UC San Diego

UC San Diego Electronic Theses and Dissertations

Title

Improving Biological Object Classification in Plankton Images Using Convolutional Neural Networks, Geometric Features, and Context Metadata

Permalink

https://escholarship.org/uc/item/8f18p61p

Author

Ellen, Jeffrey Scott

Publication Date

2018

Peer reviewed|Thesis/dissertation

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

UNIVERSITY OF CALIFORNIA SAN DIEGO

Improving Biological Object Classification in Plankton Images Using Convolutional Neural Networks, Geometric Features, and Context Metadata

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

Doctor of Philosophy

in

Computer Science

by

Jeffrey Scott Ellen

Committee in charge:

Professor Charles Elkan, Co-Chair
Professor Mark D. Ohman, Co-Chair
Professor Virginia R. de Sa
Professor Lawrence K. Saul
Professor Zhuowen Tu

2018

The dissertation of Jeffrey Scott Ellen is approved, and it is acceptable in quality and form for publication on microfilm and electronically:

_____

_____

_____

_____
Co-Chair

_____
Co-Chair

University of California San Diego

2018

DEDICATION

To my family for all their support,

especially:

my loving wife,

two patient children,

and my parents.

EPIGRAPH

"Education is what remains after one has forgotten what one has learned in school."

Albert Einstein

"Don't Panic"

Douglas Adams; *The Hitchhiker's Guide to the Galaxy*

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

I am privileged to have been able to contribute to such a meaningful application area at such a fantastic institution with such wonderful people. I could not have completed this research without a great deal of assistance. My heartfelt gratitude goes out to:

- SSC Pacific, ONR, and the UCSD Computer Science program for the opportunity via the SMART scholarship program, and providing a long enough leash for me to create a truly unique, enriching, and meaningful PhD experience.
- The Gordon and Betty Moore Foundation for ZooGlider construction and data, and the faith in the SIO team.
- The SIO Instrument Development Group for innovating and troubleshooting the ZooGlider, creating an engaging collection of images for me to use as a platform for my experiments.
- The National Science Foundation for providing XSEDE Computation Time, including extending the duration of their support to match my research timeline.
- The National Science Foundation again, for providing Plankton sample analysis via grants to the Ohman Lab.
- The NVIDIA Corporation for the donation of the Tesla K40 GPU which I used in my own research, but also for the commitment to donate GPUs to academics everywhere. When I had a meaningful interaction with another researcher, they were almost certainly utilizing a donated NVIDIA GPU to achieve their results.
- Contribution from the NSF–supported California Current Ecosystem Long Term Ecological Research site (ccelter.ucsd.edu) including the tangible support and expertise of Tristan Biard, Laura Lilly, Catherine Nickels, Mark Ohman, Linsey Sala, Stephanie

Sommer, Emma Tovar, and Ben Whitmore who conducted numerous hours of organism identifications and validations, without which these experiments would not have been possible. However, CCE-LTER provided far more intangible support to my experience. Being a CCE-LTER member firmly embedded me in a community where I could interact with a dozen other labs and research areas at SIO while introducing me to friends that I would have otherwise not have met. CCE-LTER also provided me access to the field of Oceanography that it would be impossible for me to obtain on my own, allowing me to attend conferences, meetings, and working groups and providing a breadth of knowledge and experiences that I value as much as my own research. Not the least of these experiences was the opportunity to participate in research cruises, which not only solidified my friendships but also expanded my horizons in a way few get to experience.

- UC Ship funds for facilitating the student-lead cruises I participated in.
- Benjamin Whitmore for Zooglider data extraction and analysis. By helping with the dirty work, including deploying and recovering the glider, I was able to keep more focused on the machine learning aspects of the glider images.
- Casey Graff for coding, including tedious code for boring applications, complex code that facilitated more experimentation, and innovative code that lead to new assessments. Regardless of how far in the future this statement is being read, it is probable that I would still be performing experimentation at that time if it wasn't for Casey's assistance.
- Chris Li for assisting with my early experiments and first paper.
- Linsey Sala, SIO Pelagic Invertebrates Collection manager for taxonomic expertise but also for simultaneously providing the unilateral warmth and friendship that one can only

find in a peer, while also providing the guidance and resources that one can only find in a mentor.

- Ohman Lab members, including Amanda Netburn, Jenni Brandon, Ben Whitmore, Laura Lilly, and Stephanie Sommer for accepting me as one of your own and providing so many hours of interesting discussions, laughs, and diversions. A lab full of friends made every part of my experience better, I wish I could have helped out more.
- Ben Whitmore again for providing me just about anything I needed whenever I was in a pinch, including a refreshing can of POG in humid Hawaii and the use of an incredibly ugly Christmas sweater for the SIO holiday party.
- The many individuals that I never met who worked hard at sea to generate the data I relied on, including UCSD Ship Operations, Southwest Fisheries Science Center, and CalCOFI sea-going and technical staff.
- Especially Shonna Dovel for PRPOOS net collections and operations.
- Emma Tovar, Kayla Blincow, Kelsey Gilmore, Todd Langland, Jean-Baptiste Romagnan, Alison Cawood – ZooScan operators who spent thousands of hours painstakingly handling samples, generating images, and assessing their contents. The enormous quantity of images not only facilitated my research, but drew me to the problem in the first place. Without such a large and well curated collection, I would not have pursued classifying plankton images.
- Ben Whitmore a third time for being an incredibly understanding officemate at Scripps, roommate at conferences, and bunkmate at sea.
- Dr. LorRaine Duffy from SSC Pacific for mentorship and starting me down this path. I had not even considered returning to school until she suggested it. I did not pursue

returning to school until she nagged me about it. Most importantly, I would not have finished my degree without her unwaivering support.

- My committee members: Prof. Lawrence K. Saul, Prof. Zhuowen Tu, Prof. Virginia R. de Sa, and especially Prof. Charles Elkan for their support, advice, and patience.
- Professor Mark Ohman, for everything. Obviously for teaching so many different aspects of oceanography. For guidance in the art of research, how to craft a well-designed question. For wisdom of the practice of science, how to precisely determine what the evidence supports. For mentorship regarding the profession of a research scientist, how to navigate the non-science part of being a scientist. For being interested in machine learning. For pushing me to do more, and faster. For forcing me to take extra classes and spend time at sea (maybe I could have achieved the same machine learning results without such a total immersion, but I would not be as good of a scientist or as complete of a person). But also for having patience during my periods of slow progress. For tolerating my repeated 11th hour requests. For proofreading and wordsmithing. But most importantly, for always having more time. I can't figure out how despite having the most obligations, Mark always seems to have more time for his students than they do for him. The longer I was in the Ohman lab, the more I appreciated how many other things Mark could be doing, yet he was investing so much of his time making myself and the other Ohman lab members into the best scientists we could be.

My six year adventure was inextricably intertwined with my family. Like many graduate students, I was fortunate to have the unconditional support of my parents for favors both large and small. Also like many graduate students, I spent many long and sleepless nights, but unlike

most graduate students many of those nights were to help with tending to a sick child or changing a diaper. Superficially, my growth as a scientist and a father had many parallels, such as both myself and my children having to finish our homework at night before waking up early for class the next morning. The curiosity, interest, and glee of my young children bolstered my own joy and enthusiasm for my work, I often felt like a kid again myself.

There were much more meaningful implications of our fusion. Being part of a family unit sometimes hampered my ability to progress quickly but it also forced me to be thoughtful and deliberate in my choices. Sometimes being entangled in family unit felt like an anchor, but more often my family acted as a truss, supporting me and making it impossible for me to collapse even when I was completely overextended. Most importantly, being part of a unit meant that we all got to grow together.

In trying to explain to my kids what it meant to become a (non-medical) doctor, the definition that they best understood was that I was learning how to become the world's expert in something, even if that something was very specific. Officially my coursework, this dissertation, and the research that it represents confers that title upon me exclusively, but my entire family deserves to share the honor for becoming the world's experts at supporting a father trying to learn oceanography while focusing on computer science but also holding down a full time job in service of his country. Congratulations 'Doctors' Lindsay, Mikaela, and Blake for a job well done.

Chapter 2, “A Review of Feature Extraction for Automating Biological Object Classification in Images,” was written solely by the dissertation author. Thanks to Charles Elkan and Mark Ohman for their critical feedback.

Chapter 3, “Improving Object Detection and Segmentation for In Situ Plankton Images,” was written solely by the dissertation author. Parts of the Methods section of this chapter were submitted verbatim as “Supplemental Information” within the publication: Ohman, Mark D.; Davis, Russ E.; Sherman, Jeffrey T.; Grindley, Kyle R.; Whitmore, Benjamin M.; Nickels, Catherine F.; Ellen, Jeffrey S. “Zooglider: an autonomous vehicle for optical and acoustic sensing of zooplankton.” The dissertation author was the sole investigator and author of the reproduced portion of the material. Thanks to Charles Elkan and Mark Ohman for their critical feedback.

Chapter 4, in full, is a reprint of the material as it appears in: Ellen, Jeffrey; Li, Hongyu.; Ohman, Mark D. “Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques.” Proceedings of OCEANS'15 MTS/IEEE, pp. 1-9, IEEE, Washington, D.C., 2015. DOI 10.23919/OCEANS.2015.7404607. The dissertation author was the primary investigator and is the primary author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in: Graff, C. A.; Ellen, Jeffrey. “Correlating Filter Diversity with Convolutional Neural Network Accuracy.” 15th IEEE International Conference on Machine Learning and Applications, pp. 75-80, IEEE, Anaheim, CA, 2016. DOI 10.1109/ICMLA.2016.0021. The dissertation author is an equal contributor in the investigation and authoring of this paper.

Chapter 6 is being prepared for journal submission as: Ellen, Jeffrey S.; Graff, Casey A.; Elkan, Charles; Ohman, Mark D. “Improving plankton image classification using context

features." It is presented as part of this dissertation with the acknowledgement of the study coauthors Casey A. Graff, Charles Elkan, and Mark. D. Ohman. The dissertation author was the primary investigator and is the primary author of this material.

VITA

2001	Bachelor of Science, Computer Science, University of Illinois, Urbana-Champaign
2001-2002	Research Assistant, NCSA/Beckman Institute, UIUC
2002	Master of Science, Computer Science, University of Illinois, Urbana-Champaign
2002-Current	Research Scientist, SPAWAR Systems Center, US Navy/Federal Government
2018	Doctor of Philosophy, University of California San Diego

SELECTED PUBLICATIONS

Kauwell, D.; Levin, J.; Yu, H.; Lee, Y.; Ellen, J.; Bhalla, A. "Does Visualization Improve Our Ability to Find and Learn From Internet Based Information?" Proceedings of the 24th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA. 2001.

Tran, N.X., et al. "Wireless data glove for gesture-based robotic control." In International Conference on Human-Computer Interaction (pp. 271-280). Springer, 2009.

Ellen, J.; Dela Rosa, K. "Text Classification Methodologies Applied to Micro-text in Military Chat", Proceedings of the 8th International Conference on Machine Learning and Applications (ICMLA), pp. 700-714. IEEE., 2009.

Ellen, J. "All about microtext: A working definition and a survey of current microtext research within artificial intelligence and natural language processing", Proceedings of the Third International Conference on Agents and Artificial Intelligence. January 2011. Rome, Italy

Ellen, J. and Parameswaran, S. "Machine Learning for Author Affiliation within Web Forums--Using Statistical Techniques on NLP Features for Online Group Identification." Proceedings of the 10th International Conference on Machine Learning and Applications (ICMLA), pp. 100-105. IEEE., 2011

Macdonald, A.; Ellen, J. "Multi-level resolution features for classification of transportation trajectories." Proceedings of the 14th International Conference on Machine Learning and Applications (ICMLA), pp. 713-718. IEEE., 2015.

Ellen, J., Li, H. and Ohman, M.D. "Quantifying California Current Plankton Samples With Efficient Machine Learning Techniques." In OCEANS'15 MTS/IEEE pp. 1-9. IEEE., 2015.

Graff, C.A. and Ellen, J. "Correlating Filter Diversity with Convolutional Neural Network Accuracy." Proceedings of the 15th International Conference on Machine Learning and Applications (ICMLA), pp. 75-80. IEEE., 2016

PATENTS

Tran, N., Fugate, S., Ellen, J., Duffy, L. and Phan, H., US Secretary of Navy, 2012. RFID system for gesture recognition, information coding, and processing. U.S. Patent 8,279,091.

ABSTRACT OF THE DISSERTATION

Improving Biological Object Classification in Plankton Images
Using Convolutional Neural Networks, Geometric Features, and Context Metadata

by

Jeffrey Scott Ellen

Doctor of Philosophy in Computer Science

University of California, San Diego, 2018

Professor Charles Elkan, Co-Chair
Professor Mark D. Ohman, Co-Chair

The thousands of different species of drifting organisms that comprise the plankton form the base of the food web in the world's largest ecosystem; hence sampling plankton to increase our scientific understanding and assess their status is important for ecological, environmental, and commercial purposes. Rapidly expanding libraries of digital images of plankton necessitate new approaches for object recognition and efficient classification. In this dissertation I inventory geometric features commonly used in biological object classification and benchmark the accuracy of supervised machine learning algorithms that use these features. I then employ

convolutional neural networks (CNNs) and evaluate preprocessing techniques and augmentation strategies to improve classification of zooplankton in scientific images.

I use two types of images: those acquired open-ocean by a novel autonomous Zooglider and images of preserved zooplankton from a laboratory-based *Zooscan*. For both instruments, I compiled 350,000+ original training images of zooplankton and marine snow (i.e., detritus) in 16 or 27 classes. To improve object detection, I implement a flat-fielding background correction algorithm and an original two-pass algorithm for segmenting *Zooglider* images. I investigate techniques for preprocessing images, and find that per-image normalization (global contrast normalization) results in the highest accuracy for plankton images.

Since environmental factors force changes in the plankton assemblage, human experts use knowledge of these conditions to determine expected and observed species. Therefore, I evaluate the effect of inclusion of geotemporal (e.g., sample depth, location, time of day) and hydrographic (e.g., temperature, salinity, chlorophyll-a) context metadata on classification efficacy and find a marked accuracy boost for feature-based classifiers (e.g., random forests, support vector machines, and multilayer perceptrons).

I introduce different approaches for incorporating geometric features and context metadata into CNNs and find that these augmentations (geometric, geotemporal, hydrographic) significantly reduce error rates in CNNs, and using all three yields the most improvement. For CNNs, I evaluate the effect of changes in dropout, number of layers, connectivity, and filter size on classification accuracy. I document asymptotically increasing accuracy with more computationally intensive techniques and complex architectures, such as substantially deeper networks and artificially augmented data sets. The best CNN model achieves 92.3% accuracy with a 27-class dataset.

CHAPTER 1 Introduction to the Dissertation

1.1 The Case for Automating Biological Object Classification

Counting or labeling objects within images is a time consuming part of many scientific processes, including in the biological sciences. From *in situ* observations to lab-based experiments to medical diagnosis, assessing the presence or quantity of objects is critical to achieve the scientific assessment or objective. Applications where assessing a small number of images with 100% certainty is required do not make a good candidate for automation; a scientific process that requires counting or observing hundreds or thousands of objects, with some tolerance for a noise in the measurements is amenable to modern machine learning technologies. Automating these processes was inconceivable until recently, yet there is strong motivation to do so. First, human experts are often required for these specialized image processing tasks, including plankton classification (Robinson et al. 2017). For many repetitive tasks humans are susceptible to fatigue, boredom, or bias that causes errors, which may skew or invalidate the outcome, and plankton classification is no exception (Culverhouse et al. 2003; Culverhouse 2007). Second, automation allows for faster results. In the case of a medical diagnosis, increased speed could save lives, in the case of biological oceanography, increased speed could facilitate better sampling and understanding of oceanic phenomena in progress, particularly unusual events such as the recent warm anomalies in the Northeast Pacific (Bond et al. 2015; Ohman 2018). Third, automation reduces cost, which allows for markedly increased sample sizes and more frequent or comprehensive assessments. In the case of climate sciences and environmental monitoring, the impact could be global in scope (Hays et al. 2005; Field et al. 2006; Richardson et al. 2009). For widely distributed or longitudinal studies, an additional benefit can be consistency; an algorithm can be deployed in multiple locations worldwide simultaneously and for a longer duration than is possible for most human experts. Fourth, as more species are

discovered, data that has already been analyzed will need to be reassessed. There are thousands of species of plankton, but dozens more being identified annually. Some are obscure, such as very large (~70cm) gelatinous species from 2000m+ depth (Pugh and Haddock 2014), or so distinctive they are unable to be assigned to an existing phylum (Just et al. 2014). Also many new species are revealed from existing collections when sufficient samples are analyzed, for example, an entire taxonomic family of crustaceans within the Copepoda subclass required reorganization and further delineated into additional species (Bradford-Grieve et al. 2017), which would also trigger a need for reanalysis. Although not a new species, a physical specimen was captured by a Remotely Operated Vehicle (ROV) in the relatively shallow (100-200m) and well-sampled waters of Monterrey Bay which turned out to be a species that had been described once (Chun 1900) and not reported for 100 years (Sherlock et al. 2016). Upon review of archived ROV with a more precise description, that species had been sampled 12 times on video over the previous decades, just not recognized (Carey 2016; Sherlock et al. 2017).

1.2 Defining the Problem Space

When two or more target classes of interest are to be identified from within a set of pixel-based images, the goal can be formulated as a supervised machine learning classification problem. First, a set of 'gold standard' images is created where the desired labels are provided with authority from a trusted source, usually a manual annotation. Next, image processing techniques derive sets of numerical features as a level of abstraction from the pixel representations. Finally, the numerical features are used as input to the machine learning algorithm to train a model. The goal is to automatically assign a label using the same image processing techniques, and to evaluate the efficacy of the machine learning model.

Two of the most significant distinctions for image processing are whether the objects are identical or merely similar, and whether the objects are deformable or non-deformable. While each type of object has unique challenges, and many image processing techniques are used across all four object types, this introduction will focus on feature extraction techniques that are most clearly relevant for deformable, non-unique objects (Figure 1.1). Most biological objects are in this category.

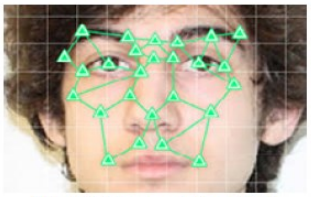
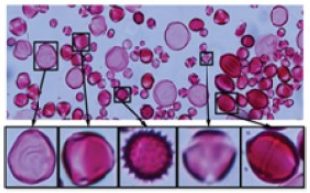
OBJECT CLASSIFICATION CRITERIA		**Rigidity**	
		Non-deformable	Deformable
Uniqueness	Unique	 Landmark (building) identification	 Facial Recognition (ID)
	Non-Unique	 Crystalline or Manufactured Object Classification	 Biological Object Classification

Figure 1.1: Representative object types for image classification problems, arranged on two different axes: uniqueness and rigidity. Biological object classification tends to involve non-unique, deformable objects**.**

Besides the target object's criteria, there are other common image processing challenges resulting from the manner in which the image is captured. Two important considerations are whether the objects will be easy to locate in the image, and whether or not they will be obscured. Segmentation is the definition of a boundary separating a `region of interest,' or *ROI*, from the `background'. Obstructions of the ROI are called occlusions. I am considering the case where segmentation has already been performed, or where the target object is visible without much

clutter. Applicable domains therefore include identifying biological objects within a fluid, such as aquatic organisms, aerial organisms, cellular level identification, and particle recognition (Figure 1.2).

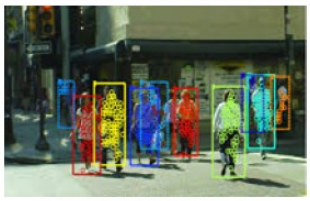
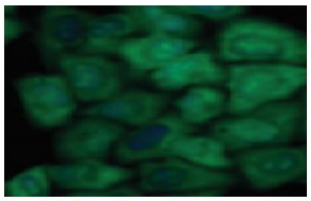
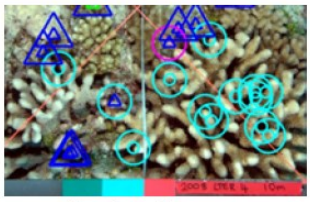
IMAGE ACQUISITION CRITERIA		Segmentation	
		Difficult	Easier
Occlusions	Frequent	 Pedestrian Detection	 Identifying densely packed cells
	Infrequent	 Coral reef census, groundcover classification	 Identifying individual fish, birds, leaves, plankton, cells

Figure 1.2: Representative biological object classification problems, again arranged on two axes: prevalence of occlusions and ease of segmentation. Plankton images tend to be infrequently occluded and easier to segment**.**

One additional consideration for the image is the manner in which it was captured. General purpose image processing such as object recognition, scene understanding, or optical character recognition are generally performed with data from a standard camera, which is intended to be used in a variety of illumination conditions, at various focal lengths (and therefore viewing angles) employing a wide range of magnification. General purpose images are generally recorded extemporaneously, and the information they convey is generally entirely contained within the pixels, or perhaps supplemented by a small amount of metadata such as date and time.

Biological object processing in support of scientific applications is usually performed in a known, controlled environment, such as a microscope, flow cytometer, or wildlife camera. This

provides a known pixel pitch and other high levels of consistency from one image to the next. Also, these images are generally not captured spontaneously, but are often acquired in a planned, controlled manner. Examples of this type of image acquisition includes processing a sequence of known samples (e.g. medical diagnostics), at a fixed interval (e.g. time lapse), or based on a known criterion (e.g. motion-activation). Because of these characteristics, the images often contain metadata that are critical to interpreting the image, such as a location or a sample ID (unique identifier). Sometimes these devices also record other information, such as fluorescence, directly with the file, while at other times, such as with a medical chart, the information is not directly measured physically by the camera. All of these factors lead to a tendency for images of biological objects to have significantly more metadata than other types of images.

This dissertation will primarily focus on images of mesozooplankton drawn from Scripps Institution of Oceanography resources, including the California Current Ecosystem – Long Term Ecological Research (CCE-LTER) project, the Pelagic Invertebrate Collection, the California Cooperative Oceanic Fisheries Investigations (CalCOFI), and a novel *in situ* autonomous *Zooglider* (Ohman et al. 2018). However, the findings and techniques presented extend to many other biological object classification tasks.

1.3 Motivating Questions

1.3.1 What is the prior state-of-the-art for plankton classification in images?

In the context of this dissertation, prior state-of-the-art algorithms are non-Deep Learning algorithms that operate on vectors of feature values. These supervised machine learning algorithms learn an association between vectors of feature values and the provided classification label. When presented with a vector of feature values without a label, the classification is inferred. Example prior state-of-the-art machine learning algorithms include Random Forests

(Ho 1995), Support Vector Machines (Cortes and Vapnik 1995), and Multilayer Perceptrons (Rumelhart et al. 1986). While single-channel plankton images are relatively simple compared to full color natural scenes, previously published results using the above algorithms vary widely in outcomes (Grosjean et al. 2004; Blaschko et al. 2005; Hu and Davis 2005; Sosik and Olson 2007; Gorsky et al. 2010; Luo et al. 2011; Ellen et al. 2015).

Even if Deep Learning algorithms have a higher overall accuracy, there are other factors which might still cause a researcher to select a feature-based approach. One important factor when comparing feature-based algorithms with one another, and with Deep Learning methods, is to establish the number of images that need to be annotated in order to form a sufficiently large training set to achieve asymptotic classification accuracy. A second factor is that as plankton imaging technologies migrate from simply being tethered to a ship *in situ* (e.g. Benfield et al. 2003; Cowen and Guigland 2008; Picheral et al. 2010; Briseño-Avena et al. 2015) to being deployed aboard autonomous vehicles (e.g. Ohman et al. 2018) and some level of on-board classification is desired as a diagnostic, the power consumption and hardware requirements of the classification algorithm may be a larger concern than the overall accuracy.

1.3.2 What is the improvement from using contemporary convolutional neural networks?

Convolutional neural networks for image processing, as an example of contemporary deep learning methods, are repetitive structures that apply a system of hierarchical filters to process their input in a manner inspired by Hubel and Wiesel's investigation of receptive fields within the visual cortex (Hubel 1959; Hubel and Wiesel 1963). Unlike a mammalian brain, however, the criteria that trigger recognition in a CNN can be quickly focused for a specific goal *a priori*. Sets of CNN filters learn to be maximally discriminative; that is given a set of training images, their configuration is optimized so that the most diagnostic features are recognized (not

necessarily the entire contents of the image). The earliest successful commercial usage of CNNs for image classification was handwritten zip codes on envelopes sent via U.S. Mail. The handwritten individual digits consisted of low resolution images resized to 16 pixels square (Fig. 1.3, left), and the usage of CNNs surpassed existing benchmarks by 30% (LeCun et al. 1989; LeCun et al. 1998).

In terms of expressive power, which is the notion of the complexity of the relationship between input images and desired output labels, CNNs are so powerful that they can essentially memorize training examples without sufficient ability to generalize or extrapolate to future input, a condition called overfitting. One of the factors in the dramatic increase of successful CNN applications for image classification since 2010 is called 'dropout', where during training the CNN has half or more of its neurons disabled in order to prevent overfitting (Krizhevsky et al. 2012; Srivastava et al. 2014). Dropout enabled successful training of larger CNNs with more sets of filters without overfitting, and these CNNs were able to classify full color images of everyday objects such as those in the ImageNet Large Scale Visual Recognition Challenge (LSVRC, Deng et al. 2009). The ImageNet images are from contemporary digital cameras, most with a shorter edge longer than 256 pixels. Altogether, ImageNet LSVRC data consists of millions of images, each labeled by one of thousands of hierarchical categories (Fig. 1.3, right). These larger CNNs have become widely adopted since their introduction, so that many trained models are available, in particular the ImageNet challenge winners (Krizhevsky et al. 2012; Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016). Recently, performance by best CNNs has even outperformed the human-level benchmark on the ImageNet challenge data (He et al. 2015). However, an open question is whether or not an existing model will be effective for classifying

plankton images. If an existing model does not provide good results, then additional design is required to optimize the parameters and hyperparameters of a Convolutional Neural Network.

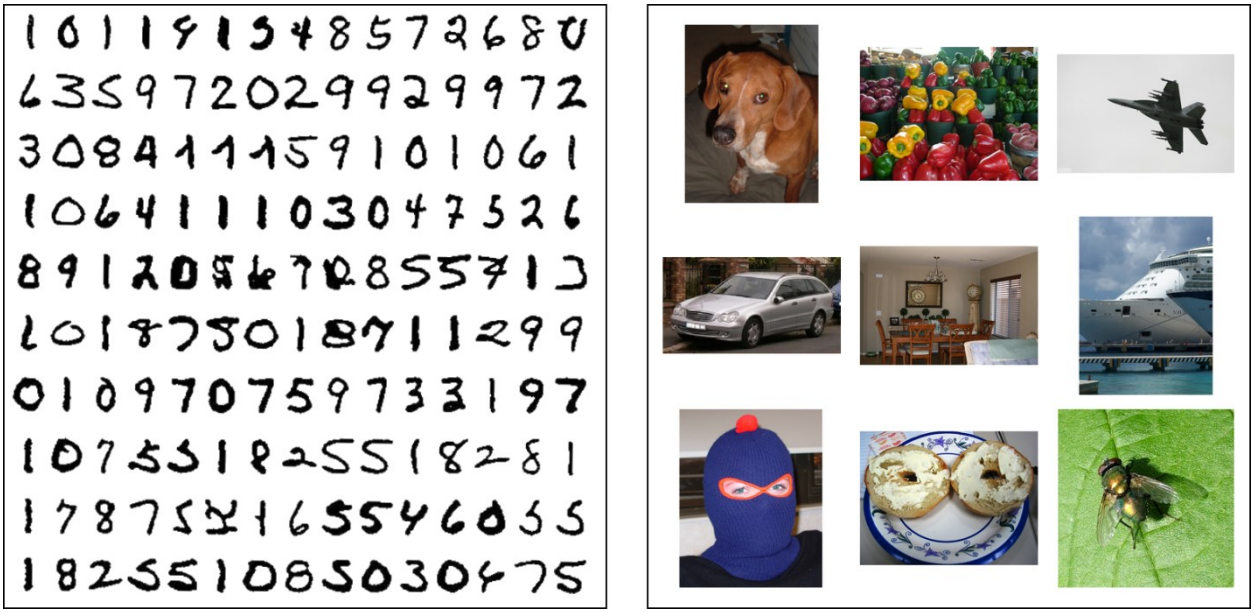


Figure 1.3: Two types of images commonly used as benchmarks for CNN image classification. On the left are low resolution handwritten digits. Image of digits adapted from LeCun et al. (1989). On the right are higher resolution samples from the ImageNet collection. Images of objects adapted from Deng et al. (2009).

1.3.3 Can context metadata be used to improve classification accuracy?

Human experts, when classifying plankton (or other organisms) often use information not present in the images. I refer to such external information as context metadata. For plankton context metadata can include measurements as diverse as the salinity of the sample, the distance from shore where the sample was acquired, or the local sea level anomaly at the time the sample was acquired which serves as an index of the El Niño/La Niña state (UH/CCE-LTER 2017) and the flow of the California Current (Chelton et al. 1982). These data points can also be reinforcing, such as the chlorophyll-a measurement, the season, and the density of particles (both visually and acoustically) which, when combined, can indicate whether or not there is currently a phytoplankton bloom. A central question is this dissertation is whether inclusion of context

metadata in machine learning applications can significantly improve the accuracy of the classifications. This question pertains to both CNNs and feature-based algorithms, although inclusion of context metadata into CNN classifiers is not straightforward because the hierarchical filters of CNNs only operate on pixels. Hence, I explore different means of incorporating context metadata into a CNN classification problem.

1.4 Outline of the Dissertation

Three different types of expertise are required for automating image recognition workflows: domain expertise, image processing, and machine learning. The chapters of this dissertation correspond to the second two elements of this workflow. The chapters are ordered in the sequence in which the tasks would be encountered when developing an end-to-end solution for other application areas, not only plankton identification.

The first required form of expertise is on the application domain for which the labeling is required. In order to train an algorithm, labeled training data must be provided, along with insights into the desired results, challenges, and potential troublesome classes. Domain expertise is outside the scope of this dissertation, except that I express appreciation for the thousands of hours of hard work that are required to acquire the source images through novel engineering and physical labor, to develop the expertise to identify organisms through extended study, and to generate thousands of human-labeled examples of objects (Figure 1.4).

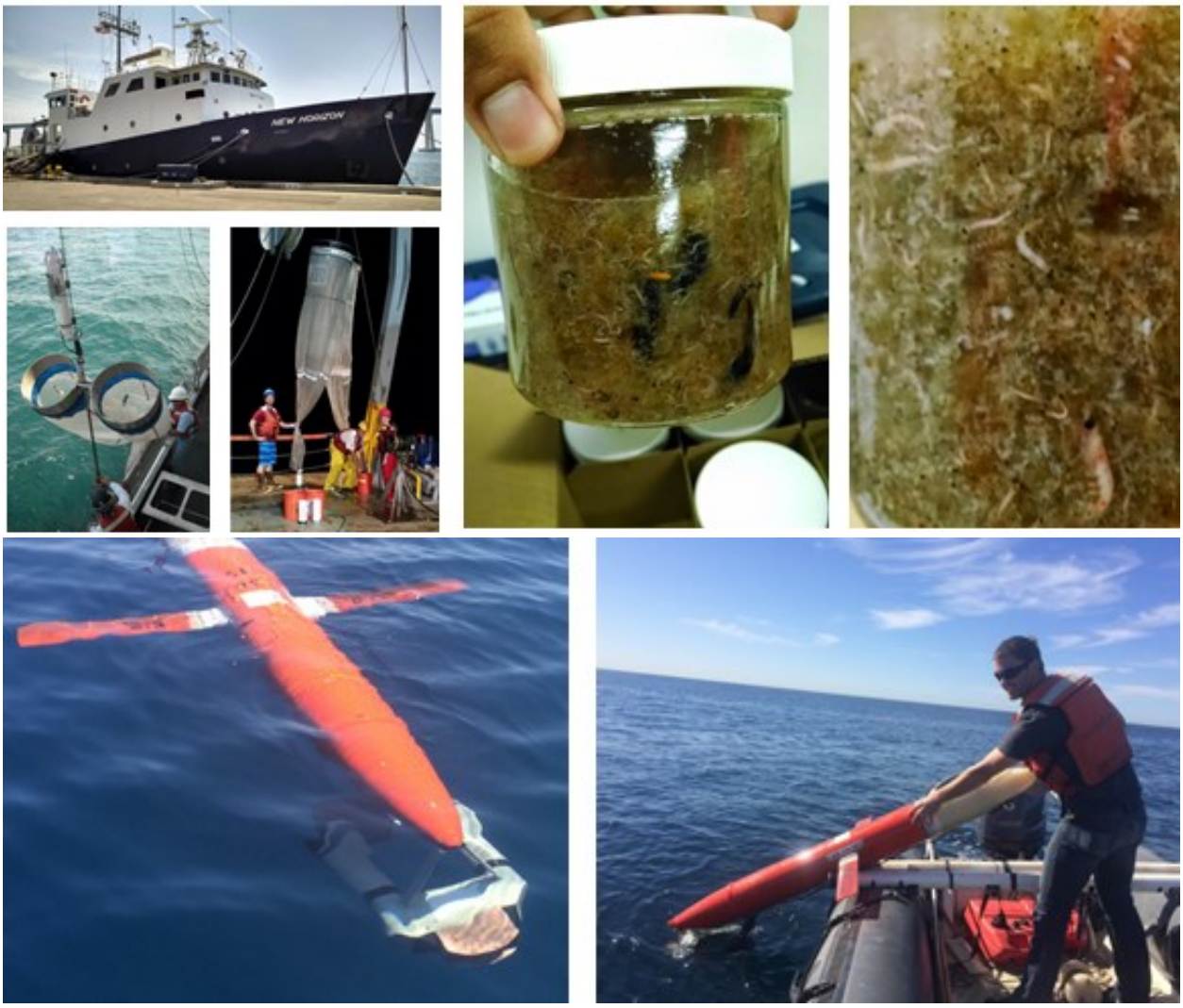


Figure 1.4: Expertise in plankton assessment requires hard work that can include deploying equipment at sea, collecting, and preserving samples. In the case of autonomous vehicles, deployment, operation, and extensive digital data processing are required**.**

The second form of expertise is the image processing required to transform the raw intensity values into a different representation that can be more easily analyzed by a machine learning algorithm. Humans naturally and rapidly develop a vocabulary and understanding of shapes, colors, and textures more easily than any digital system. However, natural language descriptions of classes are not currently part of any tractable machine learning algorithm.

Machine learning algorithms do operate on quantitative features, which are extracted from images using various algorithms.

In Chapter 2, I provide a comprehensive overview of historical and contemporary feature extraction techniques that are particularly applicable to biological object classification in images. This overview includes a literature review and categorizes previous feature extraction techniques into three different categories: statistical analysis methods, topology-based methods, and point/patch correspondence methods.

In Chapter 3, I describe methods that improve the quality of our plankton images, thereby providing better features for machine learning algorithms. My contributions include creating a uniform background with a technique called flat-fielding, and adjusting contrast and removing artifacts so that objects in the image are uniformly illuminated and distinct from the background as much as possible. In order to complete the transition from signals received on a charge-coupled device (CCD) to viable machine learning input, the images need to be segmented to detect any plankton in the images. While it is possible to simply apply machine learning algorithms to every part of the input image, I instead consider algorithms that identify which pixels from a particular image frame belong to a region of interest (ROI), and which are just noise.

The third form of expertise required is on the computer science and machine learning algorithms and statistical techniques required to achieve maximum accuracy from the derived features. Chapter 4 (published as the paper Ellen et al. 2015) establishes a baseline for attainable classification accuracy with zooplankton images using non-deep learning approaches. This publication also quantifies the number of expertly labeled images required and determines that the algorithms that work best in our case are support vector machines and gradient-boosted

random forest. Some ancillary points of investigation include whether or not having the algorithm abstain from assigning a class label for low classification scores provides any meaningful gain, and whether or not size fractionation of images improves accuracy. I also investigate whether creating an ensemble of algorithms with two diverse approaches results in better performance than either individually.

Chapter 5 evaluates normalization strategies and potential diagnostics to consider when using convolutional neural networks (CNNs, Graff and Ellen 2016). We use CNNs on various types of images, including *Zooscan* plankton images (Fig. 1.4), and observe that global contrast normalization provides the highest accuracy for our plankton images, but not for other types of images. We find that rather than starting with an existing CNN, our best results are from training CNN models *de novo*. We identify a correlation between the statistical distribution of the weights of filters in a fully trained network and the overall accuracy of that network. This correlation is potentially useful as a performance diagnostic.

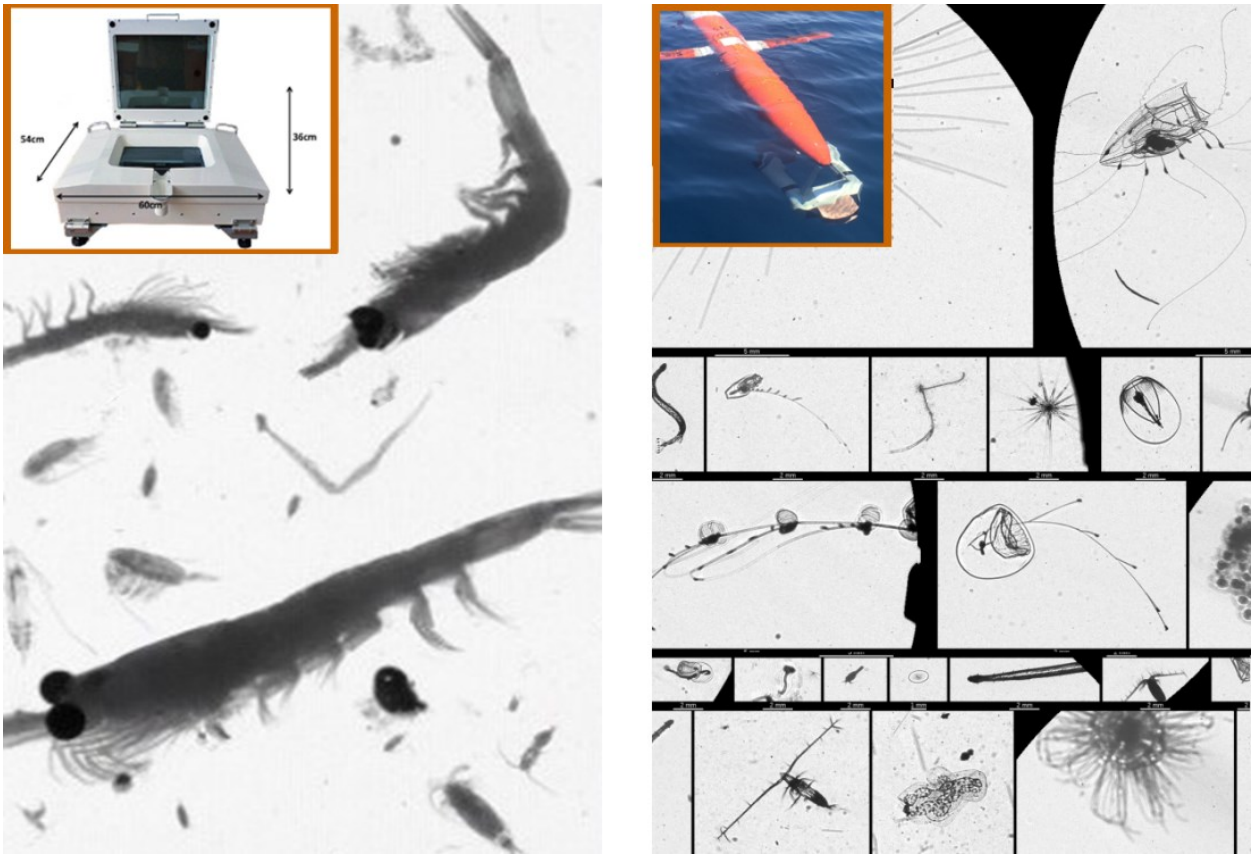


Figure 1.5: *Zooscan* (left, inset) and sample *Zooscan* images (left); *Zooglider* (right, inset) and sample *Zooglider* images (right).

Chapter 6 is a major new contribution of this dissertation. In this chapter, I thoroughly investigate the dominant contemporary approach, convolutional neural networks (CNNs) when applied to our *Zooglider* images (Fig. 1.5). I conduct the investigation using an original data set of 350,000 *in situ* images (roughly 50% marine snow and 50% non-snow sorted into 26 categories). First I establish a performance baseline using feature-based classifiers, and find that CNNs provide significantly higher accuracy than feature-based methods, even with unusually shallow and small CNNs. I then evaluate deeper networks and augmenting the data set, and observe asymptotically increasing performance, and report on which ones are the most effective at classifying *Zooglider* images.

Chapter 6 also finds that including context metadata significantly increases accuracy for all algorithms considered. The context includes the observed conditions when the image was acquired, for example geotemporal (e.g., sample depth, location, time of day) and hydrographic (e.g., temperature, salinity, chlorophyll-a) metadata. All of the feature based classifiers that I assessed benefit from the inclusion of context metadata: random forests, extremely randomized trees, gradient boosted trees, support vector machines, and multilayer perceptrons.

I find that CNNs also benefit from the inclusion of geotemporal and hydrographic context metadata, and derive further benefit from including the geometric features as auxiliary data alongside the context features. CNNs do not inherently accept any non-pixel data, so I consider multiple architectures for including metadata and identify the ones that provide the most accuracy gain. The most simple inclusion of metadata reduces errors by 5%. On my best performing architecture, the benefit metadata provides is a 10% reduction in classification errors, while also a reducing the computation time for training by 15% due to faster convergence of the algorithm.

Chapter 7 provides a brief summary of the dissertation, synthesizing my results from the previous chapter. I also analyze recent developments in the field of machine learning which are applicable to biological object classification.

1.5 References

Benfield, M., C. Schwehm, R. Fredericks, G. Squyres, S. Keenan, and M. Trevorrow. 2003. ZOOVIS: A high-resolution digital still camera system for measurement of fine-scale zooplankton distributions. *In* P. Strutton and L. Seuront [eds.], Scales in Aquatic Ecology: Measurement, Analysis and Simulation. CRC Press.

Blaschko, M. B., Holness, G., Mattar, M. A., Lisin, D., Utgoff, P. E., Hanson, A. R., Schultz, H., Riseman, E. M., Sieracki, M. E., Balch, W. M., and Tupper, B. 2005. Automatic in situ identification of plankton. Proceedings of Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05). 1: 79-86. IEEE.

Bond, N. A., Cronin, M. F., Freeland, H., and Mantua, N. 2015. Causes and impacts of the 2014 warm anomaly in the NE Pacific. Geophysical Research Letters, 42. 9: 3414-3420. doi: 10.1002/2015gl063306

Bradford-Grieve, J. M., Blanco-Bercial, L., & Boxshall, G. A. 2017. Revision of family Megacalanidae (Copepoda: Calanoida). Zootaxa, 4229. 1: 1-183.

Briseño-Avena, C., Roberts, P. L., Franks, P. J., & Jaffe, J. S. (2015). Zoops-O2: A broadband echosounder with coordinated stereo optical imaging for observing plankton in situ. Methods in Oceanography, 12, 36-54.

Carey, T. L. 2016. Taking a closer look leads to rediscovery of a prevalent deep-sea animal. Monterey Bay Aquarium Research Institute press release, November 21, 2016. Accessed October 2018.

Chelton, D. B., Bernal, P. A., & McGowan, J. A. (1982). Large-scale interannual physical and biological interaction in the California Current. Journal of Marine Research, 40. 4: 1095-1125.

Chun, C. 1900. Aus den Tiefen des Weltmeeres. G. Fisher, Jena. 519-521. (Quoted from Sherlock et al. 2016).

Cortes, C., and V. Vapnik. 1995. Support-vector networks. Machine Learning 20, no. 3: 273-297.

Culverhouse, P. F., Williams, R., Reguera, B., Herry, V., & González-Gil, S. 2003. Do experts make mistakes? A comparison of human and machine identification of dinoflagellates. Marine Ecology Progress Series. 247:17-25.

Culverhouse, P. F. 2007. Human and machine factors in algae monitoring performance. Ecological Informatics, 2. 4: 361-366.

Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. Computer Vision and Pattern Recognition, IEEE Conference on. 248-255. IEEE.

Ellen, J., Li, H. and Ohman, M.D., 2015, October. Quantifying California current plankton samples with efficient machine learning techniques. Oceans'15 MTS/IEEE Washington. 1-9. IEEE. doi: 10.23919/oceans.2015.7404607

Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. and Thrun, S. 2017. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542. 7639: 115-118.

Field, D. B., Baumgartner, T. R., Charles, C. D., Ferreira-Bartrina, V., and Ohman, M. D. 2006. Planktonic foraminifera of the California Current reflect 20th-century warming. Science, 311. 5757: 63-66.

Gorsky, G., Ohman, M. D. , Picheral, M., Gasparini, S., Stemmann, L., Romagnan, J.-B., Cawood, A., Pesant, S., Garcıa-Comas, C. , and Prejger, F. "Digital zooplankton image analysis using the *ZooScan* integrated system," Journal of Plankton Research, vol. 32, no. 3, pp. 285–303, 2010.

Graff, C.A. and Ellen, J., 2016. Correlating filter diversity with convolutional neural network accuracy. Machine Learning and Applications, 2016 15th IEEE International Conference on. 75-80. IEEE. doi: 10.1109/icmla.2016.0021

Grosjean, P., Picheral, M., Warembourg, C., and Gorsky, G. 2004. Enumeration, measurement, and identification of net zooplankton samples using the *ZooScan* digital imaging system. ICES Journal of Marine Science, 61. 4: 518-525.

Hays, G. C., Richardson, A. J., & Robinson, C. 2005. Climate change and marine plankton. Trends in Ecology and Evolution, 20. 6: 337-344.

He, K., Zhang, X., Ren, S., and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision. 1026-1034. doi: 10.1109/iccv.2015.123

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

Ho, Tin Kam. Random decision forests. In Document analysis and recognition, 1995., proceedings of the third international conference on. 1: 278-282. IEEE.

Hu, Q., & Davis, C. (2005). Automatic plankton image recognition with co-occurrence matrices and support vector machine. Marine Ecology Progress Series. 295: 21-31.

Just, J., Kristensen, R. M., and Olesen, J. 2014. Dendrogramma, New Genus, with Two New Non-Bilaterian Species from the Marine Bathyal of Southeastern Australia (Animalia, Metazoa incertae sedis) — with Similarities to Some Medusoids from the Precambrian Ediacara. PLOS One. 9: 1-11. doi: 10.1371/journal.pone.0102976

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 25: 1097-1105.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1. 4: 541-551. doi: 10.1162/neco.1989.1.4.541

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86. 11: 2278-2324. doi: 10.1109/5.726791

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. Deep learning. Nature 521, no. 7553 (2015): 436. doi:10.1038/nature14539

Luo, Q., Gao, Y., Luo, J., Chen, C., Liang, J., and Yang, C. 2011. Automatic identification of diatoms with circular shape using texture analysis. Journal of Software, Vol 6. No. 3.

Ohman, M.D. 2018. Introduction to collection of papers on the response of the southern California Current Ecosystem to the Warm Anomaly and El Niño, 2014–16. Deep-Sea Research Part I, 140. 1-3. DOI 10.1016/j.dsr.2018.08.011

Ohman M. D., R. E. Davis, J. T. Sherman, K. R. Grindley, B. M. Whitmore, C. F. Nickels, J. S. Ellen. (in review). Zooglider: an autonomous vehicle for optical and acoustic sensing of zooplankton. Limnology and Oceanography: Methods

Picheral, M., Guidi, L., Stemmann, L., Karl, D. M., Iddaoud, G., & Gorsky, G. (2010). The Underwater Vision Profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. Limnology and Oceanography: Methods, 8(9), 462-473.

Pugh, P. R., and Haddock, S. H. D. 2016. A description of two new species of the genus Erenna (Siphonophora: Physonectae: Erennidae), with notes on recently collected specimens of other Erenna species. Zootaxa 4189. 3: 401-446. doi: 10.11646/zootaxa.4189.3.1

Richardson, A. J., Bakun, A., Hays, G. C., and Gibbons, M. J. 2009. The jellyfish joyride: causes, consequences and management responses to a more gelatinous future. Trends in Ecology and Evolution, 24. 6; 312-322.

Robinson, K. L., Luo, J. Y., Sponaugle, S., Guigand, C., and Cowen, R. K. 2017. A tale of two crowds: Public engagement in plankton classification. Frontiers in Marine Science. 4: 82. doi: 10.3389/fmars.2017.00082

Rumelhart, David E., Hinton G. E., and Williams, R. J. 1986. Learning representations by back-propagating errors. Nature 323. 6088: 533-536.

Sherlock, R. E., Walz, K. R., Schlining, K. L., and Robison, B. H Robison (2016). The first definitive record of the giant larvacean, Bathochordaeus charon, since its original

description in 1900 and a range extension to the northeast Pacific Ocean. Marine Biodiversity Records, DOI:10.1186/s41200-016-0075-9

Sherlock, R. E., Walz, K. R., Schlining, K. L., and Robison, B. H. 2017. Morphology, ecology, and molecular biology of a new species of giant larvacean in the eastern North Pacific: Bathochordaeus mcnutti sp. Marine biology 164 1: 20. Doi: 10.1007/s00227-016-3046-0

Simonyan, K. and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. arXiv technical report. CoRR, abs/1409.1556.

Sosik, H. M. and Olson, R. J. 2007. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Limnology and Oceanography: Methods, 5. 6: 204–216.

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

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. 2015. Going deeper with convolutions. In Proceedings of the IEEE conference on computer vision and pattern recognition. 1-9. doi: 10.1109/CVPR.2015.7298594

University of Hawaii Sea Level Center, California Current Ecosystem LTER. 2017. Monthly average of sea level measurements from San Diego Harbor, the sea level average seasonal cycle, and the long term trend are presented, 1906 - March 2017 (ongoing). Environmental Data Initiative. doi: 10.6073/pasta/b88324bad507a400095981d10ae33563. Dataset accessed 10/06/2018.

CHAPTER 2 A Review of Feature Extraction Techniques for Automating Biological Object Classification in Images

2.1 Introduction

This review presents algorithms useful for automated biological object classification from images, and provides illustrative examples from recent scientific literature. I review techniques for extracting features from images and provide an overview of image processing for readers unfamiliar with common types of feature extraction techniques, such as domain experts in biological sciences and machine learning computer scientists. I present features in three groups organized by the morphological strategy they aim to implement: statistical analysis methods, topology-based methods, and point and patch correspondence methods. Features reviewed span the earliest techniques through the most recent advances in the field, including moment-based approaches from the 60s through SIFT and advances in `deep learning' approaches of the past few years. I provide a sample of recent results from biological object classification publications to illustrate the features discussed.

2.2 Review Organization

First, existing publications are surveyed. The rest of the review focuses on various categories of techniques. Each even-numbered section defines a category of features, and provides a high level description of the features. Each subsequent odd-numbered section provides examples from recent literature that illustrate use of the features. Section 2.4 discusses statistical analysis methods, including moments and textures. Section 2.6 covers topology based methods. These methods generally try to quantify the shape and include shape contexts, Fourier descriptors, and various turning functions in addition to skeleton-based methods. Section 2.8 addresses point and patch correspondence methods, which include SIFT descriptors and deep-learning approaches.

Sections 2.4, 2.6, and 2.8 describe the canonical, most widely used techniques encompassing the origins of image processing through current approaches. For each method cited, the level of detail provided is intentionally cursory, consisting of only the level necessary to introduce each method. My focus is to characterize input to the algorithm, and to review what it is intended to produce, with mathematical and other details intentionally omitted. Full details for each feature type presented in sections 2.4, 2.6, and 2.8 can be found in the cited works, which are generally the original publication for the algorithm, unless a subsequent publication has provided substantial clarification.

Sections 2.5, 2.7, and 2.9 provide example implementations and customizations that illustrate types of strategies that a problem-specific customization would employ. When possible, the examples given in these sections correspond to the type of clutter-free, fine-grained biological object classification task listed as the objective for this review. However, other application domains will sometimes be presented because of their notoriety or potential applicability. The summary provided for each example is intentionally brief; in-depth experimental results and minor optimizations can be found in the original work. Examples were selected on the basis of how well they represent the method and how clearly their results can be interpreted. Secondarily, examples were selected for the soundness of their overall experimentation, and the applicability of their machine learning methodology. Minimal consideration was given to the percentage accuracy, as each domain has its own nuances, and often subsequent improvements can be achieved through completely domain-specific optimizations, which are not useful for this review. On average, a dozen implementations were examined, and three to four reviewed in detail before selecting the provided citations.

While easy segmentation and lack of occlusions are relaxations, there are challenges in the type of tasks focused on in this review. If the classification target is suspended in a fluid (e.g., water or air) the images will be acquired from a variety of perspectives. Therefore, orientation of the target of classification will vary greatly and the scale may also fluctuate. As with object types, use of most features is not intrinsic to any particular image acquisition scenario, but the techniques reviewed are clearly beneficial in the easily segmented, non-occluded situation. The examples provided will originate from all four quadrants of the table in figure 1.2 with a focus on the more easily segmented, less occluded domains. This review will ignore the computational demands of each feature extraction algorithm. Computational complexity will be ignored not only for comparing features to each other, but also for considering optimized versions of the features. This is for two reasons: (1) most of the intended scientific applications will find overall accuracy, not computational resources, to be the limiting factor for implementing an automated process; (2) the details of the complexity improvements are most likely of interest to image processing experts, who are not the intended audience of this review. Many of the features covered in this review have efficient implementations in modern image processing toolboxes.

2.3 Other Feature Extraction Reviews

I aim to summarize the most crucial feature extraction techniques for biological image processing at a level that will allow those without image processing expertise to assess their applicability, but with sufficient brevity to address as many methods as possible. An equivalent review could not be located. A detailed description of the mechanics of each mentioned strategy can be found in an image processing textbook. An Introduction to Object Recognition (Treiber 2010) provides similarly provides brief descriptions, but also includes many algorithms not

applicable for this review. Feature extraction & Image Processing for Computer Vision (Nixon et al. 2012) provides more depth and details, and Pattern Recognition, Fourth Edition (Theodoridis and Koutroumbas 2008) provides perspective on how these algorithms fit within the larger field of 'pattern recognition'. These books were referenced but this review provides additional recent niche techniques not present in these textbooks due to their focus on more general image processing problems and examples of contemporary implementations. This review provides more focus most textbooks by omitting the portions which are not relevant beyond the biological object feature extraction problem. Automated taxon identification in systematics: theory, approaches and applications (MacLeod 2007) provides colorful and thorough discussion of the motivation and history of automating identification, and provides chapter-length summaries of a handful of fully implemented systems, but feature extraction techniques are not enumerated or analyzed separately.

Textbooks aside, there are many reviews, or background sections of longer papers, which overlap some aspects of this review, but do not adequately cover general biological object classification. For example, in (Shortis et al. 2013), the authors evaluate techniques for measuring fish, not classifying them. Ulrich and Steger (2002) provide a feature extraction survey which focuses on performance evaluation, but this and other assessments do not specifically focus on biology. Surveys published within biology (e.g. Pattern recognition software and techniques for biological image analysis (Shamir et al. 2010) tend to evaluate the currently available tools as finished software products and discuss their usability rather than the underlying math and algorithms. (Danuser 2011) and (Cardona and Tomancak 2012) discuss the current state of progress in image processing, and its implications and ramifications on the respective application domains, rather than providing concrete information on techniques as one

would implement them. Each one of the above references is valuable but does not help address two important considerations of the present survey: (1) providing a better understanding of the algorithm; (2) providing awareness across the field of image processing so that new algorithms or advances in theory can be leveraged to augment existing or build custom software.

Loncaric's (1998) Survey of Shape Analysis Techniques provides an overview of a number of many basic shape identification methods. However, that overview is intended for shapes only and is somewhat dated. Loncaric provides a succinct summary of the theories of human perception, in order to bound and quantify the definition of 'shape' and similarity. A more recent review of shape description techniques (Zhang and Lu 2004) provides a summary of various shape-specific descriptions. Zhang and Lu provide information on many adaptations and nuanced variations on the approaches outlined in the Topology Matching Methods section. These surveys focus exclusively on shape, while the present review includes information on color and texture matching.

For a high-level overview of the biological imaging process, refer to a 25 page chapter of Imaging Cellular And Molecular Biological Functions entitled Quantitative Biological Image Analysis (Meijering and van Cappellen 2007). This chapter details the end-to-end process of automating an image processing task. They include an introduction to pixel structure, provide some detail on common arithmetic and preprocessing operations including segmentation and rendering. This is a broader view than is presented in (Shamir et al. 2010), which focuses on pattern recognition specifically. Meijering and van Cappellen's chapter does not significantly overlap with this review, as the chapter includes only 2 pages on features. Domain specific surveys, such as Automated Processing of Zebrafish Imaging Data: A Survey (Mikut et al. 2013), similarly serves as an overview but for a more specific domain, so these publications

focus on metadata, tools, and challenges of a more specific application. While references such as the above are valuable, none provides analytical discussion of the features themselves.

2.4 Statistical Analysis Methods.

The approaches listed in this section describe the object using features where individual pixels have been abstracted. These approaches attempt to quantify the entire object. In general, they have the least number of features. Therefore, they are attempting the highest degree of dimensionality reduction. Equivalent English language descriptions would be along the lines of the image "is mostly black with some gray" or "has lots of stripes."

2.4.1 Moment Based Methods

Using *moments* (e.g. weighted averages) is a common technique for object recognition, and one of the first to be used for shape processing. *Raw* Moments describe the distribution of the intensities of an image. The zeroth order moment is equivalent to the area of the shape. The first order moment is the average of that distribution, or, the center of mass. Higher order moments describe variance, skewness, kurtosis. These can be used directly in highly controlled cases, but are sensitive to any pixels being different.

In 1962, Hu described how moments could be modified to help recognize alphabetic characters by defining variables that would be invariant under translation, scale, and rotation. Hu derived 7 specific moments that were combinations of the simple 2nd and 3rd order moments described above (skew and kurtosis). The example images Hu classified were 16x16 pixels, and 5 levels of grayscale alphabetic characters (Fig. 2.1). So the same image, when viewed at different scales and rotations, will return exactly the same values for all 7 of these parameters. These values are viable features because images which are similar should return values that are nearly the same.

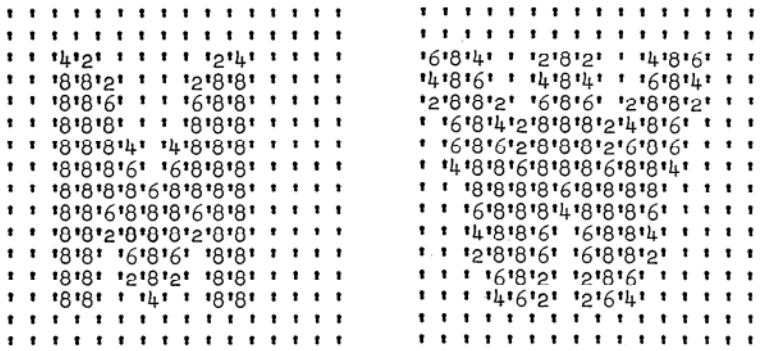


Figure 2.1: Samples from Hu's 1962 paper on *Moment Invariant*, showing images of 16x16 pixels and 5 intensity levels. Image from (Hu 1962)

The reduction of the 256 pixel image to these 7 features enables comparison using various classification algorithms. All other methods described will follow this same process, but with a different number of features. The four main automation steps are: (1) Given a number of `gold standard' images, process each of them to return a vector of floating point numbers (in this case corresponding to Hu's 7 invariant moments). (2) Use machine learning to develop a model for the relationship between these feature vectors and their expert-provided labels. (3) Generate the feature vector for each unlabeled image (in this case Hu's 7 invariant moments) (4) Assign a label for each new feature vector based on the machine learning model. Figure 2.2 shows Hu's 7 invariant moments calculated for 5 shapes, illustrating their potential for use as features. While the values are not exact matches, the orders of magnitude are similar for the corresponding shapes. Even in this simple case, heuristics are not readily apparent, hence the use of a machine learning algorithm to determine a model for label assignment.

Hu Moment	Bear	Bear @ 45	Bear @ 90	Illinois	Mirror Illinois
M_1	7.3967×10^{-04}	7.3688×10^{-04}	7.3829×10^{-04}	7.9192×10^{-04}	7.9192×10^{-04}
M_2	2.2901×10^{-08}	2.2903×10^{-08}	2.2534×10^{-08}	2.0011×10^{-07}	2.0011×10^{-07}
M_3	$.2083\times10^{-10}$	1.1591×10^{-10}	1.1868×10^{-10}	3.3700×10^{-11}	3.3700×10^{-11}
M_4	1.8013×10^{-12}	1.6077×10^{-12}	1.8385×10^{-12}	2.9091×10^{-12}	2.9091×10^{-12}
M_5	2.5436×10^{-23}	2.0917×10^{-23}	2.6149×10^{-23}	2.3478×10^{-23}	2.3478×10^{-23}
M_6	2.2451×10^{-16}	1.9861×10^{-16}	2.2992×10^{-16}	6.8809×10^{-16}	6.8809×10^{-16}
M_7	7.6962×10^{-24}	6.6421×10^{-24}	7.3323×10^{-24}	1.6687×10^{-23}	-1.6687×10^{-23}

Figure 2.2: Illustration of Hu's 7 Moment Invariants (Hu 1962) calculated for 5 instances of 2 different shapes, exhibiting scaling, rotation, and mirroring. The moments are not exactly the same due to small differences related to the discreteness of pixels, for example 50% scaling of a 5-pixel feature must either be 2 or 3 pixels. Mirroring is indicated by the sign of M_7, as indicated in the gray shading.

In 1980, Michael Teague defined a different set of moments based on Zernike polynomials; these have been used extensively and referred to as *Zernike Moments* (Teague 1980). While Hu provided solid analytical foundation for his suggested moments, Zernike moments have dominated in modern usage because they can be defined for arbitrarily high orders, thereby providing additional features for the classification algorithm to use. Calculation of a single Zernike moment is done as follows: First, the image is mapped to a unit disc. Next, the image is effectively projected onto the surface of a Zernike polynomial by applying a weighting function to the individual pixels. The center of mass of this weighted function, given in polar coordinates, is considered the Zernike moment. To achieve rotational invariance, only the magnitude of the vector is considered. Visualization of the Zernike polynomial hierarchy is shown in figure 2.3. Flusser and Suk (1993) further defined a system of moments which was additionally invariant under affine transformations.

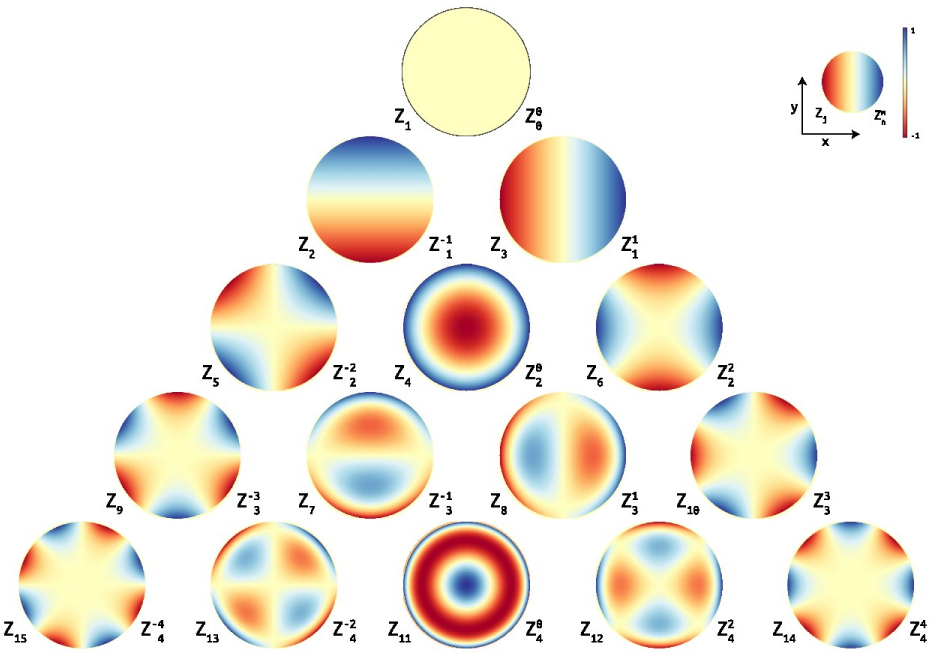


Figure 2.3: Visualizations of the first few orders of Zernike moments. Image from Wikipedia (*http://en.wikipedia.org/wiki/Zernike_polynomials*).

Moments average all of the detail of an image into a series of a few numbers. Many publications report using a few dozen at the most. While moments are a powerful technique, and provide additional advantages for other applications including compression, other approaches have been subsequently developed to capture more detail and nuance of the images.

2.4.2 Histogram Based Methods

As resolution and fidelity increased, various authors created histograms of the pixels in an image and compared them to perform classification. First in black and white, and eventually in color (Swain and Ballard 1991). The intuition is that similar objects would have similar distributions of intensity, and the size of the bins could control for variation between images. Figure 2.4 shows a histogram with a bin width of 1. By design, histograms are insensitive to location and rotation, and somewhat robust to occlusions and viewpoint changes. For example, an image captured at a shallower angle to a flat surface will cause all of the regions of each color

to shrink in proportion. The ordered set of histogram counts is the feature vector. The length of the feature vector is the number of bins in the histogram, which varies from application to application, and is often defined through experimentation.

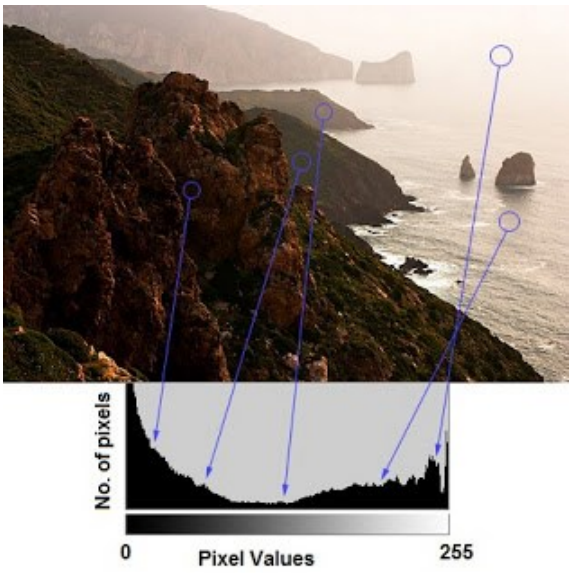


Figure 2.4: Each pixel in the original image contributes its intensity value to the histogram. Image from OpenCV Tutorial (*http://docs.opencv.org*).

2.4.3 Texture Based Methods

Moments and histograms provide information regarding the levels of intensity in a given image, but because of the averaging or binning, all positional information is lost. To address this deficiency, a seminal paper was introduced by Haralick et al. in 1973 to quantify the inherent structure in an image. Originally described as `texture' features (Haralick et al. 1973), they have been extensively used and extended since their original definition. Haralick texture descriptors are calculated in two distinct steps. First, exhaustive statistics about the location and quantity of co-occurring intensity values are tabulated as a *co-occurrence matrix*. The adjacency of every grayscale value is accounted in each of four directions as shown in figure 2.5. These co-occurrence matrices is then used as raw material for higher order mathematical functions including some of the geometric moments mentioned earlier. The aim is to quantify contrast,

orderliness, etc. in the image. In general, references to 'texture' features are statistical measurements of these co-occurrence matrices. In the paper, Haralick provided a set of 14 descriptors including contrast, variance, entropy, etc., whose formulas are concisely provided in Theodoridis and Koutroumbas (2008). Each descriptor is applied to each of the 4 co-occurrence matrices, and the mean of the four values is reported. Therefore the image is summarized as 14 features. Subsequent modifications by others include changing the stride over which the co-occurrence is calculated to a radius larger than 1, and including range and standard deviation in addition to the mean.

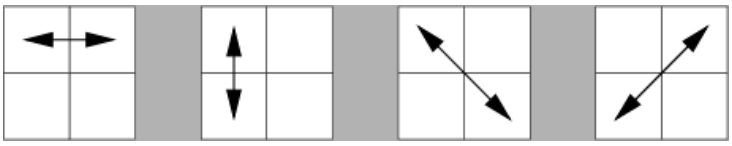


Figure 2.5: Pixel directions used to calculate co-occurrences for Haralick texture features with a radius of one pixel (Haralick et al. 1973).

The original application of Haralick's texture work included classification of overhead imagery for which the segmentation was irrelevant because the `object' was all pixels in the image. In this sense, the term `texture' is applicable not only to the feature being extracted, but to the classification label itself (e.g. 'grassland', 'water', 'urban'). Therefore, texture and its derivatives may be applicable for some types of biological object classification (identifying fields of cells) but without modification, may not be ideal for classification of individual entities. They have been used to classify ROIs but there may be other methods that would perform better.

2.5 Statistical Analysis Methods Specifically for Biological Object Classification

Recent work by Wilder et al. (2012) achieved 95+% classification accuracy on an 8-way classification task identifying reef fish using a simple set of 48 features, including 16 histogram features, 16 discrete cosine transform features, and 16 perimeter measurement features. In their

paper, they asserted that the performance success with simple features was because most of the fish did not resemble each other, and given their lighting configuration color itself (the histogram) was likely enough to be discriminatory.

Rodriguez-Damian et al. (2006) experimented with an assortment of features in a 3-way classification task of different types of pollen. This study is particularly well written, and covers in depth many aspects of the classification process, including image preprocessing and enhancement as well as various machine learning algorithms. It is by far the most comprehensive example cited in this review. They report results on 4 different types of features, and various combinations within each type. First, they use 15 low level geometric features that provided a best case accuracy of 73%. They experiment with a varying number of Fourier Descriptors (see Section 2.6.2), and achieve a best case accuracy of 80% using 128 descriptors. They experiment with four different sets of moments, and achieve 80%. They also used a set of 23 texture features: 6 first order gray level statistics, 7 Haralick textures, 10 other texture features. They experimented with each individually, but achieve 88% using a concatenation of all texture feature types. Most significantly, they claim their maximum accuracy is "much higher than palynologists can distinguish in routine analysis."

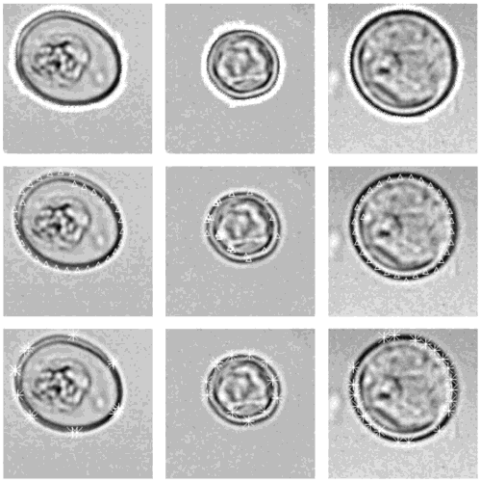


Figure 2.6: Each column contains examples from 1 of the 3 different types of pollen grains used for extensive feature experimentation in Rodriguez-Damian, et al. (2006).

Conceptually, moments require a consistant image aquisition angle, or the object to be relatively non-deformable, such as the case with the pollen grains. This constraint has not prevented moments from being used to classify other biological objects.

Luo, et al. (2004) classify binary images of plankton, as seen in figure 2.7. They used a set of 29 features: The 7 invariant moments described by Hu, the same 7 moments on a slightly modified contour of the image, 7 granulometric features, and 8 low level geometric features. They reported achieving 90% accuracy on a 5-way classification task where all ROIs had a ground truth belonging to one of the 5 types of organisms, and 76% accuracy on a related 6-way classification task where one of the classes consisted of “unidentifiable particles.”

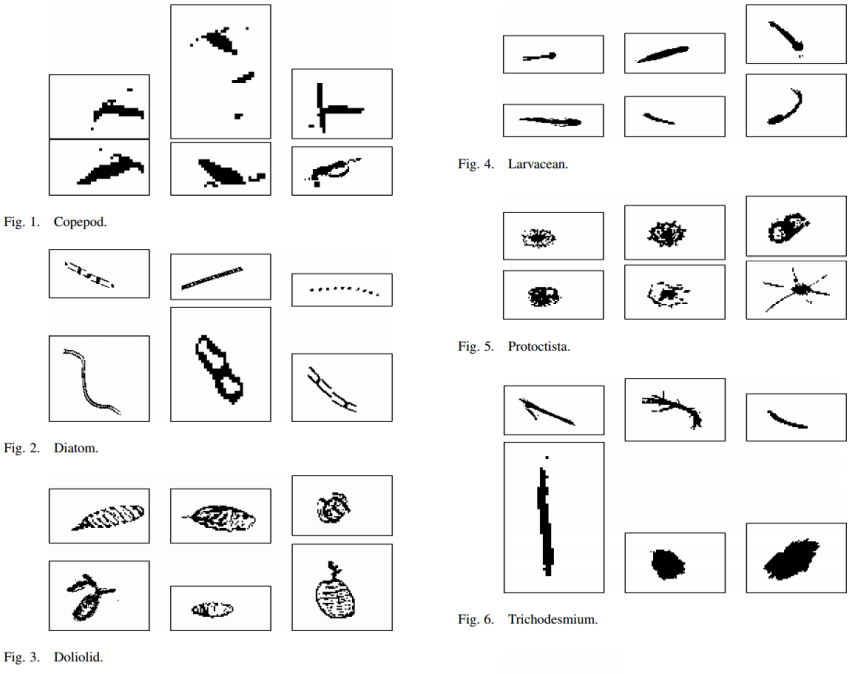


Figure 2.7: Examples of binary plankton images from Luo, et al. (2004).

Sosik and Olson (2007) classify phytoplankton using a combination of 131 features. They achieve 88% classification accuracy on a 22-way classification task of phytoplankton, as shown in figure 2.8. The features used include 6 texture features, 12 invariant moments (based on Flusser and Suk as well as Hu), 39 grayscale co-occurrence statistics, and the rest various geometric features.

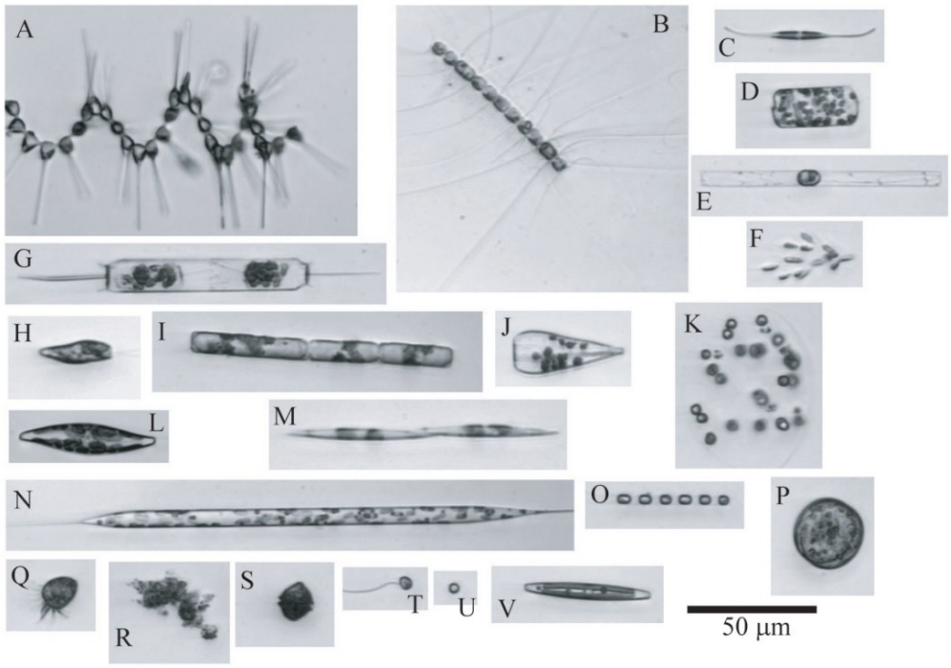


Figure 2.8: Examples of phytoplankton images used during classification by Sosik and Olson (2007).

Boom et al. (2013) use 66 features to achieve 90.1% accuracy on the 15 most prevalent classes of live reef fish in their image set, as shown in figure 2.9. They used a combination of 66 features, including two different 11-bin color histograms, and 12 grayscale co-occurrence features, some affine invariant moments (as in Flusser and Suk (1993)) and a few low level fish-specific features, such as ratio of head area to tail area.

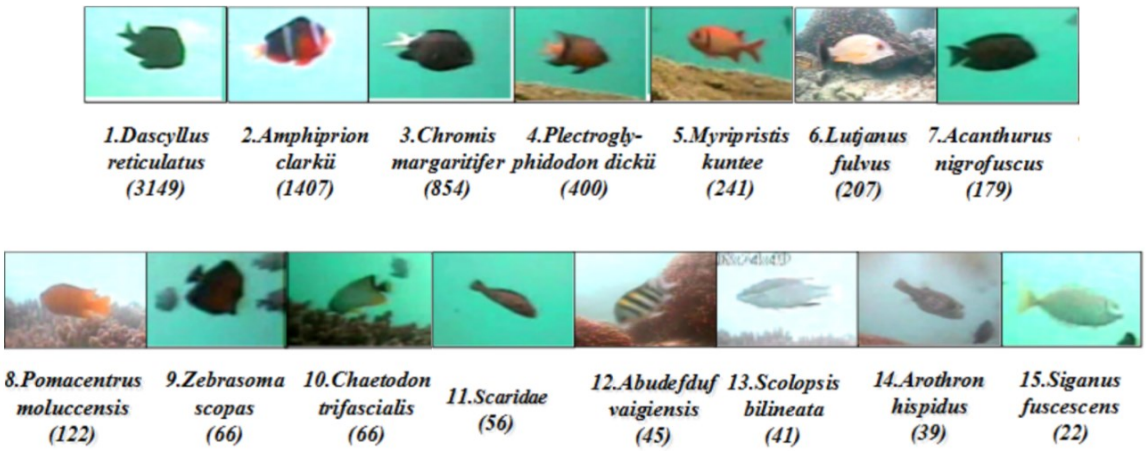


Figure 2.9: The 15 most prevalent classes of live reef fish classified by Boom et al. (2013).

Conrad et al. (2004) use a combination of 323 features and experimented with a number of classifiers and configurations to eventually achieve an average of 82.6% on a 12-way classification task of cellular structures (Fig. 2.10). The features used included 250+ Haralick textures, 50 Zernike moments, and less than 25 granularity features, edge-related features, wavelets, and handful of others. The exact number of features used varied from experiment to experiment, but are dominated by Haralick textures.

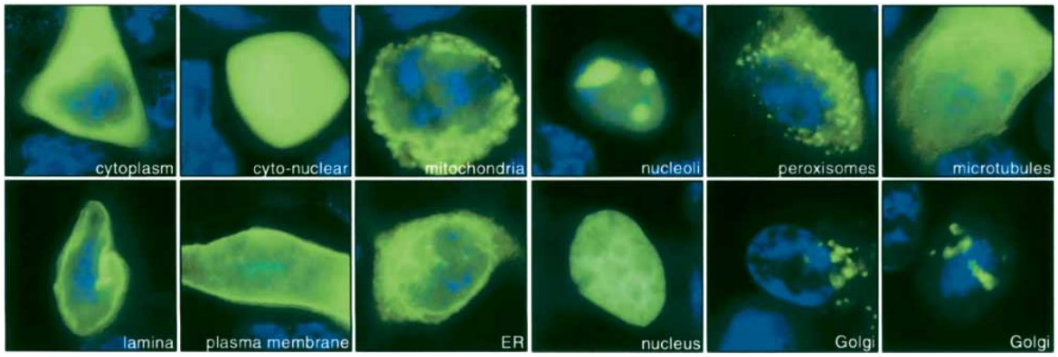


Figure 2.10: Examples of sub-cellular structures classified by Conrad, et al. (2004) primarily using Haralick textures.

Harder et al. (2006) subsequently were able to achieve 96% accuracy using support vector machines on a 4-way cellular classification task using the same base set of features with small modifications (Figs. 2.11, 2.12).

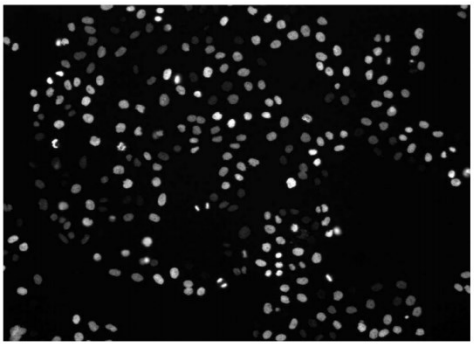


Fig. 1. Multi-cell image from a high-throughput experiment.

Figure 2.11: Sample unsegmented microscopy image of human cells from Harder, et al. (2006)

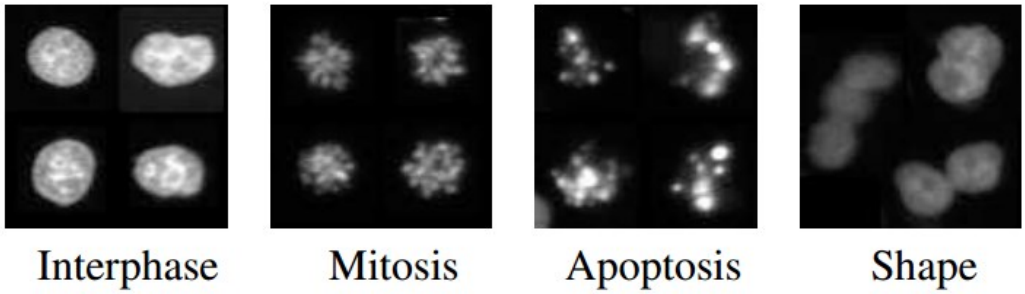


Figure 2.12: Examples of cell nuclei lifecycle stage classified by Harder, et al. (2006) primarily using Haralick textures.

Figure 2.12 illustrates why textures are an appropriate feature for this task: shape and orientation are not the key features necessary to discriminate between the classes.

Chebira et al. (2007) also use textures, but incorporate a multi-resolution approach. Their approach consisted of a combination of a horizontal and vertical filter bank to augment directional edges, and downsampling the image to generate textures at different scales (Fig. 2.13). Their application was a 10-way classification task identifying subcellular structures in an open-sourced set of single-cell images. They improved the previous best performance by 4%, achieving 95.3% accuracy. This is not the first approach to generate features at multiple scales, concisely illustrates one way in which it can be executed effectively.

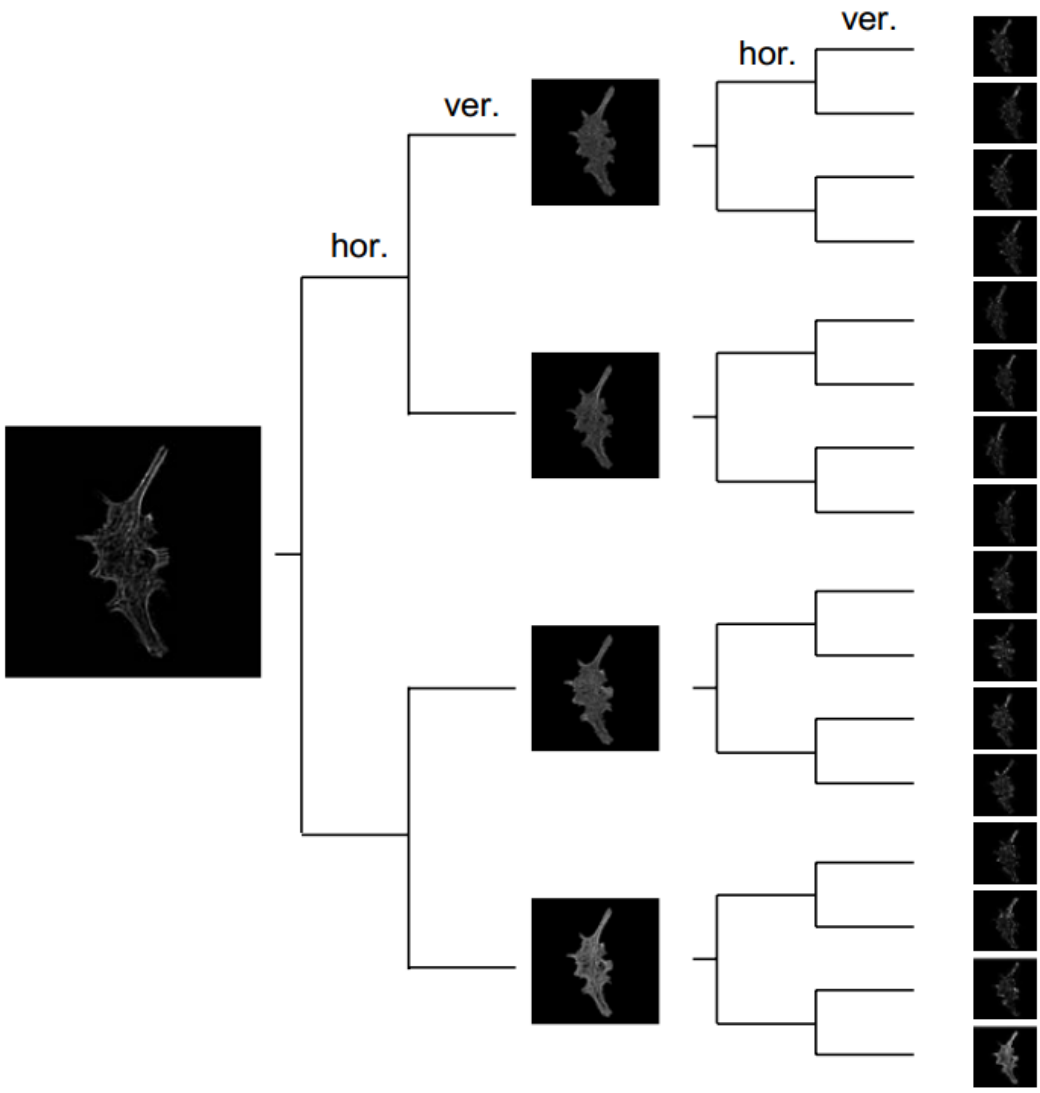


Figure 2.13: Illustration of multi-resolution approach leveraged by Chebira et al. (2007) using primarily Haralick textures at each resolution. Each branch of the tree represents enhancing of edges with filter banks, either high or low pass, and in either the horizontal or vertical direction.

2.6 Topology Based Methods

These methods are characterized by their focus on shape recognition. Once a segmentation algorithm determines which pixels are part of the object to be recognized, and which are not, these algorithms attempt to classify the resulting shape. However, unlike the moments, textures, and histograms mentioned above, which to some extent account for shape,

these algorithms focus on the shape exclusively. An equivalent English language description would say that the ROI is "an elongated oval" or "t-shaped." To restate: these methods generally operate on a *binary* version of the object or even just the object's perimeter, where pixels have a 0/1 value, no intensity information is preserved, or at least is not used as a primary feature. There are features, such as granulometry, that are applicable in cases where the shape is known in advance, or will be nondeformable. When these simplifications are not applicable, the following techniques have been used to classify shapes.

2.6.1 Boundary Matching Methods

Many approaches attempt to quantify the visual experience of `shape', thereby introducing a layer of abstraction. Hundreds of publications use combinations of single low level geometric features, where each is calculated with respect to the segmented region such as area, perimeter, circularity (Young et al. 1974), eccentricity, area excluding holes, etc. Other, similar but slightly more advanced features include convexity, circular variance, and elliptical variance (Peura and Iivarinen 1997). Some measurements, such as circularity imply a regular or uniform shape, such as a circle. For those a best-fit circle or convex hull is first applied to the irregular ROI, and the measurements are taken from that approximation. One benefit to these types of features is that many of them, particularly those expressed as percentages or ratios, are relatively insensitive to orientation or scale. Another potential benefit is that they are fairly general. The machine learning feature vector is a straightforward concatenation of as many of these low level features as desired.

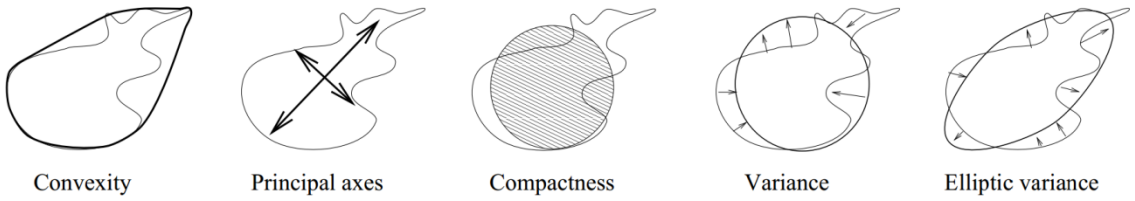


Figure 2.14: Examples of some shape descriptors from (Peura and Iivarinen 1997).

Shape Contexts are a highly cited approach that matches new items with a known prototype (Belongie et al. 2002). The core process forms correspondences between points on one shape to the second to create the actual Shape Context (Fig 2.15). First, *landmark* points are calculated using an edge detector. These are assumed to be interior or exterior contour points to the shape. For resistance to noise and speed of calculation, a randomized, roughly equally spaced subset is selected. Next, relative location from each landmark point to each other landmark point is calculated with respect to a log-polar coordinate axis, with the origin centered on a point. A histogram is generated for each point. This set of histograms is compared to each set of histograms for existing prototypes to create an ordered set of correspondences between each histogram. Then, a thin plate spline model is used to calculate the required deformation, assuming that the two shapes are of the same class (even though they might not be). Ultimately, the classification is made based on a combination of three factors. The first factor is the distance of the thin plate spline calculation (the "Shape Context distance"). The second factor is the "image appearance distance." This is the comparison of intensity values compare at each landmark point, after the images have been warped to the same configuration using the thin plate spline calculations. The third term is the "bending energy" which would have been required to complete the thin plate spline warping. As described, the feature vector is the combination of these three factors with respect to each template.

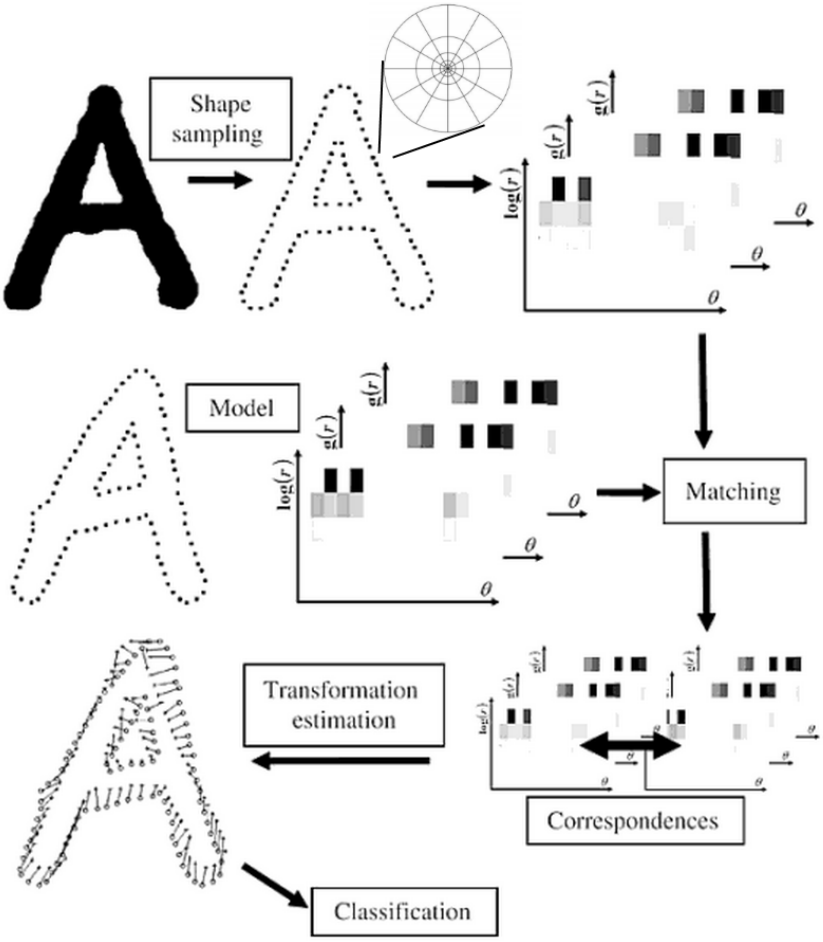


Figure 2.15: The process used to match a new sample to an existing prototype using Shape Contexts (Belongie et al. 2002).

This approach is well suited for things such as digits and trademark symbols, two of the original applications, because these are nondeformable flat objects intended to have a canonical representation that is recognizable from a number of different angles, that is, a distinctive shape.

Another drawback of this approach compared to others is that it requires the selection of particular prototypes to represent the entire class. To quote the paper: “a sparrow is a likely prototype for the category of birds; a less likely choice might be a penguin” (Belongie et al. 2002).

As with any popular and novel approach, Shape Contexts have been extended and modified a number of ways. One extension that is particularly useful for deformable or

articulated objects is *Inner-Distance Shape Contexts* by Ling and Jacobs (2007). Rather than directly creating a log-polar histogram between landmark points based on direct Euclidean distance and global orientation, Ling and Jacobs calculate distance and orientation piecewise. Inner Distance is calculated along the minimal path that stays within the object, which may not be linear, and orientation is determined by the angle of the inner distance segment, as shown in figure 2.16. This particular approach is notable because it specifically accounts for deformations, in particular articulations, which may be common for certain biological objects (e.g. limbs, antennae). The inner distance angle will remain relatively stable in proportion to how inflexible the component arms are with respect to the overall articulation. Specifically, the inner distance angle will remain constant for two completely rigid bodies connected with a joint or hinge (such as legs or wings); but would vary widely for a highly deformable object being stretched (such as a cell undergoing mitosis). These modifications change the calculations but the feature vector is the combination of the same three factors as Shape Contexts.

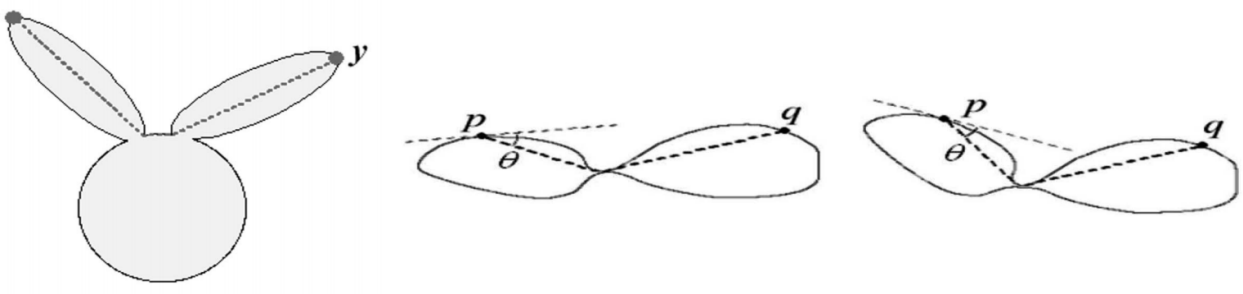


Figure 2.16: The polyline between x and y is defined as the inner distance, and the angle θ between a tangent at p and the inner distance polyline to q is defined as the inner distance angle (Ling and Jacobs 2007). Note that θ is resistant to articulation: it is the same in both figures.

As computing resources are no longer a bottleneck, more complex and less intuitive strategies have been devised. For example, Backes and Bruno (2010) have a two-step strategy. First a series of Complex Network graphs is created that correspond to the Euclidean distance between points on the shape's contour (Fig. 2.17). They then leverage prior research in Multi-

Scale Fractal Dimension to characterize the growth and changes of topological features in the network's growth. Their assertion is that similar shapes will have similar growth curves, even in the presence of noise in the contours, or occlusions which cause the boundary to be non-contiguous (this is of particular concern for methods in the next section, “Path Matching Methods”). Multi-Scale Fractal Dimension provides a feature vector consisting of 7 coefficients of a polynomial that is derived from the evolution of the Complex Network at multiple thresholds. They report a higher success rate than Fourier Descriptors, Zernike Moments, Curvature measurements, or Skeleton Path. In a longer publication, they also report performance on a fish profile classification task (Backes et al. 2009).

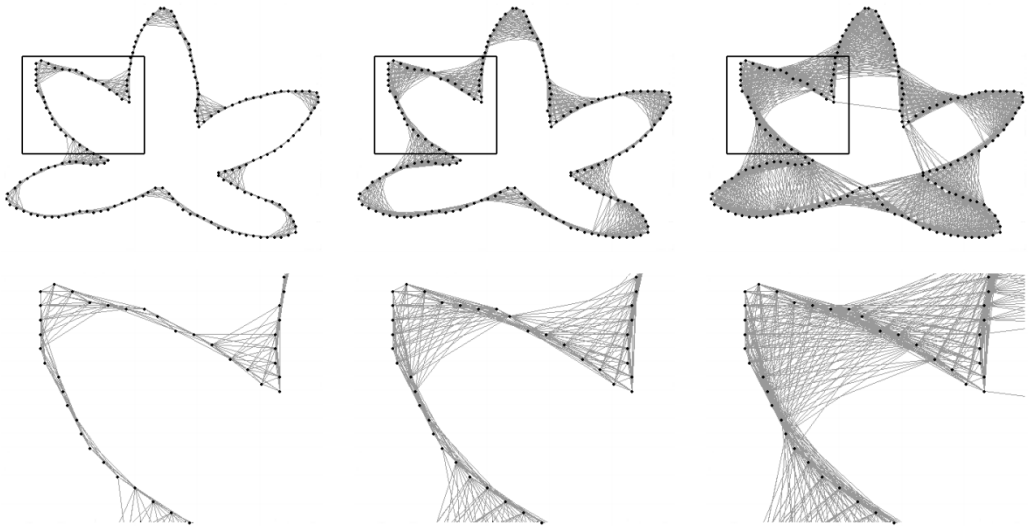


Figure 2.17: Illustration of how the Complex Network is constructed for various threshold distances (Backes and Bruno 2010). The network grows as increasing numbers of neighbors are connected from left to right.

2.6.2 Path Matching Methods

In contrast to the approaches mentioned in the previous section, which consider the whole object simultaneously, approaches in this section describe the shape in a parameterized manner. That is, these algorithms designate a particular starting point on the shape and then proceede around the perimeter.

Fourier Descriptors have been used to describe and classify curves for many decades. Zahn and Roskies (1972) used Fourier Descriptors to describe curves, and Persoon and Fu (1977) used those descriptors specifically to classify silhouettes and shapes, including digits. The contour of the curved shape is first approximated with a function. Then the Fourier coefficients that are derived are used as a descriptor of the shape, and serve as the feature vector for the machine learning algorithm. Limitations of this approach include applicability to irregular or noisy perimeters, and potentially having an undesirably different descriptor as the result of deformations or changes in viewing angle. Fourier Descriptors have also been extensively used in biological tasks, e.g. Lestrel's (2008) book entitled *Fourier Descriptors and Their Applications in Biology*.

Curvature Scale Space is another parameterized perimeter descriptor. It was specifically defined to be invariant to rotation, scale, and translation (Mokhtarian and Mackworth 1986). It is defined by repeatedly smoothing the curve using a Gaussian kernel. σ is the radius of the smoothing, and inflection points on each curve are recorded, and shown as large dots in the left side of figure 2.18. The actual Curvature Scale Space graph is constructed by traveling along each smoothed perimeter, starting from a fixed point. Inflection points are recorded in the scale space image, as shown on the right hand side of figure 2.18. The length of the curves are normalized (to 1) for each smoothed curve. Note that if the level of detail is significantly different in two original images, the lower level curves will be different, but the taller peaks should still correspond closely. The feature vectors consist of the distances from the peaks of one CSS graph to another.

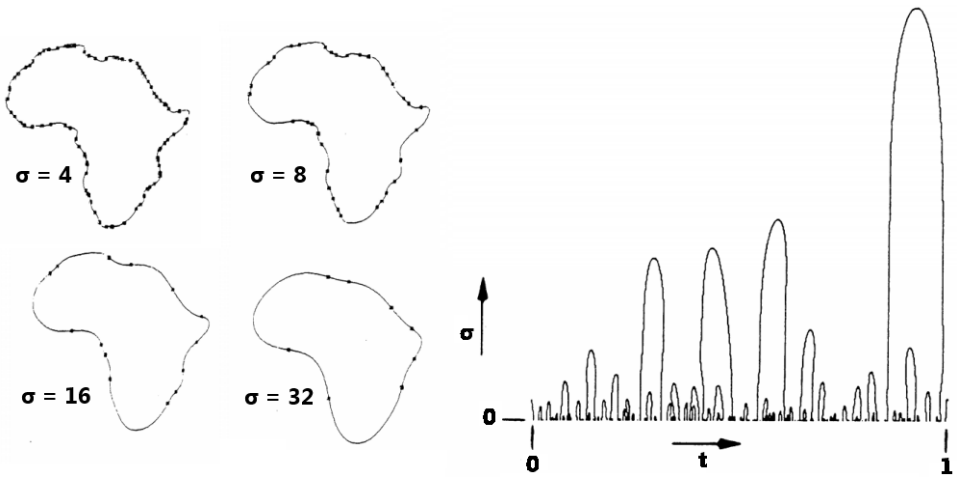


Figure 2.18: Illustration of various levels of Gaussian smoothing of the shape of Africa on the left, and the resulting Curvature Scale Space on the right. Figure adopted from (Mokhtarian and Mackworth 1986).

A more recent path matching method is the Shape Similarity Measure created by Latecki and Lakamper (2000). First, they use Discrete Curve Evolution to account for noise in the shape by smoothing it (Fig. 2.19). They assert that through evolving the curve sufficiently, it is easier to form correspondences. Then using the smoothed shape they generate an inward/outward turning function (tangent function) (Fig. 2.20). The turning function is generated by proceeding clockwise around the perimeter of the shape. X is within the range [0,1] representing the percentage of the perimeter traversed, and the corresponding y value is within the range [0,2 π) representing the angle of a tangent at that point. Shapes with increasing amounts of smoothing will have less complex (and therefore easier to compare) turning functions. As with Curvature Scale Space, the feature vector is the distances from one curve to the other.

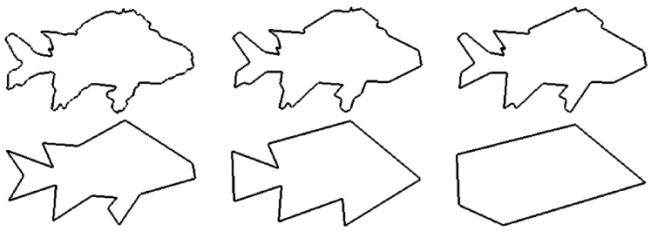


Figure 2.19: An example of a noisy fish profile polygon being smoothed five times with Discrete Curve Evolution (Latecki and Lakamper 2000)

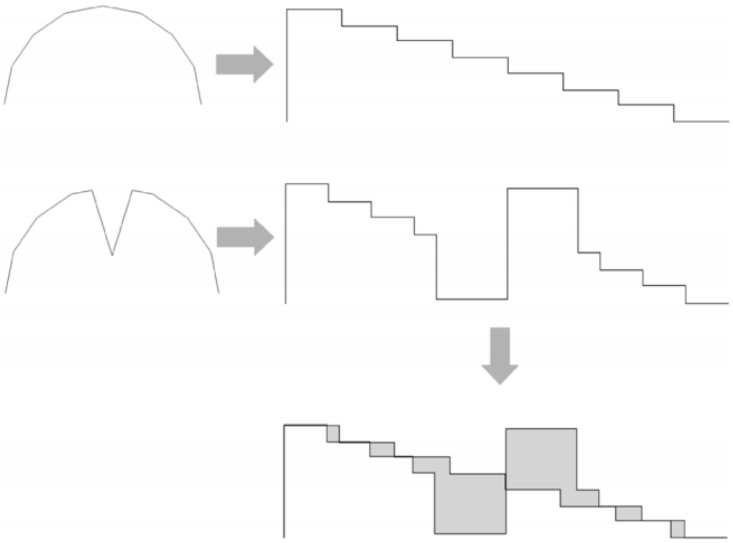


Figure 2.20: An example of the similarity and difference between two tangent curves (turning functions) used to calculate the Shape Similarity Measure (Latecki and Lakamper 2000).

Latecki and Lakamper's approach, like many others covered in this review, makes certain assumptions that are important to consider for the particular biological object classification task. For example, their Shape Similarity Measure is heavily dependent upon the amount of smoothing applied. For a task with very different morphologies, such as classifying the widely different shapes of Sosik and Olson's phytoplankton task in figure 2.8, more smoothing would likely help. For a narrowly defined task, such as classifying species of reef fish (Fig. 2.9), too much smoothing may remove subtle detail. Also, this approach was designed to adhere to generally accepted vision theory principles (Basri et al. 1998). Principles such as `bending at

boundaries/joints should be considered more similar/likely than bending in the middle of a contiguous region'; `bending likelihood should be proportional to thickness', etc. Another important consideration is that as illustrated in figure 2.21, this method is particularly sensitive to occlusion.

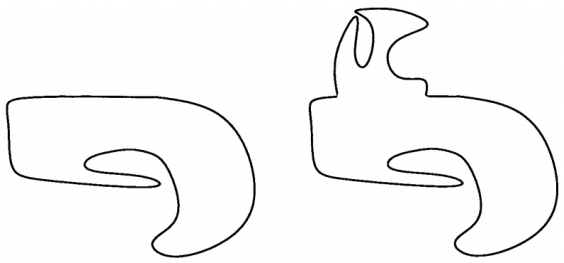


Figure 2.21: An example of two objects with very different perimeters (and therefore very different turning functions) that could potentially be considered very similar if the shape on the left is considered to be an occluded version of the one on the right. Figure from (Basri et al. 1998).

A similar approach was more recently put forth by the same research group, this time as a height function instead of a turning function (Wang et al. 2012). As shown in figure 2.22, this approach also generates an x/y functional relationship by traversing the perimeter with x representing position along the perimeter. However in this instance y represents the height above or below a line tangent to the perimeter at the point of origin. The authors claim that this method is less sensitive to localized noise in the perimeter because of the nature of the noise itself tending to have more of an effect on the tangential angle than the height, as shown in figure 2.23. It also is slightly faster to compute than Shape Similarity Measure from Latecki and Lakamper (2000), because instead of smoothing first to compute tangents, calculating the heights intermittently achieves the same effect as smoothing. The feature vector is again a comparison between these generated graphs.

Since these methods are similar, they can be computed with much of the same code, and the authors show how using both metrics in combination achieves better results than using either of the two individually.

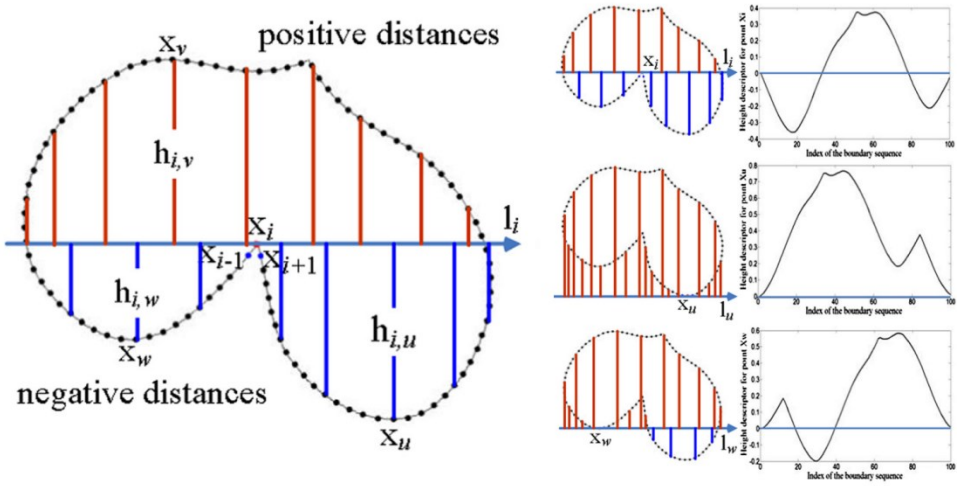


Figure 2.22: An illustration of how the height function is constant for the same shape, but with slightly different amplitude at different points on the perimeter (Wang et al. 2012). Each perimeter point shown, x_i, x_u, and x_w have a horizontal tangent line which is used to calculate the heights, with red heights above and blue heights below the tangent line.

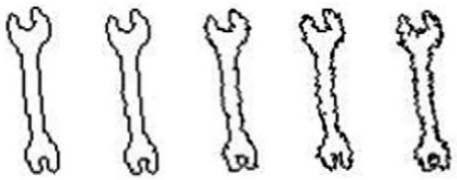


Figure 2.23: Illustration of an increasingly noisy perimeter, for which tangent would be noisier (across the allowable range of *y*-value) than the height function (Wang et al. 2012).

2.6.3 Skeleton Matching Methods

Skeleton matching methods assume that there is some rigid or deterministic structure to the coarse, overall shape in question and that deformable details are superficial. While this analogy is clearly biological and could work well to classify things such as vertebrates, shrouded

objects, and other similar circumstances, it is important to observe that classification by skeletal shape ignores finer level detail of shape. Skeleton matching methods also assume an elongated or non-convex shape. So these approaches would not work well for the pollen grains shown figure 2.6 or the cellular structures shown in figure 2.10, but might work well for the phytoplankton shown in figure 2.8 even though phytoplankton do not have a physical skeleton. There are two cases where these assumptions would not only be acceptable, but perhaps preferred. The first case is when finer grained details are ornamentation and not highly relevant to the overall shape classification. The second case is when fine grained details will be subject to noise in the image capture process, and should not be utilized because they will not be dependable from one entity to the next.

One of the earlier approaches in this area is from 1977 when Persoon and Fu attempted to define the skeleton of some objects using harmonics of their Fourier Descriptors (Persoon and Fu 1977). However, the most widely used definition of an image's `skeleton' is the centers of all the bi-tangent circles to the object. This was termed the *medial axis* (Blum et al. 1967). As a physical metaphor for the definition of the medial axis, he proposed that if the image were a field of grass, and the shape were the location where the grass was set afire, the medial axis would be where the fronts of the fires would meet and extinguish each other. An equivalent, but conceptually different description is the set of all circles which are tangent to the shape in two or more locations, and whose centers are located within the perimeter of the ROI. Both of these cases are illustrated in figure 2.24.

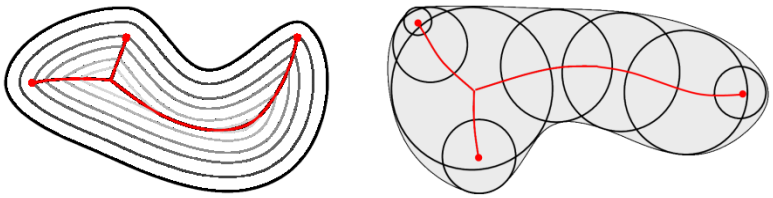


Figure 2.24: Two visualizations of the medial axis used for Skeleton matching. The grassfire/wavefront metaphor is depicted on the left, and the tangential circles are displayed on the right. In both figures, the medial axis is the bold red line in the middle of the shape with round endpoints. Image adopted from (*https://liris.cnrs.fr/david.coeurjolly/*).

Figure 2.24 also illustrates that for most classification tasks, the skeleton used does not precisely match the definition because it does not intersect the perimeter of the region. Most classification tasks define a threshold additionally constraining the skeleton to be a certain distance from the perimeter itself. This parameter controls noise, and it is important to note that it would be highly task-specific.

Unlike other features previously mentioned, the skeleton cannot be used directly as a feature by itself. Quantifying and comparing the resultant skeletons is required in order to achieve automated classification. The skeleton could then be considered a binary ROI and analyzed with techniques above. Many different comparison approaches are possible, and some unique to this construct.

One example of a specific research trajectory based on Blum's skeleton concept was developed by a series of PhD students under Benjamin B. Kimia at Brown University. These introduced and gradually refined the *Shock Graph* matching concept in order to achieve shape recognition. Shocks are a classification of the points within the skeleton defined by their topological function in the overall structure. Various investigations culminated in an approach for creating a graph from these shocks (Siddiqi et al. 1999). The concept is to create a graph based on shock classification, as shown in figure 2.25.

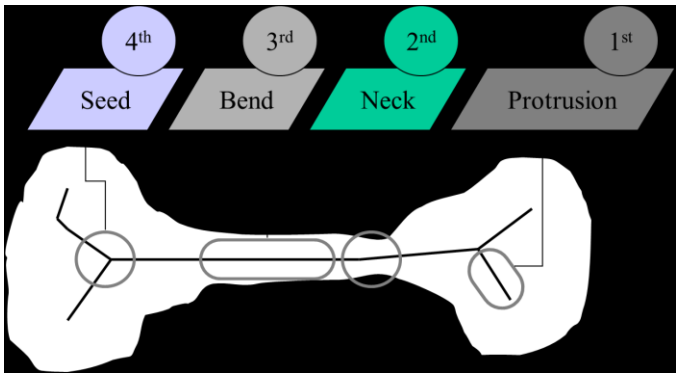


Figure 2.25: The four orders (classes) of Shock Points explained graphically. The mathematical definitions are provided in (Siddiqi et al. 1999). Image from (Johannessen 2011).

Shocks which are adjacent and of the same class are merged to form a single node. The rationale is that small amounts of noise or distortions in shape are absorbed in this step. Since the graph derived was shown to be unique for a particular 2-D shape, it can be used as a substitute for considering all pixels. The initial paper considered pose estimation and perspective in addition to classification.

Further research from the group partitioned Shock Graphs into equivalence classes, which they termed *Shape Cells* (Sebastian et al. 2001). They then considered edit distances between cells, and use that distance to classify new items the same as their nearest neighbors. Specifically, their approach was to simplify each exemplar through deformation to reach a common shape, and consider the length of those paths. A partial illustration is shown in figure 2.26. This was shown to work very well for classifying shapes from a number of different benchmark databases.

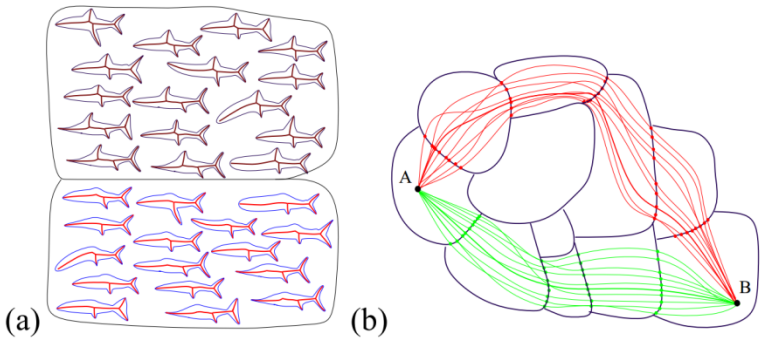


Figure 2.26: Part (a)illustrates a number of different perimeters divided into two Shape Cells. Note that each perimeter has its skeleton depicted in red, with skeletons in the top cell having one more segment than skeletons in the bottoms cell. Each shape in the cell is considered roughly equivalent because each of their skeletons would result in an identical Shock Graph, due to the requirement to merge adjacent shocks. The cells are considered adjacent because only one graph operation is required to turn the Shock Graph from the top into the bottom. Part (b) shows an example of two possible deformation paths from one Shape Cell to another (Sebastian et al. 2001). The original authors draw multiple lines to indicate that there are conceptually many different transformations that can be made that result in equivalent path distance.

The minimum path distance between Shape Cells is then directly the measure of similarity between two different images to be considered. Structured as a supervised classification task, the distance computation is performed from an unlabeled image's Shape Cell to every example class. Therefore, the classifier for this approach must be nearest-neighbor within this non-euclidean space of possible transforms (See Fig 2.26 Part (b) for an example depiction). Further extensions of this method are cited in the next section regarding biological object classification.

2.7 Topology Based Methods Specifically for Biological Object Classification

Taylor, et al. (2008) use a combination of statistical and topological features to classify scallops *in situ*. They use a combination of Haralick textures, color histograms, low level geometric descriptors, and Fourier Descriptors. They achieve an overall accuracy of 82% on their 5-way classification task, including single class accuracies of 77%-87%. In a subsquent publication, they introduce HoG features to assist with differentiating textures for their classes,

(along with other modifications to improve segmentation) and report a significant improvement. Each of their types of features only achieved at most 75% accuracy individually, but in combination they were able to achieve 98.7% accuracy on classification of already segmented objects.

ImageJ is a java based software package published as open source by the US National Institutes of Health (Abramoff et al. 2004). This software can generate 60+ geometric features for segmented ROIs, such as area, perimeter, eccentricity. It also computes grayscale information for histograms, etc. Since it is open sourced, it has been repackaged within some domain specific software packages.

One software package based on ImageJ is Fiji (Schindelin et al. 2012), which has earned 975 citations since its introduction. It does many things outside the scope of this review (for example, segmentation), but also generates additional feature types, including SIFT features and skeletons. Another popular software suite based on ImageJ is CellProfiler (Carpenter et al. 2006), which has also been cited 975 times. In addition to the geometric features, it provides Zernike moments, Haralick textures, and many others. Not all of the citations are for supervised classification tasks. Some applications are simply quantifying datasets using CellProfiler when classification of ROIs is not desired. However, CellProfiler has been successfully used for a number of cellular classification tasks, including 91% accuracy on phenotype classification (Horvath et al. 2011), as shown in figure 2.27.

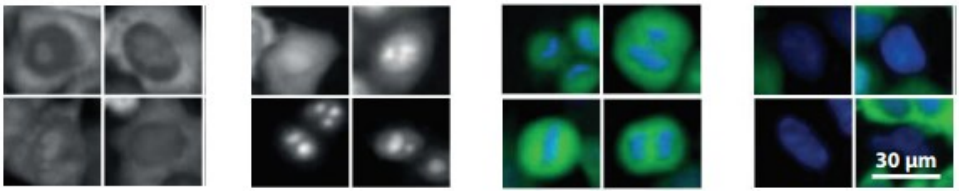


Figure 2.27: 4 different cellular phenotypes classified by the CellProfiler software in conjunction with customized machine learning software. Image modified from (Horvath et al. 2011).

ZooScan is another specialized version which uses the low level geometric features and grayscale features ImageJ provides (Grosjean et al. 2004). It has been used to count the abundance of both zooplankton (Gorsky et al. 2010) as well as fish eggs (Lelievre et al. 2012). Comparison of software suites is not the focus of this review. However the popularity of these tools helps give a metric as to whether or not researchers are finding the features that these tools generate to be useful.

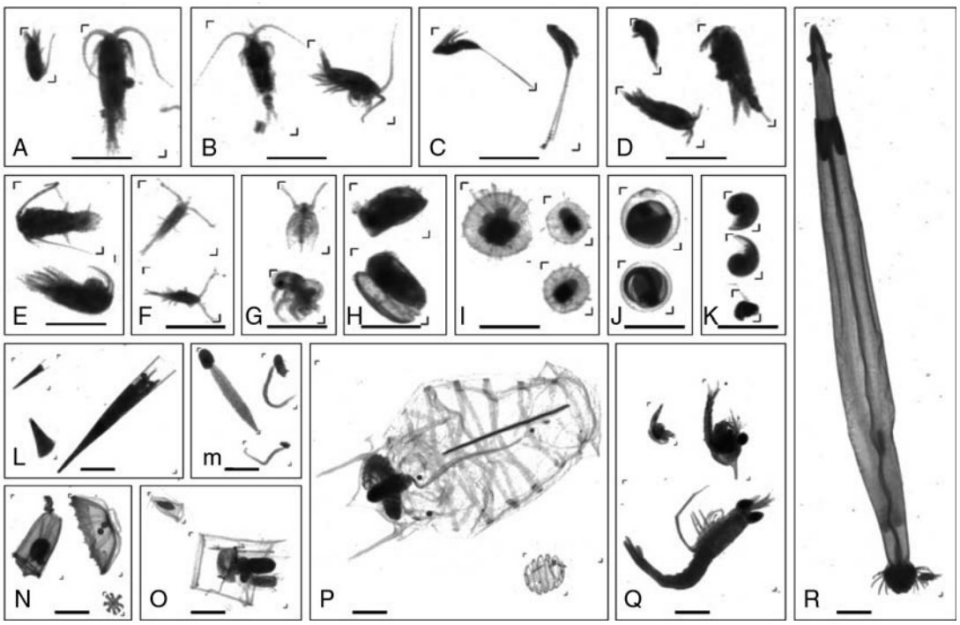


Figure 2.28: Varieties of zooplankton classified by (Gorsky et al. 2010).

Hu and Davis (2005) built a series of algorithms to classify diverse plankton including diatoms. This is a relatively broad classification task; the classes that they considered had dramatically different morphologies, as shown in figure 2.29. They experimented with a variety

of classifiers, and a few different feature sets. In 2004, they used a combination of the seminal 7 moment invariants described by Ming-Kuei Hu (Hu 1962), 6 low-level geometric features, 64 Fourier Descriptors, and 160 granulometric features. Using these features, they achieved 60-70% accuracy on 7-way classification, and 79-82% accuracy on binary classification tasks (Davis et al. 2004). In 2005, they derived 64 Haralick texture based features (Hu and Davis 2005). Specifically, they calculated a total of 8 different co-occurrence summarization matrices, which calculated the mean and range of gray-scale values at 4 different angles and distances. They then calculated a set of 8 different statistics suggested by Haralick on each of the 8 matrices, resulting in the 64 texture based features. They reported 10% improvement on the 7-way classification problem. In 2006, they combined these approaches into an ensemble method, and reported 50% fewer errors on low abundance (and therefore more difficult) classes (Hu and Davis 2006).

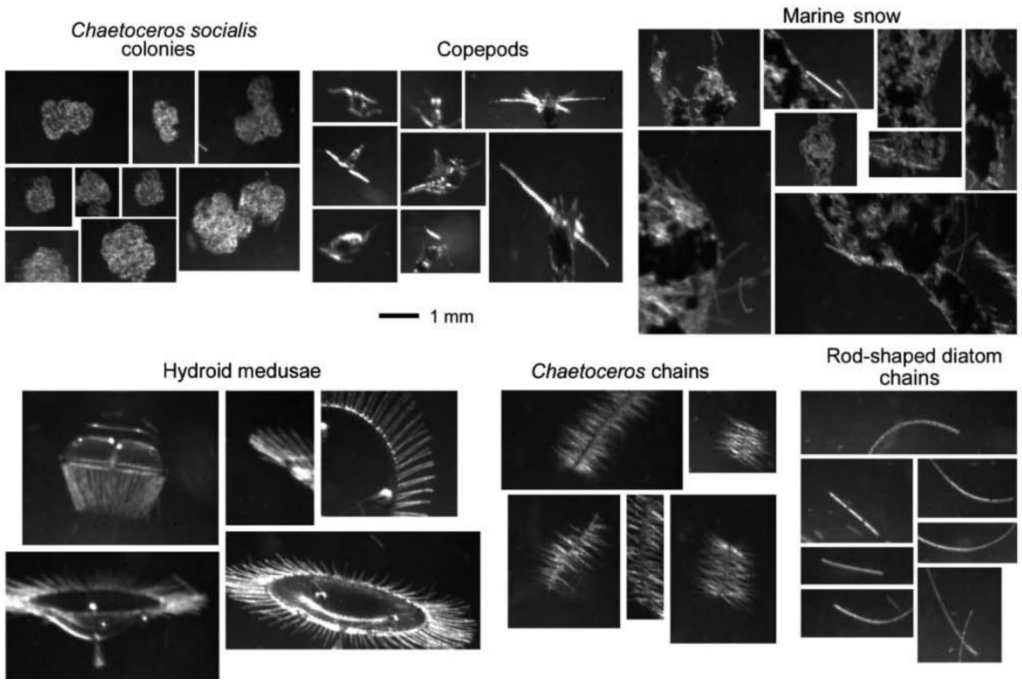


Figure 2.29: Examples of the broad planktonic classification categories classified by Hu and Davis (Davis et al. 2004).

A variation of the turning function is used by Lee to classify fish species (Lee et al. 2008). Before generating the turning function, the perimeter is reduced to between 30-50 points, as shown in figure 2.30 and these are the only features used. 97.5% accuracy is achieved for a 4-way classification task. However accuracy decreases to 87.4% for a 5-way classification task, and 73.3% with the addition of another class, for a 6-way classification task.

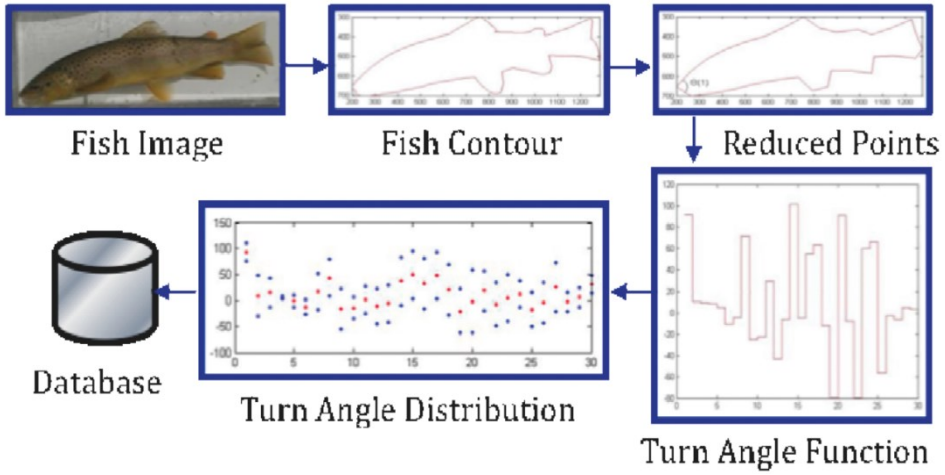


Figure 2.30: Illustration of turning function used to identify fish by their perimeter. Image from Williams et al. (2012).

Temlyakov et al. (2010) augment the Inner Distance Shape Context with two additional constructs, explicitly to handle biological objects. Their first addition is to better handle `strands', which they define to be thin, elongated, smaller than the rest of the structure, and attached at a single point. These structures are located by decomposing the original object into triangles, creating a graph based on the shared edges (Fig. 2.31), and calculating a distance based on similarity of this graph to corresponding graphs from template objects. Their observation is that the strands would tend to be highly deformable or inconsistent from one image to the next, but have minimal impact on our classification of the `core' shape, according to human perception studies they cite. Arguably, this is application dependent, but would likely be applicable in many biological classification tasks. Their second addition is the definition of an aspect ratio based on

bilateral symmetry. They then scale candidates shapes to match the aspect ratio of the templates before executing the rest of the comparison. Temlyakov et al. compare accuracies with over a dozen other methods on a standard data set, and using their additions, achieve 2% better than the next best performance.

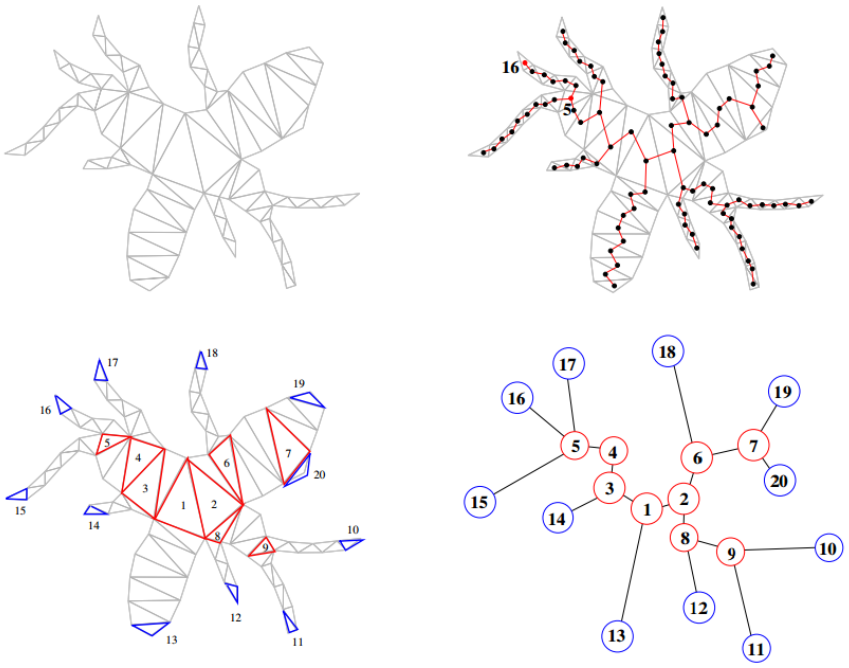


Figure 2.31: Illustration of identification of strand structures by shape-decomposition and graph transformation (Temlyakov et al. 2010). Strands are capped with blue triangles, and represented blue nodes in the graph.

The distance between Shape Cells of equivalent Shock Graphs was shown to work well for relatively simple shapes with broad variations. As later researchers applied the technique to more complex shapes, they determined that better results could be achieved by pruning the skeleton in order to facilitate comparisons. One pruning approach is outlined by Bai et al. (2007). For their approach, they simplify the perimeter of the object by first partitioning the contour with Discrete Curve Evolution, as shown in figure 2.32.

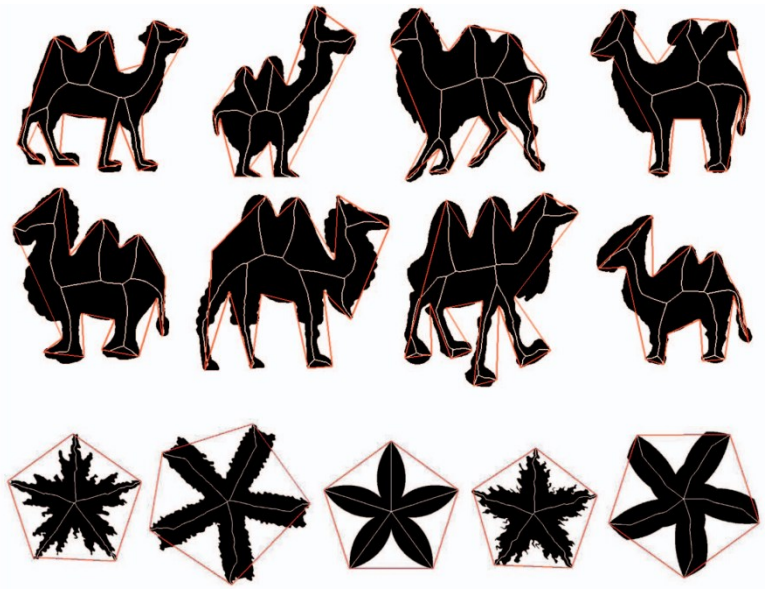


Figure 2.32: Examples of the stability of skeletons pruned with DCE (Bai et al. 2007). For each example, The red shape is the result of the original black contour being heavily smoothed with DCE. Skeleton segments which do not terminate in a convex vertex of the red simplified shape are pruned.

Nguyen, et al. (2013) introduce Spike Count to assist in the classification of pollen grains. Starting with the same set of texture features discussed earlier from Rodriguez-Damian, et al. (2006), Spike Count is introduced to help differentiate features on the perimeter of the grains. As shown in figure 2.33, Spike Count is calculated in by graphing the intensity of pixels in a band immediately outside of the segmented perimeter. Unlike the other functions in this section, these functions are not compared. Instead heuristics are used to calculate the number of local minima. This integer is then used as an additional feature. They report and increase of accuracy of around 5% on the 4 classes for which the feature is relevant. Overall the accuracy attributed to this single feature on their 9-way classification task increases from 89% to 92%.

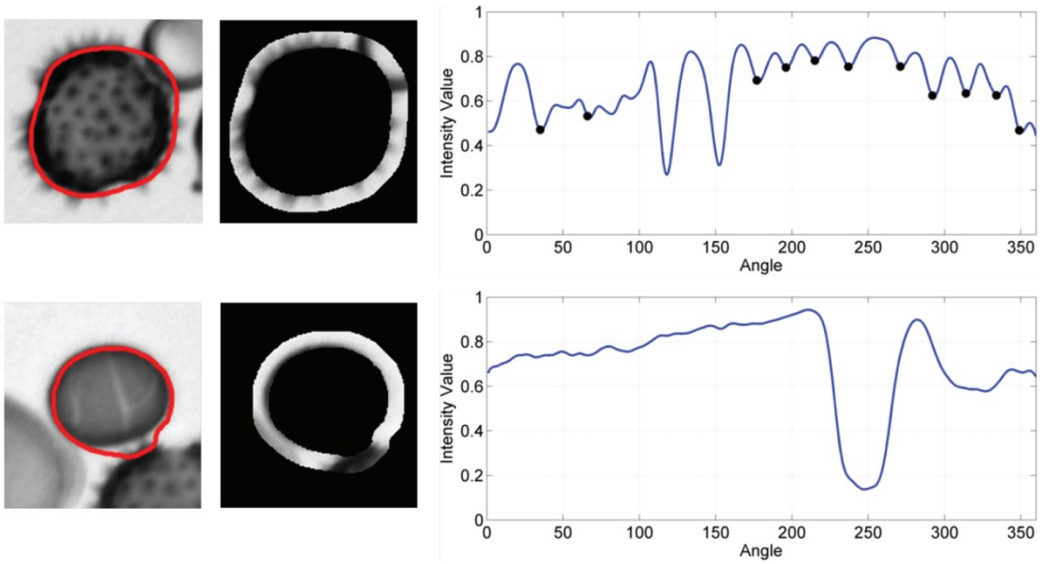


Figure 2.33: Two depictions of Spike Count from Nguyen, et al. (2013), with the top pollen grain having 11 spikes and the bottom sample having none. The graphs have the angles 0-360 on the x axis, and intensity from 0-1 as the y axis. Note that the large dip in the bottom graph, and the two largest dips in the top graph, which are all caused by adjacent grains, are correctly ignored.

2.8 Point/Patch Correspondence Methods

These methods consider portions of images at a level smaller than the whole shape. Many of these approaches involve some type of *sliding window* approach, where a very small window is repeatedly sought within the larger ROI. Although named part decomposition is not how they operate, English language descriptions in the same vein would be something such as the ROI "has two antennae" or "has a few large dark spots."

2.8.1 Fixed Heuristics

Approaches within this section can be thought of as straightforward 'Pattern/Template Matching'. They seek to match large patches of the candidate image with representative samples. The most naive is called 2D correlation, which involves looking for patches of exactly matching intensity values. The `feature' is generally a single value for each pixel and template pair, roughly equivalent to the percent match when a template centered on that pixel. This approach is

simple and effective, but suffers greatly when the viewpoint, object orientation, or lighting is inconsistent. This may not be a problem for some well-controlled biological image settings. In order to compensate for lighting, correlation can also be performed with gradient patches, which is essentially using relative intensities rather than absolute intensity values. A further refinement is to match on gradient orientation, rather than simply gradient values, which results in rotational invariance (Steger 2001)

2.8.2 Point Correspondence Based Methods

From 2004-2012, point correspondence based methods were some of the most commonly used methods with a large number of nuances and optimizations that all follow the same basic procedure. The general process for these methods is to identify visually `critical' or `interesting' points within the image, often referred to generically as *keypoints*. Each keypoint is then given a *descriptor*. The idea is to create something as unique as an English description, so that keypoints of the same visual phenomenon, but from different images, can be matched. Keypoints are ideal for recognizing one specific object, such as a landmark or building, in multiple different images. As a classification technique, keypoints identify visual elements that would directly correspond to each other in different samples. Although there are nuances in some classification methods, most often the unknown image is given the same label as the image with which it has the closest correspondence of keypoints. Or, a feature vector is created based on how similar the list of keypoints are to each other, which is similar to the methods in the *Path Matching* section above. The complexity and variations within this general approach lie in how the keypoints are determined, how the keypoint descriptors are created, and how the goodness of fit for the correspondences is evaluated.

The initial keypoint identification is covered in detail in modern computer vision textbooks (e.g., Gonzalez and Woods 2007; Szeliski 2010). In general, keypoints are designed to identify the areas with the highest contrast, because those areas should be identifiable from the widest range of lighting and angles. These high contrast keypoints are commonly located by techniques such as edge detectors (Canny, Sobel, Hough Transform), corner detectors (Shi/Tomasi), Haar wavelets, Difference of Gaussians, etc. Additionally, pre-processing techniques are frequently employed to increase the contrast in the image. This reduces noise, similar to the smoothing step in the topology based techniques covered earlier.

Computer vision textbooks also provide extensive discussion of the many types of keypoint descriptors. Scale Invariant Feature Transforms, or SIFT descriptors are one of the first and most highly cited (Lowe 2004). I summarize SIFT descriptors to provide a basis for comparison with other methods presented in this review.

A SIFT descriptor for a keypoint summarizes the magnitude and orientation of nearby gradients. Specifically the descriptor is created by computing the gradient at each nearby pixel (an 8x8 window in the left hand side of figure 2.34). These local gradients are multiplied by a Gaussian weighting factor, so that the gradients closer to the keypoint have a larger magnitude. In order to account for noise, gradients are binned. Lowe settled on 8 bins as optimal (North, North East, East, etc). Rather than simply counted as a normal histogram, magnitudes are added. The right hand side of figure 2.34 shows an illustration of the final SIFT descriptor for the keypoint, in which individual gradients have been combined by quadrant with respect to the original keypoint. In order to achieve rotational invarience, each SIFT region is not applied with respect to the image's pixels, but with the gradient at the keypoint itself set to due north. Figure 2.34 depicts a 2x2 descriptor computed from 8x8 samples, but Lowe used a 16x16 pixel window

and a grid of 4x4 descriptors when reporting results. Since each descriptor contains 8 bins, the total length of the SIFT descriptor is 128. SIFT descriptors are features for an individual point in the image.

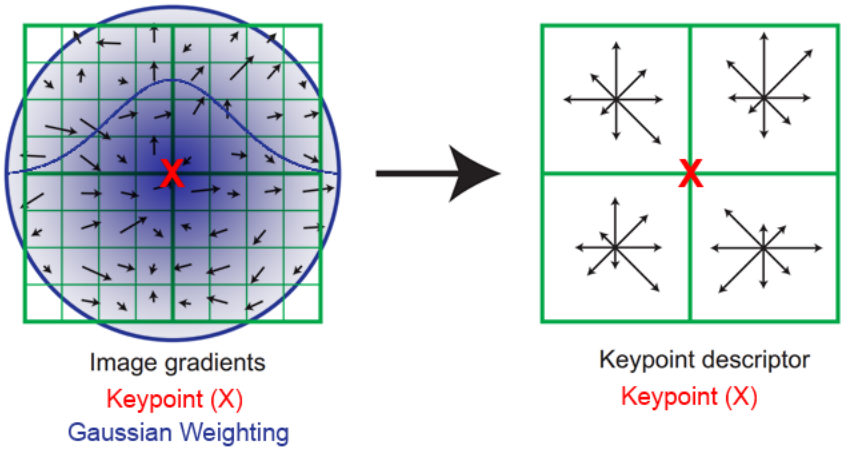


Figure 2.34: Illustration adopted from Lowe's original SIFT descriptor paper (Lowe 2004). The keypoint is at the center of both images. Image gradients are calculated for every pixel. Gradients within the radius contribute to the keypoint descriptor calculation.

Other common descriptors include SURF, HoG, GLOH, RIFT, FREAK, BRISK, FAST, BRIEF, ORB, and Gist. One of the main reasons why so many variations exist is because SIFT descriptors are computationally demanding, so many of the alternatives were generated in order to be more efficient. Tables from the performance comparison by Hudelist et al (2014) provide their relative complexity.

Finally, the descriptors must be compared in order to achieve final labeling. This can be done via Euclidean distance, Hamming distance, or any number of suitable classification or similarity strategies commonly used in machine learning.

2.8.3 Patch/Filter Based Methods

Unlike the methods in the previous section, which focus on locating particular points in the image and trying to generate correspondences, methods in this section define a particular

cluster of pixels, usually in the form of a rectangular *patch*, and then assess their presence in the image. A nuance that is necessary for semantics and important for the mathematics to work correctly is that the patches are generally not directly identified in the image, but multiplicatively applied to the original image as a *filter*. To ease the semantics of discussion, the filter will be referred to as the feature, when in actuality the filter is merely the feature detector.

Filter banks are one common technique, and there are a variety of them (Varma and Zisserman 2005). One particular set, the Leung-Malik set was originally used to classify textures (Leung and Malik 2001). It is a good representative because it includes a variety of filters, as seen in figure 2.35. The authors selected "36 oriented filters, with 6 orientations, 3 scales, and 2 phases, 8 center-surround derivative filters and 4 low-pass Gaussian filters" (Leung and Malik 2001). Because filter banks look primarily at fine-grained detail, they are also sometimes referred to as `textons.' In this case, the feature vector is a summarization of the responses of each filter in the filter bank.

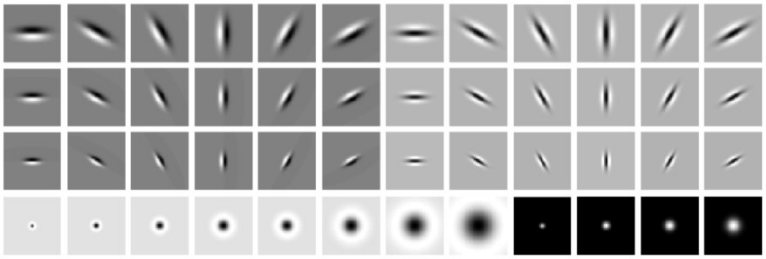


Figure 2.35: Illustration from Varma and Zisserman (2005) showing the diversity of filters in the LM Filter bank (Leung and Malik 2001).

Another commonly used technique to generate filters is Gabor Wavelets (Gabor 1946). They attempt to encode the image as if it were generated by series of sine waves in the abstract, rather than reflected photons. Although figure 2.36 depicts a symmetric arrangement, their mathematical form of a combination of a sinusoid with Gaussian decay allows them to be generated arbitrarily.

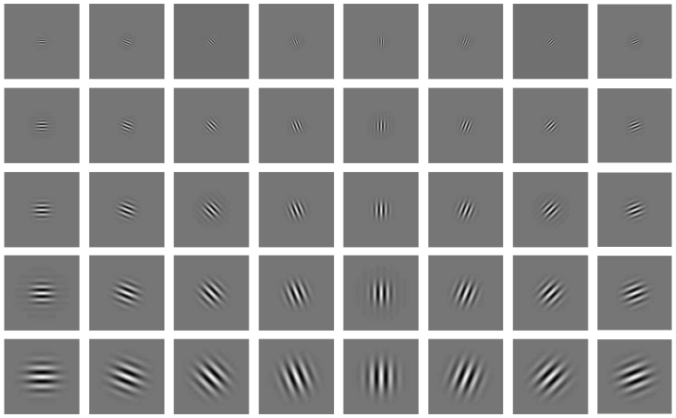


Figure 2.36: Illustration of Gabor Wavelets showing 8 different orientations and 5 different scales. Image from (Liu and Wechsler 2002).

For the past few years, there has been extensive focus on `deep learning' techniques for many kinds of learning, including supervised image classification tasks. Most of their architecture, including their ability to learn features at multiple scale, is well beyond the scope of this review. There are a number of different kinds of architecture claiming to be `deep'. Convolutional Neural Networks(CNNs) are one implementation. CNNs consist of multiple levels of neurons, each of which has a small image patch as input. As far as feature extraction is concerned, there are two primary differences between the CNN approach and the Filter Banks used by Leung and Malik (2001).

The first difference is that instead of using a set of filters *a priori*, the network is generally allowed to adjust the construction of the filters themselves to find the ones that most efficiently match the features. The method of patch construction and evolution is beyond the scope of this review, but it is frequently sparse coding, which not only has efficiency constraints, but possibly matches the behavior being executed by the neurons in humans' visual cortex (Olshausen and Field 1997). Because the filters are learned, they do not have the same level of regularity as the manually selected set of filters in a filter bank, as shown in figure 2.37.

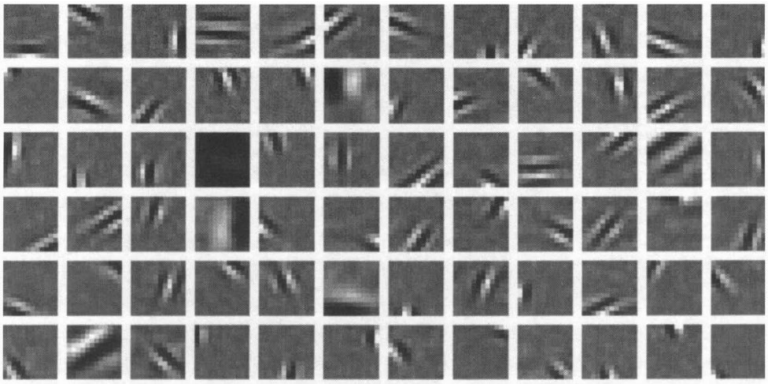


Figure 2.37: A selection filters evolved and encoded using sparse coding (Olshausen and Field 1997). This figure represents half of a bank of 144 filters. Note the lack of uniformity when compared to figures 2.35 and 2.36.

The second difference is that a separate set of filters is used (or learned) at various scales with respect to the original image. For example, if there were three sets of patches evolved, the filters typically correspond to edges at the lowest level, object parts at an intermediate level, and object models at the highest level, as seen in figure 2.38. Because calculating these image patches is only a portion of the overall computations required during the training phase of these techniques, and likely because the lowest layer tends to end up having very similar construction from one application to the next, some papers report simply using an existing, named filter bank as the lowest set of patches to expedite training.

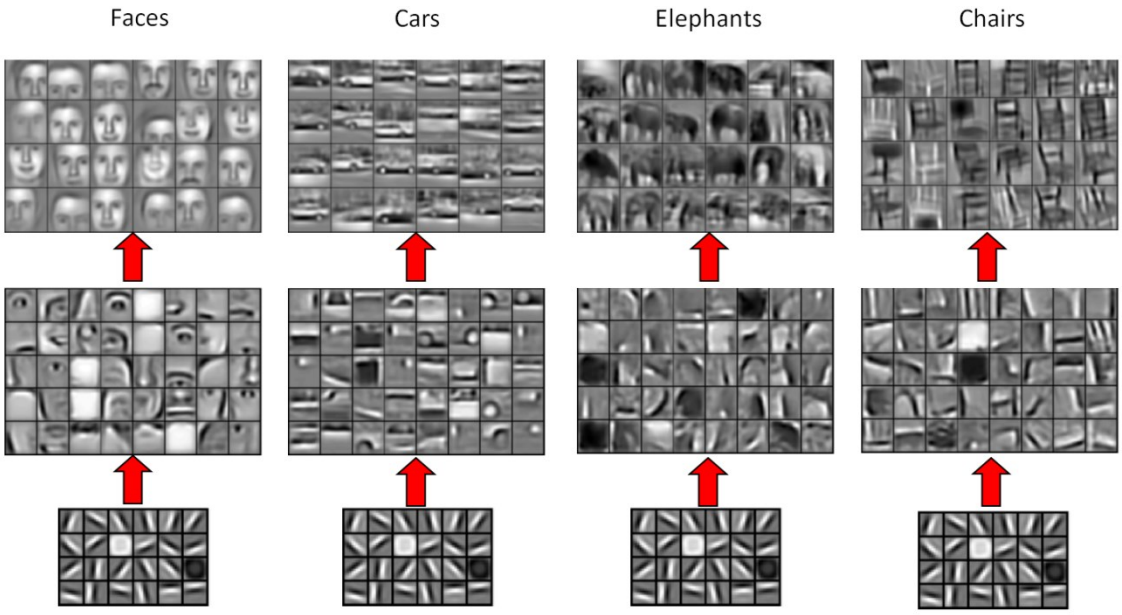


Figure 2.38: Illustration of 3 levels of filters learned by an object classifier. The lowest level would be at a small scale, along the lines of 7x7 pixel regions in the original image. Filters in successive levels are applied to pooled applications of the lower level filter responses. So, if each additional layer was also a 7x7 filter, and applied to a 4x pooled 'filter' image, the middle layer would correspond to a 28x28 pixel region in the original image, and the top layer would correspond to a 112x112 pixel region in the original image. Image adopted from (Ng and Yu 2010).

2.9 Point/Patch Correspondence Methods Specifically for Biological Object Classification

A research group at Stanford recently found simple template matching to increase performance for fine-grained object classification (Yao et al. 2012). They directly sampled `gold standard' data to acquire the templates, as shown in figure 2.39. The approach is straightforward with respect to features because it looks for exact patch matches. The key to their success is how they reduce noise by discarding nondiscriminative patches, and other machine learning optimizations.

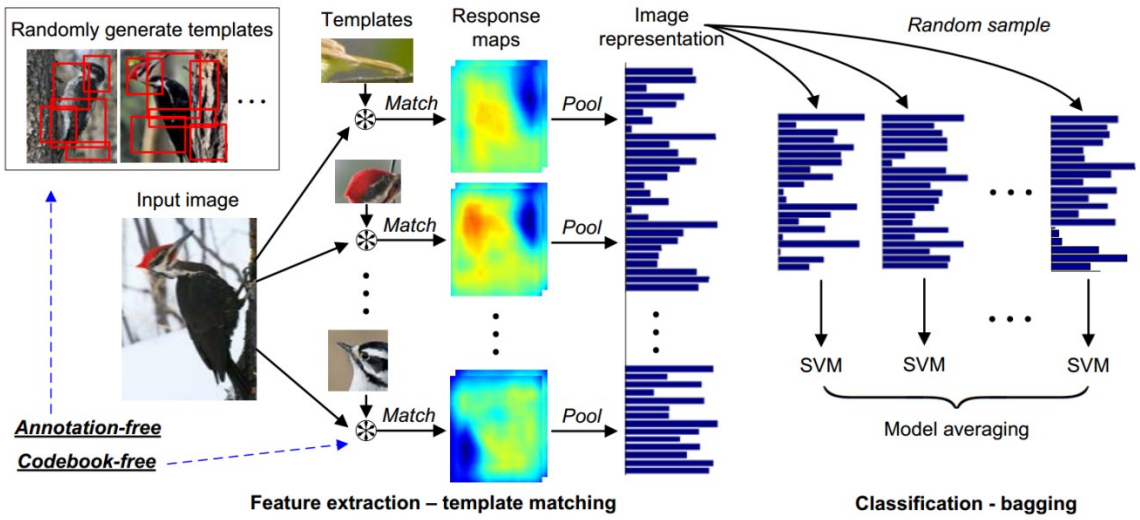


Figure 2.39: Illustration of a direct image patch matching process for classification from (Yao et al. 2012).

Ranzato, et al. (2007) customized another well-known point descriptor called *local jets* for various biological particle classification tasks. They generate a feature vector of length 108 for each pixel in the image. They reported 93.2% accuracy for a 12-way classification task of particles from urinalysis, as shown in figure 2.40. This same system was used by Edgington, et al. (2006; also Kline and Edington 2010) was used to process underwater video for invertebrate classification. They reported 95%-100% success on classifying larger sea urchins and sea cucumbers directly from frames of video. However, they reported terrible performance with similar classes of organisms, incluidng jellyfish, from time-lapse still imagery. Unlike many other approaches mentioned in this review, this particular technique skips the segmentation step, and classifies the ROIs directly from raw, unsgemented images.

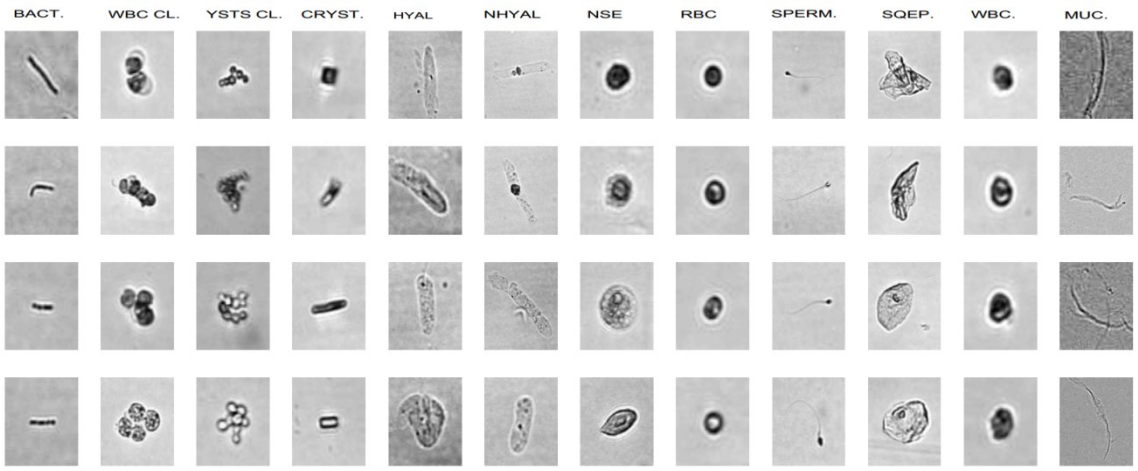


Figure 2.40: Examples of 12 classes of particles from urinalysis samples from Ranzato et al. (2007).

Zhang, et al. (2004) achieved 97% accuracy on a 5-way classification task of pollen grains. They use a set of 6 features, two second-order moment invariants, and 4 Gabor features on images such as those shown in figure 2.41. Subsequent research by Holt et al. (2011) cites the same features and adds Haralick textures and additional Gabor features to use a total of 43 features for a 6-way classification of pollen. They report accuracies ranging from 77-94%, but more significantly, that those results are within 1-4% of agreement with expert level annotations.

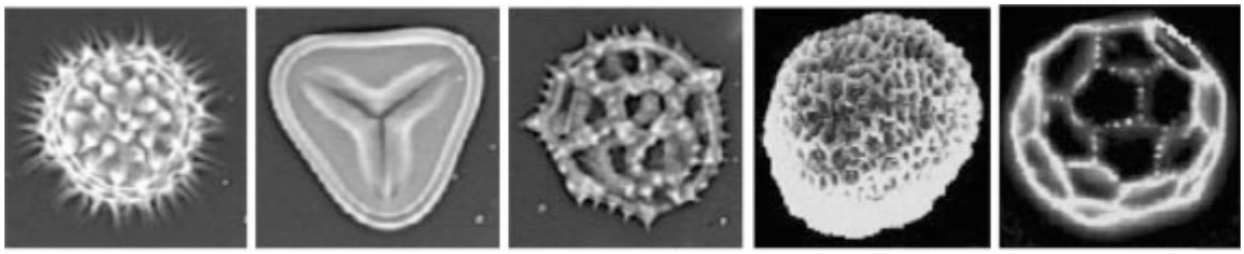


Figure 2.41: Examples of pollen classified by Gabor features in Zhang et al. 2004).

Beijbom, et al. (2012) used 24 textons generated at 4 different scales to classify types of coral within reefs survey images. Superficially, this task requires extensive segmentation, as the corals generally grow to fill all available space, as shown in figure 2.42. However, the actual census is taken by labeling the coral at individual points. They address their 9-way classification

task hierarchically, first by splitting into coral/non-coral class, and then within one of 5 coral classes, or 4 non-coral classes. The highest overall accuracy reported on the end-to-end classification of all 9 classes is 83%.

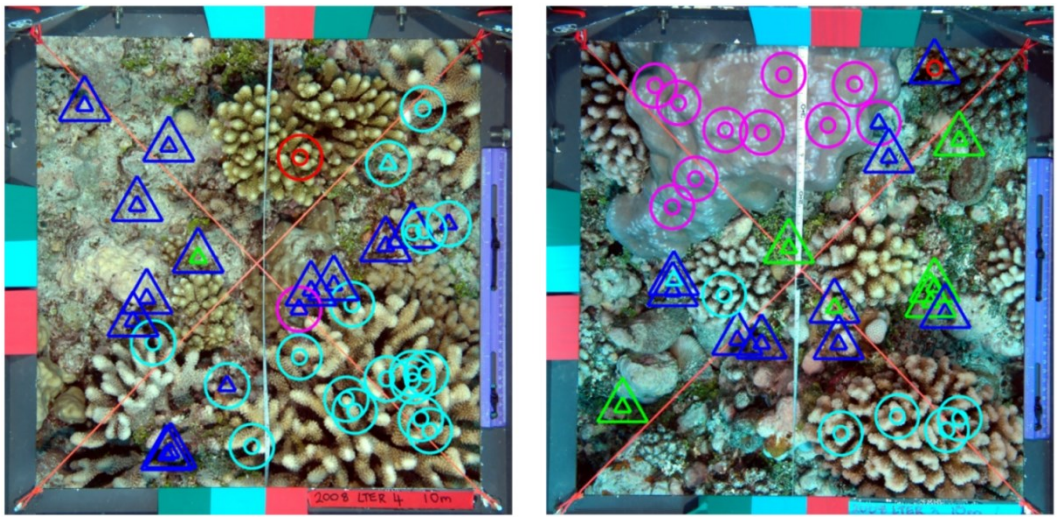


Figure 2.42: Illustration of coral reef scenes classified by textons in Beijbom, et al. (2012).

Mander, et al. (2013) also use textons to classify 12 different species of grass pollens. Overall accuracy achieved is 85.8%, which is notable because they consider the task particularly difficult: their automated process achieved better accuracy than any individual human expert, out of 7 subjects for which they reported data. The task is so difficult, they note that consensus was only achieved on 28.3% of ROIs.

Most deep learning approaches, which are patch based, report results on competition datasets such as ImageNet.

Habibzadeh, et al. (2013) classify 5 different types of white blood cells. They use a CNN, with a first level window size of 5x5 pixels. Their target images have been `segmented' as 28x28 pixels, as shown in figure 2.43. They achieve 85% accuracy with the CNN, which outperforms the 74% accuracy they report using a combination of raw intensity values and grayscale distribution moments.

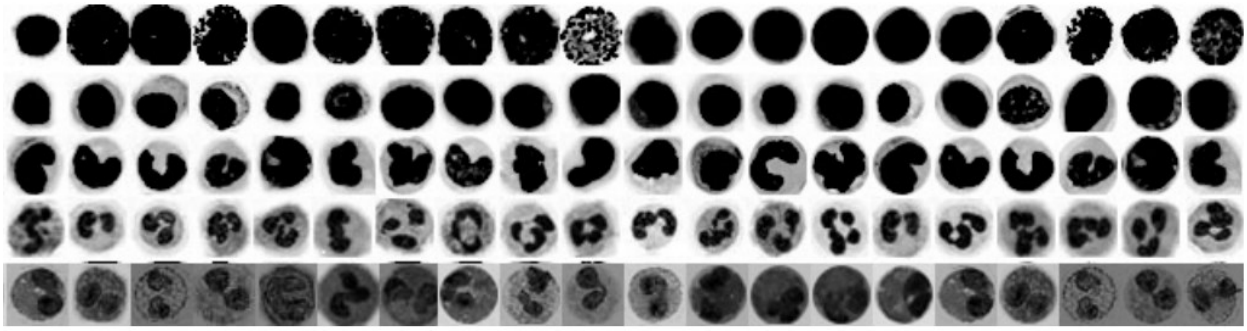


Figure 2.43: Each row represents one of the 5 different types of white blood cells classified by a Convolutional Neural Network in Habibzadeh, et al. (2013).

2.10 Discussion

There are three additional feature extraction considerations that span all of the individual algorithms mentioned above.

2.10.1 Algorithm Tuning

The important consideration is that any code which purports to implement these approaches would not be able to be used for various classification tasks directly `out of the box'. That is, some methods mentioned have parameters which would need to be derived and optimized for the classification task at hand. Prior to, and unlike the training of the classification algorithm itself, there would potentially be some application or domain specific choices required. Two examples of these choices would be how much smoothing to apply, or how far from the perimeter a skeleton should be pruned. These parameters could only be derived from careful experimentation with sample data for a particular task. Specifically, care should be given when selecting values to not only achieve optimal results on the sample data, but to accommodate future images configurations (or an understanding that reconsidering the parameters may be required).

Similarly, some of the features could be noisy, and potentially affect performance with some algorithms. Many of the references in this paper use Principal Component Analysis to reduce the number of dimensions of the features eventually made available to the machine learning algorithm. Sometimes this was done to expedite the training of the model, other times it was in order to identify the most informative features.

2.10.2 Ensemble Methods

For the most part, the cited methods were chosen not only for their applicability to biological object classification, but also for the purity, uniqueness, and originality of their approach. A logical extension for many of these methods would be to combine them, and this was illustrated in some of the biological object classification examples. For the most part, these approaches are not mutually exclusive, and could often be used in combination. The biggest potential issue with combining methods is that the calculations required to extract their features may be completely independent, so there may be no shared calculations or savings gained. The computational cost will simply be the sum of computing each individually. Also, features may not provide additional value, thereby generating noise for the machine learning algorithm to learn to ignore.

Many different types of ensembles could be considered. The first and most straightforward is simply computing all statistics for a number of different methods, and plugging them all into the same classifier. This is the approach taken by the multiple different research groups that have used the WND-CHARM image classification software published by the Goldberg research group at NIH (Orlov et al. 2008). This software computes 2873 individual features for each image, including many mentioned in this review (e.g. edge statistics, textures

(including Haralick), moments (including Zernike), fractal features, and Fourier transforms). Another ensemble architecture would be to use a separate machine learning classifier for each distinct feature type, and then to combine the results of each individual classifier, either through simple voting, or through a more complex scheme.

2.10.3 Deep Learning

For the past few years, deep learning techniques have been setting high-water marks in most vision contests as well as many contests outside of vision (e.g., Russakovsky et al. 2014). So perhaps in the near future, this entire review will be moot, if deep learning turns out to be a panacea for all vision tasks. However, there are a few reasons why this review is currently still potentially relevant even in that case. First, the deep learning approaches as currently deployed are computationally expensive. An ICML tutorial from 2013 given by Yann LeCun referenced a network with 8 layers, training on 1.3 million images, which required 10 days of training time on a single GPU (LeCun and Ranzato 2013). For individual researchers this may be a significant cost with no guarantee that everything will work correctly, so there might be multiple 10-day waiting periods during debugging and model creation. And while Google's submission to the 2014 ImageNet competition raised the bar on performance, it also raised the bar on resources. They trained 7 separate 22-layer networks, and used an ensemble method to combine the output of all (Szegedy et al. 2014). They claim to be more efficient than a similar submission from 2012, but in the near term, this approach may be beyond the resources for a classification effort without access to supercomputer level resources.

Additionally, most contests are designed with different objectives than those stated at the start of this review. For example, ImageNet is a popular contest, but contains 1000 categories, which are randomly selected subclasses of broader categories such as “mammal, bird, fish,

reptile, amphibian, vehicle, furniture, musical instrument, geological formation, tool, flower, fruit" (Deng et al. 2009). The 1000 target classes contain things such as "English setter", "Australian terrier", etc. These type of full-color, macroscale images are not representative of the type of biological image classification targeted by this review, and performance may not translate. In addition, these types of 1000-way classification tasks still have error rates of 7.4% (Russakovsky et al. 2014). So even if these techniques are more powerful overall for general tasks, they may not be sufficiently accurate for a narrow, high precision classification task.

Finally, Deep Learning approaches may greatly improve performance, but by design they are built to be general-purpose, so that they may be learned in an unsupervised fashion. This is so that they scale to accomplish large-scale performance across a huge variety of tasks without requiring customization for each one. However, if a particular task is very specific, and very high value, it may still be worth engineering a specific methodology. The `custom' approach may or may not perform better, but combinations are also an option. For example, in a recent paper entitled *DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition* (Donahue et al. 2013), the authors claimed to have achieved best in class performance on a number of classification tasks, including fine-grained classification of very similar species of birds, from the Caltech-UCSD birds 200 data set (Welinder et al. 2010). While it is true that their DeCAF architecture outperformed the other reported methods by two percent, an additional six percent gain was achieved by combining their DeCAF approach with one of the previous best approaches, *Deformable Part Descriptors* (Zhang et al. 2013).

2.11 Conclusion

This chapter provides a summary of many types of features that could be used to automate biological object classification. I organize them into three different categories as shown

in figure 2.44, provide a brief interpretation, and cite published implementations for representative domains. The feature extraction methods covered in this review are presented comprehensively, not comparatively. As the cited examples illustrate, each feature has been found more useful than some other feature for at least one application. More importantly, there is a clear tendency to aggregate features rather than to replace older ones outright. Therefore, the information in this review should remain valuable even as methods evolve and new ones are invented.

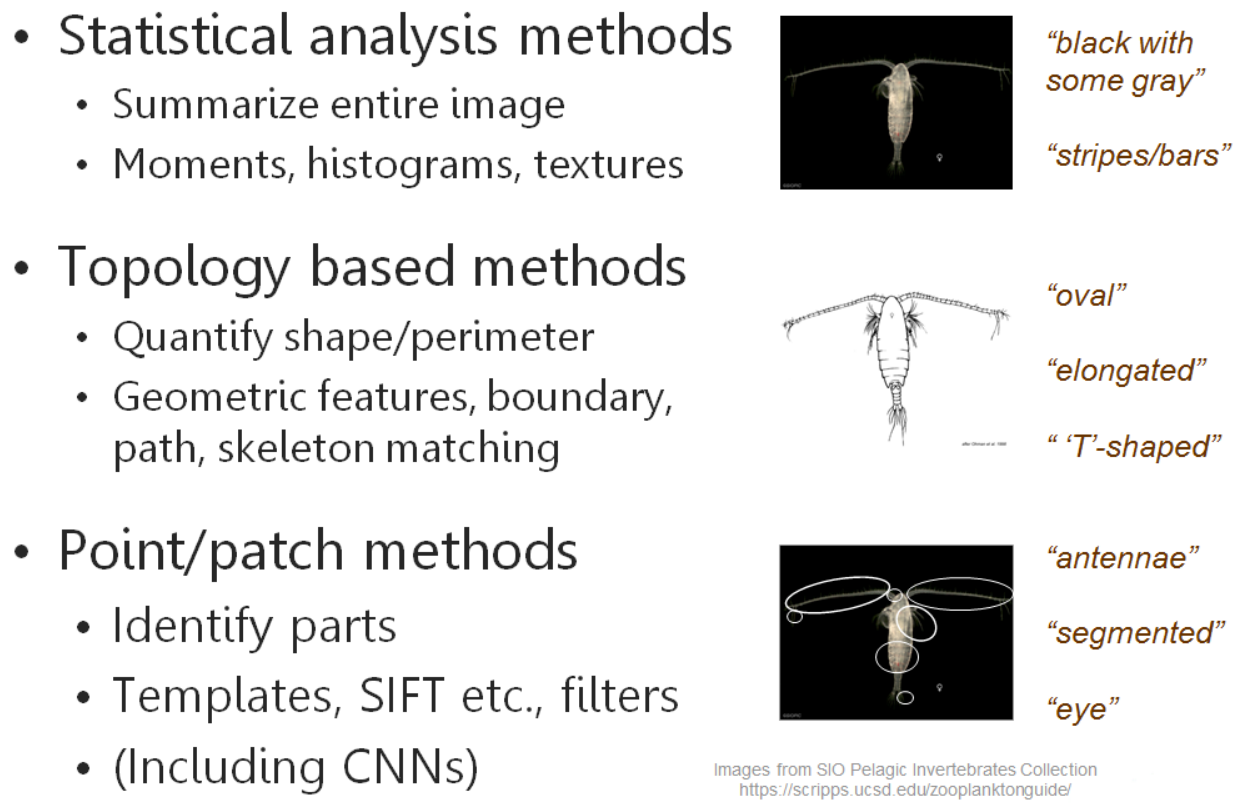


Figure 2.44: Summary of the three categories of feature extraction methods presented in this review.

2.12 Acknowledgements

Thanks to my co-advisors, Charles Elkan and Mark Ohman for their constructive commentary and patience. Additional thanks to my research exam committee: Pavel Pevzner, Lawrence Saul, and Ravi Ramamoorthi.

2.13 References

Michael D Abr`amoff, Paulo J Magalh˜aes, and Sunanda J Ram. 2004. Image processing with ImageJ. Biophotonics international 11, 7 (2004), 36–43.

Andr´e Ricardo Backes and Odemir Martinez Bruno. 2010. Shape classification using complex network and multi-scale fractal dimension. Pattern Recognition Letters 31, 1 (2010), 44–51.

Andr´e Ricardo Backes, Dalcimar Casanova, and Odemir Martinez Bruno. 2009. A complex network-based approach for boundary shape analysis. Pattern Recognition 42, 1 (2009), 54–67.

Xiang Bai, Longin Jan Latecki, and Wen-Yu Liu. 2007. Skeleton pruning by contour partitioning with discrete curve evolution.

Pattern Analysis and Machine Intelligence, IEEE Transactions on 29, 3 (2007), 449–462.

Ronen Basri, Luiz Costa, Davi Geiger, and David Jacobs. 1998. Determining the similarity of deformable shapes. Vision Research 38, 15 (1998), 2365–2385.

Oscar Beijbom, Peter J Edmunds, David I Kline, B Greg Mitchell, and David Kriegman. 2012. Automated annotation of coral reef survey images. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 1170–1177.

Serge Belongie, Jitendra Malik, and Jan Puzicha. 2002. Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24, 4 (2002), 509–522.

Harry Blum and others. 1967. A transformation for extracting new descriptors of shape. Models for the perception of speech and visual form 19, 5 (1967), 362–380.

Bastiaan J Boom, Jiyin He, Simone Palazzo, Phoenix X Huang, Cigdem Beyan, Hsiu-Mei Chou, Fang-Pang Lin, Concetto Spampinato, and Robert B Fisher. 2013. A research tool for long-term and continuous analysis of fish assemblage in coral-reefs using underwater camera footage. Ecological Informatics (2013).

Albert Cardona and Pavel Tomancak. 2012. Current challenges in open-source bioimage informatics. nature methods 9, 7 (2012), 661–665.

Anne E Carpenter, Thouis R Jones, Michael R Lamprecht, Colin Clarke, In H Kang, Ola Friman, David A Guertin, Joo H Chang, Robert A Lindquist, Jason Moffat, and others. 2006. CellProfiler: image analysis software for identifying and quantifying cell phenotypes. Genome biology 7, 10 (2006), R100.

Amina Chebira, Yann Barbotin, Charles Jackson, Thomas Merryman, Gowri Srinivasa, Robert F Murphy, and Jelena Kova˘cevi´c. 2007. A multiresolution approach to automated

classification of protein subcellular location images. BMC bioinformatics 8, 1 (2007), 210.

Danelle E Cline and Duane R Edgington. 2010. A Detection, Tracking, and Classification System for Underwater Images. VAIB 2010 - Visual Observation and Analysis of Animal and Insect Behavior (2010).

Christian Conrad, Holger Erfle, Patrick Warnat, Nathalie Daigle, Thomas L¨orch, Jan Ellenberg, Rainer Pepperkok, and Roland Eils. 2004. Automatic identification of subcellular phenotypes on human cell arrays. Genome research 14, 6 (2004), 1130–1136.

Gaudenz Danuser. 2011. Computer vision in cell biology. Cell 147, 5 (2011), 973–978.

Cabell S Davis, HU QIAO, Scott M Gallager, TANG XIAOOU, and Carin J Ashjian. 2004. Real-time observation of taxa-specific plankton distributions: an optical sampling method. Marine ecology. Progress series 284 (2004), 77–96.

Jia Deng,Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on. IEEE, 248–255.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2013. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. CoRR abs/1310.1531 (2013). http://arxiv.org/abs/1310.1531

Duane R Edgington, Danelle E Cline, Daniel Davis, Ishbel Kerkez, and J´erˆome Mariette. 2006. Detecting, tracking and classifying animals in underwater video. In OCEANS 2006. IEEE, 1–5.

Jan Flusser and Tomas Suk. 1993. Pattern recognition by affine moment invariants. Pattern recognition 26, 1 (1993), 167–174.

Dennis Gabor. 1946. Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering 93, 26 (1946), 429–441.

Gonzalez, R. C., & Woods, R. E. 2007. Digital image processing, 2nd ed. Pearson Prentice Hall, Upper Saddle River, NJ.

Gaby Gorsky, Mark D Ohman, Marc Picheral, St´ephane Gasparini, Lars Stemmann, Jean-Baptiste Romagnan, Alison Cawood, St´ephane Pesant, Carmen Garc´ıa-Comas, and Franck Prejger. 2010. Digital zooplankton image analysis using the ZooScan integrated system. Journal of Plankton Research 32, 3 (2010), 285–303.

Philippe Grosjean, Marc Picheral, Caroline Warembourg, and Gabriel Gorsky. 2004. Enumeration, measurement, and identification of net zooplankton samples using the ZOOSCAN digital imaging system. ICES Journal of Marine Science: Journal du Conseil 61, 4 (2004), 518–525.

Mehdi Habibzadeh, Adam Krzy˙zak, and Thomas Fevens. 2013. White Blood Cell Differential Counts Using Convolutional Neural Networks for Low Resolution Images. In Artificial Intelligence and Soft Computing. Springer, 263–274.

Robert M Haralick, Karthikeyan Shanmugam, and Its' Hak Dinstein. 1973. Textural features for image classification. Systems, Man and Cybernetics, IEEE Transactions on 6 (1973), 610–621.

Nathalie Harder, Beate Neumann, Michael Held, Urban Liebel, Holger Erfle, Jan Ellenberg, Roland Eils, and Karl Rohr. 2006.

Automated recognition of mitotic patterns in fluorescence microscopy images of human cells. In Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on. IEEE, 1016–1019.

K Holt, G Allen, R Hodgson, S Marsland, and J Flenley. 2011. Progress towards an automated trainable pollen location and classifier system for use in the palynology laboratory. Review of Palaeobotany and Palynology 167, 3 (2011), 175–183.

Peter Horvath, Thomas Wild, Ulrike Kutay, and Gabor Csucs. 2011. Machine Learning Improves the Precision and Robustness of High-Content Screens Using Nonlinear Multiparametric Methods to Analyze Screening Results. Journal of biomolecular screening 16, 9 (2011), 1059–1067.

Ming-Kuei Hu. 1962. Visual pattern recognition by moment invariants. Information Theory, IRE Transactions on 8, 2 (1962), 179–187.

Qiao Hu and Cabell Davis. 2005. Automatic plankton image recognition with co-occurrence matrices and support vector machine. Marine Ecology Progress Series 295 (2005), 21–31.

Qiao Hu and Cabell S Davis. 2006. Accurate automatic quantification of taxa-specific plankton abundance using dual classification with correction. (2006).

Marco A Hudelist, Claudiu Cobˆarzan, and Klaus Schoeffmann. 2014. OpenCV Performance Measurements on Mobile Devices. In Proceedings of International Conference on Multimedia Retrieval. ACM, 479.

Keiichiro Ide, Kazutaka Takahashi, Akira Kuwata, Miwa Nakamachi, and Hiroaki Saito. 2008. A rapid analysis of copepod feeding using FlowCAM. Journal of plankton research 30, 3 (2008), 275–281.

Dina Riis Johannessen. 2011. Summer School on Graphs in Computer Graphics, Image and Signal Analysis. (2011). http://www2.imm.dtu.dk/projects/graph/Graphs.html

Sotiris B Kotsiantis, ID Zaharakis, and PE Pintelas. 2007. Supervised machine learning: A review of classification techniques.

Longin Jan Latecki and Rolf Lakamper. 2000. Shape similarity measure based on correspondence of visual parts. Pattern Analysis and Machine Intelligence, IEEE Transactions on 22, 10 (2000), 1185–1190.

Yann LeCun and Marc'Aurelio Ranzato. 2013. Deep Learning Tutorial. ICML Tutorial. (2013). http://www.cs.nyu.edu/~yann/talks/lecun-ranzato-icml2013.pdf

Dah-Jye Lee, James K Archibald, Robert B Schoenberger, Aaron W Dennis, and Dennis K Shiozawa. 2008. Contour matching for fish species recognition and migration monitoring. In Applications of Computational Intelligence in Biology. Springer, 183–207.

St´ephanie Leli`evre, Elvire Antajan, and Sandrine Vaz. 2012. Comparison of traditional microscopy and digitized image analysis to identify and delineate pelagic fish egg spatial distribution. Journal of plankton research (2012), fbs015.

Pete E Lestrel. 2008. Fourier descriptors and their applications in biology. Cambridge University Press.

Thomas Leung and Jitendra Malik. 2001. Representing and recognizing the visual appearance of materials using three dimensional textons. International Journal of Computer Vision 43, 1 (2001), 29–44.

Haibin Ling and DavidWJacobs. 2007. Shape classification using the inner-distance. Pattern Analysis and Machine Intelligence, IEEE Transactions on 29, 2 (2007), 286–299.

Chengjun Liu and HarryWechsler. 2002. Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition. Image processing, IEEE Transactions on 11, 4 (2002), 467–476.

Sven Loncaric. 1998. A survey of shape analysis techniques. Pattern recognition 31, 8 (1998), 983–1001.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 2 (2004), 91–110.

Tong Luo, Kurt Kramer, Dmitry B Goldgof, Lawrence O Hall, Scott Samson, Andrew Remsen, and Thomas Hopkins. 2004.

Recognizing plankton images from the shadow image particle profiling evaluation recorder. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 34, 4 (2004), 1753–1762.

Norman MacLeod. 2007. Automated taxon identification in systematics: theory, approaches and applications. CRC Press.

Luke Mander, Mao Li, Washington Mio, Charless C Fowlkes, and Surangi W Punyasena. 2013. Classification of grass pollen through the quantitative analysis of surface ornamentation

and texture. Proceedings of the Royal Society B: Biological Sciences 280, 1770 (2013), 20131905.

Erik Meijering and Gert van Cappellen. 2007. Quantitative biological image analysis. In Imaging cellular and molecular biological functions. Springer, 45–70.

Ralf Mikut, Thomas Dickmeis, Wolfgang Driever, Pierre Geurts, Fred A Hamprecht, Bernhard X Kausler, Mar´ıa J Ledesma-Carbayo, Rapha¨el Mar´ee, Karol Mikula, Periklis Pantazis, and others. 2013. Automated processing of zebrafish imaging data: a survey. Zebrafish 10, 3 (2013), 401–421.

Farzin Mokhtarian and Alan Mackworth. 1986. Scale-based description and recognition of planar curves and two-dimensional shapes. Pattern Analysis and Machine Intelligence, IEEE Transactions on 1 (1986), 34–43.

Andrew Ng and Kai Yu. 2010. ECCV-2010 Tutorial: Feature Learning for Image Classification. (2010). http://ufldl.stanford.edu/eccv10-tutorial/

Nhat Rich Nguyen, Matina Donalson-Matasci, and Min C Shin. 2013. Improving pollen classification with less training effort. In Applications of Computer Vision (WACV), 2013 IEEE Workshop on. IEEE, 421–426.

Mark Nixon, Mark S Nixon, and Alberto S Aguado. 2012. Feature extraction & image processing for computer vision. Academic Press.

Bruno A Olshausen and David J Field. 1997. Sparse coding with an overcomplete basis set: A strategy employed by V1? Vision research 37, 23 (1997), 3311–3325.

Nikita Orlov, Lior Shamir, Tomasz Macura, Josiah Johnston, D Mark Eckley, and Ilya G Goldberg. 2008. WND-CHARM: Multi-purpose image classification using compound image transforms. Pattern recognition letters 29, 11 (2008), 1684–1693.

Eric Persoon and King-Sun Fu. 1977. Shape discrimination using Fourier descriptors. Systems, Man and Cybernetics, IEEE Transactions on 7, 3 (1977), 170–179.

Markus Peura and Jukka Iivarinen. 1997. Efficiency of simple shape descriptors. In Proceedings of the third international workshop on visual form, Vol. 443. Citeseer, 451.

MRanzato, PE Taylor, JM House, RC Flagan, Yann LeCun, and Pietro Perona. 2007. Automatic recognition of biological particles in microscopic images. Pattern Recognition Letters 28, 1 (2007), 31–39.

Maria Rodriguez-Damian, Eva Cernadas, Arno Formella, Manuel Fern´andez-Delgado, and Pilar De Sa-Otero. 2006. Automatic detection and classification of grains of pollen based on shape and texture. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 36, 4 (2006), 531–542.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2014. ImageNet Large Scale Visual Recognition Challenge. (2014).

Johannes Schindelin, Ignacio Arganda-Carreras, Erwin Frise, Verena Kaynig, Mark Longair, Tobias Pietzsch, Stephan Preibisch, Curtis Rueden, Stephan Saalfeld, Benjamin Schmid, and others. 2012. Fiji: an open-source platform for biologicalimage analysis. Nature methods 9, 7 (2012), 676–682.

Thomas Sebastian, Philip Klein, and Benjamin Kimia. 2001. Recognition of shapes by editing shock graphs. In Computer Vision, IEEE International Conference on, Vol. 1. IEEE Computer Society, 755–755.

Lior Shamir, John D Delaney, Nikita Orlov, D Mark Eckley, and Ilya G Goldberg. 2010. Pattern recognition software and techniques for biological image analysis. PLoS computational biology 6, 11 (2010), e1000974.

Mark R Shortis, Mehdi Ravanbakskh, Faisal Shaifat, Euan S Harvey, Ajmal Mian, James W Seager, Philip F Culverhouse, Danelle E Cline, and Duane R Edgington. 2013. A review of techniques for the identification and measurement of fish in underwater stereo-video image sequences. In SPIE Optical Metrology 2013. International Society for Optics and Photonics, 87910G–87910G.

Kaleem Siddiqi, Ali Shokoufandeh, Sven J Dickinson, and Steven W Zucker. 1999. Shock graphs and shape matching. International Journal of Computer Vision 35, 1 (1999), 13–32.

Heidi M Sosik and Robert J Olson. 2007. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Limnol. Oceanogr. Methods 5 (2007), 204–216.

Carsten Steger. 2001. Similarity measures for occlusion, clutter, and illumination invariant object recognition. In Pattern Recognition. Springer, 148–154.

Michael J Swain and Dana H Ballard. 1991. Color indexing. International journal of computer vision 7, 1 (1991), 11–32.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2014. Going deeper with convolutions. arXiv preprint arXiv:1409.4842 (2014).

Richard Szeliski. 2010. Computer vision: algorithms and applications. Springer.

Richard Taylor, Norman Vine, Amber York, Steve Lerner, Dvora Hart, Jonathan Howland, Lakshman Prasad, Larry Mayer, and Scott Gallager. 2008. Evolution of a Benthic Imaging System From a Towed Camera to an Automated Habitat Characterization System. Technical Report. 1–7 pages.

Michael Reed Teague. 1980. Image analysis via the general theory of moments*. JOSA 70, 8 (1980), 920–930.

Andrew Temlyakov, Brent C Munsell, Jarrell W Waggoner, and Song Wang. 2010. Two perceptually motivated strategies for shape classification. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2289–2296.

Sergios Theodoridis and Konstantinos Koutroumbas. 2008. Pattern Recognition, Fourth Edition. Academic Press.

Marco Alexander Treiber. 2010. An Introduction to Object Recognition. Springer.

Markus Ulrich and Carsten Steger. 2002. Performance comparison of 2d object recognition techniques. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences 34, 3/A (2002), 368–374.

Manik Varma and Andrew Zisserman. 2005. A statistical approach to texture classification from single images. International Journal of Computer Vision 62, 1-2 (2005), 61–81.

Junwei Wang, Xiang Bai, Xinge You, Wenyu Liu, and Longin Jan Latecki. 2012. Shape matching and classification using height functions. Pattern Recognition Letters 33, 2 (2012), 134–143.

PeterWelinder, Steve Branson, Takeshi Mita, CatherineWah, Florian Schroff, Serge Belongie, and Pietro Perona. 2010. Caltech-UCSD birds 200. (2010).

Joseph Wilder, Chetan Tonde, Ganesh Sundar, Ning Huang, Lev Barinov, Jigesh Baxi, James Bibby, Andrew Rapport, Edward Pavoni, Serena Tsang, and others. 2012. An automatic identification and monitoring system for coral reef fish. In SPIE Optical Engineering+ Applications. International Society for Optics and Photonics, 84991H–84991H.

Kresimir Williams, Chris Rooper, and John (eds) Harms. 2012. Report of the National Marine Fisheries Service Automated Image Processing Workshop. NMFS-F/SPO-121 (2012), 48.

Bangpeng Yao, Gary Bradski, and Li Fei-Fei. 2012. A codebook-free and annotation-free approach for fine-grained image categorization. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 3466–3473.

Ian T Young, Joseph E Walker, and Jack E Bowie. 1974. An analysis technique for biological shape. I. Information and control 25, 4 (1974), 357–370.

Charles T Zahn and Ralph Z Roskies. 1972. Fourier descriptors for plane closed curves. Computers, IEEE Transactions on 100, 3 (1972), 269–281.

Dengsheng Zhang and Guojun Lu. 2004. Review of shape representation and description techniques. Pattern recognition 37, 1 (2004), 1–19.

Ning Zhang, Ryan Farrell, Forrest Iandola, and Trevor Darrell. 2013. Deformable part descriptors for fine-grained recognition and attribute prediction. In Computer Vision (ICCV), 2013 IEEE International Conference on. IEEE, 729–736.

Y Zhang, DW Fountain, RM Hodgson, JR Flenley, and S Gunetileke. 2004. Towards automation of palynology 3: pollen pattern recognition using Gabor transforms and digital moments. Journal of quaternary science 19, 8 (2004), 763–768.

CHAPTER 3 Improving Object Detection and Segmentation for In Situ Plankton Images

3.1 Introduction

This chapter will discuss the steps between digital image capture and image analysis, sometimes referred to as image preprocessing. I describe techniques to improve object detection in images acquired by our autonomous Zooglider (Ohman et al. 2018). Like any new imaging system, we seek to improve its performance in order to optimize the scientific value of the images captured, this includes reducing noise in the image capture process to improve not only the aesthetics for the human, but also the suitability of the images for machine learning. These improvements primarily consist of a dynamic flat-fielding algorithm to correct for uneven background illumination and a novel two-pass segmentation algorithm for object detection, together with the open standard that we use for embedding metadata into image files. After this introduction, the rest of this chapter provides background on flat-fielding, segmentation, and embedding metadata with respect to plankton images in general, detail on our implementation, and results of applying these methods.

In image processing, "detection" is defined as determining whether or not a region of interest (ROI) is present in an image, and "segmentation" is defined as "subdividing an image into its constituent regions or objects" (Gonzalez and Woods 2007). In the case of plankton images, the background of the images will be relatively consistent, and the ROI will generally occupy a small portion of the field of view, but not necessarily densely or completely covering it.

This scenario is not unique to plankton images, but is common to many biological image analysis tasks, including cell detection in biomedicine. This scenario is also not new; the research area of biomedical cell segmentation is mature enough to have retrospective articles such as "Cell segmentation: 50 years down the road" (Meijering 2012). While the duration of previous

research activity provides an indication of the level of difficulty of the problem, it also indicates that there is a substantial body of existing methods to draw upon.

A successful segmentation algorithm will consistently identify which pixels are part of a ROI, and which are not. This process includes successful handling of the irregular, elongated, and nearly transparent structures typical of many zooplankton species. Once segmented, various properties of the ROI can be calculated for independent use or use as machine learning features, such as its topology (e.g., shape, size), or statistical summaries of the properties of the constituent pixels (e.g., average pixel value, texture descriptors).

Once measured, these calculated properties need to be associated with the image. While many software solutions exist to manage data, we elect to embed the metadata in the file itself using the Extensible Metadata Platform (XMP) format (Adobe 2001; ISO 16684-1:2012). Two reasons for embedding the metadata within the file are that embedding prevents the metadata from getting separated from the image or applied to the wrong image, and embedding the metadata using a well-known format allows other researchers to use or quickly examine a subset of the images without having to procure or deploy secondary software. Embedding metadata also facilitates their use in machine learning applications.

3.2 Prior Image Processing Techniques Extended for *Zooglider* Images

3.2.1 Flat-fielding of scientific images

Plankton images, like all scientific images, exist as a proxy that allows scientists to more easily observe or measure phenomena. The images themselves are not the object of study, therefore the intensity values recorded in the image are not sacrosanct. Manipulating intensity

values is desirable to the extent that the modifications more accurately represent the original phenomena. Improving the aesthetics of the image is secondary.

Even before digital images, scientists grappled with correcting for photographic artifacts for use in scientific measurement (Shaw 1978). Working with a cryogenically cooled CCD used in astronomy, Leach et al. (1978) described a method they called "Flat-Field Correction" to try to reduce inherent noise. Flat-fielding has been applied to plankton images by Faillettaz et al. 2016, but their algorithm was not published. Other image correction methods including histogram normalization, per image normalization, per pixel normalization, and dehazing (He et al. 2011) did not provide results as consistently as our flat fielding implementation.

3.2.2 Segmentation of plankton images

Many different segmentation algorithms have been applied to plankton. Most algorithms either locate discontinuities (intensity gradients) to define the boundary between image segments or locate similarities (uniformity by some metric) to define membership to an image segment (Gonzalez and Woods 2007).

One type of similarity-based segmentation is thresholding; the threshold can be defined a priori or dynamically determined. In the case of plankton images, the goal is to identify the uniform characteristics of the background and then remove it, leaving the remaining pixels as ROIs. Thresholding is the approach used to segment zooplankton in Zooscan images (Grosjean et al. 2004, Gorsky et al. 2010) due to the high uniformity of the background as in figure 3.1. Similarly, after transforming the image from intensity to phase congruency, thresholding is then used to segment phytoplankton in Imaging Flow CytoBot images by Sosik and Olson (2007) in figure 3.2 (left).

Figure 3.1: Example Zooscan image that can be segmented by thresholding.

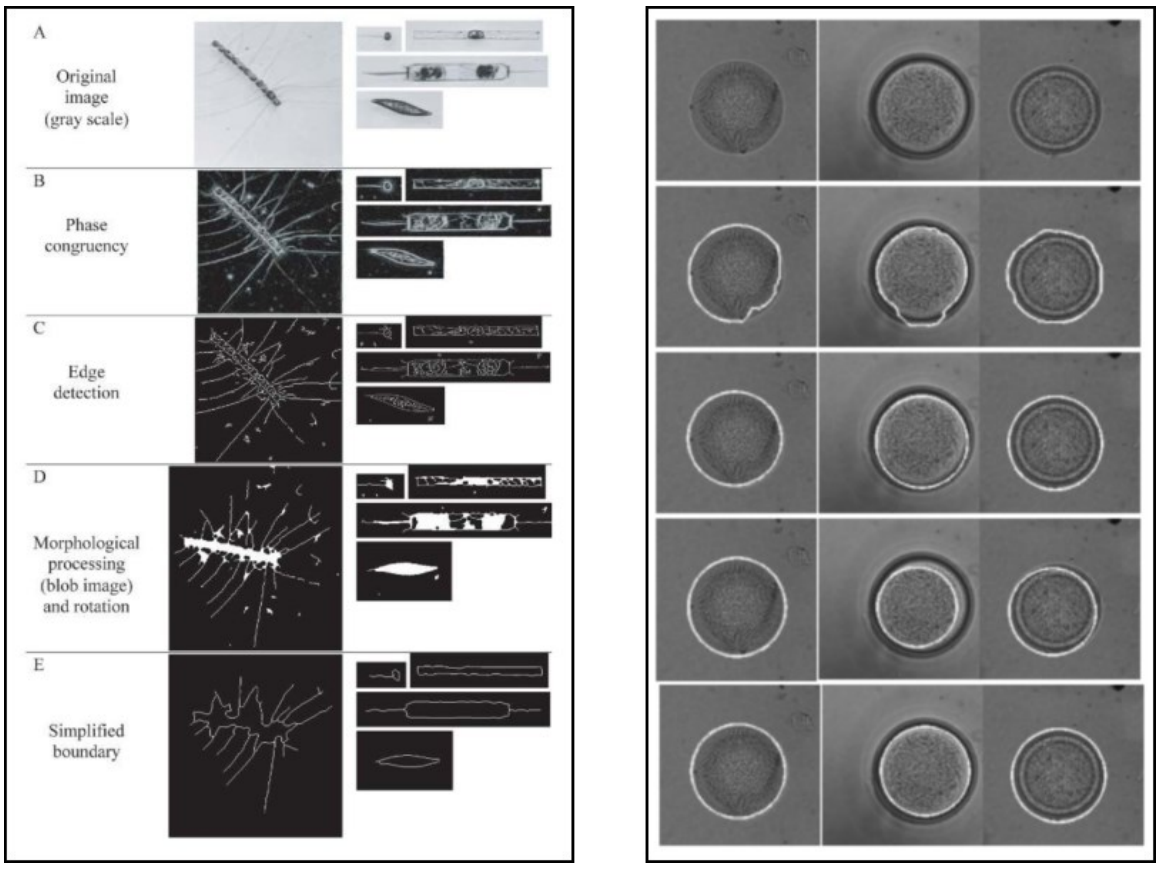


Figure 3.2: Two segmentation algorithms applied to phytoplankton. On the left, all stages of segmentation including thresholding at step C. Image from Sosik and Olson (2007). On the right, all stages of segmentation are shown for three different circular diatoms. The Canny edge detector is used to create the initial segmentation in the second row, before subsequent refinements specific to circular diatoms. Image from Luo et al. (2011).

One common discontinuity-based method is a Canny edge detector (Canny 1986). In the case of plankton images, since an edge detector only identifies boundary pixels, an additional step is required to link the edges (Gonzalez and Woods 2007). A Canny edge detector was used to segment circular diatoms in microscope images by Luo et al. (2011) in figure 3.2 (right). A different discontinuity-based method is called Active Contours, which deforms a spline curve based on an initial segmentation, yielding only smooth contours. This is implemented for

phytoplankton and small zooplankton in FlowCam images by Blaschko et al. (2005), as shown in figure 3.3 (top).

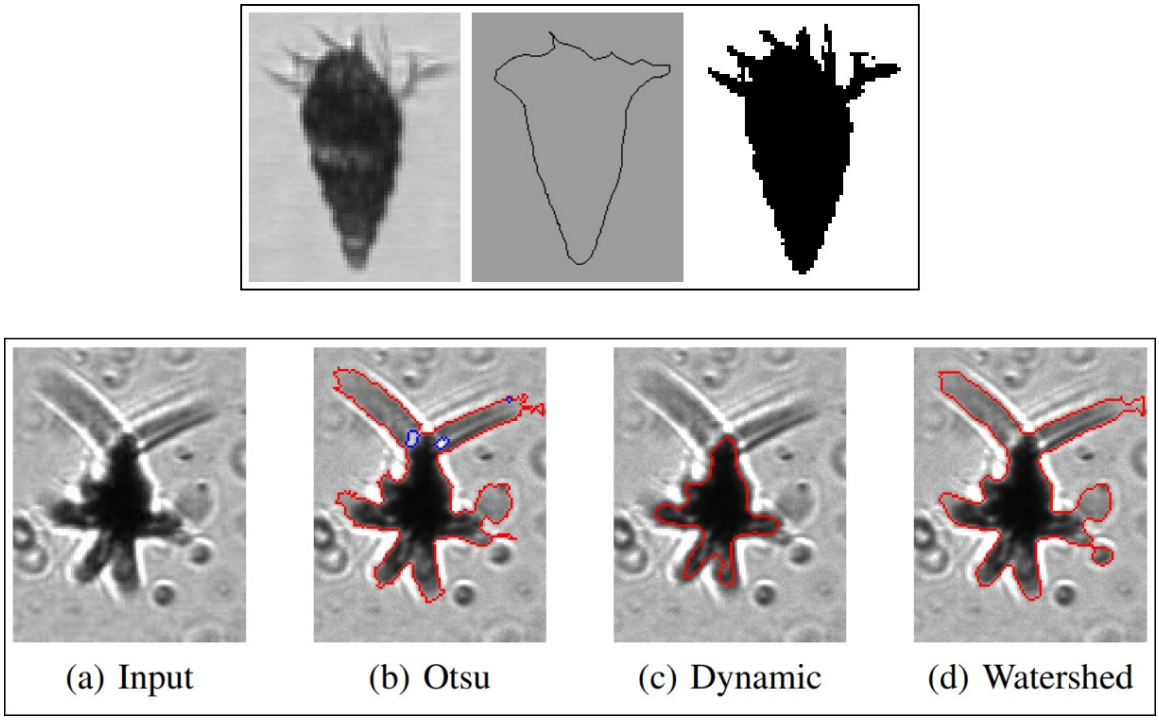


Figure 3.3: Combining segmentation algorithms. The top row has the original ciliate image (left), active contour segmentation (middle), and thresholding (right). Image from Blaschko et al. (2005). The bottom row has three additional segmentation types. Image from Hirata et al. (2016).

The choice of segmentation method is not mutually exclusive. Many workflows include mixtures of methods (Blaschko et al. 2005; Sosik and Olson 2007). Hirata et al. (2016) created an ensemble of segmentations using multiple approaches (Fig. 3.3, bottom) and let the subsequent machine learning algorithm determine which is useful. Because plankton segmentation presents challenges such as elongated structures for the ROI, all six of the previously cited studies modify the baseline segmentation algorithm, rather than just evaluating as a stock algorithm.

In addition to modified approaches, a completely novel segmentation algorithm specifically designed to capture setae (thin, elongated, hair-like extensions) is the "Grayscale Surface Direction Angle Model" by Zheng et al. (2014; Fig. 3.4). As digital image resolution improves, specialized segmentation algorithms for particular species or structures should become more feasible.

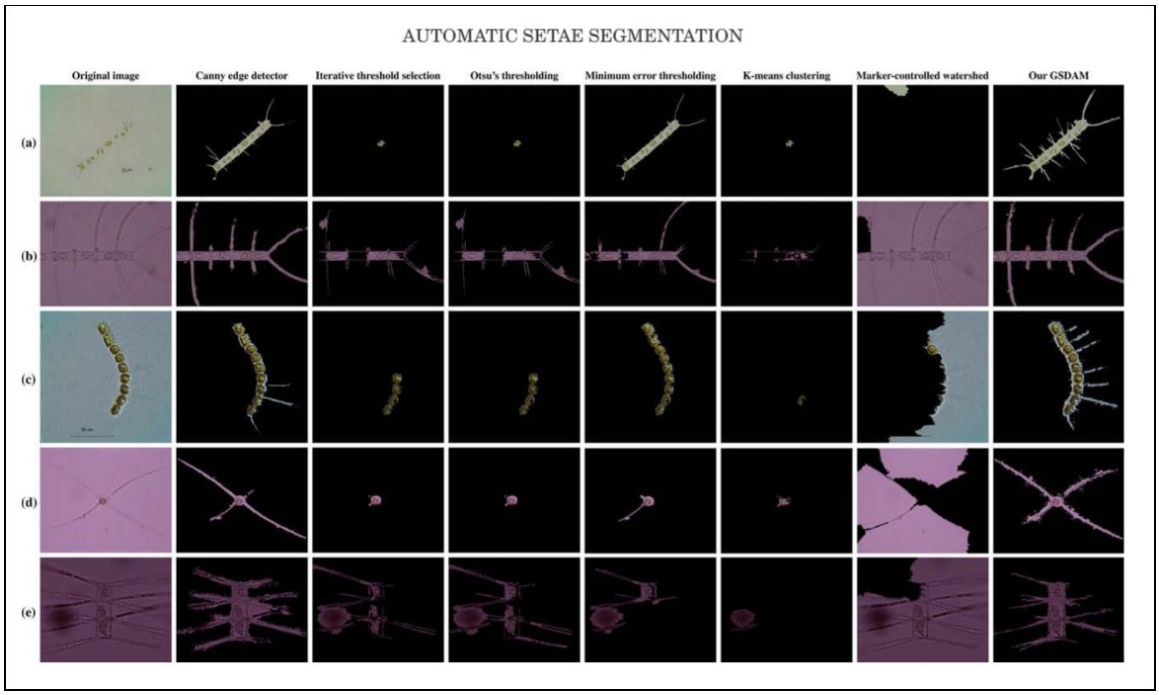


Figure 3.4: Microscopic images (leftmost column), with results shown from standard segmentation algorithms (middle columns) and from the greyscale direction angle model capturing setae (rightmost column). Image from Zheng et al. (2014).

3.2.3 Embedding Metadata with the Extensible Metadata Platform (XMP) format

When used in the context of images, the term 'metadata' refers to information about the origin and contents of the image, frequently including time, date, exposure information, and also potentially including location, description of the intended subject, etc. Rather than being maintained as a separate catalog, most contemporary digital image formats

(e.g., JPEG, PNG) allow text or binary information to be stored within the image file itself. The most common standards governing the structure and contents of this embedded data are EXIF, ITPC, and XMP (Tesic 2005). The eXtensible Metadata Platform (XMP) was first announced by Adobe Systems Incorporated in 2001 and eventually published as an open standard in 2012 by the International Organization for Standardization (ISO 16684-1:2012).

3.3 Original Image Correction and Segmentation Algorithms

3.3.1 Acquisition and Characterization of Zooglider images

The images analyzed here were acquired with *Zooglider* (Ohman et al. 2018), which is a modified *Spray* glider (Sherman et al. 2002; Davis et al. 2008). *Spray* gliders are autonomous underwater vehicles capable of 50-day ocean deployments during which they sample physical properties of the ocean during dives as deep as 400 m (Sherman et al. 2002; Davis et al. 2008). *Zooglider* additionally has a Zoocam (Fig. 3.5), a low power camera with a telecentric lens that acquires in situ images of ~250 mL of ocean per frame in order to quantify plankton and plankton-sized particles (Ohman et al. 2018). Images recorded are single-channel transmission images, like a common X-Ray, but with illumination provided by a red LED centered at 620-630 nm (Ohman et al. 2018). Images can be captured at a frame rate of up to 2Hz, which, based on the typical ascent rate of the glider of 0.1 m s^{-1}, yields a profile of images with as little as 5cm vertical separation

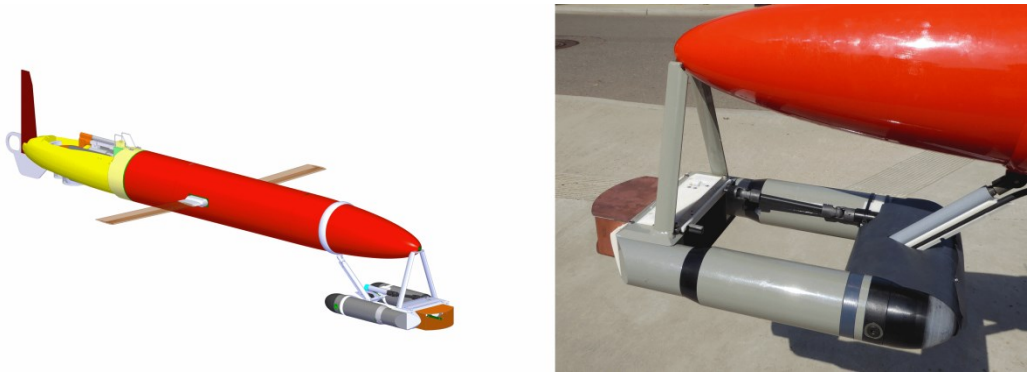


Figure 3.5: Zooglider schematic (left) and photograph of the Zoocam camera system (right)

The Instrument Development Group at Scripps Institution of Oceanography engineered and deployed the fully autonomous Zooglider, which can sample parts of the ocean not possible with any shore-based, towed, human operated, or cabled system. It is navigated remotely via the internet using Iridium satellite communications. Each time it surfaces, Zooglider telemeters ashore hydrographic measurements, chlorophyll-a fluorescence, and dual-frequency acoustic backscatter, together with an indication of size distributions of particles imaged by the Zoocam. The images themselves are downloaded upon recovery of the vehicle.

A single Zooglider ascent from 400 m depth captures approximately 8,000 full frame images over the span of approximately 65 minutes. More than a hundred such 0-400m profiles can be acquired per deployment at the full 2 Hz acquisition rate. A full frame image may have no objects present, or hundreds. Thus, efficient means for object detection and segmentation in Zooglider images are essential.

3.3.2 Flat-fielding of Zooglider Images

Similar to the telescope images considered by Leach et al. (1978), our *Zooglider* images of uniformly illuminated fields exhibited a consistent gradient of sensor response (uneven

brightness during exposures of deep water/empty frames) as in figure 3.6. Our raw intensity values also have a slight banding artifact. Therefore I implemented a correction process inspired by previous approaches, which removes these artifacts, standardizes the intensity values, and improves image contrast. The main difference between our implementation and that of Leach et al. is that instead of recording baseline frames with no expected input, we use a rolling average of adjacent frames (prior and subsequent exposures) as our average, or flat, image.

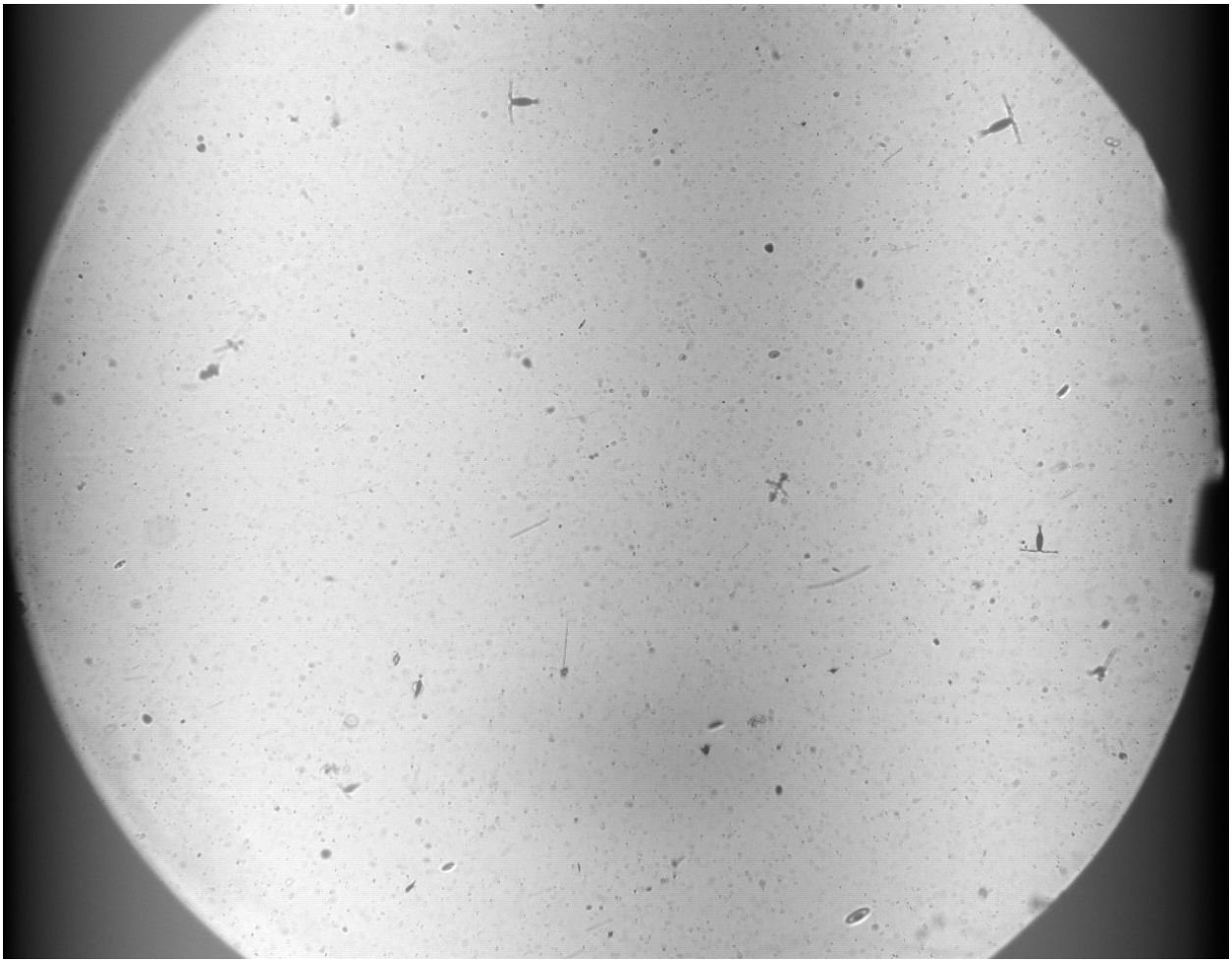


Figure 3.6: A typical Zooglider image with raw pixel values rendered as recorded in situ.

The flat field correction begins with a 100 frame rolling average (i.e., the 50 frames before and after an exposure, excepting the first 50 and last 50 images of each dive). The raw

pixel values for each frame are corrected by subtracting the 'flat-field' as follows. We calculate the single mean intensity value across the 100 adjacent images for all pixel locations, a single value between 0 and 255. We then calculate the mean intensity value across the 100 adjacent images for each pixel location, calculate the mean of those mean values, and divide each component mean by the singular mean intensity to create a correction factor matrix of values (clipped at a maximum of 1.75) yielding a correction factor matrix the same size as the image frame of values [0-1.75]. We then multiply the raw pixel values pointwise by this correction factor matrix and divide the result by the maximum value in the frame to rescale pixel values to [0-1], at which point the contrast of background areas is uniform throughout the image. We additionally increase contrast by performing gamma correction (Gonzalez and Woods 2007) of 2.2 and re-center these new pixel values to have a mean intensity of 0.812 (corresponding to greyscale value of 207). Finally we clip values below 0.0 and above 1.0 and convert back to 8-bit values [0, 255].

Pseudo-code for our flat-fielding algorithm using a rolling average

```
#Calculate mean of stack of 100 frames, and the global mean
Pix_c[j,k] = image_pixels              #Pixels of current frame
PixMean[j,k] = mean( Σ Pix_i[j,k] : for i = [c-50,c+50] )
PG = mean( PM[j,k] )


#Use the clipped mean to adjust the current image
CF[j,k] = PG / PM[j,k]                 # / is element-wise division
if CF[j,k]>1.75:
      set CF[j,k]=1.75
PixCorr[j,k]  = Pix_c[j,k] * CF[j,k]  # * is element-wise mult
```

Figure 3.7: Pseudocode for flat-fielding *Zooglider* images.

```
#Convert pixel values from 8-bit to 0-1 space for gamma corr.
PCmax = maximum( PixCorr [j,k] )
PixCorrNorm[j,k] = PixCorr [j,k] / PCmax


#Now normalized so 0 <= PCN[j,k] <=1, perform gamma correction
gamma = 2.2
PixCorrGamma[j,k] = PCN[j,k]^gamma


#For aesthetics and consistency, Re-center distribution so that
mean intensity is 207/255. Clip values that are adjusted too far.
PixCorrGammaMean = mean ( PCG[j,k] )
PixCorrRecenter[j,k] = PCG[j,k] * 0.812/PCGM
if PixCorrRecenter[j,k]<0:
      PCR[j,k]=0
if PixCorrRecenter[j,k]>1:
      PCR[j,k]=1
PixFlatField[j,k] = PCR[j,k] * 255
```

Figure 3.8: Pseudocode for flat-fielding *Zooglider* images (continued).

3.3.3 Segmentation of Zooglider Images

Segmentation of Zooglider images presents challenges, with the primary concern that large portions of nearly transparent ROIs, such as medusae, are difficult to distinguish from the background. Based on the dynamic background within a single frame, the numerous small particles that are also present, and the highly variable ocean conditions, we did not find success using thresholding or any other similarity-based approach (Gonzalez and Woods 2007).

Regions of Interest (ROIs) were segmented with a novel algorithm using two passes of an edge detector (Canny 1986). Our first pass uses less sensitive settings to generate detection

regions where at least some strong edges will be present. Because the perimeters of many of our target ROI contain portions that are extremely thin or nearly transparent we perform a second, more sensitive pass to capture these fine-grained details. This second pass is used as the actual segmented perimeter used for geometric feature calculation and ROI retention, but only if the first pass also indicates that some portion of the perimeter was part of a strong detection region. We also created unique handling for high coincidence frames and ROI at the edge of the frame. We used the Python implementation of OpenCV (Bradski and Kaehler 2000) as well as Scipy (Oliphant 2007) and its scikit-image component (van der Walt et al. 2014), because neither implementation alone provided direct access to all parameter values required. All thresholds and kernel values were determined after extensive evaluation of possible values.

For our first pass, we blur using a 13-pixel-wide Gaussian kernel with $\sigma = 1.5$ and calculate directional gradients using the same filter. We then perform Canny segmentation with a low threshold of 8 and a high threshold of 20 (note that Canny (1986) recommended a ratio of 2:1 or 3:1). Following Canny (1986), we retain all of the highest threshold edges and moderate edges if they are 8-connected to a strong edge (adjacent or diagonal). To merge nearby line segments into continuous perimeters, we perform dilation with a 5x5 structuring element, and since the purpose of the regions is detection we leave them dilated. We then perform flood-fill using 4-connected neighbors (adjacent but not diagonal). We retain these detection regions if their area exceeds 100 pixels (corresponding to a roughly 30 pixel or larger area before dilation).

Our second pass also uses blurs and calculates gradients using a Gaussian kernel of size 13, but with $\sigma = 1.75$. This pass performs thresholding using low and high values of 25 and 35. We again perform dilation with a 5x5 structuring element, but then perform erosion with the

same 5x5 element so that these perimeters closely match the intensity boundaries. We fill as before and discard all regions with an area less than 30 pixels.

We then use the first pass as a detector, discarding all second-pass regions that do not overlap with a region from the first pass. If the candidate region has an area less than 100 pixels, we count it but do not record the image tile or any geometric statistics. If the candidate region's area is greater than 100 pixels we retain the ROI as an individual PNG image and calculate geometric features (e.g. area, min/mean/max intensity) and embed these as XMP formatted metadata.

As quality control, we perform a check against coincidence. We found that in frames with a large number of diatoms or marine snow, the entire field of view is returned as a single latticed ROI. So if the second pass returns greater than 5% of the pixels as edges of candidate regions, the edges are discarded, and another attempt is made using a low threshold of 38 and a high threshold of 52, with an otherwise identical procedure. If that still yields greater than 5% of the pixels as edges, a tertiary attempt is performed with a low threshold of 50 and a high threshold of 104, and these ROI are retained regardless of the ratio of edges to original frame.

The Canny algorithm does not include frame boundaries as edges. Since many objects of interest, including larger ROI, will be partially out of the frame, we consider all top and bottom pixels to be candidate edges, as well as all the pixels that are eroded when a 3x3 cross is applied to the mask.

Pseudo-code for segmenting an image

First Pass Canny

```
# Blur the current frame
Pix[j,k] = image_pixels             #Pixels of current frame
PixBlur[j,k] = Pix[j,k] * GaussianKernel( 13,1.5 )
                                    #where * is convolution


#calculate the magnitude of the directional gradients
#element-wise, e.g. Python's numpy.hypot
GradX[j,k], GradY[j,k] = Directional_Gradients( PB[j,k]  )
Mag[j,k] = hypot( GX[j,k], GY[j,k] )


#Bin edges per direction by the low and high thresholds
LowThresh = 8, HighThresh = 20
PixStrong[j,k], PixWeak[j,k] =
(edges_per_direction( GX[j,k], GY[j,k], Mag[j,k], LT, HT ))


#Following Canny (1986) to keep only edges meeting criteria
Edges[j,k] = hysteresis_thresholding( PixStrong[j,k],
                                      PixWeak[j,k] )


#Dilate candidates, fill holes, and save as first pass binary
#mask which will subsequently be used as detections
Edges[j,k] =dilate( Edges[j,k], Ones[5,5] )
                                    #Ones = 5x5 array of all 1's
RegionsFirstPass[j,k] = fill_holes( Edges[j,k] )
RegionsFirstPass[j,k] = remove_small_objects( RFP[j,k], min=100 )
```

Figure 3.9: Pseudocode for two-pass segmentation of *Zooglider* images.

Second Pass Canny

```
# Again, Blur the current frame, slightly different blur
PixBlur[j,k] = Pix[j,k] * GaussianKernel( 13,1.75 )
                                #where * is convolution


#calculate the magnitude of the directional gradients
#element-wise, e.g. Python's numpy.hypot
GradX[j,k], GradY[j,k] = Directional_Gradients( PB[j,k] )
Mag[j,k] = hypot( GX[j,k], GY[j,k] )


#Bin edges per direction by the low and high thresholds
LowThresh = 25, HighThresh = 35
PixStrong[j,k], PixWeak[j,k] =
(edges_per_direction( GX[j,k], GY[j,k], Mag[j,k], LT, HT ))


#Following Canny (1986) to keep only edges meeting criteria
Edges[j,k] = hysteresis_thresholding( PixStrong[j,k],
                                      PixWeak[j,k] )


#Check to see if secondary or tertiary settings are required
If Edges[j,k] > 5% of image:
      restart second pass with LT= 38; HT = 52
If Edges[j,k] > 5% of image again:
      restart second pass with LT= 50; HT = 104


#Dilate and Erode candidates, fill holes, and save as second pass
#binary mask which will subsequently be used as boundaries
```

Figure 3.10: Pseudocode for two-pass segmentation of *Zooglider* images (continued).

```
Edges[j,k] = dilate( Edges[j,k], Ones[5,5] )

Edges[j,k] = erode( Edges[j,k], Ones[5,5] )

                                    #Erode, unlike 1st pass

RegionsSecondPass[j,k] = fill_holes( Edges[j,k] )

RegionsSecondPass [j,k] = remove_small_objects( RSP[j,k],min=30 )


Detection & Segmentation

For region in RSP[j,k]:

      If region overlaps RFP[j,k]:

            If region.area > 100:

Calculate geometric features and retain

Else if region.area > 30:

      Increment ROI count and discard
```

Figure 3.11: Pseudocode for two-pass segmentation of *Zooglider* images (continued).

3.3.4 Embedding Metadata as XMP

We use XMP as the format for embedding data in our images because it is an open, published standard supported by numerous image processing programs, and because XMP supports arbitrary (extensible) information. We use the Python-XMP-Toolkit (ESA/ESO/CRS4 2017) to store the key/value pairs of information in the image file. This includes dozens of geometric features extracted from the ROI (Gorsky et al. 2010; Ellen et al. 2015) as well as hydrographic measurements (CTD, Chl-a fluorescence) and geotemporal location and timing information (Ohman et al. 2018). An example is shown in figure 3.9.

Embedded XMP

Dive_Number	d0025
Latitude	32.8758
Longitude	-117.6428
Pressure_dbar	234.92
Local_Time	02:14:30_PST
Local_Date	2017-Sep-08
Feret_Diam_mm	29.8
ECD_mm	17.23
Major_Axis_Len_mm	29.59
Minor_Axis_Len_mm	15.98
Temp_deg_C	9.55
Salinity	34.24
Rho_kg_m-3	1027.5
Fluor	61
Pitch_deg	18
Roll_deg	9

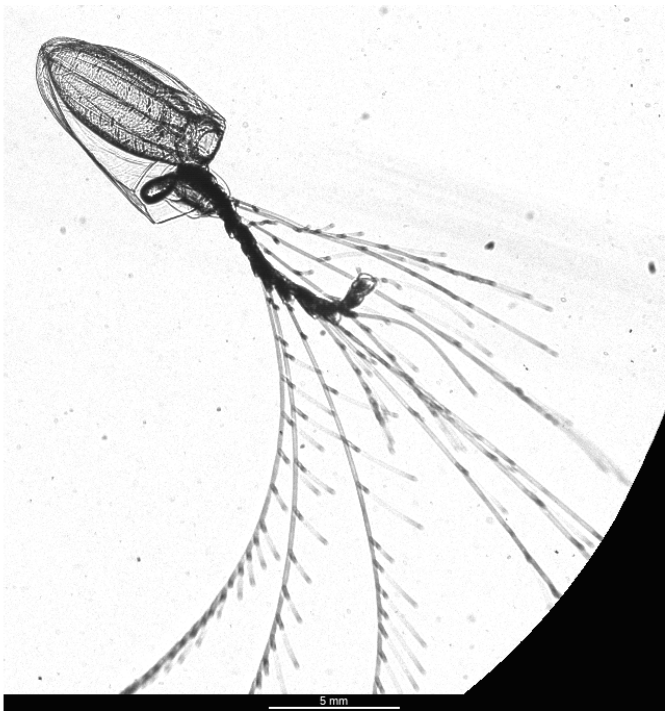


Figure 3.12: Zooglider image of a siphonophore and the first 16 data elements embedded as XMP which describe the location and time the image was captured, dimensions of the segmented boundary of the siphonophore, and hydrographic properties of the water

3.4 Results

3.4.1 Flat-fielding Successes and Limitations

Our Zooglider images Figure 3.2 shows the difference between our raw pixel values, an implementation in keeping with Leach et al. (1978), and our modification using a rolling average of 100 adjacent frames (prior and subsequent exposures) as our ‘flat’ image, as shown in figure 3.10. Our flat-fielding process worked both on images with few objects as well as images with many diatoms and marine snow particles, as shown in figure 3.11.

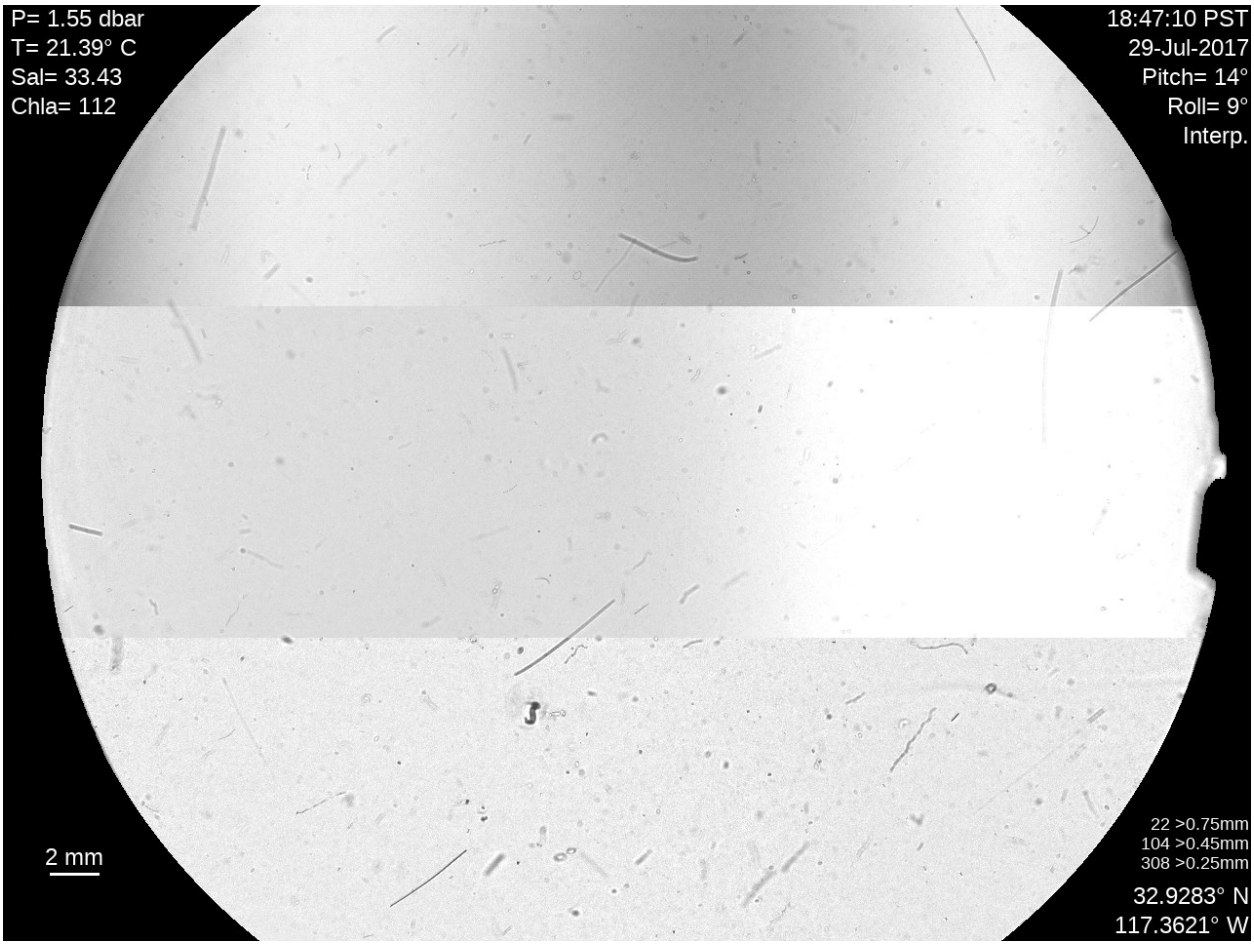


Figure 3.13: The original input (top), standard flat-fielding without a rolling average computation (middle), and our flat-fielding algorithm (bottom).

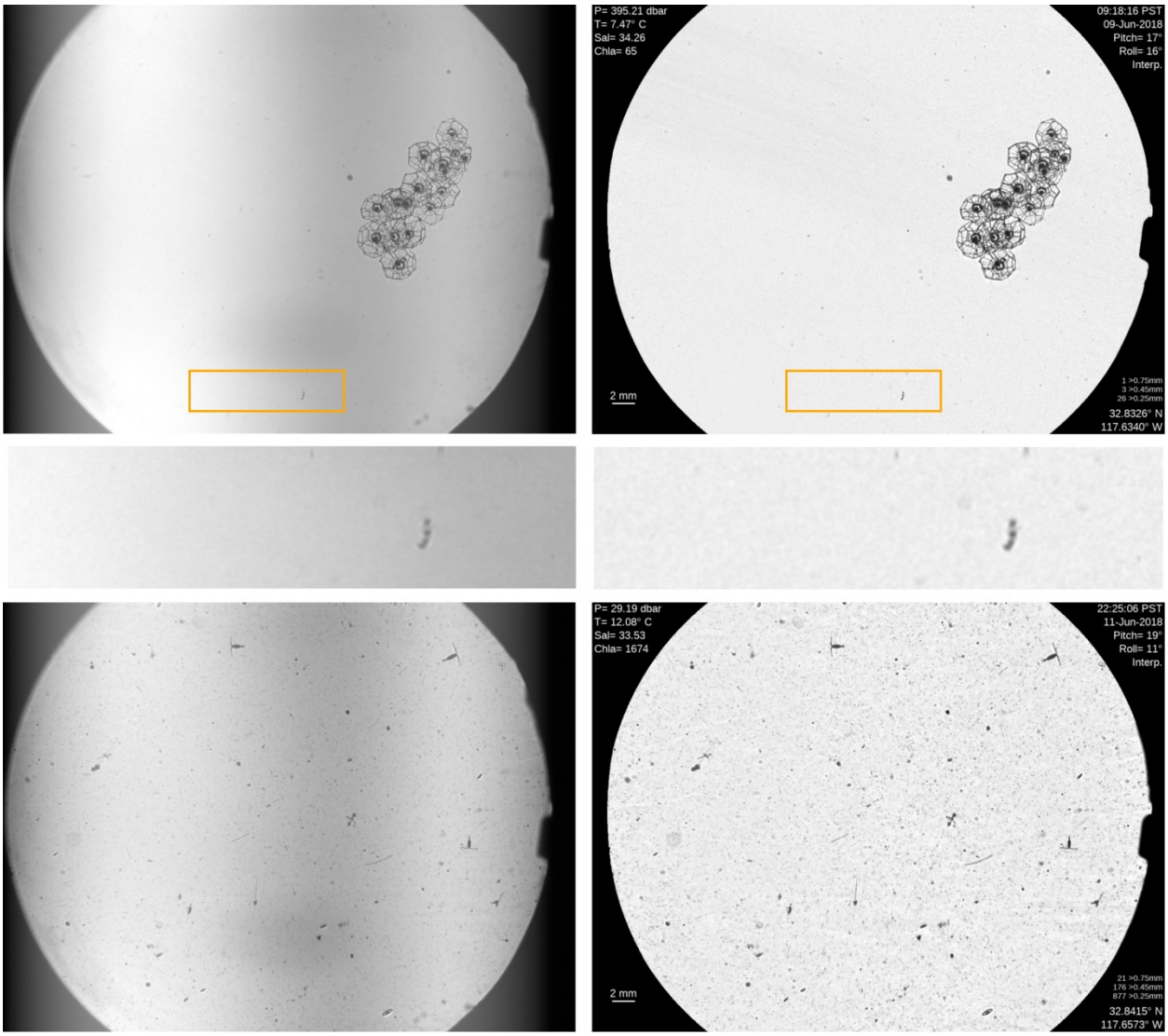


Figure 3.14: Raw images (left) and flat-fielded versions (right). Detail shown in center, showing that the gradient in intensity and banding has been corrected.

Flat-fielding not only corrects baseline image artifacts; it also corrects image defects. One deployment had an anomalous dark region in images acquired, due to a small leak caused by a faulty O-ring. The flat-fielding process was able to significantly improve the region of the image affected, as shown in figure 3.12. Larger ROIs within the damaged area are still visible.

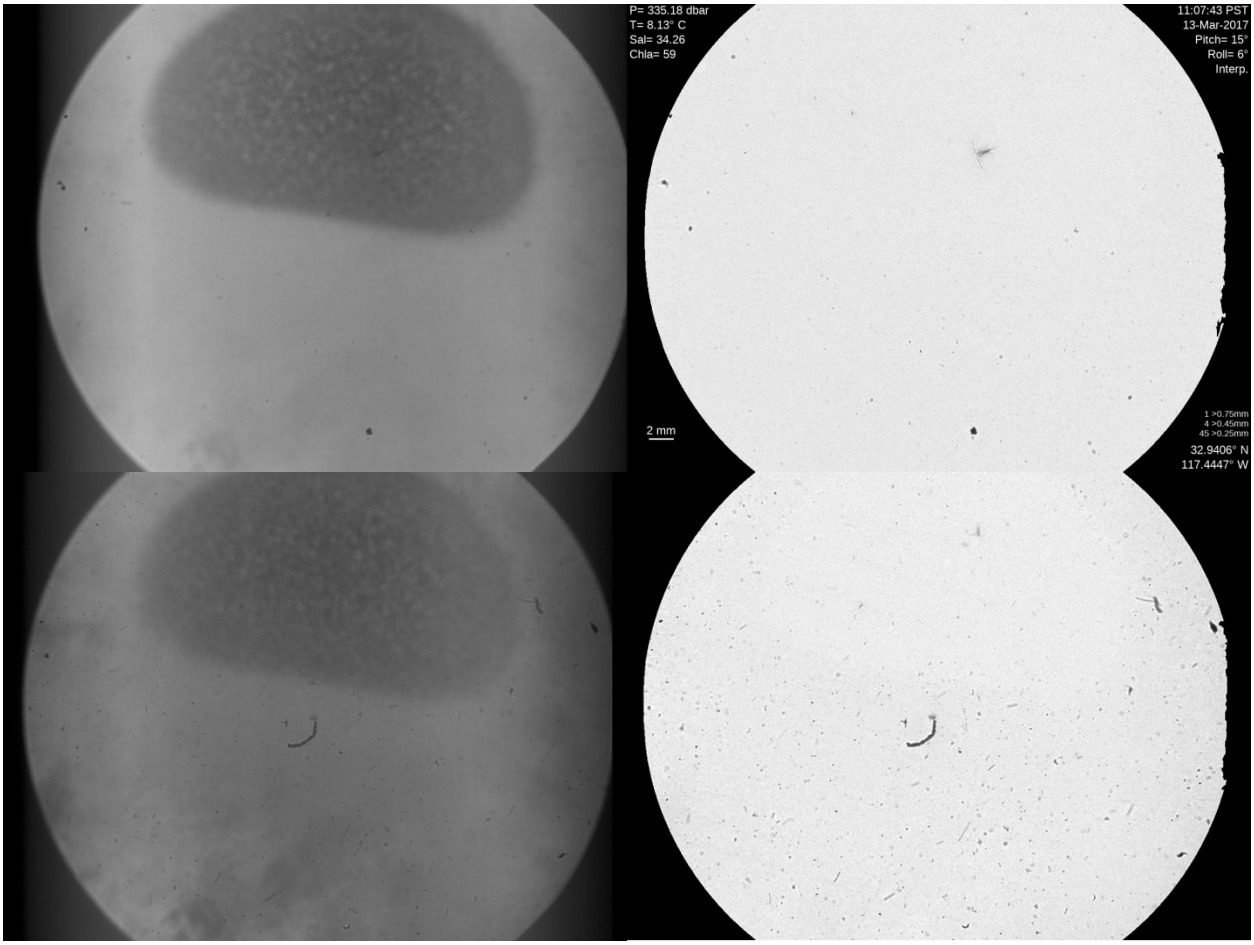


Figure 3.15: Raw image with leak (left) and flat-fielded version (right). In the top row, the blotch is removed, and the copepod that was imaged in the obscured region is preserved mostly intact. In the bottom row, the distribution of the particles is roughly uniform throughout the frame, except for the region where the dark zone had occurred, and a smaller copepod has been preserved.

3.4.2 Segmentation Successes and Limitations

Our best algorithm used a combination of two different Canny edge detectors. Our first Canny edge detector was tuned to capture only strong gradients, specifically avoiding the under 100 μm unidentifiable particles in our images (visible as hundreds of objects a few pixels in diameter). Our second Canny edge detector was tuned to capture much weaker gradients in the image in order correctly segment edges with very small gradients, as shown in figure 3.13. We

use generous edge linking criteria to capture critical elongated structures such as copepod antennae, ctenophore tentilla, and acantharian spicules, at the cost of occasionally admitting extraneous pixels.

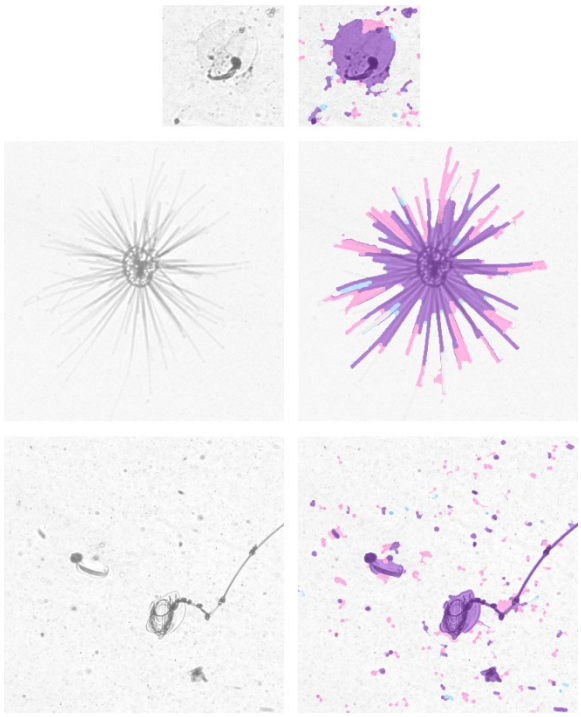


Figure 3.16: Flatfielded image (left) and pixels identified by our two different Canny edge detectors (right). Pixels highlighted in blue were designated as part of the ROI by the algorithm with less sensitive settings only. Pixels in pink were designated as part of the ROI by the algorithm with more sensitive settings only. Pixels highlighted in purple were recognized by both. Bottom image shows discarded false positive ROI (ROI with only red pixels).

In figure 3.14, pixels in blue are identified by the first pass, and serve as the initial detection. Pixels in red are identified by the second pass. In order to be retained, a candidate ROI must have at least one purple pixel, hence many of the faint, small particles in the bottom row fail to meet that criterion. A single retained ROI identified by our two-pass algorithm is the union of the purple and red pixels in a contiguous block. Some pixels are blue because of differences in our thresholds and edge-linking criteria from one pass to the next.

We then count and save a ROI as follows: ROIs that meet the segmentation algorithm but have an Equivalent Circular Diameter of <0.25mm (i.e. ECD of less than 6 pixels) are discarded, because they are near the resolution limit of the camera and therefore very inconsistent, and seem to include many false positive particles. ROI that have an ECD of <0.45mm (roughly 100 pixels in area) but ≥ 0.25 mm are enumerated but not retained. All larger ROIs are saved as individual image tiles. In figure 3.15, the ROIs that are only enumerated are highlighted in blue, while the retained ROIs are highlighted in green. ROIs are saved as individual image tiles with a scalebar and a padded margin of additional pixels around the original image (right).

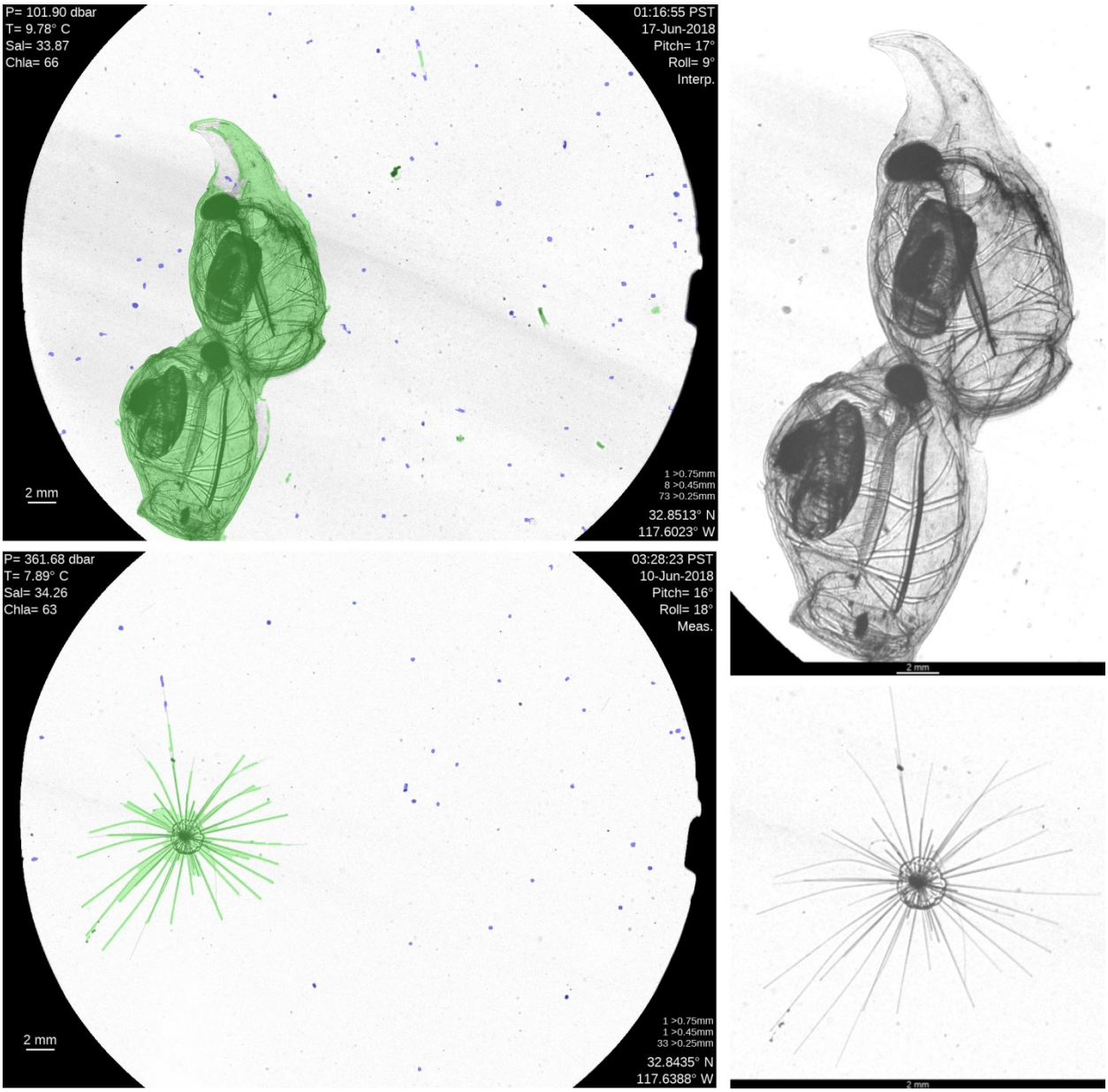


Figure 3.17: Two segmented images. Full frame images are segmented with our algorithm based on two passes of Canny segmentation (left). Blue ROIs are enumerated, green ROIs are retained as individual image tiles (right).

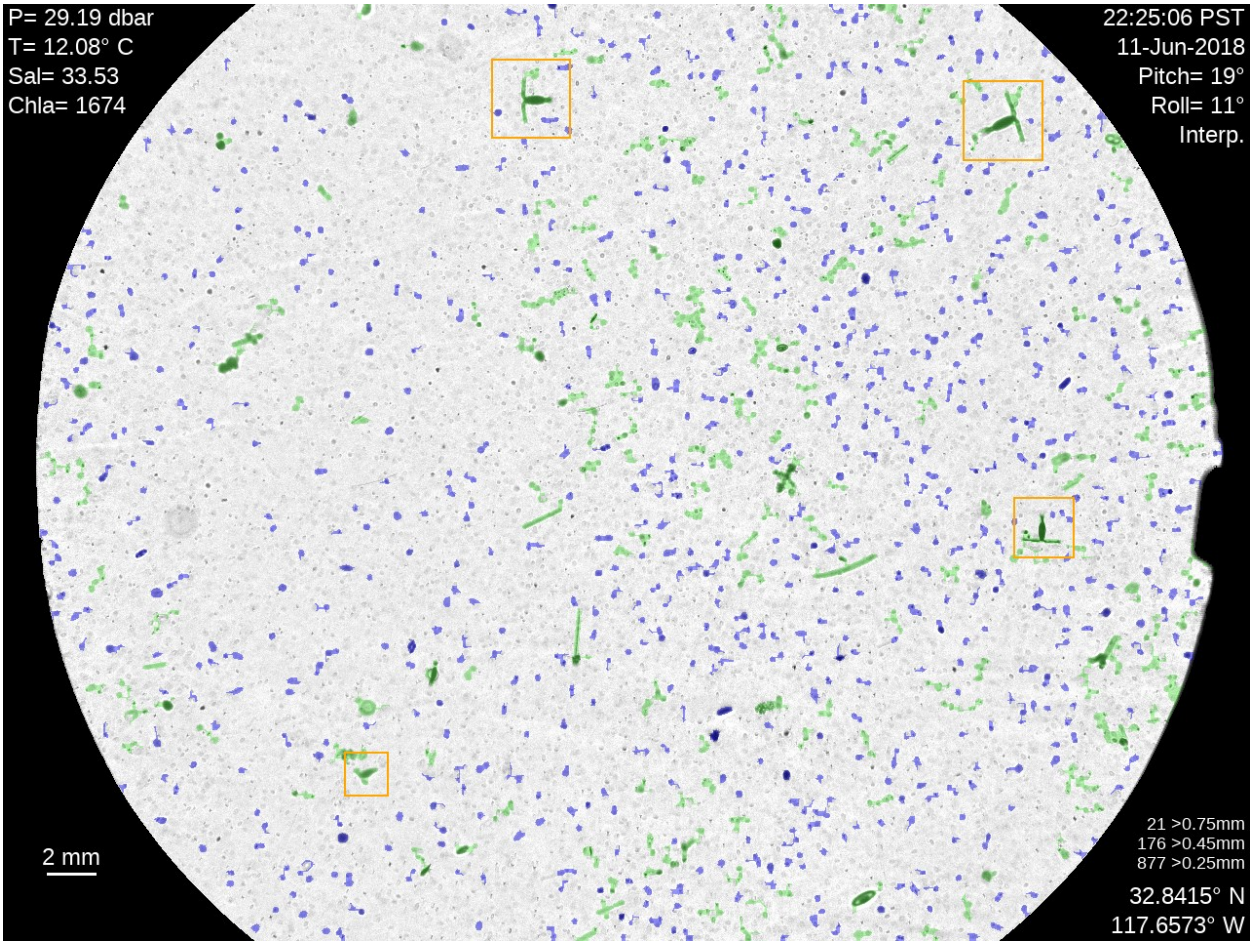


Figure 3.18: The full frame image from figure 3.10 after being processed by our segmentation algorithm. Four distinguishable copepods are located in the orange boxes.

Most of the time, the algorithm performs as desired, as in figure 3.16 (top), but transparent structures and images with numerous occlusions cause challenges.

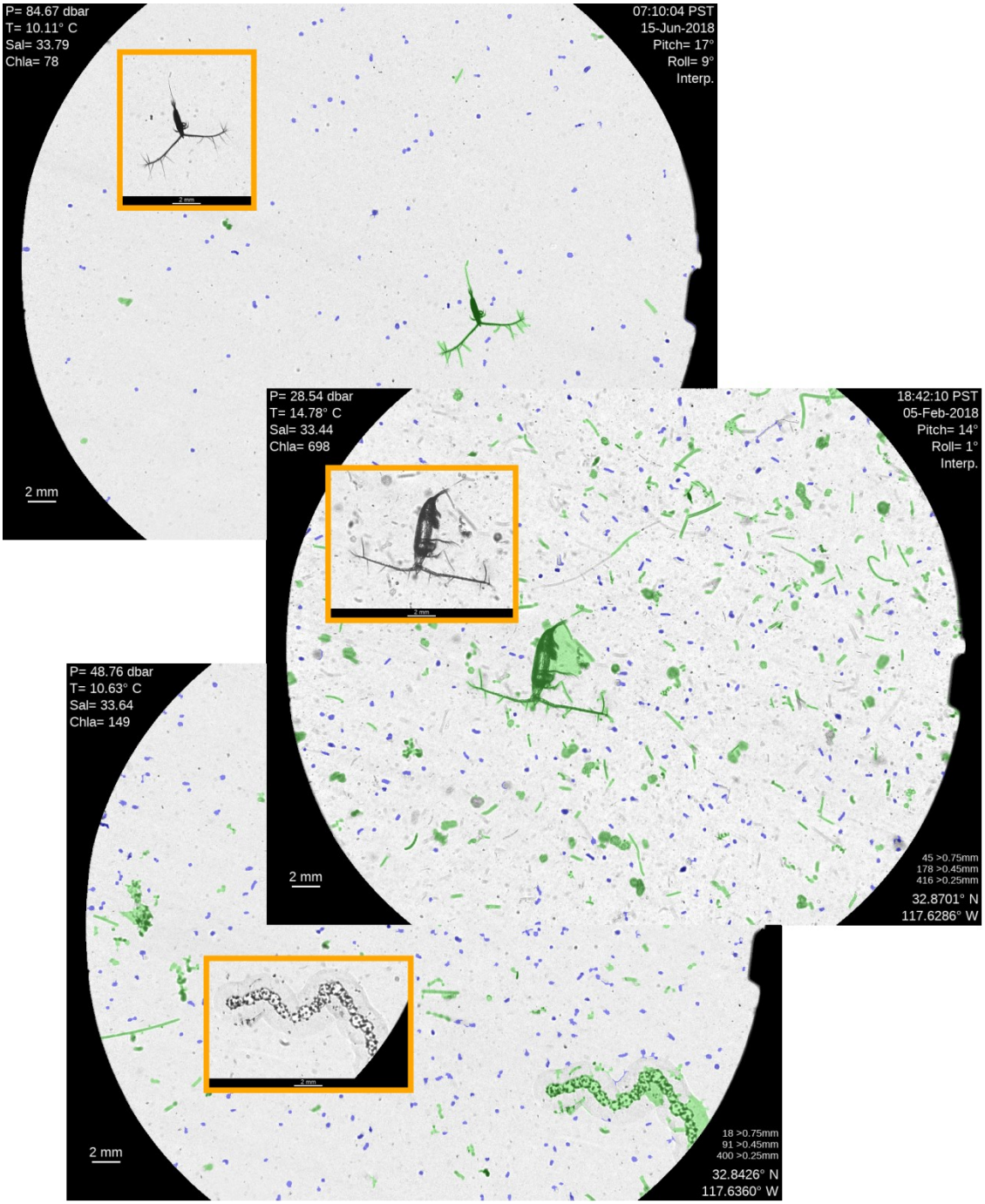


Figure 3.19: Three *Zooglider* frames. A typical frame (top) with fewer objects and accurate segmentation including setae on the copepod antennae, captured by our algorithm's sensitivity. This sensitivity occasionally causes issues such as enclosing regions due to adjacent objects at higher densities(middle), and also is not sensitive enough to capture the most transparent structures of some organisms (bottom).

Having a very sensitive segmentation algorithm causes issues when there is a high number of diatoms or marine snow in the frame. Figure 3.17 shows a flat-fielded frame on the left, and an intermediate step in the Canny algorithm on the right, where the white pixels have been identified as edge candidates by our second, more sensitive pass. These pixels are only candidates to be identified as edge pixels, since this step is before the edge-linking.

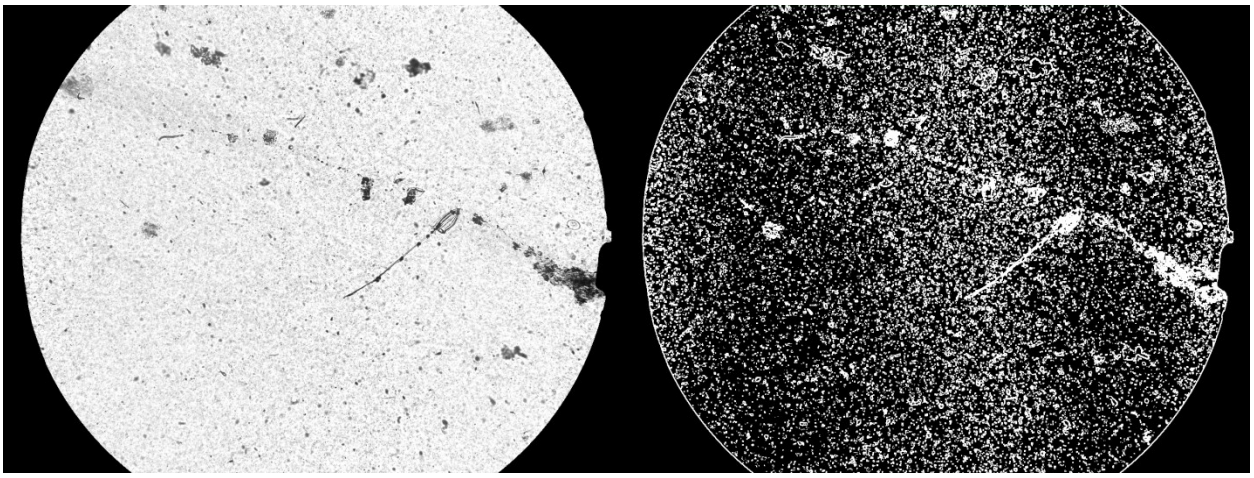


Figure 3.20: Original image with many particles (left) and candidate edge pixels identified in white (right).

Based on our edge-linking criteria, these candidate edge pixels are merged into a single large ROI, which is an unacceptable outcome in this case. I developed a heuristic algorithm to identify this situation. Even when there is a single large ROI, such as a chain of salps or a large siphonophore, the number of edge candidate pixels does not exceed roughly 3 or 4 % of the total number of pixels under consideration. However, fields of view with thousands of small particles have 5 to 10 % of the total number of pixels identified as edge candidates. So if the candidate pixels exceed 5% of the field of view, I adjust the thresholds used by the Canny algorithm to be roughly half as sensitive, and identify candidate edge pixels that meet the new criteria (Fig. 3.18). If the number of candidate pixels still exceeds 5%, I use a tertiary setting and proceed regardless of the number of pixels returned (Fig 3.19).

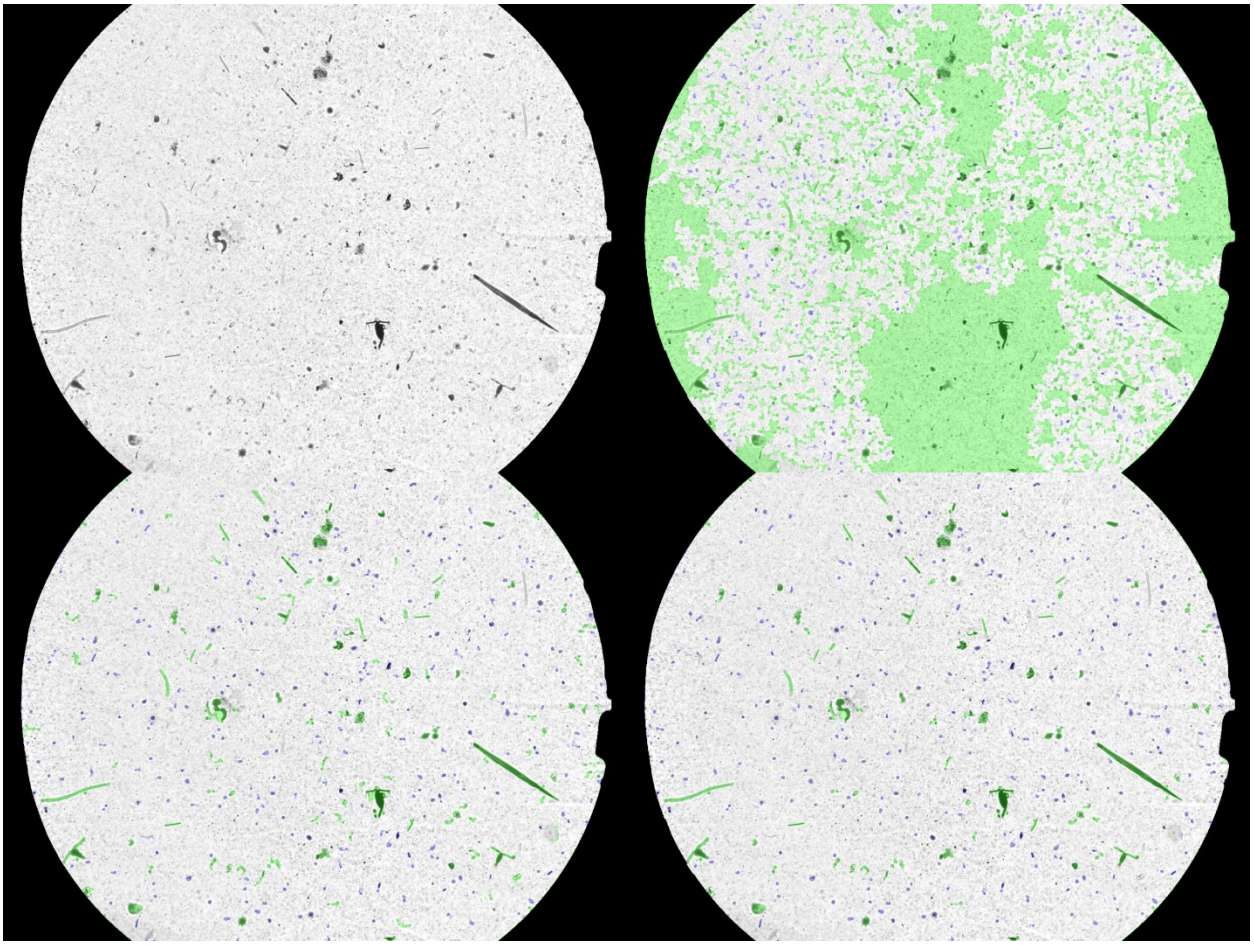


Figure 3.21: A full frame (upper left) and segmentations performed at our three sensitivity thresholds. ROIs were retained with the secondary sensitivity threshold (bottom left).

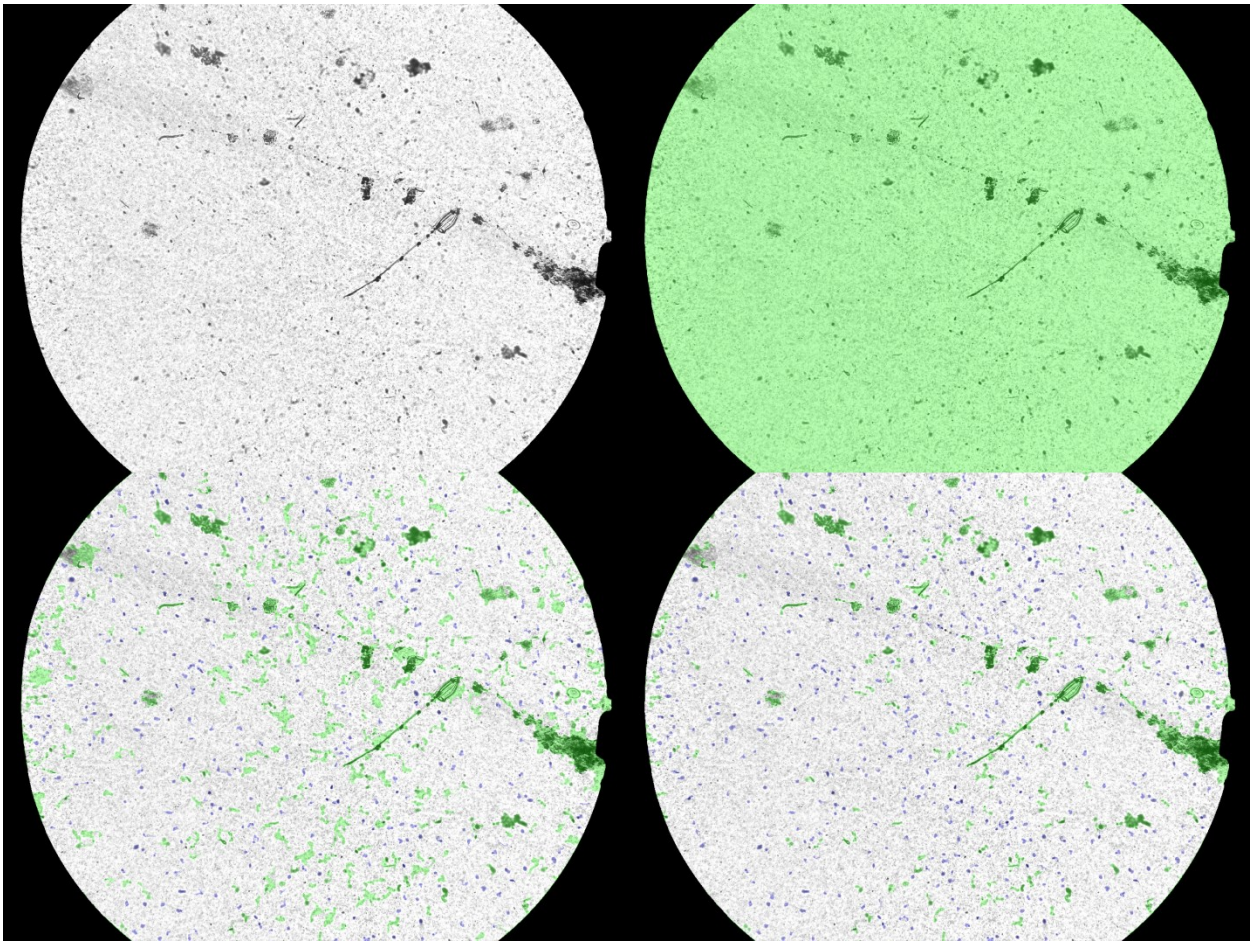


Figure 3.22: A full frame (upper left) and segmentations performed at our three sensitivity thresholds. ROIs were retained with the tertiary sensitivity threshold (bottom right).

Once our ROIs are properly segmented, I measure properties of the segmentation perimeter to be retained and embedded as XMP metadata.

3.5 Summary

I presented a complete workflow for processing images of plankton *in situ* acquired by a novel *Zooglider*. I adjust images to have a more uniform appearance with a flat-fielding technique inspired by the correction of images in astronomy. I introduced a novel segmentation

algorithm that is intended to capture delicate and nearly transparent zooplankton structures. This algorithm includes two different passes of a Canny edge detector with secondary and tertiary criteria to be used in the presence of dense aggregations of particles. After segmentation, properties of ROIs are calculated and embedded in the image in XMP format.

Deep Learning is becoming prevalent, and Convolutional Neural Networks and their predecessors can accurately classify objects in images without performing segmentation as a separate action (LeCun et al. 2015). Yet segmentation algorithms are still necessary for plankton images because in addition to the classification, biologists use quantitative morphometric aspects of the image (e.g. ROI length, area) extensively when conducting investigations using the images. Segmentation using Deep Learning is an area of active research that could address some the limitations of current approaches that rely on heuristics. CNNs have been used to segment by densely predicting a label for each pixel in the image (Long et al. 2015), and combining dense predictions with explicit human annotations of segmentation boundaries to improve generalization (Xie and Tu 2015). Future work on segmentation without deep learning has value for ensembles and embedded systems. Ensemble approaches combining multiple segmentation strategies have been shown to work for plankton (Blaschko et al. 2005; Sosik and Olson 2007; Hirata et al. 2016). As automated and high volume imaging systems acquire more plankton images, there is also a need for segmentation algorithms that achieve results at a high enough speed and low enough computational demands to allow for applications such as embedded hardware or autonomous vehicles (Ohman et al. 2018). Therefore plankton image segmentation should continue to be an open area of research.

3.6 Acknowledgements

The Gordon and Betty Moore Foundation funded the development of the *Zooglider* that collected these images, and their subsequent analysis. The Scripps Institution of Oceanography's Instrument Development Group performed the engineering and fieldwork needed to implement the *Zooglider* concept. Mark Ohman and Ben Whitmore assisted with the numerous hours of rigorous but subjective analysis required. Parts of the Methods section of this chapter were submitted verbatim as "Supplemental Information" within the publication: Ohman, Mark D.; Davis, Russ E.; Sherman, Jeffrey T.; Grindley, Kyle R.; Whitmore, Benjamin M.; Nickels, Catherine F.; Ellen, Jeffrey S. "Zooglider: an autonomous vehicle for optical and acoustic sensing of zooplankton." The dissertation author was the sole investigator and author of the reproduced portion of the material.

3.7 References

Adobe Systems Incorporated. 2001. The XMP Toolkit, Version 2.8, September 14, 2001. Retrieved from http://xml.coverpages.org/XMP-MetadataToolkit.pdf

Bradski, G. and Kaehler, A. 2000. The OpenCV Library. Dr. Dobb's. The World of Software Development: 1 Nov. 2000

Blaschko, M. B., Holness, G., Mattar, M. A., Lisin, D., Utgoff, P. E., Hanson, A. R., Schultz, H., Riseman, E. M., Sieracki, M. E., Balch, W. M., and Tupper, B. 2005. Automatic in situ identification of plankton. Proceedings of Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05). 1: 79-86. IEEE.

Canny, John. 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence. 6: 679-698, doi:10.1109/tpami.1986.4767851

Clausen, S., Greiner, K., Andersen, O., Lie, K. A., Schulerud, H., and Kavli, T. 2007. Automatic segmentation of overlapping fish using shape priors. Proceedings of Scandinavian conference on Image analysis. 11-20. Springer, Berlin, Heidelberg.

Davis, R. E., Ohman, M. D., Rudnick, D. L., Sherman, J. T., and Hodges, B. 2008. Glider surveillance of physics and biology in the southern California Current System. Limnology and Oceanography. 53: 2151-2168.

European Space Agency, European Southern Observatory, Centre for Advanced Studies, Research and Development in Sardinia. 2017. Python-XMP-Toolkit. (Version 2.0.2, 2017). Retrieved from: https://github.com/python-xmp-toolkit/python-xmp-toolkit

Faillettaz, R., Picheral, M., Luo, J. Y., Guigand, C., Cowen, R. K., and Irisson, J. O. 2016. Imperfect automatic image classification successfully describes plankton distribution patterns. Methods in Oceanography, 15, 60-77. doi: 10.1016/j.mio.2016.04.003

Gorsky, G., Ohman, M. D., Picheral, M., Gasparini, S., Stemmann, L., Romagnan, J. B., Cawood, A., Pesant, S., García-Comas, C., and Prejger, F. 2010. Digital zooplankton image analysis using the ZooScan integrated system. Journal of Plankton Research 32. 3: 285-303. doi: 10.1093/plankt/fbp124

Gonzalez, R. C., & Woods, R. E. 2007. Digital image processing, 2nd ed. Pearson Prentice Hall, Upper Saddle River, NJ.

Grosjean, P., Picheral, M., Warembourg, C., and Gorsky, G. 2004. Enumeration, measurement, and identification of net zooplankton samples using the ZOOSCAN digital imaging system. ICES Journal of Marine Science 61.4:518-525. doi: 10.1016/j.icesjms.2004.03.012

He, K., Sun, J., and Tang, X. 2011. Single image haze removal using dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence 33. 12:2341-2353. doi: 10.1109/tpami.2010.168

Hirata, N. S., Fernandez, M. A., and Lopes, R. M. 2016. Plankton image classification based on multiple segmentations. Computer Vision for Analysis of Underwater Imagery (CVAUI), 2016 ICPR 2nd Workshop on. 55-60. IEEE. doi: 10.1109/CVAUI.2016.022

International Organization for Standardization (ISO). 2012. Graphic technology -- Extensible metadata platform (XMP) specification -- Part 1: Data model, serialization and core properties. (ISO 16684-1:2012).

Leach, R. W., Schild, R. E., Gursky, H., Madejski, G. M., Schwartz, D. A., and Weekes, T. C. 1980. Description, performance, and calibration of a charge-coupled-device camera. Publications of the Astronomical Society of the Pacific. 92:233-245. doi: 10.1086/130654

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. Nature 521, no. 7553: 436-444. doi:10.1038/nature14539

Long, J., Shelhamer, E., & Darrell, T. 2015. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition . 3431-3440. doi: 10.1109/CVPR.2015.7298965

Luo, Q., Gao, Y., Luo, J., Chen, C., Liang, J., & Yang, C. 2011. Automatic identification of diatoms with circular shape using texture analysis. Journal of Software 6. 3:428-435 doi: 10.4304/jsw.6.3.428-435

Meijering, E. 2012. Cell segmentation: 50 years down the road. IEEE Signal Processing Magazine 29. 5:140-145. doi: 10.1109/MSP.2012.2204190

Oliphant, T. E. 2007. Python for scientific computing. Computing in Science and Engineering 9: 10-20. doi: 10.1109/mcse.2007.58

Sherman, J., Davis, R. E., Owens, W. B., and Valdes, J. 2002. The autonomous underwater glider Spray. IEEE Journal of Oceanic Engineering. 26: 437-446. doi: 10.1109/48.972076

Sosik, H. M., & Olson, R. J. 2007. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Limnology and Oceanography: Methods, 5(6), 204-216.

Tesic, J. 2005. Metadata practices for consumer photos. IEEE MultiMedia 12. 3:86-92.

Van Der Walt, S. and others 2014. scikit-image: image processing in Python. Peerj 2:e453 doi 10.7717/peerj.453

Xie, S., and Tu, Z. 2015. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision. 1395-1403. doi: 10.1109/iccv.2015.164

Zheng, H., Zhao, H., Sun, X., Gao, H., and Ji, G. 2014. Automatic setae segmentation from Chaetoceros microscopic images. Microscopy research and technique, 77(9), 684-690. doi: 10.1002/jemt.22389

CHAPTER 4 Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques

Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques

Jeffrey Ellen, Hongyu Li
Department of Computer Science and Engineering
University of California, San Diego
La Jolla, CA 92093-0404
Email: jellen@ucsd.edu, holi@ucsd.edu

Mark D. Ohman
Scripps Institution of Oceanography
University of California, San Diego
La Jolla, CA 92093-0218
Email: mohman@ucsd.edu

***Abstract*— This paper improves on the accuracy of other published machine learning results for quantifying plankton samples. The contributions of this work are: (1) Clarifying the number of expertly labeled images required for machine learning results. (2) Providing guidance as to what algorithms provide the best performance, and how to tune them. (3) Leveraging an ensemble of models to achieve recall rates beyond any single algorithm. (4) Investigating the applicability of abstaining. (5) Using size fractionation to learn more efficiently. (6) Analysis of efficacy of simple geometric features for plankton identification.**

Keywords—machine learning; image analysis; zooplankton; Zooscan

I. INTRODUCTION

Quantifying plankton is important, requires a high level of taxonomic skill, and is expensive. Automation of plankton sample enumeration can enable higher throughput, more efficient processing, and improved scientific understanding., Specific applications of interest include understanding plankton spatial distributions, parameterizing oceanographic models, and investigations of population ecology. In this paper, we address methods to improve automatic classification of images from preserved plankton samples.

II. MACHINE LEARNING EXPERIMENTATION

A. Data Set Description

The California Cooperative Oceanic Fisheries Investigations (CalCOFI) is a field program that has been sampling the ocean, including plankton, since 1949 [1]. The CalCOFI plankton samples are collected at sea according to a standardized bongo net protocol [2] and immediately preserved. Substantial portions of the preserved CalCOFI samples recently collected in conjunction with the California Current Ecosystem Long Term Ecological Research site have been scanned with ZooScan [3]. The resulting grayscale images are very accurately controlled in terms of contrast, noise, and other variations (Fig. 1).

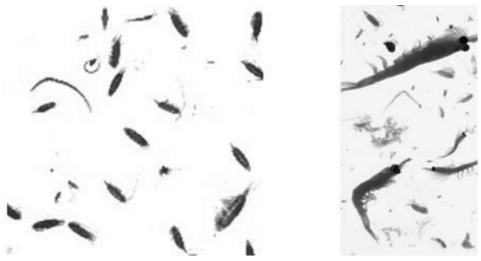


Fig. 1. Parts of two different scanned images of plankton samples, illustrating variety of ROI shapes and sizes, as well as the relative ease of ROI segmentation.

Because of the controlled conditions, the Regions of Interest (ROIs) are readily segmented from the larger image, which contains 1,000-2,000 ROIs. Figure 2 (upper row) shows two examples of animals scanned by the ZooScan, with photographs of similar animals for comparison (lower row).

Fig. 2. Left: *Nyctiphanes simplex*, a euphausiid common off the California coast. Right: a chaetognath. Both are relatively large for CalCOFI plankton: the scale bar in all 4 images is 1mm. Both have substructures and opacity differences that are preserved in ZooScan images. Photos from SIO Pelagic Invertebrates Collection [4].

Not all of the classes are so easily recognized. Figure 3 shows three more categories of varying size, shape, and

Contribution from the National Science Foundation–supported California Current Ecosystem Long Term Ecological Research site. Plankton sample analysis supported by NSF grants to M.D. Ohman, and by the SIO Pelagic Invertebrates Collection.

contrast. Some plankters are inherently more fragile, with gelatinous parts or thin appendages that are frequently damaged by net collection. Transparency is more variable in preserved samples than in live ones. Less rigid animals also have a less consistent posture and orientation. Some classes have a wide variety of sizes, and the smallest plankters have a lack of detail due to the limits of the scanning resolution. These identifications can be challenging for a human.

These challenges are not unique to plankton imaging, but are different from mainstream image processing/classification tasks, such as the ImageNet competition.

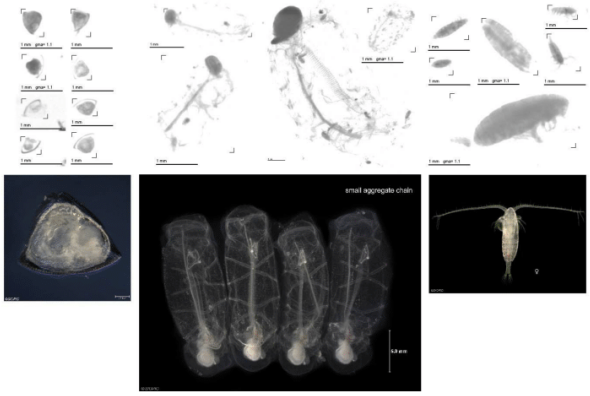


Fig. 3. Left: bryozoan larvae, of relatively uniform size and shape. Middle: a small chain of the salp *Pegea socia*, a gelatenous pelagic tunicate whose ZooScan samples exhibit some variation of scale and irregular shapes. Right: copepods, which have an even larger size range and variation within the image, despite being relatively rigid compared to the gelatinous tunicates. All ZooScan images have a scale bar of 1mm, the bryozoan larvae photo has a scale bar of 0.2mm, and the tunicate photo has a scale bar of 5mm. Photos from SIO Pelagic Invertebrates Collection [4].

The data used in this paper consists of 725,516 individual ROIs taken from samples collected during 46 different ocean transects from July 2005 to July 2012. The transects are line 80 and line 90 in the CalCOFI grid (Fig. 4), and samples are taken quarterly. Most ROIs contain a single entity, and are labeled with one of 24 categories of organisms such as 'siphonophore' or 'calanoid copepod'. There are also categories for 'detritus', 'multiples', and 'others'. The splits are functional rather than biological or genetic. A complete list is provided in Appendix A.

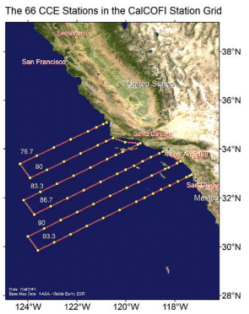


Fig. 4. The CalCOFI grid has been sampled for 67 years. Samples from line 80 and line 90, from July 2005 to July 2012 were used in this data set.

The data described in this paper will shortly be made available through the CalCOFI DataZoo website (http://oceaninformatics.ucsd.edu/datazoo/)

B. Machine Learning Features

The ZooScan software can generate low level geometric features and grayscale features for the purposes of biological object identification [5]. ZooScan has been used to count the abundance of both zooplankton [3] as well as fish eggs [6].

We used a subset of 51 of these features in the experiments described in this paper. Each feature is computed on the pixels within the ROI only, not the bounding rectangle. Features used include 19 size/shape measurements, such as area, circularity, major/minor axis length, feret diameter, and some ratios of these values. Also included are 17 grayscale distribution measurements, such as the min, max, mean, standard deviation, quartiles, skew, and cumulative histogram slope. The remaining features are positional, such as the centroid location, or more derived, such as the fractal dimension or the symmetry. Complete descriptions of the referenced features are on the ZooScan website at http://www.zooscan.obs-vlfr.fr/, and the complete list of features and some illustrations are provided in Appendix B.

C. Experimental Procedure

We conducted a series of machine learning experiments. We carried out two different 8-way classification experiments in order to compare our efforts with contemporary results. We also conducted two different 16-way classification experiments in order to examine the tradeoff between complexity and performance.

For each of the 8-way experiments, we varied the data set size from 500 to 76,800 ROIs. For the 16-way experiments, we used data set sizes from 6,000 to 725,516 ROIs. We formed balanced data sets (equal types of each image class) to facilitate experimental design in addition to interpretation of results. For example, when evaluating the impact of adding classes, or which class is the most difficult, it is important that those be held constant. Also, balanced classes allow for more simple summary statistics, such as recall, to be used to measure performance.

We built our classifiers using Python's Scikit-learn [7]. We evaluated 2 types of support vector machines (SVM), 3 types of random forests (RF) including an extra trees ensemble (XTR) and a gradient boosted random forest classifier (GBC), stochastic gradient descent with two different types of loss functions (SGD), 2 types of k-nearest-neighbor algorithms (standard (kNN) and nearest neighbor Ball Tree (nnBT)), and neural nets (implemented as a multi-layer perceptrons with a single hidden layer - MLP). For each algorithm mentioned, we experimented extensively with hyperparameters, including hundreds of combinations for SVMs to thousands of combinations for RFs as described in Section III.B and

Appendix C. We consistently used an 80/20 split for training data vs testing data, with the exact same ROIs made available to each algorithm for training.

III. Experimental Results

Our experiments were designed to provide insight into six important facets of this machine learning problem: how many data to use, which algorithm, whether ensembling helps, whether abstaining helps, whether size fractioning the data helps, and the effectiveness/efficiency of using geometric features. The results presented are a representative sample, not an average. Each experiment was repeated multiple times.

A. Determining Data Set Size Requirements

Since machine learning algorithms can be computationally expensive, and obtaining training data can be expensive, we want to quantify the 'rate of return' on hand-labeled training data. To investigate, we trained suites of classifiers with different numbers of examples per class. Each point in Fig 5 represents an independently trained 8-way classifier. 500 examples per class seemingly provides asymptotic performance. However, we continued to conduct experiments and found continued improvement as training set size increased to ~4,000 (Fig. 6).

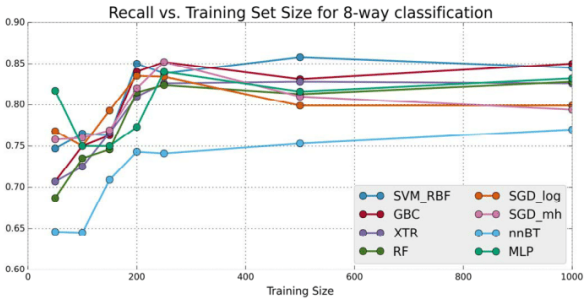


Fig. 5. Performance grouped by algorithm, shown with respect to training set size. Small data set sizes are noisy. The increase in performance apparently levels off after 500 examples per class. SVM_RBF is an SVM with a radial basis function for a kernel. SGD_log is stochastic gradient descent with log loss (logistic regression), and SGD_mh is SGD with 'modified huber' as a loss function. GBC,XTR,RF,nnBT, and MLP correspond with the descriptions in Section II.B.

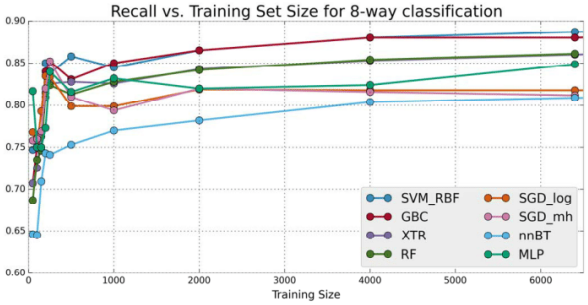


Fig. 6. Performance grouped by algorithm including larger training set sizes. Performance levels off after 4,000 examples per category for most algorithms.

For the numeric results summarized in Table I, the following holds regardless of algorithm: initially, doubling the training set size provides a 3-5% increase in performance; this rate decreases to a 1-2% improvement at larger training set sizes. The number of available expert-annotated ROIs in the 8 classes limited us to 7,680 training examples.

TABLE I. Recall Results for 8-way Classification Task

Training Size	SVM RBF	GBC	RF	SGD	nnBT	MLP
50	0.747	0.707	0.687	0.768	0.646	0.817
100	0.765	0.750	0.735	0.750	0.645	0.750
150	0.763	0.763	0.746	0.793	0.709	0.750
200	0.850	0.840	0.815	0.835	0.743	0.773
250	0.838	0.852	0.824	0.834	0.741	0.840
500	0.858	0.831	0.813	0.799	0.753	0.816
1000	0.845	0.850	0.828	0.799	0.770	0.832
2000	0.865	0.865	0.842	0.819	0.782	0.820
4000	0.880	0.880	0.854	0.818	0.804	0.824
6400	0.887	0.880	0.861	0.818	0.809	0.849
7680	0.888	0.883	0.864	0.818	0.811	

Figure 7 illustrates performance with respect to individual classes for the 8-way classification problem. Only four algorithms are shown: the results were consistent across all classifiers. In small sample sizes, the data are noisy, but as the training set size grows sufficiently large, calanoid copepods were consistently the most difficult category to classify, and eggs were the easiest. Not surprisingly, the more difficult classes also had the largest performance gain from additional training examples.

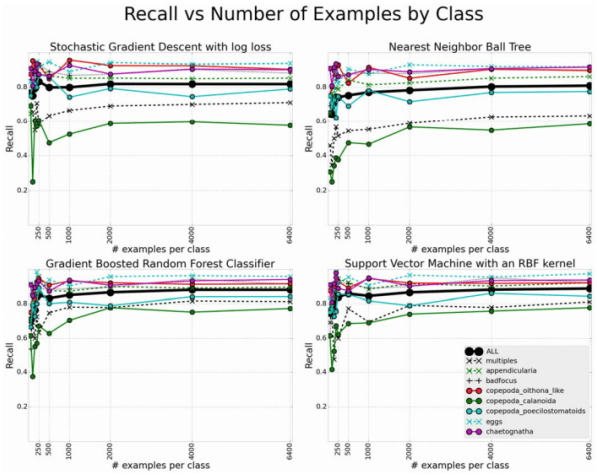


Fig. 7. Difficult classes tended to remain difficult, regardless of algorithm or data set size.

We consistently found SVM with an RBF kernel (SVM-RBF) and GBC to provide ~1-2% better accuracy than other methods.

Our overall results compare favorably with previous results. Our 8-way classification algorithm consisting of an ensemble of the GBC and SVM has an overall recall rate of 88.6%, which is 10 percentage points better than the best recall rate of 78% presented in Gorsky et al. [3]. In addition, the two best performing classes in Gorsky et al. are 'Bad Focus' and 'Fibers,' two inorganic classes.

Our results are also efficient for specific individual classes. For appendicularians we have a recall rate of 96.5% and a precision of 92.9%, as illustrated in Fig. 9, which is approximately 20-30 percentage points better than the performance in Forest et al. [8]. Our task is more difficult because we are attempting 8-way classification, where Forest et al. attempted 4-way classification. Their data set was much smaller, consisting of only 2,100 ROIs. For our training set of that size, 200 each for training and 50 for testing on each of 8 classes (2,000 total vignettes) we have an overall recall of 85%. For appendicularians specifically from that smaller data set we achieve a recall of 91.5% and a precision of 88.5%.

We achieve similar results to the phytoplankton classification task in [9]; Sosik and Olson had some simple classes for which 100% accuracy was achieved, and some ''difficult classes, such as detritus'' where only 68% accuracy was achieved. These percentages are on the order of our results. For example we also found detritus to be the most difficult category in our 16-way classifier, as shown in Figure 8. Sosik and Olson also found SVM with an RBF kernel to achieve the best results.

While minimal increases are obtained with larger sets, significant gains in recall are observed up through 4,000 examples. This value is a significantly different finding from Gorsky et al., who found that their performance plateaued at ~300 examples per class [3].

A more complicated, but closer to real world example is shown in Fig 8, which illustrates a confusion matrix for our best result for the 16-way classification problem. This classifier trained on 3,600 examples per class, which was the highest number available for all 16 classes, and achieved an overall recall rate of 0.813. When we trained a classifier on only 300 examples per class, as in [3], our best effort resulted in an SVM with an overall recall of 0.741, with similar types of errors as the confusion matrix shown.

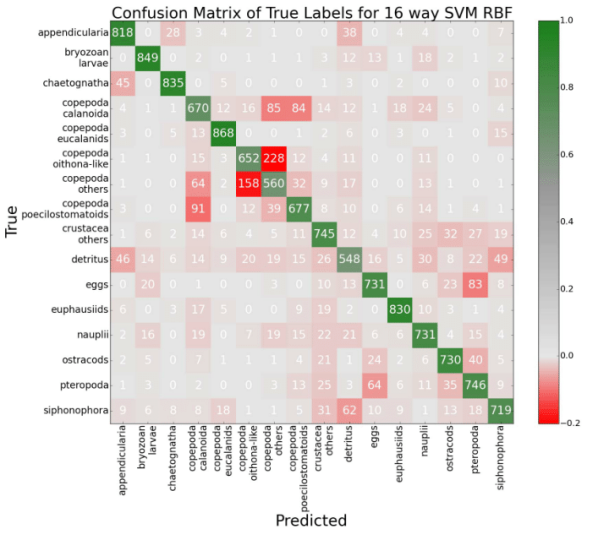


Fig. 8. Confusion matrix for our best results for the 16-way classification task. Depicted are the results for an SVM trained on 3,600 samples per class, and tested on 900 samples per class.

Ultimately, relative results are more important than absolute results, particularly because our ROIs are not a benchmark data set. For example, our categories of interest are often arbitrarily defined by the taxonomic resolution desired by a given lab. Note that in Fig 8, 862 of the errors are misclassifications of one type of copepod for another. Grouping all types of copepod into a single 'copepod' category would then raise performance by 0.075 to 0.888 on the 16-way task. Similarly, 'multiples' and 'bad focus' are two distinct classes in our data, but since they both refer to malformed ROIs, errors confusing these classes for each other should be considered less severe than all other errors.

Our relative results are competitive with other work. Our California Current sample data were previously processed with a RF classifier developed according to the results of [1]. The algorithm's recall is poor on all rare classes (often single digit accuracy), and in the teens for some common classes. Recall only exceeded 0.60 for a single class, detritus at 0.886. For a fair comparison of the SVM algorithm used in the present paper with a RF algorithm developed according to [1], classes were removed to create one 8-way classification task. Accordingly, the previously used RF implementation performed with 0.618 recall, compared to our recall of 0.887 in the present paper. For a 16-way classification task, the previous RF implementation had a performance of 0.580, compared to 0.813 for our implementation using 3,600 training items for each of the 16 classes. This 0.23-0.27 gain gives an idea of the improved performance, but slightly overestimates it because the real-world problem is harder than the treatment presented in this paper.

The 8 most prominent classes cover 85% and the 16 most prominent classes cover 92.9% of the data. So while the models presented in this paper were not trained with balanced data and not for our full 24-way classification problem, they

are useful as is, and could not perform worse than getting every single image from the rare classes incorrect. In the case of the 8-way classifier, this would yield an effective recall rate of 0.762 (an improvement of 0.221) and the 16-way classifier would yield an effective rate of 0.756 (an improvement of 0.215). Since the effective rate of the 8-way algorithm was better, it ultimately may be more effective to use a classifier with fewer classes and higher performance, which results in the need to completely sort difficult classes by hand.

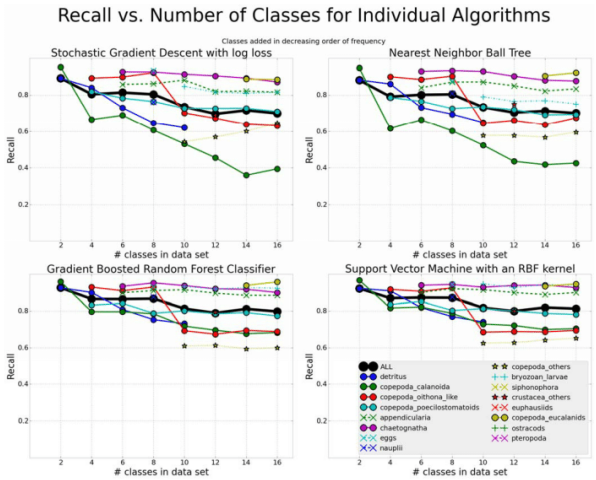


Fig. 9. An illustration of how increasing the number of classes affects recall rates. Four different algorithms are shown, and each algorithm has results for 8 different classification tasks presented. (2-way, 4-way, 6-way, 8-way, 10-way, 12-way, 14-way, and 16-way.)

Figure 9 shows how overall recall decreases when more classes are added. We trained a series of models where each algorithm had 3,600 training items, and additional classes were added to the training set in reverse order of overall abundance. Unsurprisingly, with each pair of additional classes performance decreased, although apparently more related to the difficulty of the added class than the overall number of classes. Note that improved performance on a more complex problem may be obtained by increasing the amount of training data.

B. Hyperparameter Tuning

The three algorithms with the best performance were the multi-layer perceptron (MLP), gradient boosted random forest (GBC), and support vector machine (SVM). In general, for the MLP, we found learning rate to be the most important single parameter as suggested by Bengio [11], and found that a large number of nodes in the hidden layer was not required. For the GBC, we ended up with shallow trees, a large number of samples per leaf, and moderate regularization. For the SVM, we found the regularization parameter needed to be increased on larger data sets, while the free parameter (gamma) was relatively constant.

For all experiments, cross validation consisted of a minimum of 5 folds during the hyperparameter search, but the final model was refit with the entire dataset. More information about hyperparameter tuning is provided in Appendix C.

C. Using an Ensemble to improve accuracy

Combining the results of two best performing classifiers consistently resulted in up to a 0.6% gain in recall, at no additional expense. Example results for one of our 8-way classification problems are shown in Table II. Our ensemble was done by averaging; each algorithm with the ability to returned a probability, rather than a classification. The estimated probabilities from all algorithms were then averaged pairwise, and evaluated as though they were the results of a single classifier.

Not surprisingly, the combination of the two best single-performing algorithms resulted in the strongest performance. Larger combinations of three or more algorithms sometimes achieved better results, but not as consistently as combining the SVM and the GBC. Overall, averaging provided improved results than either individual algorithms 33% of the time. Also, while not a strictly an improvement, note that SVMs helped every single other classifier exceed the recall that the other algorithm achieved independently.

TABLE II. RESULTS FOR AVERAGING 8-WAY PREDICTIONS (4000 TRAINING ELEMENTS/CLASS)

Algorithm(s) – Trained on 4,000 each	**Recall**	**Avg. Yields Improvement**
GBC and SVM_RBF	0.8866	Y
SVM_RBF and XTR	0.8855	Y
RF and SVM_RBF	0.8835	Y
SVM_RBF	0.8805	N/A
GBC	0.8799	N/A
GBC and RF	0.8783	N
GBC and XTR	0.8780	N
SGD_log and SVM_RBF	0.8758	N
GBC and SGD_mh	0.8755	N
GBC and SGD_log	0.8751	N
GBC and nnBT	0.8744	N
nnBT and SVM_RBF	0.8733	N
SGD_mh and SVM_RBF	0.8733	N
RF and XTR	0.8546	Y
RF	0.8536	N/A
XTR	0.8528	N/A
SGD_mh and XTR	0.8450	N
RF and SGD_mh	0.8441	N
RF and SGD_log	0.8439	N
SGD_log and XTR	0.8429	N
nnBT and RF	0.8401	N
nnBT and XTR	0.8375	N
nnBT and SGD_log	0.8278	Y
nnBT and SGD_mh	0.8276	Y
SGD_log and SGD_mh	0.8185	Y
SGD_log	0.8184	N/A
SGD_mh	0.8155	N/A
nnBT	0.8036	N/A

In addition, the improvement is primarily in the most difficult class, Chaetognatha, and improvement is consistent across numerous examples. The full impact of the best ensemble is shown in Fig 10.

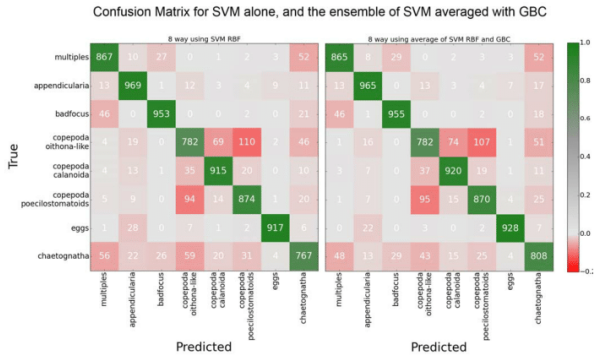


Fig. 10. The two confusion matricies shown illustrate the modest gains to be had by averaging the two best performing algorithms. The improvement is slight, 49 net additional correct classifications out of 8000, for an improvement of 0.61%

D. Improving Abundance Estimation Through Abstentions

By changing our classifier to output probabilities rather than labels we can allow abstentions. We allow for abstentions for ROIs with low confidence by ignoring guesses below a particular probability threshold. This technique eliminates false positives at the expense of some images remaining unlabeled. Therefore, this approach may be useful in circumstances where there is a high penalty for a false positive, but little penalty for a false negative. Table III provides an example.

TABLE III. ALLOWING ABSTENTIONS IN THE 8-WAY CLASSIFICATION MODEL (AVERAGE OF SVM AND GBC - 4000 TRAINING ELEMENTS/CLASS)

Confidence Threshold	% Labeled	Recall	% Labeled Correctly
0.3	0.9995	**0.8868**	0.8864
0.4	0.9941	**0.8904**	0.8851
0.5	0.9746	**0.8987**	0.8759
0.6	0.9279	**0.9196**	0.8533
0.7	0.8674	**0.9412**	0.8164
0.8	0.7943	**0.9600**	0.7625
0.9	0.6755	**0.9782**	0.6607
0.95	0.5445	**0.9867**	0.5372
0.99	0.2375	**0.9953**	0.2364

For example, setting the confidence threshold at 0.95 results in 0.9867 recall. This threshold results in the correct labeling of 4,299 of the original 8,000 ROIs and only 57 incorrectly labeled ROIs, as shown in Fig 11.

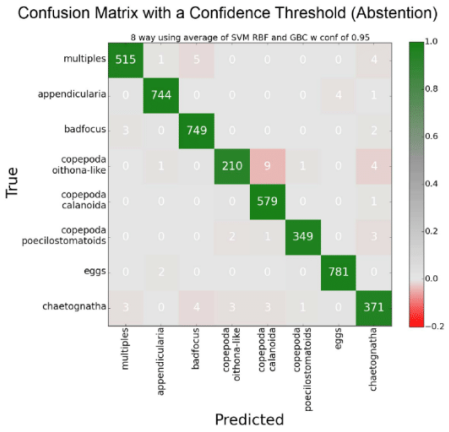


Fig. 11. Confusion matrix when the classifier is allowed to abstain from labeling images andonly classifying when probability exceeds 0.95. This approach greatly reduces the number of false positives compared to other classifiers.

If only some labeled images are required, and not a complete census, this technique may be useful for quickly labeling some images without incurring the expense of extensive manual resorting. We did not investigate whether the ratio of true positives to false negatives was more stable when allowing abstentions, but if so, an estimate of total abundance could be achieved through simply scaling these results.

E. Efficiency Through Size Fractionation

More examples in a training set improve recall but algorithm training times grow non-linearly with respect to the number of examples SVMs, for example, usually have a runtime of $O(n^2)$. According to Bottou [10], "(Runtime) grows at least like n^2 when C is small and n^3 when C gets large."

We show that size fractioning the data set can combat this penalty, and potentially allows accuracy beyond what the hardware could not otherwise achieve.

We performed a series of experiments creating specialist classifiers on different sizes of ROI. For example, we split the ROIs into quartiles by pixel area, and trained four independent classifiers. One model was trained on the smallest quartile of the ROIs, a second, independent model was trained on the next quartile larger ROIs, etc. The effect on recall was negligible. But most importantly, training four smaller classifiers is markedly faster than training one larger classifier. Completing the initial, coarse-grid search with cross validation on the single large model took 48 hours on our hardware. Training 16 specialist classifiers on size fractions of the ROIs, each over the same grid search, took 2 minutes per classifier, for a 100x speedup over the single classifier.

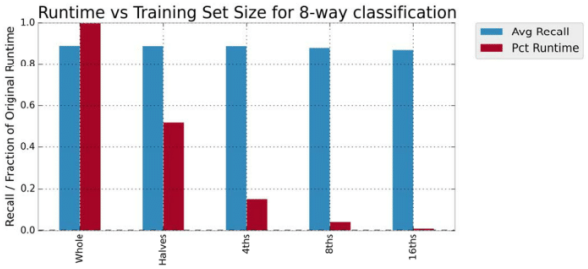


Fig. 12. Size fractioning the data results in significant time savings. Halving the data by size had minimal or no impact on recall, but drastically reduced execution time. Fraction of original runtime includes all classifiers from the group.

In one of our 8-way classification problems, size fractionating did not result in an overall gain, as none of the ensemble specialists is better than the baseline classifier on the whole data set. However, training multiple size-fractionated classifiers provides slightly better results than training a single, smaller classifier on all of the data, as shown in Fig.13.

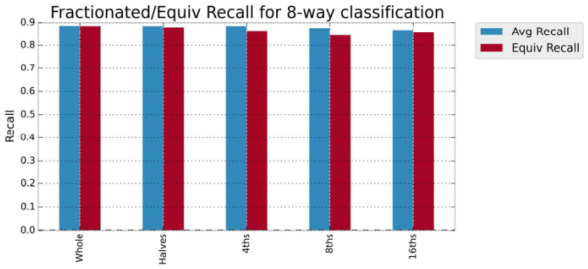


Fig. 13. The bar on the left is the baseline, a single classifier. The average of each ensemble (blue bars) is slightly higher than the recall of a single classifier of equivalent size (red bars).

To achieve maximum accuracy for our ROIs, creating a single large classifier slightly outperforms ensembles of size-fractionated models. However, size fractionating the data greatly reduces the training time, and in cases where the machine resources are limiting, creating multiple size-fractionated models will improve accuracy beyond creating a single classifier containing a selection of all of the data, as shown in the table below. In Table IV, if 8,000 examples per class are available then training a single 8-way SVM with 8,000 training examples per class yielded the best results. However, if the largest model able to be trained given hardware constraints is 1,000 examples per class, the results would be 0.02-0.03 better by training 8 models on various size fractions than trying to train a single model on data encompassing all size ranges. Size is a deterministic, objective criterion that does not require any human prescreening, and has a basis in the problem space (animals generally fall within certain size clusters per species) and therefore makes a reasonable separation criterion.

TABLE IV. SIZE FRACTIONATED RECALL VS. EQUIVALENT RECALL

Split	Recall			
	8-way classification Task		16-way classification Task	
	Split Avg	*Equiv Recall*	*Split Avg*	*Equiv Recall*
Whole	0.8869	N/A	0.8131	N/A
Halves	0.8856	0.8805	0.7948	0.775
4ths	0.8857	0.8650	0.7902	-
8ths	0.8770	0.8445	0.7843	-
16ths	0.8684	0.8580	0.7654	0.74

F. Efficiency and Feature Set Size

We use a set of only 51 features, and our algorithms learn on the order of thousands of parameters (weights) depending on the algorithm. 'Deep Learning', which usually means convolutional neural networks, has performed well in many image competitions and publications. In deep learning architectures, hundreds of millions of weights are learned, for example 133M to 144M weights were learned for 224x224 pixel images in [12].

Even with our much simpler features, our grid search and cross validation for some individual models took multiple days to complete on a system with 50 available CPU cores. While a few days may be an acceptable wait, the first model will not be the one ultimately used, and many will need to be trained before results are reliable. Deep Learning algorithms can take advantage of the high level of parallelism to utilize GPUs, but the computational cost of deep learning algorithms is still orders of magnitude above our method. While the cost of computing is cheap and classification accuracy frequently is maximized above processing costs, the number of parameters required for such networks is substantial.

To alleviate some of the computational expense of deep learning approaches, many researchers are using networks where the first set of filters has been copied from, or 'pre-trained' on a different data set, such as ImageNet data [13] (UCSD students, SciPy attendees, US Navy Scientists, personal communications, 2014-2015). Even copying filters still requires training millions of parameters in the later stages of the network.

However, in cases such as embedded systems, extremely large data sets, or initial investigations, a smaller set of features, such as the ones presented here is sufficient.

IV. CONCLUSION

In order to more effectively quantify our plankton samples, we executed a series of experiments to determine how to improve classification accuracy.

Carefully tuned support vector machines slightly outperformed gradient boosted random forest and multi-layer perceptron neural networks. Regardless of algorithm, performance increased until at least 4,000 training examples per class, although performance continued to increase with more data. Data set size impacted performance and had a bigger effect than choice of algorithm. In Table I, the first few rows of data with smaller training sets have a 10 percentage

point range between the lowest and highest performing algorithm. However, the columns show that training set size has an even bigger impact; the gain between an algorithm with less training data and the same algorithm with more data boosts recall by 15 percentage points or more. Correct hyperparameter tuning is also an important consideration. We share our methodology in Appendix A.

Our results are consistent across classes and algorithms. SVMs almost always performed best. Most hyperparameter searches ended up in the same narrow ranges across experiments. We found that creating an ensemble of our two best performing classifiers also increases performance at no additional computation or training cost.

Geometric features are inherently efficient compared to other approaches, and size fractioning the ROIs increases run time efficiency further. Our best results improve upon our previous random forest implementation by 22 percentage points. We found that our simple geometric features can achieve a recall of 0.887 for our best ensemble.

V. Appendix

A. Classification Labels

The 24 classification labels present in our data are: ['detritus', 'copepoda_calanoida', 'copepoda_oithona_like', 'copepoda_poecilostomatoids', 'multiples', 'badfocus', 'appendicularia', 'chaetognatha', 'eggs', 'nauplii', 'copepoda_others', 'bryozoan_larvae', 'siphonophora', 'euphausiids', 'crustacea_others', 'copepoda_eucalanids', 'ostracods', 'pteropoda', 'doliolids', 'others', 'radiolarians', 'polychaete', 'bubbles', 'copepoda_harpacticoida']. This list is sorted in order of frequency of occurrence.

B. Machine Learning Features

The 51 features we used for learning, spelled as provided by ZooScan are ['Angle', 'Area', 'Area_exc', 'CDexc', 'CV', 'CentroidsD', 'Circ.', 'Circexc', 'Convarea', 'Convperim', 'Elongation', 'Feret', 'FeretAreaexc', 'Fractal', 'Height', 'Histcum1', 'Histcum2', 'Histcum3', 'IntDen', 'Kurt', 'Major', 'Max', 'Mean', 'MeanPos', 'Median', 'Min', 'Minor', 'Mode', 'Nb1', 'Nb2', 'Nb3', 'Perim.', 'PerimAreaexc', 'PerimFeret', 'PerimMaj', 'Range', 'SR', 'Skelarea', 'Skew', 'Slope', 'StdDev', 'Symetrieh', 'Symetriehc', 'Symetriev', 'Symetrievc', 'ThickR', 'Width', 'X', 'XM', 'Y', 'YM'] [ref. 1]

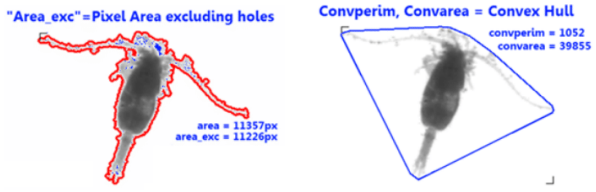


Fig. 14. Illustrations of how some of the feature values are calculated for actual ZooScan images.

C. Hyperparameter Optimization

For the multi-layer perceptron, a single hidden layer of 50 nodes often provided the best results. More nodes, even thousands, did not provide improved results. Learning rate was the most extensively searched hyperparameter, as it is reportedly the most important [11]. Initial searches covered many orders of magnitude. The final searches were fine-grained, with spacing of 2x (e.g. 0.05, 0.025, 0.0125). The optimal learning rate varied with training set size, and was most frequently 0.025 for smaller data sets with less than 500 examples per class, and decreased to 0.0025 for our largest data sets. The optimal L1 and L2 regularization hyperparameters were searched independently and consistently found to be 10^{-5} or 10^{-6}, with larger values found to be detrimental to performance.

For the Gradient Boosted Random Forest Classifier we evaluated four hyperparameters. For the maximum tree depth we tried values up to 25, but frequently a low value, such as 6, was optimal. We tried the odd-number values 3, 5, 7, 9 for the minimum samples per leaf, and the larger values, such as 7 or 9 provided the best performance. Maximum features were evaluated on deciles from 0-1, and intermediate values such as 0.3 performed best. Values of the number of estimators up to 2,500 were tried, and there was little discernable pattern.

For the support vector machine, the radial basis function with degree=3 was used for all reported results. For the regularization parameter, *C*, we experimented with various orders of magnitude from 1 to 100 million, but all results were obtained with values in the narrow range of 10,000, 100,000, or 1,000,000 with stronger regularization consistently providing better results on the larger datasets. For the free parameter, *gamma*, we again experimented with various orders of magnitude from 0.1 to very small, and found that except for very small sized data sets, gamma of 0.001 or 0.0001 was optimal.

References

[1] S. J. Bograd, D.A. Checkley, and Warren S. Wooster, "CalCOFI: A half century of physical, chemical, and biological research in the California Current System," Deep Sea Research Part II: Topical Studies in Oceanography, vol. 50, no. 14, pp. 2349-2353, 2003.

[2] M. D. Ohman, and P. E. Smith. "A comparison of zooplankton sampling methods in the CalCOFI time series." California Cooperative Oceanic Fisheries Investigations Report, pp. 153-158, 1995.

[3] G. Gorsky, M. D. Ohman, M. Picheral, S. Gasparini, L. Stemmann, J.-B. Romagnan, A. Cawood, S. Pesant, C. Garc ıa-Comas, and F. Prejger, "Digital zooplankton image analysis using the zooscan integrated system," Journal of Plankton Research, vol. 32, no. 3, pp. 285–303, 2010.

[4] L. Sala and M. D. Ohman, "Zooplankton of the San Diego Region," https://scripps.ucsd.edu/zooplanktonguide/, accessed 20 August 2015.

[5] P. Grosjean, M. Picheral, C. Warembourg, and G. Gorsky, "Enumeration, measurement, and identification of net zooplankton samples using the zooscan digital imaging system," ICES Journal of Marine Science: Journal du Conseil, vol. 61, no. 4, pp. 518–525, 2004

[6] S. Leli'evre, E. Antajan, and S. Vaz, "Comparison of traditional microscopy and digitized image analysis to identify and delineate pelagicfish egg spatial distribution," Journal of Plankton Research, vol. 34, no. 6, pp. 470-483, 2012.

[7] F. Pedregosa, et al. “Scikit-learn: Machine learning in Python.” The Journal of Machine Learning Research vol. 12, pp. 2825-2830, 2011

[8] A. Forest, L. Stemmann, M. Picheral, L. Burdorf, D. Robert, L. Fortier, and M. Babin, “Size distribution of particles and zooplankton across the shelf-basin system in southeast beaufort sea: combined results from an underwater vision profiler and vertical net tows,” Biogeosciences, vol. 9, no. 4, pp. 1301–1320, 2012.

[9] H. M. Sosik and R. J. Olson, “Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry,” Limnology and Oceanography: Methods, vol. 5, no. 6, pp. 204–216, 2007.

[10] L. Bottou and C. Lin, “Support vector machine solvers,” Large scale kernel machines, MIT Press, 2007, pp. 301-320, 2007.

[11] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures,” Neural Networks: Tricks of the Trade, Springer Berlin Heidelberg, pp. 437-478, 2012.

[12] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,”CoRR, vol. abs/1409.1556, 2014.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton. “Imagenet classification with deep convolutional neural networks,” Advances in neural information processing systems, pp. 1097-1105. 2012.

Chapter 4, in full, is a reprint of the material as it appears in: Ellen, Jeffrey; Li, Hongyu.; Ohman, Mark D. “Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques.” Proceedings of OCEANS'15 MTS/IEEE, pp. 1-9, IEEE, Washington, D.C., 2015. DOI 10.23919/OCEANS.2015.7404607. The dissertation author was the primary investigator and is the primary author of this paper.

CHAPTER 5 Correlating Filter Diversity with Convolutional Neural Network Accuracy

Correlating Filter Diversity with Convolutional Neural Network Accuracy

Casey A. Graff
School of Computer Science and Engineering
University of California San Diego
La Jolla, CA 92023
Email: cagraff@ucsd.edu

Jeffrey Ellen
School of Computer Science and Engineering
University of California San Diego
La Jolla, CA 92023
Email: jellen@ucsd.edu

***Abstract*—This paper describes three metrics used to asses the filter diversity learned by convolutional neural networks during supervised classification. As our testbed, we use four different data sets, including two subsets of ImageNet and two planktonic data sets collected by scientific instruments. We investigate the correlation between our devised metrics and accuracy, and propose that these metrics could be used for a variety of tasks related to training CNNs. Including determining the best preprocessing method for non-standard data sets, diagnosing training efficacy including potentially reducing training time, or predicting performance in cases where validation data is expensive or impossible to collect.**

***Index Terms*—Convolutional Neural Network, regularization, normalization, preprocessing.**

I. Introduction

Convolutional neural networks have been demonstrated to achieve excellent results on a wide variety of supervised learning tasks. Our goal is to develop useful metrics to understand and enhance results with these networks.

We use four different, balanced data sets to help explore the generality of the metrics that we developed. The data from ImageNet is well documented and frequently used in CNN research; whereas the other two planktonic data sets selected are relatively obscure.

Our metrics all aim to measure the diversity in the weights of a network's first convolutional layer. As will be demonstrated, there is significant importance in these weights and their variance. Since we are unable to directly manipulate the variance of filters, we use normalization and L2 regularization as proxies; they impact the filter diversity indirectly.

By demonstrating the general purpose usage of our metrics we believe that they can be applied to many other data sets. The metrics may be useful for a variety of purposes, including diagnosing network performance, identifying over-fitting, and potentially improving weight initialization for large networks.

Contribution from the National Science Foundationsupported California Current Ecosystem Long Term Ecological Research site. Plankton sample analysis supported by NSF grants to Mark D. Ohman (mohman@ucsd.edu), and by the SIO Pelagic Invertebrates Collection.

II. Experimental Design

A. Data Set Description

In order to ensure meaningful results when comparing the effects of normalization on different datasets, and assessing the construction of the filters, it is vital to control the sets to be as similar as possible. This helps to ensure that observed differences are a result of the properties of the image classes or the treatments, and not a result of some property of the set itself, such as the number of classes or number of images per class. Each constructed dataset contains twenty-one classes of 1,000 images each. We re-sized all images, using center-padding and scaling, to 224x224 pixels with three color channels. Nearly every class sampled from the original datasets contained more than 1,000 images, in which case the 1,000 were selected randomly.

For each dataset an "other" class was included that contained samples of a large number of other logical classes that occurred infrequently. Our zooplankton dataset includes this class, but the ILSVRC dataset does not, so a similar class was artificially generated for the ILSVRC (All) and ILSVRC (Dog) constructed sets to be consistent.

1) ImageNet (All): The second dataset comes from the ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2015 data set [1] for the object localization challenge. Specifically, from the 1,000 classes (called synsets) used in the challenge, twenty classes were randomly selected. For the "other" class, an uneven distribution of fifty other synsets was used to simulate an approximate equivalent of the other category found in the plankton dataset.

2) ImageNet (Dogs): The third dataset comes from the same ILSVRC dataset and is comprised of twenty hand-picked synsets that are closely related. The intent is to construct a data set that mirrors the consistent visual similarity between classes that is present in the plankton data sets. In this case, all of the classes used were dog breeds. For this dataset, the "other" class contained uneven distribution of fifty other dog synsets.

3) Zooplankton: Our zooplankton images are acquired by a technology called Zooscan[2]. In essence, this is a extremely fine-tuned monochromatic flatbed scanner which is used on

preserved samples. Example ZooScan images are shown in (Fig. 1).

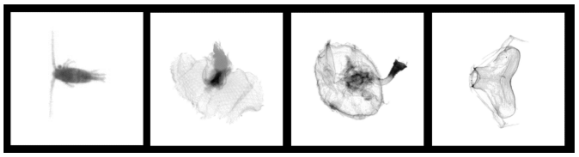


Fig. 1. Example ZooScan images: a copepod, jelly, pteropod, and siphonophore, all preserved and in unnatural postures and various states of completeness.

As shown, the background of these images is white. Prior to any normalization, the images in the plankton data set were centered by calculating the pixel value center of mass and shifting each sample to place this in the center of the image. This improved plankton validation and testing accuracy across all normalization techniques.

4) Phytoplankton: Our phytoplankton images are acquired by a technology called an Imaging FlowCytobot[3]. This technology images live cells $<$10 micrometers through use of a focused laser. Phytoplankton images are selected from [4], in a manner consistent with 20 of the major classes identified in [3]. As shown, the background of these images is noisier than the zooplankton images which results in the introduction of an artificial edge, not present in the Zooplankton data set, between the original background and the center-padding.

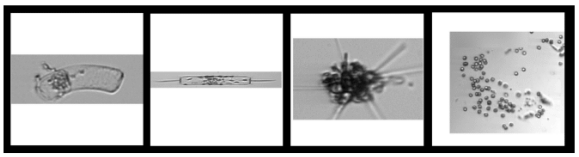


Fig. 2. Example FlowCytobot images: three diatoms (*Guinardia Striata*, *Ditylum* and *Asterionellopsis*, and Flagellate *Phaeocystis*, imaged alive and fully in tact.

B. Architecture

The original network architecture was based on the VGG-11 network architecture [5]. However, initial trials and investigation revealed that this architecture vastly over-fit to the training data. This was observed through analysis of validation loss curves, visualization of under-utilized input filters, and empirical testing of smaller network sizes. The final network selected has similar accuracy to the larger network on the datasets, while containing substantially fewer layers and parameters.

The final network architecture selected is as follows. Input (224x224 RGB image), Conv3-16, MaxPool-2, Conv3-32, MaxPool-4, Conv3-64, MaxPool-4, FC-1024, FC-21, Softmax.

All convolutional layers listed as "Conv⟨receptive field size⟩-⟨number of channels⟩" use a stride and padding of one. All fully-connected layers listed as "FC-⟨number of nodes⟩" use 50% dropout (except for the final fully-connected layer). Max pooling layers are listed as "MaxPool-⟨pool size⟩".

During our investigation several training batch sizes were compared. Initially a batch size of 50 was used with the originally larger network architecture. Lower batch sizes could only be used at the expense of additional training time. After reducing the size of the network substantially, it was found that the batch size could be reduced, yielding improved training accuracy with a negligible increase in training time.

We believe our implementation, although it uses a smaller amount of data and a smaller network than ImageNet provides a roughly equivalent testbed. Also, our implementation provides similar accuracy to support vector machines as reported for both the zooplankton [6] and phytoplankton data [3].

III. Normalization Investigation

Pixel values need to be normalized before used as input to a Convolutional Neural Network. For images where each pixel value is considered to be a feature, and not independent from its neighbors, there are many strategies to normalize the input. One option is to normalize all the values within a particular image, this per image normalization is frequently referred to as "Global Contrast Normalization". Another strategy is to normalize each pixel location across the whole stack of images separately, this per pixel normalization is frequently referred to as "Standardization". Another option is to decorrelate features and normalize their variance, whitening and Zero-phase Component Analysis, which is frequently called "ZCA whitening". ZCA whitening is commonly used for images. A fourth option is to normalize pixel values across patches of a single input, rather than the whole set of features, and this is referred to as "Local Contrast Normalization"[7]. We implemented all of these, and we found that per image normalization and per pixel normalization worked the best on our data.

A. Normalization Results

Normalization was applied by first separating the data into a training (80%) and testing (20%) set. Five of these splits were generated for each of the datasets. Once separated, the normalization parameters were fit on the training portion of the split, then applied to the entire split. We applied each of our normalization techniques separately to the three color channels.

$$Loss = E_{train}(W) + \lambda W^2 \tag{1}$$

We used L2 regularization which applies a weight λ to the squared values of the network's weights W and adds it the training error E_{train} to compute the loss value that is used to update the network. For each L2 regularization weight examined three trials were conducted (each using a unique split) for each dataset and normalization pair; with the exception being regularization value 0.001 which had five trials conducted per pair and regularization value zero which had two trials conducted per pair.

TABLE I
NETWORK TESTING ACCURACY AVERAGED ACROSS FIVE TRIALS

Dataset	Per Image	Per Pixel
ImageNet(All)	56.85	63.21
ImageNet(Dogs)	30.44	36.11
Zooplankton	71.06	69.18
Phytoplankton	79.25	79.70

IV. FILTER VARIANCE

For the rest of this paper, we will refer to the sets of weights from the first layer of the network as filters. These filters serve as the lowest level detectors, and they often evolve to respond highly to changes in intensity, such as edges.

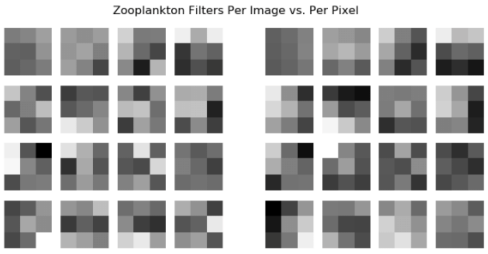


Fig. 3. Example of top performing Zooplankton filters; per image normalization on the left, per pixel on the right.

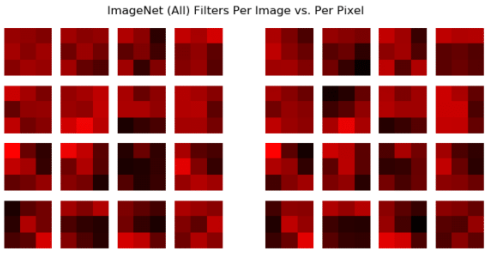


Fig. 4. Example of top performing ImageNet (All) filters (red channel only); per image normalization on the left, per pixel normalization on the right.

The pairs of filters presented in (Fig. 3) and (Fig. 4) are very similar because the are trained on the same split of the images in the same order. The values were transferred to the $0-255$ range for visualization for each image separately, so generalizations about intensities between images can not be made, however, they illustrate not only the nature of the filters, but also the relative difference between corresponding pairs of filters in the per image vs per pixel normalization strategies.

When considering the variance with respect to the filters, there are a few different ways to consider the variance. First is the variance within the weights of an individual filter. We hypothesize that this will correlate to how sharply adjacent features vary within a particular image. Second is the variance within the weights of a particular set of filters built by a single model. We hypothesize that this will correlate to how much feature values vary within all regions of all images in a particular dataset. Third is the variance between individual filters within a model. We hypothesize that this will also correspond to the variety of the values of features within a particular data set.

We define relevant metrics for these three concepts in the context of our investigation. The ImageNet images are standard RGB images, and the planktonic data sets are single-channel due to their acquisition mechanisms, so we render them in greyscale by copying their input across all three channels to keep the networks exactly the same; operating on 3-channel 224x224 images. While each channel in the ImageNet is marginally different from the others, overall the pattern holds and for simplicity we present all ImageNet results as the average across all three color channels.

A. Variance within Individual Filters

First, we consider the variance within individual filters. Each of our filters has 9 weights (3x3) and use the standard deviation of these 9 values as a measure of the variance within the filter. Since we have 16 filters learned per model trained, we take a simple arithmetic mean of these values to provide a single number reflecting the variance learned by that particular model, $\overline{\sigma}_F$, which is shown in (Eq. 2)[1], where x_i is an individual weight for filter f.

$$\overline{\sigma}_F = \frac{1}{16}\sum_{f=0}^{16}\sqrt{\frac{1}{8}\sum_{i=1}^{9}(x_{f,i}-\overline{x})^2} \tag{2}$$

Our results are shown in (Fig. 5).

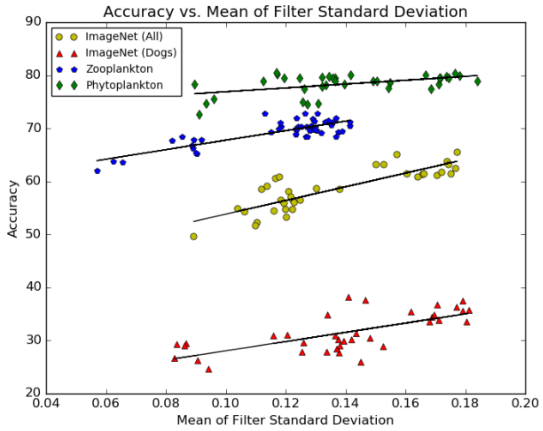


Fig. 5. The regression lines indicate a positive correlation between accuracy and $\overline{\sigma}_F$ across all datasets.

Each datum in (Fig. 5) corresponds to a single model trained as described in our Normalization Experimentation section.

[1]Given our small sample, we use Bessel's correction when calculating our standard deviation

Also shown are regression lines for each set of data points. Correlation results are in (Table II), and the last column, Combined PCC, refers to the regression lines in (Fig. 5).

TABLE II
PEARSON CORRELATION COEFFICIENT OF ACCURACY VS $\overline{\sigma}_F$ FOR EACH NORMALIZATION

Dataset	Per Image PCC	Per Pixel PCC	Combined PCC
ImageNet(All)	0.717	0.581	0.844
ImageNet(Dogs)	0.512	0.742	0.700
Zooplankton	0.809	0.874	0.824
Phytoplankton	0.669	0.687	0.491

Given that some of the trials had different data splits, and the noise present in the learning process, we do not expect a strict tolerance in the results, so we interpret values above 0.8 to indicate a very strong correlation, and values above 0.6 to indicate a strong correlation between accuracy and $\overline{\sigma}_F$. Table II also shows that the correlation holds whether considering across both normalizations, as pictured in (Fig. 5) or considering the effects of a single normalization strategy.

If there is a causal relationship between $\overline{\sigma}_F$ and accuracy, then to maximize accuracy, we should try to intentionally increase $\overline{\sigma}_F$.

The data points in (Fig. 5) are on a fixed size network for a particular data set. Many network hyperparameters were held constant, including the initialization of the weights. The only three things creating variation are the type of normalization, the amount of regularization, and the split of the data. The split of the data is not something that can be controlled in a valid matter, but the regularization is directly a hyperparameter.

The effect of regularization on $\overline{\sigma}_F$ is straightforward. The filters are the solution to an optimization problem of responding most strongly to the image patches that are most diagnostic of discriminating between classes. An individual filter having a higher $\overline{\sigma}_F$ means that its individual weights are more spread out. Since regularization is designed to reduce the magnitude of the weights, any filter weights with a high standard deviation must be very rewarding to avoid being regularized. This relationship is evident in (Fig. 6).

Individual data points represent separate trials with the same parameters on different splits of the data, and the lines drawn connect the average values of each data set. The bimodal distribution is due to the two different types of normalization. As the regularization increases, $\overline{\sigma}_F$ decreases. Since there is a high correlation between acuracy and $\overline{\sigma}_F$, the same relationship exists between accuracy vs. regularization as shown in (Fig. 7).

Again, the trials are the individual data points and the lines connect the averages. The lines of (Fig. 7) appear to be more flat than the lines in (Fig. 6), but this is because of the scale of the y-axis. But both graphs peak in similar places, as we would expect with them being highly correlated. The shape of this graph is well known, and why the optimal amount of regularization is sought via a search. But our investigation provides insight into the mechanism for this behavior.

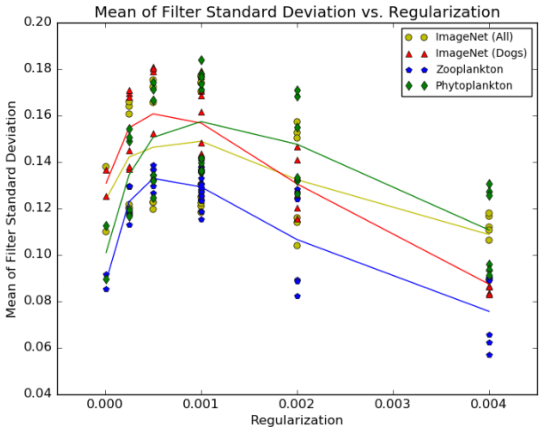


Fig. 6. The relationship between $\overline{\sigma}_F$ and regularization across all trials for all datasets.

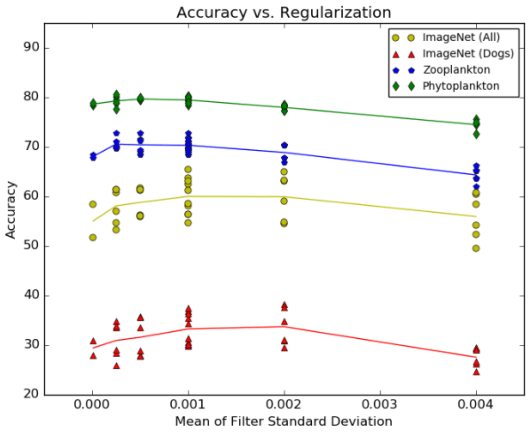


Fig. 7. The relationship between accuracy and regularization across trials for all datasets.

B. Variance Between Filters

We previously enumerated two other concepts describing the variance of filter weights. To investigate the impact of the distribution of the feature values within a particular image on the weights, we calculate the model's global filter standard deviation, specifically the standard deviation of all 144 weights in the first layer of the matrix as shown in (Equation 3).

$$\sigma_{\forall F} = \sqrt{\frac{1}{143}\sum_{f=0}^{16}\sum_{i=1}^{9}(x_{f,i} - \overline{x})^2} \tag{3}$$

We also want to investigate the variance between filters within an individual model. Since the $3x3$ weights comprising our filters in our convolutional neural network are always applied in the same orientation, simple matrix subtraction is

appropriate, and will function similar to a Hamming Distance, roughly describing how far apart the two filters are. We calculate this distance in (Equation 4).

$$\overline{\Delta}_F = \frac{\sum_{f=0}^{16} \sum_{g=f}^{16} \sum_{i=1}^{9} |x_{f,i} - x_{g,i}|}{{}_{16}P_2} \quad (4)$$

We then calculate these metrics for our data, as shown in (Fig. 8).

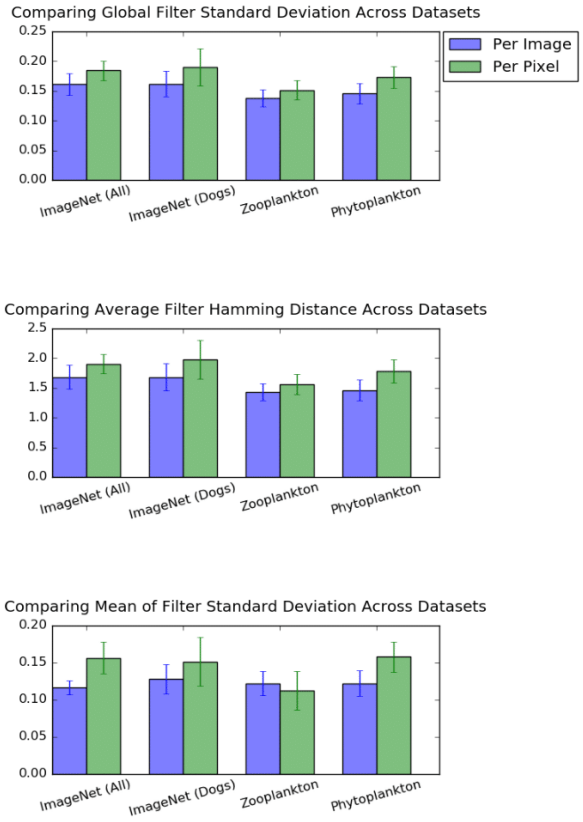


Fig. 8. Bar graph demonstrating increased diversity when using per pixel normalization.

This graph views all three metrics, $\overline{\sigma}_F$, $\sigma_{\forall F}$, and $\overline{\Delta}_F$ for all trials, including all regularization strengths. In 11 out of 12 cases, per pixel normalization yields higher values for filter diversity than per image normalization. And this is not just an artifact of considering each image set individually, but also occurs when considering the trials in aggregate. as shown in the histogram (Fig 9).

Given the diversity of our image types, we feel this would hold for any types of images. The second pattern is that for both $\sigma_{\forall F}$ and $\overline{\Delta}_F$, the values are larger for the two ImageNet data sets than for the two planktonic data sets. This backs

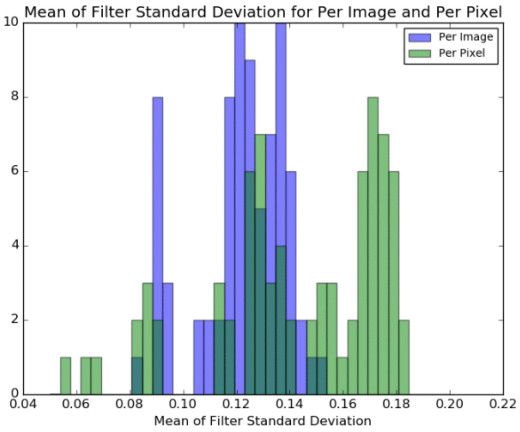


Fig. 9. Histogram illustrating the difference in distribution of filter diversity between normalization techniques.

an intuition that the two planktonic data sets are more 'subtle' than the ImageNet ones, and despite their better overall higher accuracy, they have lower filter diversity in two metrics.

V. Implications of Filter Variance on Classification Accuracy

As shown in (Table I), normalization strategy has a clear impact on accuracy. We found that one method of normalization generally outperformed the other regardless of regularization strength and other experiments not included for succinctness, such as network size. The superficial conclusion is therefore that one type of data is better suited to a particular normalization strategy than another. Our investigation sheds light on why this is the case. Figure 8 isolates the effect of normalization on all three of our metrics.

In every case where per pixel normalization results in higher $\overline{\sigma}_F$ than per image normalization, per pixel normalization also results in the highest accuracy. Agreeably, zooplankton is the one data set for which per pixel normalization had higher $\overline{\sigma}_F$, but it also achieves better accuracy using per pixel normalization.

To investigate further, we examined a confusion matrix of the results of a particular split (so the images are exactly the same for each normalization strategy). The results are shown in (Fig. 10).

The three classes with the biggest improvement (larger on-diagonal numbers) are polychaetes, nauplii, and pteropoda, three classes which have delicate, feathery appendages which are the most distinguishing feature between them and their closest neighbor in shape (chaetognath, copepods, and eggs respectively). The next largest improvements are in others, copepoda others, and crustacea others. These are three categories that obviously have a larger diversity of organisms in their images than some of the other classes. So it seems feasible that the reduced filter diversity is actually a benefit in this case.

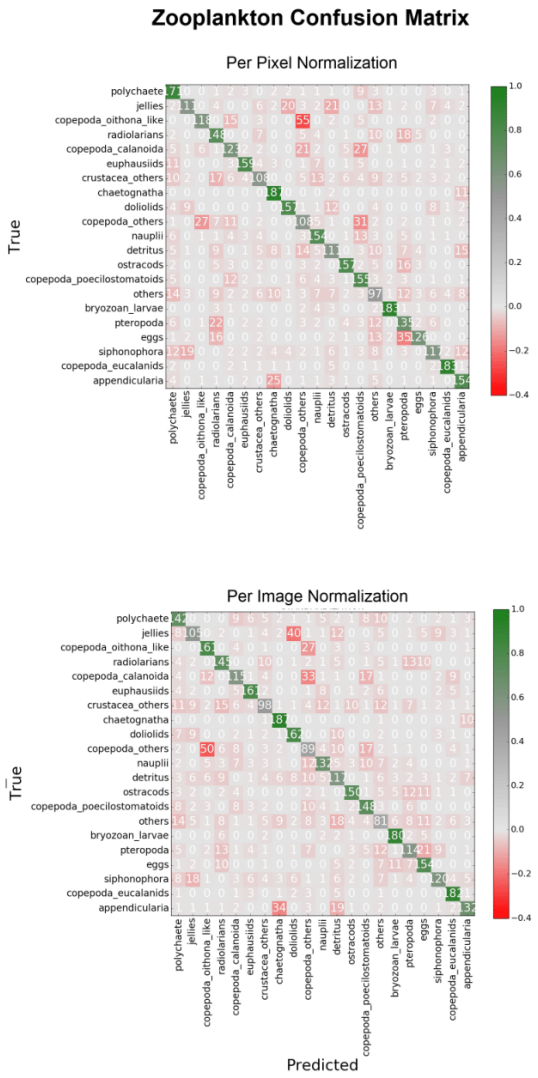


Fig. 10. Histogram illustrating the difference in distribution of filter diversity between normalization techniques.

We then infer that for images with less statistical variation in the pixels, such as the zooplankton images, per pixel normalization performs the best. Note that the phytoplankton images, with a moderately noisy background, show very little difference per normalization method, and therefore our dataset may not be diverse enough for the normalization strategy to matter.

We also propose that knowledge of this metric could be used to assist in training large networks. Our network, along with many others, uses initialization. This initialization strategy, along with many others, is designed to speed the convergence of the network. We propose a modified strategy that generates a number of different candidate initializations, calculates the $\overline{\sigma}_F$ for each one, and selects the one with the highest $\overline{\sigma}_F$ as the best candidate. This one-time calculation would trivially add to the network run time. As our network was relatively shallow, and our convergence times fast, we did not assess this with our data.

Similarly, we believe this metric could potentially help reduce the size of the validation set. There is a class of potential supervised classification problems, particularly in the scientific domain (such as medical imaging), for which many thousands or millions of training examples would be difficult or impossible to obtain. In this case, strategies such as leave-one-out cross validation attempt to overcome this lack of data, but would require as many models to be computed as folds of the data. This potentially makes grid search and other operations prohibitively expensive. Instead, we propose that filter diversity could potentially be used to assess the best performing model. More investigation on larger and diverse data sets would be required to fully verify this claim.

VI. Conclusion

In this paper, we described three metrics used to asses the filter diversity: $\overline{\sigma}_F$, $\sigma_{\forall F}$, and $\overline{\Delta}_F$. These metrics are intended to measure the diversity within a single filter, as well as across all filters. For all four of our data sets, we found a strong correlation between our devised metrics and accuracy. We feel that these metrics could potentially be used in a variety of ways to improve training models as well as determining the best model for deployment. This include identifying the best preprocessing method for non-standard data sets, potentially improving convergence time, and predicting performance in cases where validation data is valuable because it is expensive or impossible to collect.

Acknowledgment

The authors would like to thank Professor Mark Ohman and Professor Charles Elkan for their support.

References

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[2] P. Grosjean, M. Picheral, C. Warembourg, and G. Gorsky, "Enumeration, measurement, and identification of net zooplankton samples using the zooscan digital imaging system," *ICES Journal of Marine Science: Journal du Conseil*, vol. 61, no. 4, pp. 518–525, 2004.

[3] H. M. Sosik and R. J. Olson, "Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry," *Limnology and Oceanography: Methods*, vol. 5, no. 6, pp. 204–216, 2007.

[4] H. M. Sosik, E. E. Peacock, and E. F. Brownlee, "Annotated plankton images - data set for developing and evaluating classification methods." [Online]. Available: http://dx.doi.org/10.1575/1912/7341

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[6] J. Ellen, H. Li, and M. D. Ohman, "Quantifying california current plankton samples with efficient machine learning techniques," in *OCEANS 2015 - MTS/IEEE Washington*, Oct 2015, pp. 1–9.

[7] Y. LeCun and M. Ranzato, "Deep learning tutorial," in *Tutorials in International Conference on Machine Learning (ICML13)*. Citeseer, 2013.

Chapter 5, in full, is a reprint of the material as it appears in: Graff, C. A.; Ellen, Jeffrey. "Correlating Filter Diversity with Convolutional Neural Network Accuracy." 15th IEEE International Conference on Machine Learning and Applications, pp. 75-80, IEEE, Anaheim, CA, 2016. DOI 10.1109/ICMLA.2016.0021. The dissertation author is an equal contributor in the investigation and authoring of this paper.

CHAPTER 6 Improving plankton image classification using context metadata

6.1 Abstract

This chapter shows how to boost the performance of CNN classifiers by incorporating metadata of different types, and illustrates how to assimilate metadata beyond simple concatenation. We utilize both geotemporal (e.g., sample depth, location, time of day) and hydrographic (e.g., temperature, salinity, chlorophyll-a) metadata and show that either type by itself, or both combined, can substantially reduce error rates. Incorporation of context metadata also boosts performance of the feature-based classifiers we evaluated: Random Forest, Extremely Randomized Trees, Gradient Boosted Classifier, Support Vector Machines, and Multilayer Perceptron. For our assessments, we use an original data set of 350,000 in situ images (roughly 50% marine snow and 50% non-snow sorted into 26 categories) from a novel in situ Zooglider. We document asymptotically increasing performance with more computationally intensive techniques, such as substantially deeper networks and artificially augmented data sets, each bringing slightly greater accuracy apparently approaching a limit. Our best model achieves 92.3% accuracy with our 27-class dataset. We provide guidance for further refinements that may provide additional gains in classifier accuracy.

6.2 Introduction

The burgeoning number of digital imaging methods available to aquatic ecologists, both in situ (Davis et al. 1992; Samson et al. 2001; Benfield et al. 2003; Watson 2004; Olson and Sosik 2007; Cowen and Guigand 2008; Picheral et al. 2010; Schulz et al. 2010; Thompson et al. 2012; Briseño-Avena et al. 2015; Ohman et al. 2018) and in the laboratory (Sieracki et al. 1998; Gorsky et al. 2010), is generating rapidly expanding libraries of digital images useful in a variety of scientific applications. However, the accumulation of large numbers of images increases the

need for much more efficient machine learning methods in order to automate the processes of image classification, data extraction, and analysis.

Until recently, most automated image classification has employed methods we refer to as 'feature-based,' in that they operate on a set of descriptive geometric features calculated from the digital images, such as area, shape, aspect ratio, fractal dimension, textures, and gray scale histograms. The feature-based algorithms then derive a mapping from the calculated values to labels corresponding to the type of organism. Ideally this mapping will extrapolate to future images. Some of the feature-based algorithms that have been applied to classification of plankton images with varying degrees of success include random forests (Grosjean et al. 2004; Gorsky et al. 2010), support vector machines (Hu and Davis 2005; Sosik and Olson 2007; Ellen et al. 2015), and multilayer perceptrons (Wilkens et al. 1996), among others.

Since 2012, “deep learning” algorithms (Krizhevsky et al. 2012; LeCun et al. 2015) have outperformed feature-based methods in a variety of fields, including natural language processing (Socher et al. 2013), time series analysis (Graves et al. 2013), variational autoencoders (algorithms that learn to generate or alter existing data, such as image correction; Kingma and Welling 2013), zooplankton image analysis (Orenstein et al. 2015; Dieleman et al. 2016b; Dai et al. 2016; Graff and Ellen 2016; Wang et al. 2016; Zheng et al. 2017), and others. Multiple algorithms have been characterized as examples of deep learning, the commonality being the use of repetitive layers of algorithmic structure that operate on the prior layers rather than the original input. Deep learning algorithms tend to require orders of magnitude more computation, although often such computations are highly parallelizable and can be done rapidly given appropriate hardware. Among the most commonly adopted deep learning methods are convolutional neural networks (CNNs). CNNs have been applied to a spectrum of image

recognition problems (e.g., LeCun et al. 1998; Matsugu et al. 2003; Yue-Hei Ng et al. 2015; Esteva et al. 2017). Applications of CNNs and random forests to phytoplankton image classification include Orenstein et al. (2015), while further applications of CNNs to coral, plankton, and fish classification are surveyed by Moniruzzaman et al. (2017).

Convolutional Neural Networks obviate the need for explicit geometric image measurements to be defined and generated, and instead operate directly on the 2-dimensional image contents. When a human examines an image captured with discrete pixels such as figure 1a, the Gestalt theory of perceptual grouping states that we do not primarily perceive individual dots of colored ink or light, but instead comprehend unified shapes in relation to complete objects, such as in figure 1b (Wertheimer 1923). This recognition may consist of simple objects such as “tunic,” “stomach,” and “salp,” or more specific objects based on the viewer's expertise, such as “circumferential muscle bands” or “endostyle” (Wagemans et al. 2012).

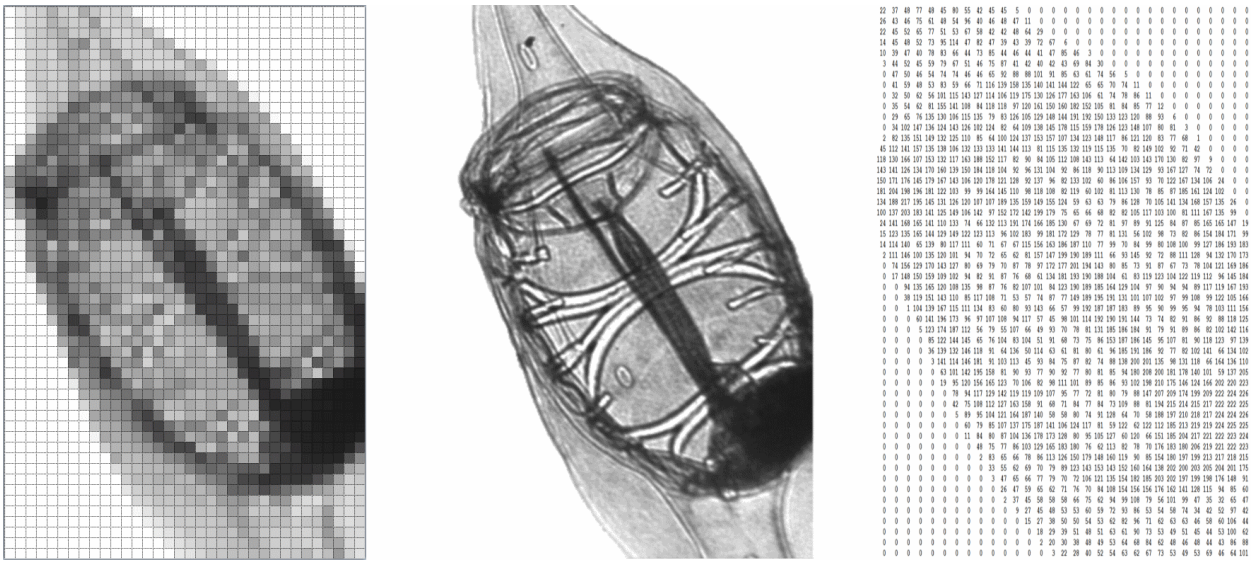


Figure 6.1: Multiple renderings of a salp zooid (a) at low resolution (b) at full resolution typical of Zooglider, which a human generally perceives as contiguous, unified shapes, and (c) a numerical representation of the intensity values in (a).

A computer’s perception is entirely different, lacking these higher level taxonomic or morphometric concepts. Computer ‘vision’ is limited to a grid of integer values (Fig. 1c) and

concepts such as “42 dark gray pixel values” or “123 contiguous non-zero pixels.” Feature-based methods use summary statistics such as perimeter or mean intensity to describe the image or object. By contrast, CNNs generate independent statistics for a lattice of sections of the original image, and repeat this process at multiple scales to build a statistical summary of the entire image contents, starting with a summary of the pixels at the lowest level, and building towards higher level object concepts.

CNNs apply a system of hierarchical filters to the grid of pixels in a manner inspired by Hubel and Wiesel’s investigation of receptive fields within the visual cortex (Hubel 1959; Hubel and Wiesel 1963). The lowest layer of the CNN consists of a set of filters as in figure 2a. These filters are initialized by either generating random values, or adopting a set of filters from a previously trained CNN. The filters are then convolved against the input image, i.e., performing element-wise multiplication between the filter and the region of the image that it covers for every possible region in the image. Every filter’s convolution is input for a neuron, which sums these inputs, and applies a non-linear activation function that produces higher valued output when the match between the filter and input region’s high values are closely correlated (Fig. 2b). The neuron’s output is used as input for the next layer of filters. Each subsequent layer of filters is similarly applied to its predecessor. During the training phase, as labeled images are assessed, the algorithm gradually adjusts these filters so that they are the most useful for determining differences between classes. Early layers of filters usually evolve to identify low level visual concepts such as colors, corners, and edges at a particular orientation as in the example in figure 2. Secondary filters typically correspond to mid-level concepts such as curves and textures, potentially equating to muscle bands or outer tunic. Additional layers of filters evolve against

their predecessors' output, ideally resulting in high level objects such as peripharyngeal band or testes that are useful for determining the final classification label.

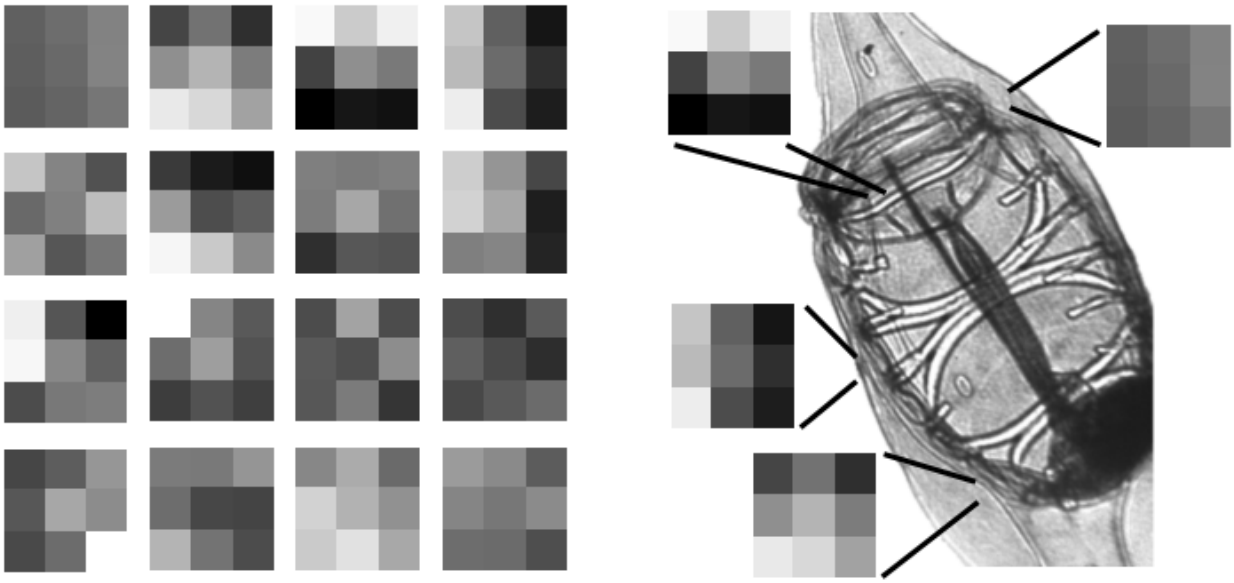


Figure 6.2: Conceptual application of filters to an input image as in the first layer of a CNN. (a) A bank of 3x3 filters. (b) Conceptual representation of regions where a particular filter from (a) would have a strong response to the salp input image: e.g., a sharp horizontal edge at the top of a muscle band, or a dark-to-light gradient mid-tunic.

Although CNNs and feature-based methods operate on different representations of the image data, a limitation of both approaches is that they utilize only the information contained in the image. In contrast, human taxonomists consider the context in which the sample was acquired when making identifications. For planktonic organisms, collection information such as geographic location, season, depth, time of day, and hydrographic conditions provide context metadata that may help constrain the realm of plausible answers and facilitate the identification process. The concept of utilizing metadata to improve image classification has been explored in other domains. One early work on classifying tourism photography used GPS information in conjunction with the images to improve identifying landmarks (Li et al. 2009). Other work incorporated GPS information to generate metadata such as elevation, average vegetation, and

congressional district and explored two different ways of incorporating the metadata to achieve a 5 point gain in accuracy on a 100-way classification task of common objects and scenes (Tang et al. 2015). While incorporating context metadata into feature-based classifications is straightforward, it is more challenging to include such metadata into CNNs.

In this chapter we assess whether incorporation of different types of context metadata improves classification accuracy for both CNNs and feature-based methods. Our numerical experiments are based on an original library of validated images from *Zooglider* (Ohman et al. 2018), a novel in situ zooplankton imaging device. We will illustrate how to optimize the use of metadata. In addition, although machine learning methods involve many parameter values that can markedly affect the efficacy of a classifier, many practitioners simply adopt default values in commonly available software packages. We illustrate the benefits of tuning hyperparameters for both CNNs and five of the most common feature-based methods, and provide guidance for selecting hyperparameter values (where a hyperparameter is an overarching parameter whose value is chosen before the learning algorithm optimizes the model's parameters). We assess the performance of feature-based algorithms against CNNs of varying size and complexity, and quantify the benefit of including metadata.

6.3 Materials and procedures

6.3.1 Machine learning algorithms and image processing software

In addition to CNNs, we used five feature-based algorithms: Random Forest Classifier (RFC), Extremely Randomized Trees (XRT), Gradient Boosted Classifier (GBC), Multilayer Perceptron (MLP), and Support Vector Machine (SVM).

The Random Forest algorithm constructs an optimal decision tree by fitting it to a bootstrap sample drawn from the training set. Once that tree is optimized, more trees are

constructed up to a threshold (Ho 1995). We also used two more recent modifications of RFC. The Extremely Randomized Trees algorithm uses stochastic partitions of the data instead of all data, and stochastic tree construction conditions instead of fully optimizing each tree (Geurts et al. 2006). These modifications usually cause faster algorithm convergence while producing similar or better results (Criminisi et al. 2012). The other RFC variation we use, Gradient Boosted Classifier, draws on the concept of boosting, where a collection of weak models can be combined into a stronger one (Freund and Schapire 1997), in this case more abbreviated decision "stumps" instead of full trees (Friedman 2001). We also assess Support Vector Machines, which construct a decision boundary that optimally divides the space between all the samples based on their overall proximity to each other in the metric space (Cortes and Vapnik 1995), rather than directly operating on sampled values of individual features, as in RFC. Finally, we assess Multilayer Perceptron (Rumelhart et al. 1986), where each neuron produces a flat subset within the decision space, and by learning these flat subsets collectively forms a complex decision surface (Haykin 2009) that is extremely flexible (Lippmann 1987).

We used the Python programming language (van Rossum 1995) for high level data handling and general computation. We used OpenCV (Bradski 2000) for image processing and manipulation. For RFC, XRT, GBC, and SVM we used Scikit-Learn (Pedregosa et al. 2011). For MLP and CNN we used the Lasagne library (Dieleman 2016a) to specify our models, which were then executed by the Theano Framework (Al-Rfou et al. 2016). Alternative CNN implementations are available in TensorFlow, Caffe, and Torch, among others.

6.3.2 Computational equipment

We performed smaller numerical experiments on a simple server with 40 CPU cores and 128GB of RAM. Because CNNs are optimized for performance on the hundreds/thousands of

weaker computational cores found in graphics processing units (GPUs); our server also had an NVIDIA K40 GPU. For larger scale experiments, we utilized NSF's Extreme Science and Engineering Discovery Environment (XSEDE.org) which provided us access to dozens of GPUs simultaneously via their nationwide supercomputing resources. While each individual model we evaluated can be assessed on a single GPU, using the computational resources of XSEDE allowed us to more thoroughly and efficiently conduct experiments, which consisted of many thousands of trials and replicates.

6.3.3 Image acquisition

Our images were acquired by *Zooglider*, an autonomous vehicle with a Zoocam bearing a telecentric lens system that enables in situ imaging of planktonic organisms and particles in a volume of~250 mL per frame (Ohman et al. 2018). Zooglider operates from 400-0 m depth and images at a frequency of 2 Hz. *Zooglider* also measures conductivity, temperature, depth, and chlorophyll-*a* fluorescence, and has a dual frequency *Zonar* (200/1000 kHz; Ohman et al. 2018) intended to measure acoustic backscatter from objects approximately the same size as those imaged by the Zoocam (0.5mm to 50mm). We performed image correction of Zoocam image frames, including de-noising and gamma correction, to improve contrast, as described in more detail in chapter 3. These operations help improve segmentation accuracy. Segmentation is the process of identifying which particular pixels serve as edges and lie on the boundary between two contiguous regions in an image. We used a custom, two-pass version of Canny edge detection (Canny 1986, Ohman et al. 2018) to segment Regions of Interest (ROI) within the field of view.

6.3.4 Image compilation

It is important that the images selected for annotation are drawn without bias, that is sampling the images to reflect what will be acquired, such as not labeling only the largest and easiest to identify ROI, or only the first ROI from a time period. This topic and other good practices for validating feature-based classification of plankton images are discussed by González et al. (2017). From a larger collection, ~2 million ROI were selected in an unbiased manner and classified. Out of the ~2 million ROI, we used the resulting 178,547 non-snow ROI in our numerical experiments. For our largest data set, we combined all non-snow ROI with 171,453 randomly sampled marine snow ROI (out of the ~1.8 million positively identified) to total 350,000 ROI (Table 1). Images were assigned to 27 categories (Table 1 and Fig. 3).

Table 6.1: Distribution of the 350,000 ROI in our largest data set. Examples for each of the 27 classes are provided in figure 3.

Acantharian_sun_like	4418	Diatoms_high_concentrations	4899	Phaeodarean	1159
Acantharians	1659	Diatoms_no_spines	20088	Quasispheres	5387
Appendicularians	14250	Diatoms_w_spines	10115	Snow	2M+
Chaetognaths	2170	Disks	2150	Spheres_egg_like	1345
Cnidarians	2358	Euphausiids	913	Spheres_white	1627
Comets	1151	Fish_Larvae	789	Tentacles	28984
Copepods	44662	Foramanifera	320	Tentacles_white_streaks	1935
Copepods_Nauplii	371	Narcomedusae	673	Threads	8043
Dense_background	2272	Overturns	8734	Translucent_spheres	981

We have previously found that approximately 1,000 images per class is a rough guideline for the number of examples a class needs to be well defined (Graff and Ellen 2016), so for the purposes of early trials and debugging, we created a limited data set of no more than 1,000 examples per class, which yielded a total of 25,047 ROI. We constructed a second data set by

capping each class at 5,000 examples, yielding 76,190 ROI. Most of our explorations were executed on this dataset.

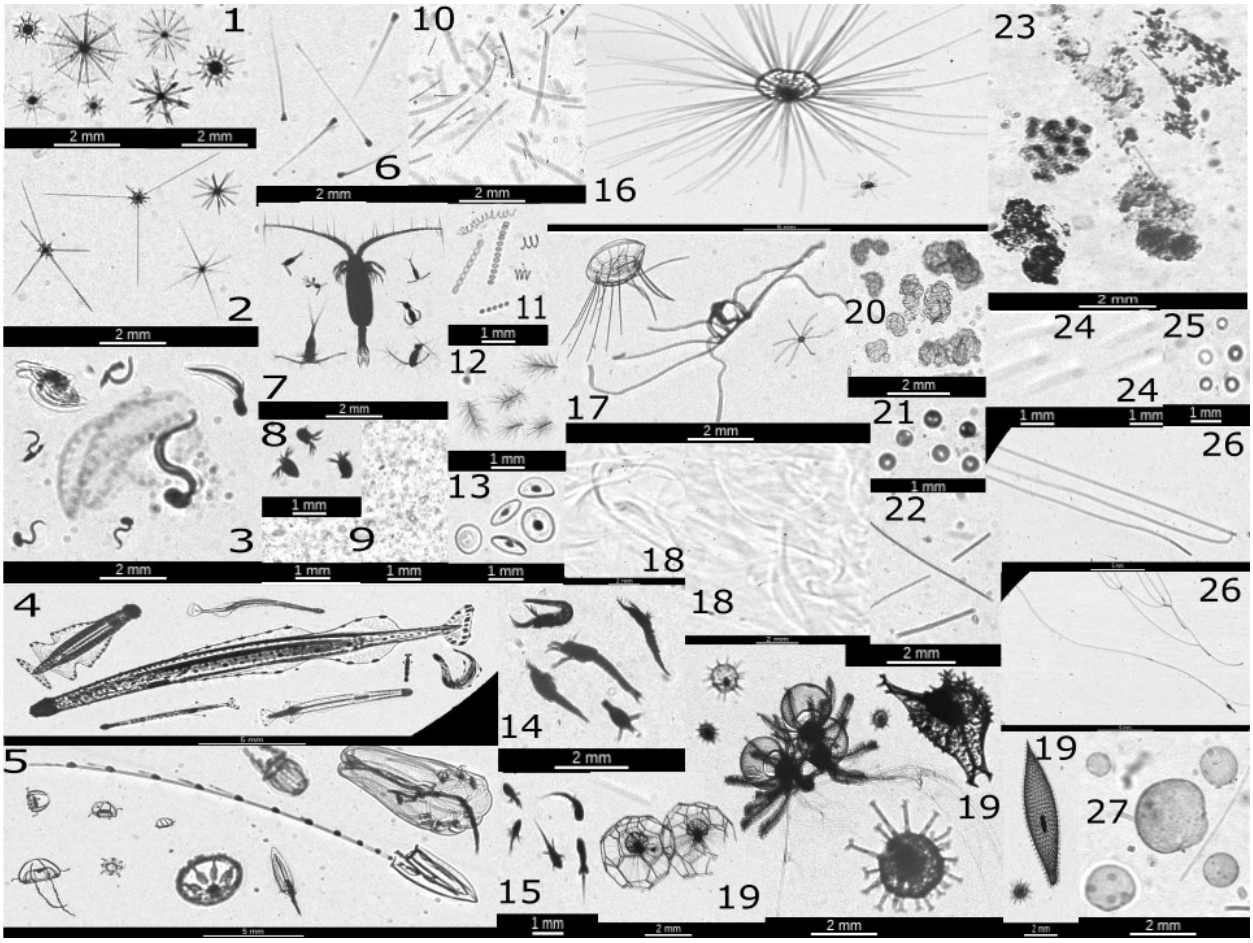


Figure 6.3: Representative ROIs for each of the 27 classes imaged by *Zooglider*.

Our main assessments use this largest dataset. We arrived at 350k by evaluating larger datasets, but found no appreciable difference in accuracy, yet incurred longer run times. All ROI including snow were randomly sampled from their respective classes to avoid introducing biased metadata or other anomalies. The overall data set sizes are 1.5GB, 4.7GB, and 21.5GB, with the metadata (described below) approximately 0.1 GB for each data set. CNNs require uniformly sized images. Based on the size of the majority of our ROI, we selected 128x128 pixels, which required rescaling some larger ROI, thus losing some detail, and adding neutral pixels to smaller

ROI in order to conform to this size. We used resampling with the Lanczos filter to resize the images (Blinn 1998).

6.3.5 Hydrographic, geotemporal, and geometric metadata

We used three types of context metadata: hydrographic, geotemporal, and geometric (Table 2). Hydrographic metadata are intended to reflect the in situ environment of the specific water parcel in which the image was acquired. These metrics include *Zooglider* measurements of chlorophyll-*a* fluorescence, salinity, density, and temperature; the local upwelling index (Schwing et al. 1986; PFEL 2018 for 33°N, 119°, averaged for the 10 days preceding each *Zooglider* image), and two different ways to approximate object concentration: acoustic backscatter and distance between ROI. Chlorophyll-*a* fluorescence, salinity, density, and temperature measurements are made by *Zooglider* every 8 seconds, while Zoocam images are acquired at 2 Hz, hence measurements are linearly interpolated to each Zoocam frame. We also use as metadata acoustic backscatter at the two acoustic frequencies, and the difference between them, which helps distinguish small and large sound scatterers. The Zonar does not ensonify the same volume as that being imaged, so the acoustic return is not a property of any recorded ROI, but provides context information about the aggregate density of nearby sound scatterers. Acoustic backscatter is averaged in 1 m depth bins. The full frame image from which ROI are segmented also provides information about nearby particle density. We calculate the individual distances from each region to its nearest neighbor ROIs in the frame, up to 5.

Our geotemporal metadata identify the place and time that the image was acquired. Values measured directly aboard *Zooglider* are hydrostatic pressure, time of image capture, and latitude and longitude interpolated between each glider surfacing. Based on these position values, bottom depth is obtained from ~100 m grid cells calculated by downsampling bathymetry

(NOAA 2012; 2018). We also calculate distance to Point Conception (a major upwelling center) and distance to the Santa Barbara Basin (a productive area). Distance to the coast and distance to the nearest continental slope (600 m) are calculated using the downsampled bathymetry. We generate four types of temporal metadata: time of day (divided into 8 time intervals); season (4 seasons, each 3 months long); El Niño-Southern Oscillation index off California (monthly, from Lilly and Ohman 2018); and Pacific Decadal Oscillation (monthly, from Mantua et al. 1997).

Geometric features extracted from the images were used as a third type of metadata for the CNN architectures (geometric features are required for feature-based approaches). The geometric values are calculated with respect to the pixels that are designated by the segmentation algorithm as being within the region (e.g., mean intensity, kurtosis, area, diameter, weighted centroid). While these values are derived from information within the image itself, the geometric features are metadata in that they describe the original image contents and ROI size before the image is rescaled and pixel values are adjusted for processing by the CNN. These values include measurements of the segmentation boundary, such as perimeter length and eccentricity, and information about the originally measured intensity values, such as minimum, maximum, and average, which are not otherwise provided to the CNN. Combined, they provide context about the illumination and scale of the original image capture. Additional detail regarding these 58 geometric measurements is provided by Ellen et al. (2015).

6.3.6 Procedures

For each of our assessments, we split the data into 80% for training, 10% for validation, and 10% as the test set. We generated 10 different randomly selected sets with these split ratios as replicate trials.

Most of the algorithms are designed to accept feature values across a defined range, usually [0-1] or [-1, 1]. In prior work, we examined four different whitening and normalization techniques, and found that with our images, per image normalization worked best (Chapter 5 - Graff and Ellen 2016). Commonly referred to as Global Contrast Normalization, the mean value of the image is subtracted from each pixel, and the result is divided by the standard deviation of the original pixel values. Since each type of metadata measurement has a scale different from the others (e.g., temperature or sampling depth) we also subtracted the mean of the measurement from its observations and divided by the standard deviation. All normalizations are calculated using the 80% split of training data for each replicate.

We calibrated each model to each replicate of the data, a process commonly referred to as hyperparameter tuning. While some of our feature-based algorithms require minimal tuning, CNNs require more careful evaluation to achieve a strong model. Training a single CNN consists of evaluating the network's performance on an image, then adjusting the network weights to reinforce good performance and alter bad performance. This is usually done by selecting one of the images at random without replacement, processing it, then selecting another. The term ‘epoch’ is used to describe the condition where the network has seen each training image one time.

This workflow creates a number of different options and hyperparameters, not all of which were evaluated. We used a batch size of 25 to evaluate multiple images simultaneously, thus increasing throughput. We imposed a limit on the number of epochs at 40, but this limit was rarely needed (see Bengio 2012 and Smith 2018 for guidance on stopping criteria and other hyperparameter choices). We also assess data set augmentation (Dai et al. 2016; Dieleman et al. 2016b), which involves generating synthetic examples to improve overall accuracy. Because our

images are captured with known pixel pitch and images are centered by our segmentation process, we only assessed horizontal reflection, vertical reflection, and rotation.

6.3.7 CNN architecture

We trained our CNNs *de novo*, rather than adopting networks from different application domains because our *de novo* results were markedly better in both this and our previous work (Graff and Ellen 2016). Initial networks have nearly random weights and no discriminative power. With each successive example, weights are adjusted. The learning rate controls the amount the weights are adjusted to respond to the most recent example and is an important hyperparameter. We used the Adam optimization algorithm (Kingma and Ba 2015), which updates all network values, in addition to modifying the initial learning rate. Two initialization algorithms are made available through the Lasagne/Theano software (Glorot and Bengio 2010; He et al. 2015). We evaluated both, found no significant difference, so we used Glorot and Bengio (2010).

Network shape has a large impact on results, and is an active area of research (Lee et al. 2015; He et al. 2016; Szegedy et al. 2017; Sabour et al. 2017). We implemented a network shape based on the VGG-16 model (Simonyan and Zisserman 2014), but on a smaller scale; since theirs was a 1000-way classification problem with images of 224x224. We also used small filters of size 3x3 for every layer and the rectified linear unit activation function (ReLU), but otherwise the convolutional portion of our network was approximately one quarter the size of their network. We had a total of 5 convolutional layers, with 16, 32, 32, 64, and 64 filters respectively, with a pooling layer between each one (Fig. 4). The pooling layers serve to reduce the input size from one layer to the next by half, using maximum value pooling: that is, for each 2x2 area of

activations, the maximum value is selected for use as a single value going forward (not the mean).

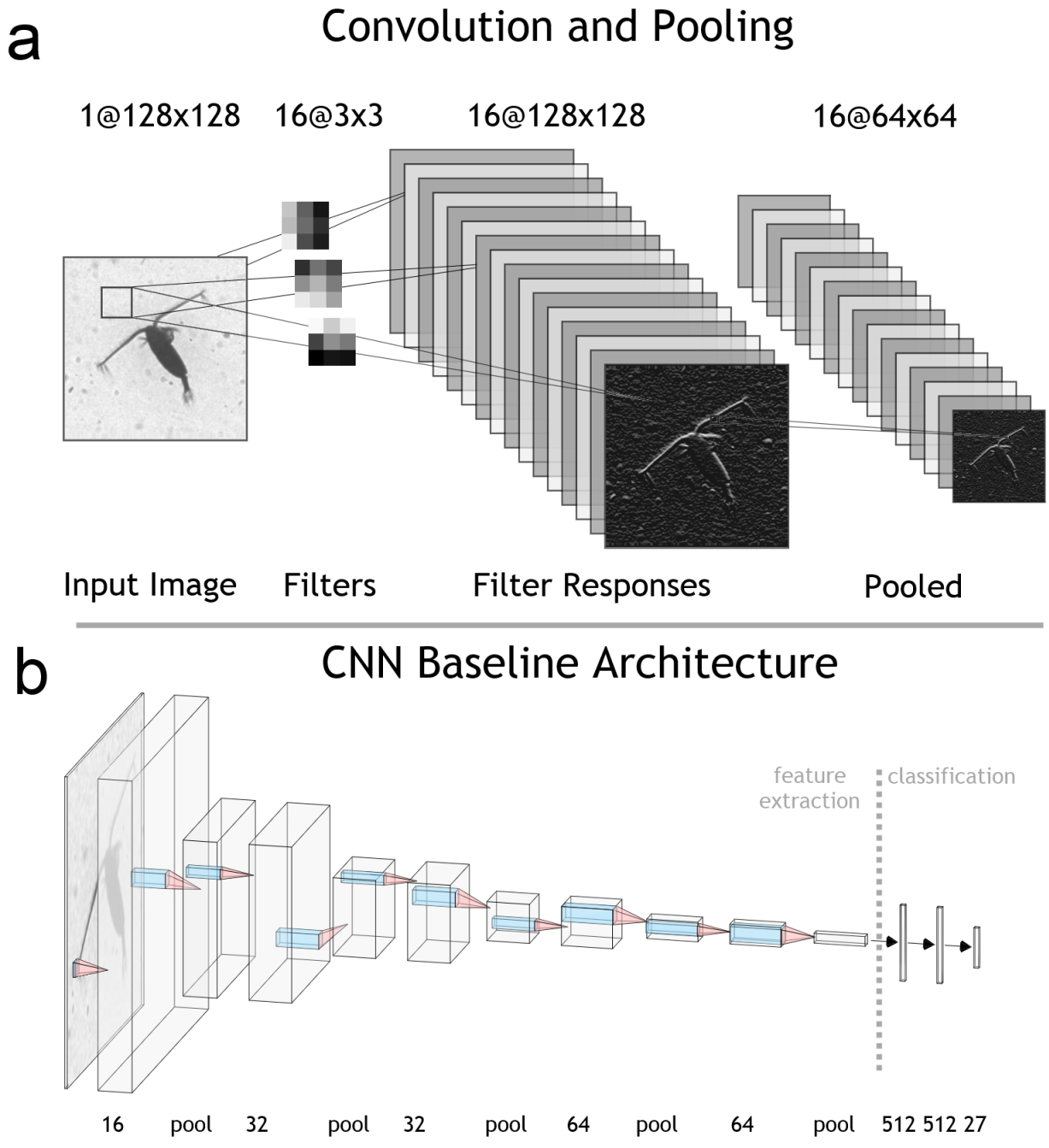


Figure 6.4: Our CNN architecture. (a) Illustration of the first convolution and pooling layers. Our input images are 128x128. Each of the 16 3x3 filters is convolved against the input, resulting in an activation volume of 16x128x128. A 2x2 max pooling layer scales the image by 50%. (b) Our baseline architecture has five convolutional layers with 16, 32, 32, 64, and 64 filters, all are 3x3. A 2x2 pooling layer follows each filter

layer. After these 10 layers are two fully connected layers, each with 512 neurons before the final softmax length 27 vector corresponding to predicted classification.

One other key architectural detail in the VGG-16 and related models is the use of fully-connected layers of neurons prior to the final softmax layer that determines the classification. We reduced the size of these fully-connected layers to one eighth the size or more of that used in VGG-16, which increased accuracy and decreased training time by 50% or more.

Since convolutional layers are designed to operate on image pixels, there is no means to fuse metadata directly into the convolutional layers. One approach to incorporating additional context metadata is to concatenate metadata values to the penultimate network layer. Instead, we find better accuracy when we incorporate the features earlier into fully-connected layers, as illustrated schematically in figure 5. Our best model, which we call Metadata Interaction, allows some interaction between the features with the output of the final pooling layer.

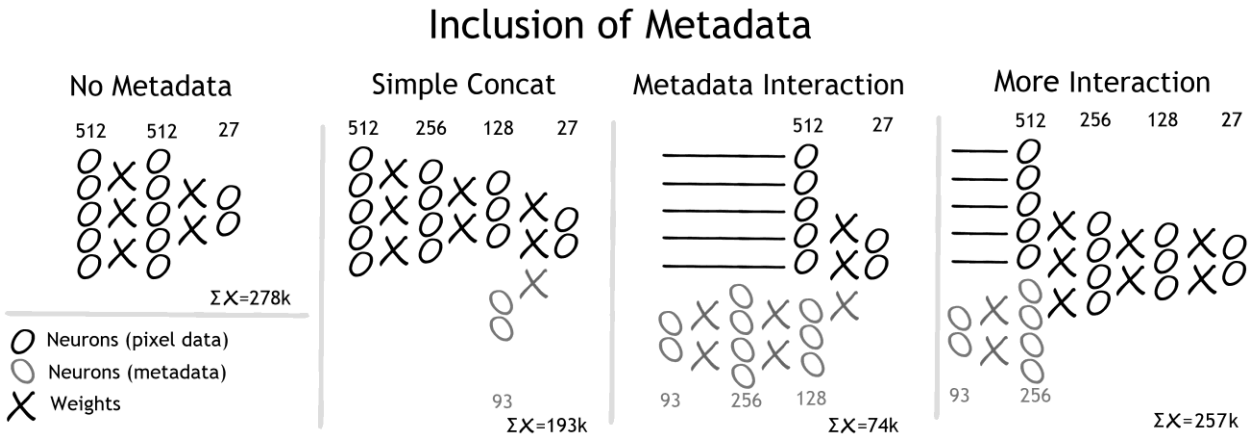


Figure 6.5: Schematic illustration of our baseline (left) and three architectures for metadata incorporation (Simple Concatenation, Metadata Interaction, and More Interaction). All convolutional layers precede illustrated alternatives, as illustrated in figure 4.

Figure 5 illustrates variations in the final fully connected layers, to the right of the dashed line labeled “classification” in figure 4. All four of these architectures have identical configurations of 5 convolutional and 5 pooling layers (Fig. 4b). In a fully connected layer, each

neuron's output is routed to every neuron's input in the subsequent layer, with a weight on each route. Therefore the number of weights applied to a fully connected layer's output is the product of the size of the layer and its successor. Our selected "No Metadata" architecture routes the convolutional layer's output to two consecutive layers of 512 neurons followed by a layer of 27 neurons, resulting in a total of ~278k weights. (Fig. 5 – convolutional layers not pictured that contain ~700k additional weights in an identical configuration for all pictured models). If we concatenated the vector of all 93 features to the penultimate layer, that CNN would have slightly more weights than the No Metadata option. Therefore, our simple concatenation model has smaller fully connected layers of 512, 256, 128, 27. After adding metadata, there are only ~193k weights, to ensure that any gain in accuracy must be from context metadata. Our Metadata Interaction model is even more restricted. We use the same layer structure as in simple concatenation (256, 128) but route the metadata from the ROI through the multiple fully connected layers instead of the CNN extracted features, so the number of weights is significantly less than either (~74k weights). Alternatively, we route the metadata through a single layer, combine them with the extracted features, and use two more fully connected layers for a total of 257k weights. These are the largest numbers, 193k , 74k, and 257k, corresponding to the usage of all 93 context metadata features.

Dropout (Hinton et al. 2012) acts as "a stochastic regularization technique" (Srivastava et al. 2014). Dropout is the concept of randomly ignoring the output of some neurons in the network in order to strengthen the rest of the network, and in most cases is beneficial. We assess the impact of dropout on both pixel data and on context metadata.

6.3.8 Performance metrics

We report binary accuracy for each of our models, where full credit is given for each correctly classified image and none for incorrect classifications, regardless of class of origin. A confusion matrix is used to interpret class-specific distribution of true/false positives and negatives. Timing information, when provided, is for single-threaded computations for training and testing a single replicate of the data. It does not include the time to load the dataset into memory. Our boxplots display whiskers equal to 1.5 times the inner quartile range, with the results of individual trials overlain as circles to indicate the distribution of the trained models. The number of trials was often as few as 10 (1 for each replicate) and rarely more than 20.

6.4 Assessment

6.4.1 Feature-based algorithm assessment

We assessed a range of different hyperparameters in order to select values that provide the best overall performance of each of the five feature-based algorithms we evaluated (Fig. 6.6). Heatmaps show averages at each combination of the two most influential hyperparameters we assessed (Fig. 6.6, left column). By examining the variance across all trials for the most important hyperparameter value (Fig. 6.6, right column) we are able to determine that no further search is warranted. The leftmost bar in each right column panel in figure 6.6 contains a suboptimal combination, the middle bar shows the results from adjusting the key hyperparameter by a single increment, and the rightmost bar shows the results from one additional increment. The rightmost values show little accuracy gains, but all have significantly higher computational cost. For example, time to convergence for SVM on a single replicate of our medium data set with regularization strengths of 100, 1000, and 10000 resulted in convergence times of 2, 6, and 34 hours, respectively. Therefore the middle bar represents the selected hyperparameter

combination for all further assessments. The hyperparameters with the most impact on our assessment for the three Random Forest based algorithms are the same (RFC, XRT, and GBC, Figs. 6.6a-f). They each have a constraint on the maximum size of the forest constructed (number of estimators) and a limit on the number of features considered in each tree/stump (maximum depth). For SVMs, best practices are to perform a grid search over the kernel coefficient for the decision boundary (gamma) and the penalty parameter that determines the strength of the error term (Fig. 6.6i-j). Both are recommended to be evaluated in geometric/exponential increments (Hsu et al. 2003). Our MLP uses the Adam optimization algorithm and the rectified linear unit activation function, which is the default parameter, and also the same as our CNN architecture. The MLP's shape is determined by the hyperparameters of the number of "hidden" layers of neurons, and the number of neurons in each hidden layer (a hidden layer is between the input and output layers). We assessed two different network shapes; one with two equally sized layers (rectangle) and one with three layers, each half the size of the preceding layer (triangle). Our second hyperparameter is the width of the base layer (Fig. 6.6g-h)

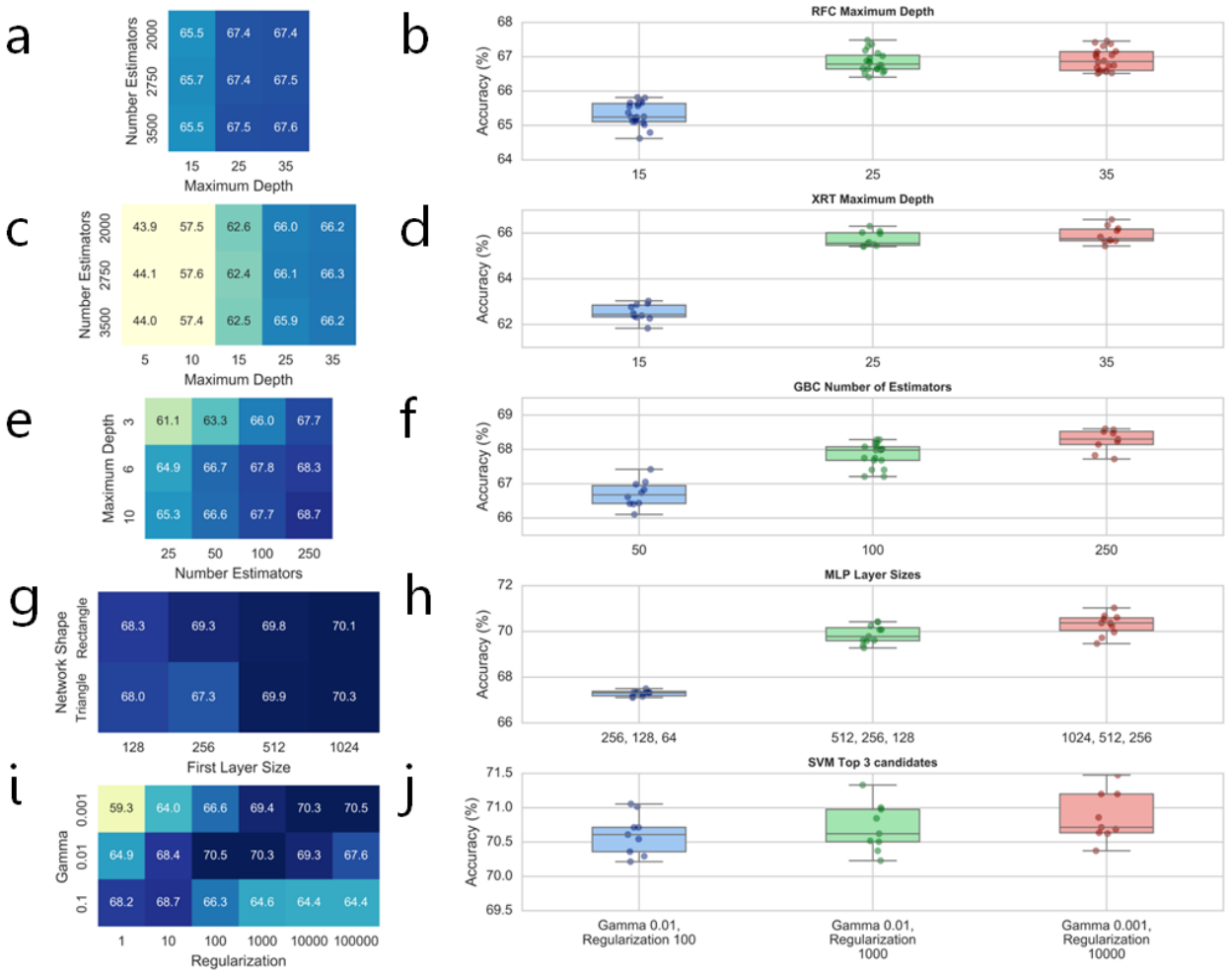


Figure 6.6: Hyperparameter grid search results for 5 different feature based machine learning classification methods (a,b – RFC, c,d – XRT, e,f – GBC, g,h – MLP, i,j – SVM). Cells in left column contain average results across all trials for a given hyperparameter combination. Boxplots in right column show all results for each hyperparameter combination across one key region of the grid search, and illustrate the variance within that configuration.

Our evaluation of the effect of data set size on classification accuracy of the feature-based algorithms showed that the largest data set consistently provided the best results (Fig. 7). Our medium data set contains ~3x more training images than the small but the large contains ~14x more training images than the small, therefore the increase in accuracy from our small to medium to large data set is less than linear with respect to the number of training images, suggesting we are approaching asymptotic performance.

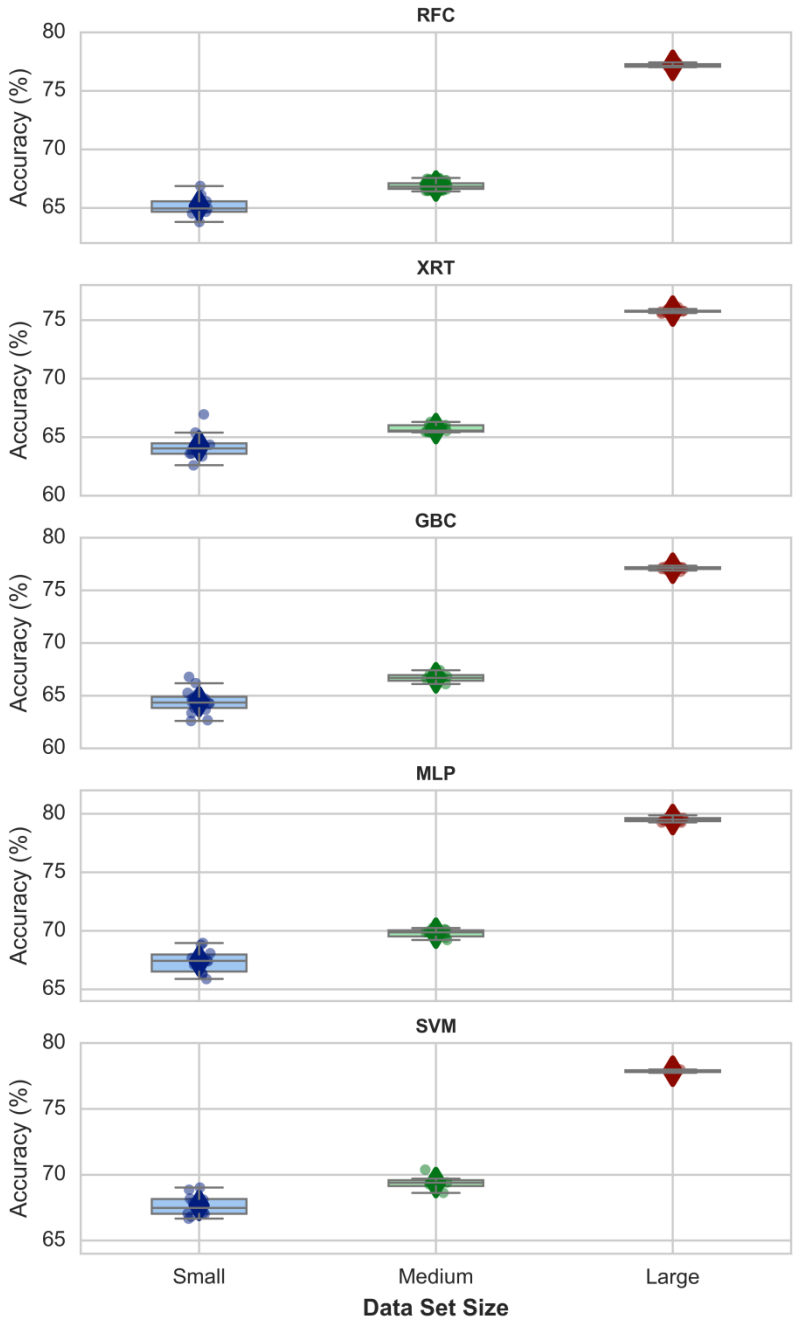


Figure 6.7: Accuracy vs Data Set size for 5 different feature based machine learning classification methods (RFC, XRT, GBC, MLP, SVM). The small data set contains ~25k images, the medium data set contains ~76k images, the large data set contains 350k images. All sets have 27 classes.

Having optimized hyperparameters and data set size, we now turn to metadata. Inclusion of context metadata significantly boosts performance for all five feature-based algorithms (Fig.

8). For algorithms assessed on the medium data set, we find gains of 6.9 to 12.2 percentage points. This gain is similar to the benefits of using the large set (Fig. 7). Geotemporal and hydrographic metadata have approximately the same influence, and inclusion of both results in the best overall classification accuracy.

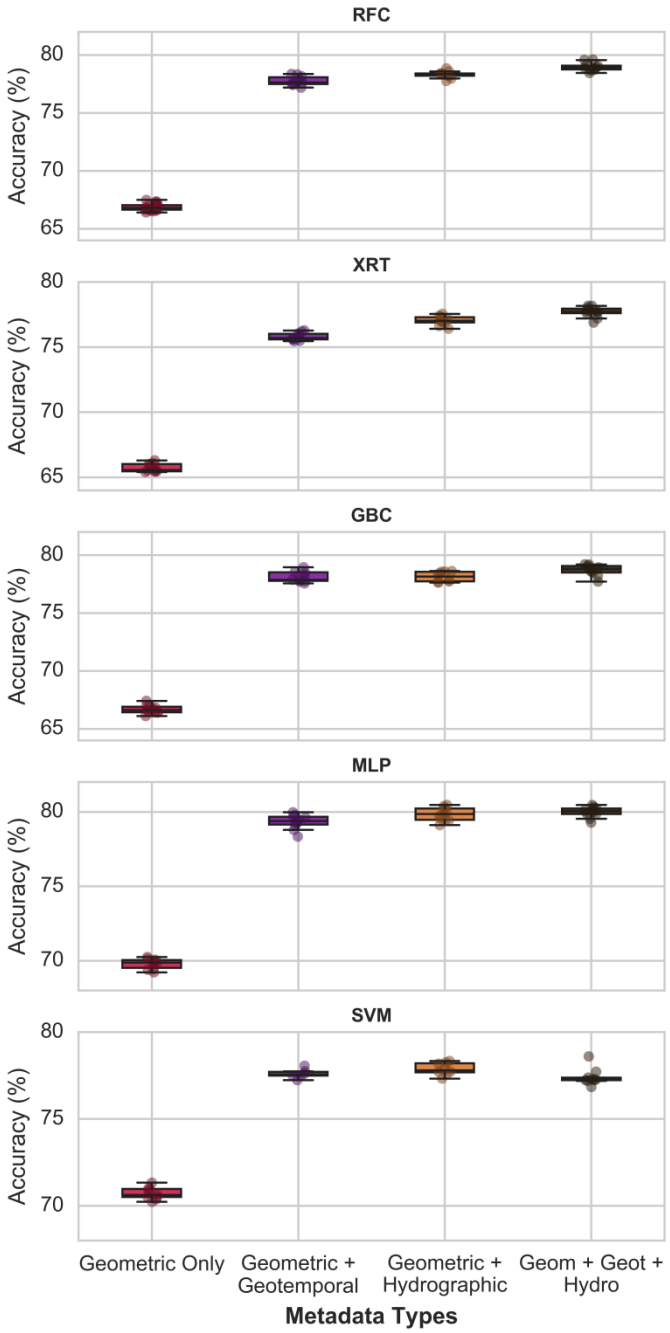


Figure 6.8: The effect of metadata on classification accuracy for 5 different feature-based machine learning classification methods (RFC, XRT, GBC, MLP, SVM) on our medium data set. The leftmost bar in each graph corresponds to a model using only the 58 geometric features, the next bar adds 22 geotemporal features, the next bar uses the 58 geometric features plus 13 additional hydrographic features. The rightmost bar utilizes all 93 features.

6.4.2 Convolutional Neural Network Assessment

CNNs have more hyperparameters that dramatically affect performance, so more preliminary investigation is required. Two that have the largest impact on performance are learning rate and regularization strength. We found the effect of learning rate to be much stronger than that of regularization, but both have a local maximum at regularization = 0.0001 (Fig. 9). As our CNN architecture matured, we revisited this assessment, but setting both values to 0.0001 remained optimal for our data. All subsequent figures use this value.

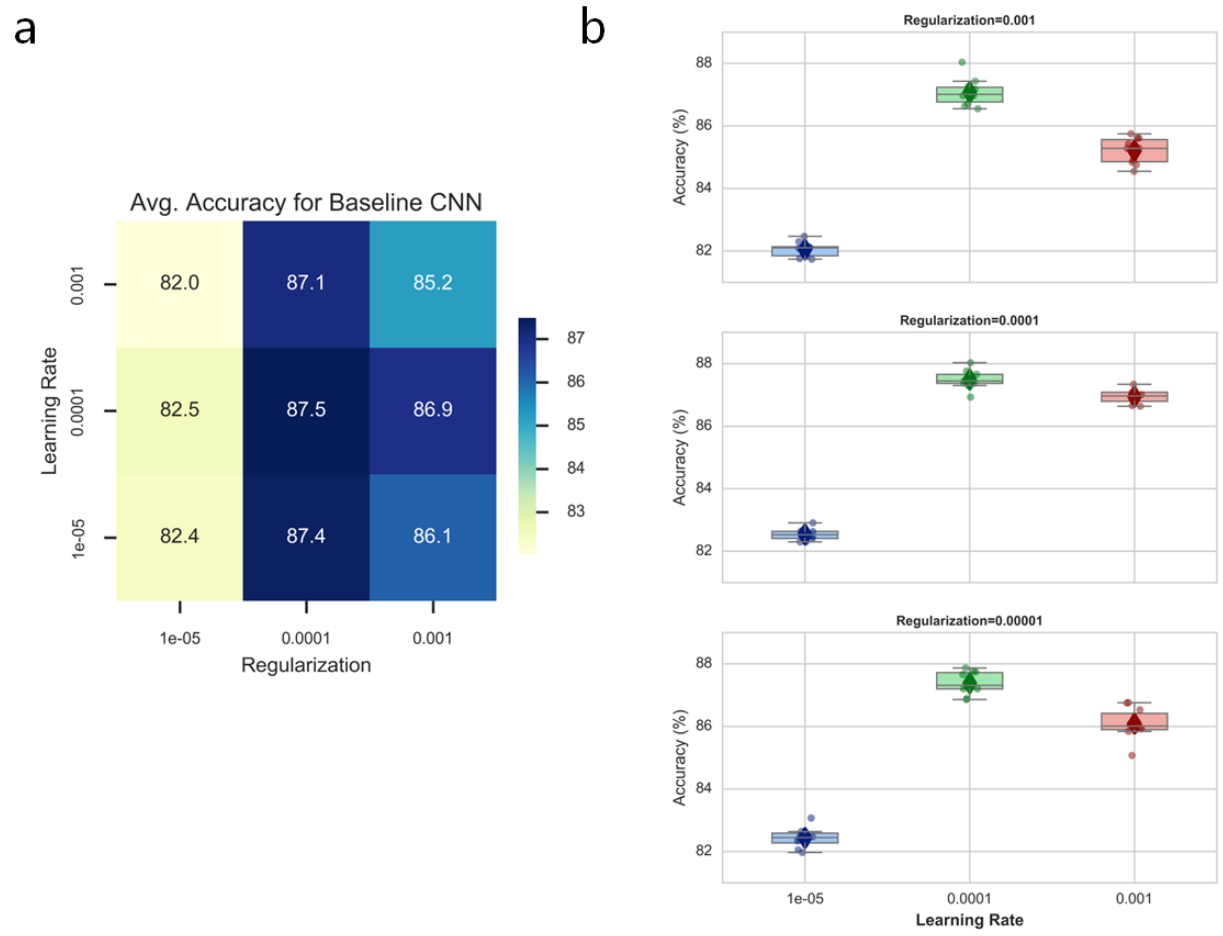


Figure 6.9: Hyperparameter optimization for CNN. (a) Heatmap cells contain average accuracy across all trials for a given combination of hyperparameters. (b) Boxplots show the distribution of results for each hyperparameter combination in the heatmap. All trials use medium data set size.

Augmentation strategies of horizontal and vertical image reflection have a stronger impact on performance with our medium than with our larger data set (Fig. 10). Our implementation used a 50% chance at runtime for each reflection operation on each image in each epoch, thus there was no additional computational demand, so we used this augmentation on all subsequent figures.

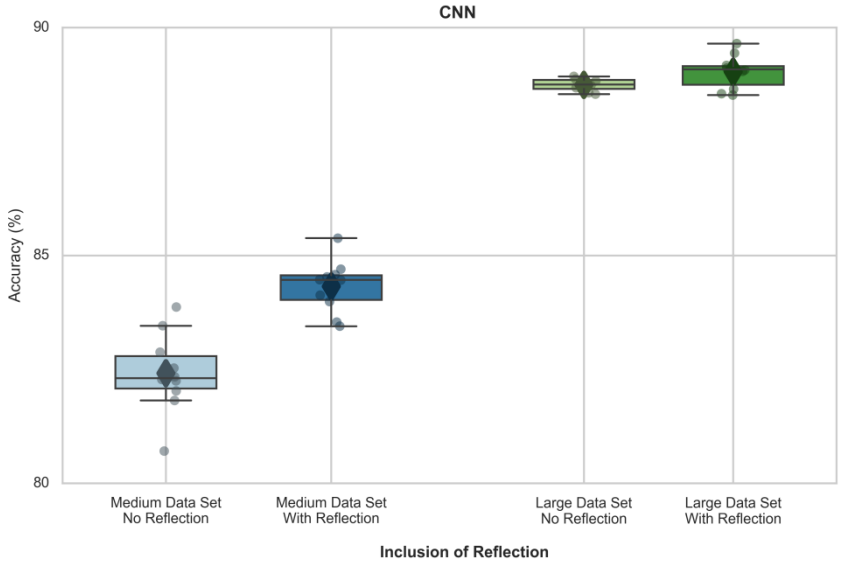


Figure 6.10: The effect on classification accuracy of using reflection as a runtime augmentation with our baseline CNN architecture, with (left) medium and (right) large data sets.

We evaluated the impact of dropout by incrementing the probability that any particular neuron will have its output ignored. We found a nearly monotonic association between dropout probability and accuracy on our medium data set (Fig. 11a) but a negligible effect with our large data set (Fig. 11b). Since using dropout still provides better results, we use it for the remainder of our assessments. Our finding of limited influence of dropout with larger datasets is notable because of the widespread use of dropout (Srivastava et al. 2014). Figure 11 reports results when we applied the dropout probabilities on only the fully connected layers of neurons.

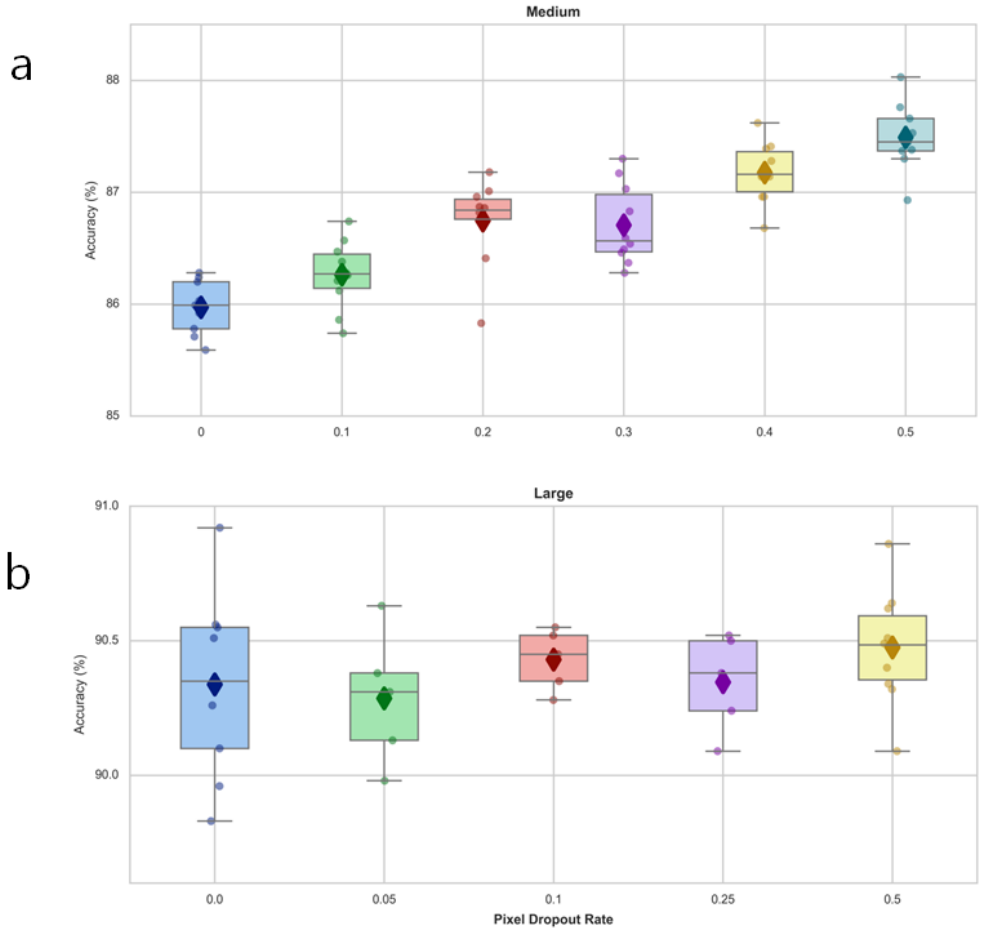


Figure 6.11: The effect on classification accuracy of using dropout with our baseline CNN architecture for (a) the medium data set and (b) the large data set. X-axis indicates the dropout probability.

We assessed numerous network configurations before arriving at our selected baseline method. This baseline method performed as well or better than the other alternatives we evaluated (Fig. 12). Our baseline model (Fig. 4), with 5 convolutional layers (16 filter convolutional layer, pooling layer, 32, pool, 32, pool, 64, pool, 64, pool) had an improvement in accuracy of 1.5 points over a similar model with 3 convolutional layers (16, pool, 32, pool, 64, pool). Models with dropout applied to the convolutional layers, and with narrower fully connected layers (512, 256, 128, 27) had lower accuracy than our selected baseline (Fig. 5 - 512, 512, 27). Larger filters (5x5 instead of 3x3) provided no significant difference in accuracy, but

the model with larger filters requires double the amount of memory as well as ~2x time as much time to train the network. Figure 12 also indicates that our networks are well tuned and not undersized since they increase in accuracy when given additional training examples.

Accuracy from our CNN is markedly better than all of the feature-based approaches: the accuracy of our baseline CNN on even our smallest dataset (25k ROI; ~1k per class) exceeds accuracy of each of the feature-based classifier accuracies on the largest dataset (350k ROI; max. 5k per class). Our CNNs exhibited a nearly linear relationship between convergence time and number of training examples, i.e., 1-2 hours per trial on our small data set and 8-12 hours per trial on our largest data set.

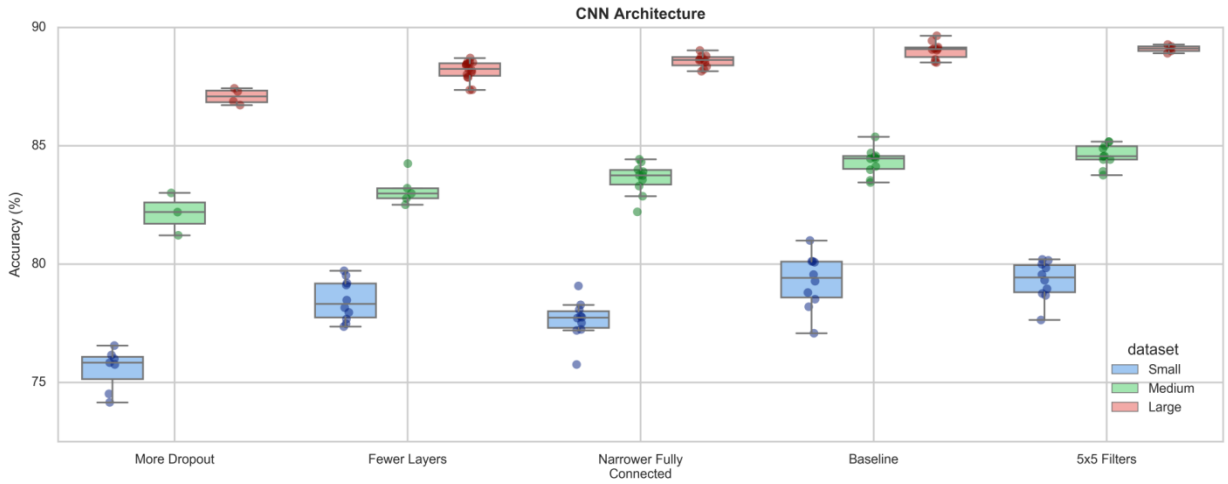


Figure 6.12: The effects of CNN architecture (i.e., changes in dropout, number of layers, connectivity, and filter size) relative to our baseline architecture (4th column). All results use pixel droput and reflection.

Having selected our baseline CNN, we then analyzed the effect of augmenting the pixel information with context metadata (Fig. 13). Both geotemporal and hydrographic context metadata individually make a significant improvement on classification accuracy ($p<0.001$; Fig. 13).

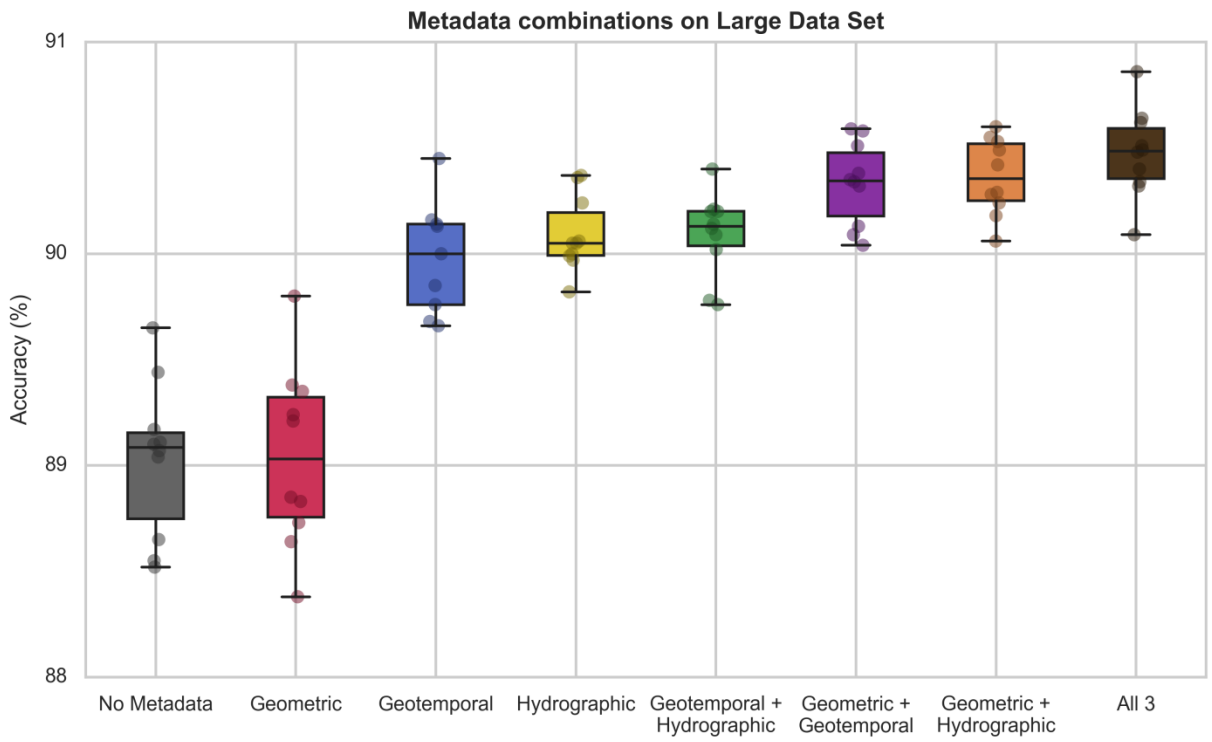


Figure 6.13: The effects on classification accuracy of adding context metadata. Experiments include no metadata and the contribution of every combination of geometric, geotemporal, and hydrographic metadata.

However, the combination of both geotemporal and hydrographic metadata yields a classification accuracy similar to each of them individually, potentially indicating overlap or redundancy between the features. Combining each individually with the geometric metadata provides a boost in performance, and using all three metadata types provides still more accuracy gain, to 90.5% accuracy. Hence, our remaining analysis will be conducted utilizing all 93 features (Table 2).

Table 6.2: Three different types of context metadata (Geometric, Geotemporal, and Hydrographic).

Geometric (58)	Geotemporal (22)	Hydrographic (13)
Size and Shape Parameters *Area Parameters* (7) Area (Filled Area), Area Excluding Holes, Convex Hull Area, Equivalent Circular Diameter, Skeletal Area, Area Excluding Holes / Area Filled Ratio, Extent (Area / Bounding Box Ratio) *Perimeter* (8) Perimeter (Filled), Perimeter (with Holes), Convex Hull Perimeter, Feret Diameter, Height, Width, Fractal Dimension, Orientation *Circularity* (12) Major Axis Length, Minor Axis Length, Circularity (Filled), Circularity (with Holes), Elongation, Eccentricity, Feret Diameter / Area Filled Ratio, Feret / Area Excluding Holes, Perimeter (Filled) / Feret Ratio, Perimeter (Filled) / Area Filled, Perimeter (Filled) / Area Excluding Holes Ratio, Perimeter (Filled) / Major Axis Length Ratio *Symmetry* (8) Centroid [X, Y], Weighted Centroid [X, Y] Centroid Distance, Centroid Distance/Area Excluded Ratio, Horizontal Symmetry, Vertical Symmetry *Grey Level Parameters* (14) Grey level normalized cumulative histogram statistics: [Slope, 1st Quartile, 2nd Quartile, 3rd Quartile], Intensity [Min, Mean, Max, Range, Std_Dev, Skew, Kurtosis], Intensity Mean Position (max-mean/range), Intensity Signal/Noise Ratio, Coefficient of variation of pixel intensity *Image Moments* (9) Central Moments	*Geographic Information* (8) Latitude, Longitude Bottom Depth Distance from [Shore, Pt. Conception, Santa Barbara Basin] Distance from closest shallow point (600m) Sampling Depth proxy (pressure in dbar) *Temporal Information* (14) Season (one-hot out of 4) Time of Day (Day/Night/Twilight status - one hot out of 8) PDO state * El Nino/La Nina state ** (as San Diego De-Trended Sea Level Anomaly) * PDO from http://research.jisao.washington.edu/pdo/PDO.latest.txt (Mantua et al. 1997) ** Anomaly from http://oceaninformatics.ucsd.edu/datazoo/data/ccelter/datasets?action=summary&id=153	*Measured* (6) Fluorescence, Salinity, Temperature, 200 kHz scattering volume (S_V) *, 1 MHz scattering volume (S_V) *, dB Difference ($S_{V\ 1\ MHz} - S_{V\ 1000\ kHz}$) Derived (7) Rho (density), Upwelling index at 33N 119W **, Distance to nearest neighbor ROI *** (in frame, up to 5 nearest, in mm) * Returns between 3 and 8.1m from the transducers were averaged into 1m depth bins (Ohman et al. 2018) ** Upwelling from https://www.pfeg.noaa.gov/products/PFEL/modeled/indices/upwelling/NA/data_download.html *** Calculated after full-frame images are segmented

We find that the manner in which metadata are incorporated affects accuracy. We obtained better accuracy when we incorporate the features earlier into fully-connected layers (Fig 14). Our Simple Concatenation metadata model not only has smaller weights overall than our model without metadata (Fig 5 - 193k vs 278k), but specifically has smaller fully connected layers of 512, 256, 128, 27. Above we have shown (Fig. 12) that this configuration is less effective than a configuration with layers of 512, 512, 27, so all accuracy gained must be from the metadata inclusion. Because the metadata interaction requires more weights for the metadata, we remove the fully connected layers from the pixel based data entirely, providing evidence that

all improvement from the Metadata Interaction model over the Simple Concatenation model is from the metadata and interaction, not network size or shape.

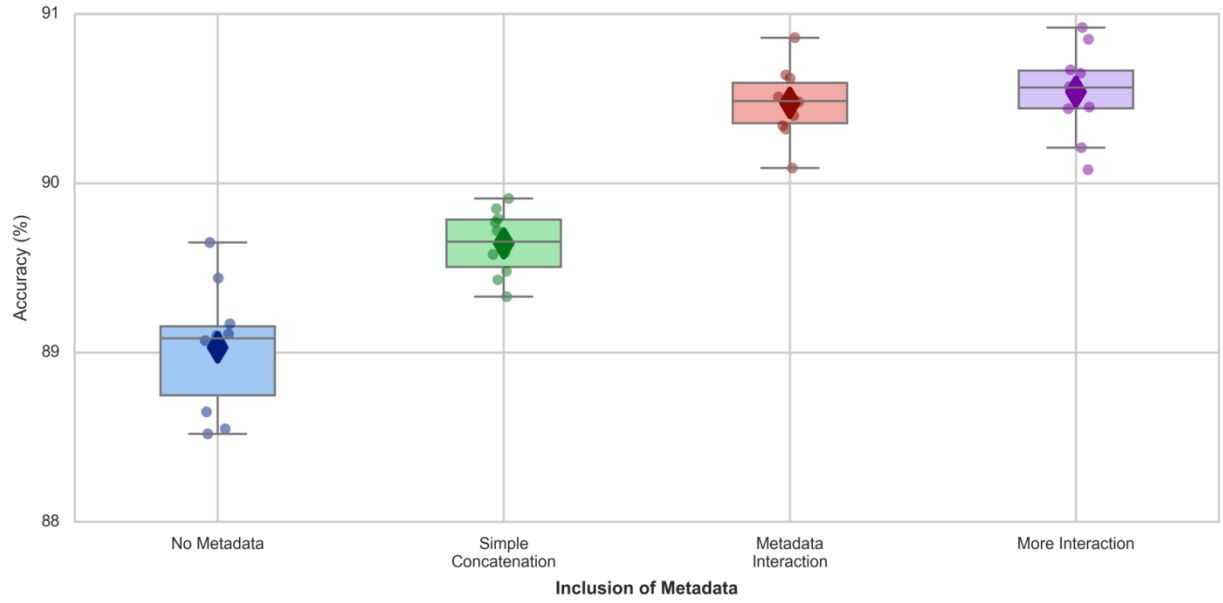


Figure 6.14: The effects on classification accuracy of different approaches to incorporating metadata (Simple Concatenation, Metadata Interaction, and More Interaction), for the large data set.

We found that applying dropout to the features derived from metadata is detrimental to accuracy (Fig. 15). Metadata dropout is detrimental even if pixel dropout is removed (Fig. 15a), especially at high dropout fractions. Metadata dropout is detrimental for the large set (Fig. 15b).

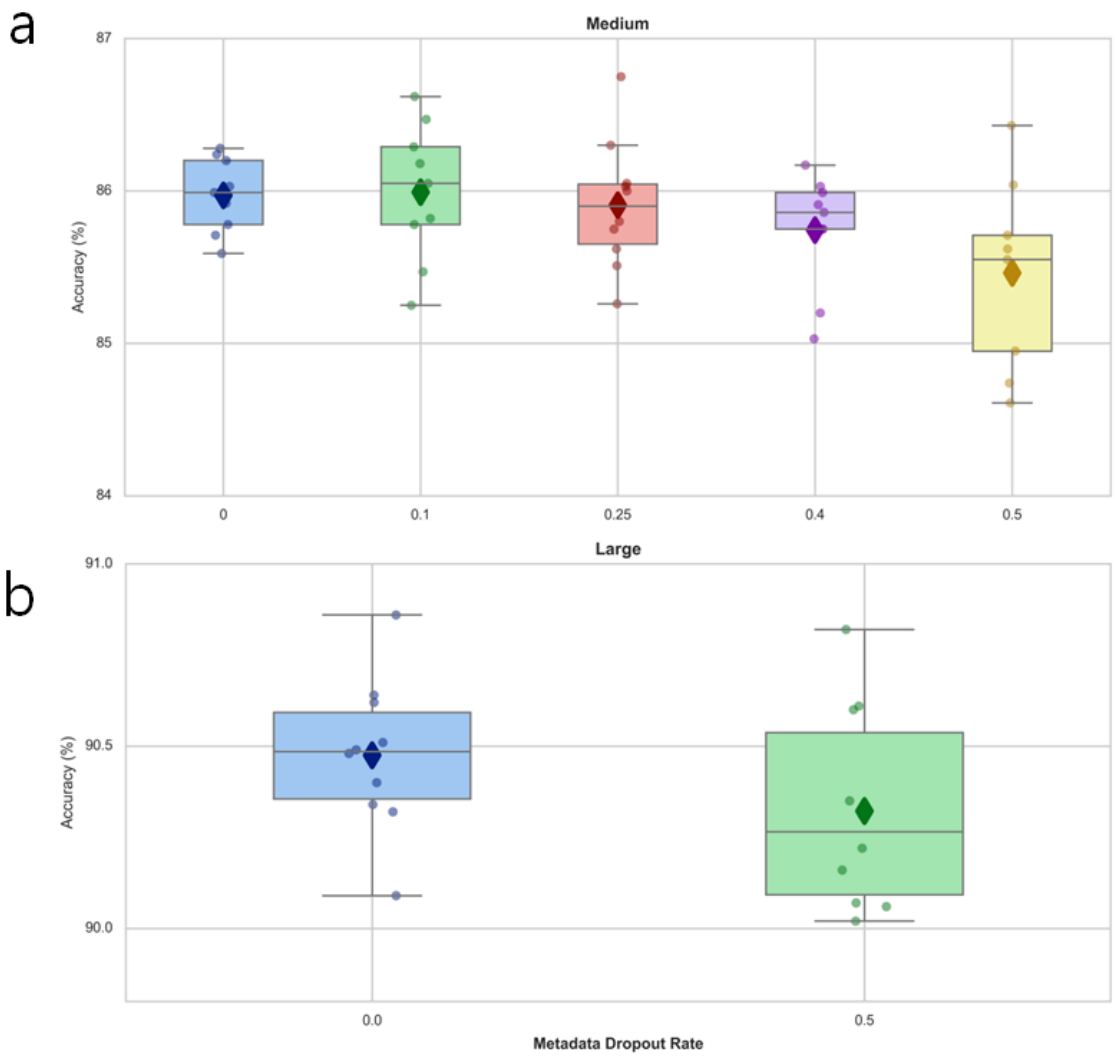


Figure 6.15: The effects on classification accuracy of including dropout with our CNN architecture, for (a) the medium data set and (b) the large data set. X-axis indicates the probability that an individual unit's value would be dropped.

We investigated more advanced CNN architectures to pursue additional accuracy (Fig. 14). Cyclic Pooling and Rolling (Dieleman et al. 2016b) have been shown to improve accuracy at the cost of much longer runtimes (5-8x longer). Our Metadata Interaction model provides almost as much benefit as Cyclic Pooling and Rolling (median 90.70% vs 90.40%). Doubling the number of filters in each layer results in a small performance gain (to 90.92%) at the cost of 50% longer runtimes. Doubling the number of layers instead results in a smaller performance gain at

the cost of 100% longer runtimes. Cycling Pooling and Rolling are still beneficial with metadata, and across different network sizes. Cyclic Pooling can be applied by itself, but Metadata Interaction plus Cyclic Pooling and Rolling is better, and this combination is significantly better than using Pooling and Rolling without metadata ($p<0.0001$). Doubling the number of filters is now not nearly as beneficial; this result makes sense if the additional filters were being devoted to learning rotations of other meaningful filters. Accordingly, doubling the number of layers and also augmenting with Cyclic Pooling and Rolling provides more of a gain than just doubling the number of filters. Our best model achieves 92.28 % accuracy with our 27-class dataset.

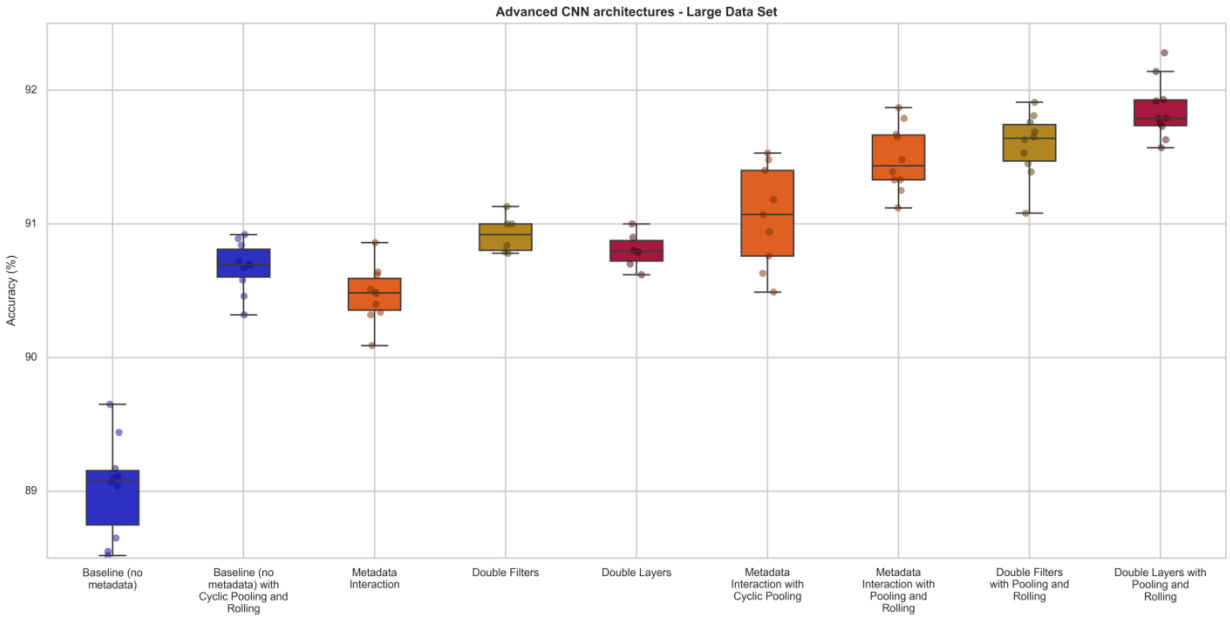


Figure 6.16: The effects on classification accuracy of advanced CNN architectures. See text for explanation.

The confusion matrix in figure 17a evaluates classwise performance of our best performing model, which includes context metadata added via Metadata Interaction, Cyclic Pooling, and Rolling (Fig. 17a). Our confusion matrix is shaded to prioritize true positive rate. For example, 21 of the 75 fish larvae in the test set were mislabeled as copepods, so that cell has a strong red shading, but the 111 snow mislabeled as copepods (corresponding to 0.06% out of

the 17,889 snow ROI in the test set) is essentially uncolored. Figure 17b illustrates the benefit of inclusion of metadata for particular classes, showing that most classes benefit. The 4 largest gains are found for nauplii, narcomedusae, euphausiids, and fish larvae, corresponding to some of the smallest classes. Prior to inclusion of metadata, some chaetognaths had previously been labeled as three other relatively thin and straight ROI classes: appendicularians, tentacles, and thread-like diatoms, while all three error types are minimized with the addition of the metadata.

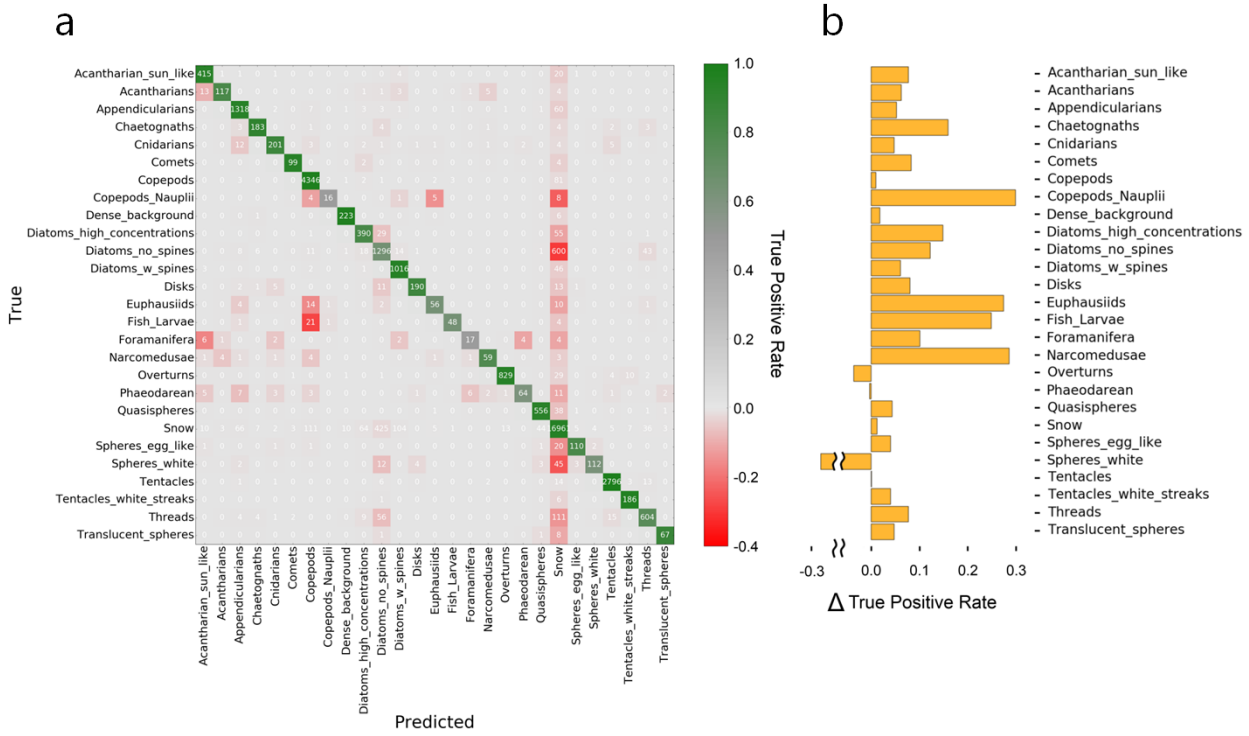


Figure 6.17: A confusion matrix indicating the specific errors made by our best performing model, which includes metadata interaction as well as cyclic pooling and rolling. (a) Rows indicate the true label, columns indicate the CNN algorithm's predicted label. Color intensity is proportional to the true positive rate. (b) Gains in classification accuracy from inclusion of metadata, for each category of organism.

6.5 Discussion

6.5.1 Impact of Context Metadata

We found that inclusion of context metadata provides gains in classification accuracy for both Convolutional Neural Networks (CNN) and feature-based classifiers. In the case of CNNs, the accuracy gain averaged 1.3 points, increasing the overall classification accuracy to 90.5% prior to enhancing CNN architecture. While the numerical increase is modest, the results were consistent across all replicates and represent a systematic improvement in overall accuracy, with appreciable gain in specific classes of images. Inclusion of metadata also improved CNN execution time, reducing convergence time by 17% (from 30.9 epochs to 25.6 epochs), likely because the metadata overrides ambiguous pixel features. In the case of feature-based classifiers, inclusion of metadata markedly increased classifier accuracy between 6.9 to 12.2 points, depending on the method considered.

Our estimate of the impact of adding metadata is likely conservative, because our feature-based models are possibly undersized for the metadata. Since models with and without metadata cannot be the same size while also being the same complexity, we favored the models without metadata. We initially tuned our models with 58 geometric measurements, held hyperparameters constant (e.g., number and depth of decision trees), but then added the geotemporal and hydrographic context metadata without retuning, nearly doubling the input to 93 features. Unlike with CNNs, metadata increased execution time for feature-based algorithms as much as 1.6x, proportional to the increase in the number of features (from 58 to 93). However, hyperparameter choices had 10-100x more impact on runtime than this increase. Overall, these are substantial gains that illustrate the clear advantage to incorporating context metadata across a variety of machine learning methods.

Although we divided metadata into three categories for illustrative purposes, they are all treated equally within our architecture, which includes them in the later layers of our CNN. Inclusion of multiple types of metadata will usually outperform a single type for two reasons. The first is due to the CNN architecture, where strong positive correlations outweigh neutral or negative correlations, so images benefitting from one type of metadata will usually not be harmed by the inclusion of additional metadata that are neutral or even slightly contradictory. The second reason is that our Metadata Interaction architecture allows for combination of features to impact the classification (e.g., a specific temperature value takes different meanings in winter vs. summer).

We assessed 12 different architectures for incorporating context metadata into our CNNs. The most naïve incorporation of metadata provided an accuracy gain of 0.6 points, less than half the benefit provided by our best architecture. Seven of our interaction architectures yielded nearly identical results at 0.8 points beyond that Simple Concatenation approach. We used efficiency of execution as a tiebreaker for designating our preferred Metadata Interaction model. However, an efficient network with a data set size of 350k could be undersized for larger data sets. Perhaps some of other architectures with additional fully-connected neurons processing context metadata would outperform the simpler architecture we presented.

6.5.2 Convolutional neural networks vs. feature-based algorithms

Prior to the development of Convolutional Neural Networks, plankton images were classified with varying degrees of success primarily using geometric features (reviewed in González et al. 2017). Recently, CNNs have been applied to plankton classification problems hinting at the potential of the approach (Wang et al. 2016; Zheng et al. 2017). A public competition (Robinson et al. 2017) stimulated new solutions (Dieleman et al. 2016b) but there

has not yet been a quantitative assessment of specific design choices when considering a CNN for plankton image analysis. Here we quantitatively evaluated CNN performance with a variety of augmentations and found that CNNs do consistently improve upon our applications of feature-based approaches as well as previous those of previous investigators (e.g., Hu and Davis 2005; Sosik and Olson 2007; Gorsky et al. 2010; Ellen et al. 2015)

On the smallest data sets, the computational requirements for CNNs exceed feature-based approaches, but as the size of the data set increases feature-based approaches require more resources because CNNs are influenced less by data set size. Since CNNs consider individual images sequentially, there is a linear relationship to data set size and number of images. Since feature-based algorithms generally consider the whole data set in the aggregate, they scale more steeply than linear with respect to data set size, with SVMs being more than quadratic (Cortes and Vapnik 1995; Pedregosa et al. 2011). Our results clearly show the benefit of larger data sets, although that benefit can only be realized if the algorithm can be successfully trained. Abstract algorithmic runtime analysis does not always hold in practice because there are confounding factors, such as whether individual calculations required by the algorithm are easily parallelizable. In practice, CNNs are also more tractable on larger data sets because GPUs have hundreds/thousands of cores well suited to the types of calculations that CNNs depend upon. For example, CNNs consider each image independently and the convolution operation with each filter is independent of the other filters in the layer. Therefore, many CNN calculations can be combined into a single multiplication of two large matrices to take advantage of the architecture of GPUs (Chetlur et al. 2014).

One disadvantage of CNNs is they currently lack direct interpretability (Zeiler and Fergus 2014). In contrast, statistics can be calculated on a trained RFC model about the relative

importance of individual features, and particular values of those features. In a CNN the first layer of filter weights can be rendered, but the interaction architecture of a CNN causes subsequent layers to lack a straightforward visualization, although this is an area of open research (Castelvecchi 2016).

6.5.3 Optimizing machine learning architectures for plankton classification

Both CNNs and feature-based algorithms require hyperparameter tuning for optimal performance. For our feature-based algorithms, we followed standard practices for hyperparameter optimization and found, as previously described in the literature, that attention to the number of estimators and depth for RFC-based approaches (Boulesteix et al. 2012), network size and activation function for MLPs (Haykin 2009), and gamma and regularization for SVMs (Hsu et al. 2003) improves performance. We found increasing the amount of training data improved accuracy. These two conclusions are consistent with our earlier investigation (Ellen et al. 2015).

We trained our CNNs *de novo*. In some applications of CNNs, starting with a pre-trained model could result in faster training times and increased accuracy, as demonstrated on phytoplankton images (Orenstein et al. 2015). In training our CNNs, we followed current good practices for CNNs (Bengio 2012; Smith 2018), although this guidance is evolving rapidly. Notably, we found that dropout (Hinton et al. 2012) produced little to no effect on our pixel-based data, and was even detrimental when applied to our context metadata. Both types of data set augmentation we evaluated were beneficial. Reflection increased accuracy by 0.34 points with no increase in runtime, while cyclic pooling and rolling (Dieleman et al. 2016b) increased accuracy by 1.6 points at a cost of a ~4x increase in execution time. Dieleman et al. (2015) first tried the concept of Cyclic Pooling and Rolling on a different class of rotationally invariant

images (galaxy morphology). An alternative method of obtaining rotational augmentations by Li et al. (2018) may be more efficient than the one we used from Lasagne (Dieleman et al. 2016a).

Since CNNs scale better with data set size than feature-based approaches, it is easier to consider more complicated and deeper architectures with them (i.e., Deep Learning). Published CNN benchmarks for image classification have increased from 19-layer networks, to an ensemble of seven separate 22-layer networks, to 152-layer networks, then 1000s of layers (Simonyan and Zisserman 2014; Szegedy et al. 2015; He et al. 2016). A particular model called ResNet (Szegedy et al. 2017) was applied to plankton by Li and Cui (2016) with modest results, which the authors suggest could be the result of insufficient training images. In limited trials we modified a version of ResNet (Szegedy et al. 2017) to fit our image dimensions with 24 layers, and it provided an increase of 0.8 points over our 5 layer Metadata Interaction model on our medium data set, at a cost of ~12x longer run time. We tried a 50-layer version of ResNet, and it performed worse than the 24-layer model (0.3 points lower, at a cost of ~1.25x longer run time). These preliminary results suggest the 50-layer network was overfitting, and the 24-layer network is closer to the optimal configuration for our images.

6.5.4 Metadata limitations

Supervised Machine Learning algorithms depend on training data being representative of future samples. For plankton image classification, this guidance is applicable not only for the distribution of the sampled organisms (González et al. 2017), but also for any context metadata used. The term "concept drift" (Widmer and Kubat 1996; González et al. 2017) describes the condition when this future distribution is not stationary. Some of the metadata distributions will drift faster than the images of the individuals themselves, as the population level responses can lag the changes in context measurements. One additional concern is that metadata will be less

useful for conditions that are not well represented in the training set; for example, time of day is not informative if all samples are collected at night.

6.6 Comments and Recommendations

6.6.1 Recommendations

Training sets should, in most circumstances, reflect the proportional distribution of classes. The percentage of marine snow in our ROIs imaged in situ exceeds 90%, but our largest data set is only 50% marine snow. We conducted limited evaluations on more unbalanced data and found a small increase in overall performance, but most of that increase was due to higher accuracy on snow only. Accuracy on non-snow classes decreased slightly, while false positives in the snow category increased. We found this outcome less desirable than the situation shown in the confusion matrix above, where very few non-snow ROIs end up with the label of snow. Many options exist for penalty functions where different types of errors are assigned different costs to create different types of confusion matrices (Elkan 2001), which then further facilitates treatment of larger datasets.

We only present results where each trained model is used to label images independently, but in practice multiple models can be used simultaneously or sequentially. Combining multiple individual models in an attempt to achieve greater accuracy than any one on its own is called ensembling. Ensembling of feature-based models without metadata on plankton images can be beneficial (Ellen et al. 2015). The concept of ensembling is well accepted, as nearly every major machine learning competition is won by an ensemble of multiple models (Robinson et al. 2017). The dynamics of an ensemble make academic analysis difficult, because the efficacy of each model needs to be examined as well as the effects of interactions between them, but evidence supports their implementation.

Most of our workflow would remain the same regardless of data set size with one exception. Small data sets with low performing models may learn so slowly or erratically as to never finish training. We set a hard limit on the number of epochs as a precaution against incurring computing costs on poorly performing models. Our 40 epoch limit was reached on ~20% of small data set trials, ~8% of medium data set trials, and ~2.5% of large data set trials. If we were doing more exhaustive investigation on smaller data sets we would resume training the model from the 40^{th} epoch for those trials.

6.6.2 Comments

Our CNNs are significantly smaller with a larger number of training examples than other contemporary evaluations of CNNs with plankton images (Dieleman et al. 2016b; Wang et al. 2016; Zheng et al. 2017; Moniruzzaman et al. 2017). The overall quality, resolution and between-class distinctiveness of our images is similar to previous studies. Based on previous publications, we did not expect our models to perform as well as they did with so few layers and filters. Some preliminary results suggest our networks with 10 convolutional layers are approaching asymptotic accuracy with respect to CNN complexity.

We outlined our calibration process for both feature-based approaches and CNNs, and found that in many situations accepted practices do hold (e.g., the importance of hyperparameters for feature-based approach hyperparameters and augmentation for CNNs) but we did find the benefit of dropout to be less significant than previously observed.

We found geometric, geotemporal, and hydrographic metadata to be useful for classification our in-situ images for both feature-based and CNN approaches. We found the context metadata to be useful not only as a straightforward augmentation at the end of the CNN

architecture, but found other incorporation strategies to be twice as beneficial for accuracy, in addition to being more computationally efficient.

Convolutional Neural Networks are rapidly evolving, with repeated layer substructures (e.g., ResNet), optimization functions , and ensembling techniques as three prominent research areas that will likely boost performance beyond our current results. The four factors that we found to provide the most benefit (data set size, appropriate network depth, data set augmentation, and inclusion of context metadata) were generally additive. We anticipate further advances by optimizing these four factors while also incorporating future structural refinements of deep learning methods.

6.7 Acknowledgements

This research was possible because of the support given by the Office of Naval Research and SPAWAR Systems Center San Diego via the SMART scholarship program. The Gordon and Betty Moore Foundation funded the development of the *Zooglider* that collected these images. The Scripps Institution of Oceanography's Instrument Development Group performed the engineering and fieldwork needed to develop and deploy the *Zooglider*. This research was conducted using the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-1548562 for machine learning experimentation. Most of the results were obtained using Comet from the San Diego Supercomputing Center and Bridges from the Pittsburg Supercomputing Center through XSEDE allocations TG-OCE150020 and TG-OCE160022. Machine Learning exploratory efforts and small scale experimentation used a Tesla K40 GPU donated by the NVIDIA Corporation. The National Science Foundation supported *California Current Ecosystem* Long Term Ecological Research (CCE-LTER) site also provided the support and expertise of Tristan Biard, Laura Lilly,

Catherine Nickels, Mark Ohman, Linsey Sala, Stephanie Sommer, Emma Tovar, and Ben Whitmore who conducted numerous hours of organism identifications and validations, without which these experiments would not have been possible.

Chapter 6 is being prepared for journal submission as: Ellen, Jeffrey S.; Graff, Casey A.; Elkan, Charles; Ohman, Mark D. "Improving plankton image classification using context features." It is presented as part of this dissertation with the acknowledgement of the study coauthors Casey A. Graff, Charles Elkan, and Mark. D. Ohman. The dissertation author was the primary investigator and is the primary author of this material.

6.8 References

Al-Rfou, R., G. Alain, A. Almahairi, and others. 2016. Theano: A Python framework for fast computation of mathematical expressions. arXiv preprint.

Beijbom, O., P.J. Edmunds, C. Roelfsema, and others. 2015. Towards automated annotation of benthic survey images: Variability of human experts and operational modes of automation. PloS one, 10:7-e0130312. doi: 10.1371/journal.pone.0130312

Benfield, M., C. Schwehm, R. Fredericks, G. Squyres, S. Keenan, and M. Trevorrow. 2003. ZOOVIS: A high-resolution digital still camera system for measurement of fine-scale zooplankton distributions. Scales in Aquatic Ecology: Measurement, Analysis and Simulation.

Bengio, Y. 2012. Practical recommendations for gradient-based training of deep architectures, p. 437-478. Neural networks: Tricks of the trade, LNCS vol 7700. doi:10.1007/978-3-642-35289-8_26

Blinn, Jim. 1998. Jim Blinn's corner: dirty pixels. 1st ed. Morgan Kaufmann.

Boulesteix, A.L., S. Janitza, J. Kruppa, and I.R. König, 2012. Overview of random forest methodology and practical guidance with emphasis on computational biology and bioinformatics. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 2, 6:493-507. doi:10.1002/widm.1072

Bradski, G. 2000. The OpenCV Library. Dr. Dobb's. The World of Software Development: 1

Briseño-Avena, C., P. L. D. Roberts, P. J. S. Franks, and J. S. Jaffe. 2015. ZOOPS- O 2: a broadband echosounder with coordinated stereo optical imaging for observing plankton in situ. Meth. Oceanogr. 12:36-54. doi: 10.1016/j.mio.2015.07.001

Canny, John. 1986. A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence 6: 679-698. doi:10.1109/TPAMI.1986.4767851

Chetlur, S., C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer. 2014. cuDNN: Efficient primitives for deep learning. arXiv preprint arXiv:1410.0759.

Cortes, C., and V. Vapnik. 1995. Support-vector networks. Machine learning 20, no. 3: 273-297.

Cowen, R. K., and C. M. Guigland. 2008. In situ ichthyoplankton imaging system (ISIIS): system design and preliminary results. Limnol. Oceanogr.-Meth. 6: 126-132.

Criminisi, A., Shotton, J., and Konukoglu, E. 2012. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and Trends in Computer Graphics and Vision 7, 2–3:81-227.

Dai, J., Wang, R., Zheng, H., Ji, G., & Qiao, X. 2016. ZooplanktoNet: Deep convolutional network for zooplankton classification. IEEE OCEANS 2016-Shanghai. 1-6.

Davis, C. S., S. M. Gallager, M. S. Berman, L. R. Haury, and J. R. Strickler. 1992. The video plankton recorder (VPR): design and initial results. Arch. Hydrobiol. Beih. Ergeb. Limnol. 36: 67-81.

Davis, Russ E., Mark D. Ohman, Daniel L. Rudnick, and Jeff T. Sherman. 2008. Glider surveillance of physics and biology in the southern California Current System. Limnology and Oceanography, 53. 5.2: 2151-2168.

Dieleman, Sander, Kyle W. Willett, and Joni Dambre. 2015. Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly notices of the royal astronomical society 450. 2: 1441-1459.

Dieleman, Sander, Jan Schlter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, Daniel Maturana et al. 2016. Lasagne: First release. doi:10.5281/zenodo 27878

Dieleman, Sander, Jeffrey De Fauw, and Koray Kavukcuoglu. 2016. Exploiting cyclic symmetry in convolutional neural networks. arXiv preprint arXiv:1602.02660. Accessed 09-2018.

Elkan, C. 2001. The foundations of cost-sensitive learning. International Joint Conference on Artificial Intelligence 17. 1: 973-978.

Ellen, J., Li, H. and Ohman, M.D. 2015. Quantifying california current plankton samples with efficient machine learning techniques. OCEANS'15 MTS/IEEE Washington. 1-9. IEEE. doi: 10.23919/oceans.2015.7404607

Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. and Thrun, S. 2017. Dermatologist-level classification of skin cancer with deep neural networks. Nature, 542. 7639: 115-118.

Fernandes, F. 2014. Python-seawater v3.3.2 (Version v3.3.2). Zenodo. http://doi.org/10.5281/zenodo.11395

Fofonoff, P. and Millard, R.C. Jr. 1983. Algorithms for computation of fundamental properties of seawater. UNESCO Technical Papers in Marine Science. 44:1-53. http://unesdoc.unesco.org/images/0005/000598/059832eb.pdf

Freund, Yoav, and Robert E. Schapire. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Journal of computer and system sciences 55. 1: 119-139.

Friedman, Jerome H. 2001. Greedy function approximation: a gradient boosting machine. Annals of statistics. 1189-1232.

Geurts, Pierre, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. Machine learning 63, 1: 3-42.

Glorot, Xavier, and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. Proceedings of the thirteenth international conference on artificial intelligence and statistics. 249-256.

González, P. , Álvarez, E. , Díez, J. , López-Urrutia, Á. and del Coz, J. J. 2017. Validation methods for plankton image classification systems. Limnolology and Oceanography Methods, 15: 221-237. doi: 10.1002/lom3.10151

Gorsky, G., Ohman, M. D. , Picheral, M., Gasparini, S., Stemmann, L., Romagnan, J.-B., Cawood, A., Pesant, S., Garcıa-Comas, C. , and Prejger, F. 2010. Digital zooplankton image analysis using the ZooScan integrated system. Journal of Plankton Research 32. 3: 285–303.

Graff, C.A. and J. Ellen. 2016. Correlating filter diversity with convolutional neural network accuracy. Machine Learning and Applications (ICMLA), 15th IEEE International Conference on, 75-80. doi: 10.1109/ICMLA.2016.0021

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

Grosjean, P., M. Picheral, C. Warembourg, and G. Gorsky. 2004. Enumeration, measurement, and identification of net zooplankton samples using the ZooScan digital imaging system. ICES Journal of Marine Science, 61(4): 518-525.

Haykin, S.S. 2009. Neural networks and learning machines (Vol. 3). Upper Saddle River, NJ, USA: Pearson.

He, K., X. Zhang, S. Ren, and J. Sun. 2015. Delving deep into rectifiers: surpassing human-level performance on imagenet classification. Computer Vision, IEEE international conference on, 1026-1034.

He, K., X. Zhang, S. Ren, and J. Sun. 2016. Deep residual learning for image recognition. Computer Vision and Pattern Recognition, IEEE international conference on, 770-778.

Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Ho, T. K. 1995. Random decision forests. Document analysis and recognition, proceedings of the third international conference on, 1:278-282.

Hsu, C. W., C.-C. Chang, and C.-J. Lin. 2003. A practical guide to support vector classification. 1-16.

Hu, Q, and C. Davis. 2005. Automatic plankton image recognition with co-occurrence matrices and support vector machine. Marine Ecology Progress Series 295: 21-31

Hubel, D. H. 1959. Single unit activity in striate cortex of unrestrained cats. The Journal of physiology 147, 2:226-238.

Hubel, D. H., and T. N. Wiesel. 1963. Shape and arrangement of columns in cat's striate cortex. The Journal of physiology 165, 3:559-568.

Kanellopoulos, I., and G. G. Wilkinson. 1997. Strategies and best practice for neural network image classification. International Journal of Remote Sensing 18, 4:711-725.

Khotanzad, A, and J-H. Lu. 1990. Classification of invariant image representations using a neural network. IEEE Transactions on Acoustics, Speech, and Signal Processing 38, 6:1028-1038.

Kingma, D. P. and M. Welling. 2013. Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

Kingma, D. P. and L. Ba. 2015. J. ADAM: a method for stochastic optimization. International Conference on Learning Representations. 2015.

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

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

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

LeCun, Y., and M. Ranzato. 2013. Deep learning tutorial. Tutorials in International Conference on Machine Learning. doi:10.1.1.366.4088

LeCun, Y, Y. Bengio, and G. E. Hinton. 2015. Deep learning. Nature 521, 7553:436. doi:10.1038/nature14539

Lee, C.Y., S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. 2015. Deeply-supervised nets. Artificial Intelligence and Statistics. 562-570.

Li, J., Z. Yang, H. Liu, and D. Cai. 2018. Deep Rotation Equivariant Network. Neurocomputing 290: 26-33.

Li, Y., D. J. Crandall, and D. P. Huttenlocher. 2009. Landmark classification in large-scale image collections. International Conference on Computer Vision, 1957-1964.

Li, X., and Z. Cui. 2016. Deep residual networks for plankton classification. OCEANS 2016 MTS/IEEE Monterey, 1-4.

Lilly, L. E. and M. D. Ohman. 2018. CCE IV: El Niño-related zooplankton variability in the southern California Current System. Deep Sea Research Part I: Oceanographic Research Papers.

Lippmann, R. 1987. An introduction to computing with neural nets. IEEE Assp magazine 4, 2: 4-22.

Loshchilov, I., and F. Hutter. 2017. Sgdr: Stochastic gradient descent with warm restarts. International Conference on Learning Representations 2017.

Mantua, N. J., S. R. Hare, Y. Zhang, J. M. Wallace, and R. C. Francis. 1997. A Pacific interdecadal climate oscillation with impacts on salmon production. Bulletin of the american Meteorological Society, 78. 6:1069-1080. (Data accessed June 2018 via http://research.jisao.washington.edu/pdo/PDO.latest.txt)

Matsugu, M., Mori, K., Mitari, Y. and Kaneda, Y., 2003. Subject independent facial expression recognition with robust face detection using a convolutional neural network. Neural Networks, 16, 5-6:555-559.

Medwin, H., and C. S. Clay. 1998. Fundamentals of Acoustical Oceanography. Academic Press, Boston.

Moniruzzaman, M., S. M. S. Islam, M. Bennamoun, and P. Lavery. 2017. Deep Learning on Underwater Marine Object Detection: A Survey. International Conference on Advanced Concepts for Intelligent Vision Systems. 150-160.

Ng, J. Y.-H., M. Hausknecht, S. Vijayanarasimhan, S., O. Vinyals, R. Monga, and G. Toderici. 2015. Beyond short snippets: Deep networks for video classification. Computer Vision and Pattern Recognition, Proceedings of the IEEE conference on. 4694-4702.

Nilsback, M.-E., and A. Zisserman. 2008. Automated flower classification over a large number of classes. Computer Vision, Graphics & Image Processing, Sixth Indian Conference on.

NOAA National Center for Environmental Information. 2016. San Diego, California Coastal Digital Elevation Model – Issued 2012-03-07, updated 2016-07-28. (Data accessed June 2018 via https://www.ngdc.noaa.gov/thredds/dodsC/regional/san_diego_13_mhw_2012.nc.html)

Ohman, M. D., D. L. Rudnick, A. Chekalyuk, R. E. Davis, R. A. Feely, M. Kahru, H.-J. Kim et al. 2013. Autonomous ocean measurements in the California Current Ecosystem. Oceanography 26. 3:18-25.

Ohman M. D., R. E. Davis, J. T. Sherman, K. R. Grindley, B. M. Whitmore, C. F. Nickels, J. S. Ellen. (in review). *Zooglider*: an autonomous vehicle for optical and acoustic sensing of zooplankton. Limnology and Oceanography: Methods

Olson, R. J., H. M. Sosik. 2007. A submersible imaging-in-flow instrument to analyze nano-and microplankton: Imaging FlowCytobot. Limnology and Oceanography: Methods 5, 6:195-203.

Orenstein, E.C., O. Beijbom, E. E. Peacock, and H. M. Sosik. 2015. Whoi-plankton-a large scale fine grained visual recognition benchmark dataset for plankton classification. The Third Workshop on Fine-Grained Visual Categorization at CVPR 2015. arXiv preprint arXiv:1510.00745.

Orenstein, E. C., and O. Beijbom. 2017. Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets. Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on. 1082-1088. doi:10.1109/WACV.2017.125

Parker-Stetter, S.L., L. G. Rudstam, P. J. Sullivan, D. M. Warner. 2009. Standard operating procedures for fisheries acoustic surveys in the Great Lakes. Great Lakes Fisheries Commission Special Publication.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel et al. 2011. Scikit-learn: Machine learning in Python. Journal of machine learning research 12: 2825-2830.

Peura, Markus, and J. Iivarinen. 1997. Efficiency of simple shape descriptors. Aspects of visual form. 443-451.

PFEL Upwelling Index: (Pacific Fisheries Environmental Laboratory), https://www.pfeg.noaa.gov/products/PFEL/modeled/indices/upwelling/NA/data_download.html, retrieved 2018.

Picheral, M., L. Guidi, L. Stemmann, D. M. Karl, G. Iddaoud, and G. Gorsky. 2010. The Underwater Vision Profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. Limnol. Oceanogr.-Meth. 8: 462-473 doi 10.4319/lom.2010.8.462

Robinson, K. L., J. Y. Luo, S. Sponaugle, C. Guigand, and R. K. Cowen. 2017. A tale of two crowds: Public engagement in plankton classification. Frontiers in Marine Science 4:82.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams. 1986. Learning representations by back-propagating errors. Nature 323, no. 6088:533.

Sabour, S., N. Frosst, and G. E. Hinton. 2017. Dynamic routing between capsules. Advances in Neural Information Processing Systems. 3856-3866.

Samson, S., T. Hopkins, A. Remsen, L. Langebrake, T. Sutton, and J. Patten. 2001. A system for high-resolution zooplankton imaging. IEEE J. Ocean. Engin. 26: 671-676 doi 10.1109/48.972110

Schulz, J., K. Barz, P. Ayon, A. Luedtke, O. Zielinski, D. Mengedoht, and H.-J. Hirche. 2010. Imaging of plankton specimens with the lightframe on-sight keyspecies investigation (LOKI) system. J. Euro. Opt. Soc.-Rapid Publ. 5: doi 10.2971/jeos.2010.10017s

Schwing, F. B., M. O'Farrell, J. M. Steger, and K. Baltz. 1996. Coastal Upwelling indices west coast of North America. NOAA Tech. Rep., NMFS SWFSC NMFS SWFSC 231: 144.

Sherman, J., R. E. Davis, W. B. Owens, and J. Valdes. 2001. The autonomous underwater glider *Spray*. IEEE Journal of Oceanic Engineering 26:437-446.

Sieracki, C. K., M. E. Sieracki, and C. S. Yentsch. 1998. An imaging-in-flow system for automated analysis of marine microplankton. Marine Ecology Progress Series 168:285-296

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

Smith, Leslie N. 2018. A disciplined approach to neural network hyper-parameters: Part 1--learning rate, batch size, momentum, and weight decay. arXiv preprint arXiv:1803.09820.

Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. Empirical methods in natural language processing, Proceedings of the 2013 conference on. 1631-1642.

Sosik, H. M. and R. J. Olson. 2007. Automated taxonomic classification of phytoplankton sampled with imaging-in-flow cytometry. Limnology and Oceanography: Methods, vol. 5, 6:204–216.

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

Szegedy, C., W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. 2015. Going deeper with convolutions. Computer Vision and Pattern Recognition, Proceedings of the IEEE conference on. 1-9.

Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning." AAAI Conference on Artificial Intelligence, Proceedings of the Thirty-First. 4:4278-4284.

Tang, K., M. Paluri, F. F. Li, R. Fergus, and L. Bourdev. 2015. Improving Image Classification with Location Context. Computer Vision, Proceedings of the IEEE International Conference on. 1008-1016.

Thompson, C. M., M. P. Hare, and S. M. Gallager. 2012. Semi-automated image analysis for the identification of bivalve larvae from a Cape Cod estuary. Limnol. Oceanogr.-Meth. 10: 538-554 doi 10.4319/lom.2012.10.538

Towns, J., T. Cockerill, M. Dahan, I. Foster, K. Gaither, A. Grimshaw, V. Hazlewood, S. Lathrop, D. Lifka, G. D. Peterson, R. Roskies, J. R. Scott, N. Wilkins-Diehr. 2014. XSEDE: Accelerating Scientific Discovery. Computing in Science & Engineering, vol.16, 5:62-74 doi:10.1109/MCSE.2014.80

van Rossum, G. 1995. Python tutorial, Technical Report CS-R9526. Centrum voor Wiskunde en Informatica (CWI), Amsterdam.

Wagemans, J., J. H. Elder, M. Kubovy, S. E. Palmer, M. A. Peterson, M Singh, and R. von der Heydt. 2012. A century of Gestalt psychology in visual perception: I. Perceptual grouping and figure–ground organization. Psychological bulletin 138, 6:1172-1217.

Wang, R., J. Dai, H. Zheng, G. Ji, and X. Qiao. 2016. Multi features combination for automated zooplankton classification. IEEE OCEANS 2016-Shanghai. 1-5.

Watson, J. 2004. HoloMar: A holographic camera for subsea imaging of plankton. Sea Technol. 45: 53-55.

Wertheimer, Max. 1923. Untersuchungen zur Lehre von der Gestalt II, in Psycologische Forschung, 4:301-350.

Widmer, G. and M. Kubat. 1996. Learning in the presence of concept drift and hidden contexts. Machine learning, 23, 1:69-101.

Wilkins, M. F., L. Boddy, C. W. Morris, and R. Jonker. 1996. A comparison of some neural and non-neural methods for identification of phytoplankton from flow cytomery data. Bioinformatics 12, 1:9-18.

Zeiler, M. D. and R. Fergus. 2014. Visualizing and understanding convolutional networks. European Conference on Computer Vision. 818-833.

Zheng, H., R. Wang, Z. Yu, N. Wang, Z. Gu, and B. Zheng. 2017. Automatic plankton image classification combining multiple view features via multiple kernel learning. BMC bioinformatics, 18(16): 570. doi:10.1186/s12859-017-1954-8

CHAPTER 7 Summary of the Dissertation

Whether strapped to a pier, perched on a lab bench, towed behind a ship, or carried by an autonomous vehicle, plankton imaging systems are challenging to develop and deploy (Davis et al. 1992; Sieracki et al. 1998; Samson et al. 2001; Benfield et al. 2003; Watson 2004; Sosik and Olson 2007; Cowen and Guigand 2008; Gorsky et al. 2010; Picheral et al. 2010; Schulz et al. 2010; Thompson et al. 2012; Briseño-Avena et al. 2015; Ohman et al. 2018 – See Fig. 7.1). As scientific instruments, these systems aim to provide objective quantification of plankton, thereby facilitating better understanding of ocean processes and furthering our understanding of Earth's functions, condition, and resources. Therefore, it is important that the contents of these images be accurately assessed, in an increasingly efficient and automated way.

This dissertation investigates an end-to-end sequence of steps for efficiently extracting and classifying plankton images. In preparatory steps, I provide new approaches to image processing in order to optimize and segment plankton images. I show how to incorporate context metadata as a new modality in the field of plankton image classification, in order to markedly improve classification accuracy. Context metadata prove beneficial for both convolutional neural networks (CNNs) and for conventional feature-based machine learning algorithms. I compare different architectures for incorporating context metadata into CNNs, which should facilitate adoption of this approach by others. I also show the importance of optimizing hyperparameters in order to maximize the performance of all machine learning methods considered. In addition to improving biological object classification and providing new approaches that advance the field of machine learning, as a by-product, this dissertation hopefully serves as a sufficient primer in machine learning to facilitate a plankton ecologist adopting and implementing machine learning technology.

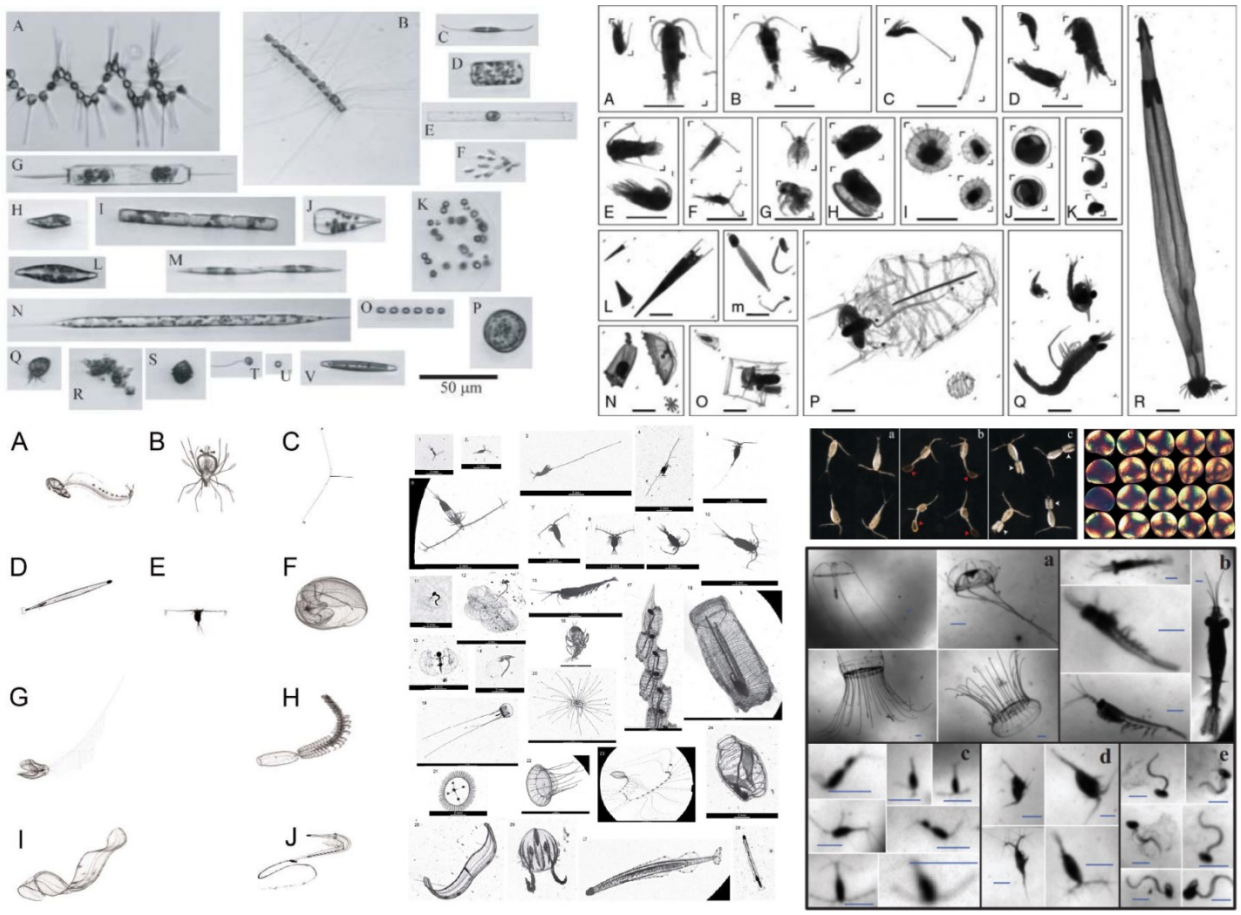


Figure 7.1: Example plankton images from Sosik and Olson (2007), Gorsky et al. (2010), Cowen and Guigand (2008), Ohman et al. (2018), Briseño-Avena et al. (2015), Thompson et al. (2012), Briseño-Avena et al. (2015).

All supervised machine learning algorithms require features. Chapter 2, "A Review of Feature Extraction Techniques for Automating Biological Object Classification in Images," describes dozens of types of machine learning features relevant to biological object classification, and provides an organizational structure to consider more precisely the underlying concepts each feature type is attempting to quantify. All types have been used to classify plankton images with varying degrees of success (Grosjean et al. 2004; Blaschko et al. 2005; Hu and Davis 2005; Sosik and Olson 2007; Gorsky et al. 2010; Luo et al. 2011; Ellen et al. 2015).

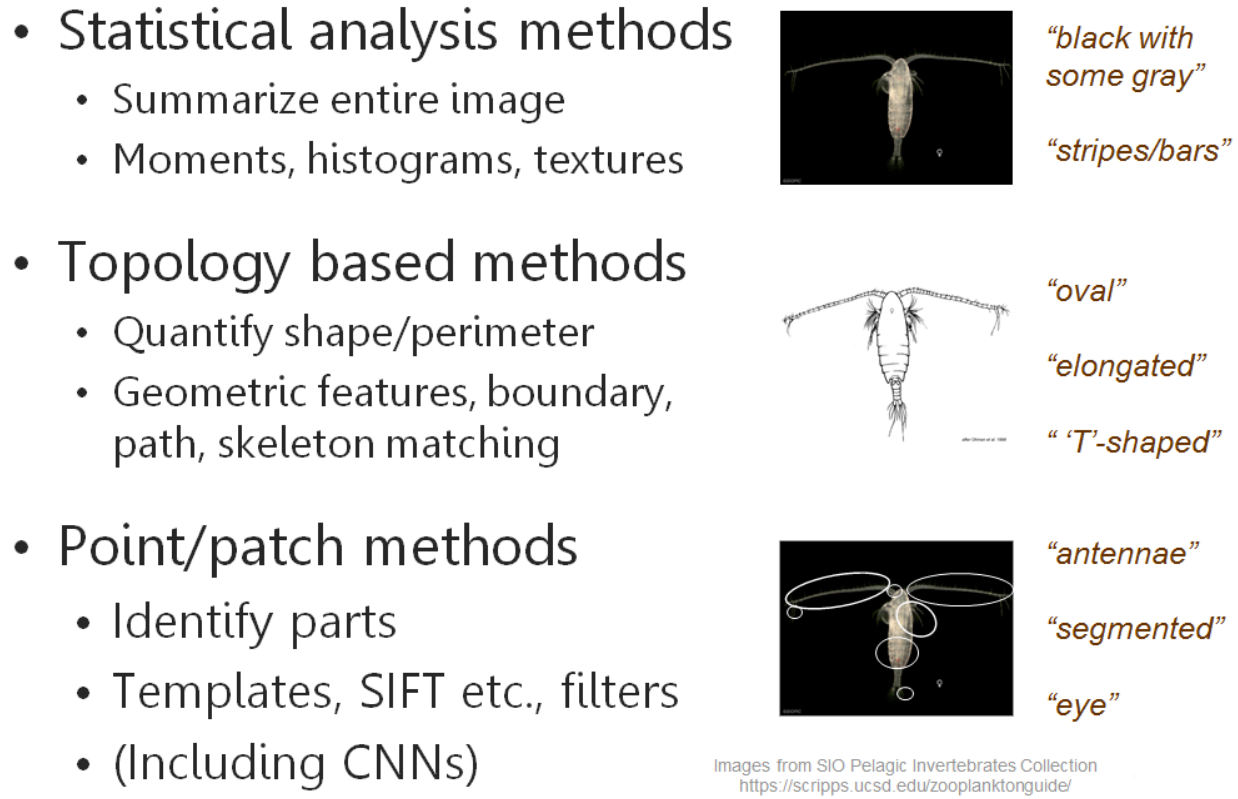


Figure 7.2: Three types of features used for plankton image classification

Features that quantify shape seem particularly promising to consider in the future, particularly because recent publications in that area have yet to be implemented with respect to plankton classification. As individual organisms with distinct boundaries, planktonic shapes always form a closed perimeter and there should be a large amount of intra-class similarity in that perimeter in addition to the interior pixel contents. Shape contexts (Belongie et al. 2002; Ling and Jacobs 2007) attempt to match shapes that are similar but deformed, and strands (Temlyakov et al. 2010) and spike counts (Nguyen et al. 2013) are methods that create features that describe the protrusions and elongated structures that are common in most types of plankton (e.g. legs, tentacles, antennae, pseudopodia, frustules). Certain types of features may outperform others based on image acquisition characteristics, such as texture features outperforming shape

features for images with irregular illumination (Hu and Davis 2005; Hu 2006). However, the choice of feature type is not mutually exclusive so unless available computation is extremely limited, many features should be included as possible for best results (Sosik and Olson 2007; Luo et al. 2011; Ellen et al. 2015).

Available computation has roughly doubled every two years for the past 50 years, a phenomenon, known as "Moore's Law" (Moore 1965), and this roughly applies not just to computation speeds, but also to memory capacity and power consumption. Therefore, feature types that seemed intractable or low value as recently as 10 years ago are potentially easy to implement today. Even if Moore's Law is abating, as suggested by Waldrop in "The chips are down for Moore's law", (Waldrop 2016) there is still an accretion of more efficient implementations and new ideas to assess as the field of machine learning continues to evolve and mature. Additionally the resolution of plankton images continue to increase (Picheral et al. 2010, Grossmann et al. 2015; Gallager 2017; Orenstein and Beijbom 2017; Ohman et al. 2018), so plankton images will continue to increase in fidelity, allowing for more accurate and innovative feature extraction algorithms.

Whether explicitly calculated geometric features as in Chapter 2, or implicitly evolved as CNN filters, computing effective features from plankton images requires clear images as a starting point, and accurate segmentation is required to produce accurate geometric features. Chapter 3, "Improving Object Detection and Segmentation for In Situ Plankton Images," illustrates a technique for improving the images captured by *Zooglider*. The flat-fielding algorithm presented in Chapter 3, adopted from astronomy, clearly improves image contrast and uniformity. The segmentation algorithm I present is based on a Canny (1986) edge detector, but

uses an original two pass approach that segments plankton from *Zooglider* images with higher fidelity that conventional single pass applications

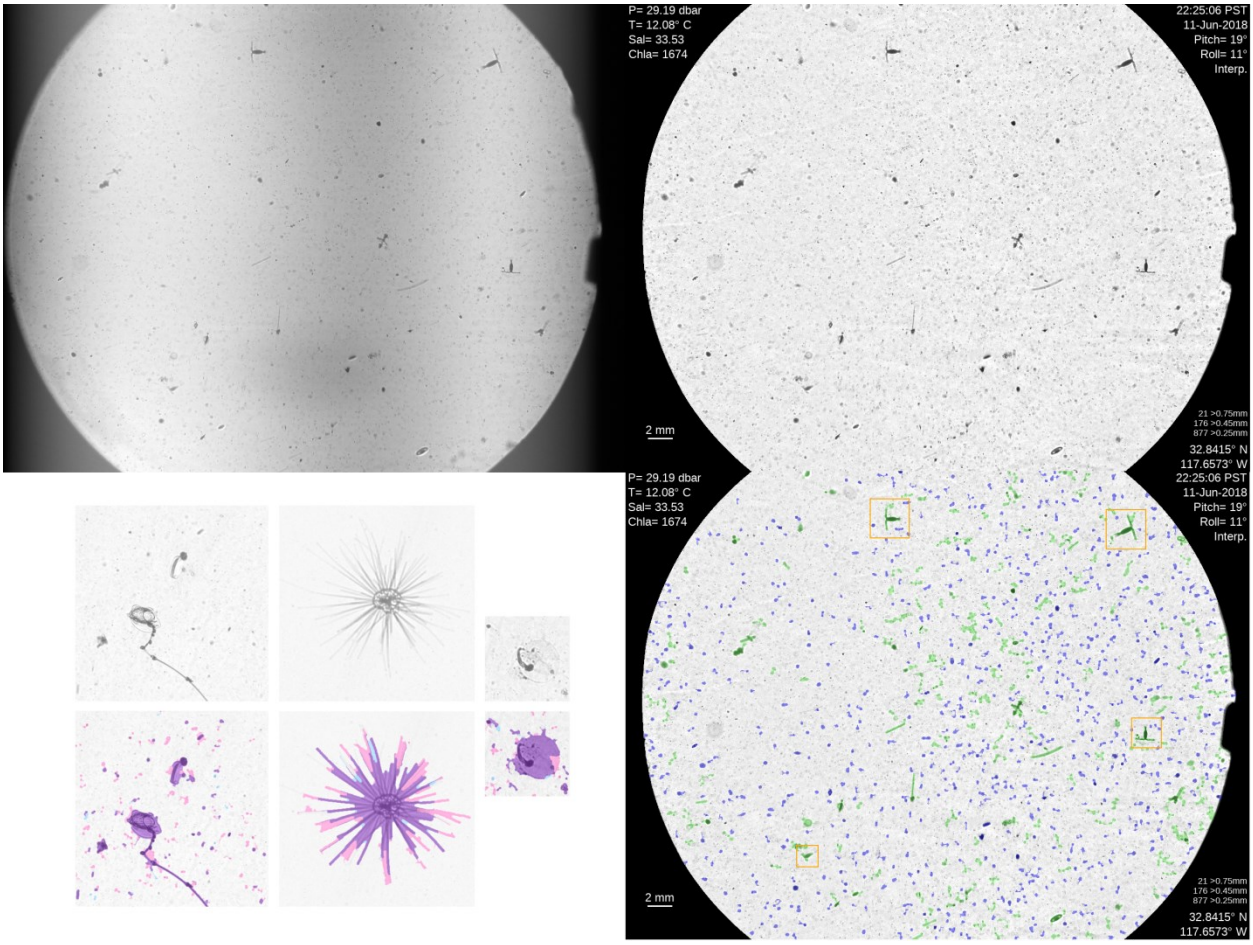


Figure 7.3: A raw zooglider image (upper left) is first flatfielded (upper right), then segmented by two different Canny edge detectors to perform detection and segmentation of thin and transparent objects (bottom left). Small ROIs are only enumerated, while large ROIs are both enumerated and image tiles are retained.

Just as CNNs do not require a priori heuristics to encode as features, and instead evolve filters to perform classification, supervised Deep Learning segmentation algorithms such as DeepEdge (Bertasius et al. 2015) and holistically-nested edge detection (Xie and Tu 2015), do not use a priori heuristics to identify intensity discontinuities, but instead evolve hierarchical features to determine segmentation boundaries. As with CNNs, these supervised algorithms

require human annotations for validation, but like CNNs they also outperform most non-deep learning segmentation benchmarks (Bertasius et al. 2015) and can even achieve this accuracy with minimal computation time per image (Xie and Tu 2015). There is no indication that these Deep Learning segmentation algorithms would achieve lower segmentation accuracy on biological objects than the natural objects in the published papers.

Chapter 4, "Quantifying California Current Plankton Samples with Efficient Machine Learning Techniques," increases the number of training examples uses with conventional feature-based algorithms until an asymptote is reached in accuracy at roughly 4,000 examples per class. However, increasing the number of examples disproportionately increases the amount of training time for feature-based algorithms: for example, the relationship between number of examples and computational complexity is worse than quadratic (Cortes and Vapnik 1995; Pedregosa et al. 2011). To mitigate this impact, I evaluate size fractionation, partitioning the training data by the area feature into non-overlapping subdivisions (e.g. small, medium, and large). As expected, this procedure greatly reduces the computation time, and I find this strategy of training multiple smaller models achieves nearly the same accuracy as a single model (Ellen et al. 2015). In turn, this size fractionation approach should allow for larger training sets to be employed, which should result in higher accuracy.

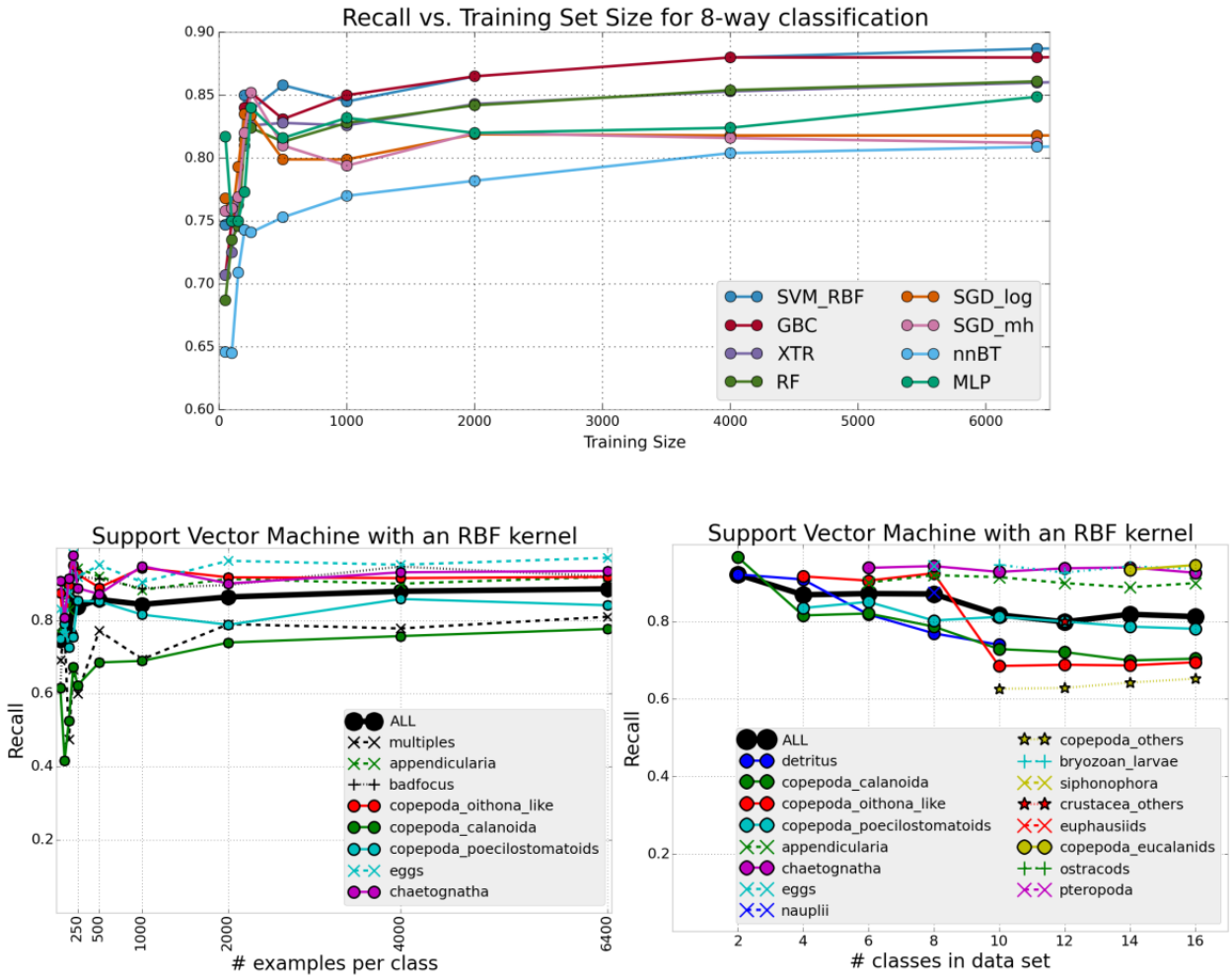
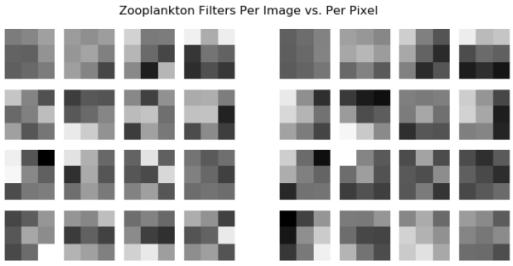


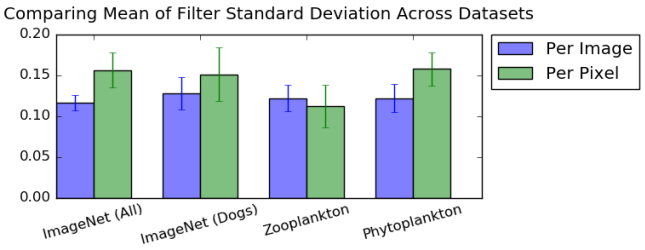
Figure 7.4: For an 8-way classification problem of Zooscan images, Support Vector Machines with a Radial Basis Function (SVM_RBF) performed the best (top). Recall by class remained consistent regardless of the number of examples per class (bottom left); and having fewer classes resulted in higher accuracy, with all trials with greater than 10 classes having very similar recall.

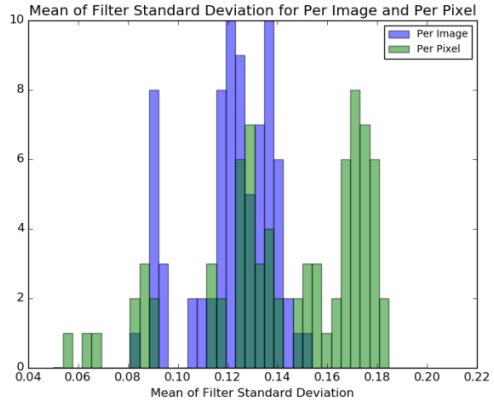
Data set augmentation is a strategy to effectively increase the number of available images. For plankton, transformations such as rotation and reflection do not alter the class label, and can be utilized without reservation, and have been found to increase accuracy (Dieleman et al. 2015). Another augmentation strategy is to select additional images with similar visual content (Johnson et al. 2015) or metadata (MacAuley and Leskovec 2012), an approach that

becomes increasingly viable as the number of available images increases. Future work with non-deep learning algorithms is important because these algorithms are better suited than CNNs to remote, on-board, low-power deployments. On board assessments performed by machine learning algorithms, even if suboptimal compared to CNNs, enable real-time adaptive sampling capabilities during the execution of a sampling mission, such as detection of zooplankton thin layers in the ocean or recognition of an unexpected population, such as a bloom, that would be valuable to sample again.

Before being classified by a CNN, images are typically normalized in some manner so that raw pixel intensities are converted into a range expected by a typical CNN library. Chapter 5, "Correlating Filter Diversity with Convolutional Neural Network Accuracy," examines three aspects of training CNNs on a new domain of images. This includes image normalization, deciding whether to train networks from scratch, and examining the first layer of filters to determine what the CNN model is learning. We find that a common technique for many image classification tasks, zero-phase component analysis or ZCA whitening (LeCun and Ranzato, 2013) does not perform best on our plankton images. Instead we find that per image normalization, sometimes referred to as global contrast normalization (GCN), works best on our plankton images (Graff and Ellen, 2016). We find that training our networks from scratch results in better accuracy than attempting to transfer networks from another domain. These two findings hold not only for our own images, but also for a set of phytoplankton images acquired with a flow-through cytometer (Sosik and Olson 2007). We also find a correlation between the diversity of filters and accuracy for trained models.







NETWORK TESTING ACCURACY
AVERAGED ACROSS FIVE TRIALS

Dataset	Per Image	Per Pixel
ImageNet(All)	56.85	63.21
ImageNet(Dogs)	30.44	36.11
Zooplankton	71.06	69.18
Phytoplankton	79.25	79.70

Figure 7.5: First layer filters (upper left) evolve throughout training. After training multiple replicates on multiple types of images, the distribution of means of the standard deviation of filters (lower left) is lower for the per image normalization. Across four different data sets, the zooplankton images are the only image type that evolved higher mean of filter standard deviation using per image normalization than per pixel normalization (upper right). This matched with zooplankton images being the only image type that had a higher accuracy using per-image normalization than per pixel normalization (bottom right).

If the diversity metric could in some way inform the design or selection of randomly initialized filters, it would potentially be useful for achieving higher accuracy, reducing computation time, or both. For example, when training a new network it would be useful to know which of two sets of filters was more likely to have higher accuracy or converge faster. Another potential accuracy gain could be obtained from algorithms that are designed to further exploit the rotational invariance or relative scale of plankton images, such as power spectra (Torralba & Oliva 2003), or rotating images of plankton to a uniform alignment before assessing them, as has been done to improve facial recognition (Taigman et al. 2014).

Once images are normalized, they can be used as training data for a convolutional neural network. Chapter 6, "Improving Plankton Image Classification Using Context Metadata," determines which configuration of layers, connectivity, filter size, and dropout works best with plankton images to establish a baseline for classification by a CNN. The accuracy of the CNN far exceeds the accuracy of feature-based classifiers, and accuracy of all algorithms increases as the number of training examples increases, up to 5,000 for most of the 27 categories, and 350,000 images total.

A major result of this dissertation is that accuracy is improved for all algorithms when geotemporal and hydrographic metadata are included. Accuracy for the CNN improves the most when the geometric features are included with the geotemporal and hydrographic metadata. The context metadata not only include measurements from the *Spray* glider platform on which *Zooglider* is based, but also includes acoustic backscatter from the dual frequency *Zonar* (Ohman et al. 2018).

In Chapter 6, I also consider multiple architectures for inclusion of metadata. The best performing architecture uses multiple hidden layers to allow for interaction between the metadata features first, and then a second layer of interaction with the features extracted from the pixels. The best performing model achieves an accuracy of 92.3%, with metadata providing the final one percentage point improvement (which equates to ~15% error reduction). Also, while fine tuning (Chu et al. 2016) or domain transfer (Orenstein and Beijbom 2017) are techniques for taking a CNN trained on one type of data and adapting it to a new type of images, I consistently found that training our own networks *de novo* provided the best results with our images (Graff and Ellen 2016).

Additionally, even the smallest CNNs considered have performance beyond feature-based approaches, a result that is notable for two reasons. First, it allows for someone new to the domain (whether experienced or not) to iterate quickly to determine at a broad level what characteristics will be successful with their particular images. Second, efficiency is a concern, because there is still a significant gap in the number of images used in our training set compared to the rapidly expanding rate at which the images can be acquired. If a particular enhancement or algorithm provides superior accuracy over a more efficient algorithm, but the more accurate algorithm cannot be put into use, as described in Robinson et al. 2017, then the enhancement is a mere novelty and does not achieve the end goal of helping sort images.

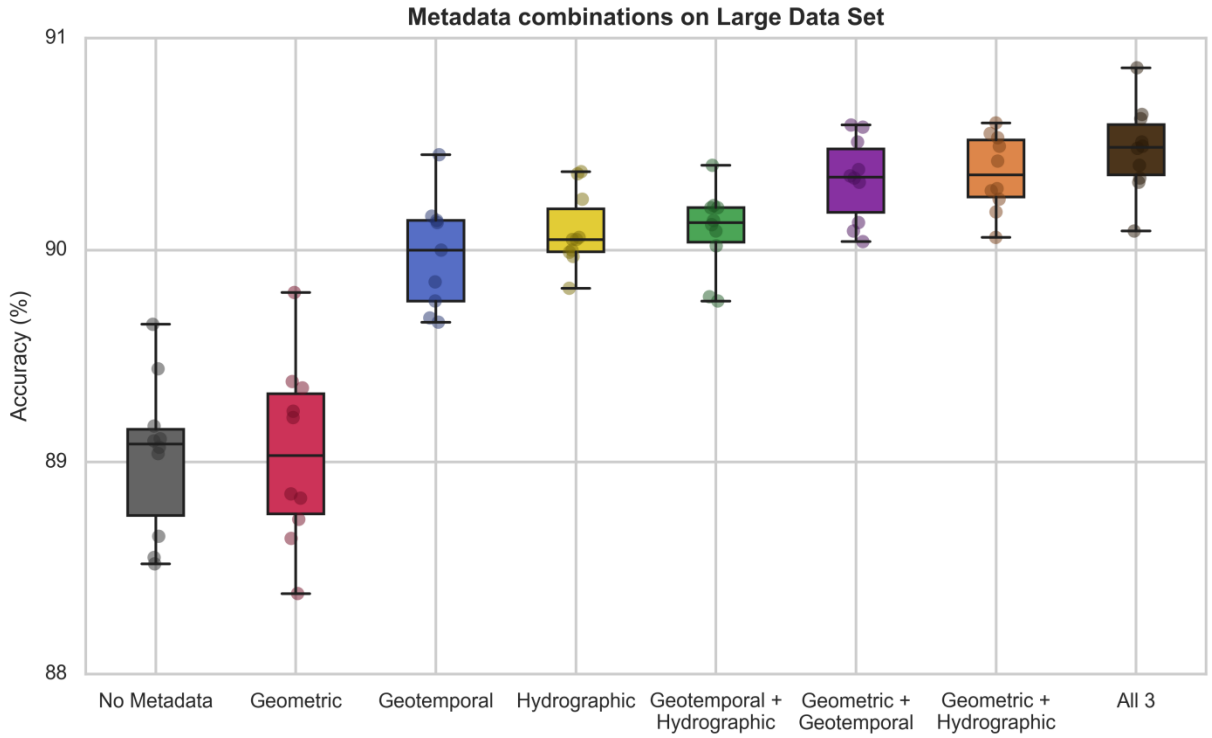


Figure 7.6: Including any of geometric features, geotemporal metadata, and hydrographic metadata improves CNN classification accuracy; including all three yields the highest accuracy.

Here, the metadata used included acoustic backscatter measurements. This kind of multi-modal machine learning will likely continue to be of interest as systems are developed that include multiple modalities such as acoustic (Briseño-Avena et al. 2015; Ohman et al. 2018) or fluorescence signatures, in addition to reflected light (Beijbom et al. 2016). One of the advantages of CNNs over previous feature-based algorithms is that the CNNs are intended to resolve issues such as recognizing objects under varying illumination, scale, and stretching (LeCun et al. 2015). These are not only a challenge for plankton images (Hu and Davis 2005) but for most image types. However, one limitation of CNNs is that they rely on patterns of pixels that are similar to previously seen pixels in training examples. These images are just a projection of a momentary posture and position of the real-world object, not just for plankton, but for any image captured by a camera (as opposed to a PowerPoint slide or abstract painting). Newer architectures, such as "deformable part descriptors" (Zhang et al. 2013) are designed to address this deficiency, and showed 6% improvement. CapsNet is a potential successor to the CNNs presented in this dissertation, a CapsNet contains a network of recursive capsules that not only learns maximally discriminative filters like a CNN, but also learns transformation matrices to represent pose and part-whole relationships (Sabour et al. 2017).

Convolutional Neural Networks clearly outpace the previous state of the art for general purpose image classification, posting annual performance gains in excess of 10 percentage points and rapidly approaching human level performance on these tasks (He et al. 2015). Part of the reason for their wide adoption is that they generalize exceptionally well. Specific features or heuristics are not required a priori, and not just the same algorithm as a design template, but the trained model can be used with very little modifications from one image data set to the next (Sermanet et al. 2013; Zeiler and Fergus 2014; LeCun et al. 2015). This included tangential

domains, such as using a model trained on RGB images being used to successfully perform object recognition and pose estimation from RGB-D images, which include a depth channel (distance from the object to the image plane), which is a common image format within robotics (Schwarz et al. 2015). Even within plankton classification, Orenstein and Beijbom (2017) found that using a model pre-trained on ImageNet images provided better classification accuracy than a model trained using only the plankton or phytoplankton images. As the concept of CNNs matures, including hybridizations and successors such as CapsNet, and as more practitioners become familiar with implementing these algorithms, an open question is whether this type of domain transfer is truly optimal, or just a convenient starting point, as general purpose CNN models have often been trained on data sets such as the 1.2 million images in the ImageNet dataset, whereas the size of most plankton datasets numbers in the tens of thousands.

The answer to this question may come from deep learning theorists, or it may be determined empirically. While the utility of a CNN is determined by a metric such as accuracy on recognizing objects within images, there is no specific mathematical notation to represent the power or expressiveness of a CNN (Bengio and Delalleau 2011). Empirical evidence of the power of CNNs is that their success was greatly increase when they were artificially dampened with dropout (Srivastava et al. 2014), which would not be the case unless they were more significantly more expressive than the variety in current data set.

If the expressiveness of a CNN is very large, it is possible that using a CNN pre-trained on common objects before being fine-tuned to plankton data would be analogous to how a human expert progressing from infancy to plankton expert gradually learns to describe and recognize broad shapes and colors before being able to identify the most subtle taxonomic keys. However, it is also possible that, similar to dropout being required, that a CNN trained on

ImageNet data may be overly specialized. For example, a CNN that has been trained on thousands of classes from ImageNet, or trained to recognize pedestrians from a self-driving car, or trained to identify facial features to unlock a mobile phone may not have evolved to be able to discriminate between more subtle features such as differentiating the internal organs of nearly transparent gelatinous organisms. Until there is sufficient evidence, domain practitioners will try to decide for themselves, as evidenced by the Tajbakhsh et al. (2016) paper entitled "Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning?"

In conclusion, deep learning, including convolutional neural networks, has proven successful in advancing the state-of-the-art for audio, video, speech, and image processing (LeCun et al. 2015), so it is not surprising that CNNs significantly improve accuracy for classifying plankton images. Even though CNNs tend to require more training images than previous algorithms and require greater computational overhead, even moderately sized CNNs with a small amount of training data outperform previous plankton processing algorithms, and are readily adaptable to microcomputers with graphical processing units (GPUs). Additionally, inclusion of context metadata is more effective per unit of computation than the CNN itself, and is not difficult to implement, especially since many of the context data measurements are likely to be acquired in conjunction with the imaging process. The inclusion of context metadata also significantly improves the accuracy of non-CNN algorithms, reducing errors by ~30% on average. Moreover, the inclusion of context metadata and geometric features significantly improves accuracy of even the optimal CNN architecture, reducing errors by ~15% on average.

Each of the four chapters above with original results (Chapters 3, 4, 5, and 6) contained some degree of blending or borrowing, leveraging existing concepts. In this sprit, additional

ensemble approaches that blend the best of different algorithms and concepts are likely to further increase classification performance.

7.1 References

Beijbom, O., Treibitz, T., Kline, D. I., Eyal, G., Khen, A., Neal, B., ... & Kriegman, D. (2016). Improving automated annotation of benthic survey images using wide-band fluorescence. Scientific reports, 6, 23166.

Bengio, Y., & Delalleau, O. (2011, October). On the expressive power of deep architectures. In International Conference on Algorithmic Learning Theory (pp. 18-36). Springer, Berlin, Heidelberg.

Bertasius, G., Shi, J., & Torresani, L. (2015). Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 4380-4389).

Briseño-Avena, C., Roberts, P. L., Franks, P. J., & Jaffe, J. S. (2015). Zoops-O2: A broadband echosounder with coordinated stereo optical imaging for observing plankton in situ. Methods in Oceanography, 12, 36-54.

Chu, B., Madhavan, V., Beijbom, O., Hoffman, J., & Darrell, T. (2016, October). Best practices for fine-tuning visual classifiers to new domains. In European Conference on Computer Vision (pp. 435-442). Springer, Cham.

Cowen, R. K., and Guigand, C. M. (2008). In situ ichthyoplankton imaging system (ISIIS): system design and preliminary results. Limnology and Oceanography: Methods, 6(2), 126-132.

Dieleman, S., Willett, K. W., & Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. Monthly notices of the royal astronomical society, 450(2), 1441-1459.

Gallager, S. M. (2017). U.S. Patent Application No. 15/512,893.

Grossmann, M. M., Gallager, S. M., & Mitarai, S. (2015). Continuous monitoring of near-bottom mesoplankton communities in the East China Sea during a series of typhoons. Journal of oceanography, 71(1), 115-124.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. Proceedings of the IEEE international conference on computer vision. 1026-1034. doi: 10.1109/iccv.2015.123

Hu, Q., & Davis, C. (2005). Automatic plankton image recognition with co-occurrence matrices and support vector machine. Marine Ecology Progress Series, 295, 21-31.

Hu, Q. (2006). Application of statistical learning theory to plankton image analysis (Doctoral dissertation, Massachusetts Institute of Technology and Woods Hole Oceanographic Institution).

Johnson, J., Ballan, L., & Fei-Fei, L. (2015). Love thy neighbors: Image annotation by exploiting image metadata. In Proceedings of the IEEE international conference on computer vision (pp. 4624-4632).

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. nature, 521(7553), 436.

Moore, G. E. (1965). Cramming more components onto integrated circuits. Electronics 38 (8): 114–117.

Orenstein, E. C., & Beijbom, O. (2017, March). Transfer Learning and Deep Feature Extraction for Planktonic Image Data Sets. In Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on (pp. 1082-1088). IEEE.

Picheral, M., Guidi, L., Stemmann, L., Karl, D. M., Iddaoud, G., & Gorsky, G. (2010). The Underwater Vision Profiler 5: An advanced instrument for high spatial resolution studies of particle size spectra and zooplankton. Limnology and Oceanography: Methods, 8(9), 462-473.

Sabour, S., Frosst, N., & Hinton, G. E. (2017). Dynamic routing between capsules. In Advances in Neural Information Processing Systems (pp. 3856-3866).

Schwarz, M., Schulz, H., & Behnke, S. (2015, May). RGB-D object recognition and pose estimation based on pre-trained convolutional neural network features. In Robotics and Automation (ICRA), 2015 IEEE International Conference on (pp. 1329-1335). IEEE.

Sermanet, P., Kavukcuoglu, K., Chintala, S., & LeCun, Y. (2013). Pedestrian detection with unsupervised multi-stage feature learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 3626-3633). doi: 10.1109/CVPR.2013.465

Sieracki, C. K., Sieracki, M. E., & Yentsch, C. S. (1998). An imaging-in-flow system for automated analysis of marine microplankton. Marine Ecology Progress Series, 168, 285-296.

Sosik, H. M., & Olson, R. J. (2007). Automated taxonomic classification of phytoplankton sampled with imaging‐in‐flow cytometry. Limnology and Oceanography: Methods, 5(6), 204-216.

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

Tajbakhsh, N., Shin, J. Y., Gurudu, S. R., Hurst, R. T., Kendall, C. B., Gotway, M. B., & Liang, J. (2016). Convolutional neural networks for medical image analysis: Full training or fine tuning?. IEEE transactions on medical imaging, 35(5), 1299-1312. doi: 10.1109/tmi.2016.2535302

Thompson, C. M., Hare, M. P., & Gallager, S. M. (2012). Semi-automated image analysis for the identification of bivalve larvae from a Cape Cod estuary. Limnology and Oceanography: Methods, 10(7), 538-554.

Waldrop, M. M. (2016). The chips are down for Moore's law. Nature News, 530(7589), 144.

Zeiler M.D., Fergus R. (2014) Visualizing and Understanding Convolutional Networks. In: Fleet D., Pajdla T., Schiele B., Tuytelaars T. (eds) Computer Vision – ECCV 2014. ECCV 2014. Lecture Notes in Computer Science, vol 8689. Springer. doi: 10.1007/978-3-319-10590-1_53

Zhang, N., Farrell, R., Iandola, F., & Darrell, T. (2013). Deformable part descriptors for fine-grained recognition and attribute prediction. In Proceedings of the IEEE International Conference on Computer Vision (pp. 729-736).