Haar Random Forest Features and SVM Spatial Matching Kernel for Stonefly Species Identification

N. Larios*, B. Soran*, L. G. Shapiro*, G. Martínez-Muñoz†, J. Lin‡, and T. G. Dietterich‡
*University of Washington
Seattle,WA 98195
Emails: {nlarios@u, bilge@cs, shapiro@cs}.washington.edu
‡Universidad Autónoma de Madrid
Madrid, Spain, Email: gonzalo.martinez@uam.es
§School of Electrical Engineering & Computer Science
Oregon State University, Corvallis, OR 97331
Email: {linju,tgd}.@eecs.oregonstate.edu

***Abstract*—This paper proposes an image classification method based on extracting image features using Haar random forests and combining them with a spatial matching kernel SVM. The method works by combining multiple efficient, yet powerful, learning algorithms at every stage of the recognition process. On the task of identifying aquatic stonefly larvae, the method has state-of-the-art or better performance, but with much higher efficiency.**

***Keywords*-object-class recognition; machine learning; SVM; Random Forests; Haar-like features;**

I. Introduction

Image classification 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. A typical classification methodology begins with detection of interesting points or regions [1], encodes them to obtain a descriptor [2], and uses the descriptors (and sometimes their spatial locations) for classification. The evaluation of random forests (RF) [3] [4] has been proposed as a way of speeding up classification feature extraction [5] and segmentation [6].

Insect classification is important for monitoring ecosystem health and for supporting ecological research. Previous work on insect identification includes the ABIS system [7] for bee identification, the work of Wen et al. [8] on identification of orchard pests, and some systems that perform species identification from acoustic data (songs, wing-beat frequencies, etc.) [9]. Automated stonefly identification from images has been the subject of several recent papers [10] [11] [12]. All of these papers used multiple low-level features encoded with SIFT descriptors [2]. In [10], an occurrence histogram of EM cluster assignments was generated from the SIFT vectors and used in classification by a bagging ensemble of decision trees. In [12], evidence trees were employed, and in [11], multiple non-redundant codebooks were used. All of these approaches are computationally intensive, due to the feature extraction process.

This paper describes a new image classification methodology designed to improve classification efficiency while achieving high accuracy. The method is based on powerful yet efficient feature extraction and a kernel for SVM classification that combines the evaluation of Haar-like features [13] of image patches and their image positions. The bag-of-words approach for image classification is extended to discriminative feature extraction with a RF that evaluates Haar-like features on densely sampled image patches. This is able to capture highly-detailed information while reducing training and classification feature computation time. The depth levels of each RF tree form a semantically meaningful implicit cluster hierarchy. Each extracted image patch is represented by its CIELab color channels and by image planes containing the bins to accumulate the gradient magnitudes of a particular orientation range. Additional accuracy gains are obtained by pairing the learning of each Haar-like feature with the image "channel" that they will be evaluated on. The generated histograms are adapted to be evaluated by a combined feature and spatial pyramid-match kernel. The aim of our kernel is to correlate the geometric correspondence of the window locations in image space with the occurrence count in each tree node from two different images.

II. Overview

Our approach has 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. The stonefly images were captured through an automated process 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 head facing left. Next, rectangular patches of three different sizes, each relative to the insect size, are extracted. From each patch, 12 feature bands, each 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 $180\,^\circ$).

A random sampling of labeled patches from the training images is used to generate the Haar random forest. Using bootstrap sampling, M subsets of the set of training patches are produced, and each subset is used to construct a decision tree for deciding if a given image patch comes from a particular insect species or not. At each branch node, one Haar feature along with one of the twelve bands is chosen as the decision mechanism. The resulting M trees form the Haar random forest.

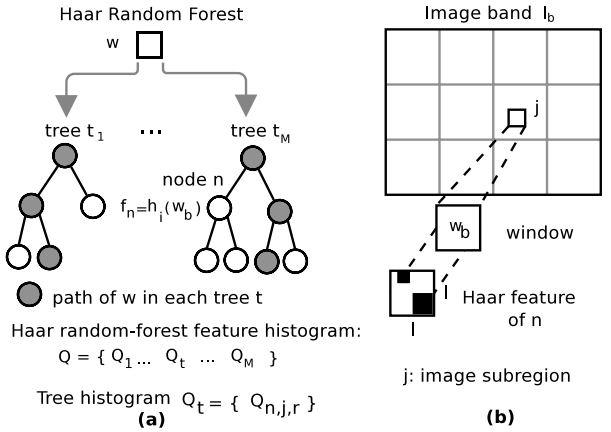

Figure 1. (a) HRF consisting of M trees with splitting function f at each internal node evaluated on image channel window w_b. (b) Haar feature h_i of $l \times l$ size.

The HRF works as a discriminative codebook to define image feature vectors (histograms) that are then employed to train an SVM classifier. The codebook extraction phase uses a regular sampling of patches from the training images, rather than the random subsets that were used to create the random forest. Square patches are sampled in three scale ranges: from the base length of the Haar-like feature mask ($l = 26$ pixels) to 0.1% of the stonefly image length and in the $[0.1 - 0.2)$ and $[0.2 - 0.3)$ ranges relative to image length. For some few small images where 0.1% image length is smaller than 26 pixels, the initial scan is replaced by a single pass extracting $l \times l$ patches. In this phase, for each training image, the patches are put through each of the M trees of the random forest. This generates, at each node of each tree, a count of how many patches passed through that node. In addition, to find spatial correspondences, a count of how many of these patches came from each image *subregion* at each scale is kept. Thus a 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, one from each tree, are extracted, concatenated to produce a single feature vector per training image, and employed to train an SVM image classifier. This paper describes HRF generation and image feature extraction (Section III), a HRF-feature spatial matching kernel (Section IV), and our experiments and results (Section V).

III. HRF Generation & Feature Extraction

A Haar random forest is an ensemble of M decision trees learned by a randomized learning algorithm [14]. 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 the patch arrives at a leaf node. Figure 1(a) illustrates a HRF with the highlighted path followed by the evaluation of a patch w through the nodes it visits. Figure 1(b) shows a Haar-like feature h_i from the set $\mathcal{H}^{l\times l}$; the set of extended Haar-like features [13] of $l \times l$ size applied to a patch w_b.

Haar Random Forest Generation: The training set $\mathcal{W}$ consists of pairs $(\boldsymbol{x}, y)$ where $\boldsymbol{x} = \{w_1, \dots w_B\}$ is a feature vector formed by the patches of the training sample window from B different image transforms, and $y \in \{0, 1\}$ is the sample class label. The transforms include the three channels of the CIELab color space and nine gradient-orientation bin image 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 spliting 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 f' that maximizes the information gain of 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'.

$$\mathcal{W}^{l_n} = \{w \in \mathcal{W}^n | f' < \tau'\}, \tag{1}$$

$$\mathcal{W}^{r_n} = \mathcal{W}^n \backslash \mathcal{W}^{l_n} \tag{2}$$

The information gain for a particular f' is

$$\Delta H_{f'} = -\frac{|\mathcal{W}^{l_n}|}{|\mathcal{W}^n|} H(y|\mathcal{W}^{l_n}) - \frac{|\mathcal{W}^{r_n}|}{|\mathcal{W}^n|} H(y|\mathcal{W}^{r_n}), \tag{3}$$

where $H(y|\mathcal{W})$ is the Shannon entropy of the label variable y given the samples from set $\mathcal{W}'$. The recursive 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.

HRF Image Feature Extraction: The HRF feature extraction process represents the image as a histogram Q. Each bin $Q[\eta, j, r]$ in Q corresponds to an image subregion j of spatial level r and index η across all the nodes n of the HRF trees. The histogram is computed by scanning a sliding window across the image (represented as a set of channels as described above). To obtain insect-part information at different sizes, the three different window-scale ranges mentioned in Section II are computed. At each sliding window scale, a HRF trained over patches extracted at that scale range is employed. The histograms from each scale are concatenated as 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 increases the count of the 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 patch counts $Q[\eta, j, r]$. 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 $Q[\eta, j, r] = Q[l_\eta, j, r] + Q[r_\eta, j, r]$.

Feature Vector Dimensionality: The HRF learning has a fixed limit on split nodes per tree $N_t = 70$, and a total fixed number of split nodes of the forest $N_{rf} = 700$. 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 approximately 11360 independent dimensions. This is large, but similar to vectors in current computer vision systems [15] [16], which show no over-fitting issues.

IV. Matching-Kernel SVM Classifier

Our method uses 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 [17]; it returns a similarity measure between image histograms and an approximate correspondence between two sets of elements. We extend this framework by combining methods using spatial [18] and tree based partitionings [16].

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

Figure 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. See [12] for the full species names.

$$\tilde{K}_t(P, Q) = \sum_{d=0}^{D-1} \frac{1}{2^{D-d}} \mathscr{S}_{d+1}(P, Q), \tag{4}$$

where d indexes across tree depth levels, D is the maximum 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

$$\mathscr{S}_d(P, Q) = \sum_{r=0}^{R-1} \frac{1}{2^{R-r}} \left(\mathcal{I}(P_r^d, Q_r^d) - \mathcal{I}(P_r^{d+1}, Q_r^{d+1}) \right). \tag{5}$$

$\mathcal{I}(P_r^d, Q_r^d)$ 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 kernel receives the vectors containing only the information at the deepest level, so all the internal split node bins are calculated and evaluated at runtime. The evaluation of these internal bins improves the similarity measure by functioning as a partial match that smoothes the harsher matches of the deepest level. 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. Furthermore, given that all the bin values in the histogram intersection distance $\mathcal{I}$ are non-negative, they can be pre-multiplied by their corresponding weight. 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 $\sum_k \min(Pweighted_k, Qweighted_k)$.

V. Experiments

To assess the performance of our method, we compared our algorithm with the best stonefly-species classifier in the literature [12], which uses stacked-evidence trees (SET). These experiments were performed on the STONEFLY9 dataset, which consists of 3826 images obtained by imaging 773 stonefly larvae specimens (See [12] for more details of the dataset). Figure 2 contains example images with individuals of each of the nine species to illustrate the challenges

posed by this classification task. Table I shows the results obtained in nine binary-classification experiments defined for this comparison. 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), which are closely related. We also show the relevance of the gradient channel information by performing these classification experiments only using the color channels (HRF Lab). The classification error with color alone is much higher than the error when using color plus gradient features. With respect to timing, our algorithm (HRF Lab+G) was compared to the SET algorithm on the overall prediction time for new test images in the cal vs. dor experiment under the same conditions. All the timing experiments were performed on a 2.8GHz processor computer with 8GB of memory. The SET algorithm requires three separate local-feature detection/decription processes ([Hessian-Affine, KB-Salient, PCBR]+SIFT), which take on average several minutes per image ($\approx$ 0.15 + 6.5 + 8.6 minutes). The average processing time of the HRF algorithm, from image load to histogram generation, is two orders of magnitude smaller (5.03 seconds) on the same data set. In both HRF and SET, the classification stage, operating only on histogram feature vectors, takes only about 0.2 seconds.

Table I
CLASSIFICATION ERROR. SET, HRF ON (LAB) COLOR CHANNELS ONLY AND WITH GRADIENT-ORIENTATION PLANES (LAB+G).

Error%	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**

VI. CONCLUSIONS

Our new HRF algorithm combines efficient low-level feature evaluation with discriminative learning, obtaining a semantic codebook-quantized image representation in a single stage. By applying learning from the initial low-level feature stage on, it is capable of obtaining lower average error rates than SET in the stonefly classification task, performing particularly well on the most difficult task. The use of scale invariant, constant-time Haar-like features in the branch nodes of the trees achieves a much lower average processing time than general region detectors with similar invariance characteristics. Future developments will include addition of spatial position information at the splitting functions and reduction of the dimensionality of the feature vectors.

REFERENCES

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

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[3] Y. Amit and D. Geman, "Shape quantization and recognition with randomized trees," *Neural Comput.*, vol. 9, no. 7, pp. 1545–1588, 1997.

[4] L. Breiman, "Random forests," in *Machine Learning*, 2001, pp. 5–32.

[5] F. Moosmann, B. Triggs, and F. Jurie, "Fast discriminative visual codebooks using randomized clustering forests," in *NIPS 19*. Cambridge, MA: MIT Press, 2007, pp. 985–992.

[6] F. Schroff, A. Criminisi, and A. Zisserman, "Object class segmentation using random forests," in *BMVC*, 2008.

[7] T. Arbuckle *et al.*, "Biodiversity informatics in action: Identification and monitoring of bee species using ABIS," in *Proc. 15th Int. Symp. Informatics for Env. Protection.*, 2001, pp. 425–430.

[8] C. Wen, D. E. Guyer, and W. Li, "Automated insect classification with combined global and local features for orchard management," in *ASABE*, 2009.

[9] D. Chesmore, "Automated bioacoustic identification of species," *Anais da Academia Brasileira de Ciências*, vol. 76, no. 2, pp. 435–440, 2004.

[10] N. Larios *et al.*, "Automated insect identification through concatenated histograms of local appearance features," *Mach. Vision Appl.*, vol. 19, no. 2, pp. 105–123, 2008.

[11] W. Zhang, A. Surve, X. Fern, and T. Dietterich, "Learning non-redundant codebooks for classifying complex objects," in *ICML '09*, 2009, pp. 1241–1248.

[12] G. Martínez Muñoz *et al.*, "Dictionary-free categorization of very similar objects via stacked evidence trees," in *CVPR' 09*, 2009, pp. 549–556.

[13] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *ICIP 2002, Vol. 1*, 2002, pp. 900–903.

[14] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 36, no. 1, pp. 3–42, 2006.

[15] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR '05 Vol. 1*, 2005, pp. 886–893.

[16] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *CVPR '08*, 2008, pp. 1–8.

[17] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *ICCV '05*, 2005, pp. 1458–1465.

[18] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR '06*, 2006, pp. 2169–2178.