# REDUCING TRAINING EFFORT IN BIOLOGICAL IMAGE CLASSIFICATION

by

Nhat 'Rich' Nguyen

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2016

Approved by:

_____
Dr. Min C. Shin

_____
Dr. Richard Souvenir

_____
Dr. Zbyszek Ras

_____
Dr. Andrew Willis

_____
Dr. Mark Clemens

# ABSTRACT

NHAT 'RICH' NGUYEN. Reducing training effort in biological image classification.
(Under the direction of DR. MIN C. SHIN)

To automatically classify biological images, machine learning techniques have been widely used to train the classifiers from labeled images. For a new category of biological object, a tedious and expensive labeling process is needed from a human expert. With the growing amount of biological data and the increasing number of categories to recognize, a more efficient method to train the classification system is required. The aim of this dissertation research is to effectively reduce the labeling effort of human experts in training the image classification methods. The contributions of this research consist of the following key components: First, the size differential regularization is employed to refine the ranking of classification rules to alleviate the risk of over-fitting in the case of a small number of training samples. Second, the spatiotemporal connectivity among the unlabeled samples is utilized to determine the weighting scheme of the existing classifiers from multiple sources. Third, the target directed sampling is proposed to focus the search for additional samples which are most likely to belong to the new class. The approaches are demonstrated to be effective in biological experiments including cell detection, insect detection, and pollen classification. The experimental results indicate that the proposed methods can achieve comparable performance to the current machine learning approaches while significantly reduce the amount of training data.

# ACKNOWLEDGMENTS

First and foremost, I am deeply grateful for my research advisor, Dr. Min C. Shin, for his continual, enthusiastic and optimistic direction throughout my academic career. I could not have completed my doctoral research without his guidance.

I appreciate Dr. Richard Souvenir, for his meticulous attention to details and high standards for quality of work. I express my gratitude to committee member, Dr. Mark Clemens, for his continued support with research opportunities and for giving me valuable inputs regarding to my research. I would also like to thank Dr. Zbyszek Ras, and Dr. Andrew Willis, for their expert interpretation for my research findings.

It was my extreme privilege and honor to complete this work under the funding provided by the National Science Foundation (NSF-DBI-0754748). I would like to acknowledge Dr. Toan Huynh, Dr. Matina Donaldson-Mastasci, and Dr. Eric Norris for collecting and providing the ground-truth marking for the biological data in this dissertation.

I would like to express gratitude to my supervisors, both present and former, Dr. Manuel Perez-Quinones and Dr. Richard Lejk for their support and guidance for me to keep pursuing the dissertation research while I am employed full-time at UNC Charlotte. I also appreciate my doctoral follow students, Scott Spurlock, Thomas Fasciano, Ayman Hajja, Hui Wu, and Hakim Touati for sharing their valuable knowledge and advices.

Finally, I give sincere thank my family, especially my fiance, Annie Ha, and all my friends for their overwhelming support and encouragement. I am grateful to have you all in my life.

TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# CHAPTER 1: INTRODUCTION

## 1.1 Biological Image Classification

Along with the maturity of information technology in biological fields and the emergence of *in-vivo* tissue staining, the number of biological images (e.g., single-organism images as well as cellular and molecular images) acquired in digital forms is growing rapidly [51]. The advances in imaging equipment have resulted in a significant number of images available to biologists and medical researchers and consequently, have led to a need for the automatic image analysis. Figure 1.1 illustrates an example of many biological images that need to be analyzed. In this figure, the different types of pollen is revealed under a microscope with diverse shapes and textures [1]. Analyzing these images is critical to find answers to many important questions in various fields of the life sciences including cell biology, developmental biology, and the medical sciences [79, 7].

For cell biologists, studying the cell population in the microscopy images enables the quantification of the cellular behaviors in response to different biological stimuli. For instance, images captured under a fluorescent microscope permit statistical analysis of various cell parameters such as apoptosis, adherence, morphology, and motility [36]. Thus, it has the potential to identify even subtle effects of many physiological stimuli on many cell types that keep humans healthy and shed light on new ways to treat disease. For ecologists, images of pollen grains provide a record of different flowers pollinators such as

[1] provided by Dr. Matina Matasci's group at the University of Arizona

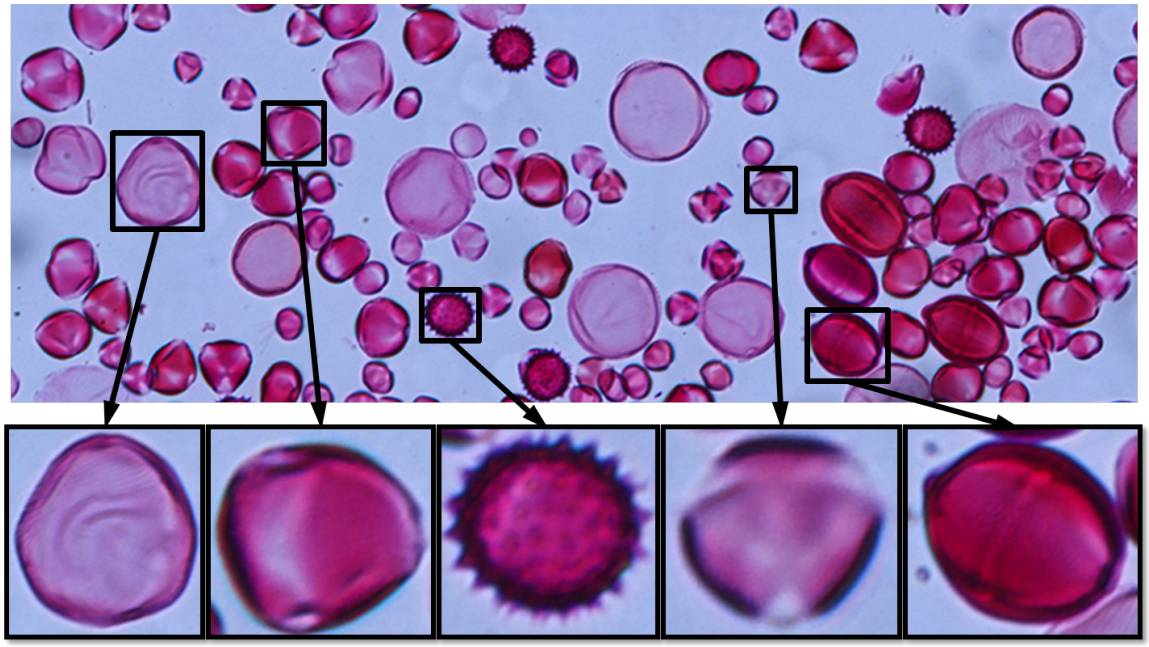


Figure 1.1: Top: a typical image of pollen contains a dense population of many pollen types viewed under a microscope. Bottom: several representative samples from different pollen types are enlarged to show a diverse shapes and textures among the pollen population. Automatic classification methods enable analysis of a large amount of image data without a biologist spending hours of labeling.

honeybees and butterflies have visited. Analysis of these images opens a window into the complex network of interactions between plants and pollinators in a community [38]. In agriculture and conservation biology, studying pollen images have practical implications for which plants are receiving pollination services, as well as the nutrition and health of the pollinators themselves.

Classification is an important technique for image analysis. It enables automatic prediction of a large number of unseen images which are usually required to give sufficient data to analyze in a large scale experiment. While gathering large quantities of images has become relatively fast and easy, categorizing them into different *classes* remains slow and time-consuming. Biology technicians usually spend hours in the laboratory to manually label each biological object into a specific class. Considering the labeling process is too te-

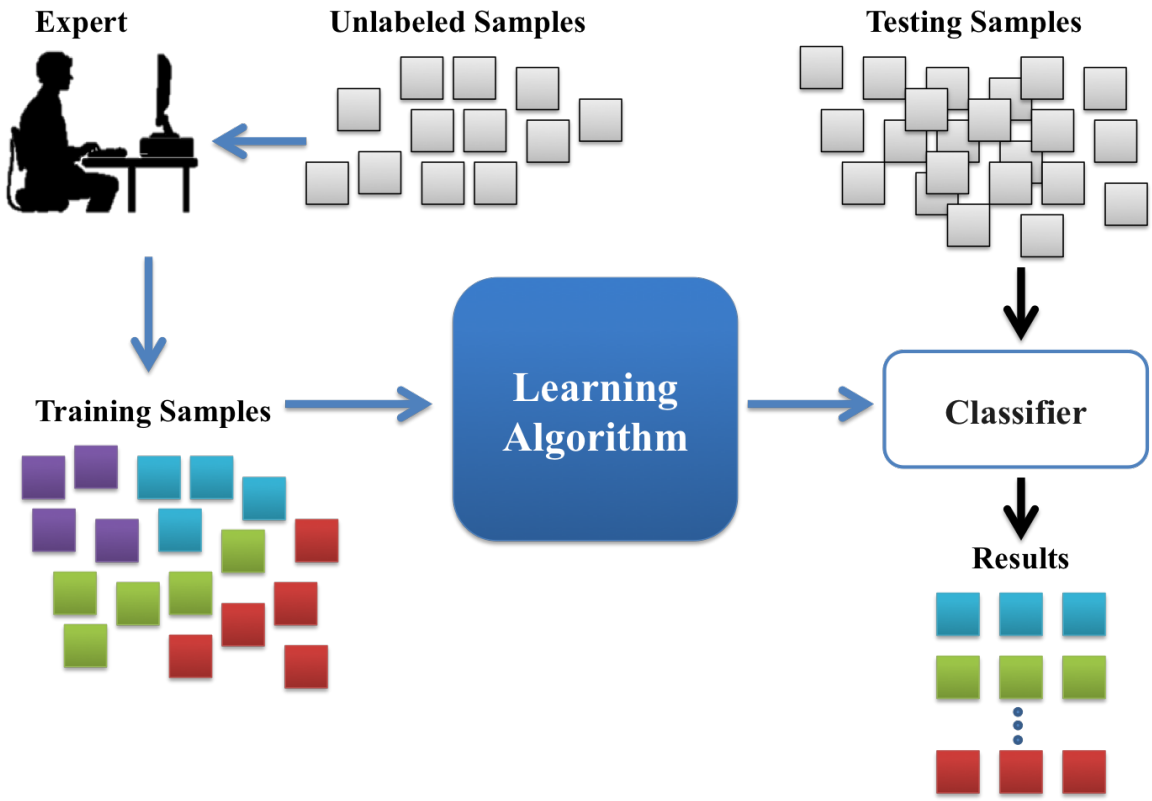



Figure 1.2: Automatic Classification Procedure.

dious, automated classification methods have been applied to analyze thousands of images. Supervised learning algorithms [82, 86, 47, 81] have been utilized to train the classification model given the labeled images as *training samples*. Since classification methods require the annotation of a number training samples which takes a significant amount of human effort, a major bottleneck becomes the laborious process of annotating the labels for these training samples. For the rest of the dissertation, this process will be referred as the *training effort*. A typical automatic classification procedure is illustrated in Figure 1.2.

With the diverse nature of biological classes to classify, obtaining training samples for automatic classification methods tends to be expensive in terms of training effort. Training samples could be obtained using crowd-sourcing label acquisition systems such as Ama-

zon Mechanical Turk (AMT) which allows any person to label the object images. For the majority of traditional object categories (e.g., car, table or chair), AMT has provided cost-effective training samples by a non-expert person. Unfortunately, these label acquisition systems are not suitable for biological data as the recognition of training images usually requires the skills and knowledge of a professionally trained expert. Consequently, the biology experts must obtain additional training samples on new image data. As it becomes increasingly important to save time and labor of biology experts in training the classifier, there is a great need for a classification method that can significantly reduce the training effort while producing a comparable performance to traditional machine learning approaches on new datasets.

## 1.2 Challenges in Reducing Training Effort

This dissertation aims to reduce the number of training samples that require specific expert knowledge in order to classify the biological images. However, reducing the number of training samples for automatic classification poses a few challenges:

First, training the classifier without a sufficient number of samples could often lead to poor performance because it is susceptible to overfitting [70]. For example, a classifier is overfit when it is 100% accurate on the training data but only 60% accurate on test data; in fact, it could have been 80% accurate on both. Overfitting can be analyzed in terms of the bias-variance trade-off [20]. Figure 1.3 explains overfitting as the relationship between bias and variance. When the size of training data is relatively small, the *variance* of the classification model is high although its *bias* on the training data is small. High-variance learning methods may be able to represent their training set well, yet are at risk

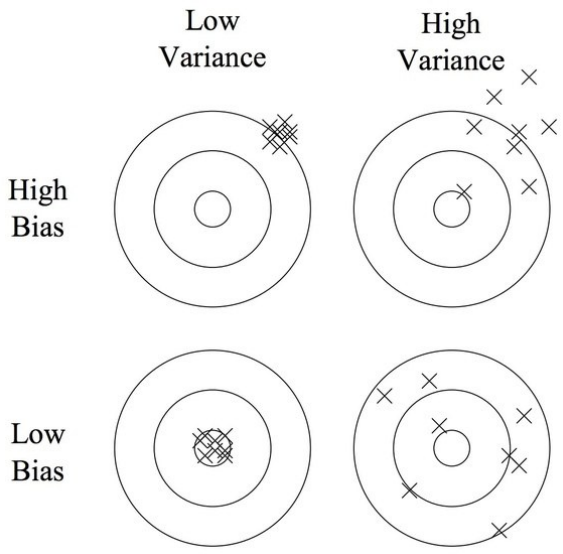



Figure 1.3: A visualization of bias and variance using a bulls-eye diagram of four different cases representing combinations of both high and low bias and variance (adapted from [21]). Imagine that the center of the target is a model that perfectly predicts the correct values. As we move away from the bulls-eye, our predictions get worse and worse. Underfitting is shown as high bias low variance while overfitting is shown as low bias high variance.

of overfitting to noise or unrepresentative training data [21]. In contrast, algorithms with high bias typically produce simpler models that don't tend to overfit, but may underfit their training data, failing to capture important regularities. Due to an insufficient number of training samples, the overfitting issue happens as the classification model becomes too sensitive to a few noisy training samples. However, the number of available samples of a *new* biological class can be limited when the class is encountered for the first time in a biological experiment. This condition can also happen when the setting changes in image capturing equipment and imaging modalities, or the image data is captured differently by separate research groups. As a result, the limited samples from the new class may be insufficient to describe its properties or to be discriminated from other classes.

Second, the distribution of training samples in biological images is usually unbalanced.

For example, images available for tumor cells may be much less than normal cell types. On a highly unbalanced class distribution, it is particularly time-consuming to obtain these samples when the human experts have to search for them in a significant number of *unlabeled samples*. While the samples of some classes are rather abundant, other classes can have significantly less training samples. Thus, a majority of training effort is spent on searching for appropriate samples to be labeled. Since the labeling cost is expensive, the labeling should only be done on the most useful training samples from a certain class that needs additional training samples. The effort required to obtain training samples of a biological class can be related to its distribution in the available set of unlabeled samples.

## 1.3 Main Contributions

Motivated by the challenges above, this dissertation provides a few approaches to reducing the effort by the human expert in training the classifier. In particular, the proposed research reduces the number of labeled samples by leveraging previously learned knowledge: the *existing classes* that are acquired from other experiments. The existing classes are assumed to be abundant and may contain some useful information to help improve the classification of the new class. In such cases, the use of this knowledge can save a significant amount of labeling effort. The reason behind this is that classification models among visually correlated categories are strongly inter-related, and the generalization power of the classifier is determined by the correlation between two sets of training samples [24]. In this classification framework, the problem is known as *transfer learning*, and the new class is also known as the *target* class, and existing classes are the *source* classes [59].

This dissertation extends the transfer learning framework by Yao and Doretto (2010) [86]

in three ways. First, we incorporate a size regularization into the transfer learning model to avoid overfitting when the training data is limited. Second, we minimize the difference in predicted labels between two samples which are spatiotemporally connected and construct a graph to learn the weighting scheme of the existing classifiers from multiple sources to supplement the small number of labeled samples. Third, we combine the transfer learning model with a sampling method to provide a more efficient way to select additional training samples. Specifically, the research contributions of this dissertation consist of the following key components:

- **Size-distribution Regularization:** To alleviate the risk of over-fitting in the case of a small number of training samples, a size distribution regularization is used to refine the ranking of classification rules. In such cases, the rules should rely on a separate estimator which can be computed from a collection of training samples. The selection of the classification rules is intended to conform with a side-distribution regularization in addition to minimizing the empirical training error. In the biological cell context, we particularly choose the cell size distribution as a regularization because it represents a common biological characteristic of cells which can quickly be estimated from the training images; thus it requires minimal effort to measure manually. Several different cell datasets are evaluated to demonstrate the applicability of our approach.

- **Spatiotemporal Regularization:** Another way to combat overfitting is leveraging the information from multiple images in a video. The regularization can penalize classifiers with more structure, thereby favoring smaller ones with less room to over-

fit. In this dissertation, we propose a transfer learning method that employs the spatiotemporal relationship among the unlabeled data. The spatiotemporal consists of the pixel coordinate as well as the time frame in the video from which an image patch is obtained. The unlabeled samples are the image patches available abundantly in the video and require no manual labeling effort. We estimate the spatiotemporal connectivity between pairs of unlabeled samples using the optical flow throughout the video. Our approach is based on the assumption that two unlabeled samples which are connected by the optical flow should have the same predicted label. This motivates us to minimize the difference in predicted labels between two samples which are spatiotemporally connected. The evaluation on three data sets of social insects such as ants and termites demonstrates that the proposed method is able to reduce the training effort while maintaining comparable accuracy to previous approaches.

- **Target-Directed Sampling to Select Training Data:** To effectively select additional training samples, Target-Directed Sampling (TDS) is proposed to focus the search toward the samples of the new (target) class. Initially, the number of training samples from the target class is small, additional target samples are more beneficial to the classifier than other samples. However, target samples are more difficult to find due to the imbalanced in training data. TDS is an active learning approach, as a more specific case of margin sampling, to select the most useful training samples from the unlabeled data [48, 73]. Particularly, the unlabeled samples which are most likely to be confused between a target class and another class are chosen *first* for training. Additionally, the confidence which is employed by TDS is constructed from the

classification rules extracted from the existing classes. These classification rules are assumed to be reliable enough to build the initial classifier. The approach has been successfully applied to classify various pollen classes.

Experimental evaluation validates that the proposed method requires significantly less training effort than some widely used learning approaches.

## 1.4 Dissertation Outline

The remainder of this dissertation is organized as follow. As the objective of the proposed research is to reduce training effort, Chapter 2 reviews relevant work on reducing training effort in both leveraging knowledge from existing data and selecting more effective training samples. The chapter also discusses the benefits and issues associated with the approaches, motivating the need for the proposed research.

Chapter 3 establishes formal definitions and mathematical constructs of boosting methods which are used extensively throughout this dissertation research. This chapter also describes a transfer learning framework necessary to understand the proposed methods.

Chapter 4 presents a regularization of the new class to refine the ranking of its classification rules to avoid over-fitting in the case of a small number of new training data. The size distribution is chosen as a regularization because it can effectively quantify the characteristics of object population in an image. For each cell type, the regularization is determined during the training step and does not need to be re-trained for each image.

Chapter 5 utilizes the spatiotemporal connectivity of the unlabeled data to regulate the training of a detector on a new insect type. Our key contribution is integrating the spatiotemporal connectivity among the unlabeled samples to determine the weighting scheme

of the existing classifiers from multiple sources. The evaluation on 3 data sets of social insects demonstrates that our method can achieve comparable performance to previous approaches while reducing the training labels.

Chapter 6 proposes a *target-directed sampling* method designed to reduce the amount of training effort required to classify a new object class. In order to reduce the number of labeled samples, auxiliary knowledge is exploited in two ways: First, the existing data is leveraged to build the classifier with a small number of target samples. Second, unlabeled samples that are likely to be valuable to the training data are identified.

Finally, Chapter 7 summarizes the future exploration of this dissertation research. This chapter also outlines open problems and directions for future research on reducing training effort and relates this dissertation to future studies.

# CHAPTER 2: RELATED WORK

Automated image classification is an important field for a broad range of applications because it enables statistical analysis in large-scale biological experiments [36]. In recent years, there has been a strong research interest in the classification methods for biological objects in images [36, 40, 58, 53], some of which have resulted in software applications [10, 41]. In this chapter, we will first discuss the construction of feature representations which have been used in the context of biological images (Section 2.1). Subsequently, we will explore different training methods for biological image classification (Section 2.2). Finally, we will establish some literature review on various training reducing methods which motivate this dissertation (Section 2.3).

## 2.1 Local Feature Representation

Many classification methods have focused on developing some feature descriptors that work well for a particular object type. Morphological operations such as multi-scale correlation kernels [1, 75], two-level adaptive thresholding [4], and weighted medial axis transform [33] can result in unsuccessful classification over a large range of object appearance. Depending on the biological application, active contour methods [85] and sliding window methods [22] has been applied to build the feature representation in biological images.

### 2.1.1 Active Contour Methods

In the context of biological images, classification has also been performed using an active contour to model the bright halo which surrounds the object, then extract representative features from the region within the boundary [50, 61, 62]. To localize the precise the boundary of each object, we utilize active contours proposed by Xu *et al.* [85] which have been used extensively in many applications. Active contour methods are able to segment the object boundary, but requires manual initialization for each individual objects. The initialization is often a time-consuming process and require much effort from the human experts. In this dissertation, we automatically estimate the initial boundary of active contours by detecting circles using the circular Hough transform (CHT) [88] under the assumption that the shapes of some biological objects (such as cells and pollens) are varied from circular to elliptical. The detection accuracy of circular Hough transform (CHT) [88] also depends on a threshold to compromise number of actual grains and errors. This threshold is difficult to determine because the image of pollen usually exhibits a high density of grains with different intensity values as well as many inhomogeneous background region. From the initial boundary estimation, a snake iteratively conforms to the boundary according to the internal energy (elasticity and rigidity) as well as the external energy which is computed from the gradient vector flow. We compute the GVF from an edge image obtained by the Canny edge operator on a Gaussian smoothed image. The parameters of the Hough transform and the active contour are empirically selected based on visual inspections and used for all images.

### 2.1.2 Sliding Windows Methods

To separate an object in an image, sliding window approaches have been well established in the literature [14, 25]. Typically, the differ in the form in which the location of an object is represented as bounding box, center point, or contour. Because the number of rectangles in an $n \times n$ image is of the order $n^4$, this process usually cannot be done exhaustively. Instead, several heuristics have been proposed to speed up the search. Typically, these consist of reducing the number of necessary function evaluations by searching only over a coarse grid of possible rectangle locations and by allowing only rectangles of certain fixed sizes as candidates [64]. When biological objects are overlap and in clusters, the classification problem becomes even more challenging. In our cases, the image used for training was selected in such way that minimizes the overlap of cells and therefore did not heavily affect the experiment. However, in cases where there are clusters among the classifying objects, segmenting the neighbor objects can be done by procedures presented in [1, 22, 87]. Recently, a rich set of carefully designed feature descriptors which adapt to the object appearance has been proposed to detect different types of objects [58]. However, the training task is still highly difficult when it is applied to various object types without some effort of parameter tuning.

### 2.1.3 Feature Descriptors

Even when taken with advanced biological imaging systems such as phase-contrast microscopy, the appearance of different biological objects types are highly diverse. To represent the diverse appearance of these objects, many of early studies on building feature types have been proposed and explored, including lines [32], edges [52], corners [90], regions

[8], among many other. While these features have worked well for certain object classes, using a single feature representation cannot handle the diversity and therefore is difficult to achieve high performance with unseen images from a new object type. In the following, we will discuss the recent work on developing a collections of features for biological objects based on boundary, texture, region, and gradient.

**Boundary-based** Previous studies have compiled a set of shape and texture features derived from the boundary [45, 3, 63]. From the boundary constructed by the active contour, we construct a feature descriptor consisting of 34 features based on pollen boundary shape, internal texture. The shape features computed from the boundary of the active contours include area, diameter, ratio of area and perimeter, compactness, roundness, rates of changes, thickness, elongation, centroid, Euclidean norm, mean size, eccentricity, and circularity. While shape features utilize the extracted boundary, the texture features are derived from the rectangular region which encloses the extracted boundary. Our texture features include the first-order statistics, Haralick's Coefficients from the gray level occurrence matrix (GLCM), and the gray-level run length. The computation details of these features are described in [63].

**Texture-based** The first procedure is first-order statistics which contained the distribution of the gray level in the region such as mean, variance, statistical moments, energy, and entropy of the gray-level. The second procedure is based on the gray level co-occurrence matrix (GLCM). From the GLCM, we computed the Haralick's Coefficients such as Energy, Entropy, Correlation, Inverse Difference Moment, Inertia, cluster shade, and cluster prominence [74]. The third procedure is the gray-level-run which consists of a set of con-

secutive pixels in a image with the same gray value [13]. Then, several texture features such as the short run emphasis, long run emphasis, gray level non-uniformity, run length non-uniformity and run percentage.

**Region-based** We adapted the features from previous work [53, 1, 54] since they were shown to work well for cell detection task. Depending on the application, additional features could be added for potential improvement. We briefly described the features as follow. First, the normalized radial mean response of sample $i$ was computed as the ratio between the mean intensity of the inner and outer circular regions surrounding the pixel location of $i$. Six different scales for the inner and outer regions were applied to accommodate a variety of cell appearances. Second, the mean of gradient magnitude was calculated within a square window around sample $i$. The length of the square window was same as the outer region from above. Third, filtering responses of sample $i$ were collected from circular averaging, low-pass Gaussian and isotropic Laplacian of Gaussian kernels. The response values were normalized by dividing with the maximum value within the entire image.

**HOG-based** In our experiment of insect classification, the features are computed based on the Histogram of Oriented Gradients (HOG) feature descriptor [18]. HOG was formulated based on an idea that local object appearance and shape can often be characterized adequately by the distribution of local intensity gradients or edge directions, without prior knowledge of the corresponding gradient or edge positions. HOG has been used widely for human detection application, here we apply it to biological images. Specifically, the biological image is divided into small $m \times m$ pixel spatial region called image patch. Each image patch is divided into 4 blocks each block has $3 \times 3$ cells with one overlapping cell

between adjacent blocks. Each cell has a size of $p \times p$ pixel with 8 bins contains the orientation histogram. Note that depending on the applications, additional features can be incorporated.

## 2.2 Biological Images Classification Methods

The training of biological image classification is still a tedious and time-consuming process that must be performed by highly skilled biologists who specialize in their field of study. Recently, several groups have developed training methods for biological classification. Most training methods involve a small number of object classes and require a large count of training labels per class. [45, 63, 38]. As the number of types increases, it seems that more and more training labels are required for each class. For example, a pollen classification system proposed Allen *et al.* requires 40 training labels per type to classify 7 types, but as many as 150 training labels per type for 17 types [3]. To build a more comprehensive classifier that could deal with an increasing number of types over time, the number of samples required for training would thus likely be prohibitive. Another study has shown good performance up to 30 classes with as few as 18 training samples per class [9]; however, the classification rules are derived by their human experts. Such an approach is likely to scale poorly as the number of classes increases, because the complexity of the classification model and the amount of expert knowledge required to build them will grow rapidly as more similar classes are added. In the following sections, we will discuss the training of three major classification methods including boosting, support vector machines, and convoluted neural network.

### 2.2.1 Boosting Methods

Boosting has been extensively successful method for image classification. Since its first introduced, AdaBoost has been shown to be equivalent to forward stage-wise additive modeling method that minimizes the exponential loss [29]. The AdaBoost algorithm is an iterative procedure that tries to approximate a strong classifier by combining many weak classifiers. If a training data point is misclassified, the weight of that training data point is increased (boosted). A subsequent classifier is built using the new weights, which are no longer equal. Again, misclassified training data have their weights boosted and the procedure is repeated. Typically, one may build 100 or 1000 classifiers this way. A score is assigned to each classifier, and the final classifier is defined as the linear combination of the classifiers from each stage. As an successful example of boosting based methods, Viola *el al.* use AdaBoost to build an efficient moving person detector to train a chain of progressively more complex region rejection rules based on Haar-like wavelets and space-time differences [82]. AdaBoost is similar to most traditional machine learning methods by assuming the distributions of training and test data to be identical. In this dissertation, we extend AdaBoost to transfer knowledge from the source domains and add regularization on unlabeled samples.

Decision Stumps Decision stump is one level decision tree that classify image samples by sorting them based on feature values. Each node in a decision stump represents a feature of the sample to be classified, and each branch represents a value that the node can take. Trained on a single dimension in the feature vector of a sample in an image, the decision stump can produce a label of either object or non-object with better accuracy than a random

guess. At its worst, a decision stump will reproduce the most common sense baseline, and may do better if the selected feature is particularly informative. An exponential decrease of an upper bound of the training error rate is guaranteed as long as the error rates of the decision strmps are less than $\frac{1}{2}$ [29].

### 2.2.2 Support Vector Machines Methods

Aside from boosting based methods, Wu and Dietterich propose a support vector machines (SVM) framework for image classification [84]. Their method uses previously trained data, which they refer to as auxiliary data, to constrain the SVM learning and identify support vectors that are applicable to a target task. Farhadi *et al.* [27] use SVM to learn word signs consisted of SIFT features [46] for head and hands among different signers at frontal and three-fourth view for building word models for American Sign Language.

### 2.2.3 Convolutional Neural Network Methods

Recently, some detection methods which are based on machine learning algorithms have been proposed on a pedestrian dataset based on hand-crafted features [16]. Notably, some methods which are based on convolutional neural network (CNN) have shown impressive performance on object detection [42]. Although reliable detection can be achieved for a particular type of object, the training task remains difficult. For instance, the training of CNN usually requires a large number of labels to fine-tune the deep learning network [76]. Unlike general object data (such as PASCAL-VOC), biological data is less suitable for labeling based on crowd-sourcing (e.g. Amazon Mechanical Turks). Instead, the training task of biological data entails an extensive collection of samples labeled by the human experts. To save the expert's time in training detection methods, there is a great need for

a method that can significantly reduce the training labels while maintaining a comparable performance to previous approaches. To this end, transfer learning has been shown to reduce in training effort by leveraging the previously acquired data (source) to new data (target) [59, 17].

## 2.3 Reduction of Training Effort

The reduction of training effort has been studied extensively in machine learning and data mining fields [16, 59]. One way to reduce the human effort in labeling new training data is applying some *transfer learning* algorithms [67, 31, 26, 59] which leverage the knowledge from the existing classes (sources) to learn the new class (target). For a classification system, the ability to transfer previous knowledge has a potential of quickly learning new tasks to achieve the objectives, thus reducing users' training effort. Transfer knowledge is effective when the classification system is able to learn a new concept with only a few labeled samples available for training [59].

### 2.3.1 Transfer Learning

Early transfer learning works raised some important issues, such as learning to learn [71], learning one more thing [77], and multi-task learning [11].

Transfer learning can provide some combination of three possible benefits to learn the new task: better initial performance, more rapid learning, or higher achievable performance [24]. Figure 2.1 demonstrates the three benefits of transfer learning and illustrates their corresponding effects on a learning curve. Ideally, transfer learning would provide all three benefits in learning the new task. If the objective is to reduce training effort, transfer learning typically yields better initial performance or more rapid learning. Additional effort in

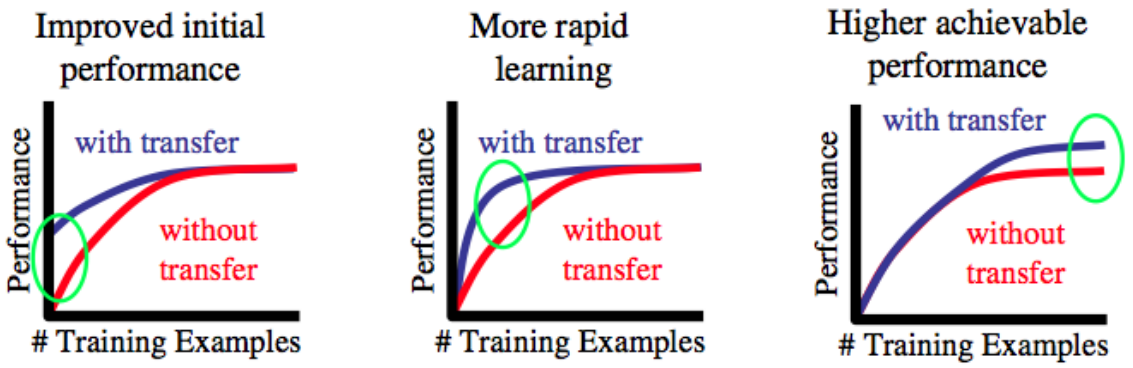



Figure 2.1: The three possible benefits of transfer and their effect on the learning curve of a new task, adapted from a 2005 DARPA publication.

training can be greatly reduced when the initial performance is almost satisfactory. Similarly, more rapid learning can help the classification method to achieve good performance with smaller number of training samples. As the right sub-figure suggests, transfer can enable a higher level of performance when the supply of training data is limited.

Research on transfer learning has been prolific in recent years over a wide variety of applications, such as sign language recognition [28], text classification [67], document categorization [2], and vehicle tracking [86]. Transferring from similar distribution sources can be intuitively understood: although the source classes cannot be reused directly, there are certain parts of the classes that can still be reused together with a few labeled data in the target [19]. Given a large training samples from the data which have been collected, transfer learning could use these training samples to build a classifier for the new data. Most recent research studies on transfer learning focus on what to transfer by implicitly assuming that the source and target domains be related to each other [59]. Based on what knowledge to leverage, we can distinguish three notable approaches: First, transferring instances from similar sources is especially useful when there is a lack of sufficient training data for the

target task, and the target task shares components of its underlying distribution with some of the source tasks. Second, sharing a common feature aims at finding "good" feature representations to minimize domain divergence and classification model error. Strategies to find effective feature representations are different depending on the types of the source data. If a lot of labeled data in the source are available, supervised learning methods can be used to construct a feature representation. Third, biassing toward previously learned model parameters can often be characterized by a vector of parameters that are fit to the training data; transfer can then bias the learning of a new model toward a specific set of parameters.

Among the first successful application instance transfer learning, TrAdaBoost proposed by Dai *et al.* employs a boosting approach to re-weight the source training samples according to their likelihood to be similar to the target samples [17]. The main idea of TrAdaBoost is to use boosting to lower the weight of the irrelevant samples in the source data in each iteration. The weights of the misclassified source samples are decreased to weaken their impact on the weak learner. The correctly classified samples, treated as the additional training data, boost the confidence of the learned model from the target data even when the number of target samples is small. TrAdaBoost discards the weak learners from the first half of the iterations, and the final classifier is constructed as the linear combination of weak learners from latter half of the iteration. However, in the boosting mechanism, each additional learner added by a boosting algorithm has, on average, less influence on the final classifier's prediction than its predecessors [69]. When existing data are shared among different groups, Dai *et al.* provide a model for transfer learning which is shown to be effective while using a small number of labels [17]. Their method extends the boosting framework to predict the label of the target data is simply to combine the training data from

the source with the target, then build a classifier based on the combined data. However, this approach will not work well with various type of biological objects because it relies on only one source at a time.

In transfer learning, weak relationship between source data and target data might lead to poor performance of the target classifier (negative transfer). Yao *et al.* argue that TrAdaBoost algorithm was vulnerable to negative transfer because it relies only on one source [86]. Thus, they extended the transfer algorithm into MultiSourceTrAdaBoost which transfers from multiple sources. The authors extracted a subset of training samples, coming from various available sources, that were the most closely related to the target. A selection criterion has been introduced such that a weak classifier is selected from the source that appears to be the most closely related to the target, at the current iteration. Negative transfer tend to be reduce using such approach because it overcomes the imposition to transfer knowledge from a single source, potentially loosely related to the target.

An alternative to the above approach is to learn the individual classifier for each source class, then combine these classifiers by some similarity distances between the source data and target data. As a notable method in this area, TaskTrAdaBoost [86] proposed by Yao *et al.* is closely related to our method. Along with the instance transfer mentioned above, Yao and Dorretto also presented a parameter transfer framework, TaskTrAdaBoost [86], that jointly selected classification rules from multiple sources. The method identified the parameters, which came from various sources, to be reused together with the target training data to improve the target classifier. The transferring parameters in this work were represented as weak classifiers. The authors assume that the new data will share some parameters with existing data and builds a classifier based on those parameters.

In address some limitations inherent in TrAdaBoost's design, Eaton introduces the TransferBoost algorithm [24] which employs set-based boosting technique to automatically select individual data from the source tasks to augment the target task's training data. Similar to TrAdaBoost, TransferBoost employs boosting scheme to automatically determine the weight assigned to each source samples in order to learn the target classifier. However, the algorithm boosts each source task based on a notion of *transferability* from the source task to the target task. Transferability is defined to be the change in performance on the target task between learning with and without transfer [23]. TransferBoost boosts each set of instances from the same task, increasing the weights of all instances from a source task if the source task shows positive transferability to the target task.

### 2.3.2 Regularization to Transfer Learning Models

Besides transferring classifier from multiple sources, several researchers investigated on a regularization framework to transfer parameters [26, 6, 5]. In a regularization framework one assumes the existence of task-specific parameters for each task and shared parameters that parameterize a family of underlying transformations. Both the structural parameters and the task-specific parameters are learned together via joint risk minimization on some supervised training data for related tasks.

Ando and Zhang [6] combine an $L_2$ regularization penalty on the task-specific parameters with an orthonormal constraint on the shared parameters. This transfer algorithm is applied in the context of asymmetric transfer where source training sets are utilized to learn the shared parameter. The shared parameter is then used to project the samples of the target and train a classifier on the new space. In their experiment on text categorization, the

source training sets were automatically derived from unlabeled data. More precisely, the source is consisted of predicting frequent content words for a set of unlabeled documents.

Evgeniou and Pontil proposed a regularization framework to transfer parameters of SVM [26]. The authors applied Hierarchical Bayesian (HB) framework for SVM from multitask learning. They modeled the relation between different tasks in terms of a novel kernel function that used a task coupling parameter. The presented method assumes that the parameter in SVM for each task can be separated into two terms: one is a common term over tasks and the other is a task-specific term. They utilize Hinge loss as the Loss function and the inner product of the parameters as regularization function.

Amit *et al.* proposed a regularization scheme for transfer learning based on a trace norm regularization penalty [5]. This norm is used because it is known to induce solution matrices $W$ of low rank. In addition to the primal formulation, the paper presented a kerneled version. It is shown that although the weigh vectors can not be directly retrieved from the dual solution, they can be found by solving a linear program on m variables. The authors conducted experiments on a multiclass image classification task where the goal is to distinguish between 72 classes of mammals. The performance of their transfer learning algorithm is compared to that of a baseline SVM multiclass classifier. Their results show that the trace-norm penalty can improve multiclass accuracy when only a few samples are available for training. Furthermore, when the new data is assumed to share some common parameters with existing data, Sarinnapakorn *et al.* propose a few transfer methods attempt to identify the shared parameters coming from various existing data in the form of classification rules [67]. These approaches have two potential shortcomings. First, they assume all sources classes are independent which means no interaction among multiple sources.

Thus, they minimize the empirical loss on the source training data separately which might not agree with the minimal loss on the target data. Second, they rely solely on the labeled training samples which require human effort and ignore the unlabeled target samples which can be useful in their variety and abundance. These shortcomings motivate the determination of the weights of the unlabeled target samples in respect of the classifiers learned from the source classes.

To combine multiple models, Chattopadhyay *et al.* estimate the weights of source classifiers based on the smoothness assumption to minimizing the difference in predicted labels between two nearby unlabeled samples in feature space [12]. Their method is able to integrate a large number of unlabeled data to supplement for the lack of labeled data into a loss minimization framework. Also following a similar approach, Gao *et al.* proposed a locally weighted ensemble (LWE) learning framework for parameter transfer [31]. They propose a graph-based approach to approximate the optimal model weights where the local weight for a source model is computed by first mapping and then measuring the similarity between the model and the target local structure around its labeled samples. This similarity is measured by comparing neighborhood graphs, and quantified in the weight assignment equation. They dynamically assigned weights of multiple classification models according to a model's predictive power on training samples in the target data. Table 2.1 summarizes the qualitative comparisons among the transfer learning methods. Our research focuses on reducing the number of training samples for transfer learning models using several regularizations. To the best of our knowledge, none of the previous methods takes into account the spatiotemporal connectivity between the pairs of unlabeled samples. We note that our method requires the connectivity of unlabeled samples collected through time, so its train-

Table 2.1: Qualitative comparisons of transfer learning approaches which transfer the classification models similarly to this research. The approaches in last three rows are the research of this dissertation.

| **Author** | **Transfer** | **Classifier** | **Domain** | **Regularization** |
|---|---|---|---|---|
| Ando [6] | Parameter | SVD | Text | $L_2$ |
| Gao [31] | Parameter | LWE | Text | Local Structure |
| Harpale [34] | Parameter | Regression | Text | Utility Gain |
| Rai [60] | Parameter | ANN | General | - |
| Amit [5] | Parameter | SVM | Text | Trace Norm |
| Evgeniou [26] | Parameter | SVM | General | Hier. Bayesian |
| Chattopadhyay [12] | Parameter | SVM | Biological | Feature Distance |
| Sarinnapakorn [67] | Parameter | Boosting | Text | - |
| Yao [86] | Parameter | Boosting | General | Utility Gain |
| Nguyen [54] | Parameter | Boosting | Biological | Size Distribution |
| Nguyen [55] | Parameter | Boosting | Biological | Target Directed |
| Nguyen [in review] | Parameter | Boosting | Biological | Spatio-Temporal |

ing is only possible only when training data is available in video form. In the following section, we employ this spatiotemporal position to regulate the training of a target classifier.

### 2.3.3 Active Learning to Select Training Samples

Although many approaches have successfully exploited the knowledge from the existing classes to construct the new classifier, little attention was paid on how to apply the same knowledge to select appropriate training samples on the new class. On the basis of selecting the most useful training samples which were most likely to improve the classification, many research studies demonstrated different *active learning* strategies which effectively reduce the training effort [49, 48, 39, 73]. Widely regarded as a querying technique, uncertainty sampling approach selected the instances which has the least confident about the label. Uncertainty sampling consisted of two components: training and querying. The training step could be employed by many kinds of classifiers such as decision tree [83], nearest neighbor [30], and support vector machines [78]. Recently, Lughofer compared

and showed that these classifiers performed comparatively in several classification systems [47]. The querying step selected unlabeled instances to be labeled by an expert, then added those instances to the training set to train the classifier again. In particular, there were three main strategies in uncertainty sampling: posterior entropy, least confidence and margin sampling.

To minimize the training set size, uncertainty sampling aimed to optimize the order in which the samples are labeled. In such way, the samples with the most information were labeled *first* [73]. Uncertainty sampling is a popular setting in real-world problems where large collection of unlabeled data (or *unlabeled pool*) can be collected at once. For example, active learning were applied in text classification [37], image classification and retrieval [89], speech recognition [80], and object recognition [81]. The main motivation behind active learning is that unlabeled samples are usually inexpensive and available in abundance, while annotating those samples can be expensive or time-consuming. Thus, the human annotator often wishes to select only the most informative samples to be labeled, thus reducing redundancy to some extent, compared to the baseline of selecting the examples randomly.

Roy *et al.* used a probability model to label example which maximizes the posterior entropy on the unlabeled dataset [65]. Although this method was applied with naive Bayes classifiers, it could be extended to any classifier with probabilistic output. Nuzhnaya *et al.* approximated the probability function that measured the likelihood of a given sample belong to a class in multi-class boosting framework using a sigmoid function [57]. If the objective is to reduce the classification error, Culotta *et al.* proposed a least confident strategy to prefer instances that would help the model better discriminate among specific

classes using conditional random field (CRF) [15]. The approach, however, only considered instances of the most probable label, and disregard to the distribution of other labels. To correct this problem, Lou *et al.* proposed a Margin query strategy, called "Breaking Ties", to select the instances that minimize the margin between the most and second most probable label [48]. Joshi *et al.* compared uncertainty sampling approaches and argued that margin query strategy, such as Breaking Ties, worked best on their selected datasets [39].

However, a recent study [47] argued that active learning strategies still required an initial classifier which already had a reasonable accuracy in order to decide which samples should be selected for further labeling and consequently offered an unsupervised approach for the training of the initial classifier. Unfortunately, the study ignored the situation where the labeled training samples from existing classes were available and could have been useful to train the initial classifier. The open problem remains about how to regularize transfer learning when the number of new samples is limited, and how to effectively select additional training samples based on the existing classes. This dissertation research focuses on these problems and contributes a few solutions to reduce training effort in biological image classification.

# CHAPTER 3: BACKGROUND

In this chapter we develop a general notation and background framework to obtain the necessary knowledge for understanding the remaining of the dissertation. First, we introduce some relevant background information for the classification problem (Section 3.1). Next, we describe the improvement in classification performance on a new class by using the rules from other existing classes (Section 3.2). Specially, we provide some details about a related transfer learning framework and its limitation that leads into our proposed solution. The size-differential regularization that we present in chapter 4 is constructed upon the transfer learning framework. Subsequently, we describe the incorporation of the regularization into a minimization framework which will be relevant in chapter 5. Finally, we describe the classification confidence which is used the sampling selection algorithm for additional training samples presented later in Chapter 6.

## 3.1 Notations and Frameworks

We first introduce the notation and define the problem statement. In the feature space $\mathcal{X} \in \mathcal{R}^N$ with $N$ dimensions correspond to the feature representation and the label space $\mathcal{Y} = \{-1, +1\}$, a classifier is trained by estimating a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. In order to train a classifier, samples consisted of the feature vector $\mathbf{x}_i \in \mathcal{X}$ with the manually label $y_i \in \mathcal{Y}$ are collected. The detection of an object of interest is in an image the same as to classify the image patch contained the object as a positive label $y_i = 1$ and otherwise as a negative

label $y_i = 1$. The target class consists of small labeled data $D_l^\tau = \{(\mathbf{x}_i, y_i)|1 \leq i \leq n_l\}$ and abundance of unlabeled data $D_u^\tau = \{(\mathbf{x}_j)|1 \leq j \leq n_u\}$ where $n_l$ and $n_u$ are the numbers of labeled and unlabeled target samples ($n_l \ll n_u$). Table 3.1 summarizes the notations used throughout this dissertation.

Table 3.1: Notations used in this dissertation and their explanation

| Notation | Explanation |
|---|---|
| $\mathcal{X}$ | The feature space $\in \mathcal{R}^N$ of $N$ dimensions |
| $\mathcal{Y}$ | The label space of $\{-1, +1\}$ |
| $\mathbf{x}_i$ | The feature vector of the $i^{th}$ sample |
| $y_i$ | The label of the $i^{th}$ sample |
| $\hat{f}$ | The estimated (composite) classifier |
| $\mathcal{H}$ | The set of possible classifiers |
| $h_t$ | The base classifier which is trained at boosting iteration $t$ |
| $\alpha_t$ | The weight of the base classifier at boosting iteration $t$ |
| $\epsilon_t$ | The empirical loss (training error) at boosting iteration $t$ |
| $D_t$ | The weight distribution of the labeled samples at boosting iteration $t$ |
| $k$ | Total number of source classes |
| $f^\tau$ | The target classifier |
| $f^s$ | The $s^{th}$ source classifier |
| $\mathcal{H}_c$ | The set of candidate classifiers which are trained from the source classes |
| $\mathcal{D}^s$ | The data of the $s^{th}$ source classes |
| $\mathcal{D}^\tau$ | The data of the target domain |
| $\mathcal{D}_l^\tau$ | The labeled data of the target class |
| $\mathcal{D}_u^\tau$ | The unlabeled data of the target class |
| $n^s$ | Number of samples in the $s^{th}$ class |
| $n^\tau$ | Number of total target domain samples |
| $n_l$ | Number of labeled samples |
| $n_u$ | Number of unlabeled samples |
| $\hat{y}_j$ | The estimated label or pseudo-label of the unlabeled sample $\mathbf{x}_j$ |
| $\lambda$ | The parameter controls the trade-off between loss and regularization |
| $\mathbf{B}^s$ | $k \times 1$ weight vector of the source classes |
| $\mathbf{F}_i^s$ | $1 \times k$ vector of predicted values of source classifiers for the $i^{th}$ sample |
| $\mathbf{W}_{i,j}$ | The weight matrix of member $w_{i,j}$ indicating the weight between the $i^{th}$ and $j^{th}$ unlabeled samples |

### 3.1.1 Adaptive Boosting

Boosting is an iterative method of constructing an accurate classifier by combining many weak classifiers, each of which may only need to be reasonably accurate [29]. One of the most popular boosting method, Adaptive Boosting (also known as AdaBoost), is used to train our classifier. Adaptive Boosting weights each base classifier based on its prediction accuracy. AdaBoost constructs an initial distribution of weights over the training set. Then, the boosting mechanism selects a base classifier that gives the least error, where the error is proportional to the weights of the misclassified samples. The training of the base classifier is executed similarly to the following minimization

$$\min_{h_t} \frac{1}{n_l} \sum_{i=1}^{n_l} \mathbf{L}(h_t(\mathbf{x}_i), y_i, d_i), \tag{3.1}$$

where $\mathbf{L}(.)$ is the empirical loss of the base classifier $h_t$ on the labeled samples, and $d_i$ is the sample weight. Initially, AdaBoost often constructs an initial distribution of weights $D_1 = \{d_i | d_i = \frac{1}{n_l}, 1 \leq i \leq n_l\}$ over the training data. For each $t = 1$ to $T$ iterations, AdaBoost selects a base classifier that gives the least error

$$h'_t = \arg\min_{\mathcal{H}} \sum_{i=1}^{n_l} (h_t(\mathbf{x}_i) - y_i)^2 d_i. \tag{3.2}$$

Next, the weights associated with the samples misclassified by the selected weak classifier are increased. The weight distribution is updated at each iteration so that previously incorrect training samples are weighted higher. Specifically, the weight of the classifier is determined based on the corresponding training error:

$$\alpha_t = \frac{1}{2} \ln \frac{1 - \sum_i (h_t(\mathbf{x}_i) - y_i)^2 d_i}{\sum_i (h_t(\mathbf{x}_i) - y_i)^2 d_i}. \tag{3.3}$$

In the next iteration, the weights associated with the samples misclassified by the selected base classifier are increased as $d_i \leftarrow d_i e^{-\alpha_t y_i h_t(\mathbf{x}_i)}$. Finally, the strong classifier $\hat{f}$ is computed as the signum function of the weighted linear combination of $T$ base classifiers

$$\hat{f} = \text{sign}(\sum_{t=1}^{T} \alpha_t h_t). \tag{3.4}$$

### 3.1.2 Extension to Multi-class Problem

To extend the binary learning problem into $k$-class, the "one-versus-one" strategy, also known as pairwise coupling [35], is adapted for the boosting framework. Instead of using the positive samples from a source class and the rest negative samples from the rest of the source classes as suggested in [86], they train a set of base classifiers which distinguished the samples of one source class from those of another source class. This approach uses a binary learning algorithm to distinguish the samples of one class from, the samples of another class. For $k$ classes, a total of $\frac{k(k-1)}{2}$ binary classifiers are trained for all possible class pairs.

Let $P_{s_1}(s_2|\mathbf{x}_i)$ be the probability output for the binary boosting algorithm to classify class $s_1$ against class $s_2$ ($1 \leq s_1, s_2 \leq k$ and $s_1 \neq s_2$). From the composite classifier, the probability output can be directly computed using a sigmoid function:

$$P_{s_1}(\mathbf{x}) = \frac{1}{1 + \exp(-\hat{f}^{s_1})}. \tag{3.5}$$

Assume that all $P(.)$ are independent, the posterior probability for class $s_1$ can be computed as

$$P(y_i = s_1|\mathbf{x}) = \prod_k P_{s_1}(s_k|\mathbf{x}). \tag{3.6}$$

Subsequently, the class label of example $\mathbf{x}_i$ is predicted as:

$$\hat{y}_i = \arg\max_k P(y_i = k|\mathbf{x}_i). \tag{3.7}$$

After transferring the base classifiers, the boosted classifier $\hat{f}^\tau$ and the target label $y_\tau$ were derived similarly as equation (3.4) and (3.7), respectively. Dictated by the above equation, the total weight of positive and negative training samples are set to be equal; and the individual weights are inversely proportional to the number of training samples. After the additional training samples are obtained from a human expert, the weights distribution is adjusted accordingly. Note that the highest probability class $\hat{y}_1 = \arg\max_k P(k|\mathbf{x})$ and second highest probability class $\hat{y}_2 = \arg\max_{\hat{y}_2 \neq \hat{y}_1} P(k|\mathbf{x})$ will be used to compute the classification margin in the later chapters.

## 3.2 Transfer Learning

In this section, we aim apply the previously knowledge from the existing classes to build the classifier. Let the $s^{th}$ source class ($1 \leq s \leq k$ where $k$ is the number of source classes) contain the source data $\mathcal{D}^s = \{(\mathbf{x}_i, y_i)|1 \leq i \leq n^s\}$ where $n^s$ is the number of source training samples. The target class $\mathcal{T}$ has some training data $D^\tau = \{(\mathbf{x}_i, y_i)|1 \leq i \leq n^\tau\}$ where $n^\tau$ is the number of target training samples (note that $n^\tau \ll n^s$). Our main assumption is that labeled training samples from $s^{th}$ source classes are available and unlabeled data contains some target samples that can be used to improve the learning of the target classifier function $\hat{f}^\tau : \mathcal{X} \rightarrow \mathcal{Y}$.

### 3.2.1 Collecting Classification Rules from Existing Data

When the number of samples of the target class is small, previous research suggested to improve the classification by leveraging some transferable knowledge from the source classes [67, 31, 26]. By conducting the training under AdaBoost [29], we are able to collect all base classifiers $h_c$ from $f^s$ of $k$ source classes into set $\mathcal{H}_c = \{h_c(\mathbf{x}_i)|h_c(\mathbf{x}_i) \in f^s, 1 \leq c \leq C\}$ where $C = k \times T$ where $T$ is the number of boosting iterations. To ensure the quality of set $\mathcal{H}_c$, we extract only the classification rules that yielded a training error less than a threshold value. Additionally, we find a redundancy problem where some rules from multiple sources were practically identical. To this end, we sort the classification rules on their splitting attributes and then eliminated ones which had a negligible difference to others. Note that this extraction process could be done off-line and prior to the encounter of a new class.

### 3.2.2 Transferring Classification Rules to New Data

A recent study [86] suggested using boosting to extract $k$ sets of base classifiers using the positive samples from a source class and the negative samples from the rest of the source classes. Subsequently, all base classifiers are collected into a candidate set $\mathcal{H}_c$. as describe in the above section. As the result, $k$ binary boosted classifiers were trained, each classifier is from a source class.

The formation of the target boosted classifier is illustrated in Figure 3.1. The target training set consisted of positive samples from the $\tau$ and negative samples from $s$. Instead of training the base classifier directly from a small number of positive samples, the candidate set $\mathcal{H}_c$ was exploited and the target training set was used for evaluating a candidate base

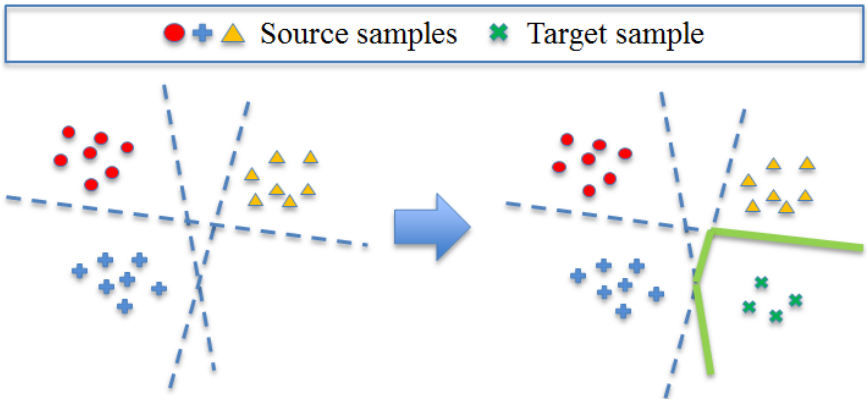



Figure 3.1: The formation of the boosted classifier of the target class with small number of target samples. Dashed blue lines represent a base classifier. Solid green line is the target classifier constructed by the base classifiers transferred from the source classes.

classifier. For each boosting iteration $t$, a base classifier $h'_t \in \mathcal{H}_c$ which minimizes the error over $D_\tau$ was chosen as:

$$h'_t = \arg\min_{\mathcal{H}_c} \sum_{i=1}^{n^\tau} (h_t(\mathbf{x}_i) - y_i)^2 d_i. \tag{3.8}$$

Interestingly, it was possible that a base classifier had the classification error over the target training data exceeds $50\%$, which is a violation of the boosting assumption [29]. This could have happened because the base classifiers were transferred from source classes instead of training using target samples. For instance, a base classifier captured a relationship that is opposite from the target class. In binary form, the base classifier had the opposite side of the decision boundary for positive class. In such case, the rule of that base classifier was inverted as $h^*_t \leftarrow -h_t$. As a result, the base classifier was adapted to the target data and the classification error consequently dropped below $50\%$. Similar to boosting, the computation of weight $\alpha_t$ is given by (3.3) and the update of weight distribution $D_t$ is same as (3.4).

## 3.3 Adding Regularizations

So far, the classifiers have been trained using only the empirical loss term as training error in (4.1). Since the objective is to learn a target classifier $h_t$ using only a small number of labeled samples, we can take advantage of the large number of unlabeled samples. The unlabeled samples are the image patches available abundantly in the video and required no manual labeling effort. In [12], the training of a classifier $h_t$ are presented as the following optimization:

$$\min_{h_t} \sum_{i=1}^{n_l} \mathbf{L}(h_t(\mathbf{x}_i), y_i) + \lambda \sum_{j=1}^{n_u} \mathbf{R}(h_t(\mathbf{x}_j)), \tag{3.9}$$

where the first term $\mathbf{L}(.)$ is the empirical loss of the target classifier $h_t$ on the labeled samples; and the second term $\mathbf{R}(.)$ is the regularization which constraints $h_t$ based on the unlabeled samples. The regularization parameter $\lambda$ controls the trade-off between the empirical loss and the regularization.

### 3.3.1 Determining the Weights of Existing Classifiers

To estimate the target classifier, a recent method in [12] proposes to construct a weighted combination of the $k$ source classifiers $f^s$. Let $\beta^s$ be the measure of relevancy of the $s^{th}$ source class on the target class, and $f_j^s = f^s(x_j)$ be the predicted value of the $s^{th}$ source classifier on $x_j$. The estimated label or pseudo-label $\hat{y}_j$ of the unlabeled sample $x_j$ based on the $k$ source classifiers is given by

$$\hat{y}_j = \sum_{s=1}^{k} (\beta^s f_j^s) = \mathbf{F}_j^s \mathbf{B}^s \tag{3.10}$$

where $\mathbf{F}_i^S = [f_j^1 \dots f_j^k]$ is the $1 \times k$ vector of predicted value of $k$ source classifiers for the $j^{th}$ target sample, and $\mathbf{B}^s = [\beta^1 \dots \beta^k]'$ is the $k \times 1$ weight vector, where $\beta^s$ is the weight corresponding to the $s^{th}$ source class.

Vector $\mathbf{B}^s$ can be estimated under the smoothness assumption that the predicted labels between any two "nearby" samples in the feature space should be similar, the $i^{th}$ and $j^{th}$ unlabeled samples are determined to be "nearby" according to their distance given by the edge weight $w_{i,j} \in \mathbf{W}_{i,j}$ as the distance in feature space. The weight vector $\mathbf{B}^s$ was computed by solving the following minimization problem

$$\begin{aligned} &\min_{\mathbf{B}^s} \sum_{i,j=1}^{n_u} (\mathbf{F}_i^s \mathbf{B}^s - \mathbf{F}_j^s \mathbf{B}^s)^2 \mathbf{W}_{i,j} \\ &\text{subject to } \sum_{s=1}^{k} \beta^s = 1 \text{ and } \beta^s > 0, \end{aligned} \tag{3.11}$$

where $\mathbf{F}_i^s \mathbf{B}^s$ and $\mathbf{F}_j^s \mathbf{B}^s$ are the pseudo-labels, and $w_{i,j} = e^{-\frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma}}$ as the distance in feature space of for $i^{th}$ and $j^{th}$ samples of the target class.

The optimization in (3.11) can be rewritten into a standard quadratic problem and be solved accordingly as presented in [12]. By enforcing that nearby points in the feature space should have similar labels, the weighting scheme of $\mathbf{B}^s$ is likely to give higher weights to the source classes which provide consistent labels to similar target samples. On the other hand, those source classes which provide inconsistent labels to similar target samples is likely to get a low weight. Furthermore, this minimization problem allows the evaluation of $\mathbf{B}^s$ for all $k$ source classes simultaneously, thus taking into account any potential interactions among multiple source classes.

### 3.3.2 Selecting Training Samples

A simple approach for selecting additional training samples is to uniformly sample the unlabeled data at random and then use the training samples which are labeled as the target class. However, in a classification problem with a large number of classes, the random sampling approach obtains only a small portion of the labeled samples as target training samples. Consequently, much labeling effort from the human annotator can be wasted.

**Margin Sampling** To help select more effective training samples, margin sampling has been frequently used in active learning literature [31, 30, 48]. In feature space, a margin of an example is defined as the distance of that example to the classifier's decision boundary. Intuitively, samples with large margins are easy to classify, since the classifier has little doubt in differentiating between the two most likely class labels. Conversely, samples with small margins are more confusing, therefore knowing the true label would help the classifier discriminate more effectively between them. An unlabeled example is said to have the smallest margin if the difference between its probabilities of belonging to two classes is minimal. Thus, we consider only the most and second most probable classes ignored the remaining ones. Consequently, we select the unlabeled example $\mathbf{x} \in D_u^\tau$ which has minimal difference between its highest probability class $\hat{y}_1 = \arg\max_k P(k|\mathbf{x})$ and second highest probability class $\hat{y}_2 = \arg\max_{\hat{y}_2 \neq \hat{y}_1} P(k|\mathbf{x})$. Following margin sampling principle, a new training sample is selected by the following function:

$$\arg\min_{\mathbf{x} \in D_u^\tau} \quad P(\hat{y}_1|\mathbf{x}) - P(\hat{y}_2|\mathbf{x}) \tag{3.12}$$

where $\hat{y}_1$ and $\hat{y}_2$ are the classes which return the highest and second highest posterior probability. It projects the input feature space $x_j$ of a unlabeled sample into a margin in probability space and select the minimum margin among the classes. v

## 3.4 Metrics for Classification Evaluation

The ground truth data consists all centroid locations of biological objects in the image. Depending on the object type, a biologist technician can collect from 25 to 100 centroids locations per an training image.

Let us consider a binary classification problem in which the outcomes are labeled either as positive or negative. There are four possible outcomes from a binary classifier. The object detected by an methods was determined as true positive (TP) if there was a corresponding ground truth location within the diameter of the object type; otherwise it is a false positive (FP). Any undetected object is considered a false negative (FP).

We compute the Recall (also called True Positive Rate) $TPR = \frac{TP}{TP+FN}$ and Precision (also called Positive Predictive Value) $PPV = \frac{TP}{TP+FP}$. To measure the overall performance, we used F-score as $2 \times \frac{Precision \times Recall}{Precision+Recall}$. By definition, a higher F-score value, which was ranged from 0 to 1, corresponded to a better detection performance. Our aim was to use a single metric such as F-score for the experiment to allow easy interpretation among the training methods yet ensure satisfactory assessment since F-score encompass both recall and precision values.

For some experiments, we provide $A_{ROC}$ as the area under the curve of the receiver operating characteristics (ROC) as the standard metric to determine the detection accuracy of all evaluating classifiers. The ROC curve is created by plotting the true positive rate

(TPR) against the false positive rate (FPR) at various threshold settings. By definition, True Positive Rate = $\frac{TP}{TP+FP}$ and False Positive Rate = $\frac{FP}{FP+TN}$ where $TP$ is the number of true positives, $TN$ is true negatives, and $FP$ is false positives.

# CHAPTER 4: CELL DETECTION WITH SIZE-DISTRIBUTION REGULARIZATION

The goal of this chapter is to reduce the number of samples required to train a classifier in the cell detection context. Learning-based cell detection tend to be specific to a particular imaging protocol and cell type. For a new cell type, a tedious re-training process must be performed by human experts. To reduce the amount of effort required by the experts to train a cell detector on the new cell type, previously collected information from the existing cell types can be exploited. Similar to the transfer learning algorithm described in Chapter 3, the cell detection in this chapter leverages useful information from existing rules of other cell types which have already been observed (Section 4.1). When the number of samples is very small, training is still susceptible to overfitting to those small number of samples. We address the overfitting issues by introducing a regularization for the new type to refine the ranking of the existing rules (Section 4.2). Since the Size-Differential (SDR) could effectively quantify the characteristic cells in an image, it is selected as a regularization for the cell detection task. For each cell type, the SDR is determined during the training step and does not need to be re-trained for each individual image. The evaluation on five cell types with 2,660 individual cells (50 real images) demonstrates that a cell detector trained using the proposed method is able to achieve good performance on the new cell type with only 10% of the training effort. We demonstrate that our method achieves the accuracy of previous approaches while reduces the training effort up to 10 times. Additionally, we evaluate the sensitivity of the SDR and show that only a few training samples are needed to

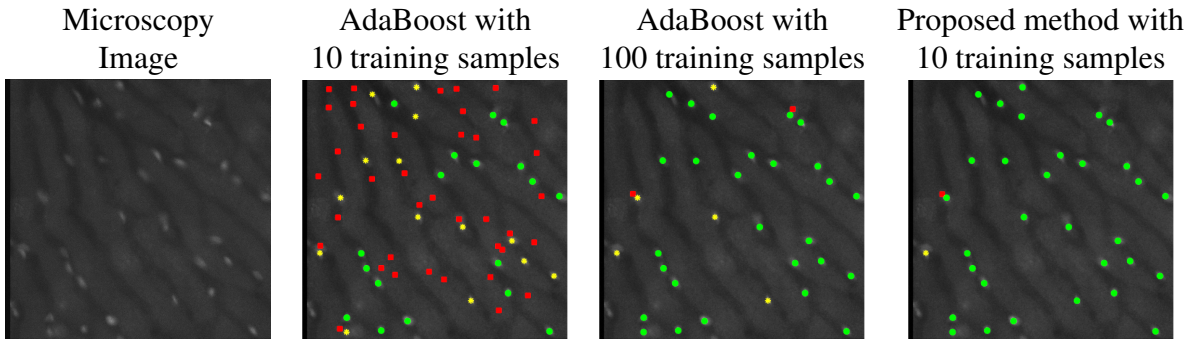



Figure 4.1: Sample detection results from a cell detector trained by a traditional machine learning method (AdaBoost) compared to one trained by the proposed method. With 10 training samples, AdaBoost yields several incorrectly detected regions (denoted as red squares) and undetected cells (denoted as yellow asterisks). In order to detect most cells correctly (denoted as green circles), AdaBoost requires up to 100 training samples. Our method only needs 10 training samples to achieve a similar performance.

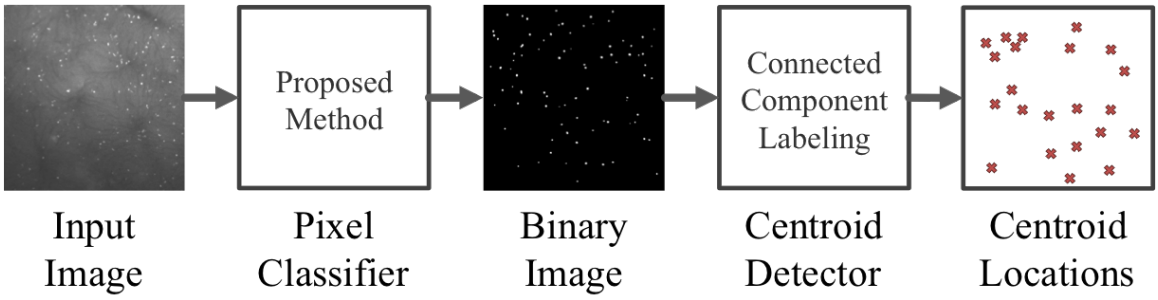



Figure 4.2: The overview of the cell detection process. Given an input cell image, the cell pixel classifier using the proposed method returns a binary image of cell pixels. Then, a cell centroid detector groups cell pixels into cell regions (using connected component labeling) and computes the location of cell centroids.

reliably estimate the SDR in our dataset. The proposed work and its applications presented in this section were published in[54, 56]. An example of detection results is shown in Figure 5.2.

## 4.1 Overview

A classification model in this chapter is described as a binary pixel classifier in the the context of cell detection. In this context, the first step is to detect cell from a given grayscale

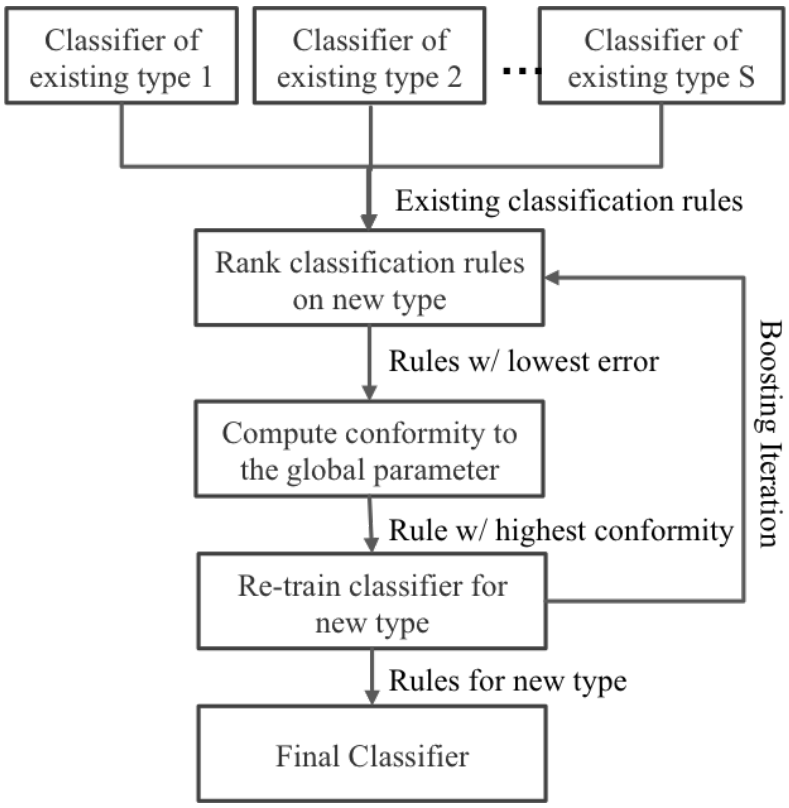



Figure 4.3: The overview of the proposed cell pixel classification method.

image where each pixel is classified as either cell or non-cell. This is the main part of our method and such step has been optimized using the proposed method. The next step is to take the binary image and group them into cell regions. This is simply done by performing connected component labeling [66]. All evaluations are conducted at the cell region level by comparing the distance to the center of cell. The overview of the cell detection process can be found in Figure 4.2.

In the feature space $\mathcal{X}$ of pixels in a microscopy image, and the label space $\mathcal{Y} = \{-1, +1\}$ denoting each pixel in the image as cell or non-cell, training a cell pixel classifier is the same as estimating a function $f : \mathcal{X} \rightarrow \mathcal{Y}$. In order to train the cell pixel classifier, multiple training samples consist of the feature vector $\mathbf{x}_i \in \mathcal{X}$ and the label

$y_i \in \mathcal{Y}$ are collected from a user expert by clicking on a training image. A cell sample (labeled as $y_i = 1$) has the corresponding feature vector $\mathbf{x}_i$ computed at the centroid pixel of a cell.

**Leveraging Classification Rules on a New Type** For a new (or target) cell type $\tau$, training the pixel classifier without a sufficient number of samples could often lead to poor performance [70]. Rather, we leveraged the classification rules extracted in $\mathcal{H}_c$ to train the cell detector similarly to a recent transfer learning algorithm, TaskTrAdaBoost [86]. In such way, the classification rules from $\mathcal{H}_c$ were employed to build the final pixel classifier $f_\tau$ for the new cell type given training data $D_\tau = \{(\mathbf{x}_j, y_j) | 1 \leq j \leq n_\tau\}$ where $n_\tau$ is the number of target training samples (note that $n_\tau \ll n_s$). The target weight distribution is initialized as $\mathbf{w}_\tau = \{w_j | w_j = \frac{1}{2n_\tau}, 1 \leq j \leq n_\tau\}$. For each iteration, we selected a transferable rule $h_\beta$ that minimizes the error over $D_\tau$ as:

$$h_\beta(\mathbf{x}_j) = \arg\min_{h_c \in \mathcal{H}_c} (\sum_j w_j [y_j \neq h_c(\mathbf{x}_j)]). \tag{4.1}$$

**Overfitting Issues** The overfitting problem was previously referred as having a set of rules that is too specific to a particular dataset thus not having a diverse enough of classification rules [86]. Such problem was addressed in TaskTrAdaBoost by using multiple sources. With a sufficient number of samples, training would certainly be improved by having a more diverse thus relevant rules available. However, when the number of samples is very small, training is susceptible to overfitting to those small number of samples. Training step involves the selection of the most valuable classification rules by assessing the training samples. We attempt to improve such assessment by using a new criterion (regularization). Additionally, this problem is not specific to the cell detection; and our experiments also in-

clude the multi-source framework. In the next section, we proposed to use a regularization to refine the ranking of classification rules.

## 4.2 Size-Differential Regularization

In addition to minimizing the error on the training samples of the new cell type as in Equation (4.1), the transferable rules $h_\beta$ were selected to conform to a regularization. We particularly chose the Size-Differential as a regularization (SDR) because of the following reasons:

- It represents a common biological characteristic of cells.
- It can be quickly estimated from a training image.
- It requires minimal effort to measure manually.

In order to compute the feature vector of a cell training sample, the centroid location of a cell is collected by a single mouse click performed by a biology technician on an image. Under the assumption that the cells were approximately circular, measuring the cell size needed only one additional mouse click on the cell boundary after collecting the centroid location. We acknowledged that some cells were elliptical and measuring the exact cell size in such case might require two clicks for the major and minor axes. Alternatively, other regularizations such as shape and color features could be integrated into our framework depending on the application domain. Under *in vivo* microscopy, the florescent intensities within the cells were different depending on the location of a cell with respect to the microscope focal lens resulting in various cell sizes [50]. We modeled the cell sizes using a Gaussian distribution. Figure 4.4 explains the selection of a classification rule based on the

cell size distribution on the training image.

Let us define a distribution $\mathcal{P} \sim \mathcal{N}(\mu_m, \sigma_m^2)$ to model the cell size with $\mu_m = \frac{1}{M}\sum_m(r_m)$, and $\sigma_m = \sqrt{\frac{1}{M}\sum_m(r_m - \mu_m)^2}$ where $r_m$ is the cell radius. In other words, the cell size distribution $\mathcal{P}$ was estimated using set $\mathcal{R} = \{r_m | 1 \leq m \leq M\}$ where $M$ is the number of size samples. On a training image $\mathcal{I}_\tau$ of the new cell type, a user measured the cell radius $r_m$ by one additional mouse click on the cell boundary after getting the cell location. Thus, acquiring $M$ samples of cell sizes required only additional $M$ mouse clicks. In Section 4.3.3, we showed that the estimation of $\mathcal{P}$ was robust enough to maintain a stable performance with $M = 6$.

After acquiring $\mathcal{P}$, we collect every classification rule $h_c$ and apply each rule to classify the training image $\mathcal{I}_\tau$ to obtain a binary classification image containing cell and non-cell pixels. Neighboring cell pixels were grouped into $U$ cell regions using the connected component labeling procedure [66]. This procedure uniquely labels a group of pixels based on its connectivity with the neighboring pixels. Assumed that cells were approximately circular, cell radius $r_u$ were automatically derived from each cell region and formed a set $\mathcal{R}_c = \{r_u | 1 \leq u \leq U\}$. To ensure the quality of $\mathcal{R}_c$, we filtered out the outlier regions which are smaller than a threshold value. Subsequently, we used set $\mathcal{R}_c$ to compute the detected cell size distribution $\mathcal{Q}_c$ as a discrete probability distribution where the values of cell sizes in $\mathcal{R}_c$ is equally spaced and contained within a discrete bin.

On training image $\mathcal{I}_\tau$, we evaluated the conformity of classification rule $h_c$, which resulted in the detected size distribution $\mathcal{Q}_c$, to the user estimated cell size distribution $\mathcal{P}$ using the Kullback-Leibler divergence [43]. In particular, we selected the classification

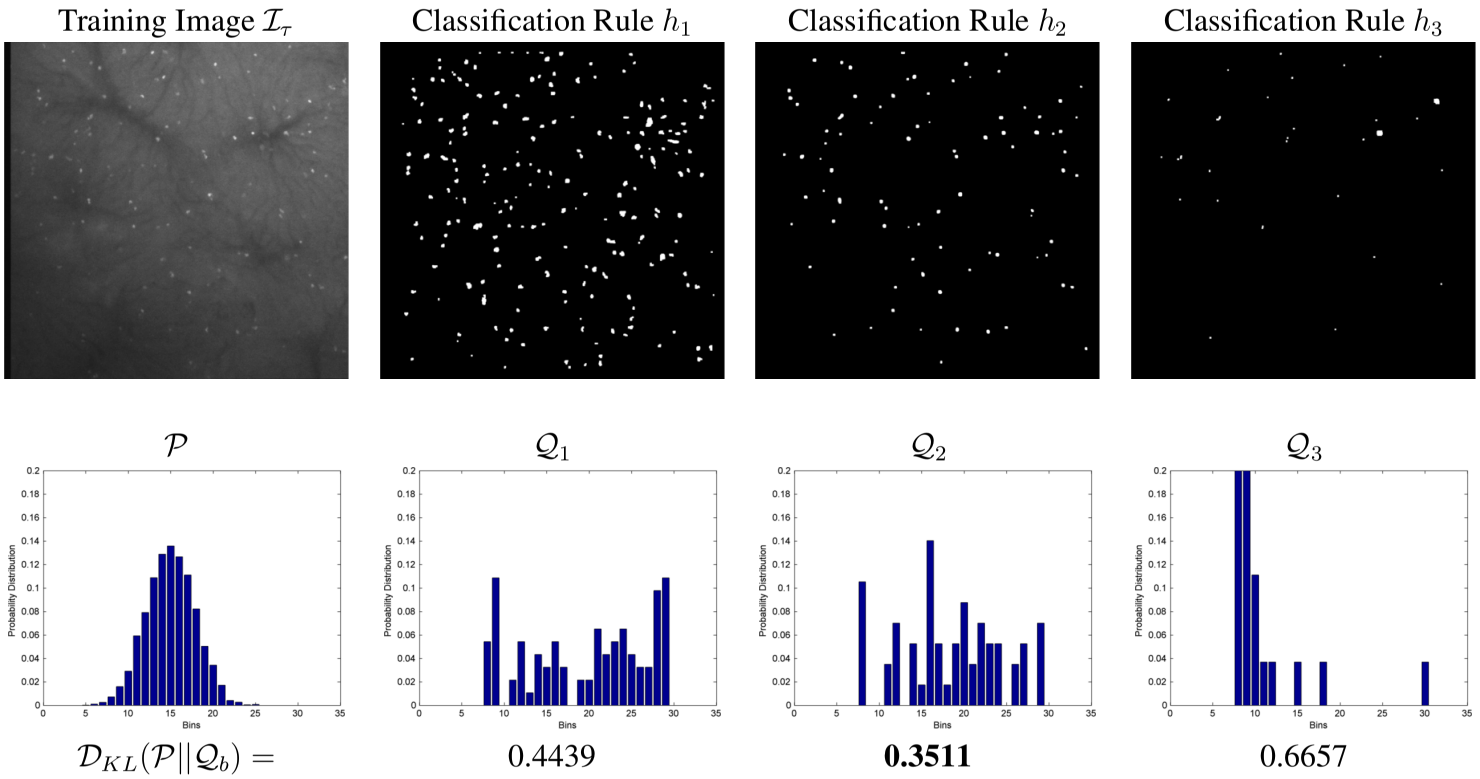



Figure 4.4: The selection a classification rule out of 3 different ones with equal training error on 4 training samples. In the first row, classification rule $h_2$ is able to returns high accuracy on the training image than other rules ($h_1$ yields too many incorrect detected cells, and $h_3$ leaves out most cell undetected). As a result, when compared with the user defined distribution $\mathcal{P}$, the corresponding size distributions of $h_1$ and $h_3$ have larger Kullback-Liebler divergence than that of $h_2$. Classification rule $h_2$, which yields the lowest divergence (in bold), is selected to construct the cell detector.

rule $h^*_\beta$ as:

$$h^*_\beta(\mathbf{x}_j) = \arg\min_{h_c \in \mathcal{H}_c}((\sum_j w_j[y_j \neq h_c(\mathbf{x}_j)]) + \lambda \mathcal{D}_{KL}(\mathcal{P}||\mathcal{Q}_c)) \tag{4.2}$$

where $\mathcal{D}_{KL}(\mathcal{P}||\mathcal{Q}_c)$ is the Kullback-Leibler divergence between distribution $\mathcal{P}$ and $\mathcal{Q}_c$:

$$\mathcal{D}_{KL}(\mathcal{P}||\mathcal{Q}_c) = \sum_p \mathcal{P}(p) \log \frac{\mathcal{P}(p)}{\mathcal{Q}_c(p)} \tag{4.3}$$

and $p$ is a bin containing the equally-spaced values of possible cell sizes. As the result, the classification rule $h^*_\beta$ conformed with the cell size distribution $\mathcal{P}$ besides minimizing the target training error.

Following the boosting scheme of $T$ classification rules $h^*_\beta(\mathbf{x}_j)$, the detector of the new cell type $f_\tau$ was constructed as:

$$f_\tau = \text{sign}(\sum_{\beta=1}^{T} \alpha_\beta h^*_\beta(\mathbf{x}_j)) \tag{4.4}$$

where the boosting coefficient $\alpha_\beta$ is computed similarly as in [86]. In our dataset, cells do not often overlap with others and the paper focuses on reducing the number of training samples. However, we noted that there were several cases where some cells were in a cluster and *touch* each other. In such cases, rather than using connected component labeling, the grouping of cell pixels can be coupled with methods such as [1, 22, 87]. Thus, our method of improving training efficiency could be integrated with other works on handling occlusion and cell clusters.

**Implementation Details** The proposed method, which was implemented in Matlab R2012a on an Intel Core 2 Duo 2.66GHz workstation, ran at an average speed of 1.8 seconds when detecting approximately 25 to 100 cells in an $1000 \times 1000$ pixel image.

Table 4.1: The description of the experimental datasets

| ID | Cell Type | Imaging Protocol | Mag. | Source |
|---|---|---|---|---|
| NKT | Natural Killer T | *in-vivo* Intravital | 10X | [75] |
| WBC | White Blood Cell | *in-vivo* Intravital | 20X | [53] |
| RBC | Red Blood Cell | *in-vivo* Intravital | 40X | [40] |
| DSP | Drosophila | Fluorescence Light | 40X | [10] |
| HTC | HT29 Colon Cancer | Fluorescence Light | 40X | [10] |

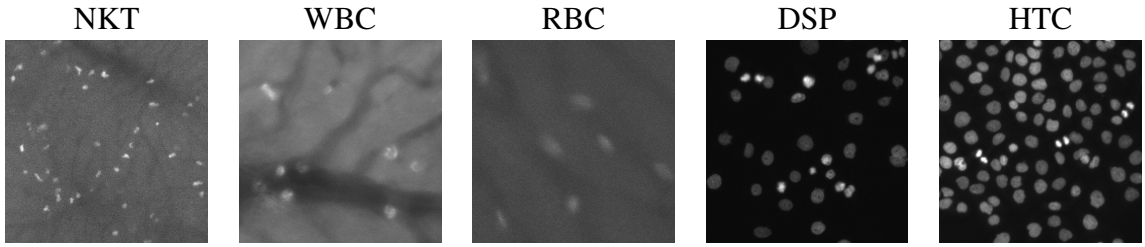



Figure 4.5: Representative images with different appearances of five cell types that are captured from various sources.

## 4.3 Experiments on Cell Detection

**Data Description:** Five different cell types including white blood cells (WBC), natural killer T-cells (NKT), red blood cells (RBC), drosophila (DSP), and HT29 colon cancer (HTC) were acquired using 2 imaging protocols (*in vivo* epi-fluorescence and isolated fluorescently labeled) and 3 magnification levels (10X, 20X, and 40X). We evaluated the performance of the detection algorithms on a total of 50 real images (10 images from each cell type). Each image might contain from 25 to 100 cells (total of 2660 cells). We divided the images from each cell type into two halves for training and evaluating. A biology technician manually determined the center and the radius of the cells in 50 images. For each cell type in our experiment, we only selected training samples from a set of 5 images and evaluate on another set of 5 images to ensure there is no possible overlap between the training and testing data. The details of each cell types were described in Table 4.1 and

sample images were shown in Figure 4.5.

**Compared Methods:** To evaluate the performance, we consider two components of the proposed method: the Size-Differential Regularization integration and the transfer learning strategy. The CSD was proposed as a Size-Differential Regularization to be integrated with the transfer learning framework (as described in Section 4.2) to form the GlobalTrAdaBoost algorithm. This proposed method is compared with two existing boosting methods that were used to train the cell detector. First, AdaBoost algorithm [29] was considered as the baseline method. Second, TaskTrAdaBoost [86] was employed to transfer knowledge from previous data to the training of new data. We showed some detection results of the proposed method compared to those of AdaBoost, and TaskTrAdaBoost in Figure 4.6.

In this section, we examined the effect of the Size-Differential Regularization on the performance of AdaBoost, TaskTrAdaBoost, and GlobalTrAdaBoost. Additionally, we discussed the sensitivity of the cell size distribution and explained how our approach was able to estimate the distribution with just a few samples. To objectively evaluate the detection performance, we replicated the training and testing of each method 30 times on each of 5 cell types. In each evaluation, one cell type was chosen as the new type, and the remaining cell types were used as the existing types.

### 4.3.1 The Effect of the Size-Differential Regularization

To evaluate the effect of the Size-Differential Regularization on the detection accuracy, the F-measures of GlobalTrAdaBoost are compared against those of AdaBoost and TaskTrAdaBoost. We measure the training effort $E_\tau$ as the number of new training samples and the number of size samples required to obtain the global regularizer ($E_\tau = n_\tau + M$ where

Table 4.2: F-Measures (Mean $\pm$ Standard Error) of training methods for a cell detector with different training efforts. A performance number is highlighted in bold if it is significantly better than other methods based on a paired t-test at $p = 0.05$. $Global_{under}$ and $Global_{over}$ (as discussed in Section 4.3.3) are two versions of GlobalTrAdaboost with fluctuated values of the Size-Differential Regularization.

| $E_\tau$ | AdaBoost | TaskTrAdaBoost | GlobalTrAdaBoost | $Global_{under}$ | $Global_{over}$ |
|---|---|---|---|---|---|
| 10 | $0.57 \pm 0.024$ | $0.69 \pm 0.018$ | $\mathbf{0.81 \pm 0.014}$ | $0.78 \pm 0.032$ | $0.79 \pm 0.038$ |
| 20 | $0.68 \pm 0.019$ | $0.72 \pm 0.018$ | $\mathbf{0.82 \pm 0.011}$ | $0.79 \pm 0.039$ | $0.80 \pm 0.042$ |
| 30 | $0.69 \pm 0.018$ | $0.75 \pm 0.015$ | $\mathbf{0.81 \pm 0.011}$ | $0.79 \pm 0.031$ | $0.80 \pm 0.012$ |
| 40 | $0.76 \pm 0.013$ | $0.78 \pm 0.012$ | $\mathbf{0.81 \pm 0.008}$ | $0.81 \pm 0.034$ | $0.81 \pm 0.008$ |
| 50 | $0.76 \pm 0.012$ | $0.77 \pm 0.012$ | $\mathbf{0.82 \pm 0.006}$ | $0.82 \pm 0.023$ | $0.82 \pm 0.023$ |
| 60 | $0.79 \pm 0.010$ | $0.80 \pm 0.010$ | $0.83 \pm 0.008$ | $0.82 \pm 0.008$ | $0.81 \pm 0.043$ |
| 70 | $0.80 \pm 0.009$ | $0.82 \pm 0.007$ | $0.83 \pm 0.004$ | $0.82 \pm 0.007$ | $0.82 \pm 0.014$ |
| 80 | $0.80 \pm 0.012$ | $0.81 \pm 0.010$ | $0.83 \pm 0.006$ | $0.83 \pm 0.008$ | $0.82 \pm 0.011$ |
| 90 | $0.80 \pm 0.012$ | $0.81 \pm 0.009$ | $\mathbf{0.83 \pm 0.004}$ | $0.83 \pm 0.009$ | $0.83 \pm 0.010$ |
| 100 | $0.82 \pm 0.009$ | $0.83 \pm 0.007$ | $0.84 \pm 0.004$ | $0.83 \pm 0.008$ | $0.83 \pm 0.010$ |

$M = 6$ for GlobalTrAdaBoost and $M = 0$ for AdaBoost and TaskTrAdaBoost). Training is conducted with $E_\tau$ varying from 10 to 100. In each execution, the target training samples $D_\tau$ are randomly selected. To keep the training set balanced, an equal number of non-cell samples are also randomly selected for all methods.

The performance of the final detector would be dependent on the training data as shown in Table 4.2; and the maximum attainable performance would be the performance with the maximum number of training samples. When trained with a large number of samples ($E_\tau = 100$), all AdaBoost, TaskTrAdaBoost, and GlobalTrAdaBoost expectedly reach similar F-measures of 0.82, 0.83, and 0.84, respectively). Theoretically, the performance of all compared methods should approach similar results with large amount of training data. The focus of the our experiment is not on the improving the final detection performance but rather reducing the training effort. Thus, we focus on comparing how the performance *improves* especially when the training size is small. However, GlobalTrAdaBoost show significant improvement in accuracy as well as stability to other methods with small training efforts. In particular, at training effort $E_\tau = 10$, 20, and 30, integrating the size distribution (as described in Section 4.2) improves accuracy as the average improvement over TaskTrAdaBoost is 17%, 14%, and 8%, respectively (see Table 4.2). The stability of the GlobalTrAdaBoost is also improved as the standard error reduces from TaskTrAdaBoost by 22%, 39%, and 27%. Evidently, the CSD regularizes the detection by refining the ranking of classification rules when only a few training samples were available.

Compared to AdaBoost, GlobalTrAdaBoost reduces the training effort up to 10 times ($E_\tau = 10$ versus 100) to achieve equal performance. When AdaBoost encounters a new cell type which has a different size from what had been seen before, providing few train-

Table 4.3: Comparison of Cell Size Distributions (Mean $\pm$ Standard Deviation in pixels) constructed using 6 selected samples and all available samples. $\Delta_\mu$ is the difference of the means and $\Delta_\sigma$ is the difference of the standard deviations. Note that the differences across the cell types are quite small (less than 1 pixel).

| Data | All Samples | 6 Samples | $\Delta_\mu$ | $\Delta_\sigma$ |
|---|---|---|---|---|
| NKT | $4.95 \pm 1.40$ | $4.73 \pm 1.48$ | 0.22 | 0.08 |
| WBC | $9.34 \pm 2.15$ | $8.30 \pm 2.02$ | 1.04 | 0.13 |
| RBC | $12.17 \pm 2.75$ | $12.79 \pm 2.79$ | 0.62 | 0.04 |
| DSP | $6.86 \pm 1.17$ | $7.35 \pm 0.81$ | 0.49 | 0.36 |
| HTC | $5.26 \pm 1.63$ | $5.51 \pm 1.91$ | 0.25 | 0.28 |

ing samples from the new cell type usually yields poor performance (F-measure$= 0.57$ when $E_\tau = 10$). However, with the same number of training samples, GlobalTrAdaBoost rapidly adapts to the new cell type by regularizing the classification rules from existing types to conform with CSD (as described in Section 4.2). In fact, with just 10 training samples, GlobalTrAdaBoost's F-measure was already 0.81, which was only 4% lower than its maximum performance.

### 4.3.2 The Estimation of the Size-Differential Regularization

The estimation of the Size-Differential Regularization played an important role in the detection performance when only a few training samples were available. The cell size distribution was estimated using a Gaussian model constructed by a few manually collected samples. Per each cell type, 6 cell samples which were most frequently selected upon the user's inputs were used to construct the *estimated* cell size distribution. In each set, we computed the mean and standard deviation of cell sizes per each value of $M$. We observed that the mean of the standard deviations of multiple sets start converging when $M \geq 6$. We compared this distribution with the *true* cell size distribution which were constructed using all available cell samples in a dataset.

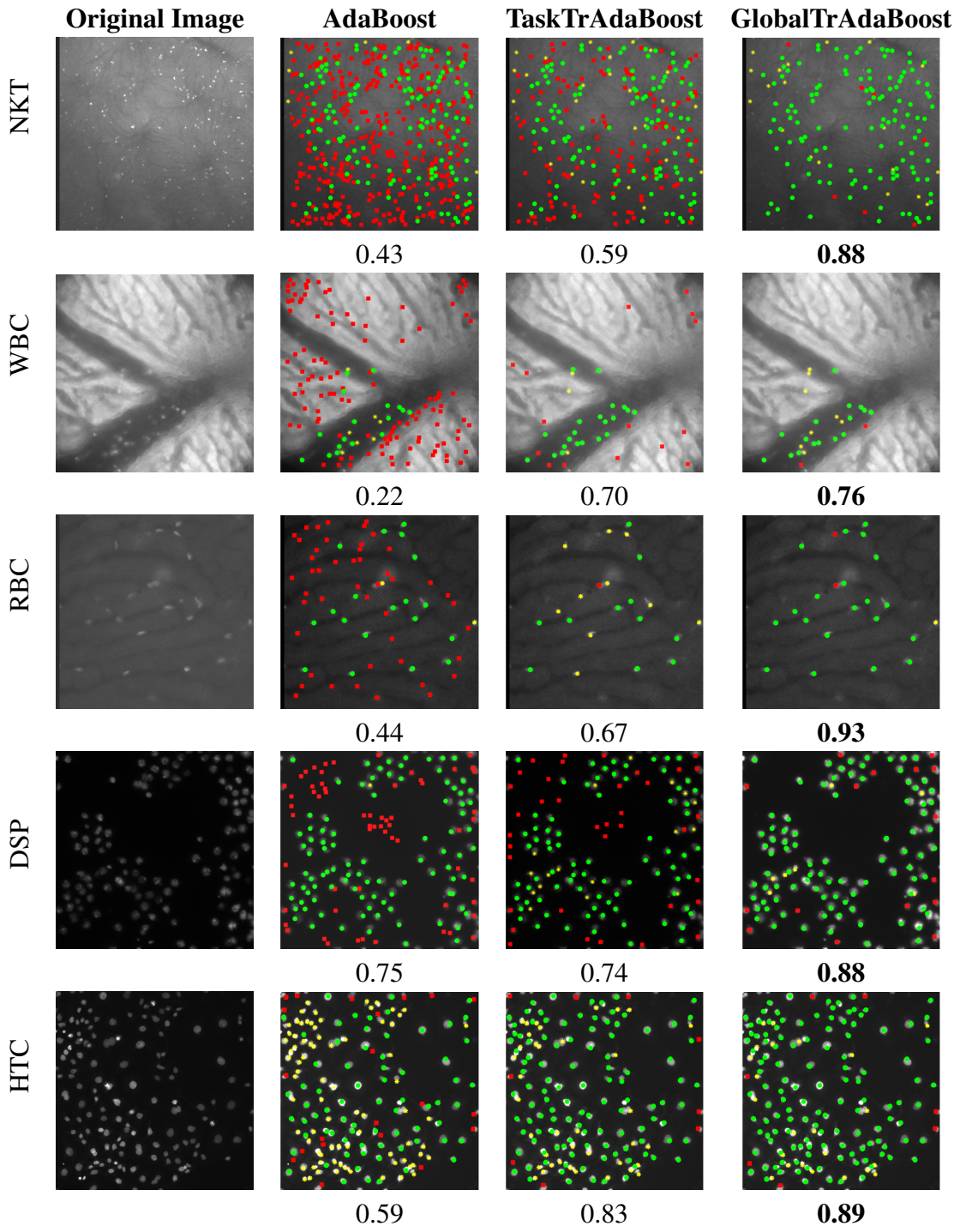



Figure 4.6: Sample results from five different cell type comparing AdaBoost, TaskTrAdaBoost and the proposed method when trained with only 10 training samples. The number under each image is the corresponding detection performance, in terms of F-measure. When provided with as few as 10 training samples, both AdaBoost and TaskTrAdaBoost provide many false positives (red squares) and false negatives (yellow asterisks). GlobalTrAdaBoost is able to achieve high numbers of true positives (green circles) using the same amount of training effort.

Table 4.3 showed the cell size distributions with mean and standard deviation in pixel values. It was notable that the standard deviations of the size distribution in WBC (at 2.02 pixels) and RBC (at 2.79 pixels) were slightly higher than other cell types because these types had more cells that are appeared to be elliptical which made them more difficult to be accurately measured under the circular assumption. This also explained why the mean difference in WBC and RBC were higher than other types at 1.04 and 0.62 pixels respectively. However, the differences between the two distributions averaged across all cell types yielded as low as 0.45 pixel in mean and 0.18 pixel in standard deviation implied that 6 selected samples were reliable enough to estimate the cell size distribution. Additionally, the experiments show that CSD works well for all of the cell types in our dataset.

### 4.3.3 The Sensitivity of the Size-distribution Regularization

The estimation of the Size-distribution Regularization plays an important role in the detection performance when only a few training samples were available. The cell size distribution is estimated using a Gaussian model constructed by a few manually collected samples. To evaluate the detection performance against the range of cell sizes, we randomly select 30 sets of $M$, which is up to 16 samples, from each cell type. In each set, we compute the mean and standard deviation of cell sizes per each value of $M$. As described in Section 4.3.3, the mean of the standard deviations of multiple sets starts converging when $M \geq 6$. Thereafter, we compute the mean $\mu_{\Delta}$ and standard deviation $\sigma_{\Delta}$ of multiple sets of 6 samples. If the GlobalTrAdaBoost performance is still higher than other methods when the Size-distribution Regularization estimation varied by the value of $\sigma_{\Delta}$, then we could infer the Size-distribution Regularization of 6 cell size samples ($M = 6$) are

sufficient to improve the detection performance. We integrate the GlobalTrAdaBoost with two fluctuated values of the Size-distribution Regularization $\mathcal{P}_{under} \sim \mathcal{N}(\mu_\Delta - \sigma_\Delta, \sigma_\Delta^2)$, and $\mathcal{P}_{over} \sim \mathcal{N}(\mu_\Delta + \sigma_\Delta, \sigma_\Delta^2)$. The corresponding methods $Global_{under}$ and $Global_{over}$ are executed 30 times in the same procedure described in Section 4.3.1. We show the F-measures in conjunction with other detection methods in Table 4.2. The performance of both $Global_{under}$ and $Global_{over}$ at each $E_\tau$ are significantly better ($p < 0.05$) than both AdaBoost and TaskTrAdaBoost up to $E_\tau = 60$.

## 4.4 Summary

To reduce the amount of effort required by the experts to train a cell detector on the new cell type, previously collected information from the existing cell types can be exploited. The classification rules extracted from existing cell types are combined with the training samples of the new cell type using transfer learning. However, when the number of samples is very small, training is still susceptible to overfitting to those small number of samples. We address the overfitting issues by introducing a regularization for the new cell type to refine the ranking of the existing rules. We particularly choose the Size-Differential (SDR) as a regularization because it could effectively quantify the characteristic cells in an image. For each cell type, the SDR is determined during the training step and does not need to be re-trained for each individual image. The estimation of the SDR played an important role in the detection performance when only a few training samples were available. Besides SDR, the proposed framework can potentially employ other regularizations such as cell shape variance and area of cell cluster. One direction of future research can be investigating on how these regularizers would be used to reduce the number of training samples. We

believe that these results demonstrated the potential of the proposed method for greater applicability in cell detection by reducing the amount of human effort.

# CHAPTER 5: SOCIAL INSECT DETECTION WITH SPATIO-TEMPORAL REGULARIZATION

## 5.1 Overview

A network formed by social insects (ants, bees, and termites) is of significant interest to biologists to understand the division of labor and task specialization, collective search and retrieval, adaptive networks, and other types of distributed problem-solving [72]. The studies of such social network require the analysis of movements and interactions over a long period of time. Traditionally, the detection process has been done by manually observing many insects in a colony. For example, two colonies containing dozens of individual insects are illustrated in Figure 5.1. Since this manual process is extremely time-consuming, it is not feasible for handling a large amount of data and limits the research progress in this field. Therefore, recently there have been many attempts to automate the detection for social insects.

In this chapter, we propose a transfer learning method that employs the spatiotemporal relationship among the unlabeled data. The spatiotemporal consists of the pixel coordinate as well as the time frame in the video from which an image patch is obtained. The unlabeled samples are the image patches available abundantly in the video and required no manual labeling effort. We estimate the spatiotemporal connectivity between pairs of unlabeled samples using the optical flow throughout the video. Our approach is based on the assumption that two unlabeled samples which are connected by the optical flow should

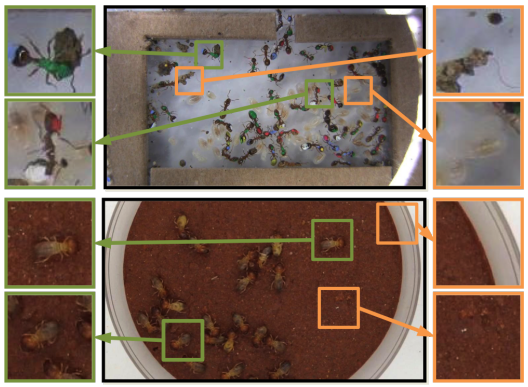


Figure 5.1: Typical images of ants and termites in colony contain at least dozens of individual objects within a controlled laboratory environment. A few positive samples of ants and termites (in green) as well as negative samples of dirt, eggs, container, and noise (in orange) are zoomed in to show greater details.

have the same predicted label. This motivates us to minimize the difference in predicted labels between two samples which are spatiotemporally connected. After that, we construct a graph to learn the weighting scheme of the existing classifiers from multiple sources to supplement the small number of labeled samples. The evaluation on three data sets of social insects such as ants and termites with over 6,000 samples demonstrates that the proposed method is able to reduce the training effort up to 16 times while maintaining comparable accuracy to previous approaches. An example of detection results is shown in Figure 5.2.

## 5.2 SpatioTemporally Regularized Adaptive Learning

In this paper, we investigate on the regularization term and formulate the proposed method in three parts. First, we provide some background on determining the relevancy of the source classifiers (Section 3.3.1). Next, we propose to weight the unlabeled samples based on the spatiotemporal connectivity (Section 5.2.1). Finally, we train the target classifier in a boosting framework (Section 5.2.2).

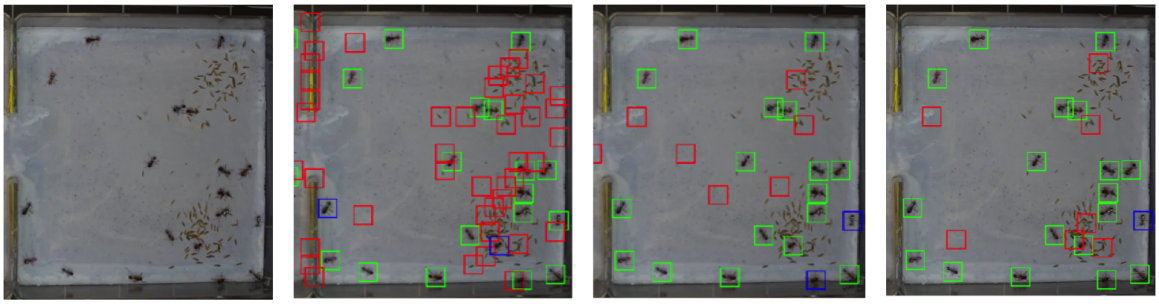
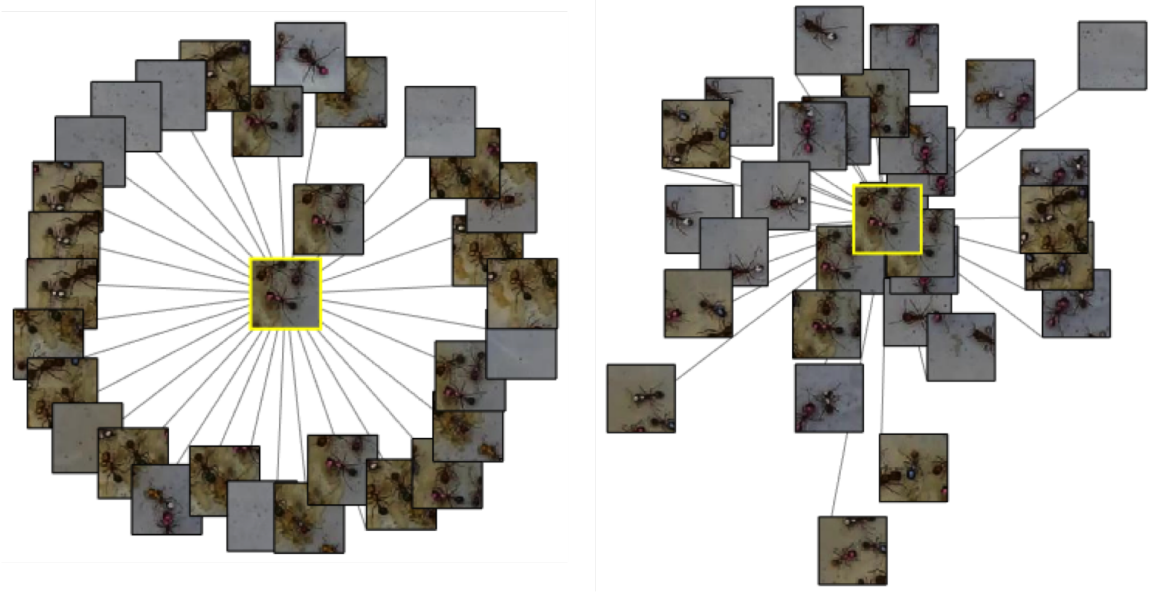


**Image of an ant colony** | **TaskTrAdaBoost with 4 training labels** | **TaskTrAdaBoost with 64 training labels** | **Proposed with 4 training labels**

Figure 5.2: Detection results from a classifier trained by a transfer learning algorithm (TaskTrAdaboost) compared to one trained by the proposed method. With 4 training samples, TaskTrAdaBoost yields many false positives (denoted as red squares) and false negatives (denoted as blue squares). In order to detect most object correctly (denoted as green squares), TaskTrAdaBoost requires up to 64 training samples. Our method only needs 4 training samples to achieve a comparable accuracy.

### 5.2.1 Spatiotemporal Regularization

Matrix $\mathbf{W}_{i,j}$ was computed in Equation (3.11) using only the distance in feature space in the previous section. In this section, we propose a new way to compute $\mathbf{W}_{i,j}$ using a spatiotemporal regularization which is based on optical flow. The rationale behind this approach is based on the assumption that two image patches which are connected in image coordinates at some time $t$ in the video should be the same predicted label. This motivates us to minimize the difference in predicted labels between two samples which are spatiotemporally connected. Thus, the weight $w_{i,j}$ between spatiotemporal connected samples should be high because they are likely to be the same labels. Formally, in an image at the temporal frame $t$ from a video, an optical flow vector was computed at every pixel location in the image. Subsequently, all of the optical flow vectors within a non-overlapping image area, called block $b_t$, was summed up into a single vector $\mathbf{v}_t$ of the block in the image at time $t$. By summing up with vectors, each block now consists of a single optical flow vector which



**(a) Connectivity using feature space**

**(b) Connectivity using spatiotemporal graph**

Figure 5.3: Visualization of the distance between a sample to its nearest neighbors in feature space (left) and in the spatiotemporal graph (right). The neighbors on the left figure have mixture of negative samples while almost all neighbor on the right figure are positive.

approximates the general orientation and magnitude of the motion of the objects contained inside the block.

After collecting these blocks, we construct a graph structure $\mathbf{G}_{ST} = \langle \mathbf{V}_{ST}, \mathbf{E}_{ST} \rangle$ consists of a set of nodes and each node $n_t \in \mathbf{V}_{ST}$ representing block $b_t$ and contains the block's spatial coordination $(c_{b_t}, r_{b_t}, t)$ where $c_{b_t}$ and $r_{b_t}$ are the column and the row of block $b_t$ in an image, and $t$ is the temporal frame. Together with the nodes in $\mathbf{G}_{ST}$ is a set of edges $e^t_{t+1} \in \mathbf{E}_{ST}$ connecting node $n_t$ and one of its 9-connected temporal neighbors node $n_{t+1}$. The neighbor node $n_{t+1}$ is determined based on the orientation and magnitude of vector $\mathbf{v}_t$ and the edge is associated with a numeric value indicating the cost to connect. For example, if node $n_t$ has the vector magnitude less than a motion threshold which indicates that little movement has been reported, it will be connected to block $n_{t+1}$ which has

the same position in the next frame. On the other hand, if node has a magnitude over a threshold value, we then determine to which of the 9-connected neighbors it will be connected using the orientation of its $\mathbf{v}_t$. As a result, we are able to construct a connected graph representing the relationship among all blocks in an image sequence.

From $\mathbf{G}_{ST}$, we build the weight matrix $\mathbf{W}_{i,j}$ which contains weights between unlabeled samples. Given any arbitrary pair of unlabeled samples $\mathbf{x}_i, \mathbf{x}_j \in D_u^\tau$, we can using $(c_i, r_i, t_i)$ and $(c_j, r_j, t_j)$ where $c, r$ is the image coordinate at time $t$ to map them directly to the corresponding nodes $n_i$ and $n_j \in \mathbf{V}_{ST}$. Subsequently, we calculate weight $w_{i,j} \in \mathbf{W}_{i,j}$ between two nodes $n_i$ to $n_j$ as the shortest path between them using Dijkstra's algorithm as follows

$$w_{i,j} = \mathbf{shortestPath}_{Dijkstra's}(n_i, n_j) \tag{5.1}$$

Using an adjacency matrix, the time complexity to add an additional node is $\mathcal{O}(|\mathbf{V}_{ST}|^2)$ where $|\mathbf{V}_{ST}|$ is the total number of nodes. While the total number of nodes is increased over time, its maximum value is the product between the total number of images in the video the total number of block per image. The number of blocks per image is carefully chosen to keep balance between the resolution of optical flow vectors and the computational expense when constructing graph $\mathbf{G}_{ST}$. Additionally, we recognize that it is sometime challenging to determine the neighbor node based solely on the optical flow vector, so we consider alternative possibilities by creating extra edges to all of the other neighbor nodes where each edge is assigned a higher associated cost. Figure 5.3 illustrates the effectiveness of sample comparison using $\mathbf{G}_{ST}$ instead of distance in feature space. While the connections using

the distance in feature space contain a mixture of negative samples and positive samples, almost all unlabeled samples connected to the same samples can be seen as positive.

### 5.2.2 Learning the Target Classifier

We learn the target classifier $f^\tau$ using only a few labeled target samples and a large amount of unlabeled samples. Using Equation (3.10), we can predict the pseudo-label $\hat{y}_j$ of any unlabeled sample $\mathbf{x}_j$ based on the weighted combination of source classifiers. Each source classifier $f^s$ is learned from the source labeled data $D^s = \{(\mathbf{x}_i, y_i)|1 \leq i \leq n_s\}$ where $n_s$ is the number of labeled samples from a source class. By conducting the training under the boosting framework, we are able to collect all base classifiers $h_c$ from $f^s$ of $k$ source classes into set $\mathcal{H}_c = \{h_c(\mathbf{x}_i)|h_c(\mathbf{x}_i) \in f^s, 1 \leq c \leq C\}$ where $C = k \times T$ where $T$ is the number of boosting iterations.

Similarly on the target class, we adopt the boosting framework to train a composite classifier $f^\tau$ from multiple base classifiers. At boosting iteration $t$, we learn a base classifier $h'_t$ by rewriting the empirical loss and regularization in (3.9) into the following:

$$h'_t = \arg\min_{\mathcal{H}_c} \frac{1}{n_l} \sum_{i=1}^{n_l} (h_t(\mathbf{x}_i) - y_i)^2 d_i + \lambda \frac{1}{n_u} \sum_{j=1}^{n_u} (h_t(\mathbf{x}_j) - \hat{y}_j)^2 d_j, \tag{5.2}$$

where $h_t(\mathbf{x}_i)$, $y_i$, $d_i$ are respectively the label by base classifier $h_c$, the actual label, and the weight of the $i^{th}$ labeled sample; $h_t(\mathbf{x}_j)$, $y_j$, $d_j$ are respectively the label of base classifier $h_t$, the pseudo-label, and the weight of the $j^{th}$ unlabeled sample; and $\lambda$ is the regularization parameter.

It can be observed that parameter $\lambda$ in (5.2) and the source classes play a role in the error bound of the target class. When $\lambda = 0$, the error bound reduces to one that uses only

**Algorithm 1:** Spatio-Temporally Regularized Adaptive Learning (STRAL)

**Input:** $k$ source classifiers $f^s$, labeled data $D_l^\tau$ and unlabeled data $D_u^\tau$, distribution $d_i$ and $d_j$ to be uniformly distributed, and regularization parameter $\lambda$

**Output:** Target classifier function $f^\tau : \mathcal{X} \rightarrow \mathcal{Y}$

1: Construct graph $\mathbf{G}_{ST}$ with optical flows through the video
2: Compute $\mathbf{W}_{i,j}$ using the shortest path between 2 nodes in $\mathbf{G}_{ST}$ using (5.1)
3: Learn source relevancy $\mathbf{B}^s$ by solving (3.11)
4: Predict the pseudo labels $\hat{y}_j$ using (3.10)
5: **for** $t = 1$ to $T$ **do**
6: Find the classifier $h_t$ that minimizes (5.2) resulting in regularized loss $\epsilon_t$
7: Set $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$ where $\epsilon_t < \frac{1}{2}$
8: Update the weight distribution

$$d_i \leftarrow d_i \times e^{-\alpha_t h_t(x_i) y_i}$$
$$d_j \leftarrow d_j \times e^{-\alpha_t h_t(x_j) \hat{y}_j}$$

9: **end for**
10: **return** $f^\tau(\mathbf{x}) = \text{sign}(\sum_t \alpha_t h_t(\mathbf{x}))$

labeled target samples. Thus, the proposed method is degraded into a traditional approach such as AdaBoost. As $\lambda$ increases, the influence of the source classifiers becomes higher. We recommend a larger value of $\lambda$ under the scenario of lack of labeled target samples and the training must resolve around the regularization. To this end, the effect of transferring knowledge from the source classes is greater than within the target class. In the extreme case when $\lambda \rightarrow \infty$, the target samples will no longer be meaningful and the proposed method relies solely on weighted source classifiers. Thus, effective transfer is possible only if the difference between source and target classes is small.

Since the proposed method is based on regularizing the unlabeled samples with respect to their spatiotemporal connectivity, we refer to it as the Spatio-Temporally Regularized Adaptive Learning (STRAL). Algorithm 1 summarizes the main procedure in STRAL. As STRAL is directly based on the boosting framework, its convergence properties can be inherited from AdaBoost [68]. Furthermore, since the condition $\epsilon_t < \frac{1}{2}$

is satisfied in algorithm 1, the prediction error $\epsilon$ over the target data $D^{\tau}$ is bounded by $\epsilon \leq 2^M \prod_{t=1}^{T} \sqrt{\epsilon_t(1-\epsilon_t)}$, and the upper bound of the associated generalization error is given by $\epsilon + \mathcal{O}(\sqrt{\frac{Md_{VC}}{n_l}})$, where $d_{VC}$ is the VC-dimension of the base classifier [86].

## 5.3 Experiments

We conduct a series of experiments on images of ants and termites to demonstrate the performance of the proposed method STRAL against three classification methods: AdaBoost [68], TaskTrAdaBoost [86], and FeatReg which is a version of the proposed method but uses the feature distance of unlabeled samples instead of using the spatiotemporal information and loosely based on a method in [12]. The AdaBoost does not employ either the transferred knowledge from the source domains or any regularization on unlabeled samples. A transfer learning approach, TaskTrAdaBoost, exploits the knowledge from source domains to be re-used on the target domain. Since TaskTrAdaBoost does not explicitly provide a regularization term, it does not take advantage of any regularization on unlabeled data. When the target training data is small, TaskTrAdaBoost is demonstrated to be effective in exploiting the existing knowledge to learn the new data in [86]. To isolate the effect of spatiotemporal regularization on the classification accuracy, we form FeatReg, a method adopted from [12] which determines the relevancy of the source classifiers solely on the feature distance of unlabeled samples instead of using the spatiotemporal information. For simplicity, we used decision stumps as base classifiers to illustrate the relative performance of the proposed method. Slightly better accuracy may be achieved by using more complex base classifiers such as decision trees or SVMs, but we do not expect the relative performance gain to differ.

Table 5.1: Description of the experimental datasets

| Species | Frames | Image Dim | Objects | Marking | Common Size |
|---|---|---|---|---|---|
| Messor | 4000 | 1240 X 960 | 49 | Light Tint | 15 X 35 |
| Temnothorax | 5000 | 960 X 540 | 50 | Color Paint | 20 X 55 |
| Macrotermes | 2500 | 982 X 982 | 29 | Unmarked | 35 X 70 |

### 5.3.1 Procedures

Data Our data consist of three videos each taken by a difference biological research group. The first video contains a colony of 50 *Temnothorax* ants taken at thirty frames per second. Some of the ants are painted to assist in identification. The second video contains a colony of 49 *Messor* ants taken at a higher resolution. We also obtain 2,500 frame video of 29 *Macrotermes* termites which are unmarked (see Figure 5.1). In each video, the positions of ants and termites were manually labeled in all frames as their respective biology experts. To evaluate the effectiveness of the proposed method, we collected 6,000 image patches (2,000 image patches from each video). One half of the image patch are positive samples while the other half are negative samples. Each image patch is cropped from each video randomly, and the label is determine as positive if its center is less than half of the average length of the respective insect from the ground truth position. In the experiment, we select the Messor dataset as the target and other data sets serve as sources. In the target data, we divide the samples into non-overlapping two halves: one half for training and another half for testing. The details of each data set are described in Table 5.1.

### 5.3.2 Reduction on Training Effort

We measure the training effort $n_l$ as the number of labeled samples. We conduct an experiment with $n_l$ varying from 4 to 256. Note that $n_l$ consists of equal number of positive

Table 5.2: $A_{ROC}$ (Mean $\pm$ Standard Deviation) of boosting-based classification methods for different number of labeled samples $n_l$. A performance number is highlighted in **bold** if it is significantly better than all other methods based on a paired t-test at $p = 0.05$.

| $n_l$ | **AdaBoost** | **TaskTrAdaBoost** | **FeatReg** | **STRAL** |
|---|---|---|---|---|
| 4 | $0.71 \pm 0.11$ | $0.72 \pm 0.09$ | $0.81 \pm 0.03$ | $0.83 \pm 0.02$ |
| 8 | $0.72 \pm 0.08$ | $0.78 \pm 0.07$ | $0.83 \pm 0.02$ | $\mathbf{0.84 \pm 0.01}$ |
| 16 | $0.76 \pm 0.05$ | $0.81 \pm 0.03$ | $0.84 \pm 0.02$ | $\mathbf{0.85 \pm 0.01}$ |
| 32 | $0.80 \pm 0.04$ | $0.81 \pm 0.03$ | $0.84 \pm 0.01$ | $\mathbf{0.85 \pm 0.00}$ |
| 64 | $0.82 \pm 0.03$ | $0.83 \pm 0.02$ | $0.85 \pm 0.01$ | $0.85 \pm 0.00$ |
| 128 | $0.84 \pm 0.02$ | $0.84 \pm 0.02$ | $0.85 \pm 0.01$ | $0.85 \pm 0.01$ |
| 256 | $0.85 \pm 0.01$ | $0.85 \pm 0.01$ | $0.86 \pm 0.01$ | $0.86 \pm 0.01$ |

and negative samples. For every number of labeled samples, we replicated the experiment 30 times and record the average performance of each method. The performance of the detector would be dependent on the training data as shown in Table 5.2; and the maximum attainable accuracy would be the performance with the highest number of training samples. We note that as expected AdaBoost, TaskTrAdaBoost, FeatReg, and STRAL reach the similar maximum accuracy close to $A_{ROC} = 0.86$ when trained with a large number of samples ($n_l = 256$). The main objective of our experiment is to reduce the training effort. Thus, we focus on comparing how the accuracy *improves* especially when the training size is small.

First, STRAL reduces the training effort significantly while achieving comparable performance to TaskTrAdaBoost and FeatReg, respectively. For example, STRAL employs only $n_l = 16$ to achieve similar accuracy of $A_{ROC} = 0.85$ as the TaskTrAdaBoost with $n_l = 256$ samples. Similarly, in order to reach the same $0.85$ performance, FeatReg requires up to 64 labels comparing to only 16 labels in STRAL. Second, the stability of the STRAL is also improved as the standard deviation of $A_{ROC}$ reduces from TaskTrAdaBoost by at least 3 times when the number of training labels is small ($n_l = 4, 8, 16$) as indicated

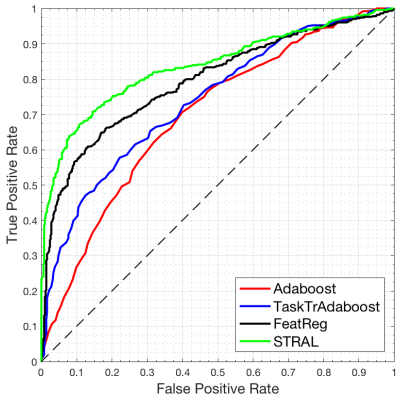



Figure 5.4: Comparisons of classification accuracy of evaluating methods using only 4 labels to simulate the rapid training ability of all evaluating methods

in the first 3 rows of Table 5.2. This result provides evidence that the usage of regularization on unlabeled samples is indeed beneficial supplement the weighting of the existing classifier and subsequently reducing the number of training labels. Third, at training effort $n_l = 8$, 16, and 32, STRAL yields higher accuracy over FeatReg which does not use the spatiotemporal regularization. Using the sample t-test with 95% confidence level, STRAIL accuracy is statistically higher than that of FeatReg. Evidently, the usage of spatiotemporal information to construct matrix $\mathbf{W}_{i,j}$ is helpful to regulate the training of the target classifier even when only a few labeled samples are available.

### 5.3.3 Improvement on the Initial Accuracy

To simulate rapid training, we compate the accuracy of STRAL against those of AdaBoost, TaskTrAdaBoost, and FeatReg at the initial round of training. We evaluate the impact of the spatiotemporal regularization on the initial detection accuracy by employing only 2 positive labels which are randomly selected for all evaluating methods. An equal number of negative labels are also selected to keep the training set balanced. To eliminate

the bias of selecting the target training samples, we also replicated the experiment 30 times, and record the average performance of each method.

Figure 5.4 illustrates the ROC curve of all comparing methods with only $4$ training samples. The proposed method gives the highest area under the curve. When encountered a new object type which has a different from what had been seen before, providing few training samples from the new domain yields poor accuracy in AdaBoost ($A_{ROC} = 0.71$) as well as TaskTrAdaBoost ($A_{ROC} = 0.72$). However, with the same number of training samples, STRAL rapidly adapted to the new object type and produces $A_{ROC}$ already at 0.83, which was only 4% lower than its maximum performance (as seen in Table 5.2).

## 5.4 Summary

In this paper, we proposed a rapid training for a biological object detection method by leveraging the spatiotemporal connections among the unlabeled data to transfer the existing knowledge from multiple sources. Our key contribution is the development of a spatiotemporal regularization term to the standard loss minimization formulation. Based on the smoothness in the predicted labels on the spatiotemporal connected samples, our proposed method learns the weights of the classifiers from multiple source classes. The evaluation on three data sets of social insects with 6,000 samples demonstrates that our method reduces the training up to 16 times while maintaining a comparable performance to previous approaches. For future studies, we plan to investigate on a more efficient learning of the source weights by addressing the unlabeled labels that display fast movements in our experiment.

# CHAPTER 6: POLLEN CLASSIFICATION WITH TARGET-DIRECTED SAMPLING

In this chapter we propose the *target-directed sampling* (TDS) method to reduce the amount of training effort required to classify the new (target) class. A majority of training effort is spent on searching for the appropriate samples to label in a pool of *unlabeled samples*. The distribution of unlabeled samples in biological images is usually unbalanced, which means that while the samples of some classes is rather abundant, other classes can have significantly less samples. On a highly unbalanced class distribution, it is particularly time-consuming to obtain these samples when the human experts have to search for them in a large number of unlabeled samples. Rather than having a human expert submit a set of labeled images as training samples, TDS chooses particular samples from the unlabeled pool for a human expert to label. Since the goal is to reduce training effort of the experts, the labeling should only be done on the most useful training samples from the target class that needs additional training samples. Initially, the number of training samples available for the target class is usually small, thus additional training samples from the target class is more needed than that from other classes. Thus, we propose the TDS method to search for the most confusing samples that are most likely to belong to the *target* class in a pool of unlabeled samples.

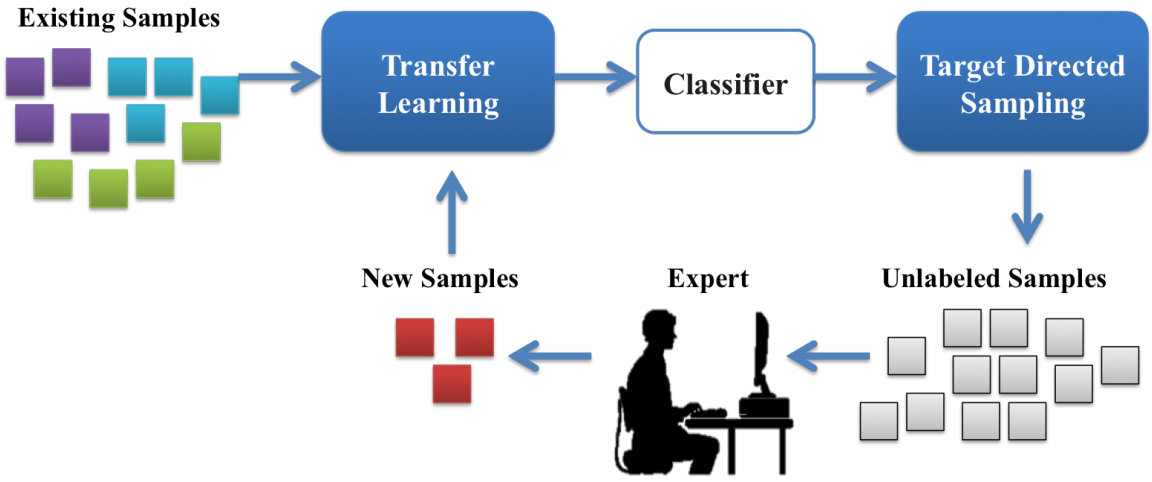



Figure 6.1: A training iteration of the proposed method. Initially, a classifier is constructed using Transfer Learning as discussed in Chapter 3. This classifier is then employed to classify the unlabeled samples with some confidence scores. These confidence scores are utilized in the Target-Directed Sampling to select valuable training samples that are most likely to belong to the target class. The selected samples are presented to the expert to provide the class labels. Then, the labeled samples in added into the training data for the next iteration.

## 6.1 Overview

The overview of the method proposed in this chapter is briefly explained in Figure 6.1. In particular, a new approach for training a classification system is discussed as follow: The target-directed sampling (TDS) is proposed to effectively choose the unlabeled samples which are likely to be from the target class. Particularly, the unlabeled samples which are most likely to be confused between a target class and another class are chosen *first* for training. Without loss of generality, we assume there are some existing data with available class labels and a large number of unlabeled data. As discussed in details in Chapter 3, the classification rules are trained on a few existing object classes (the sources) and is able to extrapolate some of that knowledge into the target class. These classification rules are demonstrated to be reliable enough to build the initial classifier which a small number of

new samples. The approaches have been successfully applied into pollen grains classification. The method presented in this chapter has been published in [55].

## 6.2 Target Directed Sampling

For our classification problem, the unlabeled samples with a small margin between the target class and a source class are more likely to be valuable. In this context, the limitations of Equation 3.12 are in its incapability to looking for the training samples from the target class. Since there's a lacking of training samples from the target class, an additional target training example would have more influence to the construction of the target classifier than an additional source training example. In other words, knowing additional target samples would help learning the target classifier more effectively. A constraint on the minimization problem is needed to guide the search for the target samples in the unlabeled pool In such way, the problem is reformulated as:

$$\begin{aligned} \underset{\mathbf{x}\in D_U}{\arg\min} \quad & P(\hat{y}_1|\mathbf{x}) - P(\hat{y}_2|\mathbf{x}) \\ \text{subject to} \quad & \hat{y}_1 = \tau \end{aligned} \tag{6.1}$$

where $\hat{y}_1$ is the maximum confident label and $\tau$ is the target class. Such formulation ensures that the samples with a small margin but has a highest confidence of the target label are selected. This would help the selection method chooses more target training samples.

**Handling Initial Classification** The formulation in Equation (6.1) works well for some scenarios where the classifier has predict the correct label of the highest confidence score. However, there is still one limitation with considering only the highest confidence label $\hat{y}_1$. Since the initially there are only very few training samples, the quality of the classification

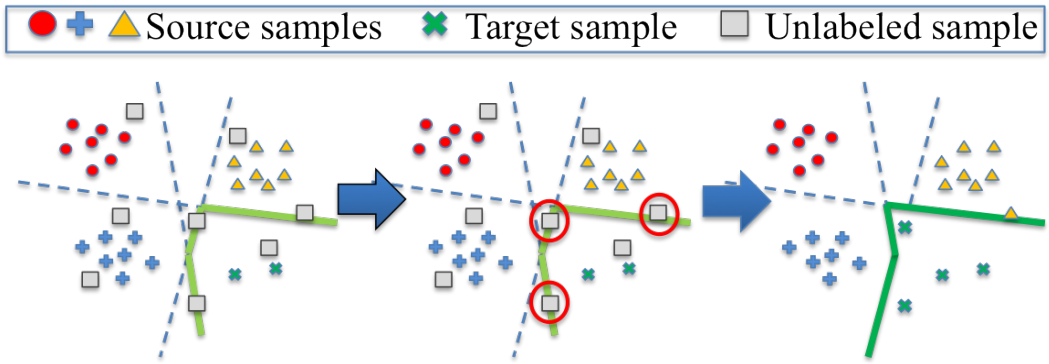



Figure 6.2: TDS employed the transferred classification rules to select the most valuable samples (in red circles) along the decision boundary between the most and the second most confusing classes. These samples are most likely to contribute to the improvement of the decision boundary.

is rather low. As a result, the confident score can be incorrect about the highest confidence level. Thus the highest confidence samples may not be the actual target class since the confident was computed on incomplete training data. To alleviate this problem, we explicitly incorporate the second highest confidence label into minimization constraint as:

$$\begin{aligned} \arg\min_{\mathbf{x}\in D_U} \quad & P(\hat{y}_1|\mathbf{x}) - P(\hat{y}_2|\mathbf{x}) \\ \text{subject to} \quad & (\hat{y}_1 - \tau)(\hat{y}_2 - \tau) = 0 \end{aligned} \tag{6.2}$$

In the above equation, we can rewrite the minimization constraint into its equivalency $\hat{y}_1 = \tau$, or $\hat{y}_2 = \tau$ to focus the search on the most probable target samples, while the margin sampling incorporated the posterior probabilities of both the most the second most likely label. Constrained by Equation (6.2), samples with a small margin between the target class and a source class are more likely to be selected. The selection criterion does not only differentiate the target class from other source classes but also improves the acquisition of target training sample, thus benefited the classification.

**Handling multiple target classes** The objective function in (6.2) can be further generalized to handle multiple target classes. This can be achieve by simply incorporating additional terms into the minimization constraints. Specifically, we can derive as the following:

$$\begin{aligned} \underset{\mathbf{x}\in D_U}{\arg\min} \quad & P(\hat{y}_1|\mathbf{x}) - P(\hat{y}_2|\mathbf{x}) \\ \text{subject to} \quad & \prod_{j=1}^{m}(\hat{y}_1 - \tau_j)(\hat{y}_2 - \tau_j) = 0 \end{aligned} \tag{6.3}$$

where $m$ is the number of target classes. At this point, multiple target classes can be acquired in order to improve the classification performance. The minimization constraint is still restricted enough so that it only allows specific classes to be considered.

**Generalizing the Formulations** The objective function in Equation (6.3) used only the margin from two probability terms $P(\hat{y}_1|\mathbf{x})$ and $P(\hat{y}_2|\mathbf{x})$. Additional confident label which may include the target class does not considered in the current formulation. In order to make this formulation applicable in the general case, the margin in the objective function can be relaxed as:

$$\begin{aligned} \underset{\mathbf{x}\in D_U}{\arg\min} \quad & P(\hat{y}_1|\mathbf{x}) - P(\hat{y}_n|\mathbf{x}) \\ \text{subject to} \quad & \prod_{j=1}^{m}\prod_{i=1}^{n}(\hat{y}_i - \tau_j) = 0 \end{aligned} \tag{6.4}$$

where $n$ indicates the number of predicted class labels to be consider as possible target samples. This formulation is also considered as the general form of the proposed Target-directed Sampling (TDS) method. The selection of valuable training samples using TDS is illustrated in Figure 6.2.

**Relations to Other Sampling Methods** Depending on the settings of $n$ and the presence of the minimization constraint, Equation (6.4) can be related to other sampling methods:

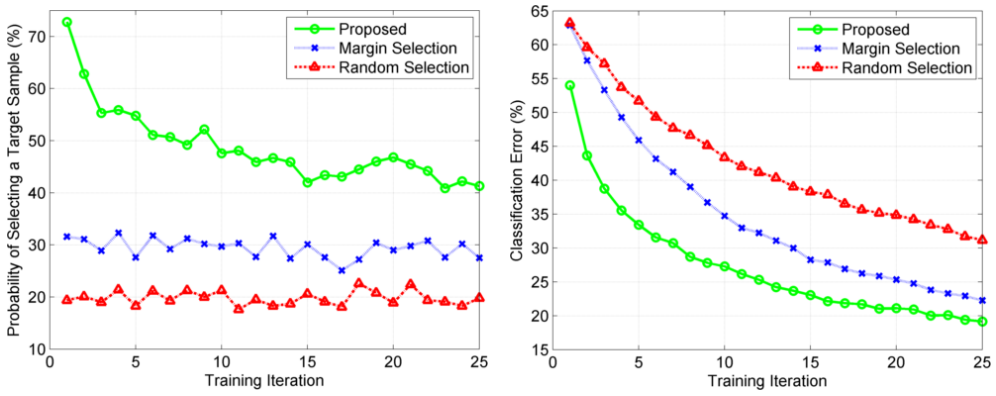
- If $n = 1$, TDS would select the predicted target label with the least confident score. When there's no minimization constraint, the objective function would reduce to the least confidence sampling method [73].

- If $n = 2$, TDS would consider only the most and second most confident term which are likely to be target class. Without the minimization constraint, TDS degenerates into the Margin Sampling method.

- If $n = k(k > 2)$, TDS becomes more generalized and consider the margin between the most and $k^{th}$ most confident term which broaden the search for the target samples that might have been mis calculated as low classification confidence.

The insight of Equation (6.4) reveals the connections of the TDS with some other sampling methods. Those methods can be regarded as special cases of TDS. In other words, TDS provides a unified framework to deal with different challenges of effectively sampling the unlabeled data. TDS can also provide flexibility to meet the requirements of different applications by adjusting parameter $n$ and the the minimization constraint. In our biological applications, TDS works best under $n = 2$ with the minimization constraint.

**Probability of Selecting a Target Sample** Figure 6.3 depicts the effectiveness of the proposed formulation. As seen in figure 3(a), the probability of selecting a target samples in a biological dataset from [48] with 3 selection methods: random sampling, margin sampling, and proposed. While margin selection has been shown to be able to select more target training samples than random selection (30% compared to 20%), our formulation has the highest probability (upper 40%) to select training samples from the target class both margin and random methods. Initially, our formulation yields a significantly high probability




(a) Target Example Probability (b) Classification Error Reduction

Figure 6.3: The probability of selecting a target example in the unlabeled data.

of selecting a target samples. The reason behind this effect is that the transferred classifiers are likely to be over the true decision boundary of the target class where there is more target samples. As our selection formulation encouraged the selection of target samples, more of those samples are selected for training. Having more target training samples, the resulted classifiers has considerably less error than other selection method. As shown in the figure 3(b), the classification error is reduced significantly particularly in the early training iteration. This effect shows that the selection criterion has coupled well with the transferred classifiers to select the most valuable training samples. Using this selection criterion, we rank all unlabeled samples into an order set. The top samples of this ordered set is selected to be labeled by the human annotator for the next iteration of training.

Although our focus is on selecting the target training samples, it is still worth mentioning that many other methods can also achieve guiding the selection into the target samples such as iterative clustering [44]. In our model, we choose to modify the minimization constraint because it is easy to interpreted and supports multiple options of the target selection. By incorporating this selection criterion, we ensure the most valuable samples for the target

classifier will be selected. While the traditional margin sampling have focused on selecting a minimal margin example from *any* class, this selection criterion favors the confusing samples that most likely to belong to the *target* class.

**Details of the implementation** Our method is implemented in MATLAB on an Intel Xeon with 4GB memory. The codes are currently available as a MATLAB toolbox on our website[2]. The base classifier is implemented as a decision stump. We expect easy integration into other datasets or frameworks. The researchers and collaborators who are interested are encouraged to use the toolbox and provide feedback and suggestions.

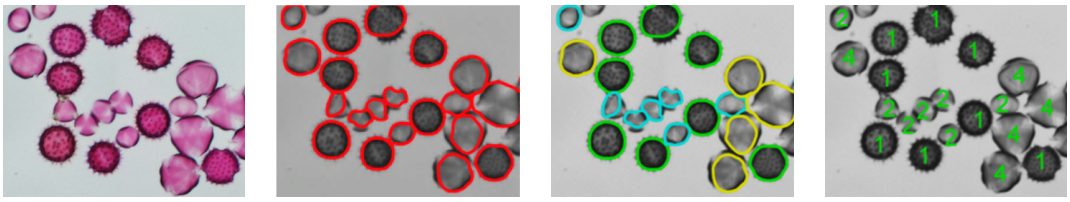
## 6.3 Experiment: Pollen Classification

In this section, we evaluate an automatic classification method to discriminate pollen grains coming from a variety of taxonomic types. Our experiment demonstrates that the proposed method reduces the training effort of a human expert up to 80% compared to other classification methods while achieving 92% accuracy in pollen classification.

### 6.3.1 Biological Background

The pollen grains of different plant taxa exhibit many different shapes and sizes, often bearing characteristic ornaments like spines or furrows. This structural diversity has made the identification of pollen grains an important tool in a variety of fields. Despite the myriad of applications, the classification of pollen grains is still a tedious and time-consuming process that must be performed by highly skilled specialists. Paleoecologists use layers of lake sediments of pollen to reconstruct the past history of vegetation in different parts of the world. Aerobiologists identify and quantify wind-borne pollen to warn allergy suf-

[2]http://fcl.uncc.edu/nhnguye1/ActiveTransfer.html



(a) (b) (c) (d)

Figure 6.4: An example of pollen classification using the proposed method. (a) A cropped region from a microscopic image containing a few pollen types. (b) The pollen grains are detected with active contours where features are extracted. (c) The classification pollen grains shown in unique colors. (d) The corresponding manual labels from a human expert.

ferers in periods of elevated risk. Since the vast majority of plant species, including most agricultural crops, depend on animal pollination for their reproduction, there are many applications for the classification of animal-borne pollen. Classifying pollen collected from pollinators like bees, hummingbirds and butterflies provides a record of different flower taxa each individual has visited. For ecologists, this provides a window into the complex network of interactions between plants and pollinators in a community. In agriculture and conservation biology, pollen classification has practical implications for which plants are actually receiving pollination services, as well as the nutrition and health of the pollinators themselves.

In chapter 6, we have shown the framework for reducing the number of training samples by using an active transfer learning approach with a minimization constraint. In this section, using a set of pollen images containing various mixtures of nine pollen types, we demonstrate that the proposed method reduces the training effort of a human expert as much as 80% compared to other classification methods while achieving 92% classification accuracy. Additionally, we show an application of the proposed method in pollen counting. Figure 6.4 illustrates an example of the pollen classification using the proposed method.

**Spike Count** Besides adopting some generic shape and texture features, we added a new feature to capture the pollen spikes and spores. Spikes and spores are important features that can be discriminative among some pollen types. Figure 6.5 provides details on the extraction of the spikes. Due to the inherent image quality and resolution, the active contours are unable to capture the spikes and spores which often are too noisy and faint. However, knowing the final position of the active contour will help identify these spikes. Thus, we extract a "ring-like" binary mask along the active contour to estimate the region of the spikes. Within this mask, a spike usually appears as a fluctuation in intensity. For each pollen grain, the average intensity of pixels at each angle is computed. Then, a 1-D signal formed by the intensity values at every angle is generated. A local minimum in the signal is detected as a spike if its difference to the values at both adjacent local maxima are within a range of $[0.05, 0.40]$. Note that we are able to distinguish a spike from the border of a neighbor grain which yields a large intensity difference (as seen in Figure 6.5). The spike count is computed as the number of local minima which satisfies the above condition.

### 6.3.2 Experimental Setup

We conduct a series of experiments on pollen images to demonstrate the performance of the proposed method against two boosting-based classification algorithms: AdaBoost [29] and TaskTrAdaBoost [86]. The AdaBoost algorithm does not employ either the transferred knowledge from the source types or any selection strategy for new training samples. A recent transfer learning approach, TaskTrAdaBoost, exploits the knowledge from source types to be re-used on the target types. Since TaskTrAdaBoost does not explicitly provide a strategy to select new samples, its new training samples are selected randomly from the

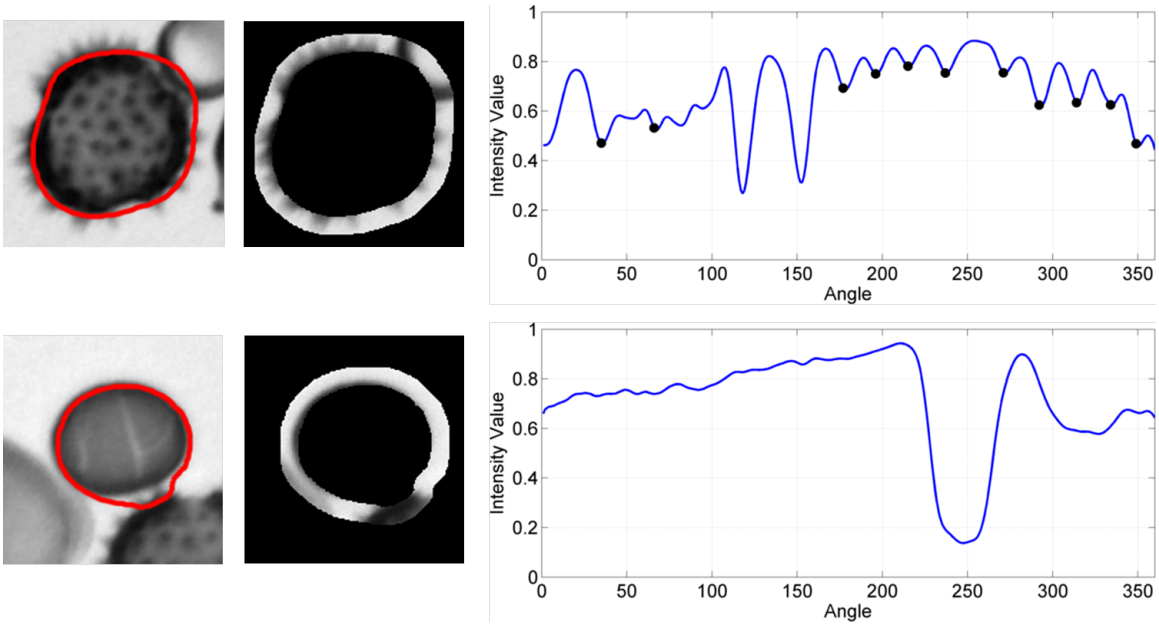



Figure 6.5: The spike count discriminates pollen type which has many spikes (upper row) from no spike (lower row). Left column: the active contour is unable to capture the spikes. Middle column: the radial mask indicates the estimated region of the spikes. Right column: a radial intensity curve consists of the average intensity of the pixels at each angle. The local minima which satisfies a specific condition are counted as spikes (as black dots along the curve). Note that although the pollen in the lower row has a neighbor with spikes, no spike is counted as the intensity fluctuation at the boundary is too large.

unlabeled data. When the target training data is small, TaskTrAdaBoost is demonstrated to be effective in exploiting the existing knowledge to learn the new data in [86]. In the following sections, we described in details the evaluation dataset, the detection performance, and the classification procedure.

**Data** The pollen images used to test the proposed method were taken from an experiment done on domestic honey bees housed at the Sonoran Arthropod Studies Institute in Tucson, Arizona. On a microscope with a motorized stage, non-overlapping regions of the microscope slide are scanned at 40x magnification into a digital image using the software NIS-Elements (Nikon). Each image covers approximately 1mm$^2$ at a resolution of 0.23

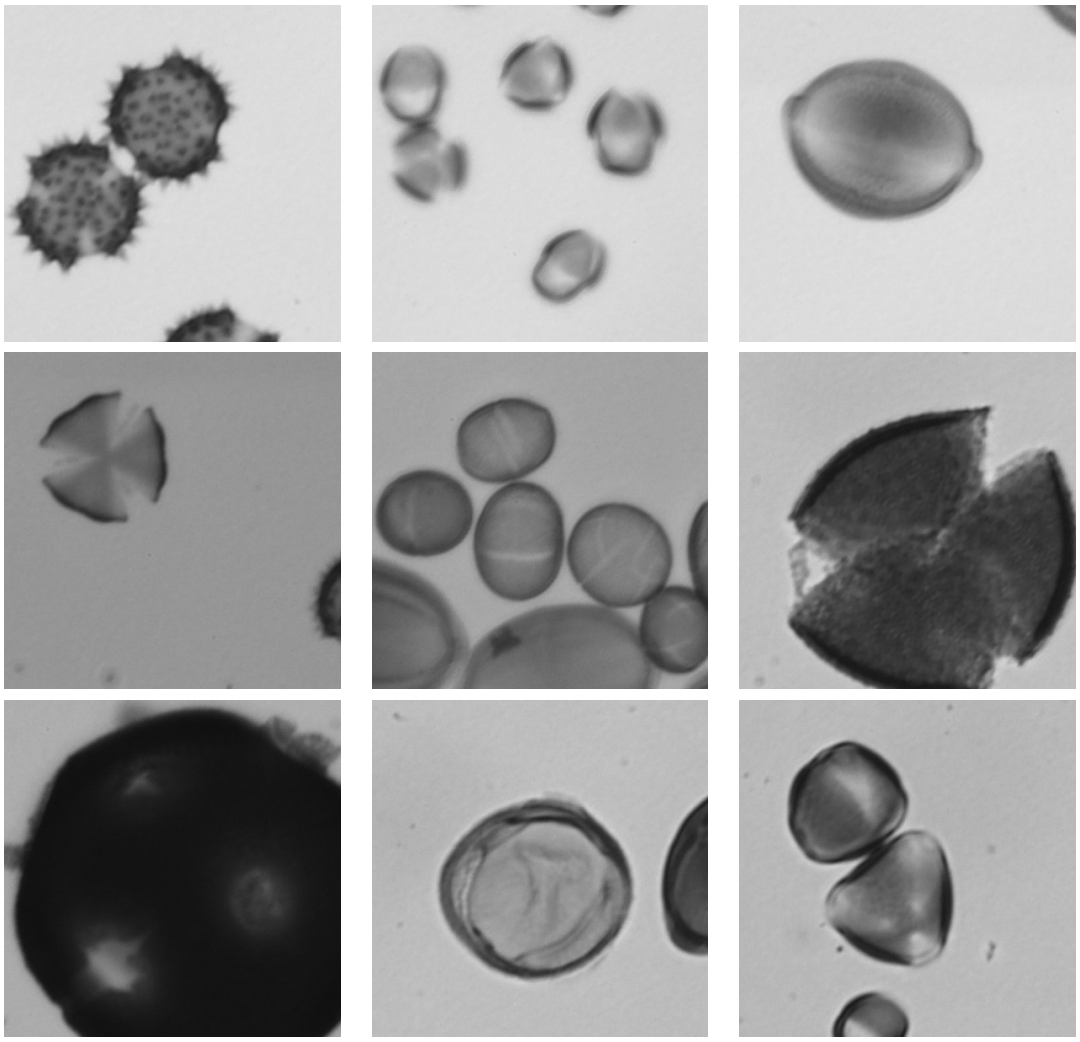


Figure 6.6: Representative samples of each pollen type.

$\mu$m/pixel. Previous classification by a human expert indicates that these samples contain a various mixtures of pollen types. We collect a total of 768 grains of 9 pollen types as shown in Figure 1.1. Types 1-3 were most common, with a frequency of at least 16% in at least one sample, while the other types were more rare, but nonetheless had a frequency of at least 3% in at least one sample. A subset of the pollen grains in these images were individually labeled for use as training and testing samples (see Table 6.1).

Table 6.1: Description of the pollen types.

| **Type** | **Name** | **# Labeled Samples** |
|---|---|---|
| 1 | Asteraceae | 289 |
| 2 | Larrea | 120 |
| 3 | Trixis | 145 |
| 4 | Phacelia | 39 |
| 5 | Lamiaceae | 31 |
| 6 | Carnegiea gigantea | 26 |
| 7 | Cylindropuntia | 22 |
| 8 | Datura | 63 |
| 9 | Prosopis | 33 |

Table 6.2: The classification accuracy of the proposed method (in %) with respect to the spike count feature. The spike count improves the classification accuracy in many types (bold faced).

| Pollen Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | Mean |
|---|---|---|---|---|---|---|---|---|---|---|
| Without Spike Count | 96 | 100 | 97 | 89 | 72 | 93 | 92 | 93 | 70 | 89 |
| With Spike Count | **100** | 100 | 97 | 89 | **77** | 93 | 89 | **98** | **83** | **92** |

**Classification Procedure** A robust assessment of the performance of the pollen classification system has been made by comparing the classified pollen grains obtained automatically by the classification system with those determined manually by a experienced palynologists. To evaluate the accuracy of a classification method, we select each pollen type as the target and execute a 3-fold cross validation. For each target, we choose randomly 6 other types from the training set to be used as the source types. All classification methods are provided initially with only 3 target samples which are selected randomly, and the rest of samples form the unlabeled pool. At each iteration of training, 5 additional unlabeled samples are selected for a human expert to label; then the classifier is re-trained based on the newly labeled samples. We repeat the training for 50 iterations. The classification accuracy on the target type is evaluated by the ratio between the number of correctly

classified target samples and the total number of target samples in the test set. We record the classification accuracy of the target type per each iteration. The overall accuracy is computed as the average over all types. To eliminate the bias of selecting the initial target training samples, the experiment is replicated 30 times and the average performance of each method is recorded.

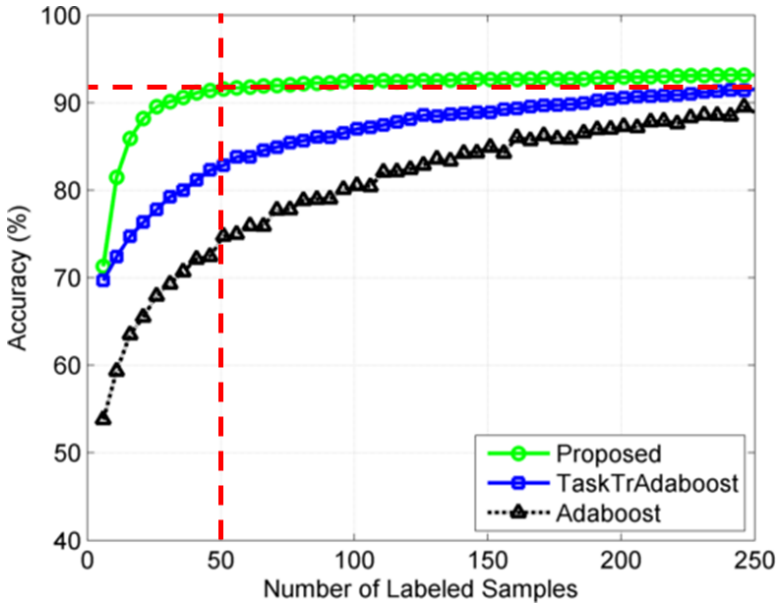



Figure 6.7: Comparison of the classification methods with respect to the number labeled samples. The proposed method achieve a comparative performance to other method while requires 80% less number of labeled samples.

### 6.3.3 Results

We assess the effectiveness of the proposed method on three different aspects: (1) the accuracy improvement with respect to the spike count feature; (2) the reduction of the number of labeled samples to achieve reasonable performance to other classification methods; and (3) the consistency to a human expert in a biological application.

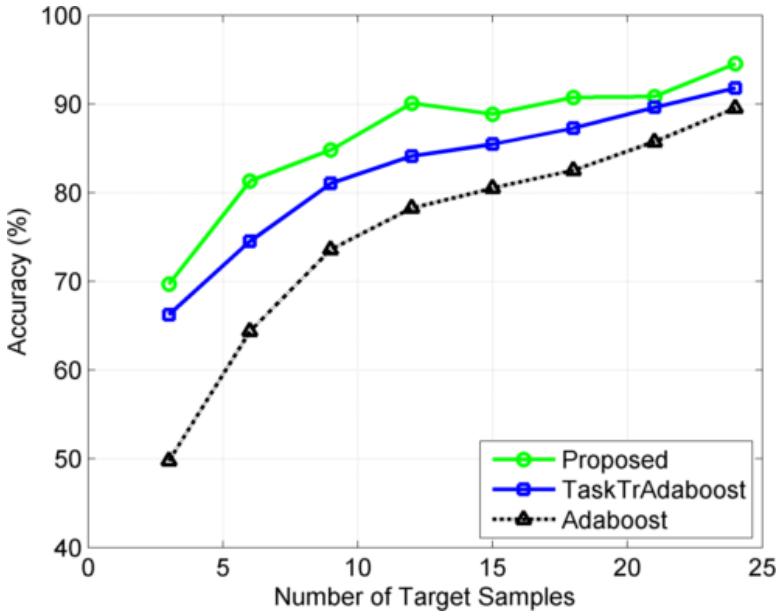



Figure 6.8: Comparison of classification methods with the same number of *target* samples. The proposed method yields a higher accuracy than other methods implying that it selects more valuable target samples which is likely to improve the classification.

**Improvement in Classification Accuracy** We first measure the improvement in classification accuracy by integrating the spike count feature. Table 6.2 provides the accuracy of the proposed method with and without the spike count. Without the spike count, type 5 and 9 are confused with type 1 with the errors rates of 13% and 18% respectively. The spike count feature improves the accuracy by 5% for type 5 and 13% for type 9 since it discriminates them with type 1 which has many spikes. Additionally, the accuracy improves in type 8 by 5% since it has a smooth boundary with no spike. We also observe a slight increase in error rate for type 7 due to its confusion to type 6 which has a similar number of spike. Overall, the error rate is reduced over 3% in all types resulting in 92% accuracy.

**Reduction in Training Effort** Figure 6.7 displays the average testing accuracy on all target types with respect to the number of labeled samples. Note that the accuracy im-

proves with the proposed method at any number of labeled samples. The improvement with a small number of labeled samples is especially large: 11% over TaskTrAdaBoost and 23% over AdaBoost when trained with 10 samples. While TaskTrAdaBoost and AdaBoost require more than 250 labeled samples to achieve a 91% accuracy, the proposed method only requires 20% of the training effort, or 50 samples, to achieve a similar performance (as shown in the red dotted lines in Figure 6.7). This result suggests that the proposed method reduces the workload of the human annotator as high as 80% and still achieves a comparative performance to other methods. The reduction in training effort is caused by two reasons. First, the proposed method selects more target samples, on average 23 target samples out of the first 50 unlabeled samples while TaskTrAdaBoost and AdaBoost select only 11 target samples. Second, the selection criterion selects more useful training samples. Figure 6.8 compares the results of all three methods with the same number of *target* samples. The performance of TaskTrAdaBoost is higher AdaBoost due to the transfer of rules which is consistent with [86]. Our method shows an additional improvement over TaskTrAdaBoost supporting that it selects not only more target samples but also the more effective ones.

**Application in Pollen Counting** Counting the grains of each pollen type on the microscope slides is a slow laborious process that must be performed by highly skilled palynologists. We compute the pollen distribution from the classified grains and compare to the ground truth established by a palynologist on the data set (described in Section 6.3.2). Using t-tests with 90% confidence level, there is no statistically significant difference between manual and automated methods; and the average error is as small as 3.6% (refer to Table

Table 6.3: Comparison between the automatic and manual methods in the application of pollen count. Since the experiment is replicated 30 times, the automatic count is provided as Mean±Standard Deviation. The manual count is done only once by a palynologist. Using t-tests at 90% confidence level, there is no significant difference found between the automatic and manual methods.

| Pollen Type | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Automatic Count | 286±2.6 | 123±0.7 | 147±3.1 | 38±1.7 | 32±3.0 | 28±1.0 | 20±0.7 | 62±2.6 | 32±2.8 |
| Manual Count | 289 | 120 | 145 | 39 | 31 | 26 | 22 | 63 | 33 |
| Count Error (%) | 1.0 | 2.6 | 1.5 | 1.0 | 3.9 | 6.6 | 9.8 | 2.4 | 3.4 |

6.3). Higher error rates are observed on type 6 (6.6%) and 7 (9.8%) due to the low sample frequency of such types. Compared to other automated pollen counters such as in [38], the difference in count in our case is indeed small (on average $\leq 2$ grains per type). In the identification of pollen collected from pollinators to record the number of visited flowers, an average error under 4% is a small fraction of the total variational responses. Thus, we believe the method has enabled a reliable pollen counter for the biology community.

## 6.4 Summary

In this chapter, we propose a *target-directed sampling* (TDS) method designed to reduce the amount of training effort required to classify a new object class (the target). The training effort can be directly relate to the number of labeled samples which the human experts have to annotate. TDS is proposed to effectively choose the unlabeled samples which are likely to be from the target class. In particular, the unlabeled samples which are believed most likely to improve the performance of the new classifier are chosen *first* for training. The classification model which is employed in TDS is constructed from the classification rules extracted from the existing classes. These classification rules are demonstrated to be reliable enough to build the initial classifier which a small number of new samples. As a result, the initial classification model is reliable enough to be used to select additional training samples. The structural diversity made the classification of pollen grains an important tool in a variety of fields. We discriminate pollen grains with a variety of taxonomic types. The pollen classifier achieves 92% accuracy in pollen classification while reducing the training effort up to 80% compared to other classification methods. We believe the proposed method shows great potential toward the automation of pollen identification and

counting which is commonly done by palynologists. We believe these results will enable a wide range of application for a object classification system with minimal training effort from a human annotator.

# CHAPTER 7: CONCLUSIONS AND FUTURE DIRECTIONS

To automatically classify the biological images, machine learning techniques have been widely used to train the classifiers from labeled images. For a new class of biological object, a tedious and expensive labeling process is required from a human annotator. With the growing amount of biological data and the increasing number of classes to recognize, training a classification system needs a significant amount of manual labeling effort. The aim of this research is to effectively reduce the training effort by applying the previous knowledge from the existing classes as well as selecting the most valuable samples from the unlabeled data. The contributions of this dissertation research consists the following key components: First, a size-differential regularization is employed to refine the ranking of classification rules to alleviate the risk of over-fitting in case of small number of training samples. Second, a spatiotemporal regularization term to the standard loss minimization formulation is developed based on the smoothness in the predicted labels on the spatiotemporal connected samples, our proposed method learns the weights of the classifiers from multiple source classes. Third, the Target-Directed Sampling is proposed to focus the search toward the samples of the new class. In this dissertation we demonstrate the first two components and propose a solution for constructing a classifiers to achieve jointly training of correlated classifiers. Further research and experiments need to be in progress to conduct a viable solution for using size-differential regularization to effectively select additional training samples. We test the proposed methods with several real datasets including

biological cells, pollen grains, and planktons. The experiments indicate that the proposed framework achieves better performance than current machine learning approaches while requires as low as 10% of the labeled data.

## 7.1 Future Directions

There are several future directions of the research in this dissertation. We have demonstrated that the proposed method shows potential toward the automation of pollen identification and counting which is commonly done by biologists. One direction would be to improve the accuracy of detection methods using a variety of regularization. In this dissertation, we have demonstrated that the proposed method shows the reduction in training labels in biological experiment. However, the accuracy of the proposed methods can be improved in order to give it full potential to software application. To this end, advanced classification methods such as deep learning can be a possible solution.

Additionally, an interesting direction would be to explore the applications of spatiotemporal regularization on video analysis in team sports. Many videos in sport settings are collected with much spatiotemporal information. These meta data can potentially provide regularizations to a variety of classification task with little training effort.

Finally, one can investigate on large-scale adaptations of the proposed methods as the number of classes increases over time. It would be interesting to know if the proposed method could be applied in a situation where the number of classes are increasing over time. As one plans to investigate on a transfer learning approach to scale with an increasing number of classes, a possible direction can be aggregating the knowledge from a class as it is learned. As more classes are acquired, this collection of classification rules will

become larger and more diverse. Extensive experiments need to be conducted in order to demonstrate if this can be scaled as the number of new classes are incorporated.

# REFERENCES


[1] Y. Al-Kofahi, W. Lassoued, W. Lee, and B. Roysam. Improved automatic detection and segmentation of cell nuclei in histopathology images. *IEEE Transactions on Biomedical Engineering*, 57(4):841–852, 2010.

[2] H. Al-Mubaid and S. A. Umair. A new text categorization technique using distributional clustering and learning logic. *IEEE Transactions on Knowledge and Data Engineering*, 18(9):1156–1165, 2006.

[3] G. Allen, B. Hodgson, S. Marsland, G. Arnold, R. Flemmer, J. Flenley, and D. Fountain. Automatic recognition of light microscope pollen images. 2006.

[4] M. B. Altman, S. J. Wang, J. L. Whitlock, and J. C. Roeske. Cell detection in phase-contrast images used for alpha-particle track-etch dosimetry: a semi-automated approach. *Physics in Medicine and Biology*, 50(2):305, 2005.

[5] Y. Amit, M. Fink, N. Srebro, and S. Ullman. Uncovering shared structures in multiclass classification. In *Proceedings of the 24th international conference on Machine learning*, pages 17–24. ACM, 2007.

[6] R. K. Ando and T. Zhang. A Framework for Learning Predictive Structures from Multiple Tasks and Unlabeled Data. *The Journal of Machine Learning Research*, 6, Dec. 2005.

[7] L. Balagopalan, E. Sherman, V. Barr, and L. Samelson. Imaging techniques for assaying lymphocyte activation in action. *Nat Rev Immunol*, 11(1):21–33, 2011.

[8] R. Basri and D. W. Jacobs. Recognition using region correspondences. *International Journal of Computer Vision*, 25(2):145–166, 1997.

[9] A. Boucher, P. Hidalgo, M. Thonnat, J. Belmonte, C. Galan, P. Bonton, and R. Tomczak. Development of a semi-automatic system for pollen recognition. *Aerobiologia*, 18(3):195–201, 2002.

[10] A. Carpenter, T. Jones, M. Lamprecht, C. Clarke, I. Kang, O. Friman, D. Guertin, J. Moffat, et al. Cellprofiler: image analysis software for identifying and quantifying cell phenotypes. *Genome Biology*, 7(10):R100, 2006.

[11] R. Caruana. Multitask learning. pages 95–133, 1998.

[12] R. Chattopadhyay, Q. Sun, W. Fan, I. Davidson, S. Panchanathan, and J. Ye. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 6(4):18, 2012.

[13] A. Chu, C. Sehgal, and J. Greenleaf. Use of gray value distribution of run lengths for texture analysis. *Pattern Recognition Letters*, 11(6):415–419, 1990.

[14] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[15] A. Culotta and A. McCallum. Reducing labeling effort for structured prediction tasks. In *AAAI*, volume 5, pages 746–751, 2005.

[16] I. Czarnowski and P. Jedrzejowicz. Data reduction algorithm for machine learning and data mining. In *New Frontiers in Applied Artificial Intelligence*, volume 5027 of *Lecture Notes in Computer Science*, pages 276–285. Springer, 2008.

[17] W. Dai, Q. Yang, G. R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, page 200. ACM, 2007.

[18] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. 1:886–893, 2005.

[19] H. Daume III and D. Marcu. Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research*, 26(1):101–126, 2006.

[20] P. Domingos. A unified bias-variance decomposition for zero-one and squared loss. *AAAI/IAAI*, 2000:564–569, 2000.

[21] P. Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[22] O. Dzyubachyk, W. A. van Cappellen, J. Essers, W. J. Niessen, and E. Meijering. Advanced level-set-based cell tracking in time-lapse fluorescence microscopy. *Medical Imaging, IEEE Transactions on*, 29(3):852 –867, march 2010.

[23] E. Eaton. PhD Dissertation: Selective Knowledge Transfer for Machine Learning. *PhD Dissertation*, pages 1–156, May 2009.

[24] E. Eaton and M. desJardins. Selective transfer between learning tasks using task-based boosting. In *AAAI*. Citeseer, 2011.

[25] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[26] T. Evgeniou and M. Pontil. Regularized multi–task learning. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 109–117, New York, NY, USA, 2004.

[27] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.

[28] A. Farhadi, D. Forsyth, and R. White. Transfer learning in sign language. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, june 2007.

[29] Y. Freund and R. Schapire. A short introduction to boosting. *Japonese Society for Artificial Intelligence*, 14(5):771–780, 1999.

[30] A. Fujii, T. Tokunaga, K. Inui, and H. Tanaka. Selective sampling for example-based word sense disambiguation. *Comput. Linguist.*, 24:573–597, December 1998.

[31] J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 283–291, New York, NY, USA, 2008.

[32] W. E. L. Grimson and T. Lozano-Perez. Localizing Overlapping Parts by Searching the Interpretation Tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(4):469–482, 1987.

[33] S. Hadjidemetriou, B. Gabrielli, T. Pike, F. Stevens, K. Mele, and P. Vallotton. Detection and tracking of cell divisions in phase contrast video microscopy. *Proc. of the Third MICCAI Worshop on Microscopic Image Analysis with Applications in Biology*, 2008.

[34] A. Harpale and Y. Yang. Active learning for multi-task adaptive filtering. 2010.

[35] T. Hastie and R. Tibshirani. Classification of Pairwise Coupling. *Annals of Statistics*, 26, Apr. 1998.

[36] E. Hodneland, N. Bukoreshtliev, T. Eichler, X.-C. Tai, S. Gurke, A. Lundervold, and H.-H. Gerdes. A unified framework for automated 3-d segmentation of surface-stained living cells and a comprehensive segmentation evaluation. *Medical Imaging, IEEE Transactions on*, 28(5):720–738, May 2009.

[37] S. C. H. Hoi, R. Jin, and M. R. Lyu. Large-scale text categorization by batch mode active learning. In *Proceedings of the 15th international conference on World Wide Web*, WWW '06, pages 633–642, New York, NY, USA, 2006.

[38] K. Holt, G. Allen, R. Hodgson, S. Marsland, and J. Flenley. Progress towards an automated trainable pollen location and classifier system for use in the palynology laboratory. *Review of Palaeobotany and Palynology*, 167(3–4):175–183, 2011.

[39] A. Joshi, F. Porikli, and N. Papanikolopoulos. Multi-class active learning for image classification. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2372–2379, 2009.

[40] W. Kamoun, S. Schmugge, J. Kraftchick, M. Clemens, and M. Shin. Liver microcirculation analysis by red blood cell motion modeling in intravital microscopy images. *Biomedical Engineering, IEEE Transactions on*, 55(1):162–170, Jan. 2008.

[41] T. Kanade, Z. Yin, R. Bise, S. Huh, S. Eom, M. F. Sandbothe, and M. Chen. Cell image analysis: Algorithms, system and applications. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 374–381. IEEE, 2011.

[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[43] S. Kullback. Letter to the editor: The kullback-leibler distance. 1987.

[44] N. Kutsuna, T. Higaki, S. Matsunaga, T. Otsuki, M. Yamaguchi, H. Fujii, and S. Hasezawa. Active learning framework with iterative clustering for bioimage classification. *Nature Communications*, 3:1032, 2012.

[45] P. Li, W. Treloar, J. Flenley, and L. Empson. Towards automation of palynology 2: the use of texture measures and neural network analysis for automated identification of optical images of pollen grains. *Journal of quaternary science*, 19(8):755–762, 2004.

[46] D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[47] E. Lughofer. Hybrid active learning for reducing the annotation effort of operators in classification systems. *Pattern Recognition*, 45(2):884–896, 2012.

[48] T. Luo, K. Kramer, D. Goldgof, and L. Hall. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6:589–613, 2005.

[49] N. A. H. Mamitsuka. Query learning strategies using boosting and bagging. In *Machine Learning: Proceedings of the Fifteenth International Conference (ICML'98)*, volume 1. Morgan Kaufmann Pub, 1998.

[50] D. Mukherjee, N. Ray, and S. Acton. Level set analysis for leukocyte detection and tracking. *Image Processing, IEEE Transactions on*, 13(4):562–572, April 2004.

[51] D. B. Murphy. *Fundamentals of light microscopy and electronic imaging*. John Wiley & Sons, 2002.

[52] R. C. Nelson and A. Selinger. Large-scale tests of a keyed, appearance-based 3-d object recognition system. *Vision research*, 38(15):2469–2488, 1998.

[53] N. Nguyen, S. Keller, E. Norris, T. Huynh, M. Clemens, and M. Shin. Tracking Colliding Cells In Vivo Microscopy. *Biomedical Engineering, IEEE Transactions on*, 58(8):2391–2400, 2011.

[54] N. H. Nguyen, E. Norris, M. G. Clemens, and M. C. Shin. Rapidly adaptive cell detection using transfer learning with a global parameter. In *International Workshop on Machine Learning in Medical Imaging*, pages 209–216. Springer, 2011.

[55] N. R. Nguyen, M. Donalson-Matasci, and M. C. Shin. Improving pollen classification with less training effort. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 421–426. IEEE, 2013.

[56] E. J. Norris, N. Feilen, N. H. Nguyen, C. R. Culberson, M. C. Shin, M. Fish, and M. G. Clemens. Hydrogen sulfide modulates sinusoidal constriction and contributes to hepatic micorcirculatory dysfunction during endotoxemia. *American Journal of Physiology-Gastrointestinal and Liver Physiology*, 304(12):G1070–G1078, 2013.

[57] T. Nuzhnaya, M. Barnathan, H. Ling, V. Megalooikonomou, P. R. Bakic, A. D. A. Maidment, and A. D. Maidment. Probabilistic branching node detection using AdaBoost and hybrid local features. *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pages 221–224, 2010.

[58] J. Pan, T. Kanade, and M. Chen. Learning to detect different types of cells under phase contrast microscopy. *Microscopic Image Analysis with Applications in Biology*, 2009.

[59] S. J. Pan and Q. Yang. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, Oct. 2010.

[60] P. Rai, A. Saha, H. Daumé III, and S. Venkatasubramanian. Domain adaptation meets active learning. *Proceedings of the NAACL HLT 2010 Workshop on Active Learning for Natural Language Processing*, pages 27–32, June 2010.

[61] N. Ray and S. Acton. Motion gradient vector flow: an external force for tracking rolling leukocytes with shape and size constrained active contours. *Medical Imaging, IEEE Transactions on*, 23(12):1466–1478, Dec. 2004.

[62] N. Ray, S. Acton, and K. Ley. Tracking leukocytes in vivo with shape and size constrained active contours. *Medical Imaging, IEEE Transactions on*, 21(10):1222–1235, Oct. 2002.

[63] M. Rodriguez-Damian, E. Cernadas, A. Formella, M. Fernandez-Delgado, and P. D. Sa-Otero. Automatic detection and classification of grains of pollen based on shape and texture. *Systems, Man, and Cybernetics, Part C, IEEE Transactions on*, 36(4):531 –542, July 2006.

[64] H. A. Rowley, S. Baluja, T. Kanade, et al. *Human face detection in visual scenes*. Carnegie-Mellon University. Department of Computer Science, 1995.

[65] N. Roy and A. McCallum. Toward optimal active learning through monte carlo estimation of error reduction. *ICML, Williamstown*, pages 441–448, 2001.

[66] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):579–586, 1988.

[67] K. Sarinnapakorn and M. Kubat. Combining Subclassifiers in Text Categorization: A DST-Based Solution and a Case Study. *IEEE Transactions on Knowledge and Data Engineering*, 19(12):1638–1651, 2007.

[68] R. E. Schapire. The boosting approach to machine learning: an overview. *Nonlinear Estimation and Classification*, 171:149–171, 2003.

[69] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. pages 1651–1686. JSTOR, 1998.

[70] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-rated Predictions. *Machine Learning*, 37(3), Dec. 1999.

[71] J. Schmidhuber. On learning how to learn learning strategies. Technical report, Technische Universitt Mnchen, 1995.

[72] T. D. Seeley. *Honeybee democracy*. Princeton Univ. Press, 2010.

[73] B. Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66):11, 2010.

[74] L. Shapiro and R. Haralick. Computer and robot vision. *Reading: Addison-Wesley*, 8, 1992.

[75] R. Souvenir, J. Kraftchick, S. Lee, M. Clemens, and M. Shin. Cell motion analysis without explicit tracking. *Computer Vision and Pattern Recognition (CVPR)*, pages 1 – 7, 2008.

[76] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. pages 1–9, 2015.

[77] S. Thrun and T. Mitchell. Learning one more thing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI)*, San Mateo, CA, 1995. Morgan Kaufmann.

[78] S. Tong and D. Koller. Support vector machine active learning with applications to text classification. *J. Mach. Learn. Res.*, 2:45–66, March 2002.

[79] D. Toomre and J. Bewersdorf. A new wave of cellular imaging. *Annual review of cell and developmental biology*, 26:285–314, 2010.

[80] G. Tur, D. Hakkani-Tür, and R. E. Schapire. Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45(2):171–186, Feb. 2005.

[81] S. Vijayanarasimhan and K. Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *International Journal of Computer Vision*, 108(1-2):97–114, 2014.

[82] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.

[83] X.-Z. Wang, J.-H. Yan, R. Wang, and C.-R. Dong. A sample selection algorithm in fuzzy decision tree induction and its theoretical analyses. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference on*, pages 3621 –3626, oct. 2007.

[84] P. Wu and T. G. Dietterich. Improving svm accuracy by training on auxiliary data sources. In *Proceedings of the twenty-first international conference on Machine learning*, page 110. ACM, 2004.

[85] C. Xu and J. Prince. Snakes, shapes, and gradient vector flow. *Image Processing, IEEE Transactions on*, 7(3):359 –369, Mar 1998.

[86] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *Computer Vision and Pattern Recognition (CVPR) IEEE Conference on*, pages 1855 –1862, Jun 2010.

[87] Z. Yin, R. Bise, and M. Chen. Cell segmentation in microscopy imagery using a bag of local Bayesian classifiers. *Biomedical Imaging: From Nano to Macro, IEEE International Symposium on*, pages 125–128, 2010.

[88] A. Yla-Jaaski and N. Kiryati. Adaptive termination of voting in the probabilistic circular Hough transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(9):911–915, 1994.

[89] C. Zhang and T. Chen. An active learning framework for content-based information retrieval. *IEEE Transactions on Multimedia*, 4(2):260–268, 2002.

[90] Z. Zhang, R. Deriche, O. Faugeras, and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. *Artificial intelligence*, 78(1):87–119, 1995.