Local-Feature Generic Object Recognition with Application to Insect-Species Identification

Natalia Larios Delgado

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

Doctor of Philosophy

University of Washington

2010

Program Authorized to Offer Degree:
Department of Electrical Engineering

University of Washington
Graduate School

This is to certify that I have examined this copy of a doctoral dissertation by

Natalia Larios Delgado

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Chair of the Supervisory Committee:

_____

Linda G. Shapiro

Reading Committee:

_____

Linda G. Shapiro

_____

Thomas G. Dietterich

_____

Maya Gupta

Date: _____

In presenting this dissertation in partial fulfillment of the requirements for the doctoral degree at the University of Washington, I agree that the Library shall make its copies freely available for inspection. I further agree that extensive copying of this dissertation is allowable only for scholarly purposes, consistent with "fair use" as prescribed in the U.S. Copyright Law. Requests for copying or reproduction of this dissertation may be referred to Proquest Information and Learning, 300 North Zeeb Road, Ann Arbor, MI 48106-1346, 1-800-521-0600, to whom the author has granted "the right to reproduce and sell (a) copies of the manuscript in microform and/or (b) printed copies of the manuscript made from microform."

Signature_____

Date_____

University of Washington

Abstract

Local-Feature Generic Object Recognition with Application to Insect-Species Identification

Natalia Larios Delgado

Chair of the Supervisory Committee:
Professor Linda G. Shapiro
Department of Computer Science and Engineering

Biological monitoring of stream water quality is a time-consuming and expensive activity for environmental protection and sustainable development. It requires intensive hand labor from experts to identify arthropod species known for their ability to reflect the status of their environment. The three generic recognition methods proposed in this dissertation are designed with the goal of developing image-based automated high-throughput insect-species identification with application to water quality assessment. These methods consist of a local-feature extraction and aggregation step to describe every image and a prediction step using this descriptor. The first method is based on the bag-of-features approach and uses unsupervised feature types and species-specific dictionaries to map sets of local features into histogram descriptors. These descriptors are concatenated and input to a logistic model tree (LMT) to discriminate among four stonefly species. The second method is a combination of efficient feature extraction and quantization using Haar-like features and random forests. The descriptors created by this process contain spatial information roughly correlated to specimen parts and are utilized by a spatial-pyramid kernel classifier for species identification. The final recognition method combines the classification scores of multiple types of local features while retaining their spatial information in a single spatial histogram of score accumulations. A stacked spatial support vector machine (SVM) is applied to these histograms in experiments with a set of 29 species matching those of a real biomonitoring task. The evaluation experiments performed demonstrate that automated insect identification is viable and can be both efficient and effective.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

GLOSSARY

AFFINE TRANSFORMATION: a transformation between two vector spaces consisting of a linear transformation that contains zero or more rotation, scaling or shear transformations followed by a translation.

BIOMONITORING: assessment of a set of environment variables (i.e. water quality, pollution) by population measurement of specific field-collected specimens.

SALIENCY DETECTOR: a feature detector whose measure of saliency is the Shannon entropy of gray-level image regions. It selects salient regions that present entropy peaks by comparing each image region to its neighbors in scale-space.

HAAR-LIKE FEATURE: an efficient image feature for object detection and recognition problems. Each such feature is defined by an image mask with positive and negative coefficients in rectangular patterns similar to the Haar wavelet basis functions.

HESSIAN MATRIX: the square matrix of second-order partial derivatives of a function. The Hessian matrix is commonly used for expressing image processing operators in computer vision by representing each pixel value as a function with a vector input of pixel coordinates.

HESSIAN-AFFINE DETECTOR: an interest operator that finds corner features invariant to affine image transformations. It employs local extrema of the trace and determinant of the Hessian matrix. These values are stable to noise while avoiding detecting areas surrounding edges.

HISTOGRAM INTERSECTION: is an approximate probability density distance between histograms that computes the sum of the minimum value of each of the corresponding bins of a pair of histograms.

IMAGE SCALE-SPACE: a theoretical representation composed by image locations combined at different scales. This means that you can determine your position in this space with the pixel coordinates and image scale. This terminology has it roots from the concept space-time in physics.

INFORMATIVENESS: desirable attribute of the regions found by an interest operators. The regions found should contain image information relevant to the computer vision task being performed, in the case of this work, the information should discriminate between different object-classes.

PCBR DETECTOR: Principal Curvature-Based Region detector. This operator performs segmentation of the curvature image to find regions with similar curvilinear structure.

PRECISION: the ratio of the number of true positives to the sum of true positives and false positives. This is a per class classification performance measure.

RECALL: is the ratio of the number of true positives to the total number of elements that actually belong to the class (true positives plus false negatives).

RELIABILITY: desirable trait of the regions output by an interest operator. The same physical region should be found by the operator in images taken under different lighting conditions, viewing angles, and object pose or orientation.

SPATIAL-PYRAMID KERNEL: a fast-to-evaluate Mercer kernel function mapping local features to multi-resolution spatial histograms. The multi-resolution histogram are built by binning features into discrete image regions of increasingly larger size. The core function to evaluate feature matching is the histogram intersection distance.

ACKNOWLEDGMENTS

I wish to express sincere appreciation to the Electrical Engineering Department and to the Computer Science and Engineering Department, where I have had the opportunity to work in the interesting subject of computer vision. I also state my gratitude for the financial support of the NSF and the Mexican SEP and CONACYT agencies. Their sponsorship has enabled me to pursue this activity. I wish to thank to my adviser Linda Shapiro as well as project PI Tom Dietterich for their guidance, knowledge, and support through the research work that allowed the creation of this dissertation. I also give my utmost appreciation to the other members of the BugID project from whom I obtained the image data to perform these experiments, including the entomologists who labeled all the insect specimens and the mechanical engineering students who imaged them. I also wish to acknowledge the love, care, and understanding of my friends and family, my parents, and my soulmate Ishtar.

DEDICATION

To my friend Javier, to my parents José Enrique and Rosa Angélica, and to Ishtar

for none of this would have been possible without all of you

Chapter 1

INTRODUCTION

One of the main areas of research in computer vision is generic object recognition. Its goal is to understand and implement, in machine vision systems, the human ability of recognizing the abstract class to which a previously unseen object belongs. This classification problem requires identifying the abstract class or category that best fits an object instance contained in an image. Generic object recognition is an interesting problem; it requires the design of recognition methods capable of abstracting the possible variations present in instances of the same class, while extracting discriminative features that are invariant to common image transformations and useful for identification. Image classification is a highly related problem where the goal is to label an image from a set of predefined tags, scene types or classes depending on its content.

There is a large body of literature [15, 21, 29, 40, 32, 65] showing the sucessful application of the bag-of-keywords approach. In this approach, local-feature data is extracted from the available images and characterized by computing a feature vector called a descriptor. A visual dictionary is created by clustering a subset of the training data in the descriptor space, creating clusters that define dictionary keywords. This dictionary is then used to map the region descriptors of a novel image into keywords, and therefore, to map the bag of features for an image into a bag of keywords that is used to construct a representation of the image. Histograms of these keywords have been successfully used as intermediate image representations. Once an intermediate representation is constructed, the reminder of the training data is employed to train a classifier to recognize object classes.

In this dissertation, we extend the state-of-the-art recognition approaches that use a set of random trees to discriminatively structure the information provided by the feature data obtained from the training and testing images. The random trees stage replaces the unsupervised cluster model learning and cluster assignment used in the visual dictionary methods. Two different image representations are introduced in this work: (1) an image-patch count histogram whose bins are associated with random-forest tree nodes and (2) a spatial histogram of classification score accumulations.

Local feature extraction is a mainstay of this methodology. The main benefit of the local-feature approach is that it takes away the design complexity and computation time from modeling the complete appearance of the objects from each class. Instead, it is directed to finding the most discriminative features, either because of their repeatability or because of their invariance, and creating the best margins to separate instances from different classes. Since it is based on the use of local features, it greatly reduces the concern for partial occlusion and for change in pose in articulated objects.

1.1 Motivating Problem

Biomonitoring is the assessment of the status and trend of the environment using counts of a set of defined species known for their susceptibility and capacity to accumulate the effects of environmental changes over time. One of the most widely used biological monitoring metrics for water quality assessment is the population count of specimens from the insect orders Ephemeroptera (mayflies), Plecoptera (stoneflies), and Trichoptera (caddisflies), often abbreviated as EPT. Species within these orders vary greatly in their susceptibility to pollution and so are robust indicators of stream water quality. Identifying and enumerating "EPT" samples is a labor-intensive and time-consuming process, and generally requires expert entomologists performing manual classification. For the experiments of Chapter 3 and Chapter 4, a subset of Plecoptera specimens (stoneflies) was selected as dataset of the initial evaluation. In Chapter 5, a set of species was selected to realistically match a complete EPT sample to evaluate the feasibility of automated monitoring.

1.2 Three Approaches

Our initial research tackled stonefly insect species identification as a generic object recognition problem with the goal of creating inexpensive computer vision methods for automated population counts of insects and other small arthropods. For this specific task, it is possible to control the image background, the lighting conditions and (to some extent) the insect orientation. Thus far we have successfully applied two different approaches. In our first method, visual codebooks are created by learning a Gaussian mixture model (GMM) over the descriptor space of the detected regions, creating a specific codebook for each detector and species pair. Each image is then represented by a feature vector computed as histograms with counts of the descriptors mapped to the learned dic-

tionary "keywords". The image representations obtained from each dictionary are combined into a concatenated feature histogram (CFH) that works as image descriptor for classification. The image classifier ensemble is trained over the feature vectors from the training set. This method obtained satisfactory identification results between four stonefly species, which demonstrated that this was possible to do via image classification methods. One of the drawbacks of this method was the use of existing interest region detectors as part of local-feature extraction method. Many of the existing detectors are computationally expensive, because they perform extensive scale-space search. Additionally, it is often necessary to employ a large number of regions with computationally-expensive high-dimensional descriptors to offset the lack of discriminative information for the current task.

Our Haar random forest (HRF) method applies discriminative learning techniques to the local-feature extraction process by evaluating image patches in a grid array with a random forest (RF) that contains Haar-like features [36] as splitting decision functions. This method is able to capture discriminative information and its spatial distribution while reducing feature extraction time due to the intrinsic efficiency of tree traversal and the constant computation time of the Haar-like feature. In this case, the depth levels of the RF tree form a semantically meaningful implicit cluster hierarchy instead of the clusters obtained by traditional clustering algorithms. The feature space partitioning provided by the Haar random forest (HRF) is employed to map the extracted image patches into assignment counts that form each histogram vector representing the image. This combination of powerful discriminative dictionary learning with an efficient low-level feature extraction process is one of the main contributions of the current method. Another relevant element of the learning algorithm is the pairing of the Haar-like feature search at every tree node with the image feature band on which it will be evaluated. Each image patch is represented by the CIELab color channels and by image planes containing gradient counts of a particular orientation. The histogram vector representing each image is evaluated by a pyramid-match kernel SVM that considers patch distributions in both feature and image space. This method achieves large time gains in comparison with previous identification methods that use existing local features [32, 40] while attaining state-of-the-art classification performance [40].

The stacked spatial-pyramid (SSP) kernel method combines multiple feature types in a discriminative stacking framework that replaces the cluster-based dictionaries of the bag-of-features approach. A first layer of tree classifiers aggregates information from different types of local fea-

tures in spatial histograms where their classification scores are accumulated. These spatial score histograms correlate the feature scores accumulated on the image region bins matching the insect body-parts of the specimens in a normalized orientation. This approach is applied to an insect-species identification problem on a data set composed of 29 species that belong to the three most common insect orders employed for biological stream-water quality monitoring. A diverse set of features is a very important factor of the SSP method given the large intra-species variations and the small inter-species differences of some subgroups of species. The spatial histogram of scores is a simple but powerful image representation allowing the combination of diverse feature types that complement each other. These histograms are a suitable input for the spatial pyramid-kernel SVM classifier that is able to use the discriminative spatial structures present in different specimen parts while remaining invariant to small changes in pose.

1.3 Dissertation Overview

The overview of the rest of this dissertation is as follows: Chapter 2 discusses related work in image feature extraction, discriminative codebook learning techniques, and kernel classifier methods for recognition. This chapter also covers existing systems for automated insect recognition. Chapter 3 reviews our initial stonefly identification work, which employs a bag-of-keywords method with a decision tree forest as classifier. Chapter 4 describes our work in efficient image classification that combines the local-feature extraction stage with the quantization stage employing discriminatively learned dictionaries. Our binary stonefly identification experiments and the results obtained by applying a spatial-kernel SVM classifier are also described. In Chapter 5, a stacking classifier framework that allows the use of spatial information and combines multiple image feature types is applied to an experiment performing realistic biomonitoring stream water assessment. Finally, Chapter 6 discusses our recognition work and the results obtained in our experiments.

Chapter 2

RELATED WORK

Generic object recognition has evolved from the use of simple features, such as line segments and regions, and simple algorithms, such as graph matching, to the use of rich features and complex machine learning algorithms. Current methods are able to perform class recognition tasks in images taken in non-controlled environments with variability in object position and orientation, with cluttered background, and even with partial occlusion. A typical classification methodology begins with detection of interesting points or regions [47], encodes them to obtain some form of descriptor [38] vector, and uses these descriptors (and sometimes their spatial locations) for classification. This performance gain obtained in the last decade can be attributed to the use of invariant representations of local features and to the intelligent use of aggregation methods to generate image representations later used by a classification method based on machine learning. These approaches compare favorably with global-feature approaches that had previously widespread application such as [49, 54].

2.1 Methods for Local-Feature Extraction

Local image features have two main advantages over global features: their intrinsic invariance to image translation transformations and robustness to modest degrees of pose change and partial occlusion. The informativeness of a set of local features from an image is currently achieved with two approaches: the use of region or interest-point detectors, where feature descriptors are computed on those regions, and an expensive descriptor vector computation following a uniform pattern such as a grid, oftentimes using methods that allow for efficient repeated computation.

For local-feature methods with a detection stage, the points or regions must be reliably detected. Image regions representing the same object or scene part should be found by the detector in images taken under different lighting conditions, different viewing angles, and with changes in object pose or orientation. Furthermore, for generic object recognition, the detected regions should be as robust possible, within the frame of this low-level stage, to variations from one object instance to another

of the same generic class. Finally, given that the number of regions obtained is expected to be much lower than the number of pixels in the image, the regions must be informative for recognition—that is, they must capture properties that discriminate between objects in different classes.

Current region detectors can be divided into two categories: intensity-based detectors and structure-based detectors. The intensity-based include the Harris-corner detector [27], the intensity extrema-based region detector (IBR) [57], the entropy-based region detector [30], the scale-invariant feature transform (SIFT) [38], and its efficient approximation, the Speeded Up Robust Features (SURF) [5]. Among the detectors in this category, special effort has been put into the development of affine-invariant detectors capable of achieving robustness to moderate changes in viewing angle, including the Hessian-affine and Harris-affine detectors [43, 44], and the maximally stable extremal regions (MSER) [41]. Structure-based detectors include the edge-based region detector (EBR) [58], the scale-invariant shape features (SISF) [28], and the ellipse feature detector [14].

The methods that obtain local features without region detectors employ random sampling of image locations or compute descriptors following some uniform pattern such as a grid. This type of extraction method proves particularly useful when the image background provides contextual information helpful to identifying the object-class of the instance present in the image. Among the feature descriptors used with this approach, the Haar-like features [59] and the Histogram of Oriented Gradients (HOG) [16] deserve particular mention due to their specific design for efficient repeated computation in a grid pattern thanks to caching. The grid pattern approach [34, 37] has also been applied using much more computationally expensive descriptors, such as SIFT [38].

2.2 Bag-of-Features Approach

The recognition research presented in this dissertation can trace its roots to the bag-of-feture approach. In this approach, histograms are used as the method to aggregate multiple local features as a descriptor for object and scene classification tasks. The standard method for this extensively used approach in computer vision is to create a "visual" dictionary from some of the available region description data, generally utilizing a clustering procedure that acts as a quantization stage. Each resulting cluster defines a "keyword" and these keywords are collected into a codebook or dictionary [15, 21, 29]. The dictionary can then be applied to map each region descriptor into a keyword,

and therefore, to map the bag of features into a bag of keywords. A classifier is trained to recognize an object category from the bag of keyword assignments. Given a new image to classify, the process of finding interesting regions, representing them as region descriptor vectors, mapping those to keywords, and classifying the resulting bags of keywords is carried out. While the new image regions are extracted, they are processed in an unordered way with no consideration for the spatial relationships between them.

Of the three generic object recognition methods put forward, the method described in Chapter 3 is heavily based on this approach, as it was the first to be developed. The histogram bins are associated with clusters learned in an unsupervised way. The subsequent methods continue to employ histograms as an image representation, but the "keywords" represented by the bins are discriminatively selected and, for the method of Chapter 5, the bins accumulate local-feature classification scores instead of simple local-feature counts. These image representations can thus be said to be based on discriminative codebooks.

2.3 Discriminative Code-books

Recognition methods such as Fergus *et al*. [21], Jurie and Triggs [29], and Lazebnik et al. [34] among many demonstrate the successful application of the bag-of-features approach with unsupervised dictionaries. Nevertheless, this approach has several possible drawbacks [40] such as *ad hoc* dictionary parameter tuning, information loss due to quantization, and the possible lack of discriminative power of high-probability cluster regions. These negative aspects are even more noticeable in challenging classification tasks with large extra-class similarities such as those found in the species-identification problems that motivated this work.

One approach to handle the bag-of-features shortcomings is to use semi-supervised methods that select the most discriminative clusters. Zhang and Dietterich [67] apply relevant component analysis to improve the discriminative power of visual keywords of unsupervised dictionaries. Winn *et al*. [62] merge a large set of clusters initially obtained by the K-means algorithm to obtain a final dictionary with increased discriminative power, and Yang and Jurie [64] use Boolean features created by a boosting algorithm that aims to merge codebook generation and classifier training.

Methods based on decision-tree ensembles (such as random forests [10]) are employed to gen-

erate discriminative codebooks. The feature-space partitions obtained by decision trees have been used as codebook entries in recent work [23, 40, 45, 53, 63]. The use of decision-tree ensembles also has the benefit of speeding up keyword assignment compared to traditional cluster assignment. Moosman *et al*. [45] used an extremely randomized forest to avoid clustering algorithms while learning visual codebooks. Shotton's [53] method uses random forests for the generation of so-called texton features for recognition and pixel-level object segmentation. Martínez *et al*. [40] employ the learning technique of "stacking" that can combine different feature types and employs a second, "stacked" layer of random forest classifiers. They propose a new scoring method for random-forest prediction using the class distribution of evidence that arrives at the leaves during training. Other computer vision areas in which random forests have been used for feature-space quantization are detection and segmentation. Winn [63] combined a probabilistic random field with potentials generated by a random forest to segment partially-occluded objects reliably.

In detection tasks, the basic aspects of the Hough transform [19], using votes in a quantized parameter space to search for adequate parameter values, have been successfully combined with the prediction mechanism of the random forests. Because of its accumulative nature over local features, the Hough transform handles the complexity of searching over parameters such as pose and scale in a robust way. This robustness can be increased by employing a probabilistic framework such as Ballard's [4] generalized Hough transform. Gall and Lempitsky's [23] Hough random forest method detects objects with a generalized Hough transform learned discriminatively. They trained a class-specific random forest so that every image patch appearance casts a probabilistic vote for the possible location of the object centroid.

2.4 Kernel Methods in Vision

There exist several frameworks and specialized kernels that have demonstrated the applicability of kernel methods for visual classification and retrieval tasks with significant results. Arbuckle *et al*. [2] developed kernels using local image features to successfully perform image classification for insect-species identification using a Support Vector Machine (SVM) classifier [52]. The local-feature Mercer kernel [11] is a similarity measure between a pair of images computed using simpler kernels as the sum of the maximum similarity measure of each feature descriptor against the descriptors

from the other image. This kernel has the advantage of allowing any existing kernels to be employed, but it is computationally very expensive. The pyramid match kernel [26] performs a very efficient comparison of sets of local features by finding matches on increasingly finer multi-dimensional histograms using weighted histogram intersections. A modified version of this idea combined with clustering applied to the bag-of-features approach can be employed to create a spatial pyramid [34] that employs global spatial information to classify image scenes. Other suitable image distance measures can be turned into kernels in order to apply an SVM to image classification. Both the χ^2 histogram distance and the earth mover's distance for signatures can be turned into suitable kernels that aggregate local-feature information for vision tasks as shown by Zhang *et al*. [65]. Based on this idea, several researchers have applied a kernel approach combined with local features for vision-based shape retrieval methods. The spatial-pyramid kernel has been shown to work for shape retrieval tasks [6]. The same basic idea of local features has been successfully transplanted from the 2D image domain to 3D mesh data. The so-called heat kernel [48] obtains the same local-feature benefits in the 3D shape domain compared to using a global representation as its counterpart in the image domain.

2.5 Automated Insect Species Identification

Other research groups have developed systems that apply computer vision methods to discriminate among a defined set of insect species. Insect classification is an important image analysis application that can be used to evaluate the ecology of a region. These existing approaches often rely on manual manipulation and image capture of each specimen. Some systems also require the user to manually identify key image features.

2.5.1 Automated Bee Identification System (ABIS)

The ABIS system [2] performs identification of bees based on features extracted from their forewings. Each bee is manually positioned and a photograph of its forewing is obtained in a standard pose. From this image, the wing veination is identified, and a set of key wing cells (areas between veins) is determined. Then geometric features (e.g. lengths, angles, and areas) and appearance features are computed from small image patches that have been smoothed and normalized. Classification

is performed using Support Vector Machines and Kernel Discriminant Analysis. This project has obtained good results, even when discriminating between bee species that are known to be hard to classify. It has also overcome its initial requirement of expert interaction with the image for feature extraction; although it still has the restriction of complex user interaction to manipulate the specimen for the capture of the wing image. The ABIS feature extraction algorithm incorporates prior expert knowledge about wing veination. This facilitates the bee classification task; but makes it very specialized.

2.5.2 Digital Automated Identification System (DAISY)

DAISY [46] is a general-purpose identification system that has been applied to several arthropod identification tasks including mosquitos, biting midges, ophionines (parasites of lepidoptera), parasitic wasps, and hawk-moths (Sphingidae). Unlike our method, DAISY requires user interaction for image capture and segmentation, because specimens must be aligned in the images. The basic algorithm of DAISY is based on the methods for human face detection and recognition via eigen-images [56]. In its latest implementation, its classifier is based on a random n-tuple classifier (NNC) [39] and plastic self organizing maps (PSOM). It employs a correlation algorithm called the normalized vector difference (NVD) algorithm. DAISY is capable of handling hundreds of taxa with efficient identifications. The use of NNC often imposes the requirement of having many instances of each species. This is specially true in species with high intra-class variability that might require many training instances to cover the whole range of appearance. Additionally, the requirement of user interaction hampers DAISY's throughput and makes its application infeasible in tasks where the identification of large samples is required.

2.5.3 Species Identification Automated and Web Accessible (SPIDA-web).

SPIDA-web [18] is an automated identification system that applies neural networks for spider species classification of the Trochanteriidae family from wavelet-encoded images. The SPIDA-web's feature vector is built using images of the spider's external genitalia from a subset of components of the wavelet transform using the Daubechies 4 function. This method also has the drawback that the spider specimen has to be manipulated by hand, and the image capture, preprocessing, and

region selection also require direct user interaction. The images are oriented, normalized, and scaled into a 128×128 square prior to feature extraction. The specimens are classified in a hierarchical manner, first to genus and then to species. The classification engine is composed of a trained neural network for each species in the group. Reported results for female specimens indicate that SPIDA is able to classify images to genus level with 95-100% accuracy; though results at the species-level still leave room for improvement.

2.5.4 Insect Species Identification for Stream Quality Assessment

One relevant application of automated species identification is the assessment of stream water quality. Thiel *et al*'s [55] work on biological water quality assessment employs basic pattern recognition methods on algae specimens [55]. Arthropod identification from images has been the subject of extensive work [32, 68, 33, 40] with the same stream assessment goals. The methods used in this work rely on combining multiple low-level feature detectors and on encoding them as SIFT descriptors [38]. In the Larios approach [32], an occurrence histogram of EM cluster assignments was generated from the SIFT vectors and used in classification by a bagging ensemble [9] of decision trees. This method is discussed in more detail in Chapter 3, where the feasibility of automated stonefly species identification with existing computer vision and machine learning methods is evaluated. In the work of Martínez *et al*., evidence trees [40] are employed in the quantization stage. A stacked ensemble of decision trees is then employed on the resulting concatenated histograms of evidence. This method was used to classify a dataset containing nine different stonefly species (*STONEFLY9*). In Zhang *et al*. [68], multiple non-redundant codebooks were used to map the local features obtained from the images. In this series of studies regarding water quality assessment, the progression of recognition methods from simple pattern recognition approaches to unsupervised and later discriminative dictionaries can be seen.

The recognition work for biological monitoring of water quality described above suffers from being computationally intensive in the local-feature extraction stage (both region or interest point detection and descriptor computation). The method introduced in Chapter 4 speeds up feature extraction by combining it with discriminative quantization using efficient Haar-like features as splitting functions at every node. Additionally, these method only included evaluation on subsets of

the species commonly utilized in EPT surveys. In [33, 40], experiments on nine stonefly taxa (Plecoptera) obtained very low classification errors. As part of this dissertation, the method described in Chapter 5 includes recognition experiments on a set of 29 species which more closely matches the possible samples for this task.

2.5.5 Other Species Identification Methods

As part of the discussion of relevant automated identification, we list a brief description of other work in this area. Wen *et al*. [60] used an approach combining both global and local image features to automatically differentiate between beneficial and damaging insect species for orchard pest management. In [2, 24, 42], generic object recognition methods were applied to the recognition of winged insects. The drawback of these methods is that they require user intervention for positioning and imaging the wings of the specimens, thus failing to achieve high-throughput identification. There exist other automated systems that perform species identification without using image data of any kind. Some techniques use the wing-beat frequencies of flying insects [35] or acoustic data obtained from the sound of insects [12]. These two systems employ commonly used machine learning methods for acoustic signal classification. As part of the trend to develop mobile applications, an automated system for on-field identification of botanical species [61] using leaf shape and venation patterns was implemented.

Chapter 3

UNSUPERVISED DICTIONARIES: CONCATENATED HISTOGRAMS OF FEATURES

The generic object recognition method described in this chapter is a modified bag-of-features approach. It employs concatenated histograms of local features as image descriptors for automated stonefly identification. Stoneflies belong to the Plecoptera insect order; they are a good initial choice to evaluate the feasibility of image-based automated species identification because of their capability to measure changes in the environment. We assess the performance of the concatenated feature histogram (CFH) method by performing multi-class and binary classification experiments on a set containing four different species of stonefly.

Figure 3.1 shows some of the images collected for these experiments. The first step in this methodology is the extraction of interesting regions from the training images. Once extracted, the regions from several different detectors are all represented by SIFT descriptors [38] that capture their appearance. The training set is divided into two sets of images: one for defining a data dictionary and the other for training the classifier. The first set is input to an EM clustering algorithm that produces a set of Gaussian components. The components are used with the second set to produce a fixed-length feature vector for each image. The feature vectors are then employed to train the classifier and to encode test images. Figure 3.2 gives an overall picture of the computation of the histograms of local features of our approach.

3.1 Dictionary Learning and Histogram Vector Computation

For this method, we apply multiple interest point detectors. For a given image and interest operator, a fixed-length vector summarizes the region output by that operator in that image. Those vectors are combined in the final classifier learning stage; the only limitation is the increase in computation time.

Three region detectors were applied to each image: the Hessian-affine detector [44], the Kadir

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

Figure 3.1: Example images of different stonefly larvae: (a) Calineuria, (b) Doroneuria, (c) Hesperoperla, and (d) Yoroperla.

saliency detector [30], and the PCBR detector [66]. We use the Hessian-affine detector implementation available from Mikolajczyk with a detection threshold of 1000. For the Kadir saliency detector, we use the binary code made available by the author (See Appendix B for the binary locations) and set the scale search range to between 25 and 45 pixels with the saliency threshold at 58. All the parameters for the two detectors mentioned above are obtained empirically by modifying the default values in order to obtain reasonable regions. For the PCBR detector, we operate at three scales with $\sigma_D = 1, 2, 4$. The higher value in hysteresis thresholding is 0.04. The two ratios applied to compute the lower thresholds are 0.2 and 0.7—producing low thresholds of 0.008 and 0.028, respectively. Each detected region is represented by a SIFT vector, using Mikolajczyk's modification to the binary code distributed by Lowe.

The use of local feature detectors poses some particular challenges for object recognition tasks. The vector descriptors must be high-dimensional in order to be discriminative, but the variable number of regions must be described by a fixed-length vector. These two issues, which always appear in the bag of features approach; can be solved by employing a feature dictionary. The feature dictionary works as the necessary dimensionality reduction step. It also handles the necessary conversion to comparable CFH vectors of the same length.

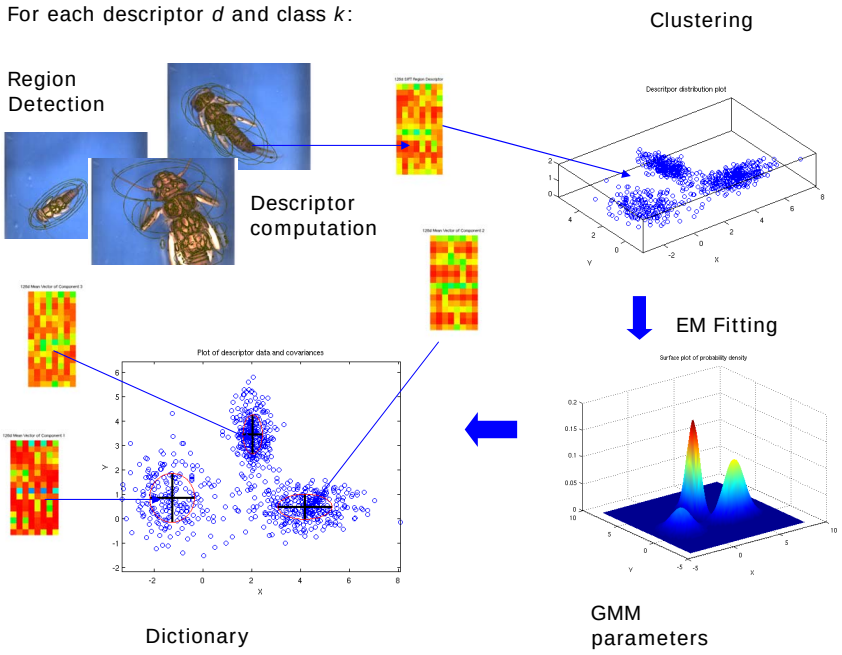

Figure 3.2: Dictionary learning procedure for the concatenated feature histogram (CFH) method.

Table 3.1: Classifier training procedure. B is the number of bootstrap iterations (i.e., the number of classifiers to learn in the ensemble).

Training

Let $T = \{(H_i, y_i)\}, i = 1, \ldots, N$ be the set of N training
examples where H_i is the concatenated histogram for
training image i and y_i is the corresponding class label.
For bootstrap replicate $b = 1, \ldots, B$

Construct training set T_b by randomly sampling N examples with replacement from T.
Let $Classifier_b$ be learned from T_b.

3.2 Classifier Learning

Object recognition can be considered as an image labeling procedure that can be performed by a classifier. The ability to compute the feature vector representation of novel images enables us to employ state-of-the-art classifiers in labeling tasks. Classifier training and class prediction for new images can be performed over these representations with an ensemble of simple classifiers instead of specialized classifiers that deal with the whole image. Table 3.1 contains the pseudo-code description of our ensemble training procedure. We believe that the possible information loss, as well as the restriction on the use of prior knowledge, due to the use of this intermediate representation, are offset by the gains in generality and robustness provided by the feature vector representation.

3.2.1 Ensemble Learning: Bagging

Ensemble learning methods employ a set of classifiers whose outputs are typically combined by (weighted or unweighted) voting. Ensemble methods are meta classification algorithms that have been empirically found to perform better than the single classifiers of which they are composed. A statistical and machine learning point of view about why ensemble classifiers can improve results is found in [17].

Bagging [9] is a general method for constructing an ensemble of classifiers. The bagging learn-

ing procedure is as follows. Given a set T of labeled training examples and a desired ensemble size B, B bootstrap replicate training sets T_b, $b = 1, \ldots, B$ are constructed. Each bootstrap replicate is a training set of size $|T|$ constructed by sampling uniformly with replacement from T. The learning algorithm is then applied to each of these replicate training sets T_b to produce a classifier LMT_b. To predict the class of a new image, each LMT_b is applied to the feature vector representation of the new image and the predictions vote to determine the overall classification.

There has been empirical proof [7] that bagging works to improve the results by reducing the variance of the base classifiers that are being employed. This characteristic of bagging couples well with the LMT base classifiers. LMTs are relatively unbiased classifiers because of their decision mechanism at the nodes [31]. The EM procedure employed to learn the feature dictionary also adds variation to the feature generation. These characteristics make bagging a suitable ensemble algorithm for our approach.

3.2.2 Base Classifier: Logistic Model Trees

The base classifiers of our approach are learned with the logistic model tree (LMT) method developed by Landwehr, Hall, and Frank [31]. The LMT has the structure of a decision tree where each leaf node contains a logistic regression classifier. Each internal node tests the value of one chosen feature from the feature vector against a threshold and branches to the left child if the value is less than the threshold and to the right child if the value is greater than or equal to the threshold.

LMTs are constructed by the standard top-down divide-and-conquer method employed by CART [8] and C4.5 [50]. At each node in the decision tree, the algorithm must decide whether to introduce a split at that point or make the node into a leaf (and fit a logistic regression model). This choice is made by a one-step lookahead search in which all possible features and thresholds are evaluated to see which one will result in the best improvement in the fit to the training data. In standard decision trees, efficient purity measures such as the GINI index or the information gain can be employed to predict the quality of the split. In LMTs, it is instead necessary to fit a logistic regression model to the training examples that belong to each branch of the proposed split. This is computationally expensive, although the expense is substantially reduced by a clever incremental algorithm based on logit-boost [22]. Benchmarking experiments show LMTs achieve robust performance [31].

The LMT base classifiers are well suited for the feature vectors generated by our approach. The attributes from the feature vector are counts of region descriptors that are assigned to the closest Gaussian component. These components work as appearance prototypes over the SIFT descriptor space of the detected regions found in the images from a class. One of the main components of the LMT induction algorithm is the attribute splitting procedure, which searches for the particular prototype and assignment-count value that best discriminates between the different classes. Another interesting element of the LMT induction procedure, which is related to the nature of the attributes, is the logistic-regression for classification. The splitting procedure is stopped when it is deemed to be non-beneficial; then a logistic-regression is performed at each leaf that only considers the attributes in the path to it. The regression considers the region assignment values for the final classification prediction. This operation at the leaves helps to reduce over-fitting by smoothing possible errors because of incorrect "keyword" assignment and unreliable region detection.

3.3 Instance Classification Procedure

The classification procedure is described in Table 3.2. The steps in the class prediction algorithm for novel images are the following: (a) the sets of interest regions are obtained by applying the three detectors, (b) the SIFT descriptor of each region is computed, (c) each set of descriptors is mapped with the respective dictionary to obtain a feature histogram for each detector, (d) the histograms are concatenated and evaluated by the learned classifier to obtain a class prediction. Figure 3.3 shows an overall picture of the classification framework for an image I, which details the computation of the component for class k in the feature vector generation procedure. This procedure is performed on I for each of the K class-related set of dictionaries and the feature vectors from each are concatenated.

3.4 Experiments and Results

The series of experiments with the stonefly dataset is to classify between very similar object classes with a relatively easy-to-segment background. The images are taken in a controlled environment, but with related species. This dataset contains 1240 images obtained from 263 specimens that were collected of four stonefly taxa from freshwater streams in Oregon. The species are *Calineuria californica*, *Doroneuria baumanni*, *Hesperoperla pacifica*, and specimens from the genus *Yoraperla*.

Table 3.2: Classification procedure. B is the number of bootstrap iterations (i.e., the size of the classifier ensemble).

Classification

Given a test image, let H be the concatenated histogram
resulting from feature vector construction.

Let $votes[k] = 0$ be the number of votes for class k.

For $b = 1, \ldots, B$

Let $\hat{y}_b$ be the class predicted by LMT_b applied to H.

Increment $votes[\hat{y}_b]$.

Let $\hat{y} = \text{argmax}_k \; votes[k]$ be the class with the most votes.

Predict $\hat{y}$.

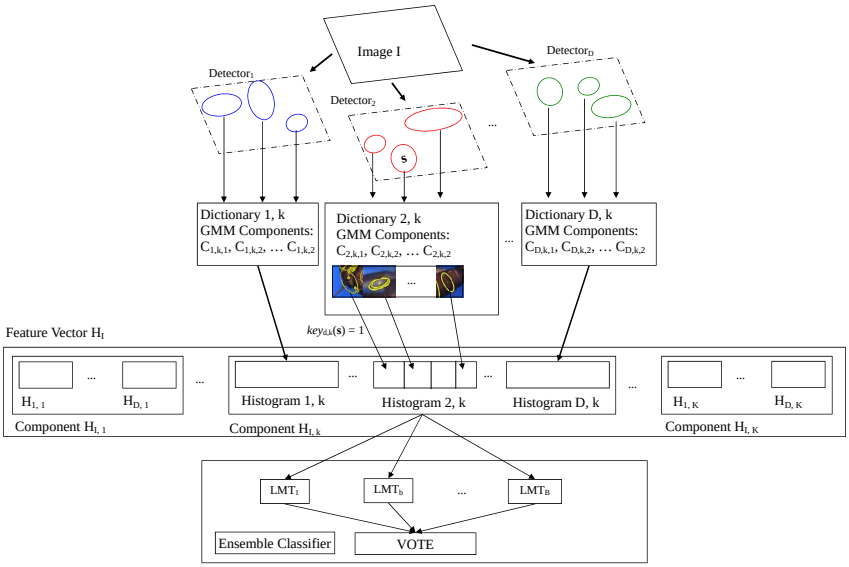

Figure 3.3: Classification Procedure for image I. Computation of the feature vector H_I and classification with the ensemble classifier. The figure details the computation of component H_k.

Table 3.3: Classification Error with all three region detectors along using a 95% confidence interval.

Task	Error [%]
CDHY	17.58 ± 2.12
JtHY	4.6 ± 1.16
CD	20.63 ± 2.70

These experiments where performed in a stratified three-fold cross-validation procedure. Each partition is roughly one third of the dataset, and takes a training or testing role in each iteration. The classification results are computed as an average of the three folds. This insures that the results are closer to performance results that would be obtained in real conditions. Three different classification tasks were defined for this data set given the characteristics of the taxa contained in it. We term these tasks CDHY, JtHY, and CD; and they are defined as follows:

CDHY: Discriminate among all four taxa.

JtHY: Merge the related species *Calineuria* and *Doroneuria* to define a single class, and then discriminate among the resulting three classes.

CD: Focus on discriminating only between *Calineuria* and *Doroneuria*.

The CDHY task assesses the overall performance of the system. The JtHY task is most relevant to possible applications, since *Calineuria* and *Doroneuria* have identical environmental responses. Finally, the CD task presents a very challenging classification problem. Table 3.3 shows the classification error achieved by this method on the three tasks.

On the CDHY task, our system achieves 82% correct classifications; achieving near perfect recognition of *Yoraperla*. As expected, the main difficulty is to discriminate *Calineuria* and *Doroneuria*. When these two classes are pooled together in the JtHY task, performance reaches 95% correct, which is excellent. It is interesting to note that if we had applied the four-way classifier and then pooled the *predictions* of the classifiers, the 3-class performance would have been slightly better (95.48% versus 95.08%). The difference is that in the JtHY task, we learn a combined dictionary

for the merged *Calineuria* and *Doroneuria* (CD) class, whereas in the 4-class task, each taxon has its own dictionaries.

The results obtained in these experiments demonstrated the feasibility of applying general image classification methods to automated stonefly-species identification. The CFH method even achieved better classification results than human attempting to identify [32] stonefly species from the same images. In the next chapter, we discuss our current classification method developed to obtain improved performance and reduce the feature extraction time.

Chapter 4

EFFICIENT APPEARANCE FEATURES: HAAR RANDOM FORESTS

In this chapter we describe our work with Haar random forests as the first-stage classifier. This work on feature extraction and classification is motivated by the existing shortcomings and possible performance improvements over our previous method. An important drawback of our previous histogram of local-features method was the use of computationally-intensive interest-region detectors for feature extraction. Additionally, it is often necessary to offset the lack of discriminative information of the current problem by generating a large number of regions through extensive scale-space search. These regions then have to be coded with high-dimensionality descriptors. This only increases the timing costs and could possibly miss discriminative regions on some classification tasks, thus having a negative task. We developed the HRF method to apply discriminative learning techniques to the local-feature extraction process. Our method is designed to evaluate image patches in a grid array in order to generate image feature vectors. Its goal is to include discriminative information (including the spatial distribution of the information) while reducing feature extraction time, because of the inherent efficiency of tree traversal and the constant computation time and scale-invariance of the Haar-like features. It is also our aim with this method to retain general object-class recognition capabilities, while initially perform experiments on stonefly species identification.

A Haar random forest is an ensemble of M decision trees learned by a randomized learning algorithm [25]. HRFs are powerful codebooks used as an intermediate image representation to cluster image patches into semantically relevant categories. Each HRF tree records the path traversed by each patch being evaluated. This path is generated by recursively branching left or right down the tree depending on the learned decision function associated with each node, until a leaf node is reached. Figure 4.1(a) illustrates a HRF with the highlighted path followed by the evaluation of a patch w through the nodes it visits. Figure 4.1(b) shows a Haar-like feature h_i from the set $\mathcal{H}^{l\times l}$; the set of extended Haar-like features [36] of $l \times l$ size applied to a patch w_b. Figure 4.1(c) shows the spatial pyramid and its subregions at every node in the forest.

We use these forests by applying a sliding window to the image and recording the count of windows that visited each node. Thus, it is possible to obtain a suitable codebook image representation to combine with a spatial matching kernel without using computationally expensive k-means and nearest neighbor assignment and without performing descriptor computation on interest regions. Decision forests have been show to be a good and efficient means of clustering descriptors [45]. We employ the tree hierarchies to cluster. Additionally, given that we know the position of each window in the image, we combine the HRF features with a spatial matching kernel to find rough geometric correspondence of features over fixed subregions between images. The way the HRF features are combined and evaluated with a spatial matching kernel is detailed below in Section 4.4.

4.1 Overview

The overall framework of this work can be divided into five parts: 1) preprocessing, 2) image patch extraction and description, 3) Haar random forest generation, 4) Haar random forest feature extraction, and 5) image classification.

For this study, the stonefly images were captured through an automated process [51] that snaps images of an insect as it passes through a mechanical apparatus with a blue background. In the preprocessing phase, the insect is automatically segmented from the background and oriented in a horizontal direction with the head facing left. This preprocessing step is specific to stonefly classification and is not required in applications where objects have a preferred orientation (e.g. cars, horses).

Once the insect is extracted and oriented, rectangular patches of three different sizes, each relative to the insect size, are extracted. From each extracted patch, $B = 12$ feature bands, each computed the same size as the patch, are produced: 3 color bands in the CIELab color space and 9 gradient orientation bands (every 20 degrees from $0\,^\circ$ to $180\,^\circ$). Each image patch is represented by the twelve bands and (during training) the label of the insect.

A random sampling of labeled patches from the training images is used for learning the Haar random forest. Using a standard bagging algorithm [9], M subsets of the set of training patches are produced, and each one is used to construct a decision tree for deciding if a given image patch comes from a particular species or not. At each branch node, one Haar feature along with one of the

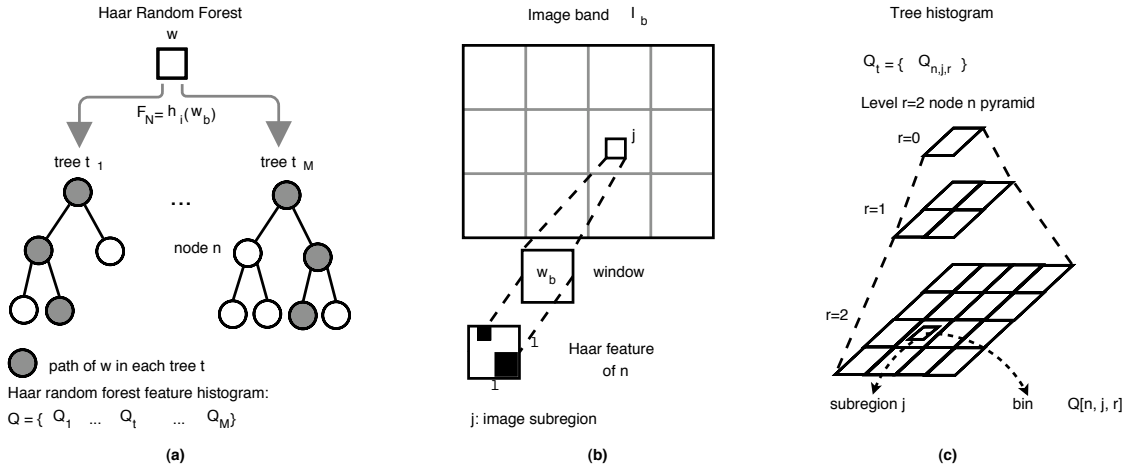

Figure 4.1: Haar random forest evaluation details. (a) HRF consisting of M trees with splitting function f at each internal node evaluated on feature band window w_b. (b) Haar feature h_i of $l \times l$ size. (c) Spatial pyramid at node n with $R = 3$.

six bands is chosen as the decision mechanism. The resultant M trees form the Haar random forest.

The Haar random forest works as a discriminative codebook to generate image feature vectors (histograms) that will be used to train an SVM classifier. The HRF feature extraction phase uses a regular sampling of patches from the training images, rather than the random subset used to create the random forest. The square image patches are obtained in three procedures of different patch-size ranges. These non-overlapping ranges are relative to the image size. They start from the base length of the Haar-like feature mask ($l = 26$ pixels in this case) to 0.1% of the stonefly image length, the next range is from 0.1% to 0.2% of the image length and final range is from 0.2% to 0.3% of the image length. For each range, the extraction procedure includes multiple passes over the whole image obtaining patches of increasing size (using a 1.1 factor) for every new iteration. In this phase, for each training image, all the patches in a given size range are put through each of the M trees of the random forest specific to that range. This generates, at each node of each tree, a count of how many patches passed through that node. In addition to finding spatial correspondences (which is roughly equivalent to part correspondences thanks to the orientation preprocessing stage), for each different image subregion in each of the three scale sampling ranges, a count of how many of these patches came from that subregion is kept. Thus each tree node contains a full patch count

and multiple subregion counts corresponding to the partitions of a spatial pyramid. Once all patches from a given image have gone through the random forest, M combined histograms are extracted, one from each tree. These histograms produce the feature vectors for image classification.

In the final classification step, the M histograms from the Haar random forest are concatenated to produce a single feature vector per training image and used to train an SVM classifier whose job is to determine if a given image belongs to a particular insect class or not. The next three sections further detail this methodology: Section 4.2 will describe Haar random forest generation, Section 4.3 explains the HRF image feature extraction, and Section 4.4 will describe the spatial matching HRF-feature kernel SVM developed for this work. Section 4.5 discusses the current set of experiments and results obtained applying this methodology to binary stonefly classification tasks.

4.2 Haar Random Forest Learning

The training set $\mathcal{W}$ consists of pairs $(\boldsymbol{x}, y)$ where $\boldsymbol{x} = \{w_1, \ldots w_B\}$ is a feature vector formed by the patches of the training sample window from B different image feature bands, and $y \in \{0, 1\}$ is the sample class label. The image feature bands include the three channels of the CIELab color space and nine gradient-orientation bin planes. Each tree in the forest is learned using a multiset $\mathcal{W}'$ obtained by sampling $|\mathcal{W}|$ pairs with replacement from the full training set $\mathcal{W}$. The learning algorithm proceeds in a top-down manner, recursively splitting the training data while adding new nodes to the tree. Each node n is defined by its associated decision function f and threshold τ. When a new node n is being added, several candidate functions $f' = h_i(w_b)\ \forall i \in I_n, b \in B$ are generated, where I_n is a set of random Haar-like feature indices considered for node n. The indices for each feature have uniform probability and $|I_n| = \sqrt{|\mathcal{H}^{l \times l}|}$. As indicated, all the existing image transform patches are considered for each feature; thus at each node, $|I_n| \times B$ candidate features f' are considered.

The candidate function f' that maximizes the information gain with respect to label y is selected. The training data $\mathcal{W}^n$ that arrives at node n is divided into subsets $\mathcal{W}^{l_n}$ and $\mathcal{W}^{r_n}$ assigned to the left and right children of n according to the optimal threshold τ' in the information gain sense for that particular f'.

$$\mathcal{W}^{l_n} = \{w \in \mathcal{W}^n | f' < \tau'\}, \tag{4.1}$$

Table 4.1: Haar random forest (HRF) learning procedure.

Forest Learning

Let $\mathcal{W} = \{(\boldsymbol{x}, y) | \, \boldsymbol{x} = \{w_1, \ldots w_B\} \wedge \, y \in \{0, 1\}\}$ be a HRF forest training set of image feature-patches $\boldsymbol{x}$ and sample class label y pairs where $\{w_1, \ldots w_B\}$ is the set of B training sample feature bands – i.e. CIELab color channels and bands of gradient orientation bins.

For bootstrap replicate $t = 1, \ldots, M$

Construct training set $\mathcal{W}'$ by uniformly sampling with replacement $|\mathcal{W}|$ examples from set $\mathcal{W}$.

Let $tree_t$ be learned by **TreeLearn** (Table 4.2) from $\mathcal{W}'$.

$$\mathcal{W}^{r_n} = \mathcal{W}^n \backslash \mathcal{W}^{l_n} \tag{4.2}$$

The information gain for a particular f' is

$$\Delta H_{f'} = -\frac{|\mathcal{W}^{l_n}|}{|\mathcal{W}^n|} H(y|\mathcal{W}^{l_n}) - \frac{|\mathcal{W}^{r_n}|}{|\mathcal{W}^n|} H(y|\mathcal{W}^{r_n}), \tag{4.3}$$

where $H(y|\mathcal{W})$ is the Shannon entropy of the label variable y given the samples from set $\mathcal{W}'$. The recursive splitting process continues until a depth limit D is reached or the number of examples falls below four instances. The values of D and M determine the number of nodes in a HRF. Table 4.1 describes the learning process of an HRF, and Table 4.2 details the tree learning procedure. The learned HRF is then utilized for feature extraction. These generated feature vectors are then employed to train an SVM kernel classifier.

4.3 HRF Image Feature Extraction

The HRF feature extraction process represents the image as a histogram Q. Each bin in Q corresponds to an image subregion j of spatial level r, and of node n from one of the HRF trees. The histogram is computed by scanning a sliding window across the image (represented as a set of image bands as described above). To obtain insect-part information at different sizes, the three different window-scale ranges mentioned in Section 4.1 are used. For each patch-size extraction

Table 4.2: HRF Tree learning procedure.

TreeLearn

Let $h_i \in \mathcal{H}^{l \times l}$ be a feature from the set of extended the Haar-like features of $l \times l$ dimensions.

Let $\mathcal{F}' = \{f' | \, f' = h_i(w_b) \; \forall \, i \in I_n \wedge b \in B\}$ be a set of candidate decision functions f' for new node n where I_n is a set of Haar-like feature random indices uniformly generated and $|I_n| = \sqrt{|\mathcal{H}^{l \times l}|}$.

Let Λ be list of tree nodes and the samples that reached to each of them. Let Λ initially contain root node n_R and the set of training samples $\mathcal{W}'$.

Let D be the depth limit and N be minimum number of samples to reach a node.

While Λ has elements

Get next node n and its associated training sample set $\mathcal{W}^n$ from Λ.

Randomly create the set of candidate functions $\mathcal{F}'$ for node n. Search for the best decision rule $f < \tau$ on $\mathcal{W}^n$ according to Eq. 4.3. Assign $f < \tau$ to node n.

If $deep(n) < D$

Create the sets $\mathcal{W}^{l_n}$ and $\mathcal{W}^{r_n}$ of training samples satisfying $f < \tau$ and Eq. 4.1 respectively. Calculate the error with respect to label y of each set.

If $error(\mathcal{W}^{l_n}) > 0$ and $|\mathcal{W}^{l_n}| > N$

Create node n_l and append it as left child of n. Append $(n_l, \mathcal{W}^{l_n})$ to Λ.

If $error(\mathcal{W}^{r_n}) > 0$ and $|\mathcal{W}^{r_n}| > N$

Create node n_r and append it as right child of n. Append $(n_r, \mathcal{W}^{r_n})$ to Λ.

Return root node n_R.

range, an HRF is trained with the patches extracted over that range. The histograms from each scale are concatenated to create the final image descriptor. To compute the bin counts, at each window position, the window w is "dropped" through each tree. As w traverses the tree, each node n that it visits counts the visit into bins corresponding to n and to the various spatial pyramid cells (j, r) to which w belongs. In order to save space, only the spatial grid at the finest level $r = R$ is stored explicitly—the bins at coarser spatial levels $r < R$ are computed on-the-fly when the histograms are being evaluated by the spatial match kernel. Hence each image histogram Q associated with an HRF tree is composed of the concatenation of all the node counts $Q[n, j, r]$. Note that the bins have a hierarchical structure, so that bins of every split node n and its children l_n and r_n satisfy $Q[n, j, r] = Q[l_n, j, r] + Q[r_n, j, r]$.

Feature Vector Dimensionality: The HRF learning has a fixed limit of split nodes per tree N_t (e.g. 70), and a total fixed number of split nodes on the forest N_{rf} (e.g. 700). Each HRF has around ten trees, each with $N_t + 1 = 71$ leaves and an average depth of $\log_2(N_t + 1)$ (≈ 21). Since the vectors are length 710 (total number of leaves) times the number of spatial regions at the deepest level ($4 \times 4 = 16$), there are 11360 independent dimensions. This is large, but similar in length to vectors in current computer vision systems [16] [53], which have not suffered from over-fitting. The evaluation of the internal spatial levels and split nodes acts as a partial match that smoothes the harsher matches of the final level, improving the similarity measure. A potential drawback of this approach is that the depth and number of trees must be chosen to to keep the length of the feature vector reasonable.

4.4 Matching-Kernel SVM Classifier

Our method employs an SVM classifier with a specialized non-linear kernel to take advantage of the discriminative hierarchy of the HRF trees and the spatial correlations between the image features. Our kernel is based on the pyramid match kernel [26]; it returns a similarity measure between image histograms and an approximate correspondence between two sets of elements. We extend this framework by combining previous ideas of using spatial [34] and tree-based partitionings [53].

The learning algorithm uses a standard SVM implementation with a specialized kernel. Consider the unnormalized matching kernel $\tilde{K}$ for just one tree t. Let P and Q be the pair of HRF feature

histograms computed across two images; then

$$\tilde{K}_t(P,Q) = \sum_{d=0}^{D-1} \frac{1}{2^{D-d}} \mathscr{S}_{d+1}(P,Q), \tag{4.4}$$

where D is the maximum tree depth level, d indexes across all the nodes of a certain depth , and $\mathscr{S}_d$ is equivalent to performing a spatial pyramid match across all the nodes at depth d. Hence $\mathscr{S}_d$ is defined as

$$\mathscr{S}_d(P,Q) = \sum_{r=0}^{R-1} \frac{1}{2^{R-r}} \big(\mathcal{I}(P_r^d, Q_r^d) - \mathcal{I}(P_r^{d+1}, Q_r^{d+1})\big). \tag{4.5}$$

$\mathcal{I}(P_r^d, Q_r^d)$ is the histogram intersection distance across all bins of nodes n at depth d and spatial cells j at level r. The normalized kernel of one tree, which can handle images of different sizes, is $K_t = \frac{1}{\sqrt{Z}}\tilde{K}_t$ where $Z = \tilde{K}_t(P,P)\tilde{K}_t(Q,Q)$. The final kernel over all trees in the forest is calculated as $K = \frac{1}{M}\sum_t K_t$, which is very efficient to evaluate. The kernel receives the vectors containing only the information at the deepest level, so all the internal split node bins are calculated and evaluated at run time. Given the definition of the histogram intersection distance $\mathcal{I}$, and the fact that all the bin values are non-negative, they can be pre-multiplied by their corresponding weights. After some manipulation, the kernel evaluation then becomes a summation of the minimum weighted values of corresponding pair of bins of P and Q of the form $\sum_k \min(Pweighted_k, Qweighted_k)$.

4.5 Classification and Timing Experiments

We now describe the series of experiments that were performed to evaluate the HRF system. A set of nine binary stonefly-species identification experiments were defined on the *STONEFLY9* [40] dataset. The dataset contains nine stonefly species referred to as Cal, Dor, Hes, Iso, Mos, Pte, Swe, Yor, and Zap. Figure 4.2 shows an example image containing one specimen of each of the nine species to illustrate the challenges posed by this task. The *STONEFLY9* dataset consists of 3826 images obtained by imaging 773 stonefly larvae specimens. Performance on all binary classification problems was evaluated via three-fold cross-validation. The images were randomly partitioned into three equal-sized sets under the constraint that all images of any given specimen were required to be placed in the same partition. In addition, to the extent possible, the partitions were stratified so that

Figure 4.2: Example images of different stonefly larvae species in the *STONEFLY9* dataset. From left-to-right and top-to-bottom: Cal, Dor, Hes, Iso, Mos, Pte, Swe, Yor, and Zap.

the class frequencies were the same across the three partitions.

We compared our algorithm to the best stonefly-species classifier in the literature [40], which uses stacked evidence trees (SET). Table 4.3 displays the classification error comparison between the SET method and our current method (HRF Lab+G), and our method only using the CIELab color channel bands (HRF Lab). Our new algorithm HRF Lab+G has lower average error and is much more accurate on the most difficult pair of species, Calineuria and Doroneuria (cal vs dor). The classification error when using only the color channels HRF Lab is much larger than the error when using color plus gradient-orientation feature bands, thus showing the relevant information provided by the gradient-orientation data.

Timing Evaluation Comparison: Our algorithm HRF Lab+G was compared to the evidence-tree algorithm SET on overall prediction time. We decided to use new test images from the cal vs dor experiment given that these are the two largest sets of species. The SET algorithm requires performing three separate processes that combine local-feature detection and description. The SET algorithm employs the Hessian-Affine [44], Kadir-Brady Salient [30], and PCBR [66] detectors; representing the obtained regions with the SIFT descriptor [38]. Each of these processes takes, on average, sev-

Table 4.3: Classification Error Comparison. SET, HRF on (Lab) color channels only and with gradient-orientation planes (Lab+G). See [40] for full species names.

	Error [%]		
Task	SET	HRF Lab	HRF Lab+G
cal vs dor	6.26	10.16	4.60
hes vs iso	3.74	9.05	3.55
pte vs swe	2.71	8.75	2.80
dor vs hes	2.25	8.09	2.20
mos vs pte	2.06	7.95	1.92
yor vs zap	1.52	6.89	1.60
zap vs cal	1.52	7.02	1.76
swe vs. yor	1.44	6.85	1.50
iso vs mos	1.29	6.90	1.30
average	**2.53**	**7.96**	**2.25**

eral minutes per image ($\approx$ 0.15 + 6.5 + 8.6 minutes = 915 seconds). The average processing time of the HRF Lab+G algorithm, from image load to histogram generation, is two orders of magnitude faster (5.03 versus 915 seconds). In both the HRF Lab+G and SET methods, the classification stage after the histogram vector loading takes only about 0.2 seconds. This experiment shows that the Haar random forest method is more accurate and much faster than the stacked evidence tree method.

Chapter 5

MULTIPLE FEATURES: STACKED SPATIAL-PYRAMID KERNEL

Realistic stream water quality evaluations require identifying specimen samples containing several dozens of different species. Each of these species has specimens with very different appearance, size, and shape. Furthermore, several groups of species within these large samples are very similar and difficult for an untrained observer to distinguish. The stacking classifier method proposed in this chapter combines the results from multiple classifiers and benefits from allowing each classifier to handle a different feature space. In this application, the combination of multiple features is clearly necessary, because it is evident that no single feature can discriminate among this large number of species while generalizing across the large variations present among the specimens of each species. Our object-class recognition method combines different feature types in a new stacking framework that efficiently quantizes input data and boosts classification accuracy, while allowing the use of spatial information with a pyramid kernel SVM. In contrast, standard stacking methods (e.g. Abdullah *et al*. [1], Martínez Muñoz *et al*. [40]) by their own nature discards any spatial information contained in the features, because only the combination of raw classification scores are input to the final classifier. This is a very relevant characteristic of this method, because, as already shown in Chapter 4, the spatial information of local features can be made to roughly match specimen parts and obtain classification accuracy gains from that match.

We evaluate our stacked spatial-pyramid kernel method on an image data set collected to emulate the species distribution found in typical biomonitoring samples. One of the most widely used biological monitoring metrics for water quality assessment is the population count of specimens from three insect orders: Ephemeroptera, Plecoptera, and Trichoptera, often abbreviated as EPT. The set of EPT specimen images employed in this work contains individuals from 29 species, and thus referred as the *EPT29* dataset. Table A.1 lists the species included in the dataset used for these experiments.

The SSP kernel classifier method retains the simplicity and elegance of combining features of

the evidence tree method [40], while allowing the use of local-feature spatial information in a robust way. The use of spatial information has proven useful in visual dictionary [34] and tree ensemble methods [33, 53] for generic object recognition and image classification tasks. Our work, which uses a spatial pyramid-kernel SVM classifier [34] while allowing diverse feature types that can complement each other to be combined, has been shown to be a successful technique to boost classification accuracy [1, 40]. Our classification framework is able to consider the discriminative structures from the parts of different objects invariant to small changes in position and size.

A diverse set of features is a very important factor given dataset characteristics of this application. The specimen images from the *EPT29* set contain wide intra-species appearance variation due to different 3D specimen position and orientation, natural diversity, developmental stages, and degradation after capture as well as species subsets with small inter-species differences. We employ shape and edge keypoints and a dense grid of patches to obtain useful regions for local features, which are described by the HOG [16] and SIFT [38] appearance descriptors and by the beam-angle histogram for shape [3]. The relative position information represented by a spatial histogram is generated by the initial random-forests stage using the appearance and shape feature scores. This combination of features and the direct use of spatial information is shown to achieve a large increase in classification accuracy for the challenging *EPT29* identification task.

5.1 Stacked Spatial Classifier for Generic Object Recognition

The proposed classification architecture draws on the idea of using evidence trees for dictionary-free classification [40] and on the use of spatial information at the final classification level [33] to identify the species of the specimen. Insect identification research in [32, 40] has shown that combining multiple interest region detectors and their respective descriptors gives better results than single detectors. Stacking is a framework that is able to combine diverse features. The benefit of this feature combination has also been shown to work well in object recognition [1]. The large variation present in all 29 species chosen for our current dataset highlights the need for a diverse set of features. It becomes obvious that no single type of feature would distinguish between all of them, which makes this recognition problem more challenging.

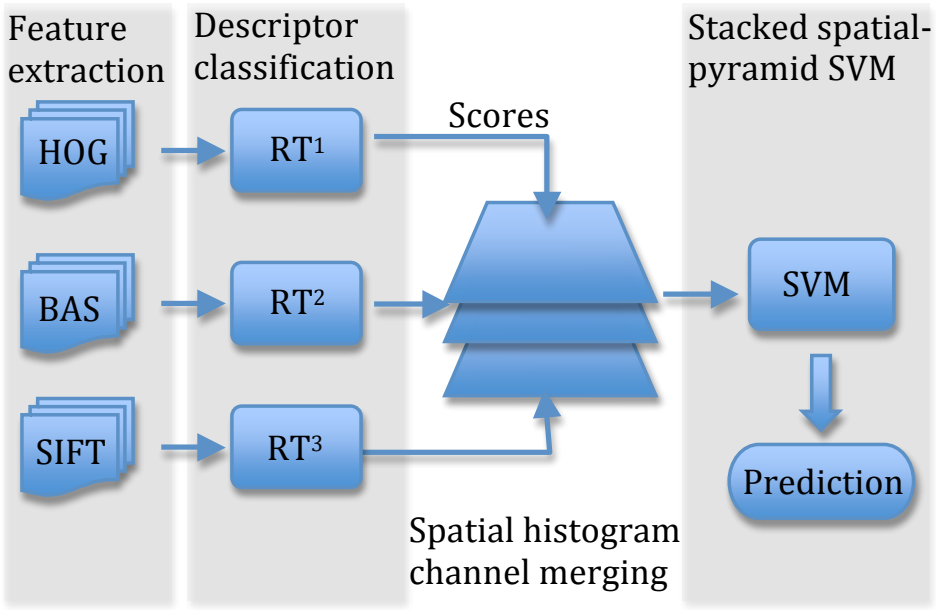

Figure 5.1: Overview of the stacked spatial-pyramid (SSP) kernel classification architecture.

5.1.1 Overall Class Prediction

We now give an overview of the prediction process of our method. For the *EPT29* dataset, the image background is first segmented out and the specimen is automatically aligned with the horizontal axis of the image. The specimen is then oriented facing left by a linear-kernel SVM classifier evaluated on a global HOG [16] descriptor trained to distinguish between specimens of this dataset facing left or right. The orientation classifier of this preprocessing stage is trained with a much smaller subset of the whole dataset (64 images with at least two images of each species). The performance is near perfect (above 99%) even with this small number of training images because the classifier only needs to capture the overall shape of the specimens to predict their orientation. Samples of the resulting images are shown in Figure 5.2. After this preprocessing step, our classification framework is composed of three main stages: (1) region detection and descriptor extraction of low-level local features, (2) local-feature classification and spatial histogram computation, and (3) insect-species prediction. The overall prediction framework of our system is shown in Figure 5.1.

The initial classification stage of the system is composed of a set of C classifiers that match every (detector, descriptor) feature type employed. $\mathcal{Q}$ is the set of (detector, descriptor) pairs that are applied to each new image I to be evaluated. The stacking component of our method enables the combination of very different types of pairings, each with different invariant characteristics as well as the retention of their spatial information. In this work the employed pairs are (1) salient points of high curvature [13] with a beam angle descriptor [3], (2) a dense grid of overlapping image patches with the HOG descriptor, and (3) the difference of Gaussians detector and SIFT descriptor [38]. The application of each pair in $\mathcal{Q}$ generates a set of detections represented by a set of descriptor vectors $B_I^c = \{x_{I,1}^c, \ldots, x_{I,N_c}^c\}$ where N_c is the number of detections obtained by the (detector, descriptor) pair c.

The set of initial local feature classifiers is composed of a random-forest classifier RT^c for each combination c. These classifiers are employed to obtain a probability classification score of each of the M classes for every descriptor vector $x_{I,j}^c$. The probability score $p \in \mathbb{R}^M$ is an M-dimensional vector employed to build the spatial histogram of classification scores $H_I^{c,L}$ of the finest spatial-grid resolution level L. This histogram is then used to construct a sequence of histograms with grid density-levels $\ell = 0, \ldots, L$ in the same manner as in [34]. Each histogram $H^{c,\ell}$ with resolution level ℓ has 2^ℓ cells along every spatial dimension (a total of $2^{2\ell}$ cells) and M channels in every cell $H^{c,\ell}(i)$. The set of histograms $\{H_I^{c,\ell} | c = 1, \ldots, C\}$ obtained with the set of random trees $\{RT^c\}$ is merged along the class-channel dimension into a singe spatial histogram H_I^ℓ of $C \times M$ channels and $2^{2\ell}$ spatial cells. The sequence of histograms $\{H_I^\ell | \ell = 0, \ldots, L\}$ constitutes the feature vector of image I for the final classifier, which is well suited for the spatial-pyramid kernel.

5.1.2 Combining Local Classification Scores

The procedure to generate the histogram $H_I^{c,L}$ of image I is as follows. For every classifier c of $\mathcal{Q}$, every descriptor $x_{I,j}^c$ of B_I^c is evaluated by each tree in RT^c. Every tree votes for the class of the descriptor. The class probability score $p_{I,j}^c$ is computed by summing the votes for each class m in the vector component $p_{I,j}^c[m]$. The values of $p_{I,j}^c$ are then normalized to sum to 1, to estimate the posterior probability $\mathbf{p}(m|x_{I,j}^c)$. After the scores are obtained for every element in B_I^c, for each spatial cell i at grid density L, all the score vectors $\{p_{I,j}^c | l_{I,j}^c \in i\}$ whose descriptor location

$l^c_{I,j}$ falls within the ith cell grid are accumulated via vector addition in the respective channels of bin $H^{c,L}(i)$. The accumulation of discriminative information in the form of scores is the main difference between our method and the original spatial-pyramid kernel SVM method [34], which only accumulates cluster assignment counts. These histograms carry the spatial information of local features to the stacked classifiers, which contrasts with [1], where spatial information is only employed at the first classification level. Thus our method has all the benefits of the bag-of-features approach while accumulating descriptor classification scores. The possible drawback of methods using the stacking framework, such as this one, is that the number of classes is often much smaller than the possible number of dictionary keywords for the input feature space, so less information is available to the final classifier. Although the classifier accuracy could be affected, this is often offset by the more discriminative information.

The use of the spatial histograms also has the advantage of indirectly maintaining information about the number of features detected in different image regions, which correlates with the image content. The results presented in Section 5.3.3 using the histograms generated with a single pair c compared with the ones obtained by combining all pairs in $\mathcal{Q}$ show that as with standard stacking techniques, our method benefits from combining complementary feature types in H^L_I.

5.1.3 Stacked Spatial-Pyramid (SSP) Kernel SVM

In order to perform the pyramid matching in two-dimensional image space, a sequence of histograms $\{H^\ell_I\}$ matching the grids at different resolution levels $\ell = 0, \ldots, L$ is built. Let $\jmath_{1,\ldots,4}$ be the grid cells at level $\ell + 1$ that subdivide cell i at level ℓ. The recursive process to compute the sequence of histograms starts with H^L_I. All the bins in cell i at level ℓ are computed with the subdivisions at level $\ell + 1$ by the following vector relationship $H^{c,\ell}(i) = \sum_{k=1}^{4} H^{c,\ell+1}(\jmath_k)$. We refer to the whole sequence of resolution-level histograms of image I as H_I. Like the feature counts in the original spatial-pyramid kernel, the class probability scores are amenable to this accumulation process. For a pair of score histograms H_{I_1} and H_{I_2} computed across all the initial classifiers c representing image I_1 and I_2, the spatial-pyramid matching kernel $\mathcal{K} = K(H_{I_1}, H_{I_2})$ is

$$\mathcal{K} = \sum_{\ell=0}^{L} \frac{1}{2^{L-\ell}} \big(\mathcal{I}(H^\ell_{I_1}, H^\ell_{I_2}) - \mathcal{I}(H^{\ell+1}_{I_1}, H^{\ell+1}_{I_2}) \big) \tag{5.1}$$

where ℓ indexes the spatial resolution levels. $\mathcal{I}(H_{I_1}^{\ell}, H_{I_2}^{\ell})$ denotes the histogram intersection distance across all class channels and spatial cells of level ℓ. Note that for $\ell = L + 1$ this distance has value zero. The kernel $\mathcal{K}$ can handle different numbers of detections in each image. The similarity value that $\mathcal{K}$ represents is directly related to the number of descriptors and their classification score values. The weight associated with level ℓ is inversely proportional to the cell-width value; thus penalizing matches found in larger cells. These coarser-level matches are still used; they account for larger changes in position. Matches at the finest level are weighted the most, while still being robust to small changes in position. Our SSP method then employs kernel $\mathcal{K}$ with the standard learning and prediction SVM algorithms. Experimental results described in Table 5.1 show the benefits of this image histogram descriptor with this type of kernel classifier, which outperformed the other stacked classifiers on the same histograms of feature combinations.

5.2 Stacked Training Set Creation and Learning

As indicated in the classification overview, our method is composed of two classification stages. The learning process thus requires three different procedures: (1) learning of the random forest classifiers $\{RT^c\}$ that will generate the local feature classification scores, (2) creation of the set of spatial histograms of scores $\mathcal{H}$ that constitutes the final classifier training set, and (3) learning of the final stacked spatial-pyramid classifier. This procedure is aimed at obtaining robust classifiers capable of handling all the variations present in the dataset while achieving high classification accuracy.

5.2.1 Random Tree Learning

For each (detector, descriptor) pair c, a set of random trees RT^c is created from a training set $\mathcal{B}^c$. For each training image I with category y_I, every descriptor of set B_I^c of image I forms a training pair $(x_{I,j}^c, y_I)$. The training data $\mathcal{B}_\tau^c$ of size N of each tree τ is obtained through a bootstrap sampling procedure by drawing at random with replacement $N = |\mathcal{B}^c|$ descriptors with uniform probability. A set of Υ random trees is learned [10] from different $\mathcal{B}_\tau^c$ training sets, constrained for maximum tree depth and a minimum of 10 examples arriving at each leaf in the learning procedure. The parameter values for this learning step were determined experimentally on the training data. Figure 5.3 shows the relative insensitivity of the overall accuracy after 150 trees. In the tree learning

procedure, every time a node is added, a subset of the attributes of the training examples of c (region descriptor and normalized region location) is randomly selected along with a threshold value as the node splitting function. This combination of attributes allows the specialization of local classifiers in the first stage, which benefits the coupling of position and score accumulation as input of the stacked classifier. As part of the training process, for every tree τ, all the out-of-bag (OOB) training examples of τ (descriptors not used to train τ) are recorded. This information is then used in the next learning step to generate the training set for the stacked classifier.

5.2.2 Stacked Classifier Training Set

Following the learning of the random trees classifiers, the training set for the stacked classifier is created. This set contains one spatial histogram per image. Let $\mathcal{H}^c$ be the set of labels and training histograms pairs obtained using single feature c and $\mathcal{H}$ be the set of label and training histogram pairs combining all the features from $\mathcal{Q}$. After the training of each RT^c classifier, the spatial histogram H_I^c of each image I in the training set is constructed using the same training descriptors from set $\mathcal{B}^c$ in a procedure similar to the one described in Section 5.1.2. The only difference in computing H_I^c is that for every descriptor $x_{I,j}^c$ of B_I^c being evaluated, its class-probability score $p_{I,j}^c$ is computed using only the votes from trees where this descriptor was an OOB element during training. Thus the values of $p_{I,j}^c$ are normalized by the number of OOB tree votes instead of the total number of trees. It is important to only use the trees where the example $x_{I,j}^c$ was never seen to model the behavior for unseen descriptors of a novel image being classified in the training set $\mathcal{H}^c$. The class label of image I is then assigned as the label of H_I^c. All the training pairs (H_I^c, y_I) are combined to create the stacked training set $\mathcal{H}^c$. The combined training set $\mathcal{H}$ is built by merging all the histograms H_I^c, of every training image I along the class-channel dimension. The experiments are carried out in a 3-fold cross-validation setup; thus at every fold the training set is composed of approximately two-thirds of the image data ($\approx$ 3148 images). The OOB voting setup is especially suitable for cross-validation experiments, because it enables the use of all the training data for the learning of the random forest classifiers and the SSP kernel classifiers.

5.2.3 Stacked Spatial Classifier Learning

The parameters for the stacked spatial-pyramid kernel SVM learning algorithm are obtained by a logarithmic grid-search with a 5-fold cross-validation using the training set on each iteration of our experiments. For the multi-class experiments performed in this paper, a one-versus-all framework was employed. This learning stage of the stacked SVM classifier can be performed with a single-feature training set $\mathcal{H}^c$ or with a multiple-feature training set $\mathcal{H}$. The spatial-pyramid kernel $\mathcal{K}$ is used in the loss and discriminant functions of the SVM learning procedure. This kernel is well suited for the score histogram image representation H_I, because classification scores just like assignment counts [34], can be accumulated in the spatial pyramid.

5.3 EPT Insect-Species Identification Experiments

In this section, we describe the series of experiments evaluating our method in the challenging *EPT29* dataset and report species and overall classification accuracy results. The *EPT29* dataset contains 4722 images divided into 29 species (See Appendix A for a list of the codes employed to identify every species) as indicated in the first column of Table 5.1. Figure 5.2 contains example images including one specimen of each of the 29 species and images of the caddisfly larvae cases present in the *EPT29* dataset to show the problems posed by this recognition task. Cases are built by the larva from surrounding material (e.g. stones, twigs, leaves, sand) and they are thought to protect the larva from predators. These case images are included to match the specimens that will be captured in an actual biological monitoring task; thus for the purpose of this experiment these species have multimodal appearance distributions. The specimens used to create this dataset were captured and identified by experts aiming to approximately emulate a biomonitoring sample for stream quality assessment. Note that some of these species have as many as ten times more images than the species with the fewest images. This type of imbalance tends to make identification tasks more challenging.

5.3.1 Random Tree Local Features

For the *EPT29* dataset experiments, our method has $C = 3$ initial RT^c classifiers. The (detector, descriptor) pairs c were selected to use regions with complementary position, shape, and orientation

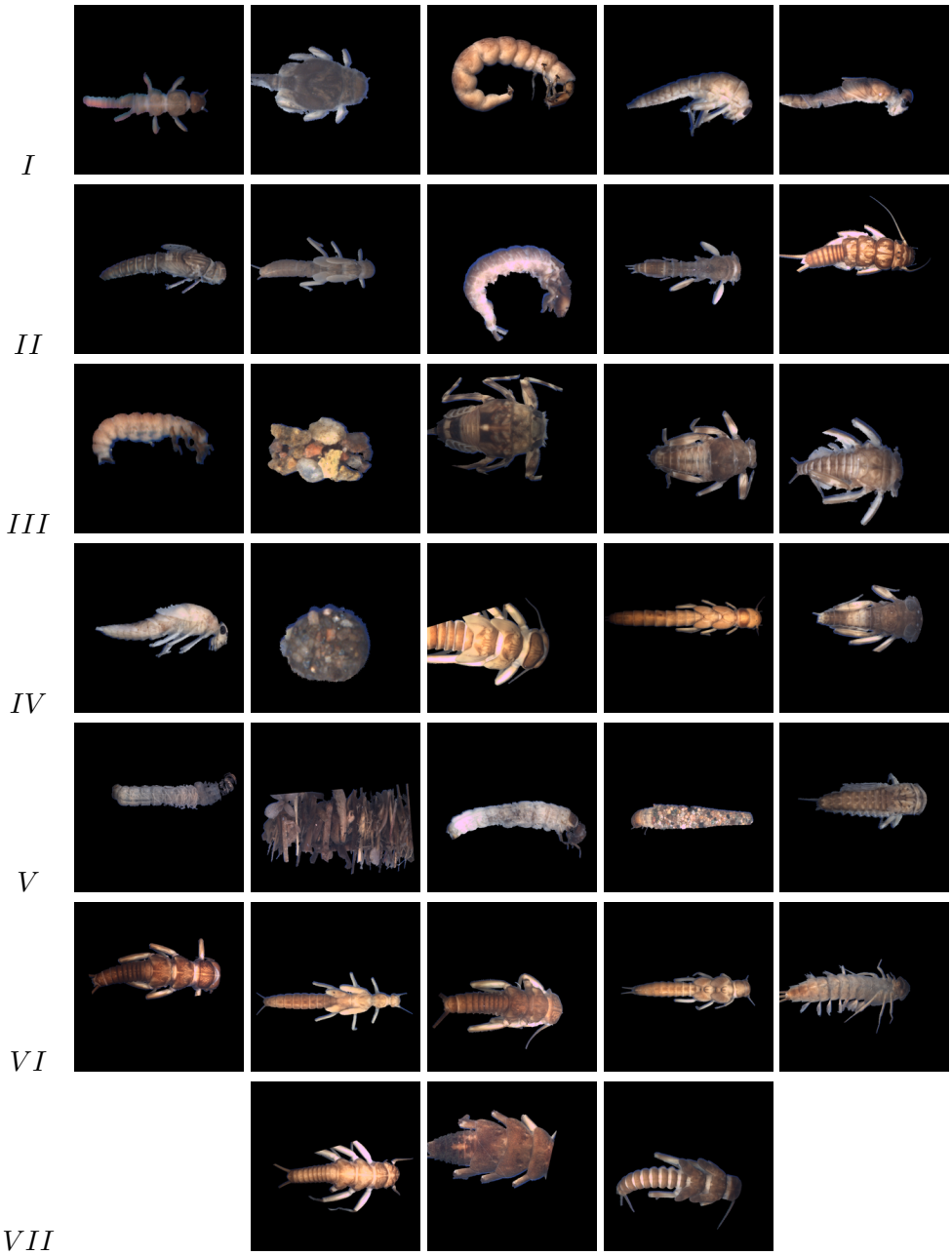

Figure 5.2: Example images from the Ephemeroptera (mayflies), Plecoptera (stoneflies), and Trichoptera (caddisflies) insect orders that conform the EPT29 dataset. These include larvae specimens and/or their cases. From left-to-right and top-to-bottom: (I) Amphin, Asiop, Atops, Baets, Calib, (II) Camel, Capni, Cerat, Cinyg, Cla, (III) Culop, Culop (case), Drunl, Drunl, Epeor, (IV) Fallc, Hlpsy (case), Isogn, Kat, Leucr, (V) Limne, Limne (case), Lpdst, Lpdst (case), Lphlb, (VI) Meg, Mscap, Per, Plmpl, Siphl, (VII) Skw, Sol, and Taenm.

attributes; the descriptors for those regions were selected with those criteria as well. The pairs c employed are denoted as follows: dense grid of overlapping image patches with the HOG descriptor (*HOG*), salient points of high curvature with a beam angle and SIFT descriptors (*BAS+SIFT*), and the SIFT difference of Gaussians detector and descriptor (*SIFT*). A detailed description of the feature-extraction procedures follows.

HOG descriptors are obtained over a dense grid of 16×16 overlapping image patches, regardless of the image size. These patches overlap by 50% (8 pixels) in both the horizontal and vertical direction to overcome small changes in position. The HOG [16] descriptor has proven really useful in detection tasks given the highly-redundant nature of its data. This grid-based feature can be applied even with the large changes in 3D position present in the data thanks to the orientation normalization process and the properties of the local features.

BAS+SIFT descriptors are obtained on salient points of high curvature along the contour by using the IPAN [13] algorithm. For each salient point in the contour, a beam angle statistics (BAS) descriptor similar to [3] is generated and concatenated with a SIFT descriptor to combine shape and appearance. To compute the BAS descriptor, a set of lines, so-called beams, that originate from the anchor points connecting to each of the remaining salient points in the supporting region are constructed. The angles between pairs of lines are calculated and accumulated with a weight inversely proportional to the perimeter distance between salient points. The BAS descriptor is the weighted histogram of these angles. The radius of the supporting region is selected adaptively depending on the length of the contour. A multiscale descriptor is generated by concatenating descriptors of different region size.

SIFT features are found with the difference of Gaussians detector and represented by the SIFT descriptor as described by Lowe [38]. The region descriptors are scale invariant, isotropic, and invariant to rotation in the image plane.

5.3.2 Experimental Setup

All the species-identification experiments are performed with a stratified 3-fold cross validation setup. In order to make the results of every experiment completely comparable, equal-sized random dataset partitions are the same in every fold of all the experiments. Given that 1–4 images were obtained for every specimen, the partitions are constrained to keep all the images of any given specimen in a single fold. In the different images of a specimen, it appears in different 3D positions, orientations and poses. At every iteration of the experiment, two folds are used for training and one for testing. The results combining the predictions of the three testing procedures are reported. The values of the tree count Υ and the maximum tree depth parameters of every classifier c are determined experimentally using only the training data of one iteration. Figure 5.3 shows the behavior of the overall accuracy of *BAS+SIFT* features in relation to different parameter value combinations. The graph shows that accuracy increases flatten after 150 trees with a maximum depth of 25. For the other two pairs of $\mathcal{Q}$, the accuracy presents a similar behavior; thus the parameter values for the tree count Υ and maximum depth were set to 200 and 25 respectively for all three ensemble classifiers. For the spatial histogram H_I^c, the number of resolution levels L is set to 2 for a 4×4 grid at the finest level. The number of channels M is 29 constituting a 464-dimension image descriptor.

5.3.3 Experiments and Results

Table 5.1 reports the species and overall classification accuracy rates of the stacked spatial-pyramid kernel SVM method (SSP), stacked random-forest classifier (RTs 3Cmb), RBF kernel SVM (RBF 3Cmb), and a single-level classifier with a χ^2 kernel SVM on a global HOG descriptor. For convenience, every species reported in these experiments is represented by a code. Table A.1 indicates the full name of the species represented by these codes.

The SVM classifier utilizing a global HOG descriptor (χ^2 HOGgbl) is used as a baseline for the classifier comparisons. This is a similar classification setup to the one employed for the specimen orientation, but it uses a multi-class SVM classifier to predict species. This method is based on the approach often used for state-of-the-art object-class detection [20]. For this experiment, accuracy results with the χ^2 kernel are reported; results with the standard linear kernel were even lower. The accuracy differential shown between the single-stage global HOG descriptor and the stacked

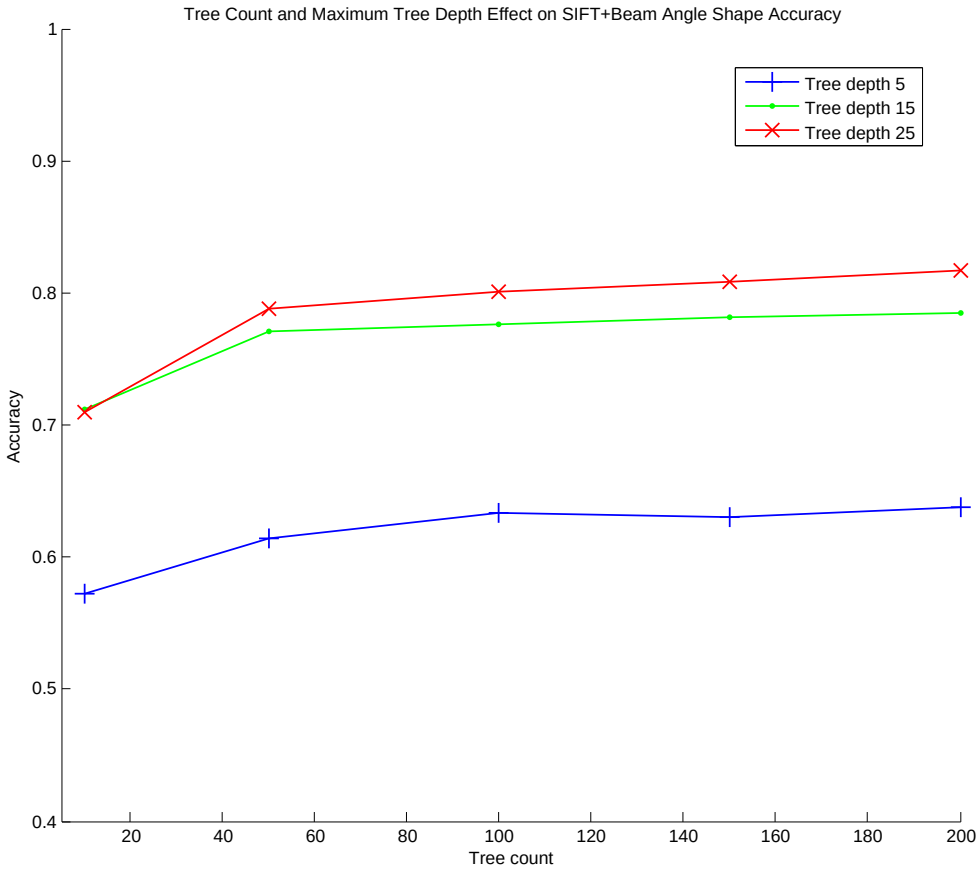

Figure 5.3: Overall accuracy graphs for the stacked spatial-pyramid kernel SVM with single feature pair BAS+SIFT showing the effect of the first-stage tree count Υ and maximum tree depth parameters.

Table 5.1: Species accuracy (Bold shows the highest species accuracy). From left to right, single-level global-HOG descriptor with χ^2 kernel. Following three, stacked spatial-pyramid kernel (SSP) classifier with single feature pair c: local HOG patches (HOGloc), BAS+SIFT descriptors of salient curvature points (BAS), and SIFT detector and descriptor (SIFT). Last three, stacked classification using the 3-feature-types combination (3Cmb): Random forest (RTs), RBF kernel SVM (RBF), and SSP kernel SVM.

Species Code	Image Count	Accuracy [%]						
		χ^2	SSP			RTs	RBF	SSP
		HOGgbl	HOGloc	BAS	SIFT	3Cmb		
Amphin	96	41.67	73.96	79.17	80.21	79.17	80.21	**85.42**
Asiop	292	95.55	95.21	95.89	95.21	96.92	95.21	**97.26**
Atops	254	75.20	86.61	89.37	87.80	87.40	86.22	**92.91**
Baets	251	72.11	85.26	83.67	79.68	81.27	86.06	**86.85**
Calib	299	60.20	71.24	73.58	79.93	77.93	75.59	**83.61**
Camel	287	77.00	83.62	83.28	86.76	82.58	85.71	**89.90**
Capni	130	42.31	60.77	77.69	79.23	75.38	76.15	**88.46**
Cerat	296	82.77	84.80	88.85	87.50	75.68	88.18	**90.88**
Cinyg	72	26.39	36.11	59.72	63.89	22.22	69.44	**79.17**
Cla	54	12.96	35.19	51.85	83.33	55.56	77.78	**87.04**
Culop	95	52.63	68.42	67.37	**81.05**	66.32	76.84	80.00
Drunl	42	23.81	61.90	78.57	78.57	59.52	66.67	**85.71**
Epeor	200	89.00	**93.50**	87.00	89.00	92.00	90.00	93.00
Fallc	224	36.61	44.64	54.46	62.95	46.43	**75.89**	64.73
Hlpsy	67	77.61	82.09	89.55	92.54	80.60	86.57	**95.52**
Isogn	229	78.60	86.46	87.77	**94.32**	81.22	93.45	93.89
Kat	48	58.33	41.67	64.58	**72.92**	29.17	70.83	**72.92**
Leucr	131	79.39	92.37	85.50	89.31	**96.18**	90.84	**96.18**
Limne	329	93.01	92.71	96.96	96.05	94.53	96.05	**98.18**
Lpdst	77	45.45	84.42	81.82	76.62	76.62	81.82	**87.01**
Lphlb	27	3.70	51.85	44.44	44.44	7.41	55.56	**59.26**
Meg	72	41.67	38.89	50.00	69.44	55.56	63.89	**77.78**
Mscap	132	58.33	67.42	73.48	75.00	68.94	80.30	**81.06**
Per	51	1.96	29.41	45.10	45.10	0.00	50.98	**52.94**
Plmpl	126	75.40	85.71	83.33	88.89	87.30	81.75	**92.06**
Siphl	150	60.00	78.00	**90.00**	88.67	85.33	85.33	**90.00**
Skw	292	63.01	71.92	78.42	77.74	61.99	81.51	**82.88**
Sol	129	86.82	87.60	87.60	94.57	92.25	93.02	**95.35**
Taenm	270	69.63	86.67	88.89	90.00	88.89	88.52	**91.48**
Total	4722	68.21	77.95	81.66	84.16	77.51	84.50	**88.06**

classifier method with local HOG descriptors clearly indicates the benefits of using local features as well as the ability of the SSP kernel to capture discriminative body-part information from the local feature positions.

The stacked spatial-pyramid kernel classifier is evaluated utilizing every single detector-descriptor pair and with a combination of the three feature pairs combined. The random-forest and the radial-basis function (RBF) kernel SVM classifiers are the other stacked classifiers employed for comparison with the SSP method. Even when these two classifiers are employed with the 3-feature combination, the results clearly indicate the beneficial coupling of the spatial histograms of scores with the spatial-pyramid kernel; these two stacked classifiers achieved two of the highest species accuracies. Because these classifiers are not designed to effectively use the spatial information contained in the input histogram vectors, even the single feature spatial-pyramid classifier using SIFT (SSP SIFT) had an overall classification rate similar to the highest of these classifiers (RBF 3Cmb) using the combined features.

Additionally, the table clearly shows the effects of the number of images on individual species accuracy, since most of the species with lower maximum accuracy have fewer images. This imbalance in the dataset is the reason for the poor performance of the stacked random forest (RTs 3Cmb); it appears that the tree learning process gave up classifying the species with small numbers of images. In contrast, the SSP 3Cmb classification setup was the least affected by the small number of images of those species.

Finally, we note the large accuracy boost obtained by using feature combination with our stacked spatial-pyramid kernel method (SSP 3Cmb) compared to the SSP kernel classifiers using only a single feature type. Figure 5.4 shows a graphical representation of the confusion matrix for the stacked spatial-pyramid kernel method with combined features (SSP 3Cmb) representing the image counts by actual species (row) and species prediction (column). The red pattern in the diagonal indicates that the majority of the specimens are correctly classified.

Discriminative Region Patterns

Figure 5.5 shows representative high-score region patterns of the spatial-histogram channel m corresponding to the actual species of the displayed specimen. Each of the feature-type pairs c is shown

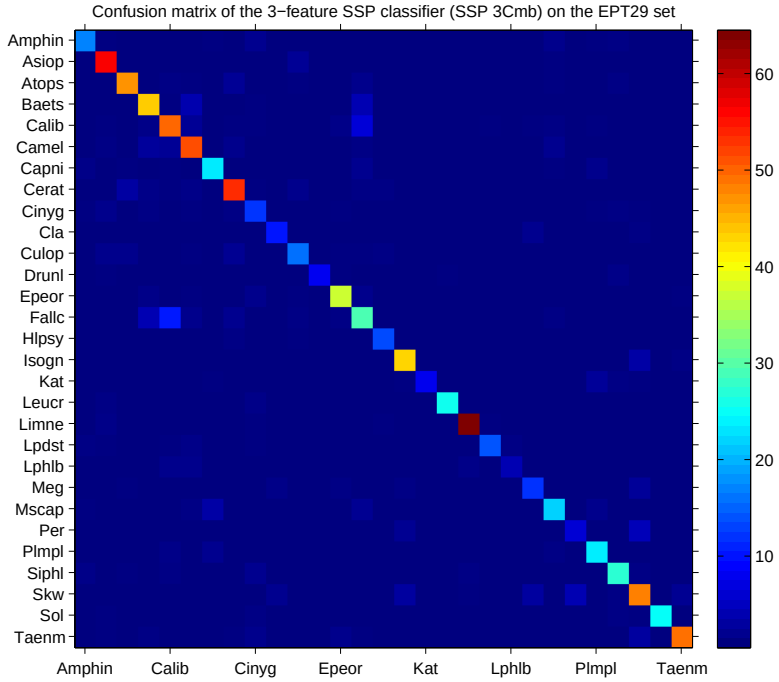

Figure 5.4: Heat-map representation of the confusion matrix for the stacked spatial-pyramid kernel using the 3-feature-types combination (SSP 3Cmb) on the EPT29 set. The figure represents the image counts by actual species (row) and species prediction (column). Red indicates a higher count; thus the majority of the predictions were correct and contained in the diagonal.

applied to individuals from the species that attained the highest species-identification accuracy of the single feature classifiers. The red parts of the insects are highly discriminative regions in the spatial histograms, which create high number of positive matches when the SSP kernel is applied to support vectors and test specimen pairs of the same species.

Evidence Histogram Results

The spatial histograms H_I^c can also be computed by using descriptor probability scores $p_{I,j}^c$ based on leaf evidence [40] instead of simple leaf votes. Note that in this procedure we go through two different normalization processes. The first process normalizes the total evidence count of classes that is computed at every node, instead of the equivalent single vote of giving a value of one to the class with the largest count found during training. The second normalization is the previously described procedure to create the spatial histograms of scores. We performed two experiments using the previously described experimental setup, but with histograms computed by evidence scores from the *HOG* and *BAS+SIFT* random tree classifiers RT^C. The overall accuracy results obtained are 77.47% and 80.62%, respectively, which shows no notable difference from the accuracy obtained using votes. The similar performance is probably explained by the large number of trees and the way the variables are aggregated in the stacked spatial-pyramid kernel SVM prediction procedure.

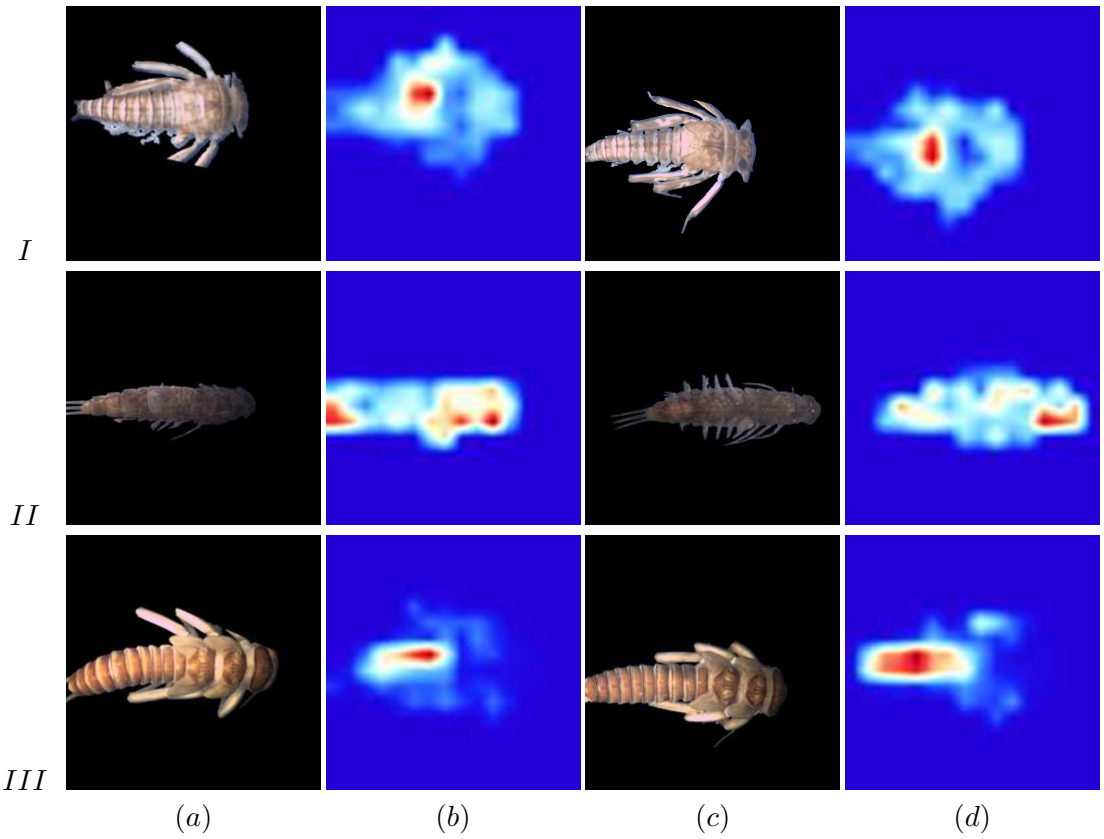

Figure 5.5: Representative heat-maps of local classification scores. (a)(c) Example original specimen images. (b)(c) Representation of a single channel m of histogram H_I^c. The channel m presented is of the actual species of the specimen. Red represents the highest scores, indicating some of the patterns of the most discriminative regions of that species. Each species is shown with the feature c that achieved the highest accuracy (Table 5.1). First row, (I) Epeor species with dense grid HOG patches, (II) Siphl with salient curvature and BAS+SIFT descriptors, and (III) Isogn with SIFT detector+descriptor.

Chapter 6

CONCLUSION

This dissertation has described three generic object recognition methods applied to arthropod-species identification with the goal of automating the identification of specimens for high-throughput, cost-effective biological monitoring of stream water quality using EPT indices. The methods include: (1) the concatenated feature histogram (CFH) method, (2) the Haar random forest (HRF) method, and (3) the stacked spatial-pyramid (SSP) method. We performed experiments with each of the three recognition methods to evaluate, first, the feasibility of automated identification by vision methods and, then, of performing realistic water quality assessments. The experiments described in this work show a progression in classifier complexity and capability as well as in the challenge posed by the image data sets.

6.1 Summary

The CFH is a bag-of-features method suitable for species classification because of its species-specific GMM-based codebook that maps sets of local features into histogram vector counts. The CFH method enables the use of multiple features by having a specific codebook for each one. For this work we employed three detector-descriptor pairs composed of the saliency, Hessian-affine, and PCBR detectors and the SIFT descriptor. Each specialized codebook creates an image representation that is later concatenated into the final representation. The concatenated histograms are the input to a LMT classifier.

The HRF method provides a combination of efficient low-level feature evaluation with discriminative learning. A semantic codebook quantized representation of the image is obtained in a single stage. The low-level feature space for this method is discriminatively selected from a set of Haar-like features. The HRF features also contain spatial information that can be roughly correlated with the appearance of insect body parts by using bins for specific tree nodes and image regions. These features are then applied by the spatial-pyramid kernel SVM for successful stonefly classification.

The SSP method also allows combination of dissimilar local features generated by different detectors and descriptors but using a discriminative framework. Shape and edge keypoints and a dense grid of patches are used to obtain useful regions for local features, which are described by the HOG and SIFT appearance descriptors and by the beam-angle histogram for localized shape. The relative position information is represented by the spatial histogram of scores. These histograms are constructed using the feature scores obtained by the initial random-forest stage evaluated on the image appearance and shape descriptors considering the image location where they were obtained. The spatial histogram of scores is proven to be a simple, yet powerful representation when employed by a stacked spatial-pyramid classifier to predict the species of novel specimen images. The combination of features and the direct use of spatial information in a stacking framework are shown to be successful by a large classification accuracy increase in the challenging *EPT29* identification task.

6.2 Discussion

Each of the recognition approaches described in this work have relative advantages, this is particularly noticeable between the more complex HRF and SSP kernel methods, which achieved good results on more challenging data sets. Although, heavily dependent upon specific types of applications, The CFH method has some advantage through its use of the simple bag-of-features representation and unsupervised dictionaries. This is certainly not the case for the insect-species identification application where feature locations correlate with body part information, and the focus is on differentiating very similar objects.

The HRF method demonstrated the benefit of learning a task-specific feature extraction procedures; its classification performance was slightly better than the SET method that uses multiple types of features obtained by general purpose detectors. The learnt HRFs showed that efficient use of Haar-like features and random forests achieves very low average feature-extraction times. However, experimental evaluations of the HRF method on the *EPT29* set clearly indicate that shape features are necessary to successfully discriminate between the specimens of this data set. The SSP method combines different types of local feature maintaining some of their image location information, which is undoubtedly beneficial in this species-classification task. Just as expressed in Chapter 4, a relative disadvantage of the SSP kernel method is the time expense of computing the features uti-

lized and the possibility of missing some discriminative information by using off-the-shelf feature detectors and descriptors. We follow with a detailed discussion of the findings specific to each of the methods.

Our first object-class recognition method performed reasonably well on three different stonefly-larvae identification tasks involving only four species. This method showed the promise for performing this task using computer vision techniques. We also found that combinations of dissimilar features can be beneficial to boost species identification accuracy rates. The CFH method attained classification error scores of 17.58% in the 4-species and 4.6% in the 3-species experiment. The classification error for the binary classification problem between Calineuria and Doroneuria was 20.63%. The error obtained in this binary task indicates that discriminative information is necessary in the codebook learning stage, and the spatial relationships between local features is important. The two main disadvantages of this method is the lack of discriminative learning of visual dictionaries and the impracticability of scaling this method to a large number of classes.

The HRF features were capable of obtaining, on average, lower error rates than the SET method, the previous best method for identifying stoneflies species. This performance gain is most notable against both the SET and CFH methods on the Cal vs Dor task in which the species are closely related. The use of scale-invariant constant-time Haar-like features to evaluate image patches at every node splitting decision increases the already efficient decision-tree evaluation procedure. The HRF method achieves an average processing time that is two orders of magnitude lower than the general-purpose detectors with similar invariance characteristics combined with region descriptor computation. Also given the related nature of the species in the *STONEFLY9* set, variation in color, shape and developmental and degradation stage are not as dominant; thus the HRF is capable of obtaining satisfactory classification rates using only highly-specialized HRF appearance features that employ the information from texture patterns of different parts of the specimens. The drawback of the HRF feature vectors is their high dimensionality, which could have a negative effect on performance and sometimes on classification accuracy. The HRF method is also clearly at a disadvantage in classification problems where shape is necessary, such as the *EPT29* data set.

The results obtained with the SSP kernel classifiers indicate that practical automated biomonitoring systems are possible. A single-stage SVM classifier with a global HOG descriptor, which is often successfully applied in object-class detection tasks, was used as a baseline comparison of the

two-level classification methods and was significantly outperformed by our method. Our stacked spatial kernel SVM method achieves higher classification rates than the stacked random trees and RBF kernel classifiers, reaching similar performance with just a single type of feature. Additionally, the unbalanced nature of the *EPT29* data set (one species has only 27 images, while other sets have more than 250) clearly has a negative effect on the performance of all classifiers. Still, we can see in the Lphlb and Per species accuracy rates that the random-trees method is most affected, and the 3-feature stacked spatial-pyramid classifier is the least affected by the small number of representatives of these two species.

These findings demonstrate the suitability of the spatial score histogram as an input feature for the stacked spatial-pyramid classifier. A theoretical drawback of the SSP feature combination procedure is that it cannot deal with identification tasks when they require multiple feature or multiple "word" combinations to represent subparts, because of the way spatial information is captured with the spatial histograms.

As a final point, given the wide inter-species appearance variation due to different 3D specimen position and orientation, natural diversity, developmental stages, and degradation after capture, we find that a diverse set of features is an important factor for EPT species identification employing the SSP method. This method benefited greatly from combining multiple different features, attaining an overall classification boost of almost 4 percentage points over the single-feature classifier with the highest accuracy.

BIBLIOGRAPHY

[1] Azizi Abdullah, Remco C. Veltkamp, and Marco A. Wiering. Spatial pyramids and two-layer stacking SVM classifiers for image categorization: a comparative study. In *IJCNN'09*, pages 1130–1137, Piscataway, NJ, USA, 2009. IEEE Press.

[2] T. Arbuckle, S. Schroder, V. Steinhage, and D. Wittmann. Biodiversity informatics in action: identification and monitoring of bee species using ABIS. In *Proc. 15th Int. Symp. Informatics for Environmental Protection*, volume 1, pages 425–430, Zurich, 2001.

[3] Nafiz Arica and Yarman Vural. BAS: a perceptual shape descriptor based on the beam angle statistics. *Pattern Recogn. Lett.*, 24(9-10):1627–1639, 2003.

[4] D. H. Ballard. Generalizing the Hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008.

[6] Anna Bosch, Andrew Zisserman, and Xavier Munoz. Representing shape with a spatial pyramid kernel. In *CIVR '07*, pages 401–408, New York, NY, USA, 2007. ACM.

[7] L. Breiman. Bias, variance, and arcing classifiers, 1996.

[8] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Chapman & Hall, New York, NY, USA, 1984.

[9] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[10] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[11] Barbara Caputo, Christian Wallraven, and Maria-Elena Nilsback. Object categorization via local kernels. In *ICPR '04: Volume 2*, pages 132–135, Washington, DC, USA, 2004. IEEE Computer Society.

[12] E.D. Chesmore and C. Nellenbach. Acoustic methods for the automated detection and identification of insects. *Acta Horticulturae*, 2001.

[13] Dmitry Chetverikov. A simple and efficient algorithm for detection of high curvature points in planar curves. In Nicolai Petkov and Michel A. Westenberg, editors, *CAIP*, volume 2756 of *LNCS*, pages 746–753. Springer, 2003.

[14] A.Y.-S. Chia, S. Rahardja, D. Rajan, and M.K. Leung. Object recognition by discriminative combinations of line segments and ellipses. In *CVPR '10 IEEE Conference on*, pages 2225 –2232, jun. 2010.

[15] G. Csurka, C. Dance, L. Fan, J. Williamowski, and C. Bray. Visual categorization with bags of keypoints. *ECCV'04 workshop on Statistical Learning in Computer Vision*, pages 59–74, 2004.

[16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR '05 Vol. 1*, pages 886–893, 2005.

[17] Thomas G. Dietterich. Ensemble methods in machine learning. *Lecture Notes in Computer Science*, 1857:1–15, 2000.

[18] M.T. Do, J.M. Harp, and K.C. Norris. A test of a pattern recognition system for identification of spiders. *Bulletin of Entomological Research*, 89(3):217–224, 1999.

[19] Richard O. Duda and Peter E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[20] Pedro F. Felzenszwalb, Ross B. Girshick, and David A. McAllester. Cascade object detection with deformable part models. In *CVPR'10*, pages 2241–2248, 2010.

[21] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proceedings of the CVPR'10*, volume 2, pages 264–271, Madison, Wisconsin, June 2003.

[22] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting, 1998.

[23] J. Gall and V. Lempitsky. Class-specific Hough forests for object detection. In *CVPR' 09*, 2009.

[24] Yuefang Gao, Hongzhi Song, Xuhong Tian, and Yan Chen. Identification algorithm of winged insects based on hybrid moment invariants. In *ICBBE 2007*, pages 531 –534, jul. 2007.

[25] Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 36(1):3–42, 2006.

[26] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV '05*, pages 1458–1465, 2005.

[27] C. Harris and M. Stephens. A combined corner and edge detector. *Alvey Vision Conference*, pages 147–151, 1988.

[28] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 90–96, 2004.

[29] Frederic Jurie and Bill Triggs. Creating efficient codebooks for visual recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, pages 604–610, Washington, DC, USA, 2005. IEEE Computer Society.

[30] T. Kadir, A. Zisserman, and M. Brady. An affine invariant salient region detector. In *European Conference on Computer Vision (ECCV04)*, pages 228–241, 2004.

[31] Niels Landwehr, Mark Hall, and Eibe Frank. Logistic model trees. *Machine Learning*, 59(1-2):161–205, 2005.

[32] Natalia Larios, Hongli Deng, Wei Zhang, Matt Sarpola, Jenny Yuen, Robert Paasch, Andrew Moldenke, David A. Lytle, Salvador Ruiz Correa, Eric N. Mortensen, Linda G. Shapiro, and Thomas G. Dietterich. Automated insect identification through concatenated histograms of local appearance features. *Mach. Vision Appl.*, 19(2):105–123, 2008.

[33] Natalia Larios, Bilge Soran, Linda G. Shapiro, Gonzalo Martínez Muñoz, Junyuan Lin, and Thomas G. Dietterich. Haar random forest features and SVM spatial matching kernel for stonefly species identification. In *ICPR*, 2010.

[34] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR '06*, pages 2169–2178, 2006.

[35] Zhenyu Li, Zuji Zhou, Zuorui Shen, and Qing Yao. Automated identification of mosquito (diptera: Culicidae) wingbeat waveform by artificial neural network. *Artificial Intelligence Applications and Innovations*, 187/2009:483–489, 2009.

[36] Rainer Lienhart and Jochen Maydt. An extended set of Haar-like features for rapid object detection. In *ICIP 2002, Vol. 1*, pages 900–903, 2002.

[37] Ce Liu, Jenny Yuen, Antonio Torralba, Josef Sivic, and William T. Freeman. SIFT Flow: Dense correspondence across different scenes. In *ECCV '08: Proceedings of the 10th European Conference on Computer Vision*, pages 28–42, Berlin, Heidelberg, 2008. Springer-Verlag.

[38] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, 2004.

[39] S.M. Lucas. Face recognition with continuous n-tuple classifier. In *Proc. British Machine Vision Conference*, pages 222–231, Essex, 1997.

[40] G. Martínez Muñoz, N. Larios, E. Mortensen, W. Zhang, A. Yamamuro, R. Paasch, N. Payet, D. Lytle, L.G. Shapiro, S. Todorovic, A. Moldenke, and T.G. Dietterich. Dictionary-free categorization of very similar objects via stacked evidence trees. In *CVPR' 09*, pages 549–556, 2009.

[41] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[42] Michael Mayo and Anna T. Watson. Automatic species identification of live moths. *Know.-Based Syst.*, 20(2):195–202, 2007.

[43] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of the European Conference on Computer Vision*, pages 128–142, 2002.

[44] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *Int. Journal Computer Vision*, 60(1):63–86, 2004.

[45] Frank Moosmann, Bill Triggs, and Frederic Jurie. Fast discriminative visual codebooks using randomized clustering forests. In *NIPS 19*, pages 985–992. MIT Press, Cambridge, MA, 2007.

[46] M. A. O'Neill, I. D. Gauld, K. J. Gaston, and P. Weeks. DAISY: an automated invertebrate identification system using holistic vision techniques. In *BioNET-Intl. Group for Computer-Aided Taxonomy (BIGCAT)*, pages 13–22, Egham, 2000.

[47] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV '04*, pages 71–84, 2004.

[48] M. Ovsjanikov, A.M. Bronstein, M.M. Bronstein, and L.J. Guibas. Shape Google: a computer vision approach to isometry invariant shape retrieval. In *NORDIA'09*, pages 320–327, 2009.

[49] Constantine Papageorgiou and Tomaso Poggio. A trainable system for object detection. *Int. J. Comput. Vision*, 38(1):15–33, 2000.

[50] J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[51] M. J. Sarpola, R. K. Paasch, E. N. Mortensen, T. G. Dietterich, D. A. Lytle, A. R. Moldenke, and L. G. Shapiro. An aquatic insect imaging system to automate insect classification. In *Biological Engineering Division of ASABE*. ASABE, 2008.

[52] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, December 2001.

[53] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR '08*, pages 1–8, 2008.

[54] Kah Kay Sung and Tomaso Poggio. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51, 1998.

[55] S.U. Thiel and J.A. Ware. Determination of water quality in fresh water lakes. In *Image Processing and its Applications, 1995., Fifth International Conference on*, pages 662 –666, jul. 1995.

[56] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[57] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of the British Machine Vision Conference*, pages 412–425, 2000.

[58] T. Tuytelaars and L. Van Gool. Matching widely separated views based on affine invariant regions. *Int. Journal Computer Vision*, 59(1):61–85, 2004.

[59] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.

[60] Chenglu Wen, Daniel E Guyer, and Wei Li. Automated insect classification with combined global and local features for orchard management. In *ASABE*, 2009.

[61] Sean Michael White, Dominic Marino, and Steven Feiner. Designing a mobile user interface for automated species identification. In *SIGCHI '07*, pages 291–294, NY, USA, 2007. ACM.

[62] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1800–1807, Washington, DC, USA, 2005. IEEE Computer Society.

[63] John Winn and Jamie Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR '06*, pages 37–44, 2006.

[64] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *CVPR'08*, pages 1–8, 2008.

[65] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, 73(2):213–238, 2007.

[66] Wei Zhang, Hongli Deng, Thomas G. Dietterich, and Eric N. Mortensen. A hierarchical object recognition system based on multi-scale principal curvature regions. In *ICPR'06*, pages 1475–1490, 2006.

[67] Wei Zhang and Thomas G. Dietterich. Learning visual dictionaries and decision lists for object recognition. In *ICPR'08*, pages 1–4, 2008.

[68] Wei Zhang, Akshat Surve, Xiaoli Fern, and Thomas Dietterich. Learning non-redundant codebooks for classifying complex objects. In *ICML '09*, pages 1241–1248, 2009.

Appendix A

EPT29 SPECIES CODE LIST

Table A.1: *EPT29* listing of species code, insect order, family, and *genus*.

Code	Order	Family	*Genus*
Amphin	Plecoptera	Nemouridae	*Amphinemura*
Asiop	Ephemeroptera	Leptohyphidae	*Asioplax*
Atops	Trichoptera	Hydrobiosidae	*Atopsyche*
Baets	Ephemeroptera	Baetidae	*Baetis*
Calib	Ephemeroptera	Baetidae	*Callibaetis*
Camel	Ephemeroptera	Baetidae	*Camelobaetidius*
Capni	Plecoptera	Capniidae	*Capnia*
Cerat	Trichoptera	Hydropsychidae	*Ceratopsyche*
Cinyg	Ephemeroptera	Heptageniidae	*Cinygmula*
Cla	Plecoptera	Perlidae	*Claassenia*
Culop	Trichoptera	Glossosomatidae	*Culoptila*
Drunl	Ephemeroptera	Ephemerellidae	*Drunella*
Epeor	Ephemeroptera	Heptageniidae	*Epeorus*
Fallc	Ephemeroptera	Baetidae	*Fallceon*
Hlpsy	Trichoptera	Helicopsychidae	*Helicopsyche*
Isogn	Plecoptera	Perlodidae	*Isogenoides*
Kat	Plecoptera	Chloroperlidae	*Kathroperla*
Leucr	Ephemeroptera	Heptageniidae	*Leucrocuta*
Limne	Trichoptera	Limnephilidae	*Limnephilus*
Lpdst	Trichoptera	Lepidostomatidae	*Lepidostoma*
Lphlb	Ephemeroptera	Leptophlebiidae	*Leptophlebia*
Meg	Plecoptera	Perlodidae	*Megarcys*
Mscap	Plecoptera	Capniidae	*Mesocapnia*
Per	Plecoptera	Perlodidae	*Perlinodes*
Plmpl	Plecoptera	Chloroperlidae	*Plumiperla*
Siphl	Ephemeroptera	Siphlonuridae	*Siphlonurus*
Skw	Plecoptera	Perlodidae	*Skwala*
Sol	Plecoptera	Peltoperlidae	*Soliperla*
Taenm	Plecoptera	Taeniopterygidae	*Taenionema*

Appendix B

DATA, SOURCE, AND BINARY FILE LOCATIONS

The binary files for the feature extraction of Chapter 3 are available at the Oxford vision group website in the corresponding authors' page. The source for the LMT classifier of Chapter 3 was obtained from the WEKA API site of the University of Waikato. The feature extraction methods and classifiers of Chapter 4 and Chapter 5 were implemented with the OpenCV library. The source of this library is hosted in Sourceforge. The stonefly and EPT data sets can be obtained from the Orogen State Bug-ID project page.

You can find the latest versions of these files and data in the following locations:

- Hessian and Harris affine detectors

  `www.robots.ox.ac.uk/~vgg/research/affine/`

- Kadir saliency detector

  `www.robots.ox.ac.uk/~timork/salscale.html`

- WEKA

  `www.cs.waikato.ac.nz/ml/weka/`

- OpenCV

  `sourceforge.net/projects/opencvlibrary/`

- Bug-ID project

  `web.engr.oregonstate.edu/~tgd/bugid/`

VITA

Natalia Larios is a graduate student in the department of Electrical Engineering at the University of Washington where she obtained a master's degree and where she is currently pursuing a PhD degree. She was born in Mexico where she earned a bachelor's degree in computer engineering from the National University of Mexico (UNAM) as member of the honors program of the school of engineering. Her research is focused on solving object-class recognition, pattern recognition, discriminative feature extraction, and related computer vision research problems by applying machine learning, probabilistic modeling, and optimization techniques. Her advisor is computer vision researcher Linda Shapiro, Professor of Computer Science and Engineering and of Electrical Engineering. Natalia currently works on a project led by Professor Thomas Dietterich; which aims to automate insect-species identification for environment monitoring purposes using digital images.