Automated Insect Identification through Concatenated Histograms of Local Appearance Features

N. Larios[1], H. Deng[2], W. Zhang[2], M. Sarpola[2], J. Yuen[3], R. Paasch[2], A. Moldenke[2], D. A. Lytle[2], S. Ruiz Correa[4], E. Mortensen[2], L. G. Shapiro[1], T. G. Dietterich[2]
[1] University of Washington, [2] Oregon State University,
[3] Massachusetts Institute of Technology, [4] Children's National Medical Center
elarios@ee.washington.edu, {deng, zhangwe, enm, tgd}@eecs.oregonstate.edu,
matts@videx.com, jenny@csail.mit.edu, paasch@engr.orst.edu, SRCorrea@cnmc.org,
{moldenka, lytleda}@science.oregonstate.edu, shapiro@cs.washington.edu

Abstract

This paper describes a fully automated stonefly-larvae classification system using a local features approach. It compares the three region detectors employed by the system: the Hessian-affine detector, the Kadir entropy detector and a new detector we have developed called the principal curvature based region detector (PCBR). It introduces a concatenated feature histogram (CFH) methodology that uses histograms of local region descriptors as feature vectors for classification and compares the results using this methodology to that of Opelt [11] on three stonefly identification tasks. Our results indicate that the PCBR detector outperforms the other two detectors on the most difficult discrimination task and that the use of all three detectors outperforms any other configuration. The CFH methodology also outperforms the Opelt methodology in these tasks.

1. Introduction

Population counts of stonefly (Plecoptera) larvae and other aquatic insects inhabiting stream substrates are known to be a sensitive and robust indicator of stream health and water quality. Biomonitoring using aquatic insects has been employed by federal, state, local, tribal, and private resource managers to track changes in river and stream health and to establish baseline criteria for water quality standards. However, the sorting and identification of insect specimens can be extremely time consuming and requires substantial technical expertise. Thus aquatic insect identification is a major technical bottleneck for large-scale implementation of biomonitoring. Larval stoneflies are especially important for biomonitoring, because they are sensitive to reductions in water quality caused by thermal pollution, eutrophication, sedimentation, and chemical pollution.

In addition to its practical importance, the automated recognition of stoneflies raises many fundamental computer vision challenges. Stonefly larvae are highly-articulated objects with many sub-parts (legs, antennae, tails, wing pads, etc.) and many degrees of freedom. Some taxa exhibit interesting patterns on their dorsal sides, but others are not patterned. Some taxa are distinctive; others are very difficult to identify. Finally, as the larvae repeatedly molt, their size and color change. This variation in size, color, and pose means that simple computer vision methods that rely on placing all objects in a standard pose cannot be applied. Methods that can handle significant variation in pose, size, and coloration are needed.

To address these challenges, we based our method on a bag-of-features approach [3, 11] to classification, which extracts a bag of region-based features from the image without regard to their relative spatial arrangement. The features are then summarized as a feature vector and classified via state-of-the-art machine learning methods. Our approach involves five stages: (a) region detection, (b) region description, (c) region classification into features, (d) combination of detected features into a feature vector, and (e) final classification of the feature vector. For region detection, we employ three different algorithms: the Hessian-affine detector [9], the Kadir entropy detector [5], and a new detector that we have developed called the principal curvature-based region detector (PCBR). All detected regions are described using Lowe's SIFT descriptor [7]. At training time, a Gaussian mixture model (GMM) is fit to the set of SIFT vectors, and each mixture component is taken to be a feature. At classification time, each SIFT vector is assigned to the most likely feature and a histogram consisting of the number of SIFT vectors assigned to each feature is formed. Feature

vectors from each of the separate detectors are concatenated to produce a larger vector when working with combinations of detectors. The final labeling of the specimens is based on these feature vectors and is performed by an ensemble of logistic model trees [6].

In related work, the ABIS system [1] performs identification of bees based on features extracted from their forewings. It uses both geometric features (lengths, angles and areas) and appearance features. It requires manual positioning of the insect and prior expert knowledge about the forewings. DAISY [10] is a general-purpose identification system that has been applied to several arthropod identification tasks. It uses appearance features and a random n-tuple classifier (NNC) [8]. It requires user interaction for image capture and segmentation. SPIDA-web [4] is an automated identification system that applies neural networks for species classification from wavelet encoded images. SPIDA-web's feature vector is built from a subset of the components of the wavelet transform using the Daubechines 4 function. The spider specimen has to be manipulated by hand, and the image capture, preprocessing and region selection also require direct user interaction.

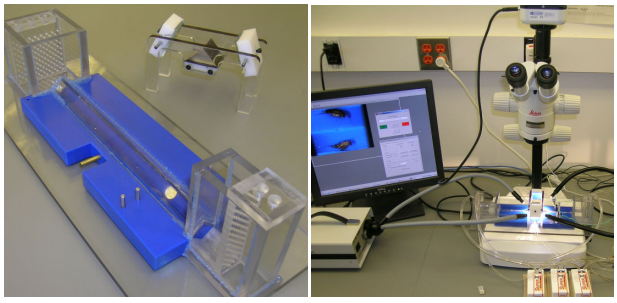
The goal of our work is to provide a rapid-throughput system for classifying stonefly larvae to the species level. To achieve this, we have developed a system that combines a mechanical apparatus for manipulating and photographing the specimens with a software system for processing and classifying the resulting images. Section 2 describes the full stonefly identification system, and Section 3 describes a large set of experiments and results.

2. Stonefly Identification System

The stonefly identification system consists of a mechanical apparatus for automated mechanical manipulation and imaging of the specimens and a software system consisting of local feature detection, description, classification and combination into a feature vector that is used by an ensemble of logistic model trees to identify the specimen.

2.1. Automated Mechanical Manipulation and Imaging of Stonefly Larvae

Figure 1(a) shows the mechanical apparatus, which consists of two alcohol reservoirs connected by an alcohol-filled tube. A specimen is manually inserted into the plexiglass well shown at the right edge of the figure and pumped through the tube. Infrared detectors positioned part way along the tube detect the passage of the specimen and cut off the pumps. Then a side fluid jet "captures" the specimen in the field of view of the microscope. When power to this jet is cut off, the specimen settles to the bottom of the tube where it is photographed by a QImaging MicroPublisher 5.0 RTV 5 megapixel color digital camera, which is attached to a Leica MZ9.5 high-performance stereomicroscope at 0.63x magnification (Figure 1b). With this apparatus, we can image a few tens of specimens per hour. Figure 3 shows some example images obtained using this stonefly imaging assembly.



(a) (b)

Figure 1. (a) Prototype mirror and transportation apparatus. (b) Entire stonefly transportation and imaging setup (with microscope and attached digital camera, light boxes, and computer controlled pumps for transporting and rotating the specimen.

2.2. Feature Detection

We apply three region detectors to each image: (a) the Hessian-affine detector [9], (b) the Kadir entropy detector [5], and (c) our new PCBR detector. The PCBR detector was developed to complement previous detectors which use the gradient orientation information directly from the image. Our method performs a watershed segmentation of an eigenimage of the principal curvatures of the insect image to obtain useful regions. The algorithm, which is illustrated in Figure 2, can be summarized as follows:

1. Compute the Hessian matrix image describing each pixel's local image curvature.

2. Form the principal curvature image by extracting the largest positive eigenvalue from each pixel's Hessian matrix (Figure 2b).

3. Apply a gray scale morphological close on the principal curvature image to remove noise and threshold the resulting image to obtain a "clean" binary principal curvature image (Figure 2c).

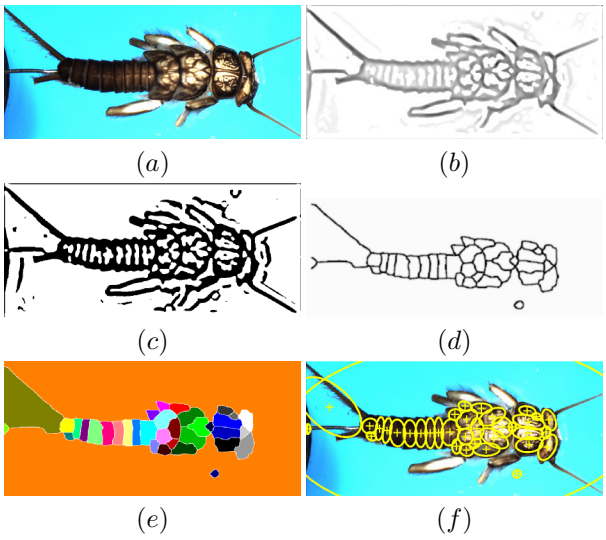


Figure 2. Regions defined by principal curvature. (a) the original image, (b) principal curvature magnitude, (c) binary curvatures, (d) watershed boundaries, (e) watershed regions, (f) fitted elliptical regions.

4. Segment the clean image into regions using the watershed transform (Figures 2d and 2e).

5. Fit an ellipse to each watershed region to produce the detected interest regions (Figure 2f).

Each detected region is represented by a SIFT vector using Mikolajczyk's modification to the binary code distributed by Lowe [7].

2.3. Training and Classification

Our approach to classification of stonefly larvae closely follows the "bag of features" approach but with several modifications and extensions. Tables 1, 2, and 3 provide pseudo-code for our method. The training process requires two sets of images, one for defining the dictionaries and one for training the classifier. In addition, to assess the accuracy of the learned classifier, we need a holdout test data set. Therefore, we begin by partitioning the data at random into three subsets: clustering, training, and testing.

Dictionary Construction We construct a separate dictionary for each region detector d and each class k (Table 1). Let $S_{d,k}$ be the SIFT descriptors that were found by applying detector d to images from class k. We fit a Gaussian

Table 1. Dictionary Construction. D is the number of region detectors (3 in our case), and K is the number of stonefly taxa to be recognized (4 in our case).

Dictionary Construction

For each detector $d = 1, \ldots, D$
  For each class $k = 1, \ldots, K$
    Let $S_{d,k}$ be the set of SIFT vectors that results from applying detector d to all cluster images from class k.
    Fit a Gaussian mixture model to $S_{d,k}$ to obtain a set of mixture components $\{C_{d,k,\ell}\}, \ell = 1, \ldots, L$.
    The GMM estimates the probability of each SIFT vector $\mathbf{s} \in S_{d,k}$ as
$$P(\mathbf{s}) = \sum_{\ell=1}^{L} C_{d,k,\ell}(\mathbf{s} \mid \mu_{d,k,\ell}, \mathbf{\Sigma}_{d,k,\ell}) P(\ell).$$
    where $C_{d,k,\ell}$ is a multi-variate Gaussian distribution with mean $\mu_{d,k,\ell}$ and diagonal covariance matrix $\mathbf{\Sigma}_{d,k,\ell}$.

    Define the keyword mapping function
$key_{d,k}(\mathbf{s}) = \text{argmax}_\ell \; C_{d,k,\ell}(\mathbf{s} \mid \mu_{d,k,\ell}, \mathbf{\Sigma}_{d,k,\ell})$

mixture model (GMM) via the Expectation-Maximization (EM) algorithm to $S_{d,k}$. The GMM has the following form:

$$p(\mathbf{s}) = \sum_{\ell=1}^{L} C_{d,k,\ell}(\mathbf{s} \mid \mu_{d,k,\ell}, \mathbf{\Sigma}_{d,k,\ell}) P(\ell)$$

where $\mathbf{s}$ denotes a SIFT vector, the component probability distribution $C_{d,k,\ell}$ is a multivariate Gaussian density function with mean $\mu_{d,k,\ell}$ and covariance matrix $\mathbf{\Sigma}_{d,k,\ell}$, and $\mathbf{\Sigma}_{d,k,\ell}$ is constrained to be a diagonal matrix. The number of components L is a parameter that must be specified. Each fitted component defines a keyword. A virtue of employing a model-based clustering method such as GMM is that it can be viewed as a classifier. Given a new SIFT vector $\mathbf{s}$, we compute the corresponding keyword $\ell = key_{d,k}(s)$ by finding the ℓ that maximizes $p(\mathbf{s} \mid \mu_{d,k,\ell}, \mathbf{\Sigma}_{d,k,\ell})$. Note that we disregard the mixture probabilities $P(\ell)$. This is equivalent to mapping $\mathbf{s}$ to the nearest cluster center μ_ℓ under the Mahalobis distance defined by Σ_ℓ.

The GMM fitting is initialized with the centers of mass of the clusters obtained by the k-means algorithm. The k-means algorithm is initialized by picking random elements. The EM steps are performed either until the error fitting the GMM is less than 0.05% from the previous iteration or 100 itrations are performed.

Table 2. Feature Vector Construction. The histograms are generated by employing the $key_{d,k}$ dictionary keys.

Feature Vector Construction

To construct a feature vector for an image:
For each detector $d = 1, \ldots, D$
  For each class $k = 1, \ldots, K$
    Let $H_{d,k}$ be the keyword histogram for detector d and class k
    Initialize $H_{d,k}[\ell] = 0$ for $\ell = 1, \ldots, L$
    For each SIFT vector $\mathbf{s}$ detected by detector d
      increment $H_{d,k}[key_{d,k}(\mathbf{s})]$
Let H be the concatenation of the $H_{d,k}$ histograms for all d and k.

Feature Vector Construction After constructing the dictionaries from the clustering image set, the next step is to construct a set of training examples from the training image set (Table 2). To accomplish this, the three region detectors are applied to each training image. Each region found by detector d is represented as a SIFT vector and then mapped to an appropriate keyword for each class k using $key_{d,s}$. These keywords are accumulated to form a histogram $H_{d,k}$, and these histograms are then concatenated to produce the final feature vector. With D detectors, K classes, and L mixture components, there are $D \times K \times L$ elements in the final feature vector.

Training and Classification Once the training set is constructed, the next step is to train the classifier (Table 3). We employ a state-of-the-art ensemble classification method: bagged logistic model trees. Logistic model trees (LMT) were developed by Landwehr, Hall, and Frank [6]. Bagging [2] is a general method for constructing an ensemble of classifiers. Given a set T of labeled training examples and a desired ensemble size B, it constructs B bootstrap replicate training sets T_b, $b = 1, \ldots, B$. 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 new image, and the predictions vote to determine the overall prediction.

3. Experiments and Results

We collected 263 specimens of four stonefly taxa from freshwater streams in the mid-Willamette Valley and Cascade Range of Oregon: the species *Calineuria californica* (Banks), the species *Doroneuria baumanni* Stark & Baumann, the species *Hesperoperla pacifica* (Banks), and the genus *Yoraperla*. Each specimen was placed in its own vial with an assigned control number and photographed using the apparatus described in Section 2.1. Approximately twenty images were obtained of each specimen, which yields 20 individual images. These were then manually examined, and all images that gave a dorsal view within 30 degrees of vertical were selected for analysis. It takes about 5 minutes for a specimen to be loaded into our imaging apparatus, transported to the imaging area, reoriented and imaged 5 times, transported to the catch basin and reinserted into its bottle. On average per image, our segmentation step takes 15.6 seconds, feature extraction takes 43.3 seconds (of which 42 seconds is for the PCBR detector), and classification takes only 6.5 ms. The code for the PCBR detector is in the process of being optimized. The mechanical specimen handling is currently the limiting factor in the system throughput.

Figure 3 shows some of the images collected for the study. Note the variety of colors, sizes, and poses. Note also that *Yoraperla* is quite distinctive in color and shape. The other three taxa are quite similar to each other, and the first two (*Calineuria* and *Doroneuria*) are exceedingly dif-

Table 3. Training and Classification. B is the number of bootstrap iterations (i.e., the size of the classifier 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 (i.e., stonefly species).
For bootstrap replicate $b = 1, \ldots, B$
  Construct training set T_b by sampling N training examples randomly with replacement from T
  Let LMT_b be the logistic model tree fitted to T_b

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

ficult to distinguish. This is emphasized in Figure 4, which shows closeup dorsal views. To verify the difficulty of discriminating these two taxa, we conducted an experiment that tested the ability of humans to separate *Calineuria* and *Doroneuria*. A total of 26 students and faculty from Oregon State University were allowed to train on 50 randomly-selected images of these two species, and were subsequently tested with another 50 images. Most of the subjects (21) had some prior entomological experience. The mean score was 78.6% correctly identified (std. dev. = 8.4). There was no statistical difference between the performance of entomologists and non-entomologists (Wilcoxon two-sample test, $W = 57.5, p \leq 0.5365$).

Given the characteristics of the taxa, we defined three discrimination tasks, which we term CDHY, JtHY, and CD as follows:

CDHY: Discriminate among all four taxa

JtHY: Merge *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*

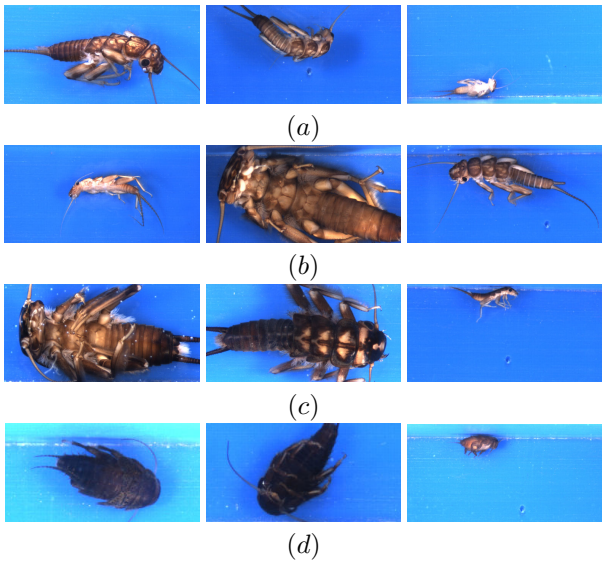


Figure 3. Example images of different stonefly larvae species. (a) *Calineuria*, **(b)** *Doroneuria*, **(c)** *Hesperoperla* **and (d)** *Yoraperla*.

Performance on all three tasks was evaluated via threefold cross-validation. The images were randomly partitioned into three sets of approximately equal size 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 the class frequencies were the same across the three partitions. In each "fold" of the cross-validation, one partition served as the clustering data set for defining the dictionaries, a second partition served as the training data set, and the third partition served as the test set.

Figure 4. Comparison images: *Calineuria* **details (left) and** *Doroneuria* **details.**

3.1. Overall Results

Our experiments were designed to achieve two goals. First, we wanted to determine how well our combined method (with three region detectors) could perform on the three recognition tasks. To establish a basis for evaluation, we also applied the method of Opelt, et al., [11], which is one of the best current object recognition systems, and compared our results to theirs. Second, we wished to evaluate the contribution of each of the three region detectors to the performance of the system. To achieve this second goal, we trained our system in 7 different configurations corresponding to training with all three detectors, all pairs of detectors, and all individual detectors.

Table 4. Percentage of images correctly classified for our system with all three region detectors along with 95% confidence intervals.

Task	Accuracy[%]
CDHY	82.42 ± 2.12
JtHY	95.40 ± 1.16
CD	79.37 ± 2.70

Table 4 shows the classification rates achieved by our combined method on the three discrimination tasks. Table 5 shows the confusion matrices for the three tasks. On the CDHY task, our system achieves 82% correct classifications. As expected, the main difficulty is to discriminate *Calineuria* and *Doroneuria*. On this binary classification

task, our method attains 79% correct classification, which is approximately equal to the mean for human subjects with some prior experience. When these two classes are pooled in the JtHY task, performance reaches 95% correct, which is excellent.

Table 5. Confusion matrices of the combined Kadir, Hessian-Affine and PCBR detectors for the three tasks. (a) CDHY (b) JtHY and (c) CD.

predicted as ⇒	Cal.	Dor.	Hes.	Yor.
Calineuria	315	79	6	0
Doroneuria	80	381	2	0
Hesperoperla	24	22	203	4
Yoraperla	1	0	0	123

(a)

predicted as ⇒	Joint CD	Hes.	Yor.
Joint CD	857	5	1
Hesperoperla	46	203	4
Yoraperla	0	1	123

(b)

predicted as ⇒	Calineuria	Doroneuria
Calineuria	304	96
Doroneuria	82	381

(c)

In order to test the performance of the CFH methodology combined with LMTs, we also applied a competing method [11] to the difficult CD task using the same image features. Opelt's method is similar to ours in that it is also based on ensemble learning principles (AdaBoost) and also able to combine multiple feature types for classification. We adapted Opelt's Matlab implementation to our features and followed the default setting of the parameters given in the paper. Euclidean distance is used as the distance metric for the SIFT features, and number of iterations T = 100. The classification rates are summarized in Table 6. As we can see, for all four combinations of detectors, our method outperforms Opelt's method by 8–12 percentage points.

3.2. Results for Multiple Region Detectors

Table 7 summarizes the results of applying all combinations of one, two, and three detectors to the CDHY, JtHY, and CD tasks. The first three lines show that each detector has unique strengths when applied alone. The Hessian-affine detector works best on the 4-class CDHY task; the Kadir detector is best on the 3-class JtHY task, and the PCBR detector gives the best 2-class CD results. On the pairwise experiments it appears that the Hessian-affine and PCBR complement each other well. The best pairwise results for the JtHY task is obtained by the Kadir-Hessian pair; which appears to be better for tasks that require an overall assessment of the shape. Finally, the combination of all three detectors gives the best results on each task.

Table 6. CD classification rates comparison of Opelt's method and CFH when applied with different combinations of detectors. A $\surd$ indicates that the corresponding detector was used.

Hessian affine	Kadir entropy	PCBR	Accuracy[%]	
			Opelt [11]	CHF & LMT
$\surd$			60.59	70.10
	$\surd$		62.63	70.34
		$\surd$	67.86	79.03
$\surd$	$\surd$	$\surd$	70.10	79.37

The PCBR detector is very stable, although it does not always identify all of the relevant regions. The Kadir detector is also stable, but it finds a very large number of regions, most of which are not relevant. The Hessian-affine detector finds very good small-scale regions, but its larger-scale detections are not useful for classification. The PCBR detector focuses on the interior of the specimens, whereas the other detectors (especially Kadir) tend to find points on the edges between the specimens and the background. In addition to concentrating on the interior, the regions found by the PCBR detector are more "meaningful" in that they correspond better to body parts. This may explain why the PCBR detector did a better job on the CD task.

Table 7. Classification rates of the CFH method when applied with different combinations of detectors. A $\surd$ indicates that the corresponding detector was used.

Hessian affine	Kadir entropy	PCBR	Accuracy[%]		
			CDHY	JtHY	CD
$\surd$			73.14	90.32	70.10
	$\surd$		70.64	90.56	70.34
		$\surd$	71.69	86.21	79.03
$\surd$	$\surd$		78.14	94.19	74.16
$\surd$		$\surd$	80.48	93.79	78.68
	$\surd$	$\surd$	78.31	92.09	68.83
$\surd$	$\surd$	$\surd$	82.42	95.40	79.37

4. Summary

This paper has presented a combined hardware-software system for rapid-throughput classification of stonefly larvae. The goal of the system is to perform cost-effective biomonitoring of freshwater streams. To this end, the apparatus is capable of nearly unassisted manipulation and photographing of stonefly specimens, and it obtains images of consistently high quality. The generic object recognition algorithms attain classification accuracy that is sufficiently good (82% for 4-classes; 95% for 3-classes) to support the application. By rejecting for manual classification the specimens in which the confidence level is not high enough; only a reasonable 30% of the samples would require further processing while the remaining identified specimens can reach an accuracy above 90% on all the defined tasks. The PCBR detector is particularly useful for discriminating between the most similar species and works well in combination with the Hessian-affine and Kadir detectors in all tasks. Finally, our CHF methodology was found to outperform Opelt's in all tasks.

Acknowledgments

We wish to thank Andreas Opelt for providing the Matlab code of his PAMI'06 method for the comparison experiment. We also wish to thank Asako Yamamuro and Justin Miles for their assistance with the dataset stonefly identification.

References

[1] 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.

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

[3] G. Csurka, C. Bray, and C. D. L. Fan. Visual categorization with bags of keypoints. *ECCV workshop*, 2004.

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

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

[6] N. Landwehr, M. Hall, and E. Frank. Logistic model trees. *Mach. Learn.*, 59(1-2):161–205, 2005.

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

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

[9] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.

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

[11] A. Opelt, A. Pinz, M. Fussenegger, and P. Auer. Generic object recognition with boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(3):416–431, 2006.

[12] C. Steger. An unbiased detector of curvilinear structures. *PAMI*, 20(2):113–125, 1998.