$ is set to identity matrix for all voxels ℓ_i . The latter results in a considerable performance loss, motivating the use of contextual cues that rotate to be consistent with the orientation estimate of each voxel ℓ_i .

on local image statistics. We believe the difference in performance between the two has to do with our degenerate version relying on decision stumps with a reduced feature set, while [57] relies on decision trees with a richer feature set.

Consistency of Contextual Cues Context cue locations are rotated according to the orientation estimate at each voxel of interest ℓ_i , which allows us to exploit synaptic context in a meaningful way. Therefore, it is interesting to observe the effects of setting the orientation estimate of all voxels to a fixed value. To do so, we have fixed $\mathbf{R}(\ell_i)$ to the identity matrix for all locations ℓ_i , which makes $\omega_3(\ell_i)$ point in the z direction. This is equivalent to our approach without pose-indexing, similar to the method proposed in [48, 98]. We call this particular implementation *Our approach: fixed orientation estimate*, and a comparison with our full method is presented in Fig. 2.10. As expected, performance drops significantly since context is relative to the orientation of the synaptic cleft, and fixing $\mathbf{R}(\ell_i)$ to an identity transformation yields inconsistent contextual cues, highlighting once again the importance of contextual information for synapse segmentation.

2.5.4 Selected Features

Our approach lets AdaBoost pick the most discriminative features for synapse segmentation at every boosting step. To observe how our method exploits context, we plot the pixel locations over which channel values $H_{m,\sigma_{n_m}}$ are summed in Fig. 2.11. To ease visualization, context cue locations \mathbf{c}_p are projected on the z_o axis (see Fig 2.2(a)).

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

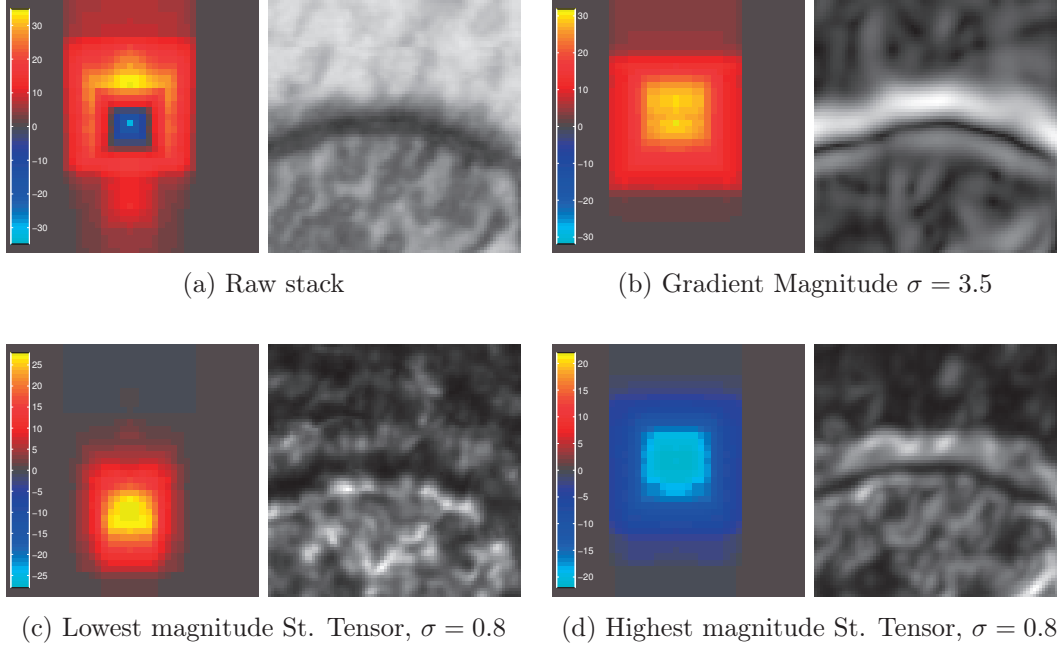


Figure 2.11: Some of the features selected by AdaBoost on four different channels for the Somatosensory Cortex dataset. The left column shows the voting map (see text for detailed description) and the right column shows an example synapse and its context.

When, according to AdaBoost, a higher value of $f_k(x, \ell_i)$ contributes to the voxel in the center being a synapse, its corresponding α_t (Eq. (2.6)) is added to the maps in Fig. 2.11. On the other hand, when the label is negatively correlated with $f_k(x, \ell_i)$, its α_t value is subtracted.

It can be seen that the raw image (Fig. 2.11(a)) provides important clues, particularly at the voxel of interest (center) but in its surroundings as well, especially at the post-synaptic region, where the classifier expects an average high image value to vote for a synapse. Another interesting channel is the lowest magnitude structure tensor, shown in Fig. 2.11(c), which signals the presence of vesicles in the pre-synaptic region, which is a strong clue used by experts to evidence the presence of a synapse.

2.5.5 Detection Accuracy

So far, we have evaluated the different approaches for voxel-wise segmentation. However, it is also interesting to evaluate synapse detection performance, that is if a synapse as a whole is detected by the algorithm or not. We measure detection performance by

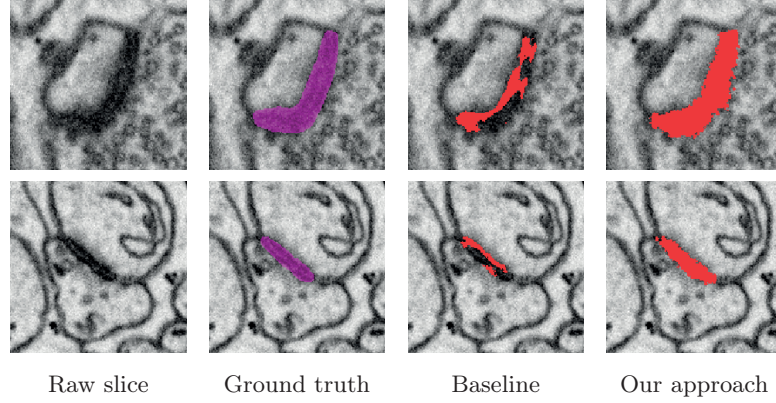


Figure 2.12: Examples of detected synaptic voxels after thresholding for the best Jaccard index. Synapses are detected by both methods, but the baseline method yields poor results from a segmentation perspective.

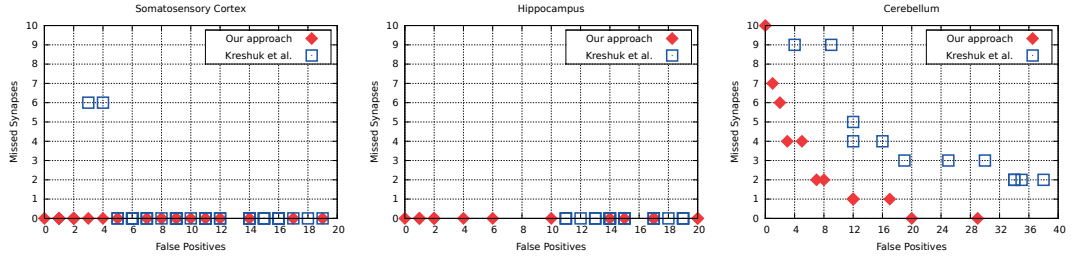


Figure 2.13: Detection performance on the three datasets, after thresholding and clustering voxels labeled as synapses by our approach and [57], at different thresholds. Our method yields less false-positive detections than [57], obtaining perfect detection performance in the first two datasets.

clustering thresholded score volumes. This can be summarized as:

1. Threshold the score volume at the value that yields the best VOC score for all d .
2. Run connected component analysis on the resulting binary volume.
3. Remove detected clusters with less than 1000 voxels, as in [57].
4. Count the number of missed and false-positive clusters/detections.

False-positive detections are clusters of positive-predicted voxels which do not intersect with the ground truth, while missed (false-negative) detections are ground truth clusters that do not intersect with the predicted score volume after step 3 above.

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

As shown in Fig. 2.13, our approach yields a significant reduction in the number of false positives when compared to the baseline, obtaining perfect detection on the first two datasets. Furthermore, because our method yields much better voxel-wise accuracy the shape of the recovered synapses is much closer to the ground-truth, as depicted in Fig. 2.12. This is important when computing biologically relevant statistics such as synapse shape and size, as discussed below.

2.5.6 Biological Statistics

We applied the classifier trained on the Somatosensory Cortex data to a large volume consisting of 1500x1125x750 voxels or $9.8 \times 7.4 \times 4.9 \mu\text{m}$. This is the original volume from which the train and test sub-volumes were extracted.

The resulting score volume was smoothed with a Gaussian filter with unit variance and thresholded at the value that corresponds to the maximum VOC score in the test volume. Afterwards, clusters of positive detections of less than 1000 voxels were discarded, as in [57] and §2.5.5. This resulted in a total of 405 clusters of voxels that our approach labeled as synapses. Finally, an expert went through the resulting segmentation volume, discarding 31 false positive synapses, obtaining a total of 374 verified synapses. This number is in agreement with the expected synapse density in the Somatosensory Cortex region (layer II) [27].

A 3D visualization of the detected synapses is shown in Fig. 2.15. It is interesting to observe the large variation in synapse shape and size, which is evidenced in the histograms of synapse size and synapse flatness Fig. 2.15(a) and Fig. 2.15(b).

Another interesting observation comes from Fig. 2.15(c), which is a scatter plot of synapse volume and flatness. There is a strong correlation between synapse volume and flatness. This occurs because synapses are membrane-like structures and the synaptic cleft width is constant across different synapses, independently of their size. Therefore, larger synapses are flatter than smaller ones, which is evidenced in Fig. 2.15(c).

In addition to generating segmentation results, our approach can also be used to determine the location of the post-synaptic and pre-synaptic regions. This can be highly relevant in practice to determine the location of the axons and dendrites relative to a given synapse, helping reveal neural circuit connectivity.

As discussed in §2.4.6, our technique evaluates two different scores for location ℓ_i at test time, one for each possible orientation polarity. The polarity with the highest score

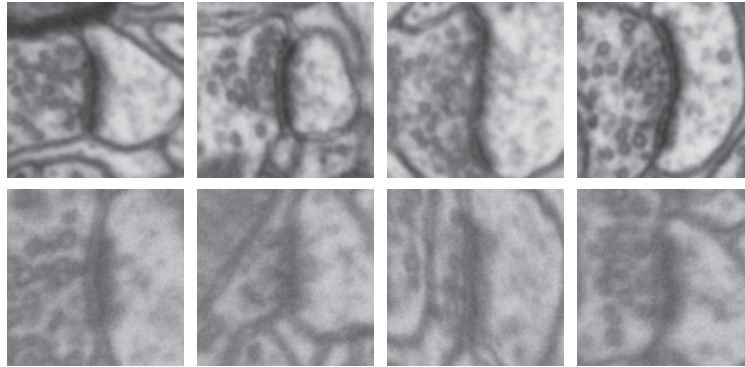


Figure 2.14: Examples of polarity prediction. Images have been aligned such that the horizontal axis travels from the pre- to post-synaptic region from left to right. Our approach achieved 100% accuracy for this task.

can be employed as a polarity estimate. Once the score image is thresholded, majority voting can be used to determine the most likely polarity.

We applied this technique to the predicted scores for the three datasets and observed 100% accuracy at predicting synapse polarity. Examples are shown in Fig. 2.14, where the synapse normal has been aligned to the horizontal axis. This is also an indication that our approach is exploiting context information.

2.5.7 Computational Complexity

In terms of computational cost, our approach is faster in both training and testing than [57] and similar to that of the U-Net [88]. However, note that the U-Net operates in 2D, while our approach and that of [57] work directly in 3D.

The training time for our approach benefits from the techniques mentioned in §2.4.5. Using stumps as weak learners is also an important advantage at test time, in contrast to Random Forest’s trees which are built with as many splits as necessary to separate training data perfectly. Another speed up is obtained by using supervoxels to over-segment the EM volumes. The chosen seed size of 2 voxels translates in an average supervoxel size of $2 \times 2 \times 2 = 2^3$ voxels [1], which yields a 8x speed factor since only supervoxel centers are classified.

Table 2.3 summarizes timings obtained for both methods. The number of trees in the Random Forest classifier was set to 500. To make comparison fair, we have modified the Random Forest implementation used by Kreshuk et al. [57] to use multiple threads,

2. CONTEXT CUES FOR SYNAPSE SEGMENTATION

Table 2.3: Train and test times for our approach and baselines. Timings obtained with multithreaded implementations for our approach and Kreshuk et al. [57], and on a GPU for the U-Net [88].

Dataset		Our approach		Kreshuk et al. [57]		U-Net [88]
		Total ($T = 2000$)	Single stump (average)	Total (500 trees)	Single tree (average)	Total
Somatosensory Cortex	Train	0:57 hs	1.69 sec	2:16 hs	16.4 sec	1:48 hs
	Test	0:13 hs	6.2 msec	0:21 hs	2.5 sec	0:09 hs
Hippocampus	Train	2:40 hs	4.8 sec	21:22 hs	154 sec	5:45 hs
	Test	1:20 hs	2.4 sec	1:46 hs	12.8 sec	0:14 hs
Cerebellum	Train	1:41 hs	3.05 sec	10:31 hs	75.7 sec	3:03 hs
	Test	1:35 hs	2.8 sec	2:47 hs	20.1 sec	0:19 hs

speeding up training and testing substantially¹. Note that, in the case of using the default Vigna implementation, the timings for the baseline method in Table 2.3 would be an order of magnitude higher.

The parameters used for our approach are the default ones described in Table 2.1. Note that the test timings for our approach already consider evaluating the two possible polarities for each location ℓ_i .

If further speed up was needed, soft cascades [17] could be employed to stop early during the evaluation of the boosted classifier. This is likely to provide a considerable speed up since most background voxels can be discarded with a simple intensity check, given that synapses appear as dark structures.

2.6 Conclusion

In this chapter we presented a novel approach to synapse segmentation. It relies on a large set of image features, specifically designed to take spatial context into account, which are selected, weighed and combined using AdaBoost. We used three different EM datasets to demonstrate that our algorithm effectively distinguishes true synapses from other organelles that exhibit the same local texture.

Moreover, our method also flawlessly identifies synaptic orientation, a key and hitherto unexplored task that could be exploited to assess other measures such as synapse

¹The original version is found in the Vigna library (<http://hci.iwr.uni-heidelberg.de/vigna/>) and our version can be found online at <http://cvlab.epfl.ch/%7Ecjbecker>

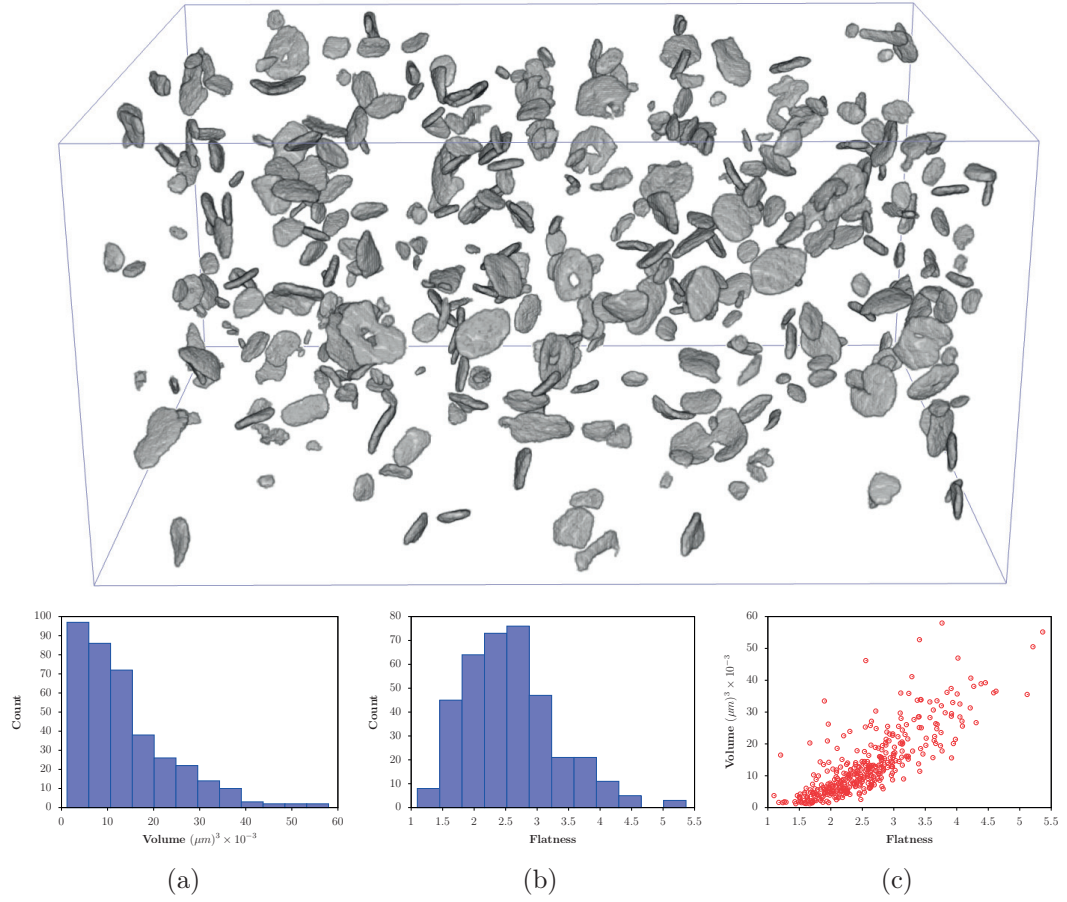


Figure 2.15: Results of our approach on a large Somatosensory Cortex volume of 1500x1125x750 voxels: (top) 3D Visualization of the detected synapses and (bottom) some of the statistics than can be extracted from voxel-wise segmentation. The correlation between synapse volume and flatness in (c) evidences the fact that synapses are membrane-like objects.

convexity and concavity.

AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MULTIPLE INSTANCE LEARNING

The segmentation approach presented in the previous chapter requires that the polarity of every synapse is manually labeled for training. In this chapter we aim at removing this requirement, as it incurs into an extra burden for the annotator. Moreover, mislabeling polarities may have a negative impact on prediction performance, as we show in our experiments.

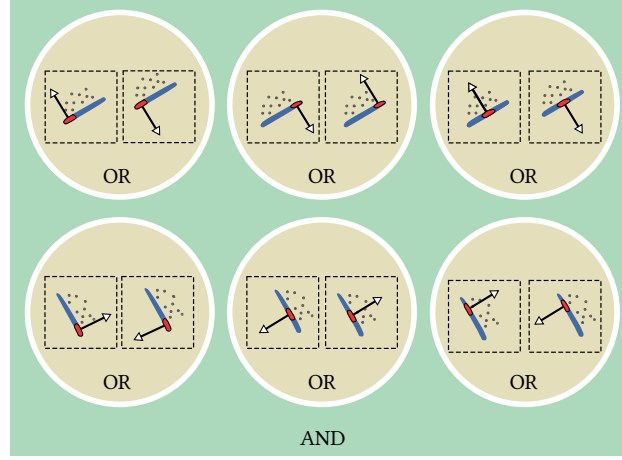
Our solution is based on Multiple Instance Learning (MIL)[5, 105] techniques and operates on clusters of training instances. Our approach results in segmentations that are as good or better than using manually-labeled polarities, and is able to flawlessly estimate synapse polarity in the training data.

3.1 Introduction

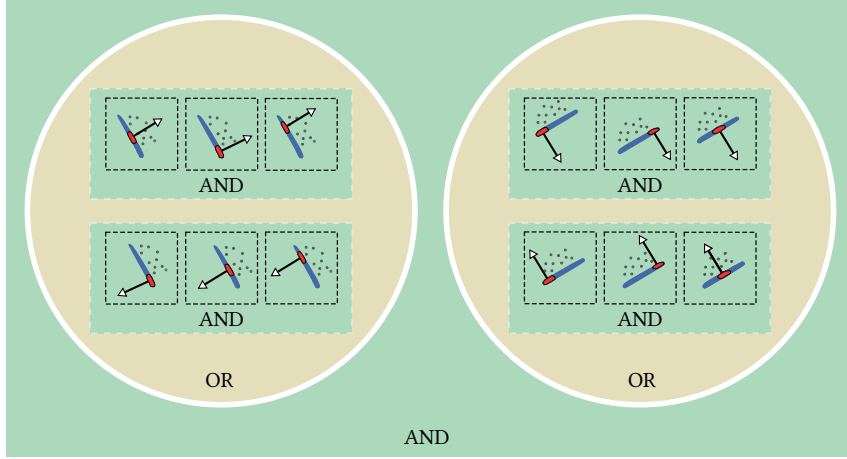
Multiple Instance Learning can be categorized as a weakly-supervised learning approach [111]. The main idea behind MIL is that there exists latent information in the training data that can be exploited to reduce annotation requirements and improve learning. Typically, MIL assigns latent variables to each training instance, whose values are to be inferred during learning. For example, in the synapse segmentation scenario of Chapter 2, there would be a binary latent value per training sample, associated to its orientation polarity.

Though various MIL method variants have been introduced and investigated [5, 8, 29, 38, 105, 110, 111], to our knowledge there is no MIL technique that can impose the

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL



(a) Classic MIL.



(b) Our approach with CMIL.

Figure 3.1: Illustration of classic MIL and our proposed CMIL in the case of synapse polarity estimation. A supervoxel or sample is illustrated in red, and its assumed orientation with an arrow. In classic MIL, each sample is associated a binary latent variable that determines its polarity. In contrast, our approach clusters samples that belong to the same synapse, constraining supervoxels of each synapse to share the same polarity, and reducing the number of latent variables at training time.

hard constraint to force nearby training samples share a common latent variable. For synapse segmentation this is essential to ensure that the samples that belong to a given synapse share the same polarity, as illustrated in Fig. 3.1.

3.2 Related Work

Multiple Instance Learning dates back to 1997 [28]. Since then, MIL has been incorporated into a variety of learning methods including Support Vector Machines (SVMs) [5, 59, 69], Random Forests [63, 102] and Boosting [8, 105, 110, 111]. It has as well been applied to a wide range of problems, including object detection [8, 105], object tracking [8, 63, 113] and image segmentation [82, 83, 84, 102, 103, 110, 111].

In computer vision, one of the main applications of MIL is to reduce annotation effort in supervised learning. This achieved by introducing latent variables whose values are estimated as training proceeds. A known example is that of [105], that employs a boosting-based MIL approach to improve object detection results when the bounding boxes in training data are not accurately aligned. Similarly, [8] introduces a MIL framework to image alignment that splits data into groups and trains a classifier for each group, improving prediction performance.

MIL has also been applied to image segmentation. For example, [102] combines MIL and Multi-Task Learning for semantic segmentation, dropping the need for expensive pixel-wise labeling, and requiring only tag-based annotations per image. Structure models can also benefit from MIL, as shown in [103], where a pairwise CRF is learned to infer latent superpixel labels at training time.

Biomedical imaging has also profited from MIL [29, 38, 110, 111]. An impressive result is that of [109], where MIL is applied to simultaneously segment and cluster pixel regions in histopathology cancer images. Their approach only requires per-patch weak labels and is able to automatically segment and cluster image regions. In a later publication [110], the same authors extend their framework to incorporate smoothness constraints at the pixel level.

Recent work with Convolutional Neural Networks (CNNs) to reduce the amount of supervision has also gained from MIL [82, 83, 84]. [83, 84] apply CNNs for image segmentation, using MIL to leverage image-level annotations, such as bounding boxes or image tags. To avoid overfitting, they both rely on pre-trained CNNs to regularize learning. On the other hand, [82] uses Expectation-Maximization methods to learn the CNN parameters with mixed image-level and pixel-level annotations, bypassing pre-training.

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL

Though MIL has been largely explored in the literature, it has been mostly applied to aggregate instance-level information in the form of Fig. 3.1(a). To our knowledge, there is no current MIL approach to deal with problems such as that of Fig. 3.1(b) that combine AND and OR operations at different semantic levels. The closest to our work is that of [110], though in the latter the problem is a very different one, namely that of automated image region clustering, and therefore not compatible with that of Fig. 3.1(b).

In the remaining of this chapter we introduce the mathematical formulation of MIL and detail our approach to solving Fig. 3.1(b). We then present results on the three datasets introduced in the previous chapter and show that our method can flawlessly estimate synapse polarity.

3.3 Classic Multiple Instance Learning

In this section we introduce and define the basics of Multiple Instance Learning, with a special emphasis on the boosting-based variants, as our goal is to incorporate MIL into the approach of Chapter 2. We then show how classic MIL can be applied to synapse segmentation and discuss its limitations.

3.3.1 Classic MIL Model and Formulation

In MIL, data is typically grouped into *bags*, where each bag contains one or more training instances. In contrast to classic supervised learning that requires one label per instance, MIL relaxes this constraint, requiring only one label per bag, hence reducing the data annotation effort.

For example, in the synapse segmentation scenario described in the previous section, we can think of each positive bag as containing two instances of opposite polarity. We know that one of them is the correct one, but it is up to the training algorithm to determine which one, so as to minimize the overall training loss. This happens at the same time as the discriminative function $f(\mathbf{x})$ is learned.

Assume we have a set of N bags, b_1, b_2, \dots, b_N . Bag i contains $|b_i|$ instances, and is assigned a label $y_i \in \{-1, +1\}$. The instances in bag i are in the set $b_i = \{\mathbf{x}_{ij}\}_{j=1, \dots, |b_i|}$, and $y_{ij} \in \{-1, +1\}$ is the latent label of sample \mathbf{x}_{ij} .

3.3 Classic Multiple Instance Learning

	$g(v_1, \dots, v_m)$	Domain
Log-Sum Exponential (LSE)	$\frac{1}{r} \log \left[\frac{1}{m} \sum_{\ell} \exp(r v_{\ell}) \right]$	$[-\infty, \infty]$
Generalized Mean (GM)	$\left(\frac{1}{m} \sum_{\ell} v_{\ell}^r \right)^{\frac{1}{r}}$	$[0, \infty]$
Noisy-Or (NOR)	$1 - \prod_{\ell} (1 - v_{\ell})$	$[0, 1]$

Table 3.1: Examples of commonly-used soft-max functions used in Multiple Instance Learning [8] that approximate $\max(v_1, \dots, v_m)$.

We write the probability of instance \mathbf{x}_{ij} being positive as

$$p(y_{ij} = 1 | \mathbf{x}_{ij}) = \sigma[f(\mathbf{x}_{ij})] = \frac{1}{1 + e^{-f(\mathbf{x}_{ij})}}, \quad (3.1)$$

where $f(\mathbf{x})$ is the scoring function we seek to learn.

For classic MIL, a bag is positive if at least one sample of the bag is classified as positive, expressed as

$$p(b_i = 1) = \max_{\mathbf{x} \in b_i} p(y = 1 | \mathbf{x}), \quad (3.2)$$

which can also be seen as an *OR* operation if the $p(y = 1 | \mathbf{x})$ were binary variables.

One difficulty with the max function is that it is not differentiable, making optimization difficult. Therefore, in the MIL setting it is commonly replaced by a differentiable soft-max approximation $g(\cdot)$, so that

$$p(b_i = 1) = \max_{\mathbf{x} \in b_i} p(y = 1 | \mathbf{x}) \approx g(b_i). \quad (3.3)$$

A summary of commonly-used soft-max functions is shown in Table 3.1.

The goal of MIL is to find a scoring function $f(\mathbf{x})$ that minimizes the negative log-likelihood of the labeled bags in the training data,

$$L = -\log \prod_{i=1}^N p(b_i = y_i) \quad (3.4)$$

$$= -\sum_{i=1}^N \log p(b_i = y_i), \quad (3.5)$$

where the product of the probabilities $p(b_i = y_i)$ can be interpreted as an *AND* operation, implying that all samples should be classified correctly by $f(\mathbf{x})$.

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL

Boosting-based MIL approaches apply gradient boosting to learn $f(\mathbf{x})$ and minimize the loss above. Note that the latent variables in MIL are not explicit, but rather implicitly coded by the max function that computes $p(b_i = y_i)$.

3.3.2 Applying MIL to Synapse Segmentation

We can apply the MIL formulation described above to synapse segmentation, where each positive bag contains two samples that are exactly equal except for their polarity. Negative bags, on the other hand, only contain one sample, as they should be classified as negative regardless of their polarity.

At training time we use MIL to resolve the ambiguous polarity of the positive samples. For prediction, we test both possible polarities for every voxel, and return the one with the highest score, as in Chapter 2.

The resulting MIL problem structure can be expressed graphically, as shown in Fig. 3.1(a), where the *OR* blocks represent the soft-max function and the *AND* blocks the product in Eq. (3.4). For simplicity we only show six training bags coming from two synapses. An important limitation of the classic MIL formulation for this particular problem is that it does not guarantee consistent polarity estimates within the same synapse. Simultaneously, it makes learning harder, as it creates as many latent binary variables as the number of positive examples.

In the next section we propose an approach to overcome these limitations. In contrast to classic MIL, it requires only one latent variable per synapse, ensuring consistent polarity estimates.

3.4 Proposed Approach

In this section we introduce a novel MIL approach that ensures that the polarity of the training samples within a synapse is consistent. We do this by clustering training samples according to the synapse they belong to, introducing an additional layer in the MIL formulation. This new approach is called Clustered MIL (CMIL), and is illustrated in Fig. 3.1(b).

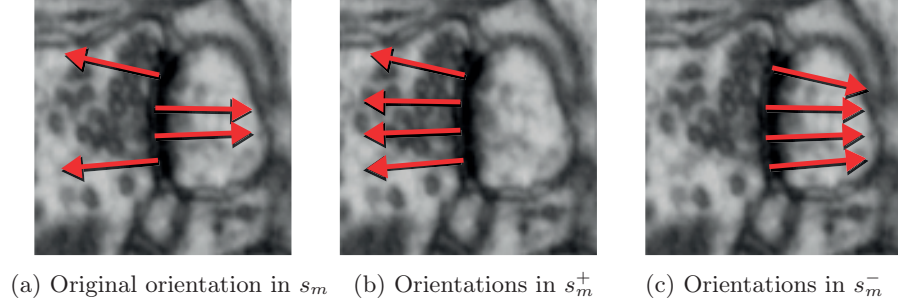


Figure 3.2: Generation of consistent synapse polarities. The eigenvector of the highest magnitude eigenvalue may not be consistent in polarity, as shown in (a). As a pre-processing step we create consistent hypothesis (b) and (c).

3.4.1 Clustering Instances and Orientations

At training time we cluster positive training instances with Connected Components into M synapses s_1, \dots, s_M . The instances in synapse m are inside the set $s_m = \{\mathbf{x}_{mj}\}_{j=1, \dots, |s_m|}$, and we denote the eigenvector with the highest magnitude eigenvalue of the Hessian of sample \mathbf{x}_{mj} as $\mathbf{n}_{mj} \in \mathbb{R}^3$.

For each synapse s_m we find the main Hessian direction $\bar{\mathbf{n}}_m$, such that the squared projected value of the orientations \mathbf{n}_{mj} on $\bar{\mathbf{n}}_m$ is maximized,

$$\bar{\mathbf{n}}_m = \arg \max_{\substack{\mathbf{v} \in \mathbb{R}^3 \\ \|\mathbf{v}\|=1}} \sum_{j=1}^{|s_m|} (\mathbf{v}^T \mathbf{n}_{mj})^2. \quad (3.6)$$

To solve for $\bar{\mathbf{n}}_m$ we use the SVD decomposition of the matrix $N_m \in \mathbb{R}^{3 \times |s_m|}$, whose columns are the orientation vectors \mathbf{n}_{mj} .

We then duplicate the training data to generate the sets s_m^+ and s_m^- with consistent polarity. The polarity of each sample in s_m^+ is modified so that it yields a positive projection on $\bar{\mathbf{n}}_m$. On the other hand, s_m^- contains the same samples as s_m^+ but with the opposite polarity. An illustration is shown in Fig. 3.2.

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL

3.4.2 CMIL Formulation

The probability of synapse m for either polarity is written as

$$p(s_m^+) = \prod_{\mathbf{x} \in s_m^+} \frac{1}{1 + e^{-f(\mathbf{x})}}, \quad (3.7)$$

$$p(s_m^-) = \prod_{\mathbf{x} \in s_m^-} \frac{1}{1 + e^{-f(\mathbf{x})}}, \quad (3.8)$$

and the probability of the respective bag containing both polarities is computed with the soft-max function $g(\cdot)$ as,

$$p(b_m) = g\left(p(s_m^+), p(s_m^-)\right) \quad (3.9)$$

$$= \left(\frac{1}{2} p(s_m^+)^r + \frac{1}{2} p(s_m^-)^r \right)^{\frac{1}{r}}, \quad (3.10)$$

where we used the Generalized Mean soft-max approximation, as in [8, 111].

The overall training loss is then written as

$$L = -\log \prod_{m=1}^M p(b_m) - \log \prod_{n=1}^{N^-} p(y = -1|x_i) \quad (3.11)$$

$$= -\sum_{m=1}^M \log p(b_m) + \sum_{n=1}^{N^-} \log \left(1 + e^{f(\mathbf{x}_i)} \right), \quad (3.12)$$

where the first term is the loss for the positive samples (synapses), while the second term is the log loss for the negatives. Note that both terms are consistent, as when there is only one element per bag we get $g(v_1) = v_1$ for all the soft-max functions introduced in Table 3.1.

Boosting can be applied to minimize Eq. 3.12. The resulting boosting weights are,

$$w_{mj}^+ = \frac{\partial L}{\partial f(\mathbf{x}_{mj}^+)} = \frac{p(s_m^+)^r}{p(s_m^+)^r + p(s_m^-)^r} (1 - p(\mathbf{x}_{mj}^+)) \quad (3.13)$$

$$w_{mj}^- = \frac{\partial L}{\partial f(\mathbf{x}_{mj}^-)} = \frac{p(s_m^-)^r}{p(s_m^+)^r + p(s_m^-)^r} (1 - p(\mathbf{x}_{mj}^-)) \quad (3.14)$$

$$w_n = \frac{\partial L}{\partial f(\mathbf{x}_n)} = 1 - p(\mathbf{x}_n) \quad (3.15)$$

It is important to take care of possible numerical instabilities, as $p(s_m^+)$ and $p(s_m^-)$ may be very small, particularly with large synapses.

3.4.3 Numerical Stability

Instead of computing $p(s_m^+)$ and $p(s_m^-)$ directly with Eq. 3.8, we compute its log-probability as well as a normalization factor to avoid underflows. More specifically, we can write

$$\log p(s_m^+) = - \sum_{\mathbf{x} \in s_m^+} \log \left(1 + e^{-\varphi(\mathbf{x})} \right), \quad (3.16)$$

and its analogous for $\log p(s_m^-)$. The normalization factor is then computed as $\beta_m = \max(\log p(s_m^+), \log p(s_m^-))$, and stores the log likelihood of the most likely polarity of synapse m . We then evaluate $\kappa_m^+ = \log p(s_m^+) - \beta_m$ and $\kappa_m^- = \log p(s_m^-) - \beta_m$, that allows us to compute a more stable version of the weights for the positive samples as

$$w_{mj}^+ = \frac{\exp(r \kappa_m^+)}{\exp(r \kappa_m^+) + \exp(r \kappa_m^-)} (1 - p(\mathbf{x}_{mj}^+)) \quad (3.17)$$

$$w_{mj}^- = \frac{\exp(r \kappa_m^-)}{\exp(r \kappa_m^+) + \exp(r \kappa_m^-)} (1 - p(\mathbf{x}_{mj}^-)). \quad (3.18)$$

This avoids low probabilities $p(s_m^+)$ from propagating numerical instabilities to the boosting weights. This same technique is applied during the linesearch step.

3.5 Evaluation

In this section we compare the segmentation performance of our approach against different baselines, on the datasets introduced in the previous chapter, with the same train and test configurations shown in Table 2.2.

3.5.1 Baselines

We compare our MIL approach, denoted *Clustered MIL*, against three baselines:

- *Classic MIL*: the classic MIL formulation applied to our problem, as illustrated in Fig. 3.1(a), which does not enforce the polarities of neighboring voxels to be the same.
- *Labeled polarity*: the synapse segmentation method presented in the previous chapter, with polarities manually labeled by the user. This is the strongest baseline and a best-case scenario.

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL

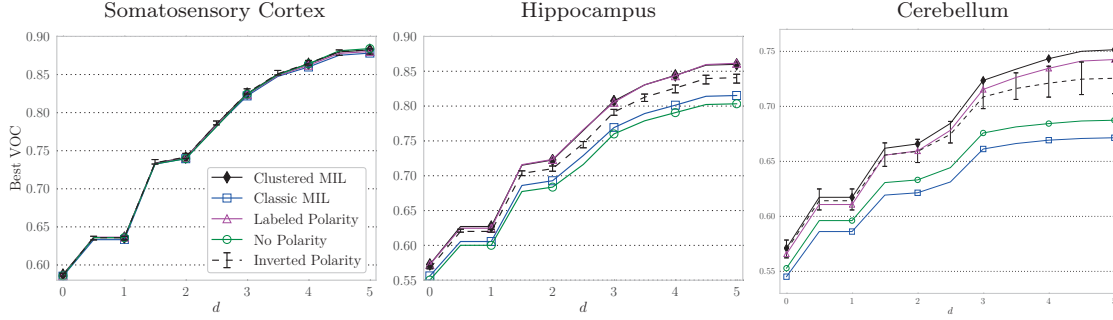


Figure 3.3: Highest Jaccard index (VOC score) as a function of exclusion zone size d for the different datasets. Our Clustered MIL approach performs as well or better than the strongest baselines for all values of d .

- *No polarity*: the same approach as above, but polarities are not labeled. It assumes that the polarity of the eigenvector of highest eigenvalue magnitude of the Hessian matrix is correct.
- *Inverted polarity*: polarities are assigned manually to each synapse, as with *Labeled polarity*, but we simulate a mislabeled polarity by inverting the polarity of one synapse, and keeping the rest untouched. For this baseline we run 10 different experiments, mislabeling the polarity of a different synapse in each run. We report the median and first and third quartile intervals in the respective plots.

3.5.2 Quantitative Results

Figure 3.3 shows the highest Jaccard index obtained by each method on the three datasets introduced in the previous chapter. It can be seen that there is a significant difference in the behavior of the baselines between the Hippocampus and Cerebellum datasets and the Somatosensory Cortex one. We first focus on the first two, and then discuss the latter.

In the Hippocampus and Cerebellum datasets, *No Polarity* and *Classic MIL* perform poorly, resulting in a drop of 5% to 8% in Jaccard Index than in comparison to *Labeled Polarity*. Similarly, mislabeling polarities typically hurts performance, missing the full potential of our segmentation approach. This is expected, as the classifier may struggle to learn the right context features for segmentation.

On the other hand, our MIL approach consistently recovers this performance drop, sometimes outperforming *Labeled Polarity*. We think the latter may happen because,

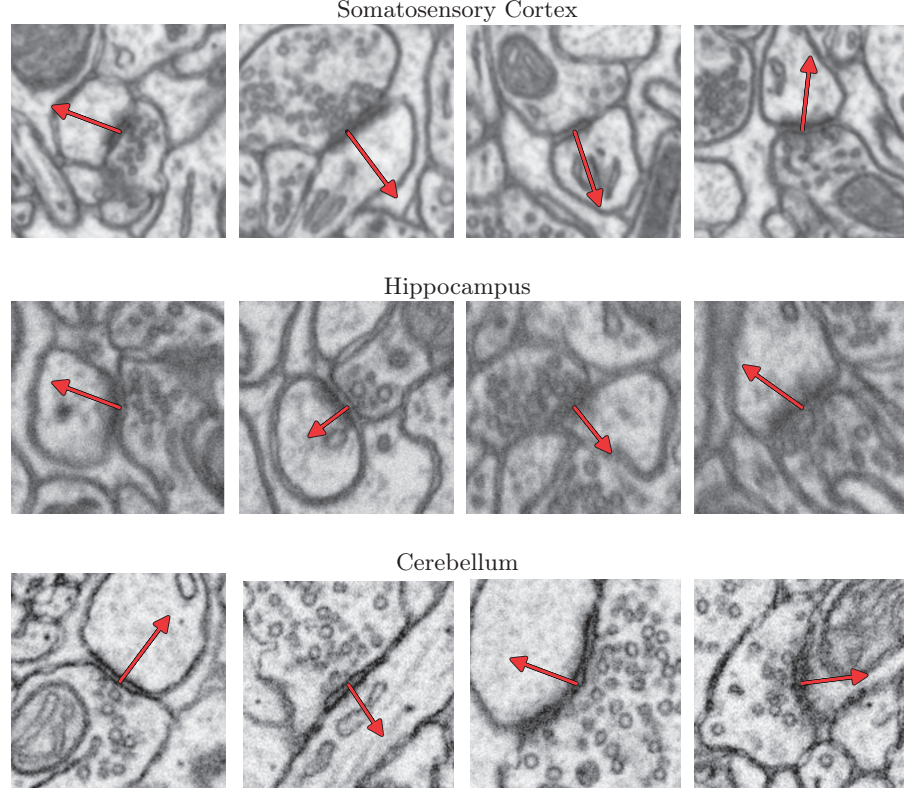


Figure 3.4: Polarity estimates computed by our MIL approach on the training data. Our method flawlessly estimates synapse polarity for all the synapses in the training set, including those that may be confusing due to organelles such as mitochondria on either side of the synaptic cleft.

as boosting proceeds, lower weight is given to samples whose polarity is not easily discriminated by MIL, which could then reduce the risk of overfitting to difficult training samples. The weight reduction effect can be seen in Eq. 3.18 if $p(s_m^-) = p(s_m^+)$.

With respect to the Somatosensory Cortex dataset, all methods do equally well, suggesting that the test set is too small or that tissue staining makes synapses easy to distinguish with little local context, and therefore much invariant to polarity estimation. This may be evidenced by looking at Fig 2.9, where *Our approach local only* performs reasonably well. Nonetheless, our MIL approach performs as well as the other baselines, and is able to estimate correctly synapse polarities, as shown next.

3. AUTOMATIC POLARITY ESTIMATION WITH CLUSTERED MIL

3.5.3 Qualitative Results

Given that our approach assigns a polarity for each synapse or group of connected positive-labeled voxels in the image, we can easily verify if the assigned polarities agree after training. Figure 3.4 shows the polarities assigned by our method to different synapses in the training data.

Furthermore, we verified through manual inspection that our approach assigns the correct polarity to all the synapses present in the training data of each dataset.

3.6 Conclusion

In this chapter we presented a new approach for Multiple Instance Learning to automatically estimate orientation polarity for synapse segmentation. In contrast to existing methods, our approach operates on clusters of training instances, ensuring that the orientation polarity of the training samples within the same synapse is consistent.

Our approach reduces annotation effort by dropping the requirement to manually assign a polarity to each synapse. Our method performs as well or better than using manually-labeled polarities, demonstrating its effectiveness. Moreover, we also verified that our approach flawlessly estimates the polarities of all the synapses in the training set.

Part II

Domain Adaptation

DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

In this chapter we present a domain adaptation algorithm that effectively leverages labeled examples across different acquisitions, significantly reducing annotation requirements. Our approach can handle complex, non-linear image feature transformations and scales to large microscopy datasets that often involve high-dimensional feature spaces and large 3D data volumes. We evaluate our approach on four challenging Electron and Light Microscopy applications that exhibit very different image modalities and where annotation is very costly. Across all applications we achieve a significant improvement over the state-of-the-art machine learning methods and demonstrate our ability to greatly reduce human annotation effort.

4.1 Introduction

Imaging modalities such as Electron (EM) and Light Microscopy (LM) can now deliver high-quality, high-resolution image stacks of neural structures, such as the ones depicted by Fig. 4.1. Typically, a combination of manual and semi-automated segmentation or annotation tools such as [77, 92, 95] are then used to extract structures of interest. However, while the ever growing amount of available imagery should help unlock the secrets of neural function, the required amounts of human annotation effort remain a major bottleneck.

While machine learning-based approaches have shown great potential, such algorithms still require significant amounts of manual annotation to train classifiers that can generalize well to previously unseen data. In microscopy, this can be a problem

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

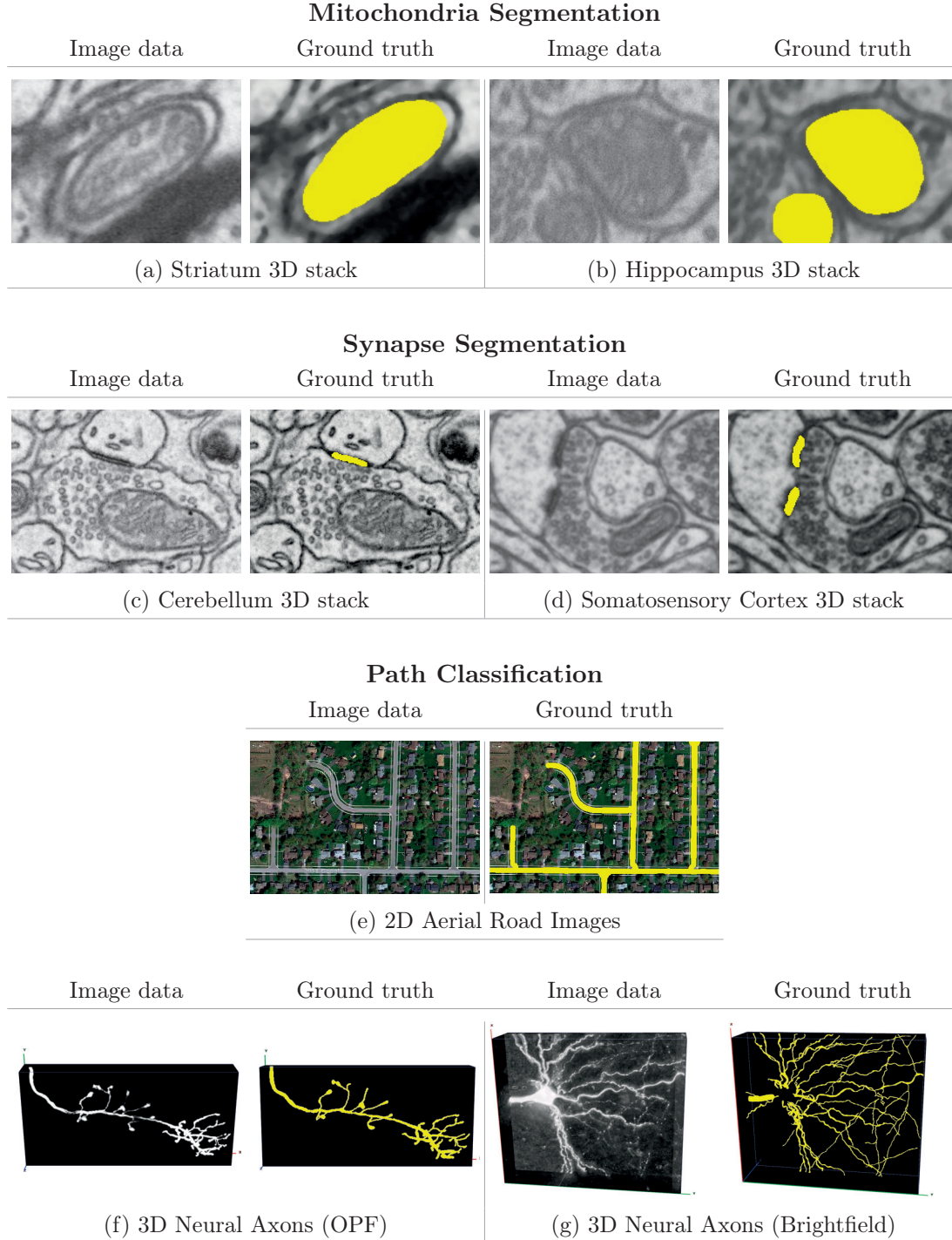


Figure 4.1: Segmentation and path classification applications we consider: (a,b,c,d) slice cuts from four 3D Electron Microscopy acquisitions from different parts of the brain of a rat. Each 3D stack contains millions of voxels to be classified. (e,f,g) 2D aerial road images and 3D neural axons from Olfactory Projection Fibers (OPF) and Brightfield microscopy. Ground truth positive samples shown in yellow. Best viewed in color.

because the data preparation processes tend to be complicated and not easily repeatable, which means that a classifier trained on one acquisition will not perform very well on a new one, even when using the same modality. This is because machine learning normally relies on the fact that the training and run-time data samples are drawn from the same distribution.

For example, acquiring the Electron Microscopy (EM) images of brain structures shown in the top two rows of Fig. 4.1 requires tissue staining to increase contrast, followed by resin encasing before the acquisition. As a result, two samples of the same brain region acquired at different times may look significantly different due to differences in their preparation. This is even more true when the samples come from different parts of the brain, so that classifiers trained for one of them perform poorly on the other. While it is theoretically possible to gather new training data after each new image acquisition, it is impractical if high-throughput is desired because manual labeling of 3D image stacks is incredibly time-consuming.

A practical solution is to use domain adaptation [49] and acquire sufficient amounts of training data after *one specific* image acquisition and then to use it in conjunction with a small amount of additional training data that can be acquired quickly after each subsequent one to retrain the classifiers. Following the terminology of domain adaptation, we refer to the acquisition with sufficient training data as the *source domain* and the one with limited supervision as the *target domain*. Our goal is then to exploit the labeled data in the source domain to learn an accurate classifier in the target domain despite having only a few labeled samples in the latter. While domain adaptation has received significant attention in the machine learning and computer vision communities, to our knowledge it has only recently been gaining interest in Medical Imaging, and remains largely unexplored for the acquisition problem depicted by Fig. 4.1. For many bio-medical applications, such as the ones considered in this work, we believe it is greatly needed to reduce annotation effort and make machine learning algorithms of practical use.

Current approaches to domain adaptation, and more generally *Transfer* or *Multi-Task* Learning [3, 21, 23, 32], treat classification in each domain as separate but related problems and exploit their relationship to learn from the supervised data available across all of them. Multi-task learning methods typically assume that the decision boundaries in each domain can be decomposed into a private and a shared term in a common feature

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

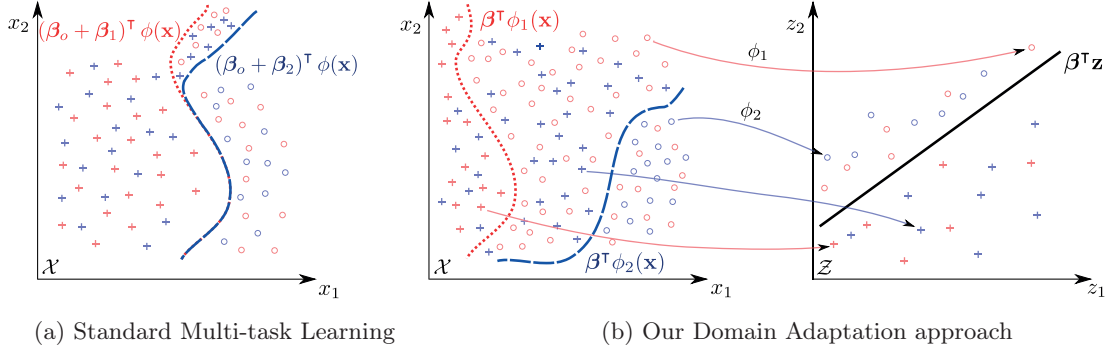


Figure 4.2: Illustration of the difference between (a) standard Multi-task Learning (MTL) and (b) our domain adaptation approach on two tasks. The feature points for each task are shown in either red or blue, and each point is drawn as a cross or circle depending on its class. The dotted and dashed curves represent the decision boundaries of each task. MTL assumes a single, pre-defined transformation $\phi(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Z}$ and learns shared and task-specific linear boundaries in \mathcal{Z} , namely β_o , β_1 and $\beta_2 \in \mathcal{Z}$. In contrast, our approach learns a single linear boundary β in a common feature space \mathcal{Z} , and task-specific mappings $\phi_1(\mathbf{x})$, $\phi_2(\mathbf{x}) : \mathcal{X} \rightarrow \mathcal{Z}$. Best viewed in color.

space \mathcal{X} , as illustrated by Fig. 4.2(a). Unfortunately, acquisition artifacts like the ones shown in Fig. 4.1(a-d) may induce a significant, possibly non-linear transformation in feature space that may violate this assumption, as shown in Fig. 4.2(b). To correct for these unknown transformations, we propose to learn a non-linear mapping of the features in each domain, such that samples can be mapped to a common discriminative latent space \mathcal{Z} , where a shared decision boundary exists, as depicted by Fig. 4.2(b). Such mappings seek to compensate for domain differences and acquisition artifacts, so that the classification task can be shared among them.

In this chapter we develop a boosting-based approach [23, 43, 115] that can simultaneously learn the non-linear mappings as well as the shared decision boundary. We boost regression trees or stumps and model the domain-specific mappings with a set of common regression trees that are shared across domains, but whose thresholds have been adapted to each of them. Our approach does not require neither specific *a priori* knowledge about the mappings' global analytical form or explicit correspondences between training samples in the different domains. This is unlike more conventional Latent Variable Models that can be applied to learn a shared mapping, such as those based on Canonical Correlation Analysis (CCA) [9, 30]. These methods generally require instance-level correspondences which limits their applicability because they rarely

are explicitly available and can be difficult to establish reliably. The situation is further complicated by the fact that the unknown mappings often are non-linear. Although kernel methods can handle this in theory [9, 73, 91], they require kernel functions that can be difficult to specify *a priori*. Furthermore, the computational complexity of kernel methods scales quadratically with the number of training samples, thus limiting their applicability when there are large amounts of data available in the source domain.

In contrast, our approach easily scales to large training datasets and high-dimensional feature spaces, often found in medical imaging [65, 97]. Moreover, unlike other methods, our approach does not require tuning any parameter except those needed by the boosted classifier it relies on. In practice, this is an important advantage, since cross-validation can be unreliable when few labeled data is afforded in the target domain.

We evaluate our approach on the four challenging bio-medical applications depicted by Fig. 4.1.

- The first two applications are mitochondria and synapse segmentation from large 3D Electron Microscopy (EM) stacks of neural rat tissue where the task is to classify voxels that belong to either structure of interest. We use as source and target domains stacks coming from different parts of the brain, each exhibiting different acquisition artifacts, making it difficult to apply standard machine learning to learn a classifier that generalizes across image stacks and for which domain adaptation is required to reduce costly annotation effort.
- We also consider the detection of Olfactory Projection Fibers from two-photon Light Microscopy stacks and axons in Brightfield imagery. Although these represent two very different imaging modalities, the task is the same in each, where we want to classify voxels as to whether they belonging to tubular structures. To showcase the power of our approach, we use as our source domain the 2D aerial images of roads shown in the bottom left of Fig. 4.1. This is of practical significance for two reasons. First, the appearance of the roads is very different from that of the fibers or dendrites. Second, delineating semi-automatically in 2D is much easier than delineating in 3D and our method makes its possible to leverage this easily obtainable 2D data to perform the much harder 3D task.

We will show that our approach consistently outperforms recent multi-task learning techniques [9, 23, 24] across this wide range of applications.

4.2 Related Work

Domain adaptation and more generally Multi-Task Learning have received considerable attention in the machine learning and computer vision communities. However, they have only recently been gaining interest in Medical Imaging [16, 100, 107], and remain largely unexplored for the acquisition problem. In this section we briefly review the state-of-the-art methods in each of these communities and clarify their connections to our work.

Initial approaches to multi-task learning exploited supervised data from related tasks to define a form of regularization in the target problem [11, 21]. In this setting, related tasks, also sometimes referred to as *auxiliary problems* [3], are used to learn a latent representation and find discriminative features shared across tasks. This representation is then *transferred* to the target task to help regularize the solution and learn from fewer labeled examples. The success of these approaches crucially hinges on the ability to define auxiliary tasks. Although this can be easily done in certain situations, as in [3], in many cases it is unclear how to generate them.

More recent multi-task learning methods jointly optimize over both the shared and task-specific components of each task [23, 26, 32, 61]. In [32] it was shown how the two step iterative optimization of [3] can be cast into a single convex optimization problem. In particular, for each task their approach computes a linear decision boundary defined as a linear combination between a shared hyperplane, shared across tasks, and a task-specific one in either the original or a kernelized feature space. This idea was later further generalized to allow for more generic forms [26, 47, 61, 112], as in [26] that investigated the use of a hierarchically combined decision boundary.

For many problems, such as those common to domain adaptation [49], the decision problem is in fact the same across tasks, however, the features of each task have undergone some unknown transformation. Feature-based approaches seek to uncover this transformation by learning a mapping between the features across tasks [73, 90, 94]. A cross-domain Mahalanobis distance metric was introduced in [90] that leverages across-task correspondences to learn a transformation from the source to target domain. A similar method was later developed in [60] to handle cross-domain feature spaces of a different dimensionality. [114] devises a surrogate kernel approach for modeling covariate shift that matches domain feature distributions in Hilbert space and avoids the need for

cross-domain correspondences. Shared latent variable models have also been proposed to learn a shared representation across multiple feature sources or tasks [9, 40, 73, 91, 94].

Feature-based methods generally require well established cross-domain correspondences and/or model non-linearities using the kernel-trick that relies on the selection of a pre-defined kernel function and is difficult to scale to large datasets. Instead, we pursue a discriminative learning approach that does not require explicit cross-domain correspondences, and exploit the *boosting-trick* [23, 43] to handle non-linearities and learn a shared representation across tasks, overcoming these limitations.

The use of boosting for multi-task learning was explored in [23] as an alternative to kernel-based approaches. For each task they optimize for a shared and task-specific decision boundary, as in [32], except that non-linearities are modeled using a boosted feature space. As with other methods, however, additional parameters are required to control the degree of sharing between tasks and can be difficult to set, especially when one or more tasks have only a few labeled samples. Similarly, [25] devises a boosting-based domain adaptation method assuming that the source domain contains out-dated samples that are down-weighted during learning. Even though [23] and [25] address different adaptation problems, both assume that there exist weak learners that can be shared between domains or tasks as a means of regularizing inter-domain learning, which may not be true in cases such as those shown in Fig. 4.1.

Another interesting method is that of [24] that learns a boosted regressor for web search ranking, using regression tree weak learners. They adapt boosted regression trees learned in the source domain to the target domain by interpolating the thresholds and leaf-node responses in each tree. In this way, similar to [23, 25], they seek to recover the private component of the target domain that in our problem corresponds to the unwanted acquisition artifacts. Furthermore, they require an interpolation parameter that weights the different domains, which, as with [23], can be difficult to cross-validate when afforded few training samples in the target domain.

In contrast to [23, 24, 25], we learn a mapping to a shared feature space that preserves the task-relevant features and learn the thresholds across domains by jointly minimizing a common loss that does not rely on a pre-defined adaptation parameter.

Within the Medical Imaging community, domain adaptation has been applied to augment training data from synthetically generated samples [44, 107], as well as to

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

modality fusion [50] and multi-task anomaly detection in CT and ultrasound [16]. However, the data acquisition problem depicted by Fig. 4.1 remains largely unexplored. An exception is [100], which targets image segmentation using labeled samples obtained across multiple image acquisitions. However, [100] is based on a sample re-weighting scheme that relies on having several labeled acquisitions, not always available in large numbers for EM and LM, and is difficult to scale to large training datasets. In contrast, our approach can leverage as little as one source acquisition, and is also easily amenable to large data volumes and high dimensional feature spaces.

4.3 Our Approach

In this section we first introduce our shared latent space model. We then discuss the specific weak learners we use.

4.3.1 Shared Latent Space Model

We consider the problem of learning a binary decision function from supervised data collected across multiple domains. In our setting, each task is an instance of the same underlying decision problem, however, its features are assumed to have undergone some unknown non-linear transformation. Even though *task* and *domain* originally denote different concepts, in the remainder of this chapter we use these terms interchangeably as is generally done in the literature [23, 25].

Assume that we are given training samples $X_t = \{\mathbf{x}_i^t, y_i^t\}_{i=1}^{N_t}$ from $t = 1, \dots, T$ tasks, where $\mathbf{x}_i^t \in \mathbb{R}^D$ represents a feature vector for sample i in task t and $y_i^t \in \{-1, 1\}$ its label. For each task, we seek to learn a non-linear transformation $\phi_t(\mathbf{x}^t)$ that maps \mathbf{x}^t to a common, task-independent feature space \mathcal{Z} , accounting for unwanted feature transformations. Instead of relying on pre-defined kernel functions, we model each transformation using a set of M task-specific non-linear functions $\mathcal{H}_t = \{h_1^t, \dots, h_M^t\}$, $h_j^t : \mathbb{R}^D \rightarrow \mathbb{R}$, to define $\phi_t : \mathcal{X}_t \rightarrow \mathcal{Z}$ as $\phi_t(\mathbf{x}^t) = [h_1^t(\mathbf{x}^t), \dots, h_M^t(\mathbf{x}^t)]^T$. In the context of boosting, the $h_j^t(\cdot)$ represent all the possible weak learners and $M = |\mathcal{H}_t|$ is the total number of them, which can be large and possibly infinite.

We consider functions of the form

$$h_j^t(\mathbf{x}^t) = h_j(\mathbf{x}^t - \boldsymbol{\tau}_j^t), \quad j = 1, \dots, M, \quad (4.1)$$

where $\mathcal{H} = \{h_1, \dots, h_M\}$ are shared across tasks, while $\boldsymbol{\tau}_j^t \in \mathbb{R}^D$ are task-specific. An interpretation of Eq. 4.1 is that all tasks share mid-level representations of the decision boundary, namely the weak learners $h_j(\cdot)$. However, for those representations to be shared among domains, the low level responses must be adapted to compensate for varying imaging conditions. The latter is accomplished through the $\boldsymbol{\tau}_j^t$. Empirically we found this model to work well in cases of domain shift resulting from differences in acquisition artifacts, such as those typically encountered in bio-medical applications.

Assuming that the problem is linearly separable in \mathcal{Z} , the predictive function $f_t(\cdot) : \mathbb{R}^D \rightarrow \mathbb{R}$ for each task can then be written as

$$f_t(\mathbf{x}) = \boldsymbol{\beta}^T \phi_t(\mathbf{x}^t) = \sum_{j=1}^M \beta_j h_j(\mathbf{x}^t - \boldsymbol{\tau}_j^t), \quad (4.2)$$

where $\boldsymbol{\beta} \in \mathbb{R}^M$ is a linear decision boundary in \mathcal{Z} that is common to all tasks, and corresponds to a non-linear boundary in each of the original task-specific input spaces via the ϕ_t . This contrasts with previous approaches to multi-task learning such as [23, 32] that learn a separate decision boundary per task, β_t , in a common input space $\phi(\cdot)$, as shown in Fig. 4.2. In the results section we show that our approach performs better for applications such as those depicted by Fig. 4.1.

We learn the functions $f_t(\cdot)$ by minimizing the exponential loss on the training data across each task

$$\boldsymbol{\beta}^*, \Gamma^* = \min_{\boldsymbol{\beta}, \Gamma} \sum_{t=1}^T c_t L(\boldsymbol{\beta}, \Gamma_t; X_t), \quad (4.3)$$

where $c_t \in \mathbb{R}$ is the weight of task t , and

$$L(\boldsymbol{\beta}, \Gamma_t; X_t) = \sum_{i=1}^{N_t} \exp[-y_i^t f_t(\mathbf{x}_i^t)] \quad (4.4)$$

$$= \sum_{i=1}^{N_t} \exp\left[-y_i^t \sum_{j=1}^M \beta_j h_j(\mathbf{x}_i^t - \boldsymbol{\tau}_j^t)\right], \quad (4.5)$$

with $\Gamma = [\Gamma_1, \dots, \Gamma_T]$ and $\Gamma_t = [\boldsymbol{\tau}_1^t, \dots, \boldsymbol{\tau}_M^t]$.

The explicit minimization of Eq. (4.3) can be very difficult because in practice the dimensionality of $\boldsymbol{\beta}$ can be prohibitively large and the h_j 's are typically discontinuous and highly non-linear. Luckily, this is a problem for which boosting is particularly well suited [43]. It has been shown to be an effective method for constructing a highly accurate classifier from a possibly large collection of weak predictors. Similar to the

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

kernel-trick, the resulting *boosting-trick* [23, 43, 115] can be used to define a non-linear mapping to a high dimensional feature space in which we assume the data to be linearly separable. Unlike the kernel-trick, however, the boosting-trick defines an explicit mapping for which β is assumed to be sparse [23, 89]. Within this setting, each h_j can be interpreted as a weak non-linear predictor of the task label.

We use gradient boosting [43, 115] to solve for $f_t(\cdot)$. Given any twice-differentiable loss function, gradient boosting minimizes the loss in a stage-wise manner for iterations $k = 1$ to K . More specifically, we use the quadratic approximation introduced by [115]. When applied to minimizing Eq. (4.3), the goal at each boosting iteration is to find the weak learner $\tilde{h} \in \mathcal{H}$ and the set $\{\tilde{\tau}^1, \dots, \tilde{\tau}^T\}$ that minimize

$$\sum_{t=1}^T \left(\sum_{i=1}^{N^t} w_{ik}^t \left[\tilde{h}(\mathbf{x}^t - \tilde{\tau}^t) - r_{ik}^t \right]^2 \right), \quad (4.6)$$

where w_{ik}^t and r_{ik}^t can be computed by differentiating the loss of Eq. (4.5), obtaining $w_{ik}^t = c_t e^{-y_i^t f_t(\mathbf{x}_i^t)}$ and $r_{ik}^t = y_i^t$. Once \tilde{h} and $\{\tilde{\tau}^1, \dots, \tilde{\tau}^T\}$ are found, a line-search procedure is applied to determine the optimal weighting for \tilde{h} and the predictive functions $f_t(\cdot)$ are updated, as described in Alg. 1. Shrinkage may be applied to help regularize the solution, particularly when using powerful weak learners such as regression trees [43].

Our proposed approach is summarized in Alg. 1. The main difficulty in implementing it is at line 4. Finding the optimal values of \tilde{h} and $\{\tilde{\tau}^1, \dots, \tilde{\tau}^T\}$ that minimize Eq. 4.6 can be very expensive, depending on the type of weak learners employed. In the next section we show that regression trees and boosted stumps can overcome this problem.

4.3.2 Weak Learners

In this section we introduce the weak learners used in our approach and their corresponding training procedure. We consider both regression tree and decision stump weak learners.

Regression trees have proven very effective when used as weak learners in conjunction with gradient boosting [22]. An important advantage is that training regression trees involves almost no parameter tuning and is very efficient when a greedy top-down approach is used [43].

Decision stumps are a special case of single-level regression trees. Despite their simplicity, they have been shown to achieve high performance in challenging tasks such

Algorithm 1 Non-Linear Domain Adaptation with Boosting

Require: Training samples and labels for T tasks $X_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{N_t}$

Task weights $c_t \in \mathbb{R}$ for each task t . Typically $c_t = 1 \ \forall \ t$

Number of iterations K , shrinkage factor $0 < \gamma \leq 1$

1: Set $f_t(\cdot) = 0 \ \forall \ t = 1, \dots, T$

2: **for** $k = 1$ to K **do**

3: Let $w_{ik}^t = c_t e^{-y_i^t f_t(\mathbf{x}_i^t)}$ and $r_{ik}^t = y_i^t$

4: Find weak learner and parameters:

$$\left\{ \tilde{h}(\cdot), \tilde{\boldsymbol{\tau}}^1, \dots, \tilde{\boldsymbol{\tau}}^T \right\} = \arg \min_{h \in \mathcal{H}, \boldsymbol{\tau}^1, \dots, \boldsymbol{\tau}^T} \sum_{t=1}^T \sum_{i=1}^{N_t} w_{ik}^t \left[h(\mathbf{x}_i^t - \boldsymbol{\tau}^t) - r_{ik}^t \right]^2$$

5: Find $\tilde{\alpha}$ through line search:

$$\tilde{\alpha} = \arg \min_{\alpha} \sum_{t=1}^T \sum_{i=1}^{N_t} c_t \exp \left[-y_i^t \left(f_t(\mathbf{x}_i^t) + \alpha \tilde{h}(\mathbf{x}_i^t - \tilde{\boldsymbol{\tau}}^t) \right) \right]$$

6: Set $\tilde{\beta} = \gamma \tilde{\alpha}$

7: Update $f_t(\cdot) = f_t(\cdot) + \tilde{\beta} \tilde{h}(\cdot - \tilde{\boldsymbol{\tau}}^t) \ \forall \ t = 1, \dots, T$

8: **end for**

9: **return** $f_t(\cdot) \ \forall \ t = 1, \dots, T$

as face and object detection [2, 104]. In cases where feature dimensionality D is very large, decision stumps may be preferred to regression trees to reduce training time.

4.3.2.1 Regression Trees

We use trees whose splits operate on a single dimension of the feature vector, also known as *orthogonal splits*, and follow the top-down greedy tree learning approach described in [43]. The top split is learned first so as to minimize

$$\begin{aligned} \arg \min_{\substack{n \in \{1, \dots, D\}, \\ \eta_1, \eta_2, \{\tau^1, \dots, \tau^T\}}} & \sum_{t=1}^T \left(\sum_{i=1}^{N_t} \mathbf{1}_{\{\mathbf{x}_i^t[n] - \tau^t\}} w_{ik}^t [\eta_1 - r_{ik}^t]^2 \right. \\ & \left. + \sum_{i=1}^{N_t} \bar{\mathbf{1}}_{\{\mathbf{x}_i^t[n] - \tau^t\}} w_{ik}^t [\eta_2 - r_{ik}^t]^2 \right), \end{aligned} \quad (4.7)$$

where $\mathbf{x}[n] \in \mathbb{R}$ denotes the value of the n^{th} dimension of \mathbf{x} , $\mathbf{1}_{\{\cdot\}}$ is the step function, and $\bar{\mathbf{1}}_{\{\cdot\}} = 1 - \mathbf{1}_{\{\cdot\}}$. As in Eq. 4.6, the weights, w_{ik}^t , and residuals, r_{ik}^t , are computed

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

by differentiating the loss of Eq. (4.5). The difference with classic regression trees is that, in addition to learning the values of η_1 , η_2 and n , our approach requires the tree to also learn a threshold $\tau^t \in \mathbb{R}$ per task. Given that each split operates on a single attribute $\mathbf{x}[n]$, the resulting $\tilde{\tau}^t$ is sparse, and learned one component at a time as the tree is built.

Once the top split is learned, new splits are learned on children leaves recursively. This process stops when the maximum depth L , given as a parameter, is reached, or there are not enough samples to learn a new node at a given leaf.

4.3.2.2 Decision Stumps

Decision stumps consist of a single split and return values $\eta_1, \eta_2 = \pm 1$. If also $r_{ik}^t = \pm 1$, which is true when boosting with the exponential loss, then it can be demonstrated that minimizing Eq (4.7) can be separated into T independent minimization problems for all D attributes for each n . Once this is done, a quick search can be performed to determine the n that minimizes Eq. (4.7). This makes decision stumps feasible for large-scale applications with very high dimensional feature spaces.

When using the exponential loss in conjunction with decision stumps, Alg. 1 reduces to a procedure similar to classic AdaBoost [37], with the exception that weak learner search is done in the multi-task manner described above.

4.3.2.3 Training Complexity

Both regression trees and decision stumps require storage linear in the number of training samples in each task. Similarly, the time complexity of training a single decision stump is linear in the total number of training examples or $\mathcal{O}(\bar{N})$, with

$$\bar{N} = \sum_{t=1}^T N_t. \quad (4.8)$$

This contrasts with kernel machines whose storage and time complexity is $\mathcal{O}(\bar{N}^2)$ [73].

Regression trees are more costly to train as they require a joint search over the thresholds across tasks whose complexity is $\mathcal{O}(\prod_t N_t)$. In this work we mainly focus on applications containing a single source and target task, representative of the most common domain adaptation setting. In such cases $T = 2$ and the complexity of training

regression trees remains smaller than that of kernel machines, since $N_1 N_2 < (N_1)^2 + (N_2)^2 + 2N_1 N_2$.

For $T > 2$, regression trees become costly and their complexity can grow faster than \bar{N}^2 . It may still be possible to train them efficiently, but we leave this as a topic for future work.

4.4 Evaluation

We evaluated our approach on four challenging and representative domain adaptation problems for which annotation is very time-consuming. We first describe the datasets, our experimental setup and baselines, and finally present and discuss our results.

4.4.1 Datasets

The experiments used for evaluation are described below, and Table 4.1 summarizes the different datasets employed, their characteristics and amount of labeled data available.

4.4.1.1 Mitochondria and Synapse Segmentation

Mitochondria and synapses are structures that play an important role in cellular functioning. Here, the task is to segment mitochondria and synapses from large 3D Electron Microscopy (EM) stacks, acquired from the brain of a rat. Example slice cuts are presented in Fig. 4.1(a-d). As in the path classification problem, 3D annotations are time-consuming and exploiting already-annotated stacks is essential to reduce labeling effort and speed up analysis. We use our boosting-based method with contextual features introduced in Chapter 2.

For mitochondria segmentation, the source domain is a fully-labeled EM stack from the Striatum region of 853x506x496 voxels with 39 labeled mitochondria. The target domain consists of two stacks acquired from the Hippocampus, one a training stack of size 1024x653x165 voxels and the other a test stack of size 1024x883x165 voxels, with 10 and 42 labeled mitochondria in each respectively. The target test volume is fully-labeled, while the training one is partially annotated, similar to a real scenario.

For synapse segmentation, the source domain is a stack acquired from the Cerebellum of size 1027x987x219 voxels with 11 labeled synapses, and the target domain is an EM stack from the Somatosensory Cortex region, which was divided in training

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

Experiment		Modality / Acquisition	Image(s)	Labeled Data (pos / neg samples)
Mitochondria Segmentation	Source Domain	EM / Striatum	853×506×496	39 mitochondria (15k, 275k)
	Target Domain			
	Train	EM / Hippocampus	1024×653×165	10 mitochondria (3k, 12k)
	Test		1024×883×165	42 mitochondria (14k, 265k)
Synapse Segmentation	Source Domain	EM / Cerebellum	853×506×496	11 synapses (3k, 645k)
	Target Domain			
	Train	EM / Som. Cortex	1024×653×165	10 synapses (7k, 510k)
	Test		1024×883×165	28 synapses (35k, 6M)
Paths: Brightfield to OPF	Source Domain	Brightfield / Neural axons	6 images ≈ 800×800×90 each	30k paths (15k, 15k)
	Target Domain			
	Train	OPF / Neural axons	4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
	Test		4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
Paths: OPF to Brightfield	Source Domain	OPF / Neural axons	8 stacks ≈ 512×512×70 each	40k paths (20k, 20k)
	Target Domain			
	Train	Brightfield / Neural axons	3 stacks ≈ 800×750×80 each	15k paths (7.5k, 7.5k)
	Test		3 stacks ≈ 700×900×100 each	15k paths (7.5k, 7.5k)
Paths: Roads to OPF	Source Domain	Aerial Images / Roads	6 images ≈ 750×850 each	30k paths (15k, 15k)
	Target Domain			
	Train	OPF / Neural axons	4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
	Test		4 stacks ≈ 512×512×70 each	20k paths (10k, 10k)
Paths: Roads to Brightfield	Source Domain	Aerial Images / Roads	6 images ≈ 750×850 each	30k paths (15k, 15k)
	Target Domain			
	Train	Brightfield / Neural axons	3 stacks ≈ 800×750×80 each	15k paths (7.5k, 7.5k)
	Test		3 stacks ≈ 700×900×100 each	15k paths (7.5k, 7.5k)

Table 4.1: Description of the segmentation and path classification experiments used for evaluation.

and testing stacks, each of size 750x564x750 and 655x429x250, with 10 and 28 labeled synapses respectively.

4.4.1.2 Path Classification

Tracing arbors of curvilinear structures is a well studied problem that finds applications in a broad range of fields from neuroscience to photogrammetry. In [99], Turetken et al. showed the advantage of using a path classifier and a mixed integer programming formulation to automatically trace such structures. Within this framework, machine learning is used to predict, based on image evidence, if a tubular path between two points in the image belongs to a curvilinear structure or not. [99] employed descriptors named Histogram of Gradient Deviations (HGD) designed to capture several characteristics of tubular structures in images. From the HGDs generated from the training images, 300 of them are randomly picked as codewords of a visual dictionary. For each given path of arbitrary length, the feature vector is generated by finding an embedding of its HGDs in the dictionary. In addition to the 300 HGDs embedding, the feature vector also contains the maximum curvature along the path, which provides information about its geometry.

This approach can be used for both 2D images and 3D image stacks, since feature vectors have a fixed size, regardless of the dimensionality of the input image. This allows us, in theory at least, to apply a classifier trained on 2D images to 3D volumes. The latter would be highly beneficial, since labeling 2D images is much easier than annotating 3D stacks. However, differences in appearance and geometry of the structures may potentially adversely affect classifier accuracy when 2D-trained ones are applied to 3D stacks, which motivates domain adaptation.

We choose images from two publicly available datasets [99] to form two separate target domains. The first one consists of 3D image stacks of Olfactory Projection Fibers (OPF) from the DIADEM challenge [6], as depicted by Fig. 4.1(f). The second one is made of Brightfield microscopy stacks, such as those depicted by Fig. 4.1(g). The latter generates a significantly harder problem, due to the irregular staining of the dendrites and axons, which produces structured noise [99].

As source domain we explore two possible choices, one that relies on 3D imagery and the other on 2D imagery, even though the target domain is 3D. The former is closer to the target domain but the latter makes sense from an operational point of view because

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

it is far easier to extract large amounts of ground truth data semi-automatically from 2D images than from 3D ones. To highlight the power of our approach, we use 2D aerial road images as our source domain, whose appearance is significantly different from that of the dendrites and axons in the target domain.

4.4.2 Experimental Setup

As in Chapter 2, we group voxels into supervoxels to reduce training and testing time for mitochondria and synapse segmentation, which yields 15k positive and 275k negative supervoxel samples in the source domain of the Mitochondria dataset and 7k positive and 645k negative samples in the source domain of the synapse dataset. This renders 12k and 510k negative training samples in the target domain of the Mitochondria and synapse datasets respectively.

To simulate a real scenario, we create 10 different transfer learning problems using the samples from either one mitochondria or synapse at a time as positives, which translates into approximately 300 and 800 positive training supervoxels per mitochondria or synapse, respectively. We use the default parameters in Table 2.1 ($K = 2000$). We evaluate segmentation performance using the Jaccard Index, computed as the number of true positives over the sum of true positives, false negatives and false positives.

For path classification, 2500 positive and negative samples are extracted from each image through random sampling, as in [99]. This results in balanced sets of 30k samples for training in the roads dataset, and 20k for training and 20k for testing for OPF, and 15k in each in for Brightfield. When the last two are used as the source domain, training and testing sets are merged together, yielding 40k and 30k samples respectively. To simulate the lack of training data, we randomly pick an equal number of positive and negative samples for training from the target domain.

The HGD codewords are extracted from the source domain dataset, and used for both domains to generate consistent feature vectors. We employ gradient boosted trees, which in our experiments outperformed boosted stumps and kernel SVMs. For all the boosting-based baselines we set the maximum tree depth to $L = 3$, equivalent to a maximum of 8 leaves, and shrinkage $\gamma = 0.1$, as in [43]. The number of boosting iterations is set to $K = 500$. For these datasets we report the test error computed as the percentage of mis-classified examples.

For all datasets we evaluate our approach with and without class balancing. With class balancing we set $c_t = \frac{1}{N_t}$ to give both tasks equal weight, while without class balancing we set $c_t = 1$ for each task.

4.4.3 Baselines

On each dataset, we compare our approach against the following baselines: training with source or target domain data only (shown as *SD only* and *TD only*), training a single classifier with both target and source domain data (*Pooling*), and with the multi-task approach of [23] (labeled *Chapelle et al.*). On the path classification datasets we evaluate our approach using regression-tree weak learners and therefore also compare to the tree-based adaptation (Trada) method of [24] on these datasets. We evaluate performance with varying amounts of supervision in the target domain, and also show the performance of a classifier trained with all the available labeled data, shown as *Full TD*, which represents fully supervised performance on this domain and is useful in gauging the relative performance improvement of each method. In a sense this represents the gold-standard that the best transfer learning technique could be expected to achieve.

We also compare to linear Canonical Correlation Analysis (CCA) and Kernel CCA (KCCA) [9] for learning a shared latent space on the path classification dataset, and use a Radial Basis kernel function for KCCA, which is a commonly used kernel. Its bandwidth is set to the mean distance across the training observations. Following [60, 90] we establish correspondence between domains using their binary category labels. The data size and dimensionality of the Mitochondria and synapse datasets is prohibitive for these methods, and instead we compare to Mean-Variance Normalization (MVN) and Histogram Matching (HM) that are common normalizations one might apply to compensate for acquisition artifacts. MVN normalizes each input 3D intensity patch to have a unit variance and zero-mean, useful for compensating for linear brightness and contrast changes in the image. HM applies a non-linear transformation and normalizes the intensity values of one data volume such that the histogram of its intensities matches the other.

4.4.4 Results: Mitochondria and Synapse Segmentation

The Jaccard Index on the test stacks of the EM segmentation datasets for 10 different runs is shown in Fig. 4.3 for our approach and the baseline methods, with varying

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

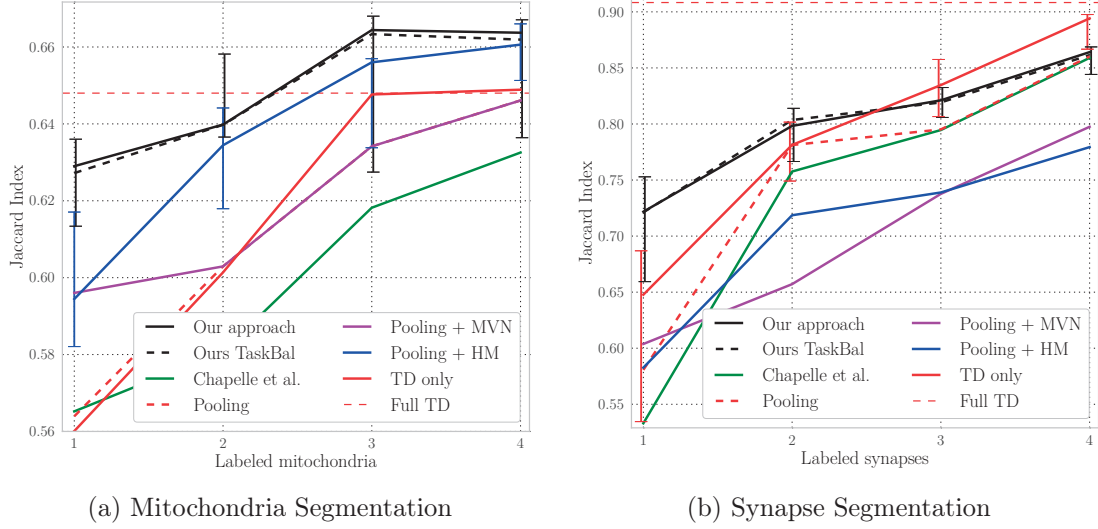


Figure 4.3: EM Segmentation: (a) mitochondria and (b) synapses. Jaccard index measure for our method and the baselines over 10 runs on the target domain, with varying supervision. Simple Mean-Variance Normalization (MVN) and Histogram Matching (HM), although helpful, are unable to fully correct for differences between acquisitions when only afforded few labeled data. In contrast, our method yields a higher performance without the need for such priors and is able to faithfully leverage the source domain data to learn from relatively few examples in the target domain, outperforming the baseline methods.

amounts of supervision in the target domain. The performance of *SD-only* is not displayed since it performs poorly on both datasets and yields a Jaccard Index below 50%.

The results for mitochondria segmentation are displayed in Fig. 4.3(a). Our approach significantly outperforms Chapelle et al. and the other baselines. The next most successful method is pooling with histogram matching (HM). However, our method yields even higher performance, its accuracy being close to that of *Full TD* when using only one labeled target mitochondria. When given more labeled data, both our approach and HM yield higher performance than *TD only* and is even able to use the source domain data to improve over *Full TD*.

Similarly, the results for synapse segmentation are shown in Fig. 4.3(b). Each labeled synapse contains only a few supervoxels. Given such limited supervision, Chapelle et al. does not improve upon *TD-only* performance. Instead, it overfits to the source domain data. Similarly, MVN and HM normalization are unable to account for the transformation between the different data acquisitions. In contrast, our approach is able

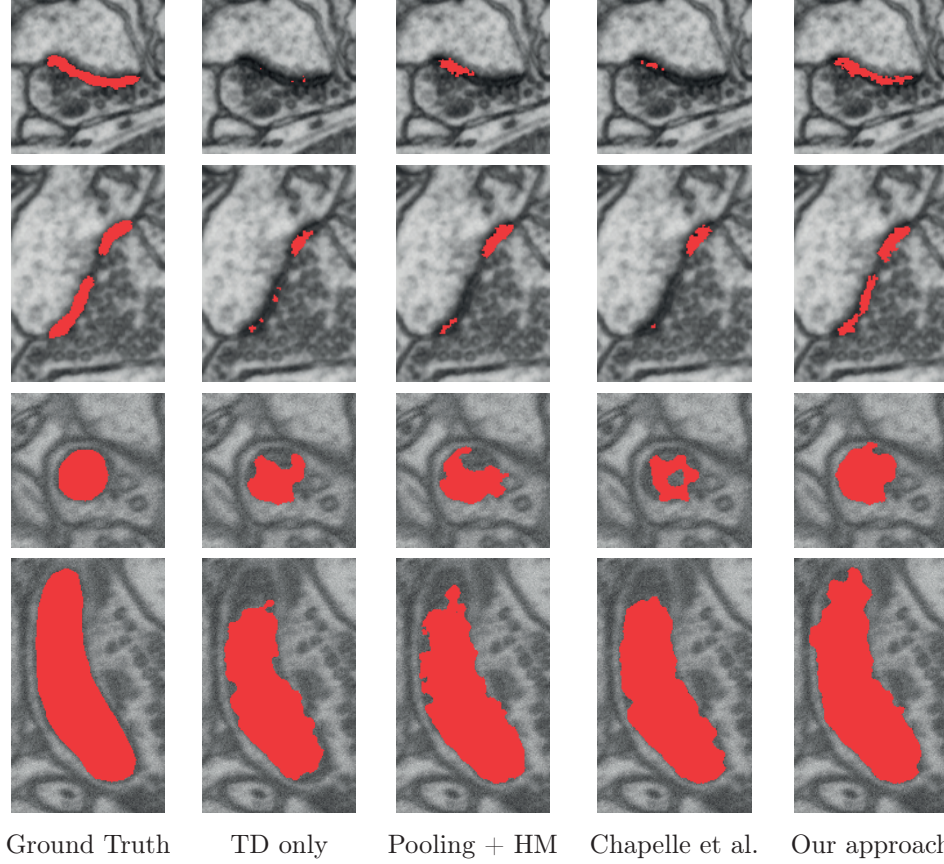


Figure 4.4: Qualitative results for the segmentation datasets when using a single labeled mitochondria or synapse in the target domain. The segmentation masks output by our approach and the baselines are shown in red for two example mitochondria and synapses. The ground-truth is also shown. Compared with baselines the segmentations output by our approach exhibit a higher accuracy and most closely resemble the ground-truth. Best viewed in color.

to effectively leverage the source domain data to obtain a more accurate segmentation even with only one labeled synapse in the target domain. Provided four labeled synapses it becomes difficult to improve over *TD-only* performance. However, as annotation in 3D is costly this already represents a significant labeling effort, and our approach still exhibits the best overall performance.

Qualitative segmentation results obtained with a single labeled mitochondria or synapse are also provided in Fig. 4.4. Compared to the baselines, the segmentations generated by our approach exhibit higher accuracy and most closely resemble the ground truth. From a practical point of view, our approach does not require parameter tuning

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

and cross-validation is not necessary. This can be a bottleneck in some scenarios where large volumes of data are used for training. For this task, training our method took less than an hour per run, while Chapelle et al. took over 7 hours due to cross-validation.

We also compare to linear Canonical Correlation Analysis (CCA) and Kernel CCA (KCCA) [9] for learning a shared latent space on the path classification dataset, and use a Radial Basis kernel function for KCCA, which is a commonly used kernel. Its bandwidth is set to the mean distance across the training observations. Following [60, 90] we establish correspondence between domains using their binary category labels. The data size and dimensionality of the Mitochondria and synapse datasets is prohibitive for these methods, and instead we compare to Mean-Variance Normalization (MVN) and Histogram Matching (HM) that are common normalizations one might apply to compensate for acquisition artifacts. MVN normalizes each input 3D intensity patch to have a unit variance and zero-mean, useful for compensating for linear brightness and contrast changes in the image. HM applies a non-linear transformation and normalizes the intensity values of one data volume such that the histogram of its intensities matches the other.

4.4.5 Results: Path Classification

We first discuss using 3D imagery as both the source and target domains and then 2D imagery as the source while the target remains 3D.

3D Neural Axons as the Source Domain Fig. 4.5 depicts our path classification results using the 3D microscopy images from one microscopy imaging technology as the source domain, and those of the other one as the target domain. As the microscopy images from each dataset depict very different imaging modalities (see Fig. 4.1), this poses a challenge for transfer learning. The performance of *SD-only* and linear CCA on these datasets is above 29% and 8% respectively, and as such they are not displayed in the figure.

The results of Brightfield to OPF are shown in Fig. 4.5 (top). With the exception of Trada and our approach, the other baseline methods have difficulty improving over *TD-only* performance, and in fact perform worse than it especially when provided only a few labeled samples in the target domain. In contrast, our approach achieves a consistent improvement over *TD-only* that is seen to be most significant when the labeled data

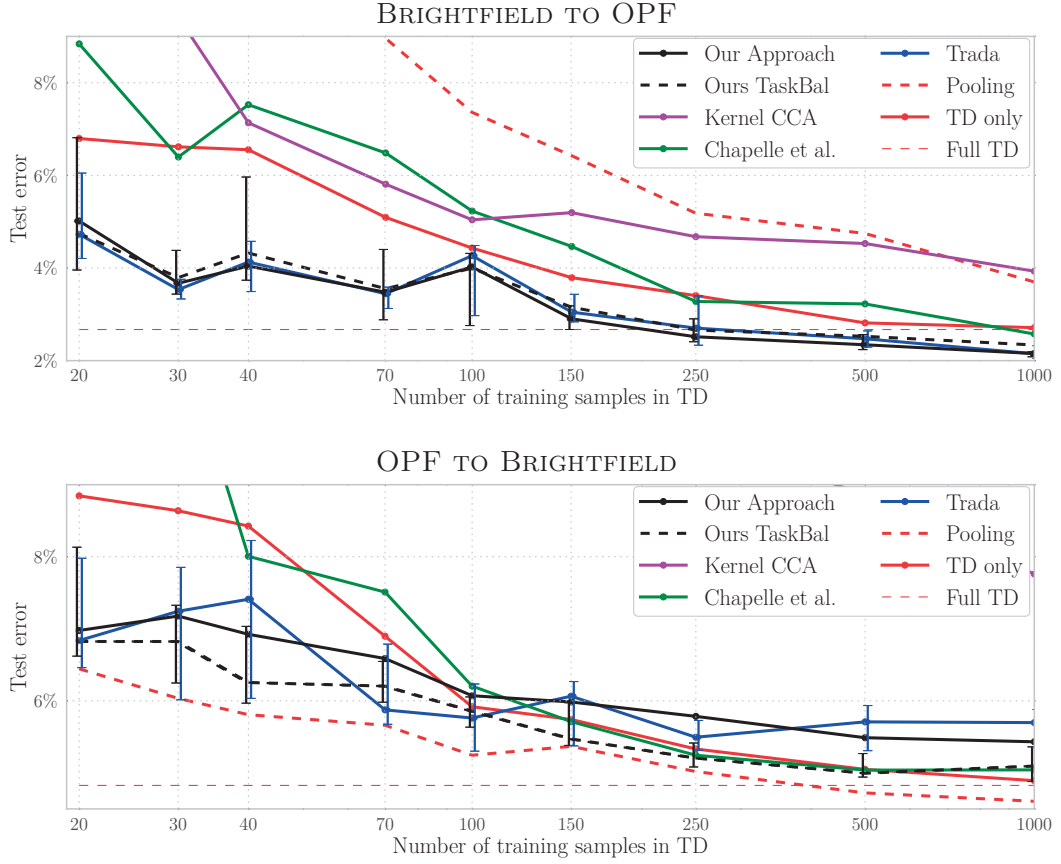


Figure 4.5: *Path Classification, 3D imagery as source domain:* Median, lower and upper quartiles of the test error as the number of training samples is varied. Our approach is able to successfully leverage the source domain data to significantly reduce annotation effort and exhibits the best overall performance across both datasets. Best viewed in color.

in the target domain is scarce, which is when domain adaptation is most needed, and it is even able to improve over *Full-TD*. The performance of our approach is matched by Trada on this dataset, which is also able to achieve a significant improvement over *TD-only* and the other baselines.

Fig. 4.5 (bottom) displays the results for OPF to Brightfield. Our approach with task balancing achieves a significant improvement over *TD-only* performance when provided few target domain training samples and outperforms the baselines. Task balancing plays a more significant role for the Brightfield dataset that can be attributed to the large appearance difference between them and the rich visual cues that are present in Brightfield but absent from OPF. Unlike Brightfield to OPF, Trada is unable to match

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

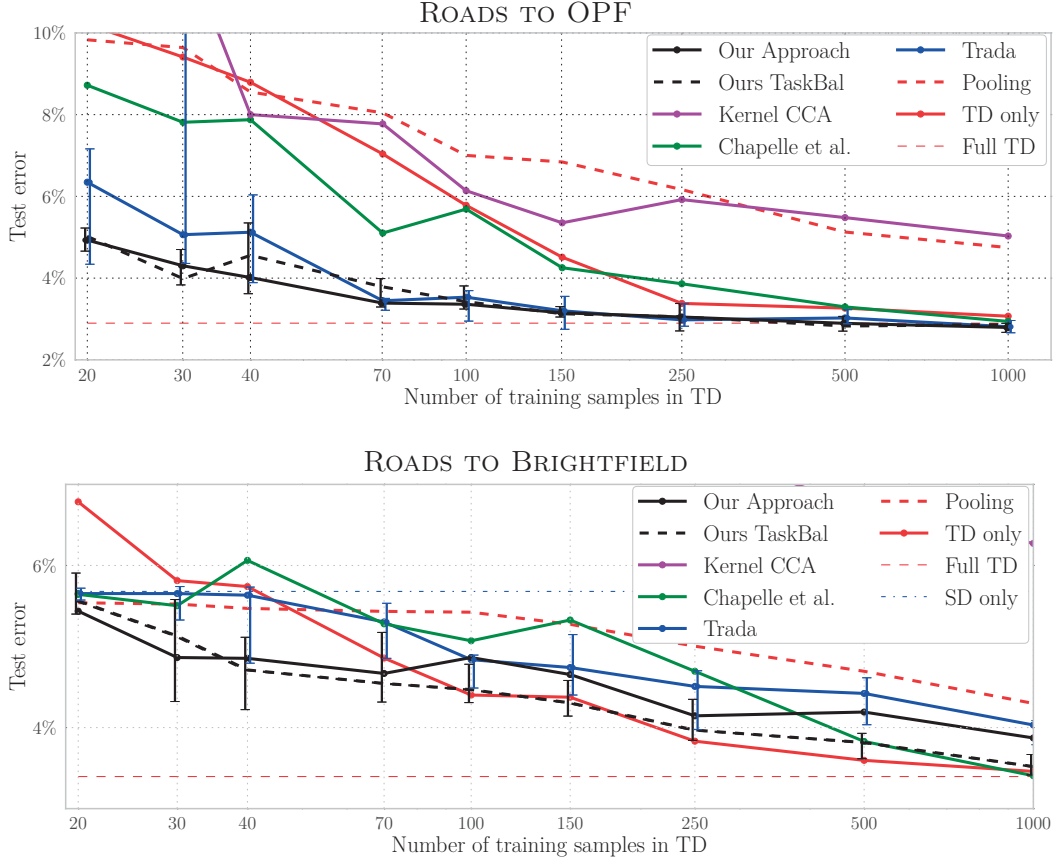


Figure 4.6: *Path Classification, 2D imagery as source domain:* Median, lower and upper quartiles of the test error as the number of training samples is varied. For OPF, our approach nears *Full TD* performance with as few as 70 training samples in the target domain and significantly outperforms the baseline methods for both experiments when afforded few training samples. Best viewed in color.

the performance of our approach when adapting OPF to Brightfield, which is likely due to its reliance on a cross-domain interpolation parameter that can be difficult to cross-validate, that is not required with our approach.

Surprisingly, naive *Pooling* achieves the best performance for OPF to Brightfield. Note, however, that while it does exceptionally well on this dataset, its preference towards Brightfield is also reflected when transferring from Brightfield to OPF where it results in the worst performance that is significantly worse than *TD-only*. In contrast, our approach is able to consistently improve over *TD-only* performance and the baselines and successfully leverage the source domain data to reduce annotation effort across both

datasets.

2D Aerial Roads as the Source Domain Using the same 3D images as before as our target domain, we now switch to aerial road images such as those in the third row of Fig. 4.1 to provide our source domain. Compared with to the 3D microscopy images, the 2D road images exhibit a much more different appearance to those of the target domain and therefore present a greater challenge.

The results on the OPF dataset are shown in Fig. 4.6 (top). Our approach outperforms the baselines, especially when there are few training samples in the target domain, and yields a similar performance with and without task balancing. The next best competitor is Trada, followed by Chapelle et al., although this method exhibits a much higher variance than our approach and both baselines perform poorly when only provided a few labeled target examples. This is also the case for KCCA. The results of linear CCA are not shown in the plots because it yielded very low performance compared to the other baselines, achieving a 14% error rate with 1k labeled examples and its performance significantly degrading with fewer training samples. Similarly, SD only performance is 16%.

Our approach comes close to *Full TD* when using as few as 70 training samples, even though the *Full TD* classifier was trained with 20k samples from the target domain. This highlights the ability of our method to effectively leverage the large amounts of source-domain data. As shown in Fig. 4.6, there is a clear tendency for all methods to converge at the value of *Full TD*, although our approach does so significantly faster. Moreover, the parameter tuning required by Chapelle et al. and Trada is done through cross-validation, which can perform poorly when only afforded a few labeled samples in the target domain, and results in longer training times. Chapelle et al. took 25 minutes to train, while our approach only took between 2 and 15 minutes, depending on the amount of labeled data.

The results on the Brightfield dataset are shown in Fig. 4.6 (bottom). Both linear and kernel CCA perform poorly on this dataset, the performance of linear CCA being only 15% using 1k labeled samples and it is not shown in the plot. Similarly, Chapelle et al. requires a fair amount of supervision in the target domain before achieving an improvement over SD only performance. Trada also performs poorly on this dataset. In contrast, our approach obtains a significant improvement with as little as 30 labeled

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

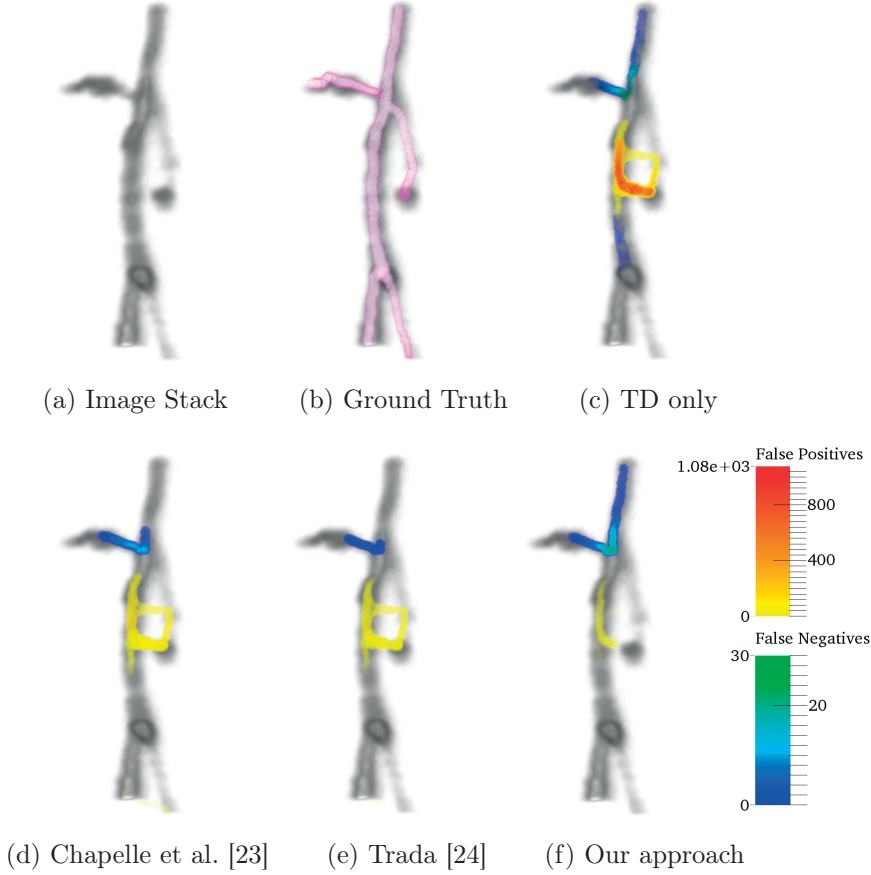


Figure 4.7: Qualitative results for the OPF path classification dataset. The 3D visualizations show the amount of false positive and false negative paths predicted by each approach at every location in the stack along with the ground-truth. The color coding displays the number of false or missed detections passing through each location. While all approaches result in only a few missed detections, compared with the baseline methods our approach produces significantly fewer false detections. Best viewed in color.

target samples, outperforming the baseline methods. For > 70 labeled target samples, although it still performs better than the other methods, our approach without task balancing performs worse than the TD only baseline. We believe this is because of task-specific attributes in the Brightfield dataset that are not modeled with our approach. This effect is diminished with task balancing, which assigns more emphasis to the target training samples during learning. Despite these differences, our approach is still able to more effectively leverage the source domain data to reduce the required amount of supervision in the target domain compared to the baselines.

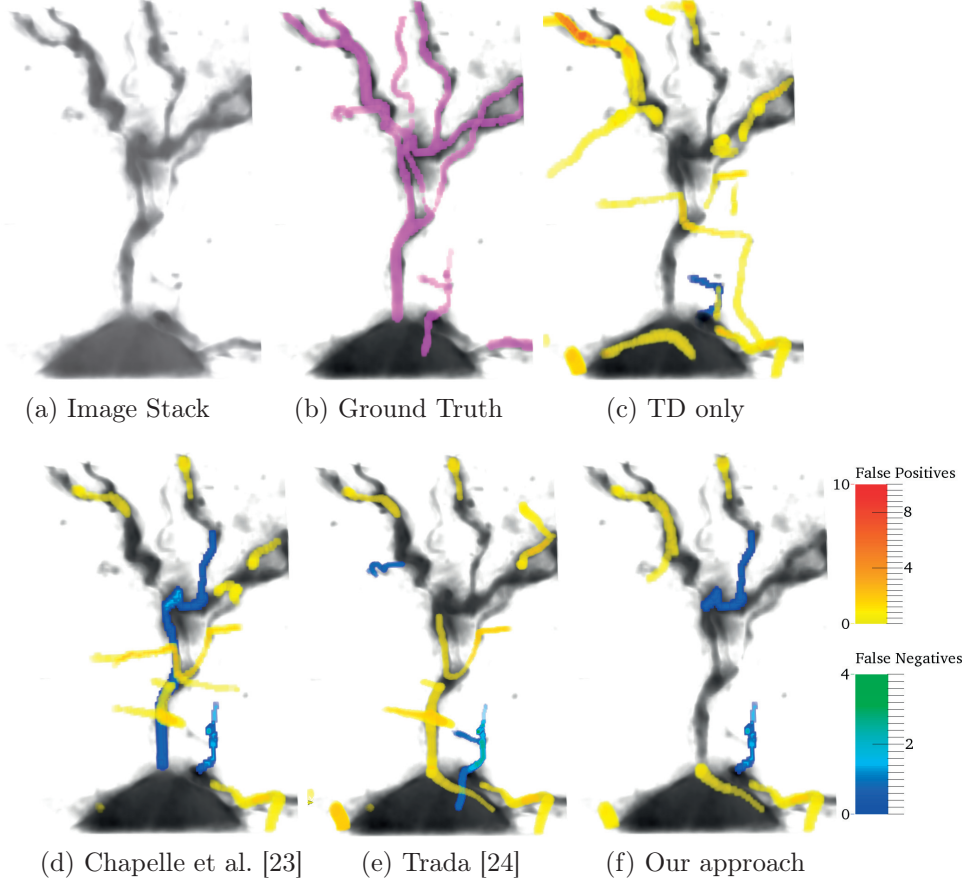


Figure 4.8: Qualitative results for the Brightfield path classification dataset. The 3D visualizations show the amount of false positive and false negative paths predicted by each approach at every location in the stack along with the ground-truth. The color coding displays the number of false or missed detections passing through each location. Compared with the baselines our approach results in the fewest overall number of false and missed detections yielding a more accurate path classification. Best viewed in color.

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

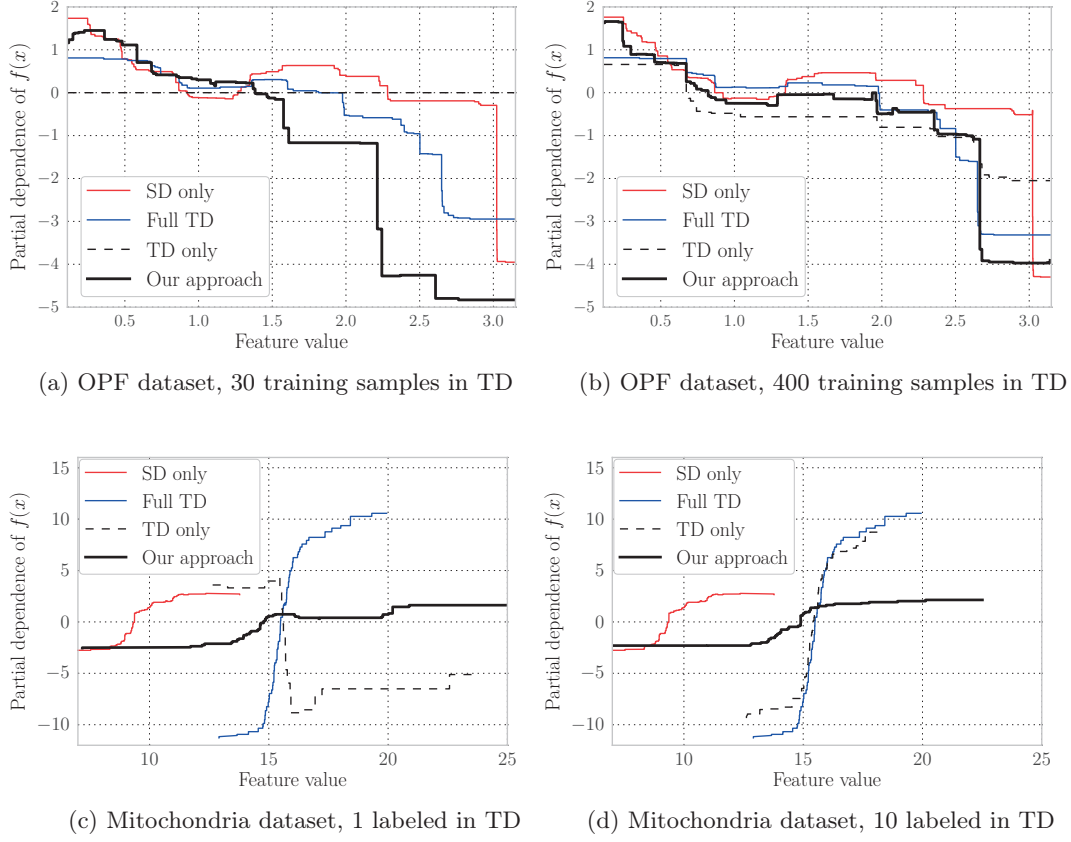


Figure 4.9: Analysis of the behavior of the trained classifiers through partial dependence plots for the OPF (top) and Mitochondria segmentation (bottom) datasets, with different amounts of training data in the Target Domain. Best viewed in color.

Qualitative results are displayed for both the OPF and Brightfield datasets in Fig. 4.7 and 4.8. The false and missed detections are shown for each of the baselines and our approach. As false detections typically concentrate about overlapping subpaths on these datasets, we display a color coding that for each voxel reflects the number of false or missed detections that include it. On OPF all approaches result in only a few missed detections, however, our approach achieves a significant decrease in false detections. Compared with OPF, the Brightfield dataset contains more complicated path structures. Our approach exhibits the best performance amongst the baseline methods on this dataset, with the fewest overall number of false and missed detections resulting in a more accurate path reconstruction.

4.4.6 Partial Dependence Analysis

To analyze the behavior of the classifiers learned with our approach, we use Partial Dependence Plots (PDPs) [43] to observe the classifier score as a function of the value of one specific feature, averaging out the effect of the other features. If $\mathbf{x} = (x[1], \dots, x[M])^T$ and features are indexed with $\mathcal{P} = \{1, 2, \dots, M\}$, denote the scoring function as $f(\mathbf{x}) = f(x[n], \mathbf{x}_c)$, where \mathbf{x}_c contains all features but the n^{th} one. The partial dependence of $f(\mathbf{x})$ with respect to the n^{th} feature is then computed as

$$\bar{f}_n(\lambda) = \frac{1}{|X|} \sum_{\mathbf{x} \in X} f(\lambda, \mathbf{x}_c), \quad (4.9)$$

where X is the set of available training data.

We choose λ to be features with high relative importance [43] for the path classification and mitochondria segmentation datasets, and then plot the PDPs for the baselines *SD only*, *TD only*, *Full TD*, and our approach in Fig. 4.9. When comparing two classifiers, what matters is their behavior as a function of the feature value, i.e., the shape of their response, while the overall scaling is classifier-dependent.

For the OPF dataset, we plot the partial dependence of the feature that encodes the maximum curvature along the path. From Figs. 4.9(a,b) it is observed that the classifier prefers paths with a low curvature, which is a sensible choice, since the shape of tubular structures is typically smooth. For the mitochondria dataset the partial dependence of one of the structure tensor eigenvalues is displayed, which has a high value when inside a mitochondria, also reflected in Figs. 4.9(c,d).

In Fig. 4.9 the PDPs of the learned classifiers are displayed with varying amounts of supervision in the target domain. Figures 4.9(a,c) depict the errors that can result from overfitting when afforded only few target domain training samples (*TD only*), such as missing important features (Fig. 4.9(a)), indicated by its constant PDP, or learning an incorrect pattern (Fig. 4.9(c)). In contrast, our approach is able to leverage the source domain data to discover relevant features and prevent overfitting. Another interesting observation is the shift between the curves for *Full TD* and *SD only*, which reflect acquisition differences that are compensated by our approach.

Finally, Figs. 4.9(b,d) show the same plots when afforded a considerable amount of training data in the target domain. In this case, the *TD only* classifier exhibits a more similar performance to *Full TD* and is able to learn a more representative pattern.

4. DOMAIN ADAPTATION FOR MICROSCOPY IMAGING

Although our approach also improves, its PDPs are fairly consistent across different amounts of supervision and it is able to learn a representative pattern even with limited supervision in the target domain.

4.5 Conclusion

We presented an approach for performing non-linear domain adaptation with boosting. Our method learns a task-independent decision boundary in a common feature space, obtained via a non-linear mapping of the features in each task. This contrasts recent approaches that learn task-specific boundaries and is better suited for problems in domain adaptation where each task is of the same decision problem, but whose features have undergone an unknown transformation. In this setting, we illustrated how the boosting-trick can be used to define task-specific feature mappings and effectively model non-linearity, offering distinct advantages over kernel-based approaches both in accuracy and efficiency. Our method relies on mid-level features and its effectiveness depends on the extent to which these features can be shared across the target and source domains. We evaluated our approach on four challenging bio-medical datasets where it achieved a significant gain over using labeled data from either domain alone and outperformed recent multi-task learning methods.

Part III

Synaptic Structure in the Aging Mouse Cortex

SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

In this chapter we apply the synapse segmentation approach introduced in Chapter 3 to analyze and compare the structure and shape of synaptic densities between adult and aged mice. Such detailed analysis requires labeling each voxel within every synapse in a stack, rendering manual annotation unfeasible for large volumes. In this chapter we demonstrate that our approach can efficiently generate full 3D segmentations that agree with expert annotations, while requiring very little annotation effort. To our knowledge, we are the first ones to analyze synapse shape in such detail on large stacks, as previous work has strongly relied on manual annotations, restricting analysis to small volumes.

5.1 Introduction

Though there is significant evidence that synaptic impairment is the main cause of age-related cognitive decline, this phenomenon is so far poorly understood [18, 41]. Although it is believed that synaptic density and plasticity are fundamental factors in synaptic impairment, most studies rely on manual annotation, imposing a limit on the level of detail and structures that can be inspected. For example, while manually labeling the existence of a synapse with a sphere to assess its size and location is typically fast, manually segmenting them voxel by voxel to analyze their shape is infeasible without automated methods.

In this chapter we show we can bridge this gap with the automated segmentation approach introduced in Chapter 3. To show its effectiveness, we analyze six large FIB/SEM brain stacks acquired from six different mice. Three of them are young

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

Mouse	Age (months)	xy voxel size (nm)	z voxel size (nm)	Stack Size (μm)	Stack size (MB)
1	4	6.0	18.6	$12.3 \times 9.2 \times 7.8$	1300
2	12	6.0	13.5	$10.3 \times 11.2 \times 12.9$	2900
3	4	5.0	11.1	$15.2 \times 10.9 \times 15.3$	8600
4	12	5.0	15.0	$12.2 \times 9.2 \times 9.5$	2700
5	12	5.0	10.2	$14.1 \times 11.1 \times 4.7$	2800
6	4	5.0	21.0	$15.2 \times 10.9 \times 14.7$	4400

Table 5.1: Details about the six C57 black 6 male mice involved in the experiments.

adults, and the other three are aged mice, offering an excellent opportunity to try our segmentation approach to analyze how aging affects the brain.

In the following sections we describe our data, the available annotations and how our automated segmentation approach was applied on the six stacks. We then validate our method and compare it to fully manual annotations, demonstrating that it is possible to obtain high quality results with much less labeling effort. Finally, we extract valuable morphological information from our segmentation output, that would otherwise be unfeasible with purely manual annotation.

5.2 Data Acquisition and EM Stacks

We analyze six large FIB/SEM brain stacks, acquired from six different *C57 black 6* males, of 4 and 24 months of age. The data was collected and acquired by Graham Knott at the BioEM Facility at the Centre of Electron Microscopy at the EPFL. The stacks were imaged from cortical layer 1, halfway between the pial surface of the brain, and the beginning of layer 2.

Sample fixation, resin embedding and sample preparation were performed as in [56]. Once acquired, the stacks were registered with the StackReg Fiji plugin [96]. A summary of the available data is shown in Table 5.1.

5.2.1 Manual Annotation with Spheres

The location and approximate size of synapses in each stack were annotated by researchers at the BioEM facility at EPFL using Fiji [93]. The total number of synapses

5.3 Automatic Segmentation

Mouse	Age (mo.)	ROI size (μm)	# Synapses Manual Annotation	# Synapses Our Approach	# Synapses Ours Corrected
1	4	$12 \times 9 \times 7.8$	1145	1321	1175
2	12	$10 \times 11 \times 12.9$	1495	1709	1538
3	4	$10 \times 10 \times 15.3$	2117	2340	2108
4	12	$12 \times 8 \times 9.5$	1061	1212	1095
5	12	$14 \times 4 \times 4.7$	774	737	700
6	4	$8 \times 6 \times 14.7$	956	1223	996

Table 5.2: Details on the Regions of Interest (ROIs) for each stack. We also show the number of synapses counted by experts on the ROI (*Manual Counting*), as well as how many were detected by our approach without any corrections (*Ours*), as well as the number of synapses after manually removing false positive detections (*Ours Corrected*).

to be annotated for the stacks of Table 5.1 is more than 6000, rendering manual voxel-wise segmentation unfeasible. Therefore, each synapse was labeled instead as a *spherical object* in Fiji, whose diameter represents the extent of the synaptic cleft, estimated from 2D slices by the annotator.

A Region of Interest (ROI) was defined for each stack to ensure the analysis is performed in the neuropil and to avoid cell bodies. Synapses outside the ROI were ignored. The size of the ROI for each stack is shown in Table 5.2.

It took on average 6 days to annotate each stack. A first observer spent 4 days labeling the stack from scratch. Later, a second annotator took two more days to check and correct the work of the first.

These manual annotations allow us to estimate synaptic density as well as the distribution of synapses according to their spatial extent. Moreover, this data is very valuable, as it allows us to compare the results obtained by experts in the field of neuroscience to those generated by our automated approach, in terms of synapse count and size.

5.3 Automatic Segmentation

We now describe how our approach was trained and applied to each stack to generate the automatic segmentations. In average, our full pipeline took 24 hours for training and prediction per stack on a single 20-core machine. This is significantly faster than the 6

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

days it took to manually annotate spherical objects. Moreover, our approach outputs a segmentation map, from which we can extract rich morphological information otherwise not available.

5.3.1 Data Annotation

Although we have been given expert annotations with the location and approximate extent of synapses for all stacks, this is not enough for training our approach, as it requires a segmentation, voxel-wise ground truth. We therefore cropped a small region of each stack of approximately 200 MB, and partially labeled it, annotating annotated between 8 and 12 synapses per stack, plus some background for the negative class. Generating such few annotations is very fast, taking between 15 and 20 minutes per stack when using a Wacom drawing tablet.

5.3.2 Training

We used the Multiple-Instance Learning extension of our approach introduced in Chapter 3, and the parameters were kept the same as those shown in Table 2.1. Training took between 30 minutes and an hour when using 20 cores on an Intel Xeon 2.90 GHz CPU. Note that there is no need for user interaction or supervision during training.

5.3.3 Prediction

To generate the segmentation maps we split each stack into smaller sub-stacks, since it is not possible to predict a whole stack at once due to the large amounts of memory needed to store the integral cubes. The size of each sub-stack was limited to 500 MB, including an overlapping region of 80 voxels.

Predictions were done on a single Intel Xeon 2.90 GHz machine, using 20 cores. The sub-stacks were processed sequentially, and each whole stack took between 7 and 35 hours to complete, depending on its size. Note that this process could be easily parallelized on a cluster to speed up prediction.

To generate the final segmentations we applied a 3D median filter with a radius of one voxel to the raw prediction, followed by thresholding. The thresholds for each stack were found by manually inspecting a few slices and their score maps. We set the thresholds to discard spurious detections (eg. thick membranes, if present). The final thresholds were either 10.0 or 20.0, depending on the stack. From our experience,

their exact values are not critical, as the statistics we want to analyze are averages over whole stacks. Finally, we applied connected components on the thresholded binary stack, removing synapses whose volume is less than $200,000 \text{ nm}^3$. In the rest of this chapter and in the tables and plots we refer to the output of this last step *Our approach*.

5.3.4 False Positive Removal

Though the output of the automated predictions can be used directly for analysis, we also created a refined version with false positive detections removed. This results in a reduced set of detections that we call *Ours Corrected* in the tables and plots. This process took on average an hour per stack.

5.4 Comparison and Results

In this section we compare the results obtained with our automated approach against the manual expert annotations from §5.2.1. More specifically, we compare the following:

- **Manual annotation:** annotations with spherical objects, containing synapse location and their approximate extent, as labeled by the neuroscientists at the BioEM facility at EPFL and detailed in §5.2.1.
- **Our approach:** the raw output of our synapse segmentation. Detections are obtained by running Connected Components on the segmentation output, as described in §5.3.3.
- **Ours corrected:** the results of our automated approach but with false positive detections removed, as detailed in §5.3.4

5.4.1 Synapse Counting & Density

Table 5.2 shows the number of detected synapses in the ROI for each method. On average we see that 10% of the detected synapses are false positives, with the exception of the sixth stack where that number raises to 18%. This is not a severe issue, as removing false positives is typically fast.

On the other hand, Fig. 5.1 shows the density estimates obtained with different methods over 1000 random sub-regions of the ROIs. The height, width and depth of the sub-regions are two-thirds the corresponding dimensions of the ROIs, and are placed

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

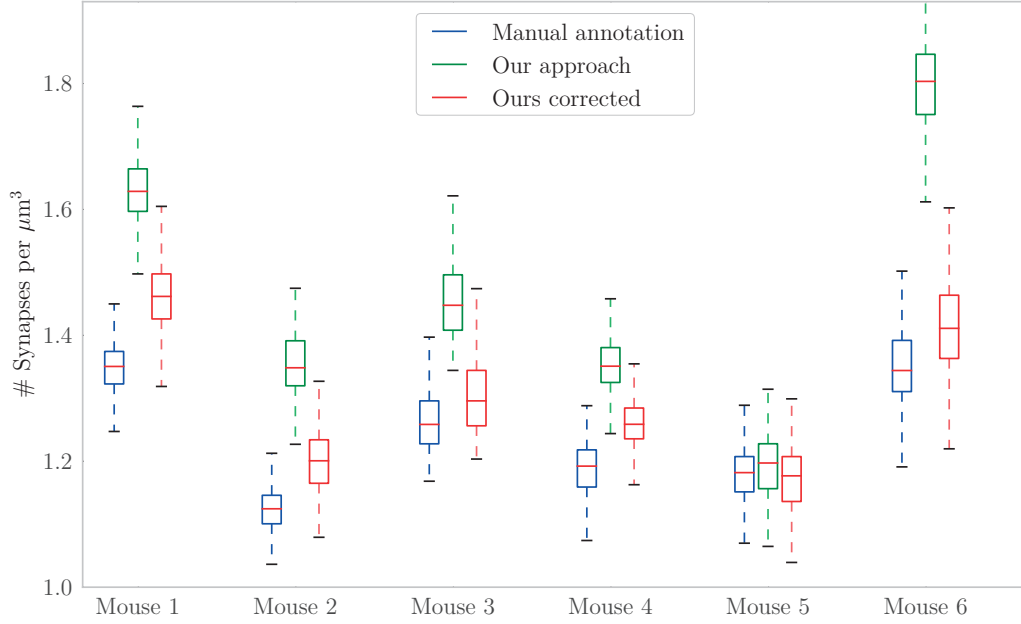


Figure 5.1: Comparison of synapse density estimates obtained with the different methods over random sub-regions of the ROIs.

at random locations within the ROIs. In this way we can generate uncertainty estimates to help us compare the different methods more reliably.

It is clear that there is an over-estimation of the synaptic density when the false positives are not removed from the output of our automated approach. Nonetheless, after doing so, the obtained densities are well within the range of those computed through manual annotation.

5.4.2 Synaptic Cleft Size

The radii of the annotated spheres in the manual annotations can be used to estimate the spatial extent of each synapse. Likewise, we can compute the Feret diameter of each connected component in our segmentation output to obtain a similar measure.

Even though both measures are correlated, they are not directly comparable, as our approach segments the synapse in full 3D, while the sphere annotations were only performed on 2D xy slices. Nonetheless, we expect that strong trends present in the data should manifest themselves in both measurements. To this end, we first plot the distribution of synaptic cleft size, measured through the diameter of the annotated

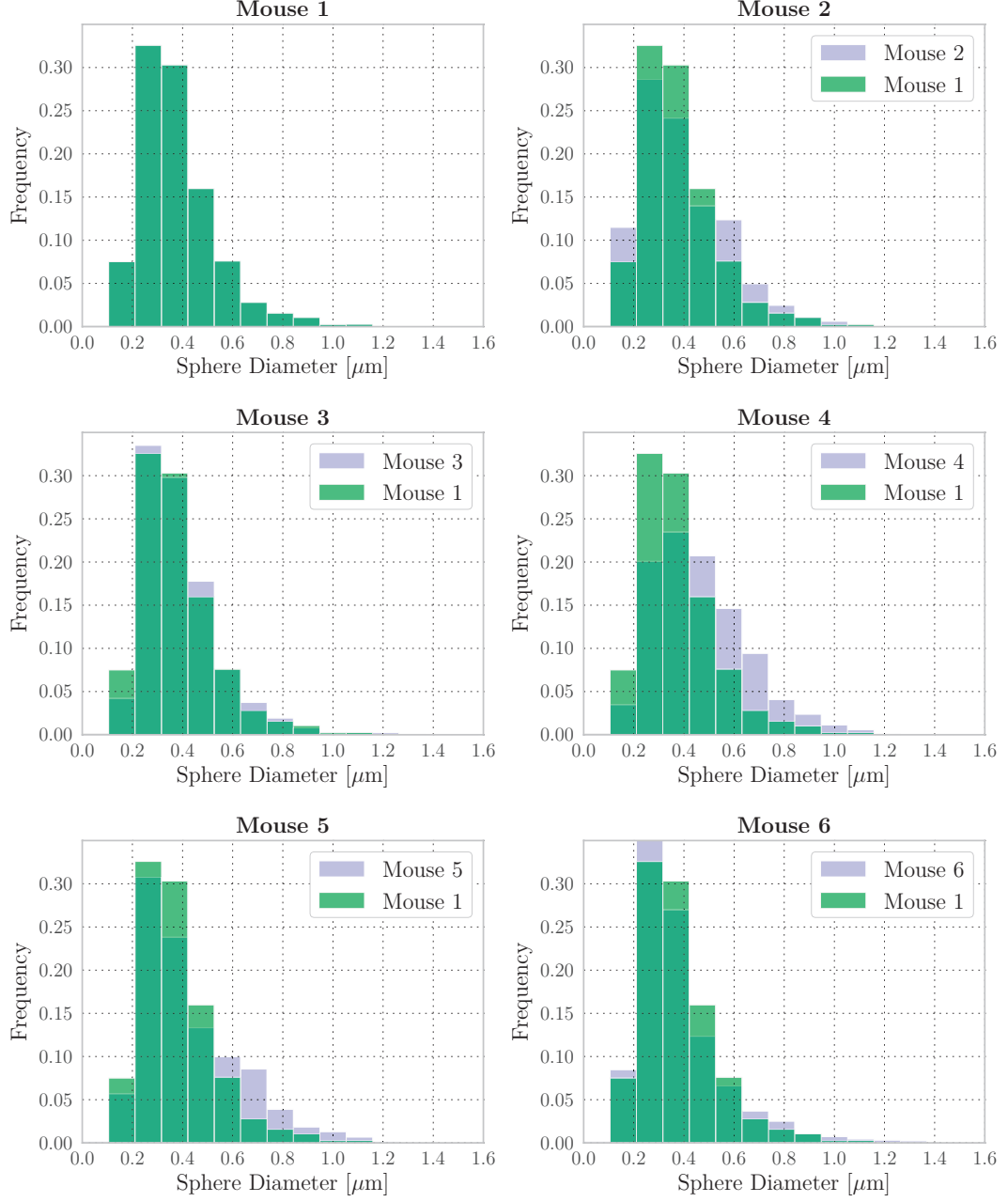


Figure 5.2: Manual annotation. Distribution of synaptic cleft size, measured through the diameter of the annotated spheres. For comparison we plot the distribution of Mouse 1 against the others. Aged mice 2, 4 and 5 present an relative increase in larger synapses.

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

spheres in Fig. 5.2. Note that there is an interesting distinction between aged (2,4 and 5) and young adult mice (1,3 and 6), as there is an increase in the number of larger synapses in the former. This was already observed by Graham Knot and the BioEM team after processing their data.

We then plot similar figures, but now using the Feret diameter of the individual detections of our automated approach. We obtain Figures 5.3 and 5.4, for our approach before and after false positive removal, respectively. We observe the trend is clear on those plots, and correctly agrees with an increase of synapses of Feret diameter between 0.5 and 0.8 μm .

This validates our approach, which yields similar results before and after false positive removal, in a fraction of the time, and with significantly less annotation effort.

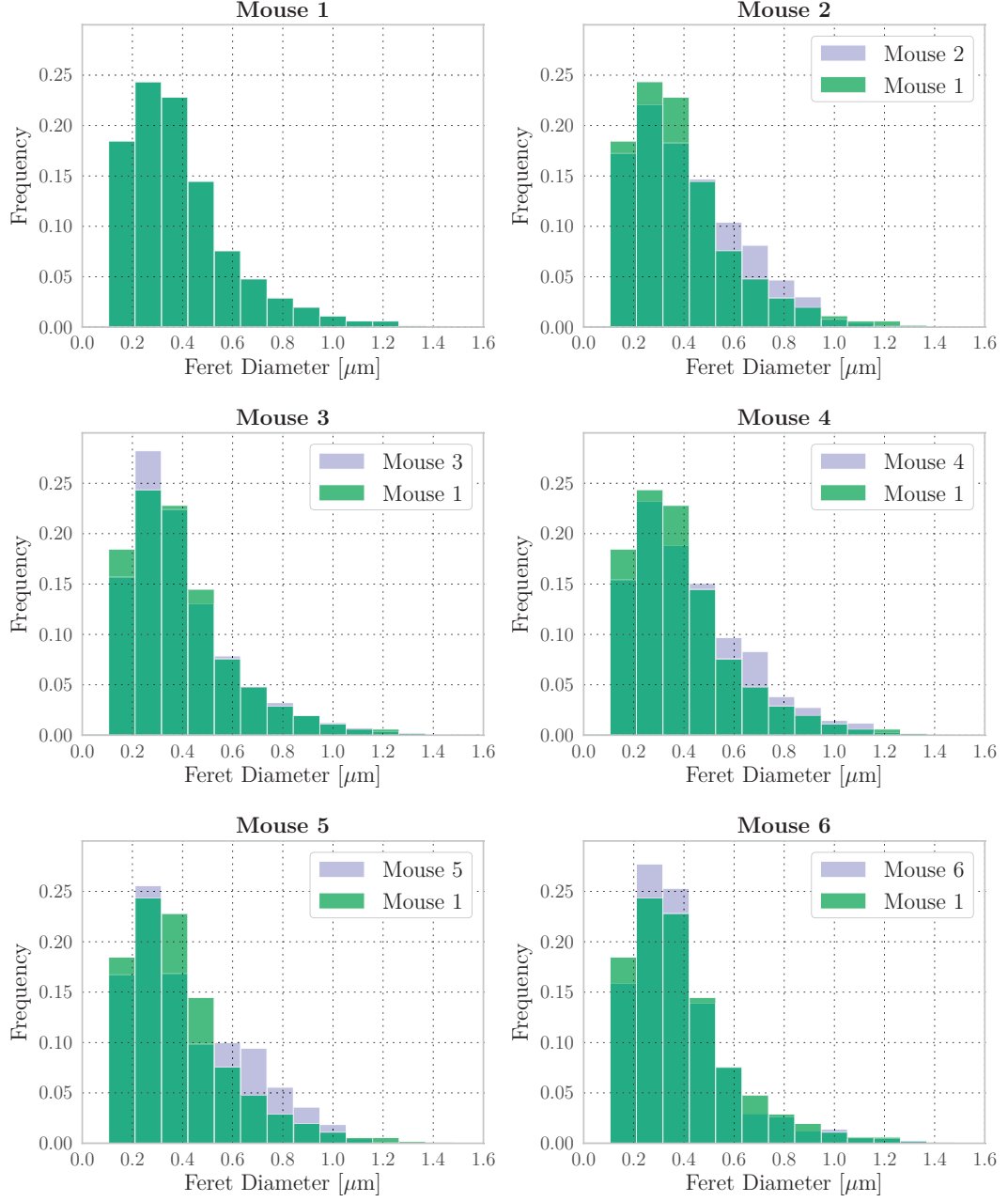


Figure 5.3: Our approach. Distribution of synaptic cleft size, measured through the Feret diameter of the automatic segmentations. For comparison we plot the distribution of Mouse 1 against the others. Aged mice 2, 4 and 5 present an relative increase in larger synapses, similar to that seen in the manual annotations in Fig. 5.2.

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

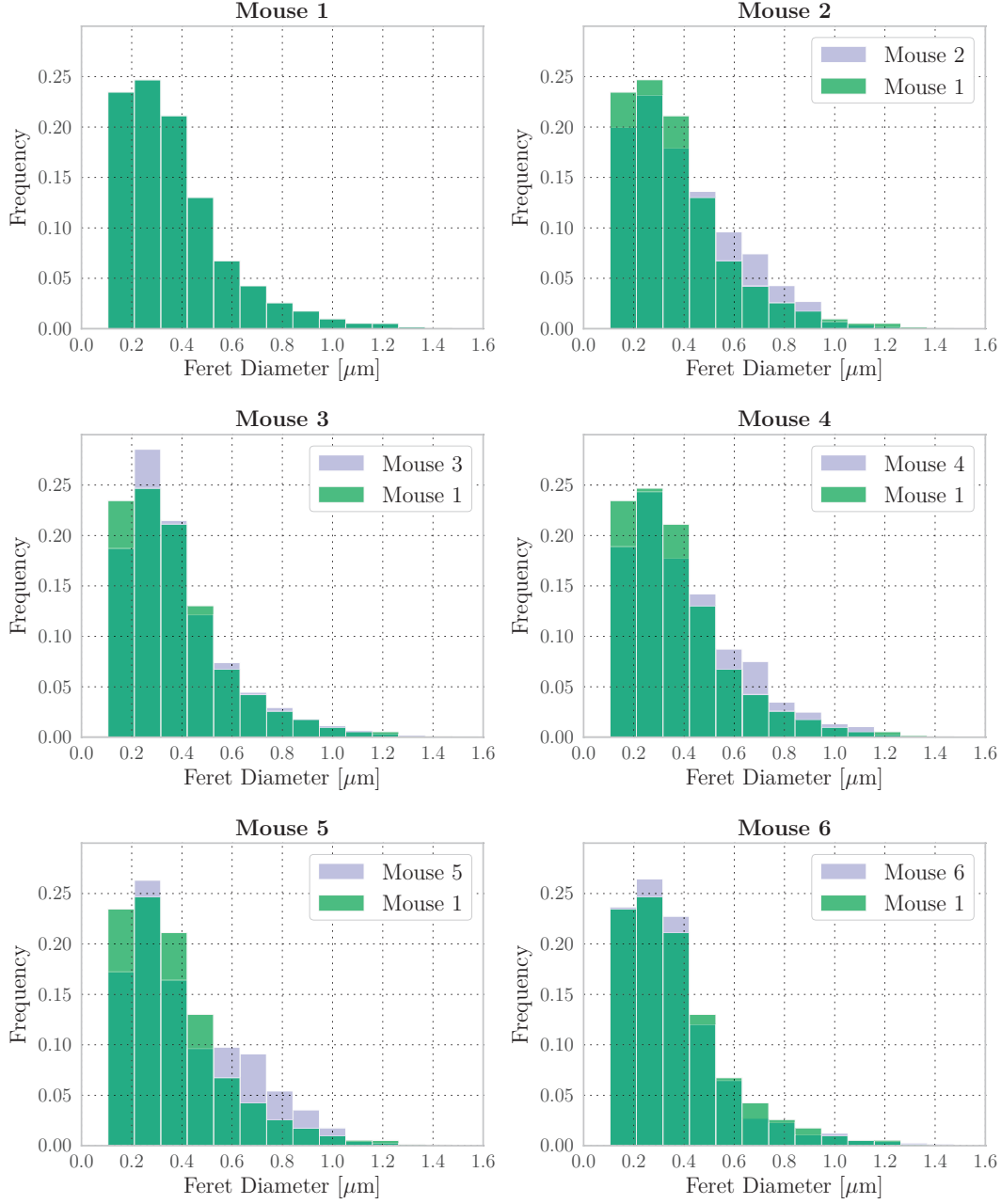


Figure 5.4: Ours corrected. Distribution of synaptic cleft size, measured through the Ferret diameter of the automatic segmentations, after manually removing false positives. For comparison we plot the distribution of Mouse 1 against the others. Aged mice 2, 4 and 5 present an relative increase in larger synapses, similar to that seen in the manual annotations in Fig. 5.2.

5.5 Exploting Segmentation Data: Synaptic Cleft Shape

So far we have concentrated on synapse count and a rough estimate of their size. While the results presented so far showcase some of the advantages of our approach, we have not yet fully exploited the availability of a voxel-wise segmentation output. In this section we will take advantage of such rich information to explore the correlation between synaptic cleft shape and aging.

5.5.1 Biological Motivation

So far we have observed an increase in larger synapses in the aged mice. It is known that synapse size is positively correlated with the number of receptors, and therefore the strength of the synaptic reaction [75]. Similarly, it has been shown that perforated synapses hold a higher number of AMPA receptors than non-perforated ones, and therefore may evoke larger post-synaptic responses [39]. On the other hand, smaller synapses are typically more labile and more plastic [71].

In light of this, in this section we analyze and compare synapse surface area and the proportion of perforated synapses among different stacks. To this end, we post-process the segmentation output of our approach and compute the synaptic surface area and number of holes in each synapse. We detail these steps next.

5.5.2 Data Processing

After running Connected Components on the segmentation output, we extract the surface area of each synapse and the number of perforations it presents. More specifically, we carry out the following steps:

1. **Rotation to canonical coordinate system.** For each synapse, we first compute its average orientation $\hat{\mathbf{n}}$, as the coordinate-wise mean of the highest-magnitude Hessian eigenvectors at each voxel in the segmentation, as in §2.4.3. We then rotate the synapse to a new coordinate system $\{x', y', z'\}$ such that z' is parallel to $\hat{\mathbf{n}}$. Note that there is a rotation ambiguity around x' and y' , but this does not pose a problem, as our measurements are independent of it.
2. **Surface fitting.** We fit a 5th-order polynomial to the segmented voxels in the $\{x', y', z'\}$ coordinate system, as a function of x' and y' . From this polynomial we

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

then extract the surface area of each synapse. Fitting a surface to the segmentation allows for measurements that are independent of the thickness of the segmented synapse, which may vary between synapses and on the training ground truth. An example of the fitted surface with a synapse with two perforations is shown in Fig. 5.5(c).

3. **Perforations.** To obtain the number of perforations and their geometry we project the voxels in labeled as positives by our segmentation method to the $\{x', y'\}$ plane, as shown in Fig. 5.5(d). We then count the number of holes in the projected image whose surface area is larger than 5000 nm^2 , since the segmentation output may contain imperfections in the form of small holes.

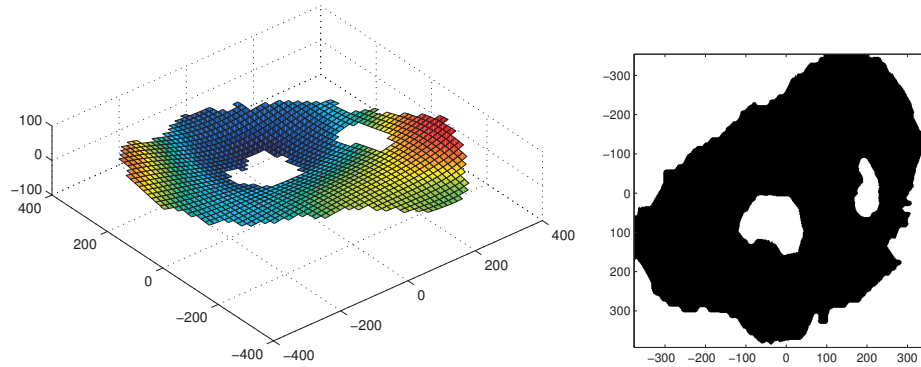
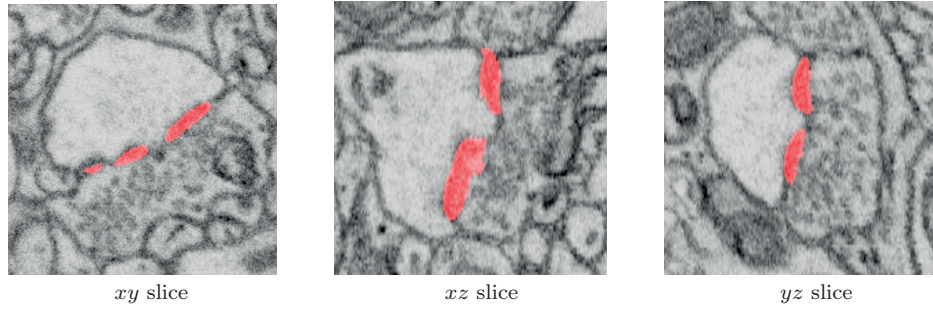
Processing each stack with the steps above took between 2 and 3 hours. We now analyze and discuss the statistics obtained in terms of surface area and number of perforations.

5.5.3 Synaptic Cleft Shape Analysis

Feret Diameter and Surface Area We first analyze whether the Feret diameter is a reasonable proxy for the synapse surface area, as the former is much easier to label and measure than the latter, for example by labeling spherical objects. To this end, in Fig. 5.6 we plot, for each detected synapse, its surface area versus the surface area of a circumference with a diameter equal to the measured Feret diameter of the synapse. The dashed black line represents the location where ideal measurements would fall if both were exactly equivalent.

As expected, the equivalent circumference area over-estimates the true surface area, and by a large factor in some cases. However, there is on average a strong correlation between both measurements, progressively deteriorating for larger synapses.

Surface Area Distribution Earlier we showed in Figures 5.4 and 5.3 that there is an increase in larger synapses in aged mice, when the extent of a synapse is measured with the Feret diameter. We now demonstrate that this phenomenon is also clearly visible with regards to synapse surface area, as shown in Fig. 5.7.



(c) Fitted surface on the $\{x', y', z'\}$ coordinate system. (d) Surface 2D projection.

Figure 5.5: Example of surface fitting on a synapse with two perforations. The top two rows show the original data and the segmentation obtained by our automated approach. The number of perforations and their area can be easily estimated from the 2D projection in (d).

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

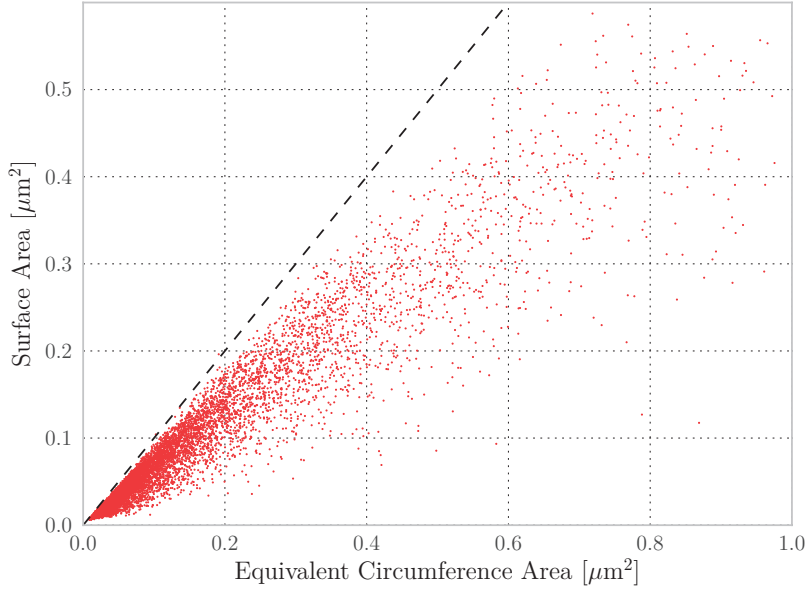


Figure 5.6: Feret diameter equivalent area and true synapse surface area. For each detected synapse, we plot its surface area versus the surface area of a circumference with a diameter equal to the measured Feret diameter of the synapse. The dashed black line represents the location where ideal measurements would fall if both were exactly equivalent.

Synapse Perforations Synaptic cleft perforations have been a subject of interest in the neuroscience literature, as it is believed that their presence is related to synaptic plasticity [20, 39, 51, 71]. The question we ask ourselves is whether there is a significant difference in the proportion of perforated synapses between the young adult and aged mice, that could indicate a link with synapse plasticity.

With this in mind, we plot in Fig. 5.8 the percentage of synapses with a single or two or more perforations. We observe a clear difference between aged (2,4,5) and young adult mice, characterized by a consistent increase in the proportion of perforated synapses in the aged ones.

From this observations, we proceeded to plot the distributions of perforated and non-perforated synapses according to synapse area, as shown in Fig. 5.9. As with Fig. 5.8, there is a marked difference between the two mouse groups, suggesting that aged mice hold a larger proportion of perforated synapses, in comparison to young adults.

5.5 Exploting Segmentation Data: Synaptic Cleft Shape

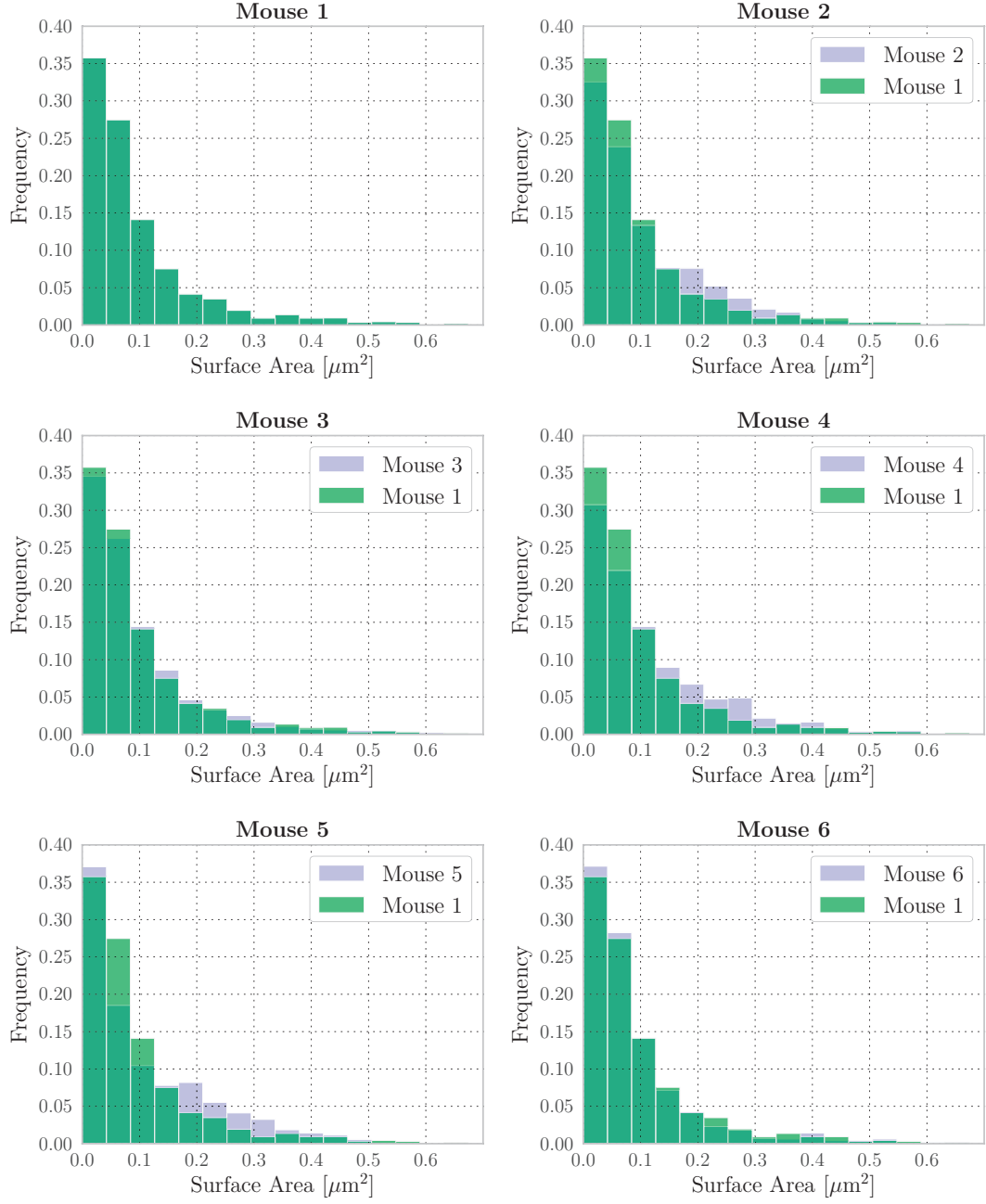


Figure 5.7: Distribution of synaptic cleft surface area of the polynomial surfaces fitted to the segmentation output *Ours corrected*. For comparison we plot the distribution of Mouse 1 against the others. Aged mice 2, 4 and 5 present an relative increase in larger synapses, similar to what is found with the Feret diameter distribution in Fig. 5.4.

5. SYNAPTIC STRUCTURE IN THE AGING MOUSE CORTEX

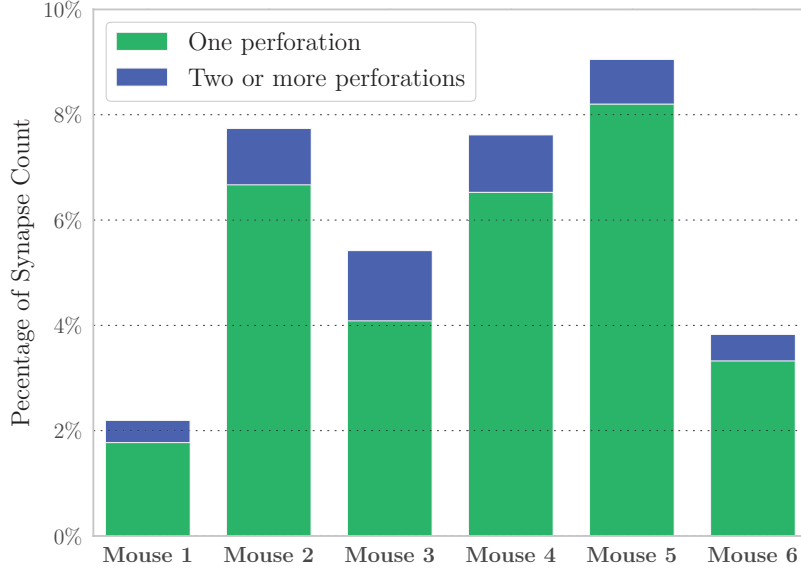


Figure 5.8: Proportion of perforated synapses in each stack. Our results show a higher proportion of perforated synapses in the aged mice, compared to young adults.

5.6 Conclusion

In this chapter we demonstrated the effectiveness and usefulness of our segmentation approach in reducing annotation effort and providing high quality segmentation maps. We showed that we can obtain equivalent results to that of experts, reducing labeling time by six times with respect to manual annotation. This process can also be further sped up, as prediction can be easily parallelized over multiple machines.

Furthermore, our approach outputs voxel-wise predictions that can be used to extract information such as synapse area and number of perforations, which would be infeasible to generate manually, given the large size of the stacks.

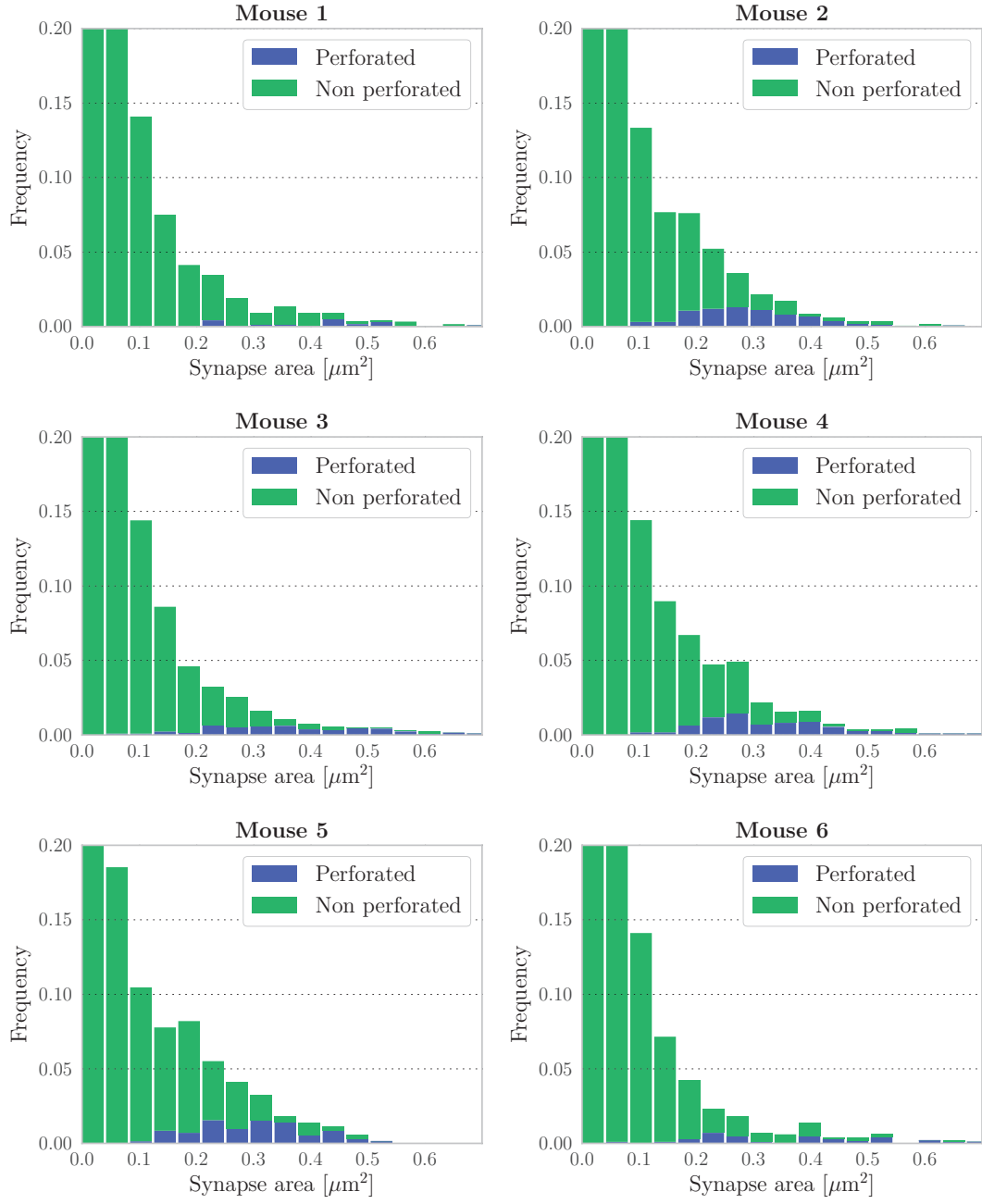


Figure 5.9: Distribution and relative quantity of perforated and non-perforated synapses. There is a significant increase in the proportion of perforated synapses in the aged mice, in comparison to the young adults.

Part IV

Final Words

CONCLUDING REMARKS

We began this thesis by identifying the need for automated methods for Electron Microscopy Imaging, as manual analysis is a slow, tedious, and error-prone procedure. Among the difficulties posed in such scenarios, we spotted three main challenges that define the requirements automated algorithms must meet to be of practical use. This led us to develop two new machine learning approaches that outperform state-of-the-art methods, scale well to large stacks, while being computationally efficient for both training and prediction.

In Chapter 2 we introduced a new automated approach for synapse segmentation in Electron Microscopy stacks. The proposed method relies on image features particularly designed to take spatial context into account. These features are inspired by the criteria human experts use to identify synapses, such as densities on the pre-synaptic and post-synaptic membranes, vesicles in the pre-synaptic axon terminal and the presence of a synaptic cleft. However, unlike other approaches, the number of our features is not fixed to a few tens or hundreds, but is instead in the order of hundred of thousands, allowing for a flexible algorithm that gracefully adapts to the training data. Rather than manually selecting a subset of the features, we rely on boosting to choose the most relevant ones and pool them adequately to optimize prediction performance. Our approach successfully distinguishes synapses from other organelles that appear within an EM stack, including those whose local appearance is similar. Moreover, the proposed approach flawlessly identifies synaptic orientation, a feature unique to our approach that had remained unexplored so far.

6. CONCLUDING REMARKS

We then showed in Chapter 3 that mislabeling polarities, a task that hitherto involved manual labeling, can have significant detrimental performance effects. This motivated the need for automatic polarity estimation during the training process, that led to the development of a new Multiple Instance Learning (MIL) algorithm in Chapter 3. Our boosting-based MIL approach can flawlessly estimate the polarities of all synapses at training time, dropping the need for manual polarity annotations, and performing as well or better than the existing baselines.

In Chapter 4 we tackled the problem of acquisition variability and proposed a new method for domain adaptation. Our approach simultaneously learns the decision boundary and a non-parametric estimation of the transformation between domains. The proposed algorithm effectively leverages labeled examples across different acquisitions, significantly reducing annotation effort. We evaluated our approach on four Electron and Light Microscopy stacks where annotation is burdensome and costly, demonstrating its effectiveness and improvement over the state-of-the-art.

In Chapter 5 we applied our synapse segmentation approach towards answering biologically meaningful questions about how aging affects the brain. To this end, we compared the structure and shape of synaptic densities in adult and aged mice. We showed that we can generate segmentations that agree with expert annotations, while requiring very little labeling effort. Our results show that aged mice enjoy a larger proportion of large synapse densities in comparison to adult mice. Moreover, our approach generates a per-voxel segmentation output from which rich morphological information can be extracted, such as the area of each synapse and the number of perforations within them.

The presented methods help advance the efficient analysis of large microscopy stacks, delivering neuroscientists with powerful tools to explore and analyze large volumes with little annotation effort.

6.1 Future Work

In this section we discuss possible extensions and improvements to our segmentation and domain adaptation frameworks.

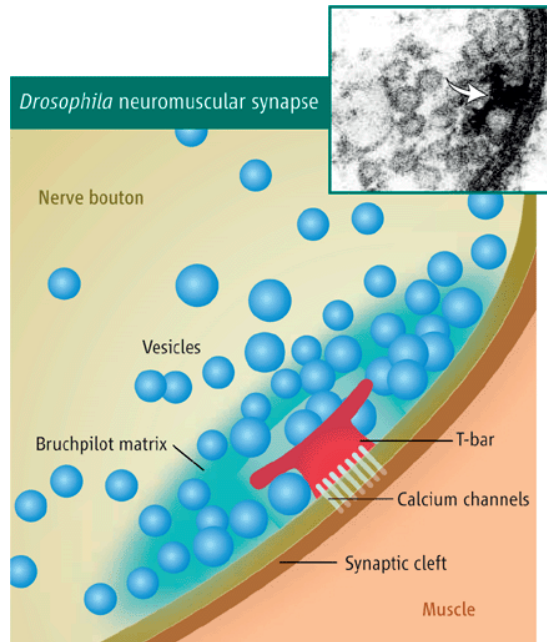


Figure 6.1: Illustration of a T-Bar synapse, such as the ones found in the fly brain. The image on the top right corner shows an example of the appearance of a T-Bar synapse in EM. Reproduced from [7].

6.1.1 Automated Segmentation of Organelles

Fly synapse segmentation. Our automated approach focused on mammalian synaptic densities that exhibit a membrane-like appearance, similar to that of a disk. However, there has also been great interest in detecting synapses in insects such as flies [46, 85] to shed light into their connectome. Though our approach could be directly applied to fly synapses, their appearance is strikingly different to that of mammals, as shown in Fig. 6.1. An important consequence of this significant difference is that the orientation estimates computed with the Hessian eigenvectors in Chapter 2 may be incorrect, making learning harder. A possible line of future research could involve designing an appropriate orientation estimate, or either learning it from user annotations. Another difficulty in fly EM stacks is that the exact extent of a synapse is hard to distinguish. Though this also happens with mammalian synapses, we have observed that it is much more pronounced in flies, making voxel-wise labeling a difficult and ill-posed task. Therefore, it would be worth investigating whether our segmentation approach can be trained directly on detection ground truth (e.g. bounding boxes). We believe

6. CONCLUDING REMARKS

this could be achieved through Multiple Instance Learning, along the lines of the work of Xu et al. [110].

Membrane segmentation. We believe our approach can also be used to segment membranes, although it may require modifications to compete with existing approaches such as CNN-based ones, for example the U-Net [88]. For membrane segmentation, CNN-based approaches trained on a large amount of annotated data have shown much higher performance in comparison to algorithms that use hand-crafted features. One possible explanation for such behavior is that membranes can take very different appearance, making feature design difficult. However, we believe this gap can be bridged by allowing for novel context features that better capture the geometry of EM membranes, improving performance and reducing the need for large amounts of training data.

Biologically-meaningful measures. In Chapter 5 we extracted measures such as surface area, Feret diameter and number of perforations for each synapse. We think it is worth exploring new measures and their link to synapse plasticity and synaptogenesis. For example, a measure of the deformation of a synapse (e.g. how deformed it is with respect to a flat membrane), or their convexity or concavity, could be linked to its plasticity or connection strength and aging [71].

6.1.2 Domain Adaptation

Non coordinate-wise transformations. Our domain adaptation method presented in Chapter 4 assumed that there is a one-to-one feature correspondence between domains. Though this may be a valid assumption for some applications, it may not be correct for other problems (e.g. rotations in feature space or histogram bin features). To deal with the latter, our algorithm needs to be modified to allow for more complex transformations. This may require estimating the transformations parametrically to be computationally feasible. Such paradigm shift would entail developing new optimization methods to estimate the transformation parameters and the decision boundary simultaneously.

Correspondence-aided Domain Adaptation. Our method looks forward to estimating coordinate-wise transformations based solely on discriminative information (i.e. isolated labels in both domains). This may be a difficult task, especially when the underlying transformations are too complex to be inferred from a few training points. One way to alleviate this situation would be to include labeled correspondences between domains to *guide* learning and avoid overfitting to the target domain. This could be implemented by modifying the loss function to force scores of correspondences in the two domains be similar.

REFERENCES

- [1] ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P. & SUESSTRUNK, S. (2012). SLIC Superpixels Compared to State-of-the-Art Superpixel Methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 2274–2282. 25, 37
- [2] ALI, K., FLEURET, F., HASLER, D. & FUA, P. (2012). A Real-Time Deformable Detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **34**, 225–239. 18, 24, 25, 65
- [3] ANDO, R.K. & ZHANG, T. (2005). A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *Journal of Machine Learning Research*, **6**, 1817–1853. 57, 60
- [4] ANDRES, B., KOETHE, U., HELMSTAEDTER, M., DENK, W. & HAMPRECHT, F. (2008). Segmentation of SBFSEM Volume Data of Neural Tissue by Hierarchical Classification. In *DAGM Symposium on Pattern Recognition*, 142–152. 15
- [5] ANDREWS, S., TSOCHANTARIDIS, I. & HOFMANN, T. (2002). Support Vector Machines for Multiple-Instance Learning. In *Advances in Neural Information Processing Systems*, 561–568. 41, 43
- [6] ASCOLI, G., SVOBODA, K. & LIU, Y. (2010). Digital Reconstruction of Axonal and Dendritic Morphology DIADEM Challenge. 69
- [7] ATWOOD, H.L. (2006). Gatekeeper at the synapse. *Science*. 107
- [8] BABENKO, B., DOLLÁR, P., TU, Z. & BELONGIE, S. (2008). Simultaneous Learning and Alignment: Multi-Instance and Multi-Pose Learning. In *Workshop on Faces in Real-Life Images: Detection, Alignment, and Recognition*. 41, 43, 45, 48
- [9] BACH, F.R. & JORDAN, M.I. (2002). Kernel Independent Component Analysis. *Journal of Machine Learning Research*, **3**, 1–48. 58, 59, 61, 71, 74
- [10] BARTESAGHI, A., SAPIRO, G. & SUBRAMANIAM, S. (2005). An Energy-Based Three-Dimensional Segmentation Approach for the Quantitative Interpretation of Electron Tomograms. *IEEE Transactions on Image Processing*, **14**, 1314–1323. 14

REFERENCES

- [11] BAXTER, J. (2000). A Model of Inductive Bias Learning. *Journal of Artificial Intelligence Research*. 60
- [12] BECKER, C., ALI, K., KNOTT, G. & FUA, P. (2012). Learning Context Cues for Synapse Segmentation in EM Volumes. In *Conference on Medical Image Computing and Computer Assisted Intervention*. 4, 21
- [13] BECKER, C., ALI, K., KNOTT, G. & FUA, P. (2013). Learning Context Cues for Synapse Segmentation. *IEEE Transactions on Medical Imaging*. 4
- [14] BECKER, C., CHRISTOUDIAS, M. & FUA, P. (2013). Non-Linear Domain Adaptation with Boosting. In *Advances in Neural Information Processing Systems*. 4
- [15] BECKER, C., CHRISTOUDIAS, M. & FUA, P. (2015). Domain Adaptation for Microscopy Imaging. *IEEE Transactions on Medical Imaging*. 4
- [16] BI, J., XIONG, T., YU, S., DUNDAR, M. & RAO, R.B. (2008). An Improved Multi-Task Learning Approach with Applications in Medical Diagnosis. In *Machine Learning and Knowledge Discovery in Databases*. 60, 62
- [17] BOURDEV, L. & BRANDT, J. (2005). Robust Object Detection via Soft Cascade. In *Conference on Computer Vision and Pattern Recognition*, 236–243. 38
- [18] BURKE, S.N. & BARNES, C.A. (2006). Neural plasticity in the ageing brain. *Nature Reviews Neuroscience*. 85
- [19] CARLBOM, I., TERZOPOULOS, D. & HARRIS, K. (1994). Computer-Assisted Registration, Segmentation, and 3D Reconstruction from Images of Neuronal Tissue Sections. *IEEE Transactions on Medical Imaging*, **13**, 351–362. 13, 14
- [20] CARLIN, R.K. & SIEKEVITZ, P. (1983). Plasticity in the central nervous system: do synapses divide? *Proceedings of the National Academy of Sciences*. 98
- [21] CARUANA, R. (1997). Multitask Learning. *Machine Learning*, **28**. 57, 60
- [22] CARUANA, R. & NICULESCU-MIZIL, A. (2006). An Empirical Comparison of Supervised Learning Algorithms. In *International Conference on Machine Learning*. 64
- [23] CHAPELLE, O., SHIVASWAMY, P., VADREVU, S., WEINBERGER, K., ZHANG, Y. & TSENG, B. (2010). Boosted Multi-Task Learning. *Machine Learning*. 57, 58, 59, 60, 61, 62, 63, 64, 71, 78, 79
- [24] CHEN, K., BAI, J. & ZHENG, Z. (2011). Ranking Function Adaptation with Boosting Trees. *ACM Transactions on Information Systems*. 59, 61, 71, 78, 79
- [25] DAI, W., YANG, Q., XUE, G. & YU, Y. (2007). Boosting for Transfer Learning. In *Machine Learning*, 193–200. 61, 62
- [26] DAUMÉ, H. (2009). Bayesian Multitask Learning with Latent Hierarchies. In *Uncertainty in Artificial Intelligence*. 60

-
- [27] DEFELIPE, J., MARCO, P., BUSTURIA, I. & MERCHÁN-PÉREZ, A. (1999). Estimation of the Number of Synapses in the Cerebral Cortex: Methodological Considerations. *Cerebral Cortex*, **9**, 722–732. 36
- [28] DIETTERICH, T.G., LATHROP, R.H. & LOZANO-PÉREZ, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, **89**. 43
- [29] DUNDAR, M.M., BADVE, S., RAYKAR, V.C., JAIN, R.K., SERTEL, O. & GURCAN, M.N. (2010). A Multiple Instance Learning Approach Toward Optimal Classification of Pathology Slides. In *International Conference on Pattern Recognition*, 2732–2735. 41, 43
- [30] EK, C.H., TORR, P.H. & LAWRENCE, N.D. (2008). Ambiguity Modelling in Latent Spaces. In *Machine Learning in Medical Imaging*. 58
- [31] EVERINGHAM, M., L. VAN GOOL AND, C.W., WINN, J. & ZISSERMAN, A. (2010). The Pascal Visual Object Classes Challenge (VOC2010) Results. 30
- [32] EVGENIOU, T., MICCHELLI, C. & PONTIL, M. (2005). Learning Multiple Tasks with Kernel Methods. *Journal of Machine Learning Research*, **6**. 57, 60, 61, 63
- [33] FIALA, J.C. (2005). Reconstruct: A Free Editor for Serial Section Microscopy. *Journal of microscopy*, **218**, 52–61. 13
- [34] FIALA, J.C. & HARRIS, K.M. (2001). Extending Unbiased Stereology of Brain Ultrastructure to Three-Dimensional Volumes. *Journal of the American Medical Informatics Association*, **8**, 1–16. 13
- [35] FLEURET, F. & GEMAN, D. (2008). Stationary Features and Cat Detection. *Journal of Machine Learning Research*, **9**, 2549–2578. 18, 24, 25
- [36] FREUND, Y. & SCHAPIRE, R. (1996). Experiments with a New Boosting Algorithm. In *International Conference on Machine Learning*, 148–156. 18, 19, 20, 23
- [37] FREUND, Y. & SCHAPIRE, R. (1999). A Short Introduction to Boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(5):771–780. 66
- [38] FUNG, G., DUNDAR, M., KRISHNAPURAM, B. & RAO, R. (2007). Multiple Instance Learning for Computer Aided Diagnosis. *Advances in Neural Information Processing Systems*, **19**, 425. 41, 43
- [39] GANESHINA, O., BERRY, R.W., PETRALIA, R.S., NICHOLSON, D.A. & GEINISMAN, Y. (2004). Differences in the expression of ampa and nmda receptors between axospinous perforated and nonperforated synapses are related to the configuration and size of post-synaptic densities. *Journal of Comparative Neurology*. 95, 98
- [40] GOPALAN, R., LI, R. & CHELLAPPA, R. (2011). Domain Adaptation for Object Recognition: An Unsupervised Approach. In *International Conference on Computer Vision*. 61

REFERENCES

- [41] GRILLO, F., SONG, S., RUIVO, L., HUANG, L., GAO, G., KNOTT, G., MACO, B., FERRETTI, V., THOMPSON, D., LITTLE, G. *et al.* (2013). Increased Axonal Bouton Dynamics in the Aging Mouse Cortex. *Proceedings of the National Academy of Sciences USA*, **110**, 1514–1523. 85
- [42] HARRIS, K.M. & WEINBERG, R.J. (2012). Ultrastructure of synapses in the mammalian brain. *Cold Spring Harbor perspectives in biology*, **4**. 2
- [43] HASTIE, T., TIBSHIRANI, R. & FRIEDMAN, J. (2001). *The Elements of Statistical Learning*. Springer. 58, 61, 63, 64, 65, 70, 81
- [44] HEIMANN, T., MOUNTNEY, P., JOHN, M. & IONASEC, R. (2013). Learning Without Labeling: Domain Adaptation for Ultrasound Transducer Localization. In *Conference on Medical Image Computing and Computer Assisted Intervention*, 49–56. 61
- [45] HEROLD, J., SCHUBERT, W. & NATTKEMPER, T. (2010). Automated Detection and Quantification of Fluorescently Labeled Synapses in Murine Brain Tissue Sections for High Throughput Applications. *Journal of Biotechnology*, **149**, 299–309. 12
- [46] HUANG, G.B. & PLAZA, S. (2014). Identifying synapses using deep and wide multiscale recursive networks. *arXiv preprint arXiv:1409.1789*. 2, 16, 107
- [47] JACOB, L., BACH, F. & VERT, J.P. (2008). Clustered Multi-Task Learning: A Convex Formulation. In *Advances in Neural Information Processing Systems*. 60
- [48] JAIN, V., MURRAY, J., ROTH, F., TURAGA, S., ZHIGULIN, V., BRIGGMAN, K., HELMSTÄEDTER, M., DENK, W. & SEUNG, H. (2007). Supervised Learning of Image Restoration with Convolutional Networks. In *International Conference on Computer Vision*, 1–8. 13, 15, 33
- [49] JIANG, J. (2008). A Literature Survey on Domain Adaptation of Statistical Classifiers. Tech. rep., University of Illinois at Urbana-Champaign. 57, 60
- [50] JIE, B., ZHANG, D., CHENG, B. & SHEN, D. (2013). Manifold Regularized Multi-Task Feature Selection for Multi-Modality Classification in Alzheimer’s Disease. In *Conference on Medical Image Computing and Computer Assisted Intervention*. 62
- [51] JONES, D., ITARAT, W. & CALVERLEY, R. (1991). Perforated synapses and plasticity. *Molecular neurobiology*. 98
- [52] JURRUS, E., PAIVA, A., WATANABE, S., ANDERSON, J., WHITAKER, R., JONES, B., MARC, R. & TASDIZEN, T. (2010). Detection of Neuron Membranes in Electron Microscopy Images Using a Serial Neural Network Architecture. *Medical Image Analysis*, **14**, 770–783. 14, 16, 17
- [53] K. REIN, K., ZÖCKLER, M., T, M.M., GRÜBEL, C. & HEISENBERG, M. (2002). The *Drosophila* Standard Brain. *Current Biology*, **12**, 227–231. 2

-
- [54] KAYNIG, V., FUCHS, T. & BUHMANN, J. (2010). Neuron Geometry Extraction by Perceptual Grouping in ssTEM Images. In *Conference on Computer Vision and Pattern Recognition*, 2902–09. 14, 15
 - [55] KNOTT, A., PERKINS, G., SCHWARZENBACHER, R. & BOSSY-WETZEL, E. (2008). Mitochondrial Fragmentation in Neurodegeneration. *Nature Reviews. Neuroscience*, **9**, 505–18. 1
 - [56] KNOTT, G., ROSSET, S. & CANTONI, M. (2011). Focussed ion beam milling and scanning electron microscopy of brain tissue. *Journal of visualized experiments: JoVE*. 86
 - [57] KRESHUK, A., STRAEHLE, C.N., SOMMER, C., KOETHE, U., KNOTT, G. & HAMPRECHT, F. (2011). Automated Segmentation of Synapses in 3D EM Data. In *International Symposium on Biomedical Imaging*. 12, 13, 16, 17, 20, 21, 26, 28, 30, 31, 32, 33, 35, 36, 37, 38
 - [58] KRESHUK, A., KOETHE, U., PAX, E., BOCK, D.D. & HAMPRECHT, F.A. (2014). Automated detection of synapses in serial section transmission electron microscopy image stacks. *PloS one*. 12, 17
 - [59] KRUMMENACHER, G., ONG, C.S. & BUHMANN, J. (2013). Ellipsoidal multiple instance learning. In *International Conference on Machine Learning*. 43
 - [60] KULIS, B., SAENKO, K. & DARRELL, T. (2011). What You Saw is Not What You Get: Domain Adaptation Using Asymmetric Kernel Transforms. In *Conference on Computer Vision and Pattern Recognition*. 60, 71, 74
 - [61] KUMAR, A. & DAUMÉ, H. (2012). Learning Task Grouping and Overlap in Multi-Task Learning. In *International Conference on Machine Learning*. 60
 - [62] LEE, P., CHUANG, C., CHIANG, A. & CHING, Y. (2012). High-Throughput Computer Method for 3D Neuronal Structure Reconstruction from the Image Stack of the Drosophila Brain and Its Applications. *PLoS Comput Biol*, **8**, 1002658. 1
 - [63] LEISTNER, C., SAFFARI, A. & BISCHOF, H. (2010). Miforests: multiple-instance learning with randomized trees. In *European Conference on Computer Vision*, Springer. 43
 - [64] LIANG, J., MCINERNEY, T. & TERZOPOULOS, D. (2006). United Snakes. *Medical Image Analysis*, **10**, 215–233. 13, 14
 - [65] LINDNER, C., THIAGARAJAH, S., WILKINSON, J., LOUGHLIN, J., WALLIS, G. & COOTES, T. (2013). Fully Automatic Segmentation of the Proximal Femur Using Random Forest Regression Voting. *IEEE Transactions on Medical Imaging*. 59
 - [66] LUCCHI, A., SMITH, K., ACHANTA, R., KNOTT, G. & FUA, P. (2012). Supervoxel-Based Segmentation of Mitochondria in EM Image Stacks with Learned Shape Features. *IEEE Transactions on Medical Imaging*, **31**, 474–486. 13, 15, 17, 30, 31

REFERENCES

- [67] LUCCHI, A., BECKER, C., MARQUEZ-NEILA, P. & FUA, P. (2014). Exploiting Enclosing Membranes and Contextual Cues for Mitochondria Segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*. 13
- [68] MACKE, J., MAACK, N., GUPTA, R., DENK, W., SCHÖLKOPF, B. & BORST, A. (2008). Contour-Propagation Algorithms for Semi-Automated Reconstruction of Neural Processes. *Journal of Neuroscience Methods*, **167**, 349–57. 14
- [69] MANGASARIAN, O.L. & WILD, E.W. (2008). Multiple instance classification via successive linear programming. *Journal of Optimization Theory and Applications*. 43
- [70] MANNELLA, C., MARKO, M. & BUTTLE, K. (1997). Reconsidering Mitochondrial Structure: New Views of an Old Organelle. *Trends Biochem. Sci.*, **15**, 37–38. 1
- [71] MARRONE, D.F. & PETIT, T.L. (2002). The role of synaptic morphology in neural plasticity: structural interactions underlying synaptic power. *Brain research reviews*. 95, 98, 108
- [72] MARSH, B., MASTRONARDE, D., BUTTLE, K., HOWELL, K. & MCINTOSH, J. (2001). Organellar Relationships in the Golgi Region of the Pancreateic Beta Cell Line, Hit-T15, Visualized by High Resolution Electron Tomography. *PNAS*, **98**, 2399–2406. 13
- [73] MEMISEVIC, R., SIGAL, L. & FLEET, D.J. (2012). Shared Kernel Information Embedding for Discriminative Inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 59, 60, 61, 66
- [74] MERCHÁN-PÉREZ, A., RODRIGUEZ, J., ALONSO-NANCLARES, L., SCHERTEL, A. & DEFELIPE, J. (2009). Counting Synapses Using FIB/SEM Microscopy: A True Revolution for Ultrastructural Volume Reconstruction. *Frontiers in Neuroanatomy*, **3**, 18. 1
- [75] MEYER, D., BONHOEFFER, T. & SCHEUSS, V. (2014). Balance and stability of synaptic structures during synaptic plasticity. *Neuron*. 95
- [76] MISHCHENKO, Y., HU, T., SPACEK, J., MENDENHALL, J., HARRIS, K. & CHKLOVSKII, D. (2010). Ultrastructural Analysis of Hippocampal Neuropil from the Connectomics Perspective. *Neuron*, **67**, 1009–1020. 12, 14
- [77] MORALES, J., ALONSO-NANCLARES, L., RODRÍGUEZ, J., DEFELIPE, J., RODRÍGUEZ, A. & MERCHÁN-PÉREZ, A. (2011). Espina: A Tool for the Automated Segmentation and Counting of Synapses in Large Stacks of Electron Microscopy Images. *Frontiers in Neuroanatomy*. 1, 3, 55
- [78] N. VU & B. MANJUNATH (2008). Graph Cut Segmentation of Neuronal Structures from Transmission Electron Micrographs. In *International Conference on Image Processing*, 725–728. 14
- [79] NARASIMHA, R., OUYANG, H., GRAY, A., McLAUGHLIN, S. & SUBRAMANIAM, S. (2009). Automatic Joint Classification and Segmentation of Whole Cell 3D Images. *Pattern Recognition*, **42**, 1067–1079. 14, 15, 17

-
- [80] NGUYEN, H. & JI, Q. (2008). Shape-Driven Three-Dimensional Watersnake Segmentation of Biological Membranes in Electron Tomography. *IEEE Transactions on Medical Imaging*, **27**, 616–628. 15
 - [81] PADFIELD, D., RITTSCHER, J., THOMAS, N. & ROYSAM, B. (2009). Spatio-Temporal Cell Cycle Phase Analysis Using Level Sets and Fast Marching Methods. *Medical Image Analysis*, **13**, 143–155. 14
 - [82] PAPANDREOU, G., CHEN, L.C., MURPHY, K. & YUILLE, A.L. (2015). Weakly-And Semi-Supervised Learning of a DCNN for Semantic Image Segmentation. *arXiv preprint arXiv:1502.02734*. 43
 - [83] PATHAK, D., SHELHAMER, E., LONG, J. & DARRELL, T. (2014). Fully Convolutional Multi-Class Multiple Instance Learning. *International Conference for Learning Representations*. 43
 - [84] PINHEIRO, P.O. & COLLOBERT, R. (2015). From Image-Level to Pixel-Level Labeling with Convolutional Networks. In *Conference on Computer Vision and Pattern Recognition*. 43
 - [85] PLAZA, S.M., PARAG, T., HUANG, G.B., OLBRIS, D.J., SAUNDERS, M.A. & RIVLIN, P.K. (2014). Annotating synapses in large em datasets. *arXiv preprint arXiv:1409.1801*. 2, 28, 107
 - [86] POOLE, A., THOMAS, R., ANDREWS, L., MCBRIDE, H., WHITWORTH, A. & PALLANCK, L. (2008). The Pink1/parkin Pathway Regulates Mitochondrial Morphology. *Proceedings of the National Academy of Sciences of the United States of America*, **105**, 1638–43. 1
 - [87] RONCAL, W.G., KAYNIG-FITTKAU, V., KASTHURI, N., BERGER, D., VOGELSTEIN, J.T., FERNANDEZ, L.R., LICHTMAN, J.W., VOGELSTEIN, R.J., PFISTER, H. & HAGER, G.D. (2014). Volumetric exploitation of synaptic information using context localization and evaluation. *arXiv preprint arXiv:1403.3724*. 16
 - [88] RONNEBERGER, O., FISCHER, P. & BROX, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*, Springer International Publishing. 13, 16, 26, 30, 37, 38, 108
 - [89] ROSSET, S., ZHU, J. & HASTIE, T. (2004). Boosting as a Regularized Path to a Maximum Margin Classifier. *Journal of Machine Learning Research*. 64
 - [90] SAENKO, K., KULIS, B., FRITZ, M. & DARRELL, T. (2010). Adapting Visual Category Models to New Domains. In *European Conference on Computer Vision*, 213–226. 60, 71, 74
 - [91] SALZMANN, M., EK, C.H., URTASUN, R. & DARRELL, T. (2010). Factorized Orthogonal Latent Spaces. In *International Conference on Artificial Intelligence and Statistics*. 59, 61

REFERENCES

- [92] SCHINDELIN, J., ARGANDA-CARRERAS, I., FRISE, E., KAYNIG, V., LONGAIR, M., PIETZSCH, T., PREIBISCH, S., RUEDEN, C., SAALFELD, S. & SCHMID, B. (2012). Fiji: an open-source platform for biological-image analysis. *Nature Methods*, **9**, 676–682, code available at <http://pacific.mpi-cbg.de>. 1, 55
- [93] SCHMID, B., SCHINDELIN, J., CARDONA, A., LONGAIR, M. & HEISENBERG, M. (2010). A High-Level 3D Visualization API for Java and ImageJ. *BMC Bioinformatics*, **11**, 274. 1, 28, 86
- [94] SHON, A.P., GROCHOW, K., HERTZMANN, A. & RAO, R.P.N. (2006). Learning Shared Latent Structure for Image Synthesis and Robotic Imitation. In *Advances in Neural Information Processing Systems*, 1233–1240. 60, 61
- [95] SOMMER, C., STRAEHLE, C., KOETHE, U. & HAMPRECHT, F. (2011). ilastik: Interactive Learning and Segmentation Toolkit. In *International Symposium on Biomedical Imaging*. 1, 3, 55
- [96] THEVENAZ, P., RUTTIMANN, U.E. & UNSER, M. (1998). A pyramid approach to sub-pixel registration based on intensity. *Image Processing, IEEE Transactions on*. 86
- [97] TU, Z., ZHENG, S., YUILLE, A., REISS, A., DUTTON, R., LEE, A., GALABURDA, A., DINOV, I., THOMPSON, P. & TOGA, A. (2007). Automated Extraction of the Cortical Sulci Based on a Supervised Learning Approach. *IEEE Transactions on Medical Imaging*, **26**, 541–552. 59
- [98] TURAGA, S., MURRAY, J., JAIN, V., ROTH, F., HELMSTAEDTER, M., BRIGGMAN, K., DENK, W. & SEUNG, H. (2010). Convolutional Networks Can Learn to Generate Affinity Graphs for Image Segmentation. *Neural Computation*, **22**, 511–538. 15, 33
- [99] TURETKEN, E., BENMANSOUR, F. & FUA, P. (2012). Automated Reconstruction of Tree Structures Using Path Classifiers and Mixed Integer Programming. In *Conference on Computer Vision and Pattern Recognition*. 69, 70
- [100] VAN OPBROEK, A., IKRAM, M.A., VERNOOIJ, M.W. & DE BRUIJNE, M. (2013). A Transfer-Learning Approach to Image Segmentation Across Scanners by Maximizing Distribution Similarity. In *Machine Learning in Medical Imaging*. 60, 62
- [101] VENKATARAJU, K., PAIVA, A., JURRUS, E. & TASDIZEN, T. (2009). Automatic Markup of Neural Cell Membranes Using Boosted Decision Stumps. In *IEEE Symposium on Biomedical Imaging: From Nano to Macro*, 1039–42. 16, 17
- [102] VEZHNEVETS, A. & BUHMANN, J.M. (2010). Towards Weakly Supervised Semantic Segmentation by Means of Multiple Instance and Multitask Learning. In *Conference on Computer Vision and Pattern Recognition*. 43
- [103] VEZHNEVETS, A., BUHMANN, J. & FERRARI, V. (2012). Active Learning for Semantic Segmentation with Expected Change. In *Conference on Computer Vision and Pattern Recognition*. 43

-
- [104] VIOLA, P. & JONES, M. (2001). Rapid Object Detection Using a Boosted Cascade of Simple Features. In *Conference on Computer Vision and Pattern Recognition*. 65
- [105] VIOLA, P., PLATT, J. & ZHANG, C. (2005). Multiple Instance Boosting for Object Detection. In *Advances in Neural Information Processing Systems*, 1417–1424. 41, 43
- [106] VITALADEVUNI, S., MISHCHENKO, Y., GENKIN, A., CHKLOVSKII, D. & HARRIS, K. (2008). Mitochondria Detection in Electron Microscopy Images. In *Workshop on Microscopic Image Analysis with Applications in Biology*. 14, 15, 16
- [107] WANG, B., PRASTAWA, M., SAHA, A., AWATE, S., IRIMIA, A., CHAMBERS, M., VESPA, P., HORN, J.V., PASCUCCI, V. & GERIG, G. (2013). Modeling 4D Changes in Pathological Anatomy Using Domain Adaptation: Analysis of TBI Imaging Using a Tumor Database. In *Multimodal Brain Image Analysis*, 31–39. 60, 61
- [108] WHITE, J.G., SOUTHGATE, E., THOMSON, J.N. & BRENNER, S. (1986). The Structure of the Nervous System of the Nematode *Caenorhabditis Elegans*. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, **314**, 1–340. 14
- [109] XU, J., WU, Q., ZHANG, J. & TANG, Z. (2012). Object Detection Based on Co-Occurrence GMuLBP Features. In *Multimedia and Expo*. 43
- [110] XU, Y., ZHANG, J., ERIC, I., CHANG, C., LAI, M. & TU, Z. (2012). Context-Constrained Multiple Instance Learning for Histopathology Image Segmentation. In *Conference on Medical Image Computing and Computer Assisted Intervention*, 623–630, Springer. 41, 43, 44, 108
- [111] XU, Y., ZHU, J.Y., CHANG, E. & TU, Z. (2012). Multiple Clustered Instance Learning for Histopathology Cancer Image Classification, Segmentation and Clustering. In *Conference on Computer Vision and Pattern Recognition*, 964–971. 41, 43, 48
- [112] XUE, Y., LIAO, X., CARIN, L. & KRISHNAPURAM, B. (2007). Multi-Task Learning for Classification with Dirichlet Process Priors. *Journal of Machine Learning Research*, **8**. 60
- [113] ZEISL, B., LEISTNER, C., SAFFARI, A. & BISCHOF, H. (2010). On-Line Semi-Supervised Multiple-Instance Boosting. In *Conference on Computer Vision and Pattern Recognition*, 1879–1888. 43
- [114] ZHANG, K., ZHENG, V.W., WANG, Q., KWOK, J.T., YANG, Q. & MARSIC, I. (2013). Covariate Shift in Hilbert Space: A Solution via Surrogate Kernels. In *International Conference on Machine Learning*. 60
- [115] ZHENG, Z., ZHA, H., ZHANG, T., CHAPELLE, O. & SUN, G. (2007). A General Boosting Method and Its Application to Learning Ranking Functions for Web Search. In *Advances in Neural Information Processing Systems*. 58, 64

CARLOS BECKER

carlosbecker@gmail.com

Avenue de France 39, 1004 Lausanne - Switzerland

<http://cvlabwww.epfl.ch/~cjbecker>

EDUCATION

Research Assistant, PhD Candidate. Research on Computer Vision & Machine Learning

CVLab, EPFL, Lausanne, under the supervision of Pascal Fua
September 2011 / May 2016

Master in Vision and Robotics (ViBOT) Joint Program - Ranked 2nd out of 22

Universitat de Girona / Heriot-Watt University / Université de Bourgogne
2009 / 2011

Electronics Engineering - GPA 9.0 out of 10

Universidad Nacional de Rosario, Argentina.
2002 / 2008

PROFESSIONAL EXPERIENCE

■ **Software & Hardware for Sonar Localization System for SAUC-E competition, Girona team.**

Won 2nd place, being the only team to use sonar localization successfully.
2007 / 2009

■ **Research & Development Position at SATELCO S.A. Rosario, Argentina.**

VoIP and telephone embedded systems development, both embedded software & hardware.
2007 / 2009

■ **Sowing Monitoring System Development. Armstrong, Argentina.**

Embedded Linux, Sensor Bus Topology (CAN)
2008

AWARDS & SCHOLARSHIPS

■ **EPFL IC Teaching Assistant Award** for work in the Pattern Recognition and Machine Learning course, 2014

■ **Erasmus Mundus 2-year Master Scholarship**, 2009 / 2011

■ **Gold medal for top grade in 2001** Colegio de la Inmaculada Concepción, Santa Fe, Argentina.

COURSES TAUGHT & TALKS

■ **Invited speaker on Gaussian Processes and Ensemble Methods**, EPFL, Pattern Recognition and Machine Learning Class. November 2014

■ **Teaching Assistant** for the Computer Vision, Introduction to Image Processing, and Pattern Recognition and Machine Learning courses at EPFL. 2012 / 2014

- **Real Time Operative Systems: Introduction and application to medium-sized embedded systems.**
UNR IEEE Student Branch. June 2009
- **Introduction to the C Programming Language.** UNR, Argentina. May / July 2005.

LANGUAGES

- **English:** Fluent. Certificate in Advanced English, December 2001.
- **French:** Intermediate, Level B1.
- **Spanish:** Fluent, mother tongue.

PUBLICATIONS

- **Domain Adaptation for Microscopy Imaging** - IEEE Transactions on Medical Imaging, 2014
C. Becker, M. Christoudias, P. Fua
- **Non-Linear Domain Adaptation with Boosting** - Neural Information Processing Systems, 2013
C. Becker, M. Christoudias, P. Fua
- **Supervised Feature Learning for Curvilinear Structure Segmentation** - MICCAI, 2013
C. Becker, R. Rigamonti, V. Lepetit, P. Fua
- **Learning Context Cues for Synapse Segmentation** - IEEE Transactions on Medical Imaging, 2013
C. Becker, K. Ali, G. Knott, P. Fua
- **Fast Object Detection with Entropy-Driven Evaluation** - CVPR, 2013
R. Sznitman, C. Becker, F. Fleuret, P. Fua
- **Learning Context Cues for Synapse Segmentation in EM Volumes** - MICCAI, 2012
C. Becker, K. Ali, G. Knott, P. Fua
- **Simultaneous Sonar Beacon Localization & AUV Navigation** - IFAC MCMC, 2012
C. Becker, D. Rivas, P. Ridao

TECHNICAL KNOWLEDGE

- **Machine Learning:** Experience with diverse machine learning methods, particularly for classification and segmentation.
- **Programming:** Advanced skills in C/C++ for both desktop and embedded systems.
- **Electronics & Embedded Systems:** Experience on embedded system design, microcontrollers and FPGAs.

