

University of Alabama in Huntsville

LOUIS

Dissertations

UAH Electronic Theses and Dissertations

2019

Local supervised methods and uncertainty quantification for boundary detection in images

Margaret Coreen Lund

Follow this and additional works at: <https://louis.uah.edu/uah-dissertations>

Recommended Citation

Lund, Margaret Coreen, "Local supervised methods and uncertainty quantification for boundary detection in images" (2019). *Dissertations*. 386.

<https://louis.uah.edu/uah-dissertations/386>

This Dissertation is brought to you for free and open access by the UAH Electronic Theses and Dissertations at LOUIS. It has been accepted for inclusion in Dissertations by an authorized administrator of LOUIS.

**LOCAL SUPERVISED METHODS AND UNCERTAINTY
QUANTIFICATION FOR BOUNDARY DETECTION IN
IMAGES**

by

MARGARET COREEN LUND

A DISSERTATION

**Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
The Department of Mathematical Sciences
to
The School of Graduate Studies
of
The University of Alabama in Huntsville**

HUNTSVILLE, ALABAMA

Spring 2019

In presenting this dissertation in partial fulfillment of the requirements for a doctoral degree from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department or the Dean of the School of Graduate Studies. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this dissertation.

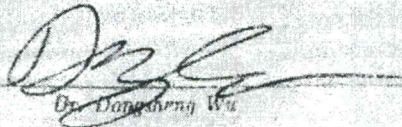
Margaret C. Lund
Margaret Green Lund

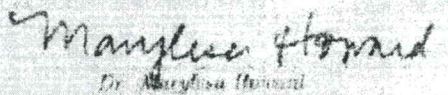
3-27-19
(date)

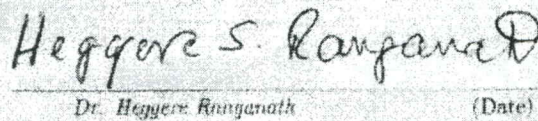
DISSERTATION APPROVAL FORM

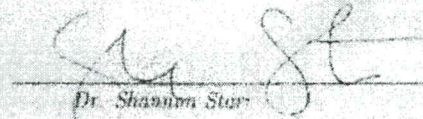
Submitted by Margaret Corcen Lund in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematical Sciences and accepted on behalf of the Faculty of the School of Graduate Studies by the dissertation committee.

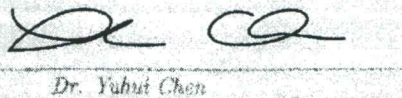
We, the undersigned members of the Graduate Faculty of The University of Alabama in Huntsville, certify that we have advised and/or supervised the candidate of the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Mathematical Sciences.



Dr. Dongsheng Wu 3/27/2019
(Date) Committee Chair


Dr. MaryLisa Howard 3-25-19
(Date)


Dr. Heggere Ranganath 3-27-2019
(Date)


Dr. Shannon Star 3/27/2019
(Date)


Dr. Yuhui Chen 03/29,2019
(Date)


Dr. Taha Diagana 03/27/2019
(Date) Department Chair


Dr. Sundar Christopher 3/29/19
(Date) College Dean


Dr. David Berkowitz 4/9/19
(Date) Graduate Dean

ABSTRACT


School of Graduate Studies
The University of Alabama in Huntsville

Degree Doctor of Philosophy College/Dept. Science/Mathematical Sciences
Name of Candidate Margaret Coreen Lund
Title Local Supervised Methods and Uncertainty Quantification for Boundary
Detection in Images

Boundary detection is a powerful tool for quantitative image analysis that allows researchers to extract crucial information about a scene. Many existing methods rely on sharp changes in luminance, chromaticity, or texture within an image to predict boundary locations between regions, and images without these features have proven difficult to partition. This work presents two new supervised statistical boundary detection methods based on image segmentation that incorporate spatial information to locate boundaries between regions with overlapping intensity histograms, specifically for images where the regions are known but precise boundary locations are unknown. The segmentation of a pixel is determined by comparing its intensity to distributions from local, user-supplied training pixels, where local is defined differently for the two algorithms, and boundaries are determined as the borders between the identified regions. The applications of each new algorithm are explored, and each algorithm's success is demonstrated on synthetic images as well as real images from lab experiments. Additionally, because of the statistical nature of the algorithms, methods for uncertainty quantification are explored and maps showing the uncertainty in the boundary location and the distinguishability between classes are provided.

Abstract Approval: Committee Chair 
Dr. Dongsheng Wu

Department Chair 
Dr. Toka Diagana

Graduate Dean 
Dr. David Berkowitz

ACKNOWLEDGMENTS

First, to Dr. Marylesa Howard and Dr. Aaron Luttmann for helping me realize I would have hated a career on Wall Street. My life took a sharp turn that first summer as your intern and I couldn't be happier with where I've ended up.

Thank you to my committee members, Dr. Dongsheng Wu, Dr. Marylesa Howard, Dr. Heggere Ranganath, Dr. Shannon Star, and Dr. Yuhui Chen, for the valuable advice and suggestions.

Thank you to all of the incredible women who inspire me everyday to be better yet love me, flaws and all. Mom, Emily, Katherine, Marylesa, Carlin and dozens of others, you've been a constant source of grace, understanding, and strength. Here's to strong women - may we know them, may we be them, may we (one day) raise them.

To my family - for being proud of me for the little things as much as for the big things. To Dad, for providing me with the best education, a passion for adventuring, and the ability to freely travel. To Mom, for every packed lunch, every science project, and every time you agreed to read just one more chapter with me. I absolutely am where I am today because you were a stay-at-home mom. And to all four of you for providing me with the most supportive and loving family a gal could ask for.

Finally, to Ben, my husband and partner in all things. The path to a doctorate is a bumpy one and only you have been there for every high and every low. One lifetime together could never be enough.

TABLE OF CONTENTS

List of Figures	x
List of Symbols	xiii
Chapter	
1 Introduction	1
1.1 Organization	3
2 Literature Review	5
2.1 Segmentation and Boundary Detection	6
2.2 Threshold Based Methods	6
2.3 Machine Learning Methods	8
2.3.1 Edge-Based Methods	11
2.3.2 Region-Based Methods	16
2.3.3 Statistical Methods	17
2.3.4 Neural Networks	19
2.4 Uncertainty Quantification Methods	22
3 Locally Adaptive Discriminant Analysis	24
3.1 The LADA Algorithm	25
3.2 LADA Example	31

4	Anisotropic Locally Adaptive Discriminant Analysis	38
4.1	The ALADA Algorithm	41
4.2	Visualizing ALADA	49
4.3	ALADA Example	53
5	Uncertainty Quantification	60
5.1	Maximum Likelihood Estimation	60
5.2	Combined t -test and F -test	63
6	Results on Real Data	70
6.1	Time Series Example	71
6.1.1	LADA Results	73
6.1.2	Results from Other Methods	75
6.2	Shear Bands ALADA Example	80
6.2.1	ALADA Results	82
6.2.2	Results from Other Methods	85
6.3	Polar Unroll ALADA Example	87
7	Conclusions	96
7.1	Future Work	98
	APPENDIX A: Supplemental Material	101
A.1	Hypothesis Testing	101
A.2	Normality Testing	102
A.3	Principal Component Analysis	104

A.4 *k*-Nearest Neighbors 105

REFERENCES **107**

LIST OF FIGURES

FIGURE	PAGE
2.1 (a) Original image of coins, (b) image segmentation using thresholding method, and (c) boundaries collected from the segmented image. . . .	7
2.2 Biological neuron and a common mathematical model for a neural network	20
2.3 Cartoon of a neural network with one hidden layer	21
3.1 Cartoon example of selecting local training data for LADA	27
3.2 A noisy synthetic image	32
3.3 Training data for a synthetic staircase image	33
3.4 Circular subimages superimposed on a synthetic staircase image	34
3.5 The LADA segmentation of a synthetic staircase image	35
3.6 Maps of the LADA-determined boundaries for a synthetic staircase image	36
3.7 Ellipses superimposed on a synthetic triangle image	37
4.1 Image of a sphere with examples of elliptical subimages	39
4.2 Comparison of intensity distributions of various local training data sets	41
4.3 Visual breakdown of ALADA using an image of a sphere	51
4.4 Intensity distributions of global vs local training data	52
4.5 A synthetic triangle image	54
4.6 Training data for a synthetic triangle image	55
4.7 Map of $r_{1_{ij}}$ values for a synthetic triangle image	56

4.8	Maps of lambda ratios and $r_{2_{ij}}$ values for a synthetic triangle image . . .	56
4.9	Map of the angles of least variation for a synthetic triangle image . . .	57
4.10	Ellipses superimposed on a synthetic triangle image	57
4.11	The ALADA segmentation of a synthetic triangle image	58
4.12	True boundaries and ALADA-determined boundaries for a synthetic triangle image	58
4.13	Original image with ALADA-determined boundaries superimposed on top	59
5.1	Maximum likelihood estimation and p-value examples	62
6.1	The experimental setup for an impactor striking a sample of synthetic olivine sand	72
6.2	Time evolution of coarse olivine sand loaded from the left by an impactor	74
6.3	Training data and LADA segmentation of a coarse olivine sand image	76
6.4	MLE p-value map and LADA boundaries with uncertainty regions for a coarse olivine sand image	77
6.5	Progression of a single boundary through time, with uncertainty regions, for a series of olivine sand images	78
6.6	Boundaries detected in an olivine sand image using four competitive methods	80
6.7	Image of 1075 steel displaying shear bands from a high strain rate impact.	81
6.8	Training data and ALADA segmentation for a shear band image	83
6.9	MLE p-value map and an F -test and t -test p-value map for a shear band image	84
6.10	ALADA boundaries for a shear band image	85
6.11	Boundaries detected in a shear band image using four competitive methods	87

6.12	Image of a laser-induced, cylindrically converging shock wave propagating in water	88
6.13	Image of a laser-induced, cylindrically converging shock wave propagating in water, after a polar transformation	91
6.14	Training data and the ALADA segmentation of a laser shock image .	92
6.15	MLE p-value map and an F -test and t -test p-value map for a laser shock image	94
6.16	The ALADA-determined bounds on the unrolled shock image and the original shock image	95

LIST OF SYMBOLS

SYMBOL	DEFINITION
X	image
x_{ij}	pixel of interest with image location (i, j)
T	set of training data
T_c	set of training data from class c
$T_{c_{ij}}$	set of training data from class c that is local to pixel x_{ij}
c	class label
C	total number of classes
n	number of neighbors parameter
r	LADA's maximum radius parameter
$r_{1_{ij}}$	ALADA's minor radius for pixel x_{ij}
$r_{2_{ij}}$	ALADA's major radius for pixel x_{ij}
θ_{ij}	ALADA's angle of least variation for pixel x_{ij}
A_{ij}	set of training pixels local to pixel x_{ij} , in a circular shape
B_{ij}	set of training pixels local to pixel x_{ij} , in an elliptical shape
$\lambda_{1_{ij}}$	larger eigenvalue from PCA covariance matrix, local to pixel x_{ij}

λ_{2ij}	smaller eigenvalue from PCA covariance matrix, local to pixel x_{ij}
$e_{\lambda_{2ij}}$	eigenvector associated with smaller eigenvalue from PCA covariance matrix, local to pixel x_{ij}
π_c	prior probability of class c
μ_{cij}	sample mean of training data from class c , local to pixel x_{ij}
σ_{cij}	sample standard deviation of training data from class c , local to pixel x_{ij}
$G(x)$	function that maps a pixel x to a class
E_V	boolean matrix of vertical boundaries
E_H	boolean matrix of horizontal boundaries
$f_c(x)$	function that maps a pixel to its most probable class
ν_{cij}	degrees of freedom from class c , local to pixel x_{ij}
t_{ij}	test statistic from Welch's t -test for pixel x_{ij}
F_{ij}	test statistic from F -test for pixel x_{ij}

For Ben, my Constant.

CHAPTER 1

INTRODUCTION

*Research is what I'm doing
when I don't know what I'm doing.*

—Wernher von Braun

Boundary detection is a vital part of extracting information encoded in images, allowing quantities of interest to be computed including density, velocity, pressure, etc. [2, 3]. For example, the U.S. Department of Energy research complex uses high-energy X-ray sources to image objects, and subsequent analysis is applied to the images to compute the boundary locations and density of the materials [4]. In remote sensing from satellite images, traditional and modern segmentation algorithms are used to identify ground cover such as forest, fields, and rivers [5–9]. The authors in [10] calculate the volume of fruit by identifying boundaries in magnetic resonance images. In medical biology applications, boundary detection methods are essential for identifying boundaries of organs, tissues, or tumors [11–14]. For example, [15] monitors changes in breast tissue location and density to infer breast cancer risk, and in biological systems, boundary detection is used to identify bacteria cell motion [16]. In each of these research fields, boundary detection methods are necessary to distinguish between classes and provide vital information about the scene.

Despite the significant contributions in literature for boundary detection, there are still images for which no existing methods perform well. Some research groups, such as the ones who provided images for this dissertation, have spent years searching for a method that will detect accurate boundaries in their images, and are forced to choose between methods that provide inadequate results or identifying boundaries by hand. In fact, because of the absence of suitable methods, many communities in physics, medicine, and biology still manually determine boundaries in images [9, 17, 18], which is time consuming and subjective. Many existing methods rely on sharp changes in luminance, chromaticity, or texture to detect boundaries and may perform poorly on images with low contrast, heteroskedasticity (variances vary across the image), low signal to noise ratio, and regions with spatial trends in intensity.

This dissertation proposes two new supervised boundary detection schemes. The first method, locally adaptive discriminant analysis (LADA), utilizes the statistics of local training data to segment the image into classes. Here “local” is defined by two user-supplied parameters in order to adapt to a wider range of images. The second approach, anisotropic locally adaptive discriminant analysis (ALADA), does something that no other existing method does- it utilizes the statistics of both pixel intensities and training pixel locations in order to identify boundaries.

Each method provides a detailed algorithm for restricting training data for each pixel to only the locally occurring classes, building a statistical segmentation model based on the Gaussian assumption of quadratic discriminant analysis [26], and identifying boundaries between the identified classes. By restricting training data for each pixel to the locally occurring classes, these methods are adequately able

to separate classes with spatially varying intensities, making them appropriate for images with shadows, heterogeneous illumination, low contrast, heteroskedasticity, and high noise.

To enhance its utility for applications, we combine our boundary identification with two methods for quantifying uncertainty in the boundary results. These methods use trainers based on probability distributions, allowing us to compute various confidence measures for each pixel in the image. The results for the first test can be interpreted as uncertainty in the boundary location, which can then be propagated through calculations, giving error bars on quantities of interest. The second test identifies regions where neighboring classes have statistically equivalent distributions, meaning we cannot confidently choose one class over another. Researchers who are interested in not only identifying boundaries, but also knowing how confident they can be in the boundary locations will greatly benefit from these uncertainty quantification methods.

1.1 Organization

The remainder of this dissertation is organized as follows. Chapter 2 explores existing boundary detection methods and existing methods for uncertainty quantification. In Chapter 3, LADA is explained and a step-by-step example is provided. In Chapter 4, ALADA is explained and another step-by-step example is provided. Chapter 5 presents two methods for quantifying uncertainty in the resultant boundaries. Chapter 6 presents three images captured during real laboratory experiments to demonstrate LADA and ALADA results, as well as uncertainty quantification.

Results from four competitive methods are also included, for comparison. Finally, Chapter 7 details the conclusions and future work. For the reader's convenience, Appendix A gives explanations of several machine learning and statistics topics that are used throughout this work.

CHAPTER 2

LITERATURE REVIEW

Digital boundary detection dates (somewhat surprisingly) back to the 1950's, when G.P. Dinneen started studying 90×90 block black-and-white images of various letters of the alphabet using the Memory Test Computer at MIT. His 1955 paper, "Programming Pattern Recognition," aimed to identify handwritten letters and used thresholding to identify edges as part of the process [19]. What began as rudimentary thresholding studies evolved to include more complex methods, resulting in the sophisticated, cutting-edge boundary detection algorithms being used today. The following sections explore the many types of boundary detection, highlighting many notable, competitive, and creative methods. The final section in this chapter highlights the need for suitable uncertainty quantification methods for boundary detection. Since segmentation is closely related to edge detection, we begin by explaining how segmentation can be used for boundary detection, and include competitive segmentation methods in this chapter.

2.1 Segmentation and Boundary Detection

One common technique for identifying boundaries in an image is to first segment the image, by partitioning it into cohesive regions or classes, such that each pixel (or part of a pixel, in the case of subpixel methods) in the image is labeled as belonging to one of the available classes, and each class has some unifying characteristic such as color, intensity, or texture [20–22, 26]. These methods typically rely on sharp changes in luminance, chromaticity, or texture between classes to aid the algorithm in identifying different classes [7, 23, 26]. If the image is segmented, the image boundaries can be defined to be the boundaries between regions in the segmented image, as shown in Figure 2.1.

2.2 Threshold Based Methods

The simplest way to segment an image, and subsequently identify image boundaries, is by thresholding the image. Thresholding an image is done by choosing an intensity value to be the cutoff point between two classes. All pixels with intensity values below the threshold parameter are assigned to one class and all pixels with intensity values at or above the threshold parameter are assigned to a second class, as detailed in Algorithm 1. Variations of the most basic thresholding technique include methods with multiple thresholds, to give a segmentation with more than two classes, and ways to compute the threshold parameter from the pixel intensities, instead of having the user choose a value [24]. More complicated versions of thresholding are included in the following section on machine learning.

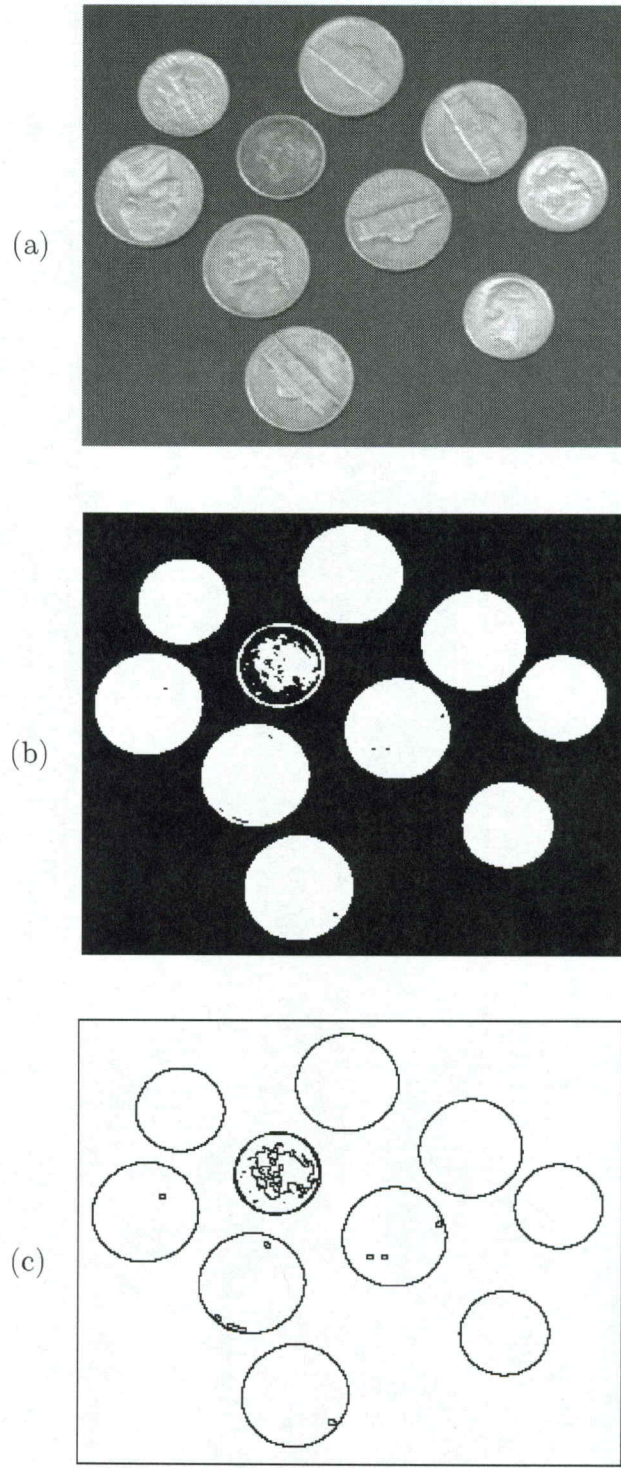


Figure 2.1: (a) Original image of coins, (b) image segmentation using thresholding method, and (c) boundaries collected from the segmented image.

Algorithm 1 Two-class thresholding algorithm for image segmentation

Given image X , define threshold parameter t ,

For each pixel $x_{ij} \in X$:

- If $x_{ij} > t$, place x_{ij} into class 1.
 - Else, place x_{ij} into class 2.
-

Threshold methods perform well on images with regions that have high contrast, such as computed tomography images, but do not take into account spatial relationships within the image [20,21,24]. When intensity scales vary across an image or when there are gradual intensity changes between classes, the threshold parameter becomes incredibly sensitive, giving vastly different results for small perturbations of inputs.

2.3 Machine Learning Methods

Many boundary detection techniques are categorized as machine learning algorithms, which is an umbrella term for algorithms that improve through experience. In other words, the program “learns” without being explicitly programmed. Machine learning algorithms become more precise as more data are analyzed.

Machine learning can be divided into two main categories, unsupervised and supervised. Unsupervised algorithms learn from an unlabeled data set (only the inputs and no desired outputs have been given) and discover relationships, structure, and patterns in the data, which can be used to group data points into categories. Unsupervised methods may be used for such purposes as identifying hidden trends in the data or separating the data into groups. In the context of boundary detection,

unsupervised methods are often used to segment the image by clustering related pixels into classes. These methods are useful when the user knows no additional information about the data set or when the user does not want to influence the results by providing a priori information.

One example of an unsupervised clustering algorithm is an iterative method called k -means clustering, which partitions pixels into k classes, with each pixel belonging to the class with the nearest mean, based on squared Euclidean distance. Each iteration consists of three steps, the data assignment step where the pixels are assigned to the classes, the centroid update step where the mean intensity of each class is recalculated, and an update of the iteration count [25]. These steps are repeated until some convergence criterion is reached, whether that may be reaching a certain number of iterations, or waiting until pixel assignments do not change. Note that pixels are not guaranteed to converge to the same clusters every time, for different initial cluster centroids, and, in fact, are not guaranteed to converge at all. Pseudocode is provided in Algorithm 2.

The k -means algorithm is widely used amongst researchers dealing with vast amounts of unlabeled data but is less common as a boundary detection algorithm. Although the algorithm iterates to find well-separated groups, it is still, essentially, a thresholding algorithm and fails to find satisfactory results in images with classes have overlapping intensity distributions.

The second branch of machine learning, supervised learning, involves algorithms that learn from training data, sets of input values and their corresponding output values. These algorithms identify patterns in the training data and build a

Algorithm 2 k -means algorithm for image segmentation

Given image X , and a number of clusters, K ,

Initialize cluster centroids, $\mu_1^{(1)}, \mu_2^{(1)}, \dots, \mu_K^{(1)}$ randomly and let $t = 1$.

Repeat until stopping criterion are reached:

1. Label every pixel $x_{ij} \in X$ to a cluster such that

$$f(x_{ij}) = \arg \min_k \|x_{ij} - \mu_k^{(t)}\|^2.$$

2. Recompute cluster centers such that

$$\mu_k^{(t+1)} = \frac{\sum_{x_{ij} \in C_k} x_{ij}}{|C_k|}$$

where C_k is the set of all pixels in cluster k and $|C_k|$ is the number of pixels assigned to cluster k .

3. $t = t + 1$
-

deterministic model in order to predict outputs for unlabeled input values. Supervised learning algorithms include classification methods, where the output values are qualitative, and regression methods, where the output values are quantitative [26]. Both types of methods can be used for image segmentation and boundary detection. For example, a researcher could use a supervised classification algorithm to segment an image (where the outputs are discrete class labels) or they could use a supervised regression algorithm to identify the probability of a boundary belonging between neighboring pixels (where the outputs are continuous values between 0 and 1).

Supervised methods are very common in image processing, as the user often has a priori information about the scene that can be used to assist the algorithm in building a model. The user may know the correct class assignment for pixels in the

image and provide these pixel-label pairings as training data in order to predict a class assignment for all the other pixels in the image.

Within the machine learning category, methods are further separated into categories based on how the algorithms approach the task of boundary detection. Techniques that detect boundaries by identifying strong changes in intensity gradient are called edge-based methods while techniques that detect boundaries by identifying homogenous regions are called region-based methods. Statistical methods use statistics of the training data to predict class labels. Neural networks are part of a new field that aims to process information in a way similar to a human brain, adopting parts of the other three categories.

2.3.1 Edge-Based Methods

By definition, an edge is a curve that follows a path of rapid change in image intensity [27], as opposed to a boundary which is a curve that marks the separation of one material or region from another. Methods that identify boundaries by computing edge gradients and selecting the regions over which the largest change occurs are called edge-based methods or gradient-based methods. These are most successful when distinguishing between regions of high contrast [20].

It is important to note that not all boundaries are edges (in the case of a gradual change in image intensity between materials) and not all edges are boundaries (sharp changes in intensity can occur as a result of shadows or heterogeneous lighting across the image) although there is significant overlap in how edges and boundaries are identified. The work presented here is concerned with detecting boundaries be-

tween regions and materials, while ignoring any edges created from shadows, irregular illumination of the scene, and edges that occur as a feature of the imaging system.

The most common edge detection algorithms today are the Sobel edge detector, created in 1968, and the Canny edge detector, created in 1986 [28–30]. Sobel detection computes the magnitude of the 2D image gradients using two 3×3 kernels, one for horizontal edges and one for vertical edges, which are convolved with the image. The vertical Sobel kernel, K_V , and the horizontal Sobel kernel, K_H , are given here on the left and right, respectively,

$$K_V = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \quad K_H = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix},$$

and pseudocode is provided in Algorithm 3.

Algorithm 3 Sobel edge detection

Given image X , horizontal kernel K_H , and vertical kernel K_V ,

1. Compute the matrix convolution of the image with the vertical kernel to get the gradient in the x -direction in order to identify the location and strength of vertical edges,

$$G_x = K_V * X.$$

2. Compute the matrix convolution of the image with the horizontal kernel to get the gradient in the y -direction in order to identify the location and strength of horizontal edges,

$$G_y = K_H * X.$$

3. Compute the combined gradient magnitude, G , where

$$G = \sqrt{G_x^2 + G_y^2}.$$

G can be viewed as a map to identify regions of strong gradient, i.e. edges.

Canny edge detection is a more complex algorithm, built on the principles of the Sobel edge detector. Like Sobel, it uses kernels convolved with the image to identify regions of strong gradients, but that is only one of the many steps involved. First, the image is smoothed with a Gaussian filter to remove noise. A 5×5 Gaussian filter with standard deviation of $\sigma = 1.4$ is common but is not required. A larger Gaussian filter will lower the detector's sensitivity to noise but will increase the localization error of the detector, while a smaller detector will do the opposite. A larger standard deviation will detect stronger edges while a smaller standard deviation will detect finer edges. Note that in real images from experiments, eliminating even low-frequency noise can mean eliminating vital boundary information. The smoothed image is then convolved with edge identifying kernels to compute gradients in the horizontal and vertical directions. Additional kernels may be used to identify the gradients in other directions, such as diagonals. These directional gradients are combined to obtain the gradient magnitude, identical to the process in Sobel edge detection, as well as the gradient direction. Then, an edge thinning technique is used on the map of gradient magnitudes in order to "clean up" any edges, giving sharp, clearly defined edges wherever the gradient was strongest. This can be done using another Gaussian smoothing filter or skeletonization methods. The fourth step takes two user-provided thresholds to further filter out some of the detected edges. Any gradients smaller than the lower threshold are ignored, any gradients larger than the high threshold are marked as strong edges, and any gradients between the thresholds are marked as weak edges. Finally, the weak and strong edges are used to create a final map of edges in the image. Strong edges are always included, and weak edges

are included if they are connected to a strong edge or sometimes if including them means forming an enclosed region. Pseudocode for Canny edge detection is provided in Algorithm 4.

Canny edge detection and its variants are customizable, allowing researchers to tailor the algorithm to their specific images. This makes it very versatile and adaptable but with the drawback that different formulations and input parameters can give disparate results. Despite its lack of robustness, Canny edge detection remains the most popular edge detection method in use today.

Advancements in technology have led to more complicated computational methods, with recent works including a variation of the Sobel method called Prewitt operation [31], zero crossing detectors that use filters to identify regions where the gradient starts increasing or decreasing (i.e. the gradient changes from positive to negative or negative to positive, thereby crossing 'zero') [32], and other gradient based methods [33, 34]. Fuzzy logic edge detection uses intensity gradients to determine the degree to which a pixel belongs to an edge or a uniform region [35]. This idea of partial membership to different groups is useful in determining uncertainty in segmentations, and some fuzzy logic methods, such as fuzzy c-means clustering, incorporate spatial information, which can help the algorithm be robust to noise in the image [36].

Many additional edge-based methods use partial differential equations to identify boundary locations. These methods involve evolving an initial contour until some cost function is minimized, where the cost function usually takes into account not only the fit of a contour to a shape but also the smoothness of the solution. The level set

Algorithm 4 Canny edge detection

Given image X , horizontal kernel K_H , and vertical kernel K_V , t_{min} , and t_{max} ,

1. Pad the image with zeros such that the $n \times m$ input X becomes the $(n + 2) \times (m + 2)$ matrix X_p .
2. Reduce noise in the image by convolving the image with a Gaussian kernel, where the kernel is found using the formula

$$H = \frac{1}{2\pi\sigma^2} \left(-\frac{(i - (k + 1))^2 + (j - (k + 1))^2}{2\sigma^2} \right); 1 \leq i, j \leq (2k + 1).$$

Default parameters are $k = 2$ and $\sigma = 1.4$.

3. Identify location and strength of horizontal and vertical edges by finding the gradient in the x and y directions,

$$G_x = K_V * X_p,$$

$$G_y = K_H * X_p.$$

4. Compute the combined gradient magnitude, G , where

$$G = \sqrt{G_x^2 + G_y^2}.$$

5. Compute the gradient direction, θ , where

$$\theta = \text{atan}\left(\frac{G_y}{G_x}\right).$$

Add 180° to results below -22.5° and then round each angle to the nearest of four directions: 0° , 45° , 90° , or 125° , representing horizontal, positive diagonal, vertical, and negative diagonal directions, respectively.

6. Compute the map of maximum edges, N , where

$$N(x_{ij}) = G(x_{ij}) \text{ if } G(x_a) < G(x_{ij}) < G(x_b)$$

where x_a and x_b are the neighboring pixels of x_{ij} , in the direction of $\theta(x_{ij})$, and

$$N(x_{ij}) = 0 \text{ otherwise.}$$

7. The final edge map is the union of the strong edge set, E_{strong} , and the weak edge set, E_{weak} where

$$E_{strong} = \{N(x_{ij}) > t_{min}\}, \text{ and}$$

$$E_{weak} = \{N(x_{ij}) > t_{min} | x_{ij} \text{ neighbors a pixel in } E_{max}\}.$$

approach is the most common of all such methods which identifies class means that are as disparate as possible while, at the same time, creating smooth boundaries between classes [37, 38].

Finally, there are edge-based methods which require prior shape information about the objects to be identified. These include atlas algorithms, which are often used in the medical community to create models or templates for certain organ shapes based on patient data. A new medical image can be analyzed but only by mapping it to an appropriate existing template [11, 39]. Another group of algorithms called active contour models, or snakes, operate similarly to level sets where an initial curve is evolved to fit an object in the image. However, for active contours, the user must know a priori what the curve shape will be and roughly the curve's location before the algorithm can find the exact shape and location [40–42].

2.3.2 Region-Based Methods

The second category of boundary detection techniques are region-based, meaning that they identify homogenous regions in the image as belonging to a single unified shape. Often, these methods will compute the statistics of some region and if the neighboring pixels sufficiently match those statistics, they are added to the region [43]. Some region-based methods, including shadow detection [44] and texture recognition [42, 45, 46], include both region criteria and spatial information and often incorporate gradients, making them particularly useful for separating homogenous regions [20, 33].

Watershed methods use an image's gradient map as a topological surface where regions with large gradients represent peaks and regions of small gradient represent valleys. Imaginary water placed on each pixel flows downhill and pixels whose "water" drains to the same valley are defined to be from the same region [47–49].

Many methods combine features of existing methods in order to provide improved results. The region competition algorithm described in [50] presents a statistical framework for image segmentation that involves a sampling window with the geometric aspects of snake models along with the statistical techniques of region growing and also imposes a minimum description length criterion to force semi-smooth boundaries.

2.3.3 Statistical Methods

Statistical segmentation and edge detection methods utilize measures such as the mean and variance of pixel intensities in order to determine the results. Region growing methods often overlap with this category since they compare each new pixel to the statistics of a region of pixels in order to decide if the pixel should be absorbed into the region or rejected and assigned to another class. However, region growing methods are a small subset of statistical methods.

Some supervised statistical algorithms use intensity distributions of the training data to build classifiers. Based on the statistics of the training pixels, non-training pixels are sorted into the most probable class using Bayes' theorem. Common methods include linear discriminant analysis, which assumes that each class of training data has the same variance, and quadratic discriminant analysis (QDA), which allows

classes to have different variances. Pseudocode for quadratic discriminant analysis is given in Algorithm 5.

Algorithm 5 Quadratic discriminant analysis for image segmentation

Given image X , training data T from C classes, and prior probabilities π_c , for each class $c = 1 \dots C$,

Compute the sample mean μ_c and sample standard deviation σ_c for the training data from each class $c = 1 \dots C$.

For each $x_{ij} \in X$:

1. Use the mapping function G to assign x_{ij} to the class which maximizes the posterior probability of x_{ij} , such that

$$G(x_{ij}) = \arg \max_c P(\text{class} = c | \text{pixel} = x_{ij}) = \arg \max_c \frac{f_c(x_{ij})\pi_c}{\sum_{k=1}^C f_k(x_{ij})\pi_k},$$

where $x_{ij} \sim \mathcal{N}(\mu_{c_{ij}}, \sigma_{c_{ij}})$.

These methods are extremely common in the literature but they assume Gaussian distributions on the global training data. Data that are well-separated in feature space result in better performance of the algorithms [26], which precludes images with low contrast, for example. The work in this dissertation is based on the fundamental ideas of quadratic discriminant analysis. Like QDA, we use statistics of the training data to sort the image’s pixels, however, we do so in a localized, adaptive way, rather than assuming that the training data can be used globally for the entire image.

Other methods such as expectation maximization (EM) iteratively fit Gaussians or Gaussian mixture models to the data in feature space and adjust class assignments to find the best clusters for each class [51]. Recent developments in EM algorithms give non-parametric approaches to this problem [52]. Similar to fuzzy logic methods, EM algorithms allow for each pixel to have partial class assignments.

2.3.4 Neural Networks

Much of the latest work in machine learning has been conducted in the field of artificial neural networks, which are computational models designed to work the way a human brain does [25, 53–56]. Figure 2.2(a) shows a cartoon of a biological neuron, which works when electrical impulses are carried to a neuron. If the impulse is strong enough, the neuron will pass on the electric pulse to other neurons, sending information on through the brain. Similarly, a node in an artificial neural network, as seen in Figure 2.2(b), will receive a weighted numerical input. This value is input to an activation function, commonly an arctangent, sigmoid, or ReLU function, given below.

$$\text{arctangent} : f(x) = \tan^{-1}(x)$$

$$\text{sigmoid} : f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

$$\text{ReLU} : f(x) = \max(0, x)$$

The output of the activation function is sent on to the next layer of nodes or set of “neuron.” This propagation of information through the neural network is called the forward pass. Figure 2.3 shows a diagram of a one-layer neural network. Note that this cartoon has three inputs and two outputs, whereas for imaging applications, the number of inputs and outputs would equal the number of pixels in the image. Each connection between neurons has a weight which is updated with each iteration, so in the case of large images, the number of weights can easily reach into the millions.

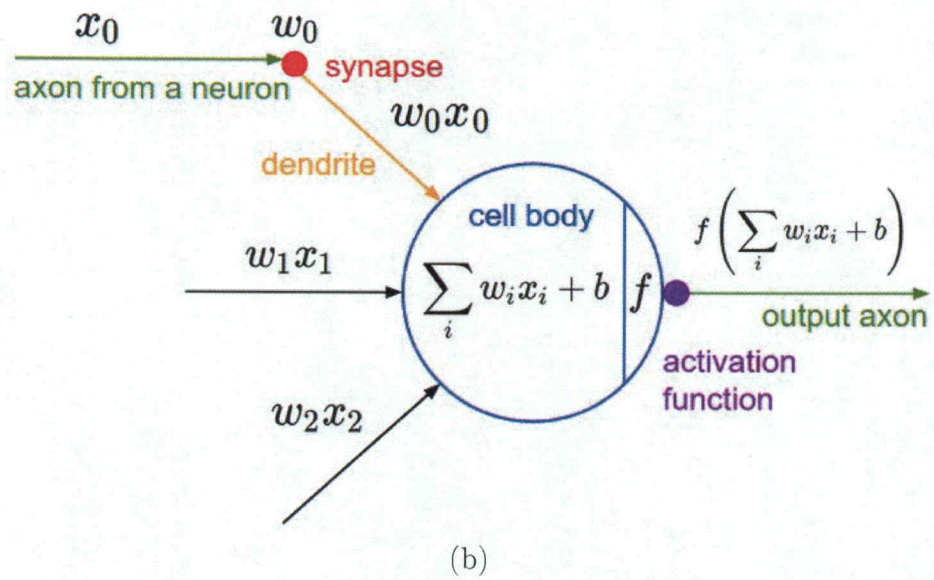
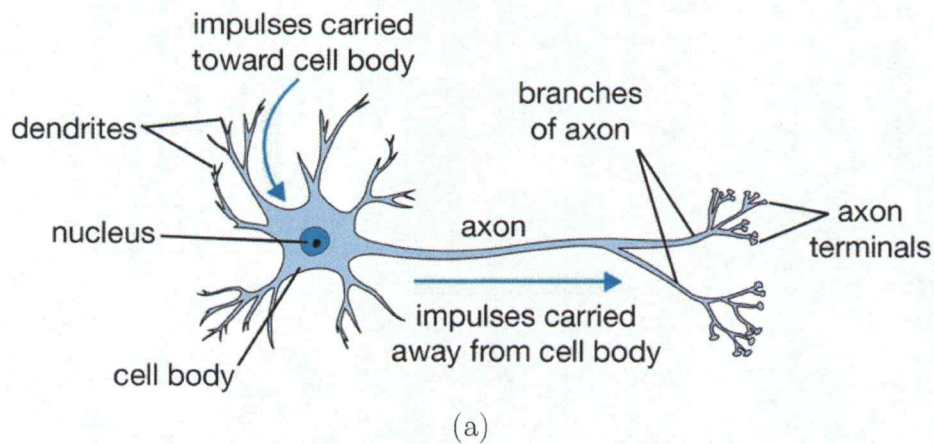


Figure 2.2: (a) Biological neuron and (b) a common mathematical model for a neural network [55]

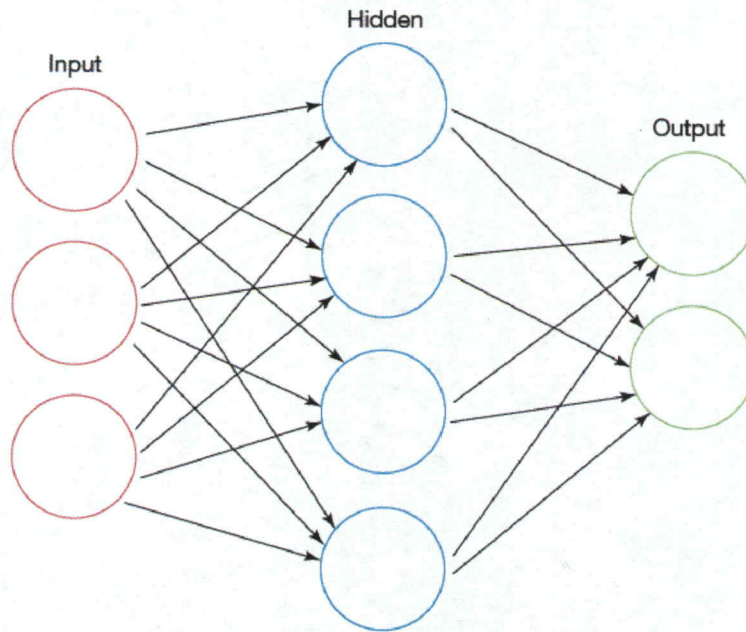


Figure 2.3: Cartoon of a neural network with one hidden layer [57]

Once the forward pass has been completed and values are given at each of the output nodes, the backward pass begins. As neural networks are machine learning algorithms, they must have an update rule through which they improve with each iteration. For most neural networks, this is done via the backpropagation of errors, or simply “backpropagation.” By calculating the derivative of a loss function between the computed outputs and the true outputs, this error can be used to adjust the weights that are present between any two nodes in the network. Alternating between forward and backward passes can be performed until the loss is sufficiently low and the network is sufficiently trained.

Boundary detection using neural networks is usually done using convolutional neural networks, algorithms that involve convolving numerous kernels with images in every layer, similar to how Sobel and Canny algorithms work. However, because

they involve numerous layers and potentially dozens of kernels of all shapes and sizes, neural networks are able to identify shapes and pattern that are not apparent at face value, making them far more powerful. Many complicated neural networks involve more than one hidden layer (with Google’s deep net employing 22 hidden layers). These networks are called deep neural networks and the field is referred to as deep learning. Many researchers are making strides in image processing using these advanced networks [58].

2.4 Uncertainty Quantification Methods

In addition to identifying class and boundary locations, applications benefit from understanding associated uncertainties, with typical measures of error confusion matrices, and kappa statistics [59]. These statistical measures provide an overall assessment of the analysis, but for each of these techniques, applied researchers are interested in knowing both the accuracy and the spatial position where the errors are most likely to occur [60–62].

For neural networks, [54] produces a confidence map indicating the number of voting networks that agree on each pixel’s predicted label, and [63] presents a method for estimating a pixel-scale confidence map when using boosting. Work by [5] takes an alternative approach to developing uncertainty maps by using the spectral domain rather than the spatial domain. Work by [64] presents an information-based criterion for computing a thematic uncertainty measure that describes the overall spatial variation of the segmentation accuracy. Despite all of these advancements, no method is

able to satisfactorily identify boundaries and provide uncertainty quantification for the types of challenging images presented here.

For the majority of images, the literature contains a vast breadth of methods for identifying object boundary locations. Techniques for edge detection and image segmentation are typically tailored for each application to address the challenges specific to each type of image [11, 22, 40, 65, 66]. However, boundary identification is challenging for images with low contrast between classes, heteroskedasticity, and objects whose intensities vary spatially. In such cases, quantitative analysis of the images becomes difficult, and the analyst is often left to manually identify materials.

For images with spatially varying regions, there is still a need for a suitable method that is able to identify boundaries between classes, even if the classes change intensity across an image or have overlapping intensity distributions. In addition to being able to handle these challenging images, an ideal method would be able to provide uncertainty quantification for the results, identifying regions where the algorithm is more or less confident in the given result.

CHAPTER 3

LOCALLY ADAPTIVE DISCRIMINANT ANALYSIS

This chapter presents a supervised statistics-based algorithm for edge detection called locally adaptive discriminant analysis (LADA). Like many other discriminant analysis methods, LADA builds its classifier from user-supplied training data. The novel idea behind LADA, however, is that instead of using all training data at once to build a single classifier for segmenting every pixel, as with global methods, we restrict the training data for each pixel of interest to the locally-occurring classes. These local training data are used to build individual classifiers for placing each pixel into a class, where region boundaries are defined to be the borders between classes.

It can be challenging to detect boundaries in images that contain shadows, heterogeneous illumination, strong gradients across classes, and weak gradients across boundaries. Global segmentation-based methods rely on the classes being well-separated in feature space and do not account for spatial variation in intensity within a class, causing them to fail on images with classes that overlap in intensity. Edge detection methods rely on sharp changes in intensity between classes and sometimes identify non-existent boundaries in images that contain misleading shadows as well as fail to identify boundaries when there are gradual changes between classes. By

restricting the training data for each pixel of interest to only the locally-occurring classes, we get a better representation of the pixel’s true class, without being misguided by the irrelevant information in other areas of the image. The training data are restricted for each pixel in the image via two user-selected parameters: a maximum radius r which limits how far away we look for training data, and a ‘number of neighbors’ parameter n which limits how many nearby training pixels we use from a class. Training pixels outside of each pixel’s subimage are temporarily ignored. The LADA algorithm is described below, with pseudocode in Algorithm 7.

3.1 The LADA Algorithm

Given an image X , a pixel x_{ij} in row i and column j , and the set of training data $T \subset X$, we are interested in determining the class, c , to which x_{ij} most likely belongs, for each $x_{ij} \in X$. The training data have class labels, c , with $c = 1, \dots, C$, for a total of C known classes occurring in X . Let $T_c \subset T$ be the set of all training pixels from class c . Since LADA is a boundary detection method, we assume that the user can identify the majority of the image, sans the boundaries, so the user is asked to provide as much training data as is reasonable.

Rather than considering the entire set T to build a trainer for x_{ij} , we temporarily reduce the training data via two user-selected parameters: r and n . Typically, training data is a single, non-changing set but with this method, the local training data are used and will likely change for each x_{ij} . Given the radius parameter, r , we

define the subimage $A_{ij} \subset X$ about pixel x_{ij} to be

$$A_{ij} = \left\{ x_{kl} \in X \mid \sqrt{(i-k)^2 + (j-l)^2} \leq r \right\},$$

which is the set of all pixels within radius r of pixel x_{ij} . All training data outside A_{ij} are temporarily ignored and have no effect on the classification of x_{ij} . Further reducing the trainer's view of local, the sets of local training data, $T_{c_{ij}} \subset A_{ij}$, are defined to be the sets of at most n nearest training pixels to x_{ij} , for each class c . Here, 'nearest' is defined using Euclidean distance to find the distance between any two pixels, such that

$$d(x_{ij}, x_{kl}) = \sqrt{(i-k)^2 + (j-l)^2}.$$

For each class, order all the training pixels within A_{ij} from smallest to largest by their Euclidean distance to x_{ij} and collect the nearest n pixels for the set of local training data, $T_{c_{ij}}$. Note that ties are included so we could have more than n training pixels from a class. Together, r and n determine our definition of local for LADA.

A simplified visual example of the restriction of the local training data for parameters $r = 3$ and $n = 4$ is given in Figure 3.1. The two training data classes that make up T are shown in pink (class 1) and blue (class 2) colored pixels, with the center pixel identified as x_{ij} . All pixels in subimage A_{ij} (within radius $r = 3$) are shaded, with elements of $T_{1_{ij}}$ being the pink, shaded pixels with demarcation of n_1 and elements of $T_{2_{ij}}$ being the shaded, blue pixels with demarcation of n_2 . Since ties are included, the algorithm is not made to choose between the two blue training

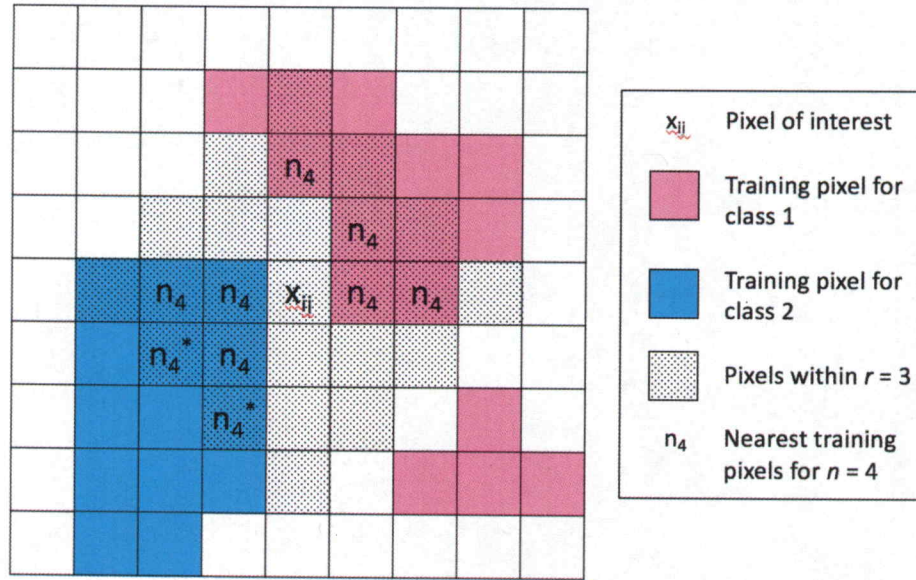


Figure 3.1: An example of defining *local* training data with $r = 3$ and $n = 4$. There are two classes with training data represented by pink and blue pixels. The pixels within A_{ij} are indicated with diagonal gray lines, and the nearest four training points for each class are marked with n_4 . Ties are included so the blue class collects 5 local training pixels instead of 4.

pixels with asterisks, which are equidistant from x_{ij} . Both are included, giving the blue class 5 local training pixels, even with the nearest neighbors parameter $n = 4$.

Each class c with $|T_{c_{ij}}| \geq 3$, where $|T_{c_{ij}}|$ is the number of elements in T_{ij} , is considered a potential class for x_{ij} and the statistics of the locally occurring training data are computed. A set with $|T_{c_{ij}}| < 3$ is possible even with $n \geq 3$ if not enough training pixels of class c occur within A_{ij} , and, in such a case, that class is not considered as a candidate for the pixel's class assignment. At least three training pixels are required so that we can compute a sample standard deviation. The sample mean, $\mu_{c_{ij}}$, and sample standard deviation, $\sigma_{c_{ij}}$, are calculated for each class that is

being considered such that

$$\mu_{c_{ij}} = \frac{1}{|T_{c_{ij}}|} \sum_{x_{kl} \in T_{c_{ij}}} x_{kl}$$

and

$$\sigma_{c_{ij}} = \sqrt{\frac{1}{|T_{c_{ij}}| - 1} \sum_{x_{kl} \in T_{c_{ij}}} |x_{kl} - \mu_{c_{ij}}|^2}.$$

The sample standard deviation is calculated with a normalization factor of $|T_{c_{ij}}| - 1$ because the local training pixels from class c are being used to represent the unknown local class c , and we wish to use an unbiased estimator. This is called Bessel's correction and is required anytime a sample mean is used instead of the true population mean [67].

The local sample mean and local sample standard deviation are used to build a Gaussian probability distribution for each potential class. Therefore, the probability density function for each class $f_c(x)$ is given by

$$f_c(x_{ij}) = \frac{1}{\sigma_{c_{ij}} \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x_{ij} - \mu_{c_{ij}}}{\sigma_{c_{ij}}} \right)^2}.$$

Maximum likelihood estimation is used to determine the class assignment for x_{ij} by finding the class that maximizes the posterior probability of pixel x_{ij} :

$$P(c|x_{ij}) = \frac{f_c(x_{ij})\pi_c}{\sum_{k=1}^C f_k(x_{ij})\pi_k}.$$

Thus each pixel x_{ij} is placed into the class which maximizes the density function. The prior probability of being in each class is denoted by π_c and we assume equal

prior probabilities, as is common in literature [7]. Equivalently, we find the class c where

$$P(x_{ij}|\mu_{c_{ij}}, \sigma_{c_{ij}}) > P(x_{ij}|\mu_{b_{ij}}, \sigma_{b_{ij}}) \forall b \neq c,$$

or the class from which it most likely came. Then a mapping function is used to map each pixel to its segmented class,

$$G(x_{ij}) = \arg \max_c P(x_{ij}|\mu_{c_{ij}}, \sigma_{c_{ij}}) \quad \text{if } |T_{c_{ij}}| \geq 3, \quad \text{where } x_{ij} \sim \mathcal{N}(\mu_{c_{ij}}, \sigma_{c_{ij}}).$$

It is possible that, via choice of r , there are too few training points within subimage A_{ij} to reliably compute a sample standard deviation (i.e., $|T_{c_{ij}}| \leq 2, \forall c$). In such a case, we place x_{ij} into the *bonus class*,

$$G(x_{ij}) = C + 1,$$

indicating there was not enough local information to identify to which of the C classes it belongs. In general, if a significant portion of the image is being placed into the bonus class, the analyst might consider choosing more training data, if more are known, or choosing a larger distance parameter r .

Finally, after every pixel has been placed into a class, the borders between class regions are defined to be the boundaries. Let E_V be a boolean matrix where an entry is true if a vertical edge exists between the pixels, i.e. the neighboring pixels are from different classes, and false if no edge exists between the pixels, i.e. the neighboring

pixels are from the same class:

$$E_V = \left[\text{boolean}(G(x_{ij}) \neq G(x_{ij+1})) \right], \forall x_{ij}, x_{ij+1} \in X.$$

Similarly, we define E_H to be a boolean matrix for the set of horizontal edges where an entry is true if a horizontal edge exists between the pixels and false if no edge exists between the pixel:

$$E_H = \left[\text{boolean}(G(x_{ij}) \neq G(x_{i+1j})) \right], \forall x_{ij}, x_{i+1j} \in X.$$

Together, E_V and E_H describe all of the LADA-determined boundaries. Pseudocode for LADA is given in Algorithm 7.

There are no specific rules for how to choose values for r and n but we provide general guidelines. As long as the user provides training data for a significant portion of the image, the user should choose smaller values for r and n . This is especially important in images with strong spatial variation, where using only the closest training pixels is important for guaranteeing distributions that appropriately represent the pixel of interest. If the user determines too much of the image is placed in the ‘bonus’ class, a larger value for r is recommended so more training pixels can be reached.

It should be noted that as r approaches the bounds of the diagonal distance of the image and n is increased to the cardinality of the largest training data set, the effect of looking at local training data diminishes and the focus becomes global. In such a case, this algorithm converges to quadratic discriminant analysis [26]. The al-

Algorithm 6 Locally adaptive discriminant analysis

Given image X , define training data T with C classes, where T_c is the set of all training pixels for a class c , (i.e. $\bigcup_c T_c = T$ and $\bigcap_c T_c = \emptyset$), and *local* parameters r and n .

For each pixel $x_{ij} \in X$:

1. Define A_{ij} to be set of all training pixel locations centered at x_{ij} , with radius r such that

$$A_{ij} = \left\{ x_{kl} \in X \mid \sqrt{(i-k)^2 + (j-l)^2} \leq r \right\}.$$

2. For $c = 1$ to C :
 - a. Define the local training data $T_{c_{ij}}$ to be the set of the n nearest training points to x_{ij} within A_{ij} that belong to class c , with ties included.
 - b. Compute the sample mean $\mu_{c_{ij}}$ and sample standard deviation $\sigma_{c_{ij}}$ of $\{T_{c_{ij}}\}$ for every $T_{c_{ij}}$ with $|T_{c_{ij}}| \geq 3$.
3. Place x_{ij} into class c for which

$$G(x_{ij}) = \left\{ \begin{array}{ll} \arg \max_c P(x_{ij} | \mu_{c_{ij}}, \sigma_{c_{ij}}) & \text{if } |T_{c_{ij}}| \geq 3 \\ C + 1 & \text{otherwise} \end{array} \right\}, \quad \text{where } x_{ij} \sim \mathcal{N}(\mu_{c_{ij}}, \sigma_{c_{ij}}).$$

After every pixel has been placed into a class, the borders between class regions are defined to be the boundaries. The set of all vertical boundaries is stored in the matrix E_V and the set of all horizontal boundaries is stored in matrix E_H where

$$E_V = \left[\text{boolean}(G(x_{ij}) \neq G(x_{ij+1})) \right], \quad \forall x_{ij}, x_{ij+1} \in X,$$

and

$$E_H = \left[\text{boolean}(G(x_{ij}) \neq G(x_{i+1j})) \right], \quad \forall x_{ij}, x_{i+1j} \in X.$$

gorithm may be restricted further to perform similarly to linear discriminant analysis if $\sigma_{c_{ij}} = \sigma_{ij}$ for all considered classes c .

3.2 LADA Example

In this section, LADA is demonstrated on an image of a noisy staircase. Figure 3.2 shows a 450×450 pixel image of a noisy staircase displaying varying intensities

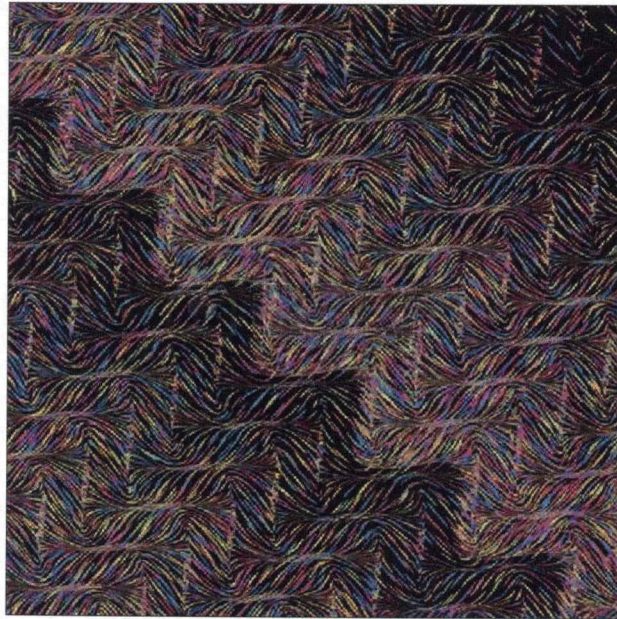


Figure 3.2: A synthetic RGB image displaying high levels of noise and gradients across classes.

across the upper right and lower left classes. Pixels in the upper right region tend to increase in intensity from the outside to the staircase while pixels in the lower left region tend to decrease in intensity from the outside to the staircase.

Figure 3.3 shows the user-defined training data, where the pink and tan regions are training data and the black region is where no training data have been selected. For this image, there are 18,332 pixels, or 9.05%, that have not been selected as training data and the average gap between training regions is 20 pixels wide. As LADA is a method for boundary detection, it is reasonable to assume the user has knowledge about the majority of the image, sans the boundaries. Images such as this

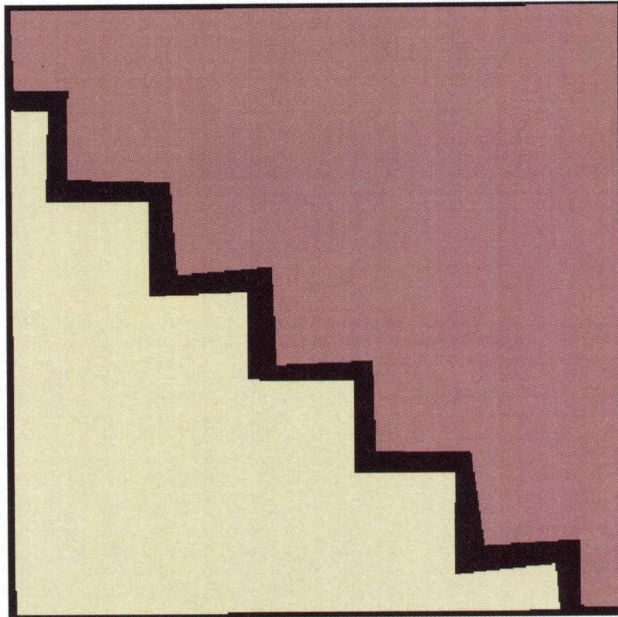


Figure 3.3: The user-selected training data for two classes. Black regions are where no training data have been selected.

are difficult for edge detection methods because of the high noise level, classes with overlapping intensities, and spatially varying class intensities.

In this example, parameter values of $n = 30$ and $r = 15$ are used. Figure 3.4 shows examples of the circular subimages for five pixels in the image. Using the local training data, each pixel is placed in a class using discriminant analysis. Figure 3.5 shows the final segmentation, where each pixel has been assigned to one of the two user-defined classes or to the bonus class. Note that the bonus class is only chosen in regions where a pixel was more than $r = 15$ pixels away from the nearest training pixels. To prevent pixels from being assigned to the bonus class, the user could choose more training data in those regions or choose a large r value.

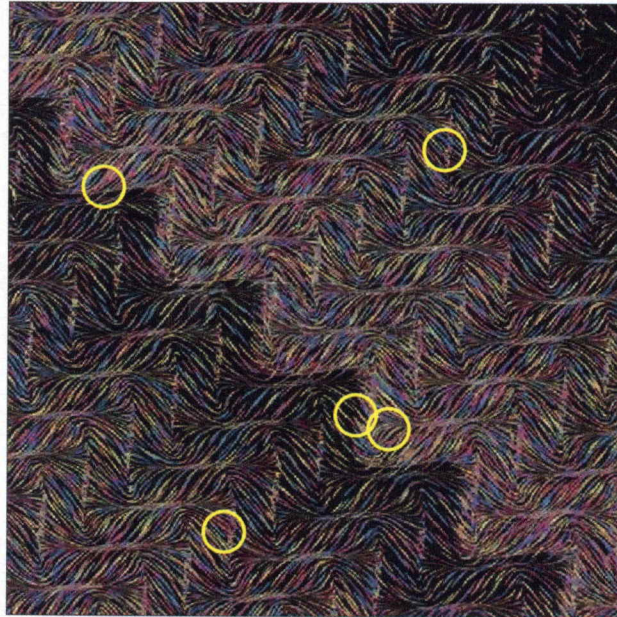


Figure 3.4: The original image with examples of circular subimages for five pixels in the image.

The set of all boundaries between two pixels belonging to different classes is defined to be the set of LADA-determined boundaries. Figure 3.6 shows the LADA-determined boundaries. For clarity and completeness, the LADA boundaries are also shown superimposed on the original image in Figure 3.7.

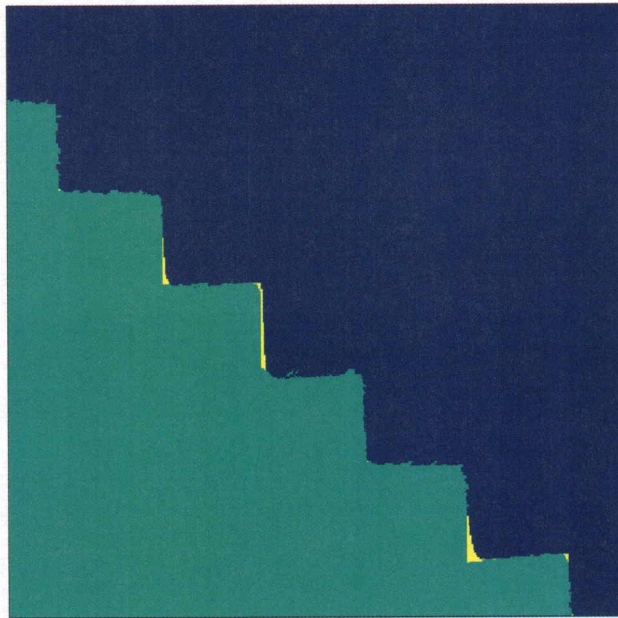


Figure 3.5: The LADA segmentation, where each pixel has been assigned to one of the two given classes (blue and green) or to the bonus class (yellow).

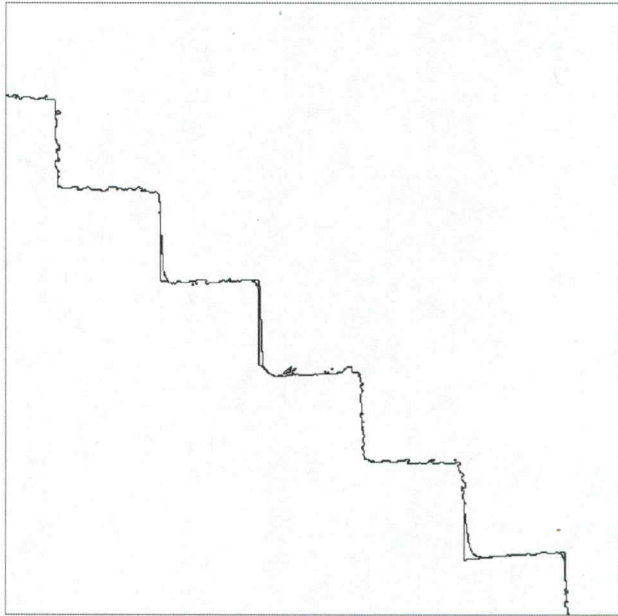


Figure 3.6: The LADA-determined boundaries, found using the above segmentation.

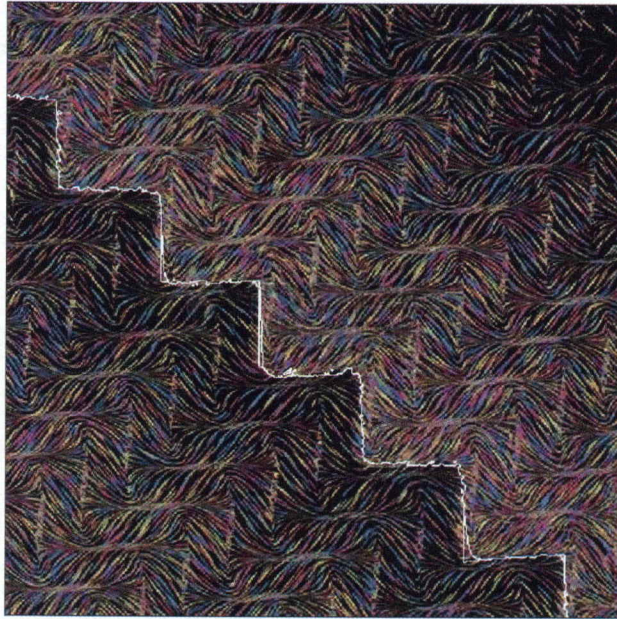


Figure 3.7: The original image with the ALADA-determined boundaries superimposed on top.

CHAPTER 4

ANISOTROPIC LOCALLY ADAPTIVE DISCRIMINANT ANALYSIS

In this chapter, we present another new method for boundary identification in images: anisotropic locally adaptive discriminant analysis (ALADA) which is an extension of the LADA algorithm presented in Chapter 3. ALADA is motivated by the fact that one size subimage may not be appropriate across an entire image. If the image has both large gaps and small gaps between training regions, there should be a way to adapt the size (and in this case, shape and direction) of the subimage, based on the available training data. As with LADA, the ALADA algorithm is a local, supervised, statistics-based method, derived from a discriminant analysis approach. Local training data are used to build a classifier for assigning a class to each pixel, where region boundaries are considered to be the borders between classes. The novelty of this algorithm comes from how local training pixels are downselected. The training data used to assign a pixel's class are collected from the regions of the image surrounding the pixel, where the size, eccentricity, and direction of the subimage are varied, based on the training data and a single, insensitive, user-supplied parameter.

ALADA retains the basic structure of the LADA algorithm and adds to it the concept that in order to assign a pixel near a suspected boundary to a class,

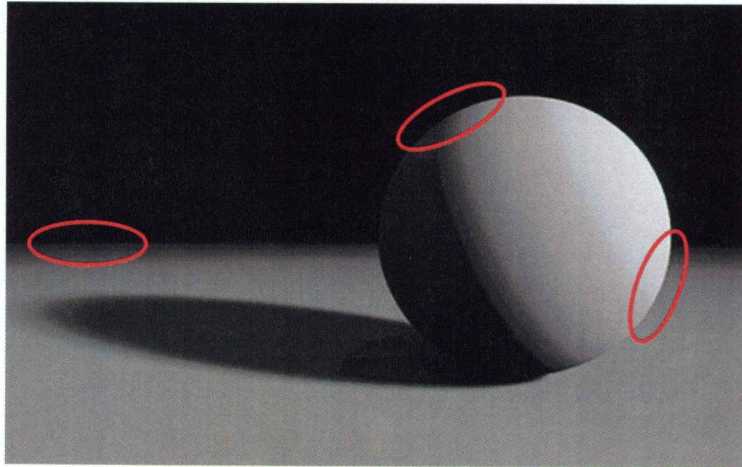


Figure 4.1: An example of ALADA ellipses superimposed on an image, demonstrating how the elliptical training data are collected along suspected boundary lines.

the pixels along that boundary are likely the most informative. This is because they are typically the best representatives of the behavior at the boundary surrounding the pixel of interest. Therefore, for pixels near a suspected boundary, we collect all local training data and then downselect it in an elliptical shape, with the major axis oriented in the direction of least variation, i.e. along the suspected boundary for each pixel. Data along a boundary are assumed to be most representative of a pixel's class, because many objects' pixel representation is consistent near the boundary but may differ as distance from the boundary increases. In such cases, this is often due to lighting and shadows in the image, or, in the case of penetration imaging, density of the object being imaged. A visualization of what these ellipses might look like are given in Figure 4.1 and their generation will be discussed.

As previously mentioned, while the pixel intensities from a class may vary across an image, they are much more consistent along a boundary. Take the right-

most ellipse in Figure 4.1, for example. Figure 4.2 shows examples of three possible local subimages, each of which would provide slightly different training data. When training data are collected in an ellipse along the boundary, densities of the intensity values are better separated in feature space. For example, the training data distributions in Figure 4.2 have sample means of .5515 and .7123 for training data taken across the boundary (top left, yellow distributions), sample means of .5248 and .7406 for training data taken from a circular region (top center, peach distributions), and sample means of .5108 and .7622 for training data taken along the boundary (top right, blue distributions).

The eccentricity of the ellipse, or how far it deviates from a circle, is also determined by the local training data. When training data are all found in a specific direction, the eccentricity approaches 1, giving a more flattened elliptical shape. When the pixel of interest is surrounded by training data, the eccentricity approaches 0, giving a circular or near-circular shape, as there is no evidence for collecting training data in a certain direction.

While LADA's parameters prove successful on many images, ALADA goes a step further and is a much more adaptive algorithm where each pixel's selected local training data are custom-tailored based on the user-provided information. The minor axis of each ellipse is determined by an insensitive user-selected parameter; pixels surrounded by training data require a smaller minor axis while pixels near a suspected boundary require a larger minor axis, to collect training data from a larger region. The major axis length and orientation are determined by the statistical relationship of local training pixel locations. Training data-dense regions will yield a smaller major

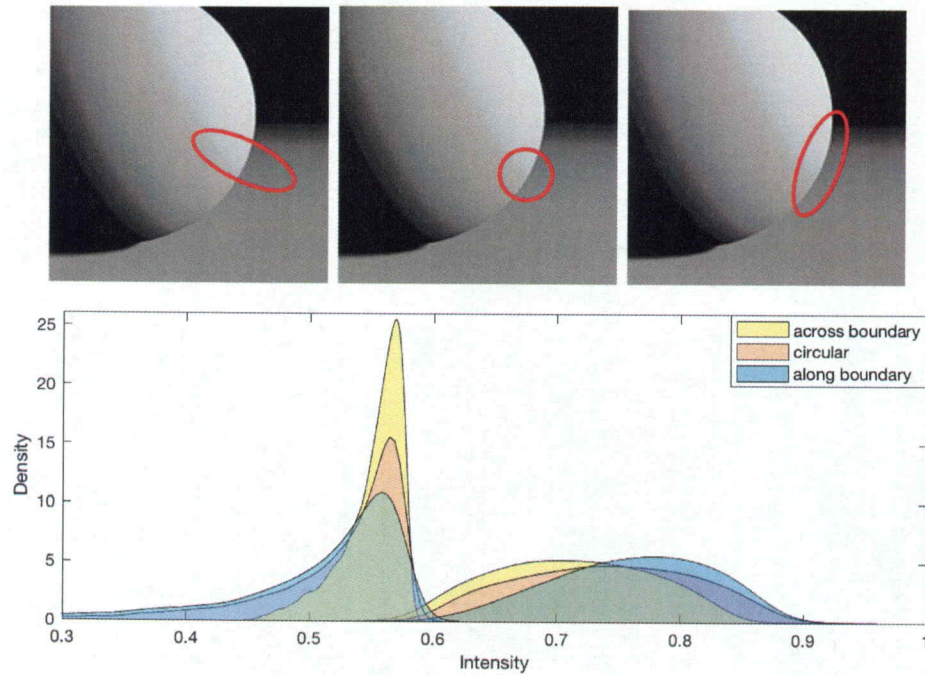


Figure 4.2: Examples of intensity densities of (top left, yellow distributions) elliptical training data across the boundary, (top center, peach distributions) circular training data, and (top right, blue distributions) elliptical training data along the boundary.

axis length such that the subimage approaches a circle, while regions near suspected boundaries will produce a major axis that is larger (sometimes *much* larger) than the corresponding minor axis and, as described before, these ellipses will lie with the major axis along the suspected boundary. These anisotropic local training data are then used to build local Gaussian distributions and maximum likelihood estimation is used to determine the most probable class for the pixel of interest.

4.1 The ALADA Algorithm

Given an image X , pixel x_{ij} in row i and column j , and a set of training data pixels $T \subset X$, we are interested in determining the class, c , to which x_{ij} most likely

belongs, for each $x_{ij} \in X$. The training data have class labels with $c = 1, \dots, C$, for a total of C known classes occurring in X . Let $T_c \subset T$ be the set of all training pixels with class label c .

The user provides a value for the ‘number of neighbors’ parameter, n , which effectively determines the anisotropic local training data for each pixel in the image. By design, we will choose a subset of T for x_{ij} such that at least n training pixels will occur for at least one class in the subset we collect. Requiring more than one class to have at least n training pixels could cause pixels surrounded by training data from a single class to look far away to collect n training pixels from a second class, causing the algorithm to no longer be local. We use a k -nearest neighbors algorithm on the training data T to first determine the n nearest training pixels (using Euclidean distance) from each class for pixel x_{ij} [68]. See Appendix A.4 for more information. The smallest radius needed to reach n training pixels from a class is identified to be the minor radius, $r_{1_{ij}}$, of the ellipse to be constructed, where

$$r_{1_{ij}} = \arg \min_r \left(\left| \{x_{kl} \in T_c \mid \sqrt{(i-k)^2 + (j-l)^2} \leq r\} \right| \geq n \right),$$

for some c . As with LADA, we collect all training pixels within the minor radius $r_{1_{ij}}$ so in the case of a tie, there may be more than n pixels from a class.

This value for $r_{1_{ij}}$ will be the minor radius of our elliptical subimage but it is temporarily used as the radius of a circular subimage. Once $r_{1_{ij}}$ is computed, define A_{ij} to be the set of row, k , and column, l , pairs for each training pixel within radius

$r_{1_{ij}}$ of pixel x_{ij} ,

$$A_{ij} = \left\{ (k, l) \mid x_{kl} \in T, \sqrt{(i-k)^2 + (j-l)^2} \leq r_{1_{ij}} \right\}.$$

These locations of the training data subset are used to calculate the major radius of the ellipse and its tilt, or angle, relevant to x_{ij} and the suspected nearby boundary.

To work towards computing the tilt, or angle, principal component analysis (PCA) is implemented to calculate the angle of least variation [69]. PCA considers the (k, l) coordinates of each locally identified training pixel in A_{ij} . By calculating the 2×2 covariance matrix of A_{ij} , we then compute its eigenvalues, $\lambda_{1_{ij}}$ and $\lambda_{2_{ij}}$, where $\lambda_{1_{ij}} \geq \lambda_{2_{ij}}$. Recall that the values of the covariance matrix represent the data's spread along the horizontal and vertical directions while the eigenvalues of the covariance matrix represent the data's spread along the eigenvector directions. The spread of the ellipse's axes should reflect the spread of the local training data so we set the ratio of the major axis to the minor axis equal to the ratio of the major direction of spread to the minor direction of spread:

$$\frac{r_{2_{ij}}}{r_{1_{ij}}} = \frac{\lambda_{1_{ij}}}{\lambda_{2_{ij}}}.$$

The major radius of the training data ellipse for x_{ij} , $r_{2_{ij}}$, is computed such that

$$r_{2_{ij}} = r_{1_{ij}} \frac{\lambda_{1_{ij}}}{\lambda_{2_{ij}}}.$$

Regions of the image saturated with training data will have $\lambda_{1_{ij}} = \lambda_{2_{ij}}$ since the training data locations in A_{ij} are uncorrelated, giving an $r_{2_{ij}}$ value equal to $r_{1_{ij}}$, i.e. the elliptical subimage is reduced to a circle, as in the LADA algorithm. Such an occurrence is most common for x_{ij} in the middle of a region of training data, far from a suspected boundary. Regions where all the training data are found in one direction, i.e. along the sides of the image, will have $\lambda_{1_{ij}} \approx 1$ and $\lambda_{2_{ij}} \approx 0$, resulting in a ratio that approaches infinity. The user may choose to set a maximum value for the ratio to prevent this. If no maximum ratio value is set and the calculated major radius approaches the dimensions of the image, the elliptical subimage shape will approach that of a band of width $r_{1_{ij}}$ that extends across the entire image in one direction.

The final value we compute using the eigenvalues and eigenvectors of A_{ij} is θ_{ij} , the assumed angle, or direction, of least variation. Note that PCA is typically used to find the principal component, or the direction of greatest variation, which lies normal to the boundary. PCA eigenvectors are, by definition, orthonormal, so by taking the eigenvector associated with the smaller eigenvalue, we obtain the direction along the suspected boundary instead of across it. Let $e_{\lambda_{2_{ij}}}$ be the eigenvector associated with the smaller eigenvalue,

$$e_{\lambda_{2_{ij}}} = \begin{bmatrix} e_{1_{ij}} \\ e_{2_{ij}} \end{bmatrix},$$

which describes the direction in which the local training data locations have smallest variance, i.e. along suspected boundaries. Note that $e_{1_{ij}}$ is a vertical distance across the rows and $e_{2_{ij}}$ is a horizontal distance across the columns. Using basic

trigonometry, we compute θ_{ij} from $e_{\lambda_{2ij}}$ such that

$$\theta_{ij} = \arctan\left(\frac{e_{1ij}}{e_{2ij}}\right)$$

where θ_{ij} is how far we want to rotate the ellipse so that it lies along the assumed boundary.

The anisotropic (elliptical) subimage is found using the equation for a shifted, rotated ellipse at an angle of θ_{ij} with minor radius r_{1ij} and major radius r_{2ij} and stored in set B_{ij} ,

$$B_{ij} = \left\{ x_{kl} \mid \frac{[(k-i)\sin(\theta_{ij}) + (l-j)\cos(\theta_{ij})]^2}{r_{1ij}^2} \dots + \frac{[(k-i)\cos(\theta_{ij}) - (l-j)\sin(\theta_{ij})]^2}{r_{2ij}^2} \leq 1 \right\}.$$

Let $T_{c_{ij}}$ be the set of training data pixels from B_{ij} that belong to class c such that

$$T_{c_{ij}} = \{x_{kl} \in (B_{ij} \cap T_c)\}.$$

Define $|T_{c_{ij}}|$ be the number of elements in $T_{c_{ij}}$. Each class where $|T_{c_{ij}}| \geq 3$ is considered as a potential class for x_{ij} . Classes that are not found in the anisotropic subimage, or that have fewer than three pixels in the subimage are ignored and will have no effect on the segmentation of x_{ij} . A set with $|T_{c_{ij}}| < 3$ is possible if not enough training pixels of class c occur within B_{ij} , but at least three training pixels are required so that we can compute a sample standard deviation.

The sample mean and sample standard deviation for each known local class are used to build probability distributions. A Gaussian distribution is assumed for local training data so, for each class, the probability density function $f_c(x)$ is given by

$$f_c(x_{ij}) = \frac{1}{\sigma_{c_{ij}} \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{x_{ij} - \mu_{c_{ij}}}{\sigma_{c_{ij}}} \right)^2}.$$

Maximum likelihood estimation is used to determine the class assignment for x_{ij} by finding the class c that maximizes the posterior probability of class c , given pixel x_{ij} :

$$P(c|x_{ij}) = \frac{f_c(x_{ij})\pi_c}{\sum_{k=1}^C f_k(x_{ij})\pi_k},$$

such that each pixel x_{ij} is placed into the class that maximizes the density function. Again, we assume equal prior probabilities, as is common in literature [7]. Equivalently, we use the mapping function $G : x_{ij} \rightarrow c$ to find the class from which x_{ij} most likely came, where

$$G(x_{ij}) = \arg \max_c P(x_{ij} | \mu_{c_{ij}}, \sigma_{c_{ij}}).$$

This process is repeated for all x_{ij} .

Note that ALADA does not have the ‘bonus’ class that LADA uses for pixels that do not have any classes with at least three local training pixels. This is because each ALADA subimage is constructed to have at least one class with $n \geq 3$ training pixels so there is always at least one class being considered.

Finally, after every pixel has been placed into a class, the borders between class regions are defined to be the boundaries. Let E_V be a boolean matrix where an entry

is true if a vertical edge exists between the pixels, i.e. the neighboring pixels are from different classes, and false if no edge exists between the pixels, i.e. the neighboring pixels are from the same class:

$$E_V = \left[\text{boolean}(G(x_{ij}) \neq G(x_{ij+1})) \right], \forall x_{ij}, x_{ij+1} \in X.$$

Similarly, we define E_H to be a boolean matrix for the set of horizontal edges where an entry is true is a horizontal edge exists between the pixels and false if no edge exists between the pixel:

$$E_H = \left[\text{boolean}(G(x_{ij}) \neq G(x_{i+1j})) \right], \forall x_{ij}, x_{i+1j} \in X$$

Together, E_V and E_H describe all of the ALADA-determined boundaries. Note that since E_V and E_H represent the boundaries between pixels, the matrix sizes are one row and one column smaller than the image itself. Pseudocode for ALADA is given in Algorithm 8.

As with LADA, there are no specific rules for how to choose parameter $n \geq 3$ but in general, the user should choose a small value, usually under 100, to ensure that only the closest training pixels are being used to build classifiers. As n approaches the cardinality of the largest training data set, T_c , ALADA grows less ‘local’ and more ‘global.’

Algorithm 7 Anisotropic locally adaptive discriminant analysis

Given image X , define training data T with C classes, where T_c is the set of all training pixels for class c (i.e. $\bigcup_c T_c = T$ and $\bigcap_c T_c = \emptyset$), and local parameter n .

For each pixel $x_{ij} \in X$:

1. Compute radius $r_{1_{ij}}$, defined to be the minimum r such that

$$r_{1_{ij}} = \arg \min_r \left(\left| \{x_{kl} \in T_c \mid \sqrt{(i-k)^2 + (j-l)^2} \leq r\} \right| \geq n \right) \text{ for some } c.$$

2. Define A_{ij} to be the set of all training pixel locations centered at x_{ij} , within radius $r_{1_{ij}}$ such that

$$A_{ij} = \left\{ (k, l) \mid x_{kl} \in T, \sqrt{(i-k)^2 + (j-l)^2} \leq r_{1_{ij}} \right\}.$$

3. Perform principal component analysis on A_{ij} to find major radius and angle of least variation:

- a. Compute the covariance matrix of A_{ij} and its corresponding eigenvalues, $\lambda_{1_{ij}}$ and $\lambda_{2_{ij}}$, with $\lambda_{1_{ij}} \geq \lambda_{2_{ij}}$, and the eigenvector associated with the smaller eigenvalue, $e_{\lambda_{2_{ij}}} = \begin{bmatrix} e_{1_{ij}} \\ e_{2_{ij}} \end{bmatrix}$.

- b. Define $r_{2_{ij}}$ to be the major radius such that

$$r_{2_{ij}} = r_{1_{ij}} \frac{\lambda_{1_{ij}}}{\lambda_{2_{ij}}}.$$

- c. Define θ_{ij} to be the angle of least variation, the direction given by $e_{\lambda_{2_{ij}}}$ such that

$$\theta_{ij} = \arctan \left(\frac{e_{1_{ij}}}{e_{2_{ij}}} \right).$$

4. Compute B_{ij} , defined to be the elliptical subimage of X centered at x_{ij} with minor radius $r_{1_{ij}}$ and major radius $r_{2_{ij}}$ at an angle θ_{ij} , as

$$B_{ij} = \left\{ x_{kl} \mid \frac{[(k-i) \sin(\theta_{ij}) + (l-j) \cos(\theta_{ij})]^2}{r_{1_{ij}}^2} + \frac{[(k-i) \cos(\theta_{ij}) - (l-j) \sin(\theta_{ij})]^2}{r_{2_{ij}}^2} \leq 1 \right\}.$$

Algorithm 8 continued

5. For $c = 1$ to C :

a. Define the local training data set

$$T_{c_{ij}} = \{x_{kl} \mid x_{kl} \in B_{ij} \text{ and } x_{kl} \in T_c\}.$$

b. Compute the sample mean $\mu_{c_{ij}}$ and sample standard deviation $\sigma_{c_{ij}}$ of $\{T_{c_{ij}}\}$ for every T_c with $|T_c| \geq 3$ where

$$\mu_{c_{ij}} = \frac{1}{|T_{c_{ij}}|} \sum_{x_{kl} \in T_{c_{ij}}} x_{kl}$$

and

$$\sigma_{c_{ij}} = \sqrt{\frac{1}{|T_{c_{ij}}| - 1} \sum_{x_{kl} \in T_{c_{ij}}} |x_{kl} - \mu_{c_{ij}}|^2}.$$

6. Place x_{ij} into class c for which

$$G(x_{ij}) = \arg \max_c P(x_{ij} \mid \mu_{c_{ij}}, \sigma_{c_{ij}}), \text{ where } x_{ij} \sim \mathcal{N}(\mu_{c_{ij}}, \sigma_{c_{ij}}).$$

After every pixel has been placed into a class, the borders between class regions are defined to be the boundaries. The set of all vertical boundaries is stored in the matrix E_V and the set of all horizontal boundaries is stored in matrix E_H where

$$E_V = [\text{boolean}(G(x_{ij}) \neq G(x_{ij+1}))], \forall x_{ij}, x_{ij+1} \in X,$$

and

$$E_H = [\text{boolean}(G(x_{ij}) \neq G(x_{i+1j}))], \forall x_{ij}, x_{i+1j} \in X.$$

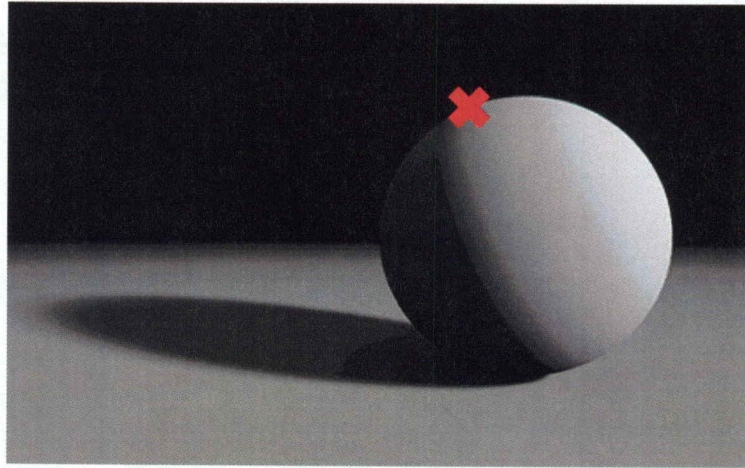
4.2 Visualizing ALADA

A simplified version of the local anisotropic training data collection process is demonstrated in Figure 4.3. The original image, shown in Figure 4.3(a) is of a grayscale sphere on a table. The shadows on the sphere and table regions cause strong changes in gradient within both classes, providing a bimodal intensity distribution on a global scale for either class, which, if training data is taken from both modes, would

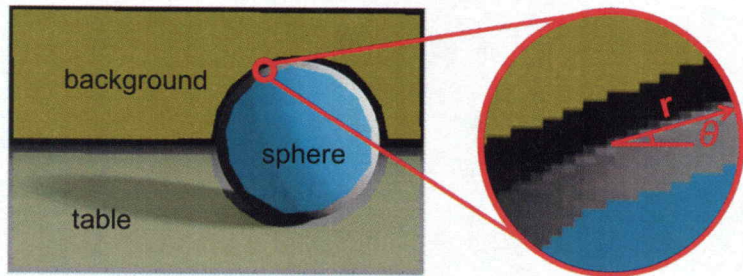
fail the Gaussian assumption of common global techniques such as linear discriminant analysis and quadratic discriminant analysis. Gradient based methods will identify edges between all the well-lit and darkened regions, even though the user knows a priori that some of those are not boundaries between materials. While our algorithm could be applied to every pixel in the image, we demonstrate the training data downselection process for pixel $x_{58,267}$, marked by a red 'x,' which has an intensity value of .60.

The user-selected training data for three classes, 'background,' 'table,' and 'sphere,' are demonstrated in Figure 4.3(b), superimposed over the original image. Given $n = 75$, a radius of $r_{158,267} = 15$ pixels is calculated. Figure 4.3(b) shows an enlargement of the circular subimage about $x_{58,267}$ with radius $r_{158,267} = 15$ pixels. The green and blue regions identify the local training data and their locations will be used for principal component analysis. From PCA and subsequent calculations, we obtain the major radius, $r_{258,267} = 45$ pixels and the angle of least variation, $\theta_{58,267} = .40$ radians. Figure 4.3(c) shows the elliptical subimage, $D_{58,267}$.

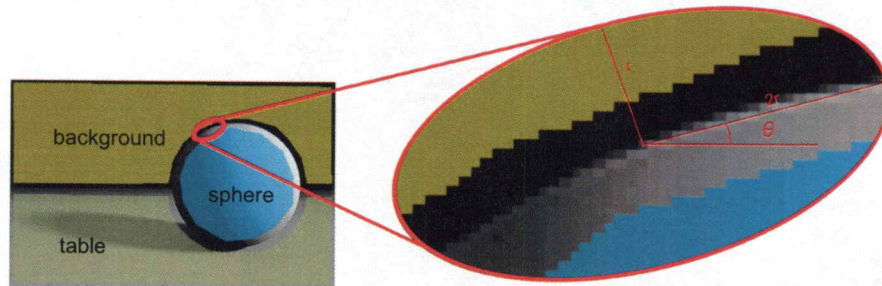
Figure 4.4(a) shows the intensity distributions for the training pixels from the entire image shown in Figure 4.3(b), left. Global image segmentation methods that compare the pixel intensity to Gaussian fits of all the training data would place this pixel into the 'table' class because, with an intensity of .60, the table class has the largest likelihood of being correct, despite the user's a priori knowledge that this pixel is nowhere near the table. Figure 4.4(b) shows the intensity distributions of the anisotropic local training data for $x_{58,267}$ that have been downselected in Figure 4.3(c). Only the 'background' and 'sphere' classes are considered because they



(a)

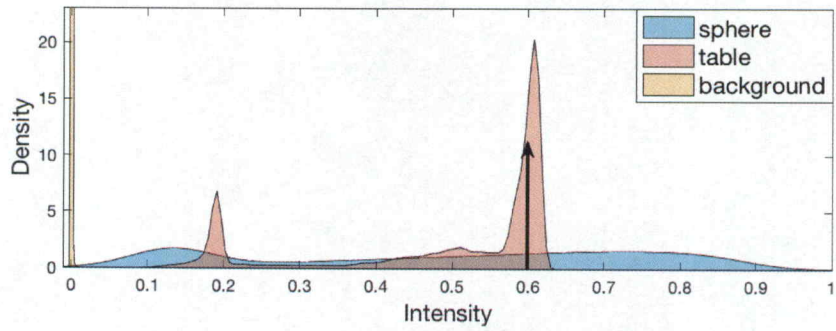


(b)

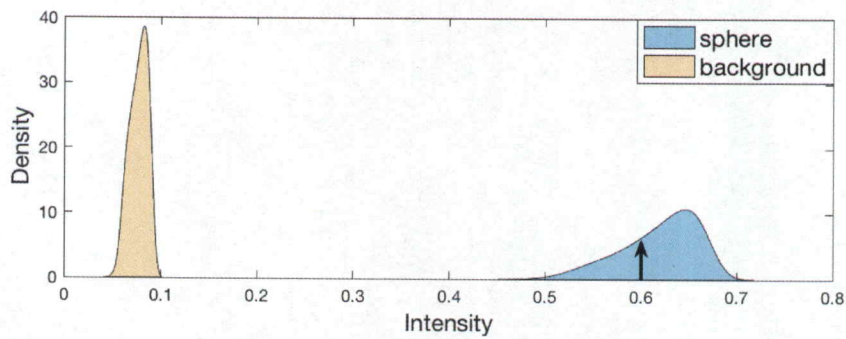


(c)

Figure 4.3: (a) A cartoon image of a sphere on a table with pixel of interest, $x_{58,267}$, marked by a red 'x.' (b) The user-selected training data for three classes and the training data within radius $r_{158,267}$ of the pixel of interest are displayed. The major radius, $r_{258,267}$, and the angle of least variation $\theta_{58,267}$ are calculated and demonstrated in (c), where anisotropic local training data are those located within the ellipse.



(a)



(b)

Figure 4.4: (a) Intensity distributions for the full set of training data pixels and (b) intensity distributions of training data pixels that are anisotropically local to $x_{58,267}$ from Figure 4.3(c), right. In both plots, the intensity of $x_{58,267}$ is indicated with an arrow.

are the only classes present in the anisotropic local data. From these intensity distributions, we would fit Gaussian distributions, using the sample mean and sample standard deviation of each local class's training data. Since Gaussian assumptions are more reasonable at the local level, the fitted distributions differ minimally from the original intensity distributions. Restricting the training data in this way makes it clear that pixel $x_{58,267}$ is best represented by the 'sphere' class.

4.3 ALADA Example

In this section, ALADA is demonstrated on an image of a synthetic noisy triangle, for which the true boundaries are known. Figure 4.5 shows a 200×200 pixel image of a triangle sitting on a surface. The image was created with intensity values of -0.5 , 0 , and 0.5 , with additive Gaussian noise generated from a standard normal distribution added to each pixel. Finally, a gradient was added to give the classes overlapping intensities in feature space. Pixels in the background region tend to increase in intensity from top to bottom while pixels in the triangle and lower region tend to decrease in intensity from top to bottom. While simple in construction, images such as this are difficult for edge detection methods because of the high noise level, heteroskedasticity, the gradual changes between classes, classes with overlapping intensities, and spatially varying class intensities, particularly in the regions where all three classes meet.

Figure 4.6 shows the user-defined training data, where the pink, tan, and white regions are training data and the black region is where no training data have been selected. For this image, there are 3110 pixels that have not been selected as training data and the average gap between training regions is 7 pixels wide. As ALADA is a method for boundary detection, it is reasonable to assume the user has knowledge about the majority of the image, sans the boundaries.

In this example, a parameter of $n = 10$ is used. Figure 4.7 shows a map of $r_{1_{ij}}$ values, calculated for each pixel using n and the user-defined training data. For this image, the minimum minor radius is $r_{1_{ij}} = 2$ and is found in regions of solid training

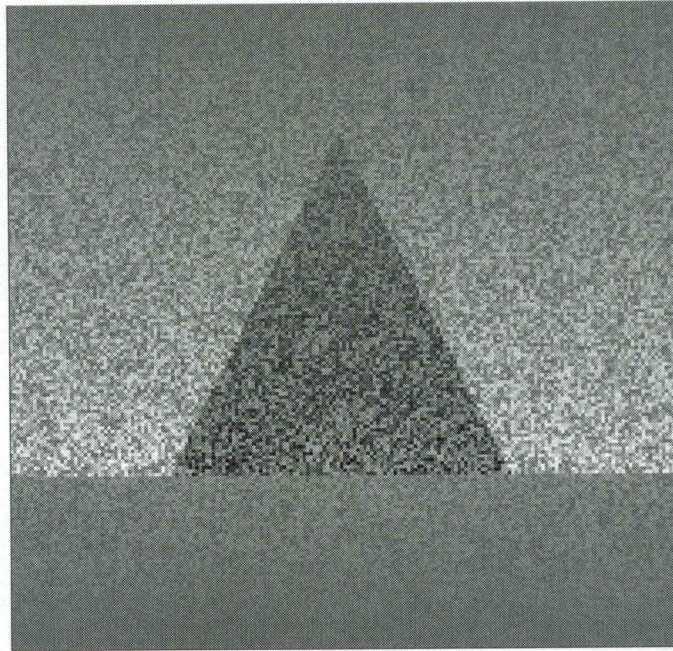


Figure 4.5: A synthetic grayscale triangle image displaying high levels of noise and gradients across classes.

data, and the maximum minor radius is $r_{1_{ij}} = 5.8310$ and is found at the widest gap between training data regions, at the top of the triangle. PCA is performed on the training data within the $r_{1_{ij}}$ radius for each pixel, and the ratio of eigenvalues is found, as given in Figure 4.8(a). Note that even though these eigenvalue ratios have values up to 13.0286, in practice we have chosen to cap this value at 4, to limit the size of the elliptical subimages. Figure 4.8(b) shows the $r_{2_{ij}}$ values for each pixel, computed by multiplying each $r_{1_{ij}}$ by the corresponding eigenvalue ratio.

Performing PCA on the training data within the $r_{1_{ij}}$ radius for each pixel also gives the information needed to find the angle of least variation for each pixel, shown in Figure 4.9.

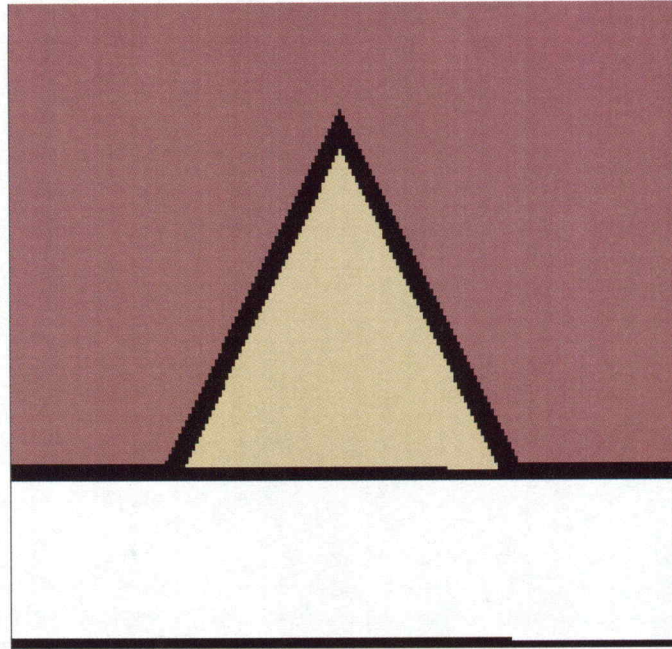


Figure 4.6: The user-selected training data for three classes. Black regions are where no training data have been selected.

Using r_{1ij} , r_{2ij} , and θ_{ij} , the anisotropic subimage is found for every pixel in the image. Figure 4.10 shows examples of the anisotropic subimages for five pixels in the image. Using the anisotropic, local training data, each pixel is placed in a class using discriminant analysis. Figure 4.11 shows the final segmentation, where each pixel has been assigned to one of the three classes.

The set of all boundaries between two pixels belonging to different classes is defined to be the set of ALADA-determined boundaries. Figure 4.12(a) shows the true boundaries for the synthetic image while Figure 4.12(b) shows the ALADA-determined boundaries. For clarity and completeness, the ALADA boundaries are also shown superimposed on the original image in Figure 4.13.

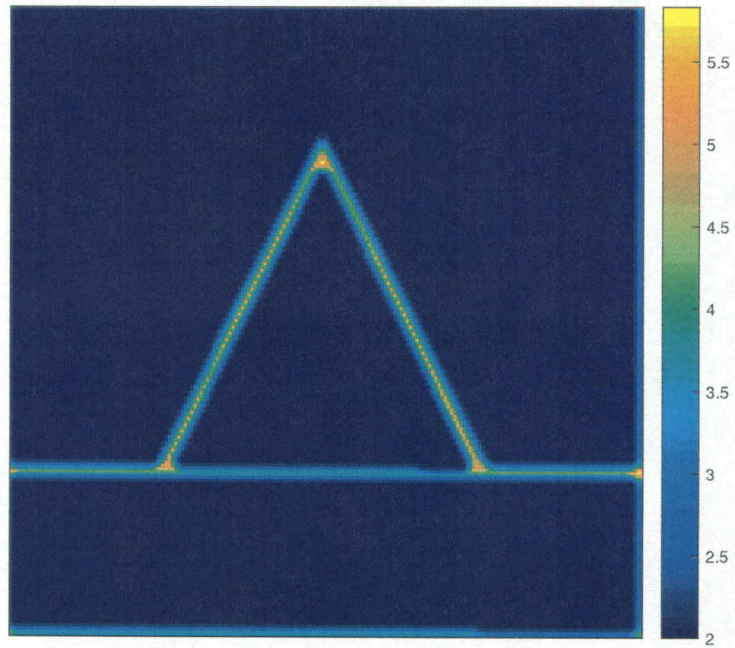


Figure 4.7: A map of $r_{1_{ij}}$ values, calculated for each pixel using parameter n and training data.

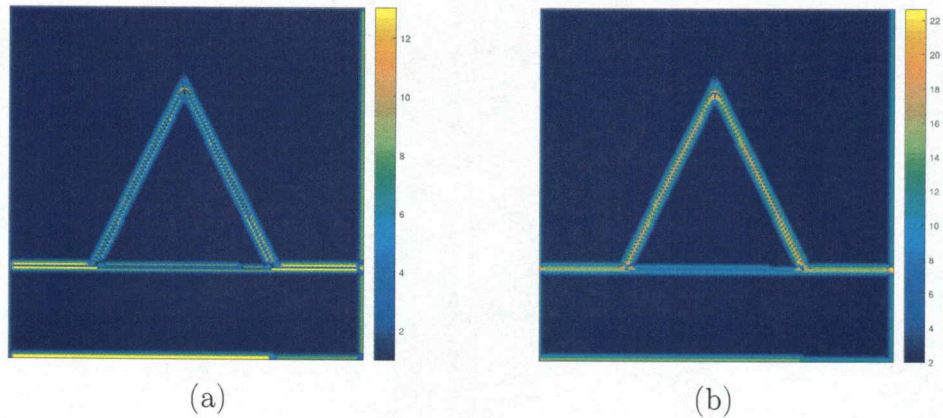


Figure 4.8: (a) A map of the ratios of eigenvalues, obtained by PCA, and (b) a map of the $r_{2_{ij}}$ values.

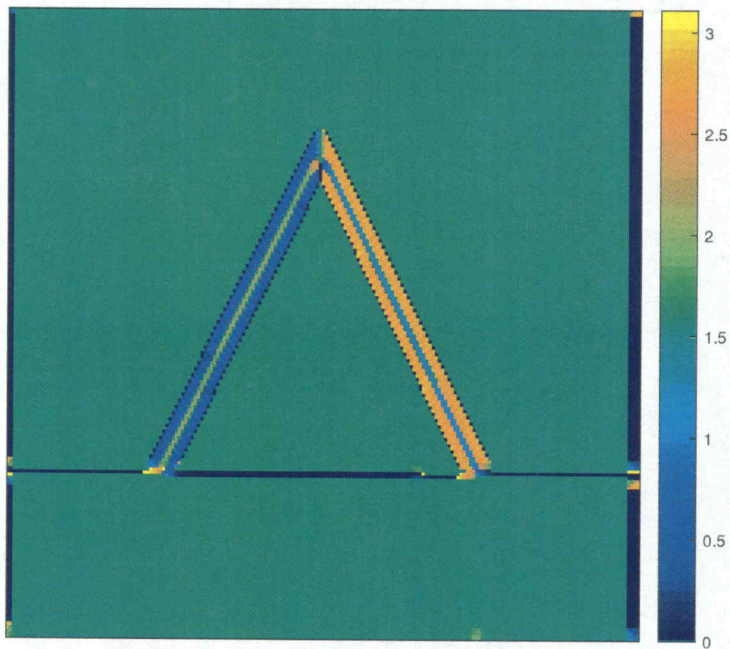


Figure 4.9: A map of the angles of least variation, θ_{ij} , measured in radians.

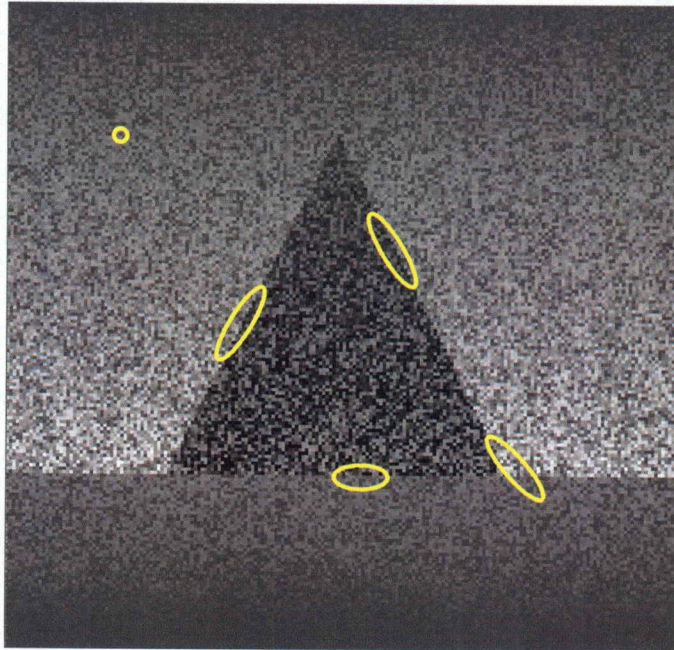


Figure 4.10: The original image with examples of anisotropic subimages for five pixels in the image.

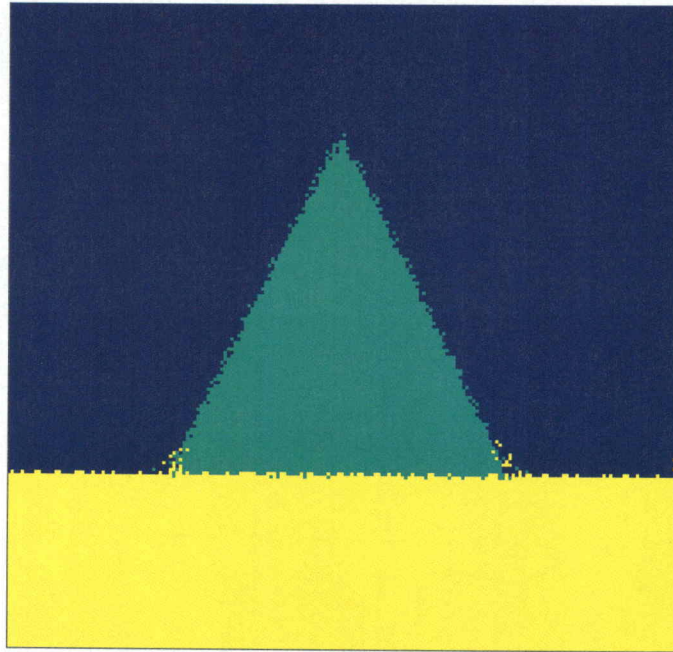
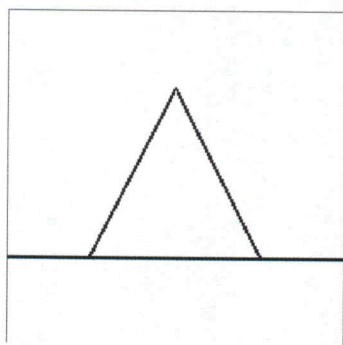
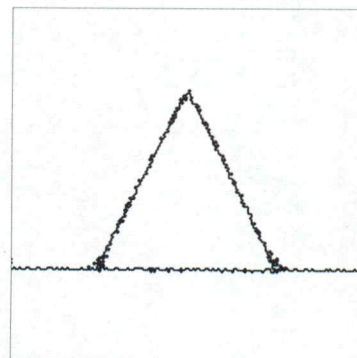


Figure 4.11: The ALADA segmentation, where each pixel has been assigned to one of the three classes.



(a)



(b)

Figure 4.12: (a) The true boundaries, and (b) the ALADA-determined boundaries.

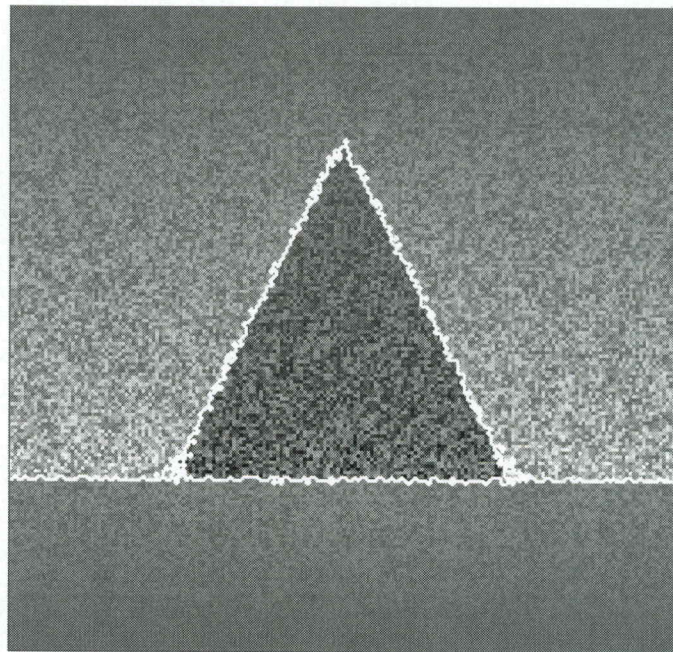


Figure 4.13: The original image with the ALADA-determined boundaries superimposed on top.

CHAPTER 5

UNCERTAINTY QUANTIFICATION

This chapter shows how the statistical natures of LADA and ALADA allow for uncertainty quantification of the detected boundaries. Using data collected during the LADA and ALADA processes, two uncertainty maps can be constructed to describe, first, the confidence in the selected segmentation based on local information, and second, the ability to discriminate between classes at boundaries. For the former, maximum likelihood estimation is used, and, for the latter, a combination of Welch's t -test and an F -test is used.

5.1 Maximum Likelihood Estimation

Given a LADA or ALADA segmentation for a pixel, we wish to quantify the probability that the pixel belongs to that class. Previous chapters have already stated that the estimation methods by which LADA and ALADA determine the segmented class, c , are considered maximum likelihood estimators (MLEs). In terms of segmentation, an MLE is a technique for determining the class that maximizes the probability distribution for an observed pixel intensity [67]. For a grayscale image, it is assumed that the local population mean $\mu_{c_{ij}}$ and variance $\sigma_{c_{ij}}$ are known for

the class c to which pixel x_{ij} is segmented, obtained via the local training data. Essentially, the class is selected using the following optimization:

$$G(x_{ij}) = \arg \max_c P(x_{ij} | \mu_{c_{ij}}, \sigma_{c_{ij}}), \quad \text{where } x_{ij} \sim \mathcal{N}(\mu_{c_{ij}}, \sigma_{c_{ij}}).$$

Consider the example in Figure 5.1(a): given the Gaussian distributions for two, hypothetical local classes and the intensity observation marked by the black star, the pixel would be segmented into class 1, given by the blue line, since it has greater probability density at that observation. However, because the observation is on the tail of the class 1 distribution, it is not well represented by that class either. MLE is, essentially, a way of quantifying how well or how poorly a pixel is represented by its segmented class, or in other words, the probability that a pixel belongs to its segmented class. The hypotheses of interest are

$$H_0 : x_{ij} \in c,$$

$$H_a : x_{ij} \notin c,$$

where pixel x_{ij} has been placed in class c .

The resulting MLE p-value is used to quantify evidence against the null hypothesis. A small p-value indicates that we have enough reason to reject the null hypothesis, i.e. that we are not confident that the pixel has been placed into its true class, while a large p-value indicates that there is not enough evidence to reject the

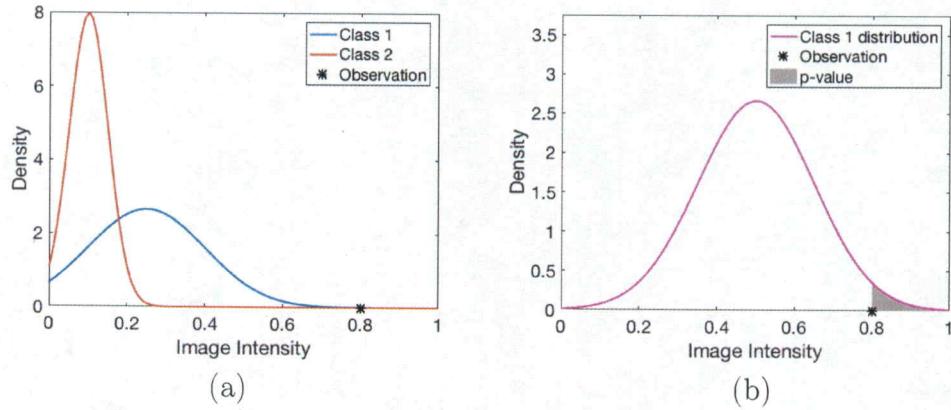


Figure 5.1: (a) While every pixel is segmented as belonging to a class, the assigned class is not always particularly likely. For the observation (black star), neither class 1 nor class 2 is very probable, though it is more likely to belong to class 1 (blue line) based on probability density. (b) For a given class distribution (magenta line) and observation (black star), the p-value is computed as the probability of observing that value or something more extreme.

null so we must accept that the null is true, i.e. that the pixel has been placed into its true class.

From the selected class's local distribution, the p-value is computed, the probability of observing the pixel x_{ij} or something more extreme, given the segmented class c and its associated parameters:

$$\text{p-value}_{\text{MLE}} = \left\{ \begin{array}{ll} P(X \geq x_{ij} | \mu_{c_{ij}}, \sigma_{c_{ij}}) & \text{if } x_{ij} > \mu_{c_{ij}} \\ P(X \leq x_{ij} | \mu_{c_{ij}}, \sigma_{c_{ij}}) & \text{otherwise} \end{array} \right\}.$$

An example of a one-sided p-value is demonstrated visually in Figure 5.1(b) for a distribution with mean $\mu_{c_{ij}} = 0.5$ and $\sigma_{c_{ij}} = 0.38$, and observation $x_{ij} = 0.8$. Notice that the further a pixel value is from the class mean, the smaller its p-value will be.

Assuming pixel x_{ij} had equal probability of having observed either positive or negative noise, the p-value is multiplied by two, for a two-sided p-value. For an image segmented by LADA or ALADA, we can produce a corresponding image of p-values from the statistical inference given here, in order to better identify parts of the segmentation of which we are less confident [70]. Regions of small p-values (which are usually found in areas of strong spatial variation - i.e. boundaries) can highlight parts of the image where the user should take special care to provide as much training data as possible or can signal that smaller parameter values might be needed in order to provide a less uncertain result.

5.2 Combined t -test and F -test

The second method we present for uncertainty quantification is a combination of two hypothesis tests to determine if neighboring class distributions are distinguishable. First, a Welch's t -test, or Welch's unequal variances t -test, is used to determine if the two distributions have equal means and second, an F -test is used to determine if the two distributions have equal variances. The p-values from the two tests are combined and together used to determine if the two classes have sufficiently different local distributions.

Similar to a student t -test, Welch's version compares two populations, or classes, in order to accept or reject a null hypothesis. Both tests assume that the data are independent, identically distributed (i.i.d.) and come from a Gaussian distribution, but, while the student t -test assumes that the classes have equal variances, Welch's t -test is designed for classes with unequal variances. Additionally, Welch's

t -test is more robust to Type I error (rejecting a true null hypothesis) than a student t -test when dealing with unequal sample sizes [71]. Since most of the images on which ALADA and LADA are performed are heteroskedastic, it would be an invalid oversimplification to assume equal variances. Note that Welch's t -test is an approximate solution to the Behrens-Fisher problem, which describes the problem of conducting a hypothesis test to determine if two Gaussian distributions with unequal variances have equal means, based on i.i.d. samples from the two distributions.

Welch's t -test is designed specifically for comparing the means of two groups. The null hypothesis states that the class means are equal and the alternative states that the class means are different. We are interested in knowing if any two means of local classes are evidenced to be equal, so we perform multiple Welch's t -tests to compare only two classes at a time. In our case, the hypotheses of interest are

$$H_0 : \mu_{c_{ij}} = \mu_{b_{ij}},$$

$$H_a : \mu_{c_{ij}} \neq \mu_{b_{ij}},$$

for all classes c and b local to pixel x_{ij} .

Evidence against the null hypothesis is quantified via the test statistic, t_{ij} , which is given by the formula

$$t_{ij} = \frac{\mu_{c_{ij}} - \mu_{b_{ij}}}{\sqrt{\frac{\sigma_{c_{ij}}^2}{|T_{c_{ij}}|} + \frac{\sigma_{b_{ij}}^2}{|T_{b_{ij}}|}}},$$

where $\mu_{c_{ij}}$ and $\mu_{b_{ij}}$ are the sample means, $\sigma_{c_{ij}}^2$ and $\sigma_{b_{ij}}^2$ are the sample variances, and $|T_{c_{ij}}|$ and $|T_{b_{ij}}|$ are the number of pixels, for the local classes c and b , respectively. A large t_{ij} -statistic indicates that we should accept the null hypothesis which states that the two classes have the same mean and are undifferentiable with such a measure. The Welch's t_{ij} -values are computed for each pair of local classes and the largest value is considered, which provides the most evidence that there are at least two classes with equal means.

The denominator of the t_{ij} -statistic is a linear combination of the independent sample variances, and the degrees of freedom associated with this are estimated using a two-class case of the Welch-Satterthwaite equation,

$$\nu_{ij} \approx \frac{\left(\frac{\sigma_{c_{ij}}^2}{|T_{c_{ij}}|} + \frac{\sigma_{b_{ij}}^2}{|T_{b_{ij}}|} \right)^2}{\frac{\sigma_{c_{ij}}^4}{|T_{c_{ij}}|^2(|T_{c_{ij}}| - 1)} + \frac{\sigma_{b_{ij}}^4}{|T_{b_{ij}}|^2(|T_{b_{ij}}| - 1)}},$$

which is rounded to the nearest integer.

The test statistic and its corresponding degrees of freedom can be used to find the associated p-value. One can either use a table of values from a two-sided Student t -distribution, or use the cumulative distribution function (CDF) of a t -distribution. To account for a two-sided p-value, this CDF value is multiplied by two, giving the formula

$$\text{p-value}_{t_{ij}} = 2CDF(t_{ij}) = 1 + 2t_{ij}\Gamma\left(\frac{\nu_{ij} + 1}{2}\right) \frac{{}_2F_1\left(\frac{1}{2}, \frac{\nu_{ij} + 1}{2}; \frac{3}{2}; -\frac{t_{ij}^2}{\nu_{ij}}\right)}{\sqrt{\pi\nu_{ij}} \Gamma\left(\frac{\nu_{ij}}{2}\right)},$$

where ${}_2F_1$ is the hypergeometric function.

Note that in the case that the two classes have equal sample sizes, the Welch's t -test statistic simplifies down to the Student t -test statistic and when the classes have equal sample sizes and equal variances, the Welch's t -test's degrees of freedom simplifies down to the Student t -test's degrees of freedom.

Next, an F -test, or Fisher's F -test is employed to compare the variances of the neighboring class distributions. Like the t -test, this test assumes that the data are independent, identically distributed (i.i.d.) and come from a Gaussian distribution. The null hypothesis states that the local class variances are equal and the alternative states that the local class variances are different. Like before, we are interested in knowing if any two variances of local classes are evidenced to be equal, so we perform multiple F -tests to compare only two classes at a time. In our case, the hypotheses of interest are

$$H_0 : \sigma_{c_{ij}}^2 = \sigma_{b_{ij}}^2,$$

$$H_a : \sigma_{c_{ij}}^2 \neq \sigma_{b_{ij}}^2,$$

for all classes c and b local to pixel x_{ij} .

Evidence against the null hypothesis is quantified via the test statistic, F_{ij} , which is given by the formula

$$F_{ij} = \frac{\sigma_{c_{ij}}^2}{\sigma_{b_{ij}}^2},$$

where $\sigma_{c_{ij}}^2 > \sigma_{b_{ij}}^2$. Note that it is not required that the larger variance be in the numerator, but doing so forces the F -test into a right tailed test, which simplifies subsequent calculations. Degrees of freedom are also required; an F -test has two degrees of freedom parameters, one for the numerator, $\nu_{c_{ij}}$, and one for the denominator, $\nu_{b_{ij}}$. Each degree of freedom parameter is the number of data points (in this case, training pixels) being used to calculate the sample variances, minus one

$$\nu_{c_{ij}} = |T_{c_{ij}}| - 1 \quad \text{and} \quad \nu_{b_{ij}} = |T_{b_{ij}}| - 1.$$

The F_{ij} -statistic and its associated degrees of freedom can be used to find the associated p-value. One can use a table of values for the p-values of an F -distribution or use the CDF of the F -distribution, which is given by the formula

$$\text{p-value}_{F_{ij}} = CDF(F_{ij}) = I_{\frac{\nu_{c_{ij}} F_{ij}}{\nu_{c_{ij}} F_{ij} + \nu_{b_{ij}}}} \left(\frac{\nu_{c_{ij}}}{2}, \frac{\nu_{b_{ij}}}{2} \right),$$

where I is the regularized incomplete beta function.

Once both p-values are computed, they are combined to form a joint hypothesis test,

$$H_0 : \mu_{c_{ij}} = \mu_{b_{ij}} \text{ and } \sigma_{c_{ij}}^2 = \sigma_{b_{ij}}^2,$$

$$H_a : \text{otherwise,}$$

where the null hypothesis is the case where the local Gaussian distributions are statistically equivalent and the alternative is that the local Gaussian distributions are distinguishable. The work presented by Frank et al. [72] describes several ways in which to combine the p-values of t -tests and F -tests for this exact purpose. Remember that the goal is to identify regions where the local classes are undifferentiable, so in order to be as conservative as possible, we want to accept the null whenever appropriate. This is done by employing the method of taking the maximum of the two p-values, since large p-values indicate evidence in favor of the null hypothesis. Therefore, the combined p-value is

$$\text{p-value}_{\text{combined}_{ij}} = \max\{\text{p-value}_{t_{ij}}, \text{p-value}_{F_{ij}}\}.$$

Analysis performed in this dissertation tests for significance at the $\alpha = .05$ level so the combined p-value is compared to .05. The following decision rule is used

$$\text{p-value}_{\text{combined}_{ij}} > .05 : \text{accept } H_0,$$

$$\text{p-value}_{\text{combined}_{ij}} \leq .05 : \text{reject } H_0.$$

Since this test is conducted for each pixel in the image with two or more local classes, the results can be visualized with a map of hypothesis test results. One color can represent accepting H_0 , another color can represent rejecting H_0 , and a third color can represent pixels where only one class is considered and no test was conducted. Thus, the image is used to indicate “problem areas” within an image

highlighting areas where it is especially difficult to discriminate between classes and may be improved upon by editing the training data.

CHAPTER 6

RESULTS ON REAL DATA

In this chapter, LADA and ALADA are demonstrated on examples from real physics experiments. First, we discuss when to use LADA versus ALADA, based on the image content and training data. Then, LADA is demonstrated on a time series of a mixture of plastic and coarse olivine sand as it is dynamically compressed. ALADA is demonstrated on an electron microscope image of shear bands in a piece of 1075 steel. Finally, an additional ALADA example is given of laser-induced cylindrical shock waves for which a polar transformation is utilized to simplify the boundary detection process.

Before delving into specific examples, let us further explore when it is appropriate to use LADA and ALADA. Because ALADA takes elliptical training data along suspected boundaries, it is important for the training data regions to have mostly smooth, straight boundaries, as in the synthetic example provided in Section 4.3. Training data regions that have cusps or concave regions can cause the elliptical training data to be taken in irregular patterns, collecting training data which is not most representative of the pixel in question. For images with these sorts of training data, LADA is best suited for detecting edges, as long as the distances between

training data regions are all relatively equal. Since LADA uses a single r value for the radius of each circular subimage when finding local training data, the r value should be appropriate for the entire image. If the user chooses training data such that there are small gaps between some regions and large gaps between other regions, they will have to choose between a small r parameter, which would be appropriate for the small gap but cause pixels in the large gap to be placed in the bonus class, or a large r parameter, which would be appropriate for the large gap but could be too large to find a clean local bound in the smaller gap. When the user is attempting to detect boundaries in an image with irregularly shaped training data or with very different sized gaps between training data, another method may be more appropriate than LADA or ALADA.

6.1 Time Series Example

Here, we demonstrate LADA on x-ray images taken from a dynamic physics experiment, conducted on the IMPULSE (IMPact System for ULtrafast Synchrotron Experiments) system at the Dynamic Compression Sector at the Advanced Photon Source [73, 74]. A sample of synthetic olivine sand (Green Diamond) encapsulated in a plastic holder is adhered to an immobile block and impacted with a cylindrical 4 mm diameter aluminum impactor at 769 m/s. A diagram of the experiment setup can be found in Figure 6.1. The compression wave is launched into the sand and plastic holder which is captured using a scintillator, which fluoresces when struck by charged particles, and an array of optical cameras. The images are attenuation x-ray phase contrast images where the lighter the color, the more the beam is attenuated

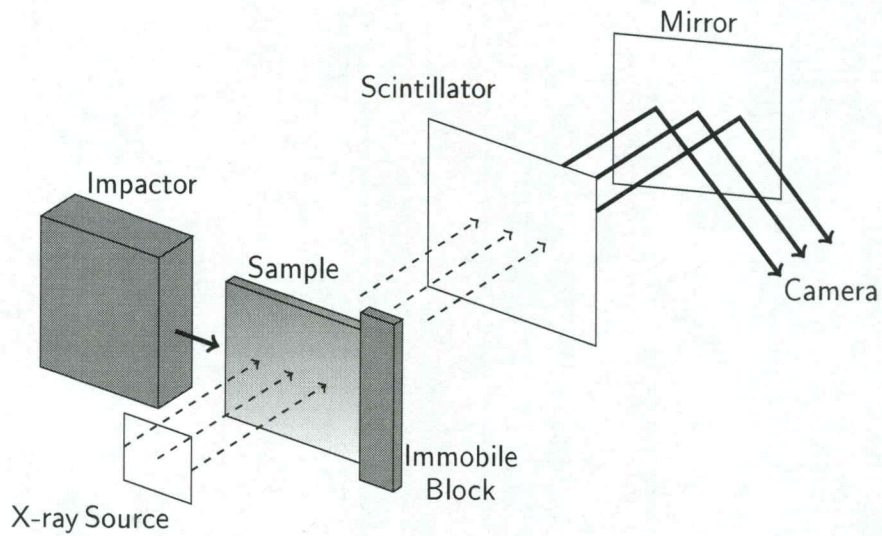


Figure 6.1: The experimental setup and imaging system for an aluminum impactor striking a sample of synthetic olivine sand encapsulated in a plastic holder from the left.

by the x-rays. Details such as motion, boundary width between regions, boundary structure, and the number of regions are desired to better understand how this sand compresses under dynamic loading and interpret the state of the material.

Traditionally, the gas-gun field has relied on velocimetry measurements which treat the sample as a black box, obtaining measurements from the back surface of the material. For homogeneous materials, this approach is often sufficient but lacks important internal sample information necessary to characterize heterogeneous materials with their additional degrees of freedom. The recent emergence of in-situ x-ray probes during an ultra-fast loading experiment like that of the IMPULSE setup [73] has provided a way to observe the internal state of a sample assembly, in real time. In the IMPULSE setup, multiple images of the same sample are recorded during a loading process which allows the sample to be characterized as it evolves in time.

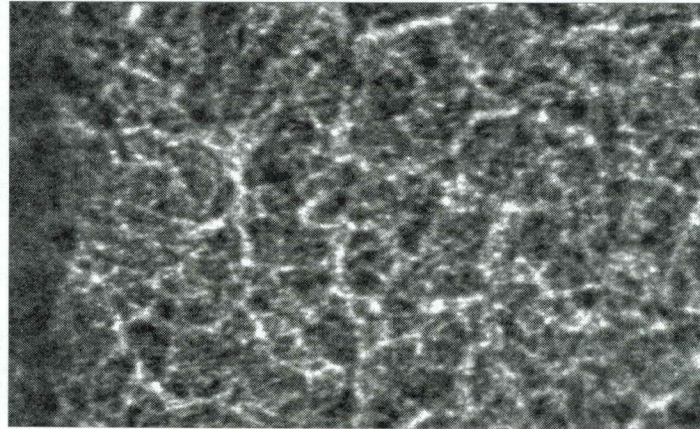
Unfortunately, every traditional segmentation and edge detection method used by this group produced unsatisfactory results because of the high levels of noise and the gradual changes between classes. Thus, a new technique is necessary. The LADA boundary detection method allows for a precise and reliable means to statistically determine the various regions of a compacted heterogeneous sample. LADA provides a productive means to detect and address the various regions of heterogeneous materials with a more scientifically objective, statistically significant, and rigorous methodology to extract the physics and dynamics of these heterogeneous samples.

Figure 6.2 shows three images from a series captured during an experiment with 769 m/s impactor velocity, recorded at 306.8 ns, 613.6 ns, and 1380.6 ns after impact, respectively. The pixel size is 1.68 μm per pixel. In the series of images, we see the compression wave propagating through the sand from left to right, with the darker region on the left being olivine sand that has been compressed. By the third image, the plastic target capsule enters on the left-hand side of the image (lighter intensity region).

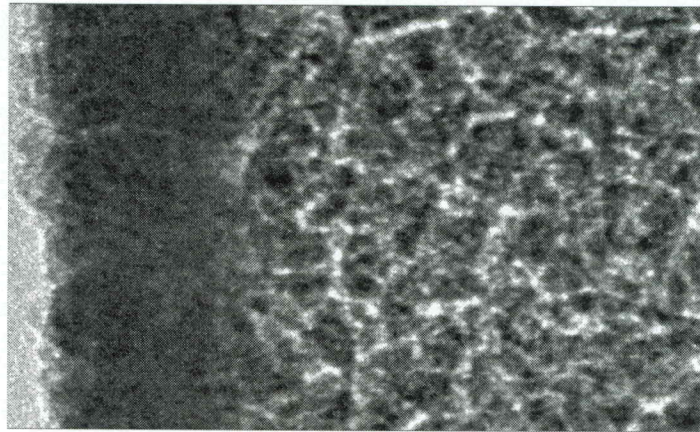
6.1.1 LADA Results

Figure 6.3(a) shows the hand-chosen training data where, from left to right, the training data regions are plastic target capsule (red), compressed sand (yellow), and uncompressed sand (light blue). The uncolored regions are areas where no training data have been chosen. As this is a difficult image, the gap between training classes is wide in many regions to avoid giving false training data. Figure 6.3(b) shows the

(a)



(b)



(c)

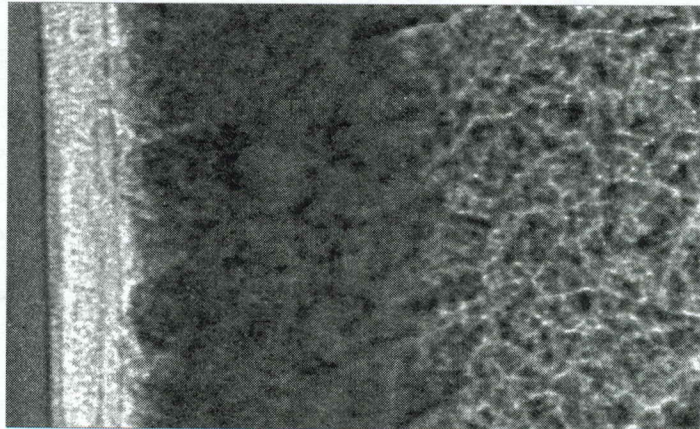


Figure 6.2: Time evolution of coarse olivine sand loaded from the left by Al impactor at 769 m/s recorded at (a) 306.8 ns, (b) 613.6 ns, and (c) 1380.6 ns after impact, respectively.

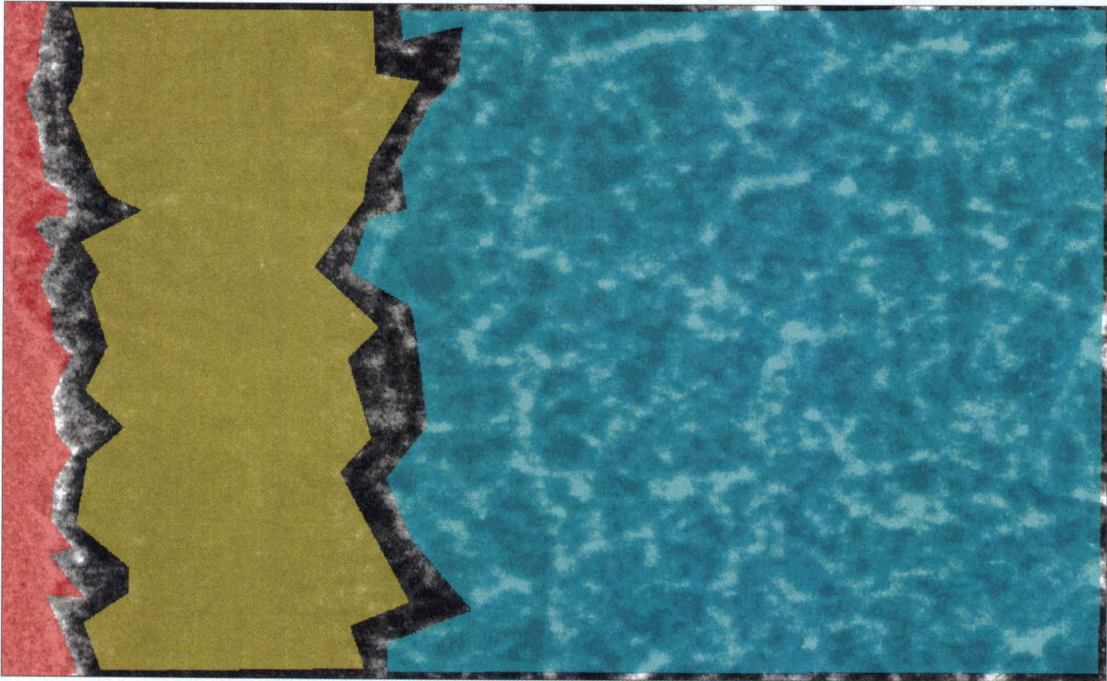
segmented image using input parameters $n = 100$ and $r = 31$, which is the smallest radius we can use without having any pixels placed in the bonus class.

Figure 6.4(a) shows the MLE 2-sided p-value map corresponding to the LADA segmentation in Figure 6.3, where dark blue corresponds to areas of higher uncertainty. Figure 6.4(b) shows the original image with boundaries (yellow), identified by LADA using input parameters $n = 100$ and $r = 31$. Uncertainty regions are given in teal, where a priori knowledge about the experiment that the compression wave is a planar wave is used to make the decision to fit linear bounds around the regions of p-values ≤ 0.05 . Here, conservative vertical bounds are used to capture the entire region of p-values ≤ 0.05 about each boundary.

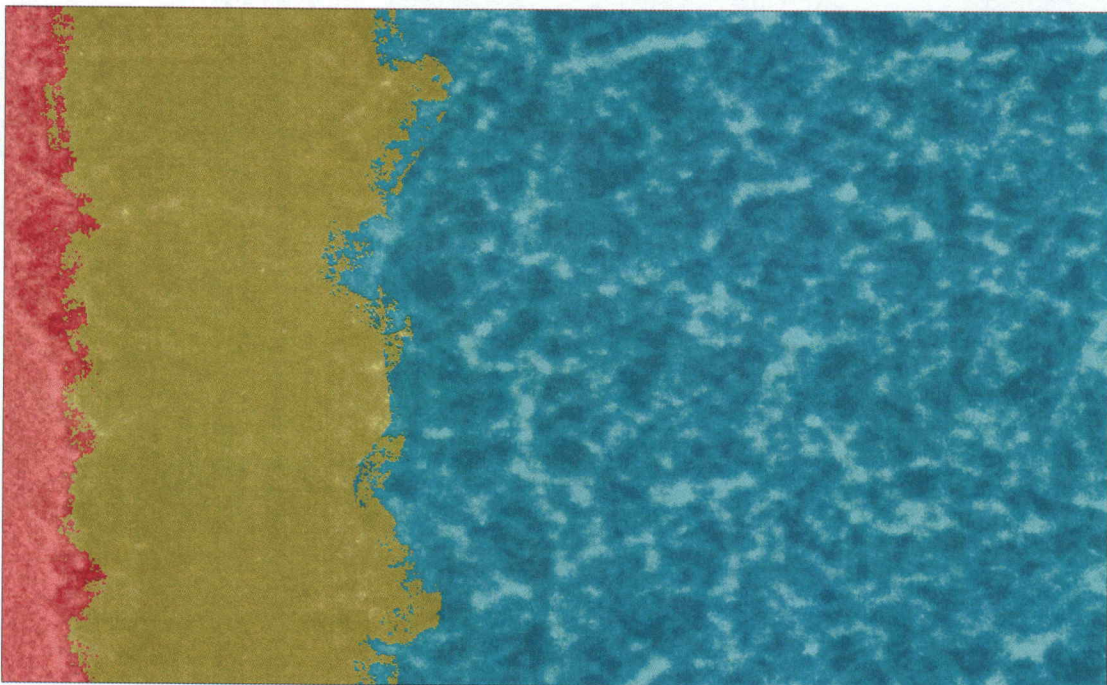
Figure 6.5 shows the progression of the front of the compression wave as it propagates through the three images shown in Figure 6.2. Using the boundaries found by LADA, and the accompanying conservative vertical uncertainty regions around each boundary, the user can compute quantities such as velocity and propagate the uncertainty through their calculations, giving scientific results that are quantitatively meaningful. Here, the author chose input parameters of $n = 100$, and the smallest r possible such that no pixels were placed into the bonus class which was $r = 42$ for the first image, $r = 31$ for the second image, and $r = 32$ for the last image.

6.1.2 Results from Other Methods

Figure 6.6 demonstrates a few alternative methods. The reader is cautioned against using the three edge detection methods on images with gradual boundaries be-

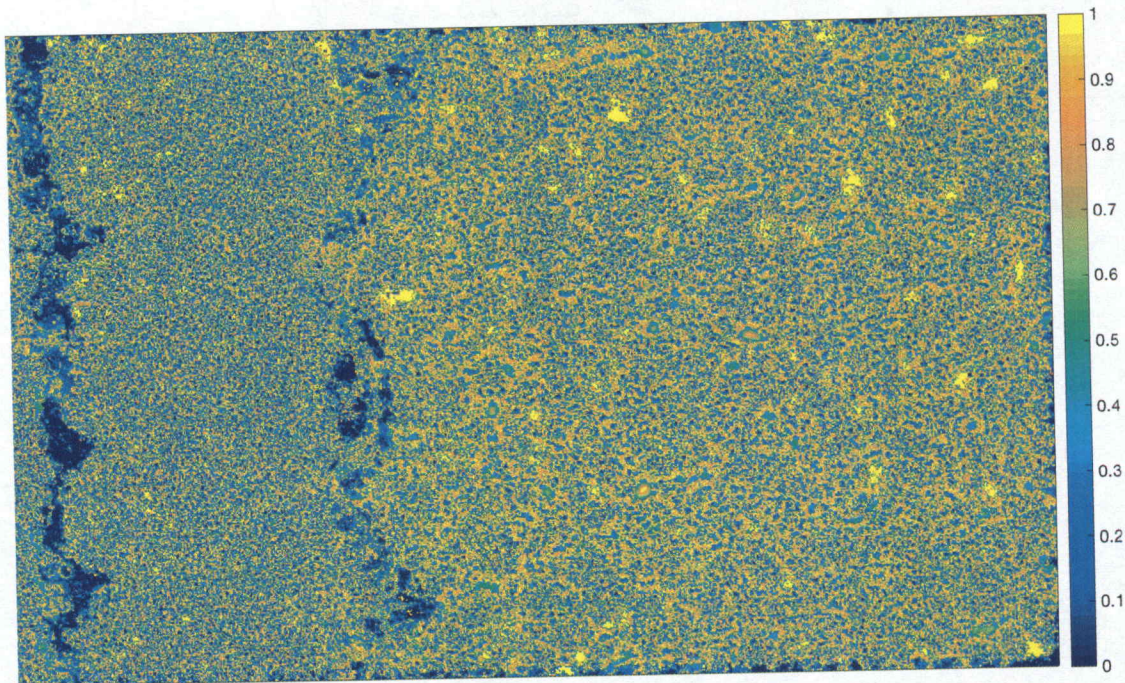


(a)

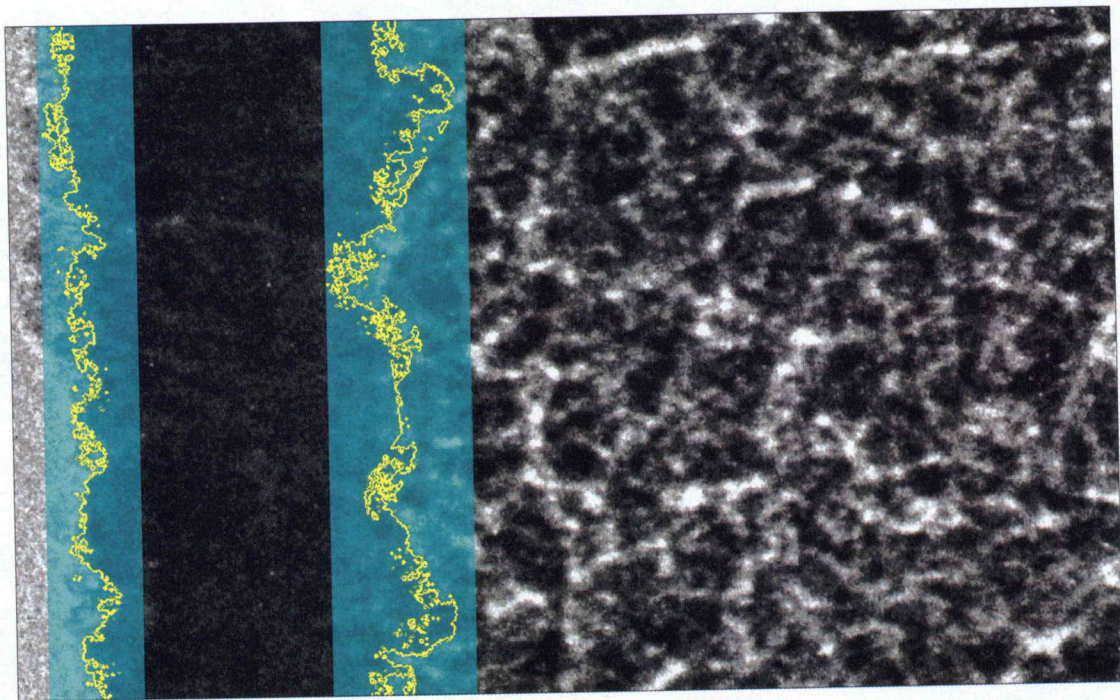


(b)

Figure 6.3: We show the (a) training data and the (b) LADA segmentation for the second image in Figure 6.2.



(a)



(b)

Figure 6.4: (a) The corresponding MLE 2-sided p-value map shows how confident we are in each pixel's assigned class. (b) The image with its final LADA-determined boundaries, using input parameters $n = 100$ and $r = 31$, and regions of uncertainty surrounding each boundary, found using the 2-sided p-value map at an alpha value of $\alpha = .05$. Vertical uncertainty regions are found because of a priori knowledge about the planar nature of the compression wave.

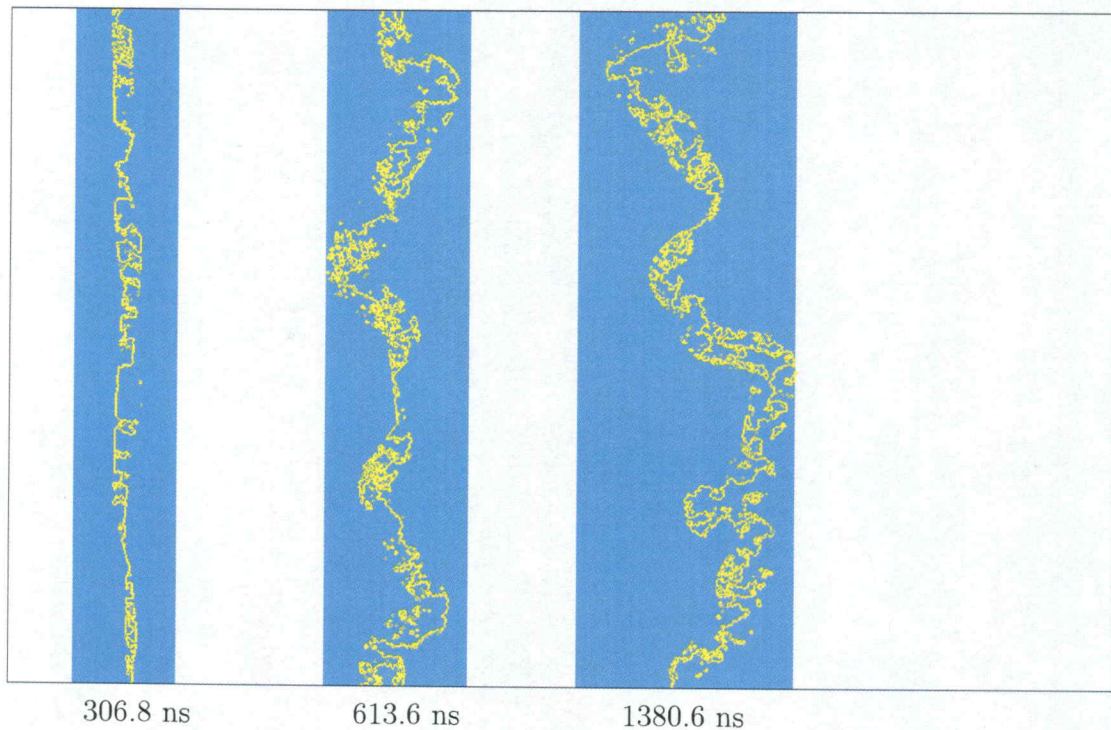


Figure 6.5: A map showing the progression of a single boundary from the three images in Figure 6.2, along with their regions of uncertainty. Again, we fit vertical uncertainty regions because of the authors' a priori knowledge about the planar nature of the compression wave. Using these boundary locations and the times at which each image was taken, we can compute quantities such as velocity of wave front.

cause, in practice, the input parameters have significant effect on location of detected edges.

The first alternative method is Sobel edge detection, which was performed using the Matlab command `edge(image, 'Sobel', t)`. The best result was obtained using a threshold of $t = .1$, as shown in Figure 6.6(a). Lowering this threshold causes even more bounds to be detected while raising the threshold causes some edges to be filtered out, including edges that are near our suspected boundaries.

The second alternative method is Canny edge detection, which was performed using the Matlab command `edge(image, 'Canny', t, σ)`, where t is a threshold parameter and σ is the standard deviation of the Gaussian filter, which determines the size of the Gaussian filter [75]. Smaller σ values smooth the image less and give results that contain weaker, more irregular edges that are more susceptible to noise while larger σ values smooth the image more, blurring noise such that only the stronger, smoother boundaries are returned [34]. The default value for this parameter is $\sigma = \sqrt{2}$ but we must increase it significantly in order to filter out all the unwanted edges that Canny finds. Here, the best result is obtained using parameters ($t = .4$, $\sigma = 12$), as shown in Figure 6.6(b).

Third, we use the edge detection backpropagation artificial neural network described in [76] and using the accompanying in [77]. The neural network uses one hidden layer with 12 hidden nodes and is trained on 16 different 4×4 edge patterns for 500 epochs. Here, a learning rate of $\nu = .01$ and a momentum term of $\alpha = .9$ are used. To account for the size discrepancy between the edge patterns and the original image, a 300×488 pixel version of the image was used instead of the original 600×976 pixel version, although training the original image with larger edge patterns would have produced a similar result.

Finally, the quadratic discriminant analysis segmentation is provided. This algorithm has no input parameters but does take in training data. The training data provided here was the same training data provided for the LADA algorithm. The boundaries could be collected from the QDA segmentation but as the reader can see, the results are extremely noisy.

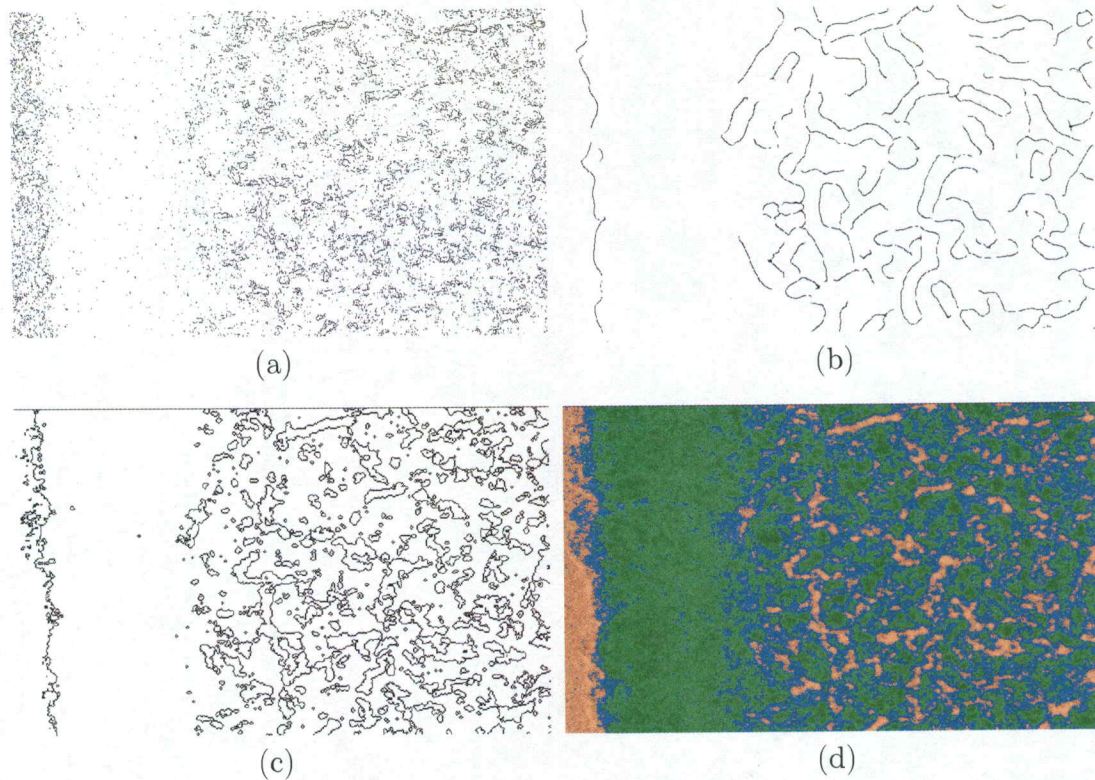


Figure 6.6: Boundary detection results using (a) Sobel edge detection, (b) Canny edge detection, (c) a backpropogation neural network, and (d) segmentation results using QDA.

6.2 Shear Bands ALADA Example

Now we demonstrate ALADA on an optical microscopy image taken of shear bands formed in a carbon steel chip under high strain rates during orthogonal cutting.

Material response exhibits a dependency upon the rate of deformation, or strain rate. At lower strain rates, deformation is typically accommodated uniformly. At higher strain rates, some materials, such as carbon steel, begin to accommodate strain non-uniformly in localized bands of severe deformation known as shear bands. In the machining process, shear bands form periodically and the spacing of the shear

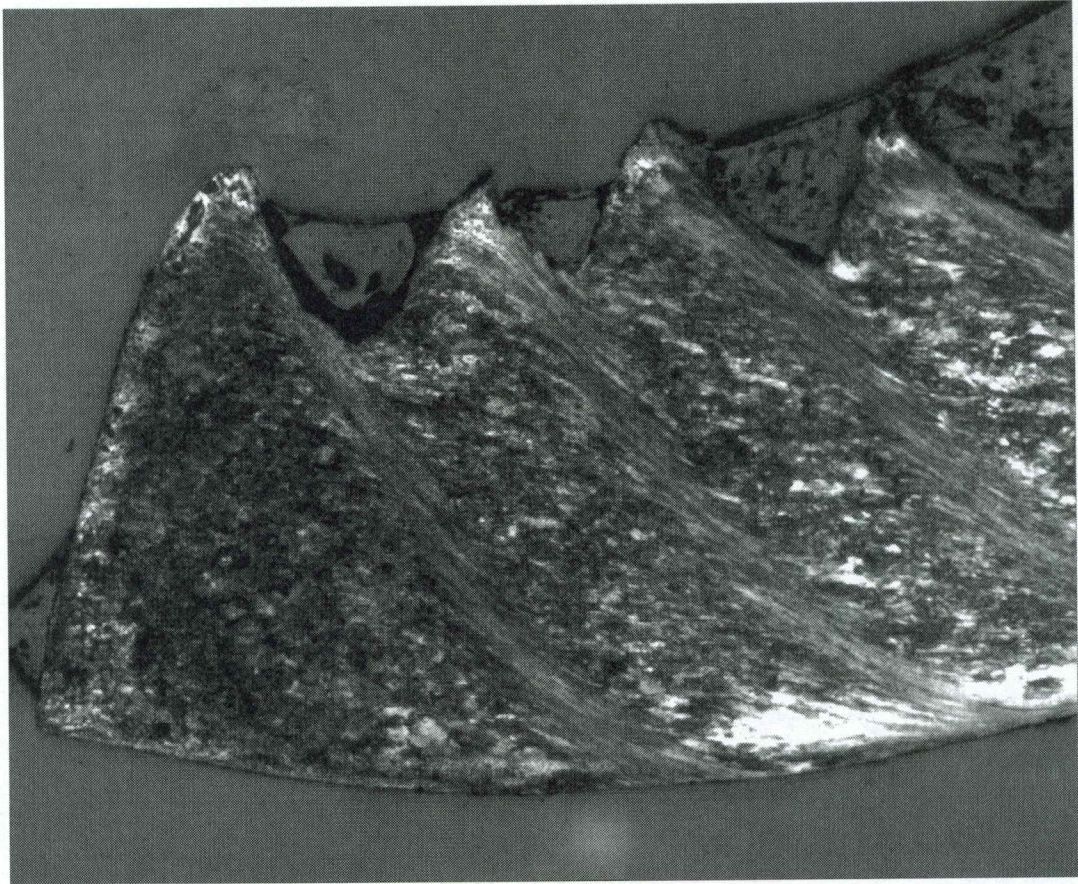


Figure 6.7: Image of 1075 steel displaying shear bands from a high strain rate impact.

bands is dependent upon the cutting speed and related to the strength of the material. When the shear bands are closely spaced, the strength approaches that of the material during uniform deformation but as the spacing increases, the average strength of the material decreases. It is thus important to characterize the conditions under which shear bands form as well as their geometric parameters such as thickness and spacing. Figure 6.7 shows a sample of 1075 steel that has been struck by a cutting block under a high strain rate.

6.2.1 ALADA Results

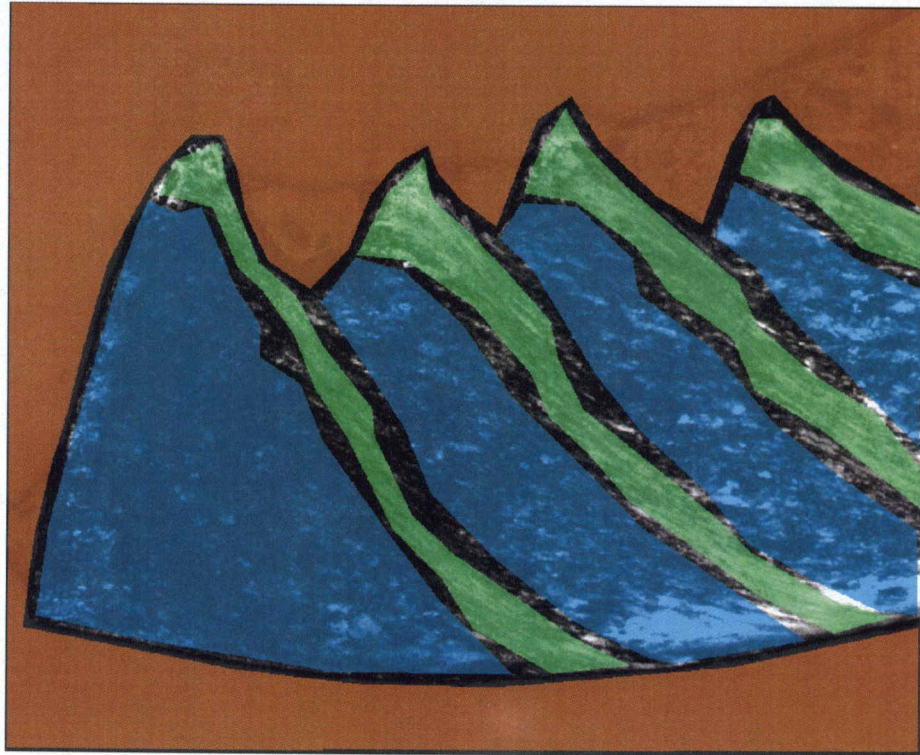
Figure 6.8(a) shows the hand-chosen training data for the shear bands (green), the steel chips (blue), and the background (orange). The uncolored regions are areas where no training data have been chosen. Figure 6.8(b) shows the segmented image using an input parameter of $n = 40$. The author has also chosen to cap the larger radius of each anisotropic subimage using the rule:

$$r_{2_{ij}} = r_{1_{ij}} \min \left\{ \frac{\lambda_1}{\lambda_2}, 4 \right\},$$

to prevent the subimages from getting too large.

Figure 6.9(a) shows the MLE 2-sided p-value map corresponding to the ALADA segmentation in Figure 6.8(b), where dark blue corresponds to areas of higher uncertainty. Figure 6.9(b) shows the F -test and t -test p-value map, which identifies regions where we can confidently distinguish the local class distributions (dark blue) and regions where the local distributions are indistinguishable (green). Yellow regions are where only one class is considered. Note that a larger value for input parameter n or choosing more training data would result in wider regions where two or more classes are considered for each pixel assignment. A smaller n parameter value or choosing less training data will result in thinner regions where two or more classes are considered.

Finally, Figure 6.10 shows the original image with boundaries (red), identified by ALADA using an input parameter of $n = 40$. Using these boundaries, the user can identify the exact location and width of the shear bands which, according to the

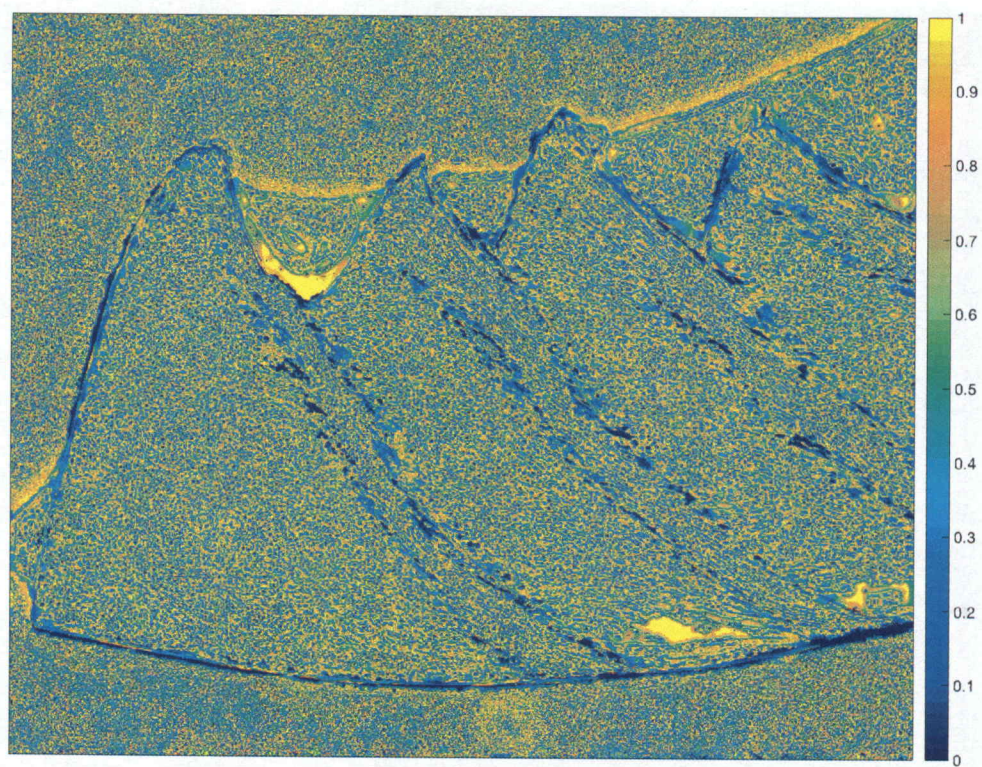


(a)

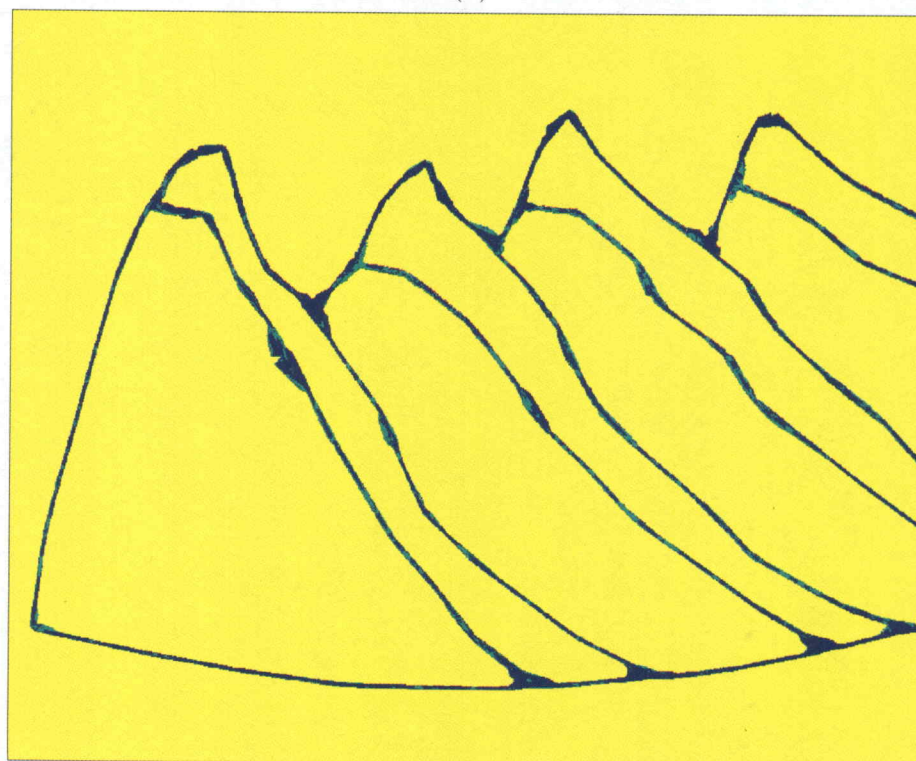


(b)

Figure 6.8: We show the (a) training data and the (b) ALADA segmentation for the shear band image in Figure 6.7.



(a)



(b)

Figure 6.9: (a) The MLE 2-sided p-value map shows how confident we are in each pixel's assigned class. (b) The F -test and t -test map shows how confident we are that the available classes are distinguishable, in areas where at least two classes are being considered.

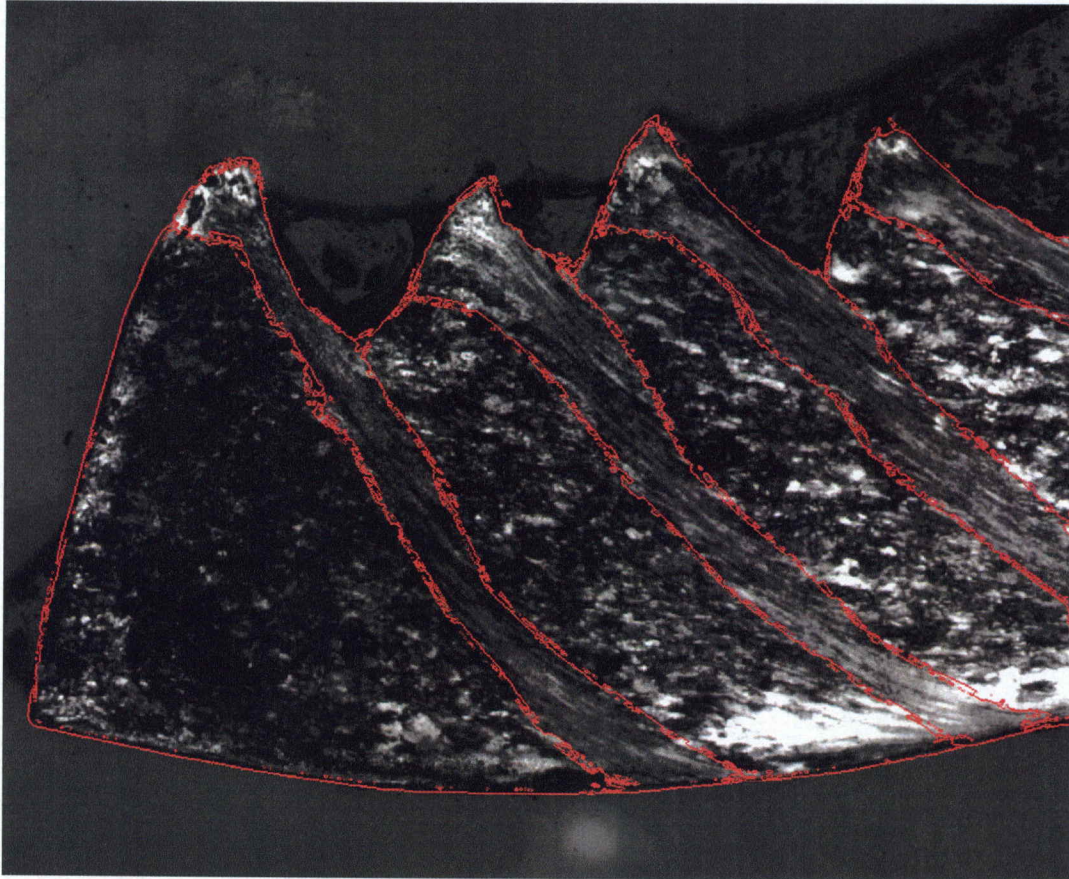


Figure 6.10: The ALADA-determined boundaries superimposed on the original image.

providers of this image, has never been done in the literature. Additionally, the user can identify uncertainty regions about each boundary using the MLE p-value map, if desired.

6.2.2 Results from Other Methods

The same four methods presented for the previous example are given here on the shear band image. We again start with Sobel edge detection, where the best result is obtained using the threshold parameter $t = .12$, and can be found in Figure 6.11(a).

Again, lowering this threshold causes even more bounds to be detected while raising the threshold causes some edges to be filtered out, including edges that are near our suspected boundaries.

Next, Canny edge detection is done using parameters ($t = .2$, $\sigma = 2$), and is shown in Figure 6.11(b). Larger σ values would smooth the image, reducing noise and unwanted edges, but doing so also removes details around the shear bands, which are the features of interest in this image [34]. Even with extensive parameter tuning, Sobel and Canny edge detection results are of little to no use, as they are not able to distinguish between actual edges of interest and edges that are the result of noise.

Figure 6.11(c) shows the results from an edge detection backpropagation artificial neural network, using the same parameters are used as in the previous neural network example: 1 hidden layer, 12 hidden nodes, trained on 16 different 4×4 edge patterns for 500 epochs, with a learning rate of $\nu = .01$, and a momentum term of $\alpha = .9$. Again, to account for the size discrepancy between the edge patterns and the original image, a 418×507 pixel version of the image instead of the original 835×1014 pixel version.

Finally, Figure 6.11(d) shows the quadratic discriminant analysis segmentation. The training data provided here was the same training data provided for the LADA algorithm.

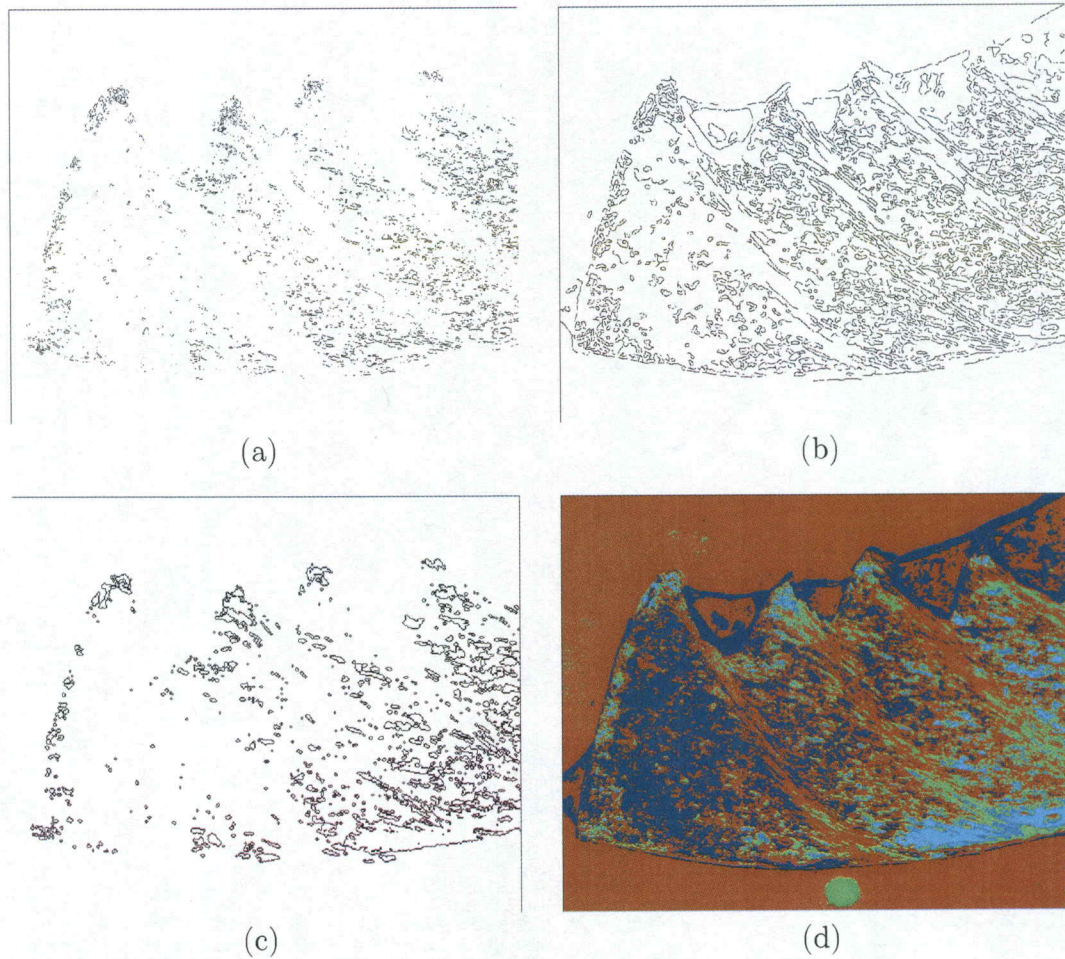


Figure 6.11: Boundary detection results using (a) Sobel edge detection, (b) Canny edge detection, (c) a backpropagation neural network, and (d) segmentation results using QDA.

6.3 Polar Unroll ALADA Example

One additional example is provided to show how an image may be adapted to be appropriate for ALADA boundary detection. Figure 6.12 shows a picture of a laser-induced converging shockwave, collected using optical imaging [78].

When shock waves travel through a material, the material is irreversibly changed. Temporal quantitative analysis of wave dynamics requires all measure-

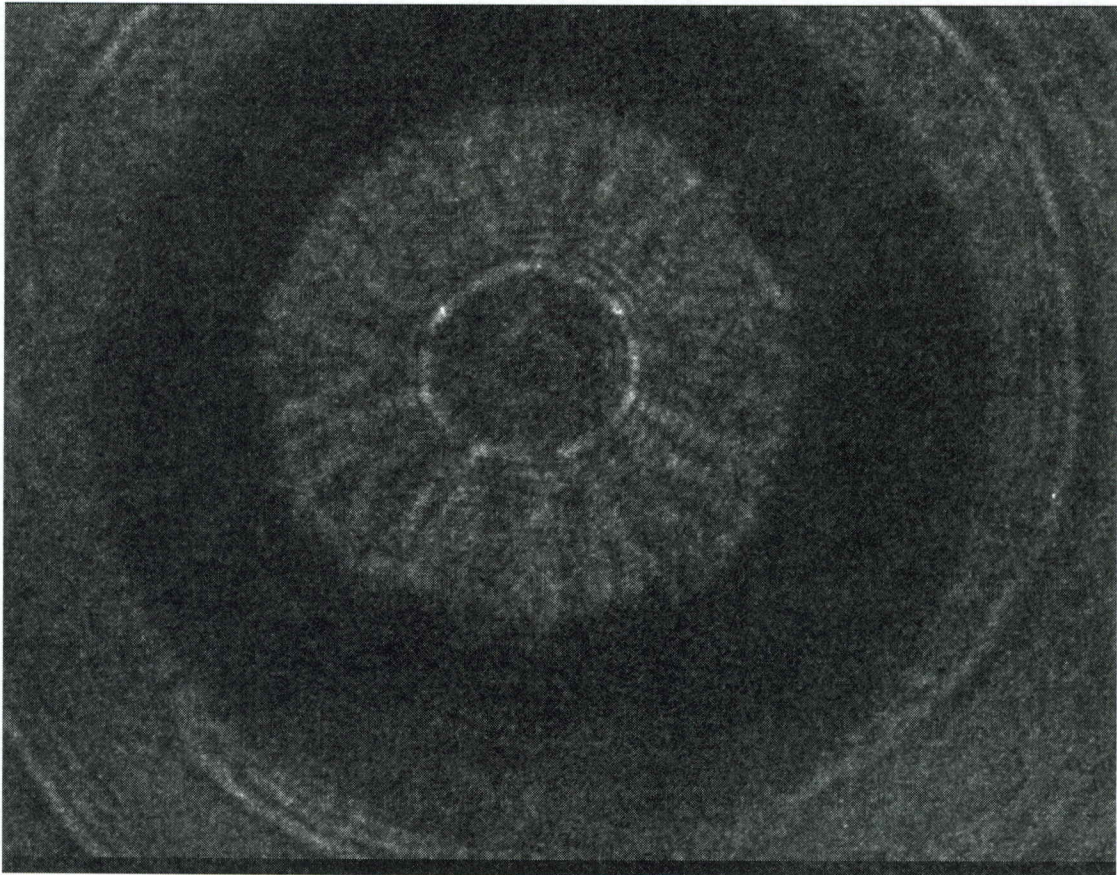


Figure 6.12: Image of a laser-induced, cylindrically converging shock wave propagating in water. The goal is to segment the image into regions of unshocked water, shock front, shocked water, laser ring, and outside (listed in order from image center moving outwards).

ments, (in this case, images) to be collected from a single shock experiment. As a result, the signal-to-noise ratio is limited and the images contain high noise content, especially for high spatiotemporal resolution [79].

When strong shock waves travel through a material, they can separate into multi-wave structures, beginning with a one-dimensional elastic wave where the material is reversibly compressed and followed by a series of plastic waves corresponding to

the irreversible changes that occur upon fast dynamic compression. In an image, the multi-wave structure can appear as a relatively uniform material with bright, narrow attributes that separate the regions. In some cases, there are changes in pixel intensity or in texture between the different regions of the shock, but in most cases these classes are difficult to separate with current boundary detection techniques because the various physics classes (e.g. shock wave) have very little to distinguish them from the other classes (e.g. are overlapping in feature space).

Figure 6.12 is an image of a cylindrically converging shock wave traveling through a thin layer of water that is between thick glass substrates [78]. The shock wave is generated from the interaction of a 200 micrometer diameter laser ring with an absorber to produce a shock wave that travels within the sample plane, perpendicular to the incident laser. A 180 femtosecond duration pulse from the same laser collects a shadowgraph image in transmission, which gives an image that is a spatial map of the second derivative of the density for the material [80]. In this experiment, six images were taken in a single experiment at 5 nanosecond intervals in order to visualize the convergence and subsequent divergence of the shock, which depicts the two-dimensional physics of the complex system. To obtain quantifiable data from these images, the shock must be precisely and accurately located with clear understanding of any error from that measurement. Identifying the location of the shock wave throughout a series of images will enable the researcher to compute quantities such as velocity.

Locating the shock is a challenging problem, but one well-suited to a local, supervised method. The boundaries of the center rings are fairly clear, allowing the

user to take training data with small gaps between training regions. However, the outer rings have more gradual boundaries, forcing the user to be more conservative when selecting training data. These differences in gap size between training data makes LADA a poor choice since there is unlikely to be a single set of parameters that is appropriate across the entire image. On the other hand, ALADA performs better when the training data are selected along linear boundaries, to ensure that ellipses follow along the suspected boundaries without protruding into one class over another. This example explores how a radially symmetric image can be transformed to be better suited for ALADA boundary detection.

Before beginning any ALADA analysis, the image is transformed or “unrolled” from its starting polar form to a rectangular coordinate system. First, we find the largest possible circular subimage in Figure 6.12, centered around the center of the inner shock. Figure 6.13 shows the “unrolled” version of this subimage, with the center of the cylindrical shock transformed to lie along the top of the unrolled shock image and the outer ring transformed to lie along the bottom of the unrolled shock image, such that the vertical axis represents the radius from the pole, or center of the shock, while the horizontal axis represents the angle from the polar axis in the counterclockwise direction. Note that this transformation stretches pixels that are close to the pole and compresses pixels that are far from the pole.

From here, ALADA is performed as usual. Figure 6.14(a) displays the hand-selected training data, with the classes beginning at the top of the image and moving down: unshocked water (blue), shock front (green), shocked water (orange), and laser ring (red). Looking at the original image, these regions begin in the center of the

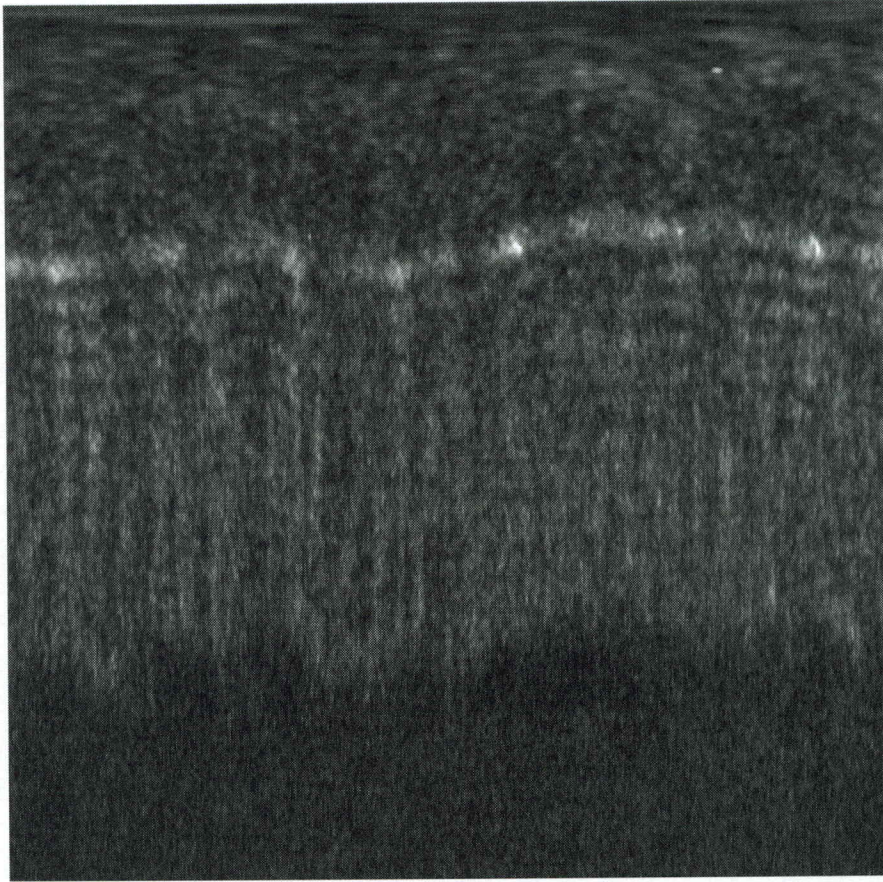
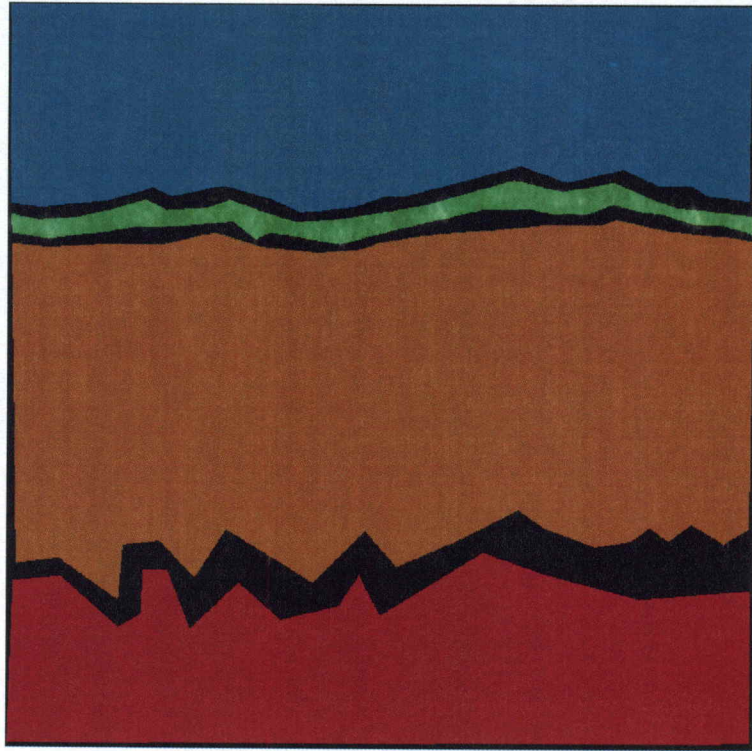


Figure 6.13: A polar transformation is used to “unroll” the image in Figure 6.12, in order to simplify the boundary-detection process.

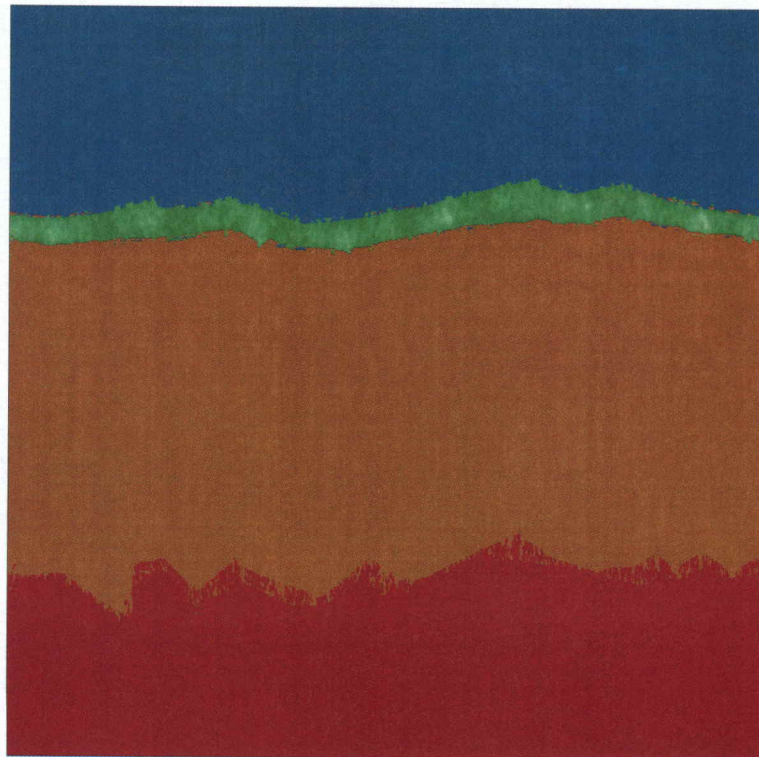
image and move out. Figure 6.14(b) shows the resulting ALADA segmentation using the input parameter $n = 40$. As with the previous ALADA example, the larger radius of each anisotropic subimage is capped using the rule:

$$r_{2_{ij}} = r_{1_{ij}} \min \left\{ \frac{\lambda_1}{\lambda_2}, 4 \right\},$$

to prevent the elliptical subimages from getting too large.



(a)

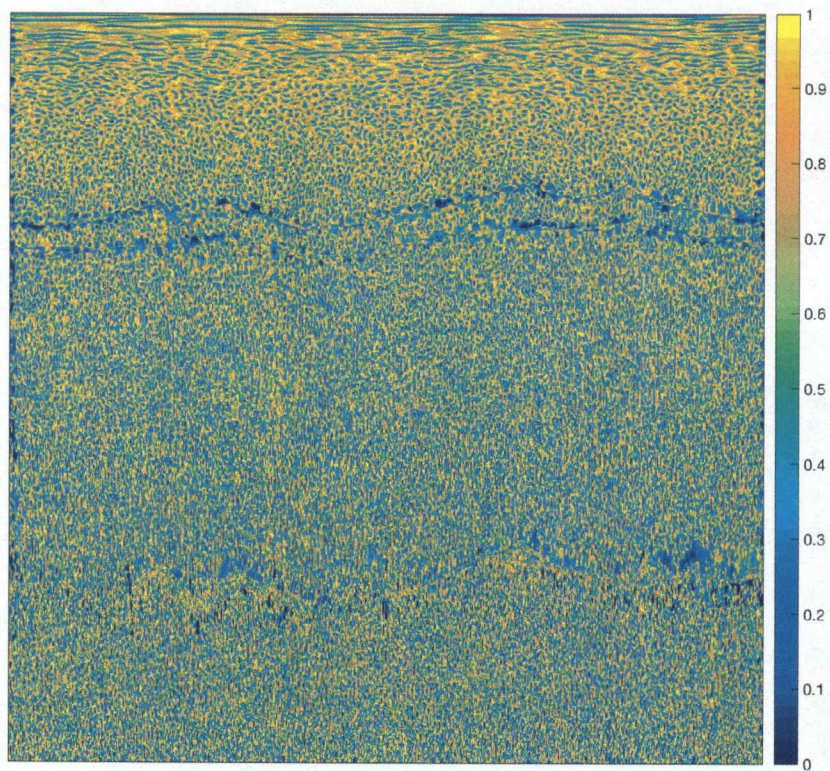


(b)

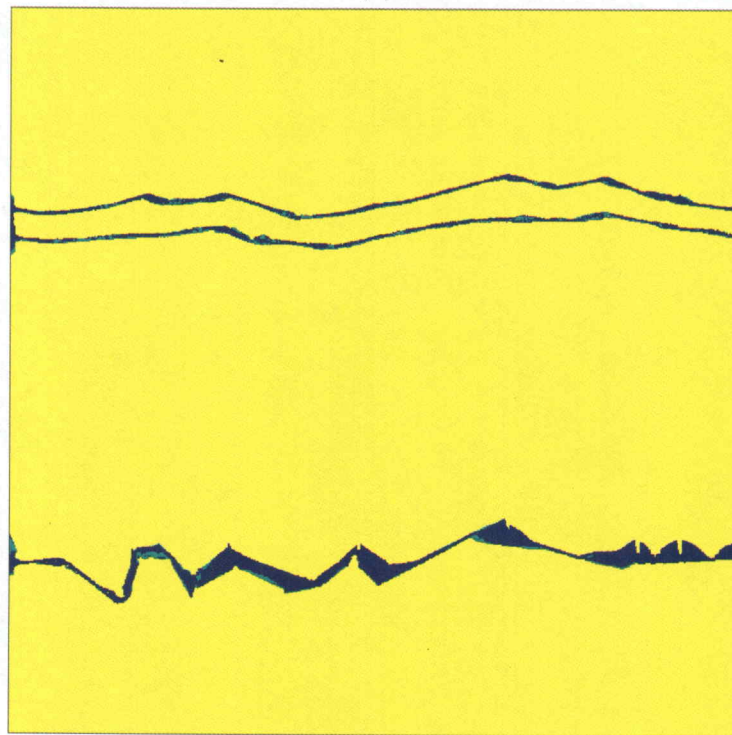
Figure 6.14: We show the (a) training data and the (b) ALADA segmentation for the laser shock image in Figure 6.13.

Figure 6.15(a) shows the MLE 2-sided p-value map corresponding to the ALADA segmentation in Figure 6.14(b), where dark blue corresponds to areas of higher uncertainty. Figure 6.15(b) shows the F -test and t -test p-value map, which identifies regions where we can confidently distinguish the local class distributions (dark blue) and regions where the local distributions are indistinguishable (green). Yellow regions are where only one class is considered.

Finally, Figure 6.16(a) shows the ALADA-determined boundaries (red) on the unrolled shock image, identified using the ALADA segmentation in Figure 6.14(b). By reversing the original transformation, we can “reroll” the image and the ALADA-determined boundaries to obtain the polar results, as shows in Figure 6.14(b). Note that transforming the unrolled version back into polar form generates a circular image, and cannot recreate parts of the image outside of the circle. The outer red circle identifies the outer limit of the image, and is not in fact part of the identified image boundaries. Using these boundaries, the user can identify the exact location of each region in the shock, and, using the other images taken during this experiment, can measure the speed and pressure of the shock front as it moves over time. Additionally, the user can identify uncertainty regions about each boundary using the MLE p-value map, if desired.

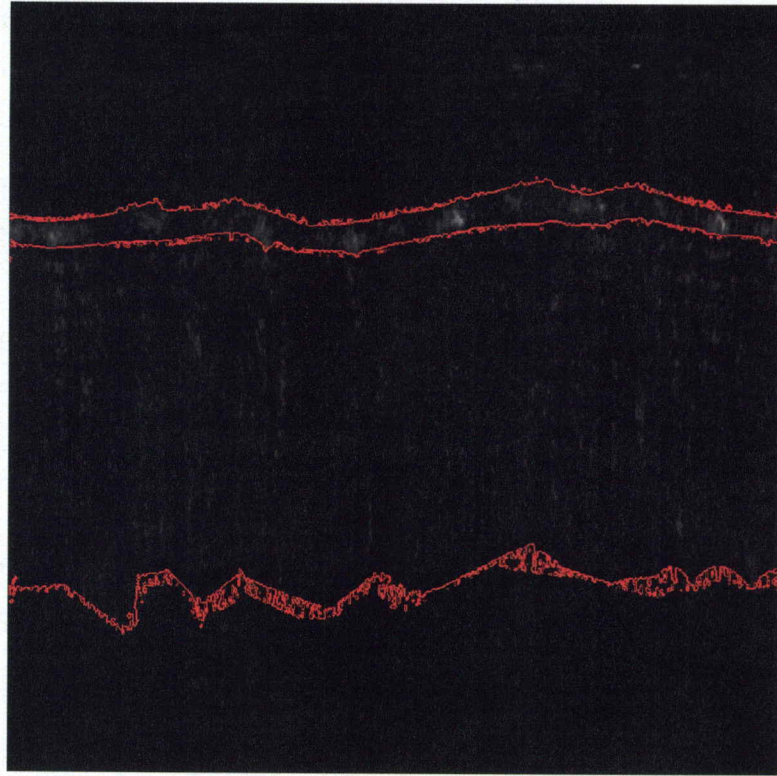


(a)

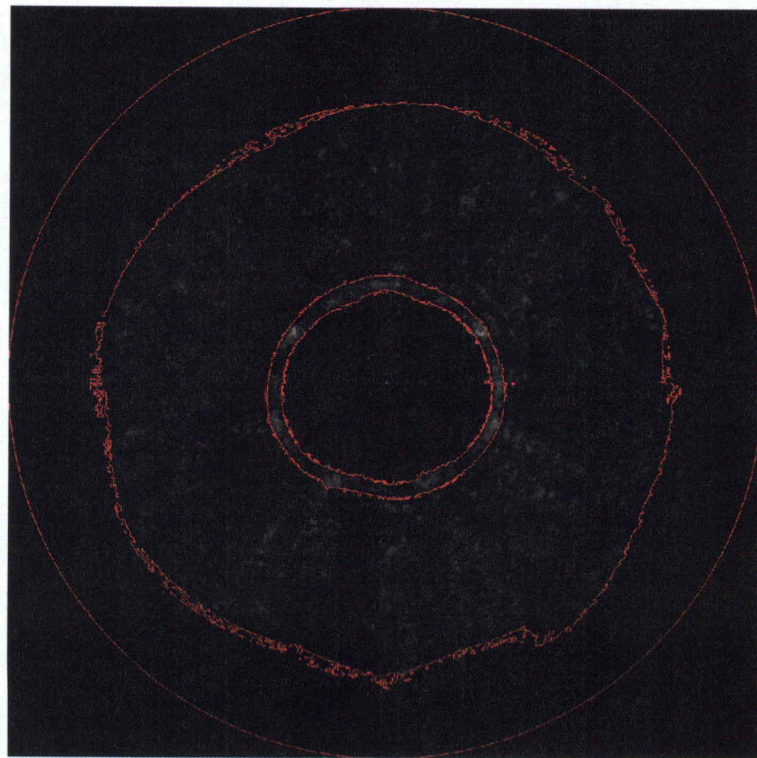


(b)

Figure 6.15: (a) The MLE 2-sided p-value map shows how confident we are in each pixel's assigned class. (b) The F -test and t -test map shows how confident we are that the available classes are distinguishable, in areas where at least two classes are being considered.



(a)



(b)

Figure 6.16: The ALADA-determined bounds on (a) the unrolled shock image and (b) the original shock image.

CHAPTER 7

CONCLUSIONS

This dissertation presents two new supervised boundary detection methods for images with low contrast between classes, heteroskedasticity, and objects whose intensities vary spatially. The first algorithm, locally adaptive discriminant analysis (LADA), uses two user-supplied parameters to subsample local training data in a circular region about each pixel. This method is best for images where the distances between training regions are relatively equal, in order for the input parameters to be appropriate for the entire image. The second algorithm, anisotropic locally adaptive discriminant analysis (ALADA), uses a single input parameter to subsample local training data in an elliptical shape along boundaries of interest. This algorithm is the first of its kind to use both the statistics of pixel intensities as well as the statistics of pixel locations. Since the algorithm relies on spatial distribution of training data, it is most appropriate for images with smooth, straight training boundaries.

After presenting LADA and ALADA, two methods for uncertainty quantification (UQ) were explored. The first, maximum likelihood estimation, can be used to describe confidence in the selected class for each pixel. When viewed as an image, the map of MLE p-values shows regions of high confidence, where the user can be satis-

fied with the result, and regions of low confidence, where the user may want to select additional training data, or choose different parameters. The regions of low p-values can also be used to place regions about the boundaries, demonstrating uncertainty in the boundary location. The second UQ method explored is a combination test using a Welch's t -test statistic and an F -test statistic. Wherever two or more classes are being considered, this test can be used to determine the separability of the local classes. Rejecting the null hypothesis means that there is not sufficient information to say that the distributions are equivalent. Accepting the null means determining that the two local classes have statistically equal means and variances, meaning we cannot confidently choose one class over another. Viewing these p-values as a map can show the user regions where they cannot be sure of a boundary's exact location because the training data distributions on each side are indistinguishable.

Finally, the algorithms and the UQ methods were demonstrated on real images from lab experiments. LADA was shown on a series of three images from a planar shock wave experiment, where the resulting boundaries and their uncertainty regions can be propagated through equations to find quantities such as velocity. ALADA was shown on a material science experiment sample showing parallel shear bands, where the boundaries and UQ can be used to find width of the shear bands and percentage of the metal that has been weakened. Four alternative methods were also used on both these images, giving inferior results to LADA or ALADA, and none of the alternative methods provide any sort of uncertainty quantification. Lastly, ALADA was also used on a special image from a cylindrical laser shock wave experiment that first required a polar transformation. Transforming the image made ALADA an

appropriate boundary detection choice and after finding the results, the image was retransformed back into the original coordinate system.

7.1 Future Work

Boundary detection is a rapidly developing field and there are many avenues yet to be explored. The largest change the author would like to explore is the possibility of eliminating the Gaussian assumption in the LADA and ALADA algorithms. The normality studies presented in this work show that while the normality assumption is more accurate at the local level than at the global level, it is still not accurate for a majority of the image. To remedy this, LADA and ALADA could be altered to fit the local training data to whichever distribution offers the best fit. Research has already been released showing how one might use the first four moments of a data set find the best fit to a distribution from the lambda family of distributions [81]. Doing this would offer more reasonable assumptions for a more confident result. Uncertainty quantification would be much more challenging, as different hypothesis tests have different assumptions on how the data are distributed, but the algorithm would be more correct.

Another possible alteration to LADA and ALADA is to iterate on the prior probabilities. As they currently stand, LADA and ALADA assume equal prior probabilities for all local classes, a common assumption when little is known about the local distributions or when the researcher wishes to use a more frequentist approach. However, there is likely some preference to one class over the other, based on prevalence of local training data or nearby class assignments. If this idea were to be pursued,

a mode filter should be used where after LADA or ALADA has been completed and all pixels have been assigned a class, the proportion of local class assignments should be used as the prior probabilities for a second run through the LADA or ALADA algorithm. Special weight could be given to the class that was originally assigned for each pixel.

Finally, as a simple way to speed up the algorithms, training pixels could be automatically assigned to their labeled class. LADA and ALADA are boundary detection algorithms and as the training data should be sufficiently far from the boundary, the author sees no reason why those pixels should not be automatically classified.

APPENDICES

APPENDIX A

SUPPLEMENTAL MATERIAL

A.1 Hypothesis Testing

In statistics, we are often interested in whether or not a mathematical statement related to random variables is true. A statistically-justified way of determining if such a statement is true is via a hypothesis test. Hypothesis tests typically have four steps [83].

First, a null hypothesis, H_0 , is proposed stating that the observations are the result of pure chance. A corresponding alternative hypothesis, H_a , is chosen stating that the observations are the result of a real effect, with some amount of chance variation. For example, a researcher may wish to determine if a set of observations, X , comes from a distribution with mean 0. They would create the hypotheses

$$H_0 : \mu = 0$$

$$H_a : \mu \neq 0$$

where μ is the mean of the distribution from which the observations are selected.

Second, the researcher chooses an appropriate test statistic based on the probability model of interest. Test statistics used in this dissertation include a Welch's t -test statistic, a two-sample F -test statistic, a combined t -test and F -test statistic, and a Shapiro-Wilk test statistic. The first three will be explained in Chapter 5 and the Shapiro-Wilk test is presented in the next section. Each test statistic has its own formula where input values such as sample mean, sample standard deviation, and degrees of freedom are found using the data set in question.

Step three uses the test statistic to compute a p-value which is the probability of obtaining a result equal to or more extreme than what was actually observed, assuming the null hypothesis is true. Smaller p-values mean a small probability of the observation occurring, indicating that the researcher should reject the null hypothesis.

Whether or not a researcher rejects the null hypothesis depends on what significance level they choose, which corresponds to the confidence level of the hypothesis test. The most common significance level is $\alpha = .05$, corresponding to a confidence level of 95%, which is what will be used for the entirety of this dissertation. If $p \leq \alpha$, the researcher concludes that the observed effect is statistically significant and rejects the null hypothesis in favor of the alternative hypothesis. If $p > \alpha$, the null hypothesis is accepted.

A.2 Normality Testing

Normality assumptions are common in statistical applications, even when such an assumption is unwarranted. Discriminant analysis methods such as LDA and

QDA assume that global training data are normally distributed and since LADA and ALADA are based on these methods, they, too, follow Gaussian assumptions. In order to determine how appropriate these assumptions are, we use the Shapiro-Wilk test to determine if each pixel's local training distributions are normally distributed.

The Shapiro-Wilk test is a hypothesis test where the null and alternative hypotheses are as follows

H_0 : population is normally distributed

H_a : otherwise.

The test statistic for the hypothesis test is given by

$$W_{c_{ij}} = \frac{\left(\sum_{i=1}^n a_i x^{(i)}\right)^2}{\sum_{i=1}^n (x_i - \mu_{c_{ij}})^2},$$

where a_i are Shapiro-Wilk coefficients and $x^{(i)}$ is the i^{th} smallest number in the sample.

The coefficients a_i can be found using the formula

$$(a_1, a_2, \dots, a_n) = \frac{m^T V^{-1}}{C}$$

where C is a vector norm given by

$$C = \|C^{-1}m\| = (m^T V^{-1} v^{-1} m)^{1/2}$$

and the column vector m is comprised of the expected values of the order statistics of iid random variables sampled from a standard normal distributions. V is the covariance matrix of those normal order statistics [84].

This test statistic can be converted to a p-value which is tested at the $\alpha = .05$ level to determine if we accept or reject the null. Below are the Shapiro-Wilk results for each image in this dissertation. The last column is the percentage of local distributions that fail to reject the null hypothesis, i.e. the percentage of local distributions that are Gaussian.

Image Name	Reference	Normality Rate
Synthetic staircase	Figure 3.2	14.25%
Synthetic triangle	Figure 4.5	84.86%
Olivine sand impact 1	Figure 6.2(a)	34.21%
Olivine sand impact 2	Figure 6.2(b)	44.70%
Olivine sand impact 3	Figure 6.2(c)	46.73%
Shear bands	Figure 6.7	18.63%
Unrolled shock	Figure 6.13	32.49%

A.3 Principal Component Analysis

For some $m \times m$ matrix, A , an eigenvector is a nonzero $m \times 1$ vector v such that $Av = \lambda v$ for some scalar λ . The scalar λ is called the eigenvalue corresponding to the eigenvector v [82]. An eigenvector's direction is unchanged when linear transformations are applied to it.

Essentially, an eigenvector is a direction and the corresponding eigenvalue tells how much variance there is in the data in that direction. Principal component analysis orders these eigenvalues, or variances, from greatest to least, to describe the underlying structure in the data by identifying the direction where there is the most variance. These vectors are always orthogonal to each other so n -dimensional data will always have n eigenvector-eigenvalue pairs or n principal components.

When PCA is used for ALADA, the data sets are images where we are interested in the spatial spread of pixels, not the spread of pixel intensities. Therefore, only two dimensions - the row and column of each pixel - are used, while pixel intensities are ignored. If a data set has, say, n pixels, then we take the $n \times 2$ matrix of pixel locations and find its 2×2 covariance matrix. This matrix's eigenvector corresponding to its larger eigenvalue is called the first principal component or direction of greatest variation, while the eigenvector corresponding to the smaller eigenvalue is called the second principal component or the direction of least variation.

A.4 k -Nearest Neighbors

The k -nearest neighbors algorithm is a non-parametric supervised classification method. It is a "majority voting" method, meaning that each new, unlabeled data point is classified according to the mode of the surrounding training data. The number of surrounding training data being considered is determined by the user-supplied parameter k .

In the ALADA algorithm, k -nearest neighbors is used to determine the radius of the circular subimage, which is also the minor radius of the elliptical subimage.

The user supplies a number of neighbors parameter, k , which in this case is how many local training pixels are required. For each pixel in the image, the algorithm is used to determine the minimum radius $r_{1_{ij}}$ needed to have at least k training pixels in the subcircle.

The general algorithm for k -nearest neighbors classification is given in Algorithm 6. For the ALADA algorithm, an abbreviated version of k -nearest neighbors is used. The algorithm is performed as listed below until Step 3, at which point the minor radius is set equal to the k^{th} distance, such that

$$r_{1_{ij}} = d_k.$$

Algorithm 9 k -nearest neighbors for classification

Given image X , define training data T with C classes, and a number of neighbors, k .

For each pixel $x_{ij} \in X$:

1. Calculate $d(x_{ij}, t) \forall t \in T$, where d denotes the Euclidean distance between the pixels.
2. Arrange the calculated Euclidean distances in non-decreasing order such that $d_1 \leq d_2 \leq d_3 \leq \dots$.
3. Take the first k distances d_1, d_2, \dots, d_k from this sorted list.
4. Find the k training points corresponding to the k distances.
5. Let k_c denote the number of training data belonging to the c^{th} class such that $k_c \geq 0$ and $\sum_c k_c = k$.
6. Place x_{ij} into the class c for which

$$G(x_{ij}) = \arg \max_c k_c.$$

REFERENCES

- [1] Vipin Tyagi. *Understanding Digital Image Processing*. CRC Press, Taylor & Francis Group, Boca Raton, FL, Sep 2018.
- [2] Leora Dresselhaus-Cooper, Marylesa Howard, Margaret C. Hock, B.T. Meehan, Kyle J. Ramos, Cindy A. Bolme, Richard L. Sandberg, and Keith A. Nelson. Machine learning to analyze images of shocked materials for precise and accurate measurements. *Journal of Applied Physics*, 122(10), Sep 2017.
- [3] Ilya Valmianski, Carlos Monton, and Ivan K Schuller. Microscopy image segmentation tool: Robust image data analysis. *Review of Scientific Instruments*, 85(3), Mar 2014.
- [4] Marylesa Howard, Michael Fowler, Aaron Luttmann, Stephen E. Mitchell, and Margaret C. Hock. Bayesian abel inversion in quantitative x-ray radiography. *SIAM Journal on Scientific Computing*, 38(3):B396–B413, May 2016.
- [5] Reza Khatami, Giorgos Mountrakis, and Stephen V. Stehman. Mapping per-pixel predicted accuracy of classified remote sensing images. *Remote Sensing of Environment*, 191:156–167, Mar 2017.
- [6] Julien Radoux and Pierre Defourny. A quantitative assessment of boundaries in automated forest stand delineation using very high resolution imagery. *Remote Sensing of Environment*, 110(4):468–475, Oct 2007.
- [7] John A. Richards. *Remote Sensing Digital Analysis: An Introduction*. Springer-Verlag, Berlin, Germany, 5th edition, 2013.
- [8] Andrés Troya-Galvis, Pierre Gançarski, Nicolas Passat, and Laure Berti-Équille. Unsupervised quantification of under- and over-segmentation for object-based remote sensing image analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(5):1936–1945, May 2015.
- [9] Yong Zha, Jay Gao, and Shaoxiang Ni. Use of normalized difference built-up index in automatically mapping urban areas from tm imagery. *International Journal of Remote Sensing*, 24(3):583–594, Jan 2003.
- [10] Satoshi Moriwaki, Yasuhiko Terada, Katsumi Kose, Tomoyuki Haishi, and Yoshihiko Sekozawa. Visualization and quantification of vascular structure of fruit using magnetic resonance microimaging. *Applied Magnetic Resonance*, 45(6):517–525, Jun 2014.

- [11] Mariano Cabezas, Arnau Oliver, Xavier Lladó, Jordi Freixenet, and Meritxell Bach Cuadra. A review of atlas-based segmentation for magnetic resonance brain images. *Computer Methods and Programs in Biomedicine*, 104(3):e158–e177, Dec 2011.
- [12] Robert A. Gatenby, Edward T. Gawlinski, Arthur F. Gmitro, Brant Kaylor, and Robert J. Gillies. Acid-mediated tumor invasion: a multidisciplinary study. *Cancer Research*, 66(10):5216–5223, May 2006.
- [13] Bjoern H. Menze, Andras Jakab, Stefan Bauer, Jayashree Kalpathy-Cramer, Keyvan Farahani, Justin Kirby, Yuliya Burren, et al. The multimodal brain tumor image segmentation benchmark (BRATS). *IEEE Transactions on Medical Imaging*, 34(10):1993–2024, Oct 2015.
- [14] Yu qian Zhao, Wei hua Gui, Zhen cheng Chen, Jing tian Tang, and Ling-Yun Li. Medical images edge detection based on mathematical morphology. In *27th Annual International Conference of the Engineering in Medicine and Biology Society, 2005*, pages 6492–6495. IEEE, Jan 2006.
- [15] J.W. Byng, N.F. Boyd, L. Little, G. Lockwood, E. Fishell, R.A. Jong, and M.J. Yaffe. Symmetry of projection in the quantitative analysis of mammographic images. *European Journal of Cancer Prevention: the official journal of the European Cancer Prevention Organisation (ECP)*, 5(5):319–327, Oct 1996.
- [16] Simon Shen, Karan Syal, Nongjian Tao, and Shaopeng Wang. Note: An automated image analysis method for high-throughput classification of surface-bound bacterial cell motions. *Review of Scientific Instruments*, 86(12), Dec 2015.
- [17] Anne Bazille, Michael A. Guttman, Elliot R. McVeigh, and Elias A. Zerhouni. Impact of semiautomated versus manual image segmentation errors on myocardial strain calculation by magnetic resonance tagging. *Investigative Radiology*, 29:427–433, Apr 1994.
- [18] Stephen M. Plaza, Louis K. Scheffer, and Mathew Saunders. Minimizing manual image segmentation turn-around time for neuronal reconstruction by embracing uncertainty. *PLoS ONE*, 7(9), Sep 2012.
- [19] G.P. Dinneen. Programming pattern recognition. In *Proceedings of the 1955 Western Joint Computer Conference*, pages 94–100. ACM, 1955.
- [20] Awais Mansoor, Ulas Bagci, Brent Foster, Ziyue Xu, Georgios Z. Papadakis, Les R. Folio, Jayaram K. Udupa, and Daniel J. Mollura. Segmentation and image analysis of abnormal lungs at ct: current approaches, challenges, and future trends. *RadioGraphics*, 35(4):1056–1076, Jul 2015.
- [21] P. Sivakumar and S. Meenakshi. A review on image segmentation techniques. *International Journal of Advanced Research in Computer Engineering & Technology*, 5(3):641–647, Mar 2016.

- [22] Aurélien Rizk, Grégory Paul, Pietro Incardona, Milica Bugarski, Maysam Mansouri, Axel Niemann, Urs Ziegler, Philipp Berger, and Ivo F. Sbalzarini. Segmentation and quantification of subcellular structures in fluorescence microscopy images using SQUASSH. *Nature Protocols*, 9(3):586–596, Mar 2014.
- [23] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, UK, 1st edition, Mar 2000.
- [24] Salem Saleh Al-amri, Namdeo V. Kalyankar, and S. D. Khamitkar. Image segmentation by using threshold techniques. *Journal of Computing*, 2:83–86, May 2010.
- [25] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, Jan 2006.
- [26] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning: data mining, inference, and prediction*. Statistics. Springer Science+Business Media, LLC, New York, NY, 2nd edition, 2009.
- [27] MathWorks. Edge detection. Online www.mathworks.com/help/images/edge-detection.html, 2018.
- [28] Josef Kittler. On the accuracy of the sobel edge detector. *Image and Vision Computing*, 1(1):37–42, Feb 1983.
- [29] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, Nov 1986.
- [30] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International Journal of Image Processing (IJIP)*, 3(1):1–11, Jan 2009.
- [31] Girish N. Chapple, R.D. Daruwala, and Manoj S. Gofane. Comparisons of robert, prewitt, sobel operator based edge detection methods for real time uses on FPGA. In *2015 International Conference on Technologies for Sustainable Development (ICTSD)*. IEEE, 2015.
- [32] Robert M. Haralick. Digital step edges from zero crossing of second directional derivatives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(1):58–68, Jan 1984.
- [33] Benjamin Hell, Marc Kassubeck, Pablo Bauszat, Martin Eisemann, and Marcus Magnor. An approach toward fast gradient-based image segmentation. *IEEE Transactions on Image Processing*, 24(9):2633–2645, Sep 2015.
- [34] Jose Anphy, Merlin Dixon K. Deepa, Joseph Naiji, George E. Silpa, and V. Anjitha. Performance study of edge detection operators. In *2014 International Conference on Embedded Systems (ICES)*, pages 7–11. IEEE, Jul 2014.

- [35] Patricia Melin, Claudia I. Gonzalez, Juan R. Castro, Olivia Mendoza, and Oscar Castillo. Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Transactions on Fuzzy Systems*, 22(6):1515–1525, Dec 2014.
- [36] Ivana Despotovic, Ewout Vansteenkiste, and Wilfried Philips. Spatially coherent fuzzy clustering for accurate and noise-robust image segmentation. *IEEE Signal Processing Letters*, 20(4):295–298, Apr 2013.
- [37] Ginmo Chung and Luminita A. Vese. Image segmentation using a multilayer level-set approach. *Computing and Visualization in Science*, 12(6):267–285, Aug 2009.
- [38] Luminita A. Vese and Tony F. Chan. A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293, Dec 2002.
- [39] Torsten Rohlfing and Clavin R. Maurer Jr. Multi-classifier framework for atlas-based image segmentation. *Pattern Recognition Letters*, 26(13):2070–2079, Oct 2005.
- [40] François Mendels, Conor Heneghan, and Jean-Philippe Thiran. Identification of the optic disk boundary in retinal images using active contours. In *Proceedings of 1999 Irish Machine Vision and Image Processing Conference (IMVIP)*, pages 103–115. IEEE, 1999.
- [41] Guillermo Sapiro. Color snakes. *Computer Vision and Image Understanding*, 68(2):247–253, Nov 1997.
- [42] Winggang Wu, Yong Gan, Bin Lin, Qiuwen Zhang, and Huawen Chang. An active contour model based on fused texture features for image segmentation. *Neurocomputing*, 151:1133–1141, Mar 2015.
- [43] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, Sep 2004.
- [44] Elena Salvador, Andrea Cavallaro, and Touradj Ebrahimi. Shadow identification and classification using invariant color models. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 3, pages 1545–1548. IEEE, May 2001.
- [45] D. Dunn, W.E. Higgins, and J. Wakeley. Texture segmentation using 2-D Gabor elementary functions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):130–149, Feb 1994.
- [46] J. Mao and A. K. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, Feb 1992.

- [47] Paul R. Hill, C. Nishan Canagarajah, and David R. Bull. Image segmentation using a texture gradient based watershed transform. *IEEE Transactions on Image Processing*, 12(12):1618–1633, Dec 2003.
- [48] V. S. Roshni and G. Raju. Image segmentation using multiresolution texture gradient and watershed algorithm. *International Journal of Computer Applications*, 22(6):21–28, May 2011.
- [49] Demin Wang. A multiscale gradient algorithm for image segmentation using watersheds. *Pattern Recognition*, 30(12):2043–2052, Dec 1997.
- [50] Song Chun Zhu and Alan L. Yuille. Region competition: unifying snakes, region growing, energy/Bayes/MDL for multi-band image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(9):416–423, Sep 1996.
- [51] Todd K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, Nov 1996.
- [52] Marco Andreetto, Lihi Zelnik-Manor, and Pietro Perona. Non-parametric probabilistic image segmentation. In *11th IEEE International Conference on Computer Vision*, Nov 2007.
- [53] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Advanced Texts in Econometrics. Oxford University Press, Jan 1995.
- [54] Gail A. Carpenter, Sucharita Gopal, Scott Macomber, Siegfried Martens, Curtis E. Woodcock, and Janet Franklin. A neural network method for efficient vegetation mapping. *Remote Sensing of Environment*, 70(3):326–338, Dec 1999.
- [55] David Fumo. A gentle introduction to neural networks series - part 1. Online <https://towardsdatascience.com/a-gentle-introduction-to-neural-networks-series-part-1-2b90b87795bc>, August 2017.
- [56] B. S. Manjunath and Rama Chellappa. A unified approach to boundary detection: edges, textures, and illusory contours. *IEEE Transactions on Neural Networks*, 4(1):96–108, Jan 1993.
- [57] J. Steven Perry. Create an artificial neural network using the neuroph java framework. Online <https://www.ibm.com/developerworks/library/cc-artificial-neural-networks-neuroph-machine-learning/index.html>, January 2018.
- [58] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. *International Journal of Computer Vision*, 125(1-3):3–18, Dec 2017.
- [59] Giles M. Foody, N.A. Campbell, Nigel M. Trodd, and T.F. Wood. Derivation and applications of probabilistic measures of class membership from the maximum likelihood classifications. *Photogrammetric Engineering and Remote Sensing*, 58(9):1335–1341, Sep 1992.

- [60] Alfred Stein, N.A.S. Hamm, and Qinghua Ye. Handling uncertainties in image mining for remote sensing studies. *International Journal of Remote Sensing*, 30(20):5365–5382, Sep 2009.
- [61] Curtis E. Woodcock. Chapter 2: Uncertainty in remote sensing. *Uncertainty in remote sensing and GIS*, pages 19–24, Dec 2002.
- [62] Xi Zhao, Alfred Stein, Xiaoling Chen, and Xiang Zhang. Quantification of extensional uncertainty of segmented image objects by random sets. *IEEE Transactions on Geoscience and Remote Sensing*, 49(7):2548–2557, Jul 2011.
- [63] Douglas K. McIver and Mark A. Friedl. Estimating pixel-scale land cover classification confidence using nonparametric machine learning methods. *IEEE Transactions on Geoscience and Remote Sensing*, 39(9):1959–1968, Sep 2001.
- [64] Patrick Bogaert, François Waldner, and Pierre Defourny. An information-based criterion to measure pixel-level thematic uncertainty in land cover classifications. *Stochastic Environmental Research and Risk Assessment*, 31(9):2297–2312, Nov 2017.
- [65] Khoobaram S. Choudhari, Pacheeripadikkal Jidesh, and Parampalli Sudheendra. Quantification and morphology studies of nanoporous alumina membranes: A new algorithm for digital image processing. *Microscopy Microanalysis*, 19(4):1061–1072, Aug 2013.
- [66] Marco A. Lopez, A. Braidot, Anbal Sattler, Claudia Schira, and E. Uriburu. An evolutionary algorithm for the segmentation of muscles and bones of the lower limb. *Journal of Physics: Conference Series*, 705(1), Apr 2016.
- [67] Morris H. DeGroot and Mark J. Schervish. *Probability and Statistics*. Pearson, Boston, 4th edition, Jan 2011.
- [68] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2), Feb 2009.
- [69] I.T. Jolliffe and B.J.T. Morgan. Principal component analysis and exploratory factor analysis. *Statistical Methods in Medical Research*, 1(1):69–95, Mar 1992.
- [70] Marylesa Howard, Margaret C. Hock, B.T. Meehan, and Leora E. Dresselhaus-Cooper. A locally adapting technique for edge detection using image segmentation. *SIAM Journal on Scientific Computing*, 40(4):B1161–B1179, Aug 2018.
- [71] Ben Derrick, Deirdre Toher, and Paul White. Why Welch’s test is Type I error robust. *The Quantitative Methods in Psychology*, 12(1):30–38, Jan 2016.
- [72] Julian Frank and Bernhard Klar. Methods to test for equality of two normal distributions. *Statistical Methods & Applications*, 25(4):581–599, Nov 2016.

- [72] B.J. Jensen, S.N. Luo, D.E. Hooks, K. Fezzaa, K.J. Ramos, J.D. Yeager, K. Kwiatkowski, T. Shimada, and D.M. Dattelbaum. Ultrafast, high resolution, phase contrast imaging of impact response with synchrotron radiation. *AIP Advances*, 2(1), Mar 2012.
- [73] B.J. Jensen, C.T. Owens, K.J. Ramos, J.D. Yeager, R.A. Saavedra, A.J. Iverson, S.N. Luo, K. Fezzaa, and D.E. Hooks. Impact system for ultrafast synchrotron experiments. *Review of Scientific Instruments*, 84(1), Jan 2013.
- [74] MathWorks. Function: edge. Online <http://www.mathworks.com/help/images/ref/edge.html>, 2018.
- [75] Hamed Mehrara, Mohammad Zahedinejad, and Ali Pourmohammad. Novel edge detection using bp neural network based on threshold binarization. In *2009 Second International Conference on Computer and Electrical Engineering*, volume 2, pages 408–412. IEEE, Dec 2009.
- [76] Morteza Mohammady Gharasuie. Edge Detection Back Propagation ANN. Online <https://github.com/mortezamg63/Edge-Detection-Back-Propagation-ANN>, October 2017.
- [77] Thomas Pezeril, Gagan Saini, David Veysset, Steve Kooi, Piotr Fidkowski, Raul Radovitzky, and Keith A. Nelson. Direct visualization of laser-driven focusing shock waves. *Physical Review Letters*, 106(21), May 2011.
- [78] Katharina Kohse-Höinghaus and Jay Barker Jeffries. *Applied Combustion Diagnostics*, volume 6 of *Combustion: An International Series*. CRC Press, Taylor & Francis Group, New York, NY, April 2002.
- [79] Gary S. Settles. *Schlieren and Shadowgraph Techniques: Visualizing Phenomena in Transparent Media*. Experimental Fluid Mechanics. Springer-Verlag, Berlin, Germany, 2nd edition, 2006.
- [80] Susanna W.M. AuYeung. Finding probability distributions from moments. *Master's Thesis, Imperial College, London*, Sep 2003.
- [81] Paul G. Hoel, Sidney C. Port, and Charles J. Stone. *Introduction to Statistical Theory*. Statistics. Houghton-Mifflin, 1st edition, 1971.
- [82] Richard M. Dudley. The Shapiro-Wilk and Related Tests for Normality. Online <https://math.mit.edu/~rmd/465/shapiro.pdf>, Feb 2015. MIT Course 465 Lecture Notes.
- [83] Stephen H. Friedberg, Arnold J. Insel, and Lawrence E. Spence. *Linear Algebra: Pearson New International Edition*. Pearson Education, Inc., Upper Saddle River, NJ, 4th edition, 2013.