Experiments with Multi-view Multi-instance Learning for Supervised Image Classification

Anonymous
Address
Email: {email}@microsoft.com

***Abstract*—In this paper we empirically investigate the benefits of multi-view multi-instance (MVMI) learning for supervised image classification. In multi-instance learning, examples for learning contain bags of feature vectors and thus data from different views cannot simply be concatenated as in the single-instance case. Hence, multi-view learning, where one classifier is built per view, is particularly attractive when applying multi-instance learning to image classification. We take several diverse image data sets—ranging from person detection to astronomical object classification to species recognition—and derive a set of multiple instance views from each of them. We then show via an extensive set of 10×10 stratified cross-validation experiments that MVMI, based on averaging predicted confidence scores, generally exceeds the performance of traditional single-view multi-instance learning, when using support vector machines and boosting as the underlying learning algorithms.**

I. Introduction

Object detection and recognition in images is an important research area, and there has always been significant interest in applying advances in machine learning to this challenging problem domain. This is particularly true for the relatively recent class of techniques developed for multi-instance learning [2], because image data can be represented naturally in multi-instance form [15], [28].

When applying multi-instance learning to this kind of data, an image is represented by a *bag* of feature vectors rather than a single vector, and classification of an image is based on this bag. Each feature vector may, for example, represent an image patch, and the bag overall represents the entire image. This increase in representational power provided by multiple feature vectors makes this type of learning more challenging than single-instance learning, because some instances will be more relevant to the classification of a bag (image) than others.

In this paper we do not propose a new technique for multi-instance learning. Rather, we investigate the benefit of using multiple multi-instance representations of the same image, where each representation is based on a different set of features. These distinct sets of features are commonly called "views" in the machine learning literature, and multi-view learning aims to exploit different views on the same entity to induce more accurate classification models. The classic example of this is a web page, where one set of features describes the textual content of the page and a second set describes the links in this page.

Multi-view learning is a very natural way to exploit different multi-instance representations of the same image. Given single-instance data, the multi-view approach is commonly applied to implement semi-supervised learning, where most of the training data is not labeled.[1] However, whether the problem is semi-supervised or not, in the single-instance case there is always the option of simply concatenating the different views into a single feature vector. In multi-instance learning, this option does not exist, as it is unclear how to effectively and efficiently join two bags of feature vectors to form a single bag—the corresponding operation to concatenating feature vectors in the single-instance case.

This makes a strong case for applying multi-view learning to multi-instance problems, even in a strictly supervised setting, provided improvements in classification accuracy can be obtained. In this paper, we investigate whether this is the case for image classification tasks, a prominent application area for multi-instance learning. We perform an extensive empirical evaluation on image classification datasets and show that MVMI can indeed, in some cases, provide significant improvements in classification accuracy over single-view learning.

TABLE I
Summary of the datasets used in the evaluation of MVMI.

Dataset	#Images	#Classes
Bikes	200	2
Cars	200	2
People	200	2
Gender	418	2
Galaxy	1733	3
Scenes	3899	13
Caltech	1703	15
Moths	880	35

II. Datasets

In this section, we discuss the image datasets used in our study, which vary considerably in the degree of difficulty and style. Table I gives summary statistics of each of the datasets.

A. GRAZ02 Bikes, Cars and People

The GRAZ02 Dataset [19] is a popular natural scene database. Although it contains only three classes, namely People, Bikes and Cars, (with an additional "background" class), it is well known to be a difficult and challenging dataset for three primary reasons: (i) significant occlusions

[1]Note that there is also some existing work on adopting this approach for semi-supervised multi-instance learning [10]

Fig. 1. Examples from the Bike, Car and Person classes in the GRAZ02 Dataset

Fig. 2. Examples of the Male and Female classes in the Gender Dataset

and background clutter, (ii) intra-class variability, and (iii) the fact that the objects of interest are often not dominant in the foreground of the images. Figure 1 gives some examples of the images from GRAZ02.

The total size of the GRAZ02 Dataset is 1280 images. We randomly selected 100 images from each of the Bike, Car and People classes. These images formed the positive classes for three binary learning problems. To obtain the negative classes for each of the three positive classes, we simply selected 100 images randomly from the remaining classes. Thus, we formed three binary image datasets, each evenly balanced and comprising exactly 200 images.

B. Gender

The Face Gender Recognition Dataset, derived from the Feret face recognition dataset [20], was first proposed in [14] for evaluating the effectiveness of complex face alignment algorithms on gender recognition. Subsequent work on this dataset is described in [17]. Each face image is neatly cropped and centered. , and Figure 2 gives some examples of images from this dataset. The dataset consists of 212 male face images and 107 female face images.

C. Galaxy Zoo

The Galaxy Zoo Dataset is a set of astronomical images obtained from the Galaxy Zoo project [11] and labeled with ground truth data in [23]. There are three classes of galaxy image in this dataset: Elliptical (comprising 215 images), Spiral (247 images), and Edge-On (107 images). Examples of these images are given in Figure 3. An interesting and A challenging aspect of these images is that they tend to be low-resolution (120×120 pixels) and very noisy.

D. Scenes

The Scenes Dataset was first proposed in [9] and consists of thirteen classes. Each class represents a natural scene such as

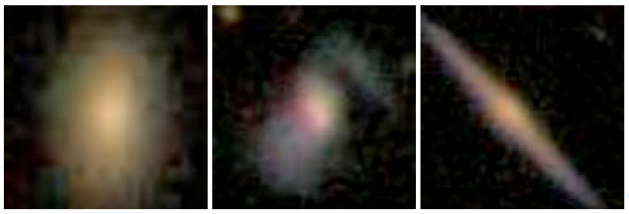

Fig. 3. Examples of the Elliptical, Spiral, and Edge-On classes in the Galaxy Dataset.

Fig. 4. Examples of the Highway, Kitchen and Forest classes in the Scenes Dataset.

Fig. 5. Examples of the Anchor, Beaver and Camera classes from the Caltech 101 Dataset.

Bedroom, Inside City or Office. This is distinct from the other datasets where a class usually denotes the presence or absence of an object of interest. The dataset as originally used contains in the order of hundreds of images per class; to reduce runtime, we decreased this by random sampling without replacement, to 50 images per class in our experiments. Figure 4 gives examples of some of the images.

E. Caltech 101

The Caltech 101 Dataset [3] is a large and well known object recognition dataset. It consists of 101 object classes and a background class. This dataset tends to be somewhat "easy" in the sense that each image contains the relevant object explicitly in the foreground, and in a dominant position. There is usually no interfering background. For the purposes of evaluating MVMI, we did not use all 101 classes, but instead selected only the first 15 classes in alphabetical order, excluding the background class. These classes are: Accordion, Airplanes, Anchor, Ant, Barrel, Bass, Beaver, Binocular, Bonsai, Brain, Brontosaurus, Buddha, Butterfly, Camera, and Cannon. Figure 5 gives examples from three of those classes. There are 1,703 images in this subset.

F. Species Recognition

The final dataset that we used, and the one with the highest number of classes (35) is a moth species recognition dataset first described in [24] and subsequently further analyzed

Fig. 6. Examples of the classes from the Species Recognition Dataset.

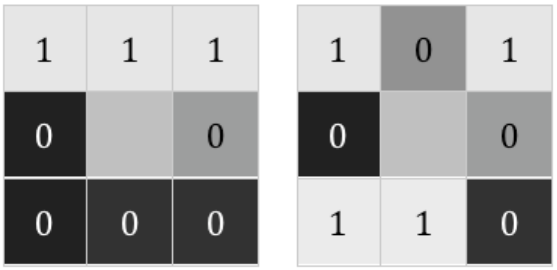

Fig. 7. Two 3×3 pixel neighborhoods used to calculate the LBP of the central pixel.

in [16]. This dataset is unique in that the moths were imaged while alive. Consequently, there is often variation in the size, position and pose of the moths in the images. Examples of three different species of moth are depicted in Figure 6. The total size of the dataset is 880 images.

III. Views

Given the image datasets described in the previous section, we now describe how the raw image data was converted into different views for the MVMI experiments. As is standard, each view corresponds to a fixed-length vector of features. In the application considered here, all feature values are numeric.

A. View 1: Spatial Pyramid of Local Binary Patterns

View 1 represents the most traditional type of view used in image recognition experiments. In this approach, each image is mapped to a single feature vector of fixed length. Hence, this first type of view actually yields a single-instance dataset– or alternatively, a "multi-instance" dataset with one instance per bag of instances. (The other views we consider are genuine multi-instance views.)

We used histograms of Local Binary Patterns (LBPs) [13] as the base image features in this view, because LBPs have been successfully applied previously in diverse applications ranging from texture classification (e.g. [13]) to face recognition (e.g. [1]). Furthermore, the computation of LBPs can be effectively approximated so that only integer calculations are performed: this significantly speeds up feature extraction while minimally affecting classification performance.

Briefly, a LBP is a property of a pixel in an intensity image, calculated by comparing the pixel to its neighboring pixels in a 3×3 block. Each neighbor is labeled with either 1 or 0, depending on whether it is lighter or darker, as shown in Figure 7. An 8-bit string is then calculated for each pixel by starting at the top left corner and following the neighbors clockwise. For example, for the neighborhood on the left in Figure 7, the string is 11100000 and for the neighborhood on the right, it is 10100110.

Fig. 8. The subdivision of an image into 21 regions using a spatial pyramid. (Image courtesy of [17].)

We ignore LBPs that are not uniform, where uniformity means that there are no more than two 0 to 1 or 1 to 0 transitions as one traverses the bit string circularly. Thus, the LBP for the pixel on the left in Figure 7 is uniform because it comprises exactly two transitions in its bitstring, whereas the LBP for the pixel on the right is not uniform, as it comprises six transitions.

The type of each uniform LBP is then computed by converting the bitstring into an integer. For example, the bitstring 11100000 is converted into 224, the bitstring 00011000 becomes 24, and 00000111 is 7. There are a total of 58 different uniform LBPs possible for any 3×3 neighborhood, and the effectively capture a variety of different classes of low-level image features, from bright points (defined by a LBP such as 00000000) to straight edges (such as the left hand example in Figure 7) to corners (for example, 111100011).

Once the LBP for each pixel has been computed, a frequency histogram of LBPs across an entire image region is computed. This histogram characterizes the image region. Following previous works, we divide the image up into 21 overlapping regions of different sizes, according to the spatial pyramid formulation [8]. A spatial pyramid involves first extracting a global feature histogram, and then dividing the image into 2×2 subregions. Four more histograms are then extracted from each subregion. Next, the image is divided again, this time into 4×4 subregions. Figure 8 illustrates the division of an image into regions, using a sample image from the Gender Dataset.

All 21 frequency histograms are normalized and then concatenated into a single feature vector describing the image overall, consisting of 1,218 numeric features in total. An advantage of using a spatial pyramid is that the feature vector describes the image at both the coarse, global scale, as well as the fine, detailed scale.

B. View 2: SIFT Keypoints

The second view we use in our experiments represents a classic method of matching images using SIFT Keypoints [12], which are essentially "interest points" detectable at multiple scales. Unlike View 1, this view is a true multi-instance view because each image can have any number of keypoints derived from it, and there is no natural way to order keypoints

consistently across images (which would enable construction of a single feature vector by concatenation). The group of keypoints from a single image, where each keypoint is described by a fixed-length numeric feature vector, forms a bag for multi-instance learning.

In our implementation of the SIFT keypoints, each keypoint consists of 256 numeric features, and on average there are 20-100 keypoints per bag.

C. View 3: Uniform Patches

A "visual dictionary" is an interesting and increasingly-used method to describe image sets (e.g. [25],[9]). Creating a visual dictionary normally involves dividing the images into patches or regions, and then clustering the patches to discover the cluster centers. These then become the "words" in a dictionary, and an image is considered a set of these words.

For View 3, we adopted this approach and divided the image up into 5×5 uniformly distributed image patches. Thus, each image was converted into a bag comprising 25 feature vectors. We constructed a LBP histogram for each feature vector, yielding 58 numeric features per vector. Rather than clustering the patches explicitly, we provide the raw patches to the MI learning algorithm instead, assuming that the MI algorithm implicitly builds a visual dictionary via the learning process.

D. View 4: Random Patches

View 4 is similar to View 3, except that, instead of extracting the patches uniformly, we extracted them from random positions and at random sizes, with a minimum patch size of 5×5 pixels and no maximum patch size other than the size of the image. A total of 25 patches were extracted per image, making the bags the same size as those from View 3.

IV. Learning Algorithms

We used two popular multi-instance learning algorithms, implemented as MISMO and MIBOOST respectively, in our evaluation. Both are implemented in the WEKA machine learning workbench [6], and both yield models that output confidence scores represented as class probability estimates. To apply them to multi-view data (yielding MVMI classifiers) we used the following straightforward strategy: Firstly, the given algorithm (MISMO or MIBOOST) was trained on each of the views individually, producing a classifier for each view. Then, at prediction time, each individual classifier's confidence scores—one for each class from each view—were averaged across views to compute the overall MVMI classifier's prediction for each class. Other more complex combining approaches such as stacking [26] are possible but initial experiments did not yield improved results and the training times were an order of magnitude greater.

A. MISMO

MISMO constructs a support vector machine classifier for multi-instance data. Support vector machines are linear models that minimize a specific penalized loss function on the training data—the so-called "hinge" loss. A quadratic penalty term is normally included to control overfitting, and a kernel function can be used to construct a non-linear classifier in the original feature space by learning a hyperplane in the higher-dimensional kernel-induced space.

As feature vectors only enter the support vector algorithm through the kernel function, which can be viewed as a similarity function with certain mathematical properties, all that is needed to apply this kind of algorithm to multi-instance data is an appropriate similarity measure for bags of instances.

MISMO is an implementation of the standard sequential minimization algorithm for support vector learning [22], applied in conjunction with a multi-instance kernel as described in [5]. We use the set kernel from that paper. Given an underlying single-instance kernel function that can be applied to pairs of individual feature vectors, the set kernel simply takes the sum of all possible pairwise kernel applications for all pairs of feature vectors from the two bags being compared. This yields a similarity score for pairs of bags. In our experiments, we use a quadratic polynomial kernel as the underlying single-instance kernel. The complexity parameter C was left at 1, which is WEKA's default. To obtain multi-class probability estimates, we configured MISMO with pairwise coupling [7] and calibration using logistic regression models [21].

B. MIBOOST

MIBOOST implements a boosting algorithm for multi-instance data [27]. Like other boosting schemes, this algorithm greedily fits an additive model to the training data. In each iteration of the sequential boosting process, an underlying "weak" learner is applied to generate one component of this additive model. The algorithm is a variant of the well-known AdaBoost.M1 algorithm [4], adapted to minimize the exponential loss function for bags of instances.

In our experiments, we used unpruned depth-limited decision trees as the "weak" classifiers (i.e. components of the additive models), generated using WEKA's fast REPTree algorithm. REPTree pre-sorts numeric attributes and chooses splits by maximizing the information gain. The depth-limit was set to three, meaning that each tree could model interactions between up to three features. To tackle multi-class datasets, we used the well-known one-vs-rest method, where each class is discriminated against all other classes, and the normalized scores are output at prediction time. We used 100 boosting iterations.

V. Evaluation of MVMI

In this section, we describe the evaluation of MVMI that we carried out and detail the results obtained.

A. Experimental Setup

Both multi-instance learning algorithms, MISMO and MIBOOST, were applied to all eight learning problems, yielding a total of 16 algorithm/dataset combinations, and we also have four different views of the datasets: View 1, a single instance representation, and Views 2-4, both of which are

multi-instance representations. This resulted in a total of 16×4 or 64 non-MVMI algorithm/dataset/view combinations. Each combination represents a single experiment on a single view that we carried out, and these experiments can be thought of as the set of "control" experiments.

We then considered two different MVMI classifiers obtained by combining views. The first of these, MV_a comprises only Views 1 and 2, because they are the two most diverse views. The second MVMI classifier, MV_b, comprises all four views. Thus, considering the eight datasets and two underlying MI algorithms, there was a total of 16×2=32 MVMI combinations, which formed the set of experimental conditions to be compared to the controls.

Overall, therefore, a total of 64+32=96 experiments were carried out. Each experiment consisted of 10 stratified 10-fold cross validation experiments to obtain estimates of classification accuracy, with the results averaged over the runs.

B. Experimental Results

Tables II and III present the average accuracy of each classifier by dataset and view (or view combination). In both tables, View 1 is the base view with which each of the other views are compared for statistical significance testing, based on a corrected resampled *t*-test [18]. We choose View 1 for this role because it outperforms all the other individual multi-instance views. We indicate in bold the best result for each dataset in each of the tables.

Examining the tables, it can be observed that when MISMO is used in conjunction with multiple views (columns MV_a and MV_b in Table II), there is an improvement in mean accuracy for five of the eight datasets compared to the first view, which itself performs very strongly against the other individual views. On two datasets, significant improvements can be obtained, and there is no significant degradation. When MIBOOST is the classifier (see Table III), it is possible to achieve a gain in accuracy for all but one dataset using multiple views instance of View 1. Significant improvements can again be obtained on two datasets, but there is also a significant degradation on one dataset.

Closer examination reveals that MIBOOST plus MVMI produces the overall best results for five datasets: Bike, Cars, Person, Galaxy, and Scenes. Interestingly, although MIBOOST+MVMI also produces a gain for the Moths dataset when compared to the corresponding single-view classifiers, it is MISMO+MVMI that produces the best result overall. Most of the greatest improvements are achieved by using all four views (namely MV_b) although occasionally the largest gains are achieved by combining only the first two views (MV_a).

On the negative side, MVMI and in fact all the multi-instance views (Views 2-4) fail to produce an improvement for the Caltech data. The best result in this case is achieved consistently by MISMO in conjunction with the single instance view, View 1. Also, with respect to the individual multi-instance views, when compared to the corresponding classifier trained on single instance data only (i.e. comparing V_{2-4} vs. V_1 in the tables), the former tend to produce less accurate classifiers in general. Sometimes the difference is very large (e.g. considering the results for the Moths dataset in Table II). However, there are exceptions to this rule, as the results for the Scenes and Moths data in Table III show (V_3 and V_4), and these are perhaps the most challenging datasets.

Nevertheless, combining the classifiers trained from the single-instance and multi-instance views does generally produce the highest accuracy overall. What is most notable is the apparent orthogonality of the classifiers that different views lead to. For example, often the multi-instance views (especially View 2, the keypoints-based view) perform worse than the single-instance views, but when views are combined, the overall MVMI classifier generally performs better than the corresponding single-instance classifier. A good example of this is the Moths Dataset in Table II: an MISMO classifier built from View 1 achieves 60.68% accuracy whereas the same classifier built from View 2 achieves only 32.03%. Their combination, however, produces a startling improvement to 65.91%.

VI. Conclusion

In this paper, we have investigated the use of multi-view learning in a strictly supervised context, namely for image classification using multi-instance learning. Due to the nature of multi-instance data, multi-view learning is particularly appropriate in this scenario. Our results, based on comprehensive experiments using six image datasets and two underlying multi-instance learning algorithms, show that the multi-view multi-instance approach can indeed, in some cases, deliver significantly improved classification accuracy compared to the standard method of using single-view (multi-instance) learning.

References

[1] Ahonen, T., Hadid, A., Pietikäinen, M.: Face description with local binary patterns: Application to face recognition. IEEE Trans. on Pattern Recognition and Machine Intelligence 28(12), 2037–2041 (2006)

[2] Dietterich, T.G., Lathrop, R.H., Lozano-Perez, T.: Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(1-2), 31–71 (1997)

[3] Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: IEEE. CVPR 2004, Workshop on Generative-Model Based Vision (2004)

[4] Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proc. of the 13th International Conference on Machine Learning (ICML. pp. 148–156. Morgan Kaufmann (1996)

[5] Gärtner, T., Flach, P.A., Kowalczyk, A., Smola, A.: Multi-instance kernels. In: Proceedings of the 19th International Conference on Machine Learning (ICML). pp. 179–186. Morgan Kaufmann, San Francisco, CA (2002)

[6] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: An update. SIGKDD Explorations 11(1) (2009)

[7] Hastie, T., Tibshirani, R.: Classification by pairwise coupling. The Annals of Statistics 26(2), 451–471 (1998)

[8] Lazebnik, S., C., C.S., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) - Volume 2. pp. 2169–2178 (2006)

[9] Li, F., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 524–531 (2005)

TABLE II

AVERAGED RESULTS OF STRATIFIED 10×10 CROSS VALIDATION EXPERIMENTS USING MISMO. RESULTS ARE ORGANIZED BY DATASET (THE ROWS) AND VIEW (THE COLUMNS).

Dataset	V_1	V_2	V_3	V_4	MV_a	MV_b
Bike	80.50±8.51	65.60±10.31 ●	74.50±10.19	71.65±10.87 ●	79.50±8.83	**80.60**±8.77
Car	**77.80**±8.17	57.85±10.28 ●	66.40± 9.69 ●	61.90±11.34 ●	77.00±9.18	74.05±9.12
People	74.05±9.12	66.10±10.14	72.65± 9.06	66.30±10.09	73.95±9.14	**77.10**±8.68
Gender	**93.75**±3.35	70.53± 7.37 ●	89.93± 4.57 ●	78.90± 6.45 ●	93.36±3.50	93.74±3.29
Galaxy	86.26±3.95	85.82± 4.24	79.07± 5.08 ●	73.81± 5.05 ●	**90.11**±3.97 ○	89.31±4.21 ○
Scenes	74.14±4.59	27.05± 4.42 ●	71.20± 5.50	71.20± 5.50	74.91±4.66	**75.82**±5.58
Caltech	**81.13**±2.24	58.46± 2.12 ●	67.31± 2.19 ●	72.38± 2.76 ●	80.17±2.37	79.88±2.39
Moths	60.68±4.52	32.03± 4.53 ●	40.66± 4.42 ●	28.80± 4.71 ●	**65.91**±4.31 ○	63.01±4.66

○, ● statistically significant improvement or degradation

TABLE III

AVERAGED RESULTS OF STRATIFIED 10×10 CROSS VALIDATION EXPERIMENTS USING MIBOOST. RESULTS ARE ORGANIZED BY DATASET (THE ROWS) AND VIEW (THE COLUMNS).

Dataset	V_1	V_2	V_3	V_4	MV_a	MV_b
Bike	80.35±8.68	71.95±8.70	75.40±8.49	76.25±10.01	81.05±8.88	**82.50**±8.21
Cars	81.50±8.02	69.60±8.69 ●	74.90±9.56	72.70± 9.78 ●	**81.80**±8.06	80.00±8.65
People	80.75±7.50	66.25±9.19 ●	76.30±8.89	76.35± 8.43	80.75±7.80	**80.80**±7.87
Gender	91.02±4.17	73.19±5.97 ●	87.44±5.65	83.74± 5.83 ●	91.09±3.97	**91.38**±4.14
Galaxy	88.44±4.24	88.00±4.04	72.69±5.08 ●	76.64± 5.77 ●	**90.14**±3.65	89.51±4.08
Scenes	69.54±5.02	36.34±6.06 ●	75.69±5.17 ○	75.69± 5.17 ○	59.60±5.62 ●	**76.72**±5.01 ○
Caltech	**78.23**±2.56	63.27±2.65 ●	70.38±2.59 ●	69.16± 2.36 ●	73.72±2.18 ●	76.43±2.36 ●
Moths	32.76±6.04	39.21±5.55 ○	41.41±5.20 ○	31.08± 4.94	49.13±6.52 ○	**54.00**±5.41 ○

○, ● statistically significant improvement or degradation

[10] Li, W.J., Yeung, D.Y.: Mild: Multiple-instance learning via disambiguation. IEEE Trans. on Knowledge and Data Engineering 22, 76–89 (2009)

[11] Lintott, C.J., Schawinski, K., Slosar, A., Land, K., Bamford, S., Thomas, D., Raddick, M.J., Nichol, R.C., Szalay, A., Andreescu, D., Murray, P., van den Berg, J.: Galaxy zoo: Morphologies derived from visual inspection of galaxies from the sloan digital sky survey. Monthly Notices of the Royal Astronomical Society 389(3), 1179–1189 (2008)

[12] Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)

[13] Mäenpää, T., Pietikäinen, M.: Texture analysis with local binary patterns. In: Chen, C., Wang, P. (eds.) Handbook of Pattern Recognition and Computer Vision, 3rd ed, pp. 197–216. World Scientific (2005)

[14] Makinen, E., Raisamo, R.: Evaluation of gender classification methods with automatically detected and aligned faces. IEEE Trans. on Pattern Analysis and Machine Intelligence 30(3), 541–547 (2008)

[15] Maron, O., Ratan, A.: Multiple-instance learning for natural scene classification. In: Proc. of the 15th International Conference on Machine Learning (ICML). pp. 341–349. Morgan Kaufmann (1998)

[16] Mayo, M., Watson, A.: Automatic species identification of live moths. Knowledge-Based Systems 20(2), 195–202 (2007)

[17] Mayo, M., Zhang, E.: Improving face gender classification by adding deliberately misaligned faces to the training data. In: Proceeding of 23rd International Conference Image and Vision Computing New Zealand 2008 (IVCNZ 2008) (2008)

[18] Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning 52(3), 239–281 (2003)

[19] Opelt, A., Pinz, A.: Object localization with boosting and weak supervision for generic object recognition. In: Proc. of the 14th Scandinavian Conference on Image Analysis (SCIA) (2005)

[20] Phillips, P., Moon, H., Rizvi, S., Rauss, P.: The feret evaluation methodology for face recognition algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence 22, 1090–1104 (2000)

[21] Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A., Bartlett, P., Schölkopf, B., Schuurmans, D. (eds.) Advances in Large Margin Classifiers. MIT Press (1999)

[22] Platt, J.C.: Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B., Burges, C., Smola, A. (eds.) Advances in Kernel Methods – Support Vector Learning. MIT Press (1998)

[23] Shamir, L.: Automatic morphological classification of galaxy images. Monthly Notices of the Royal Astronomical Society 399(3), 1367–1372 (2009)

[24] Watson, A., O'Neill, M., Kitching, I.: Automated identification of live moths (macrolepidoptera) using digital automated identification system (DAISY). Systematics and Biodiversity 1(3), 287–300 (2004)

[25] Wen, C., Guyerb, D., Li, W.: Local feature-based identification and classification of orchard insects. Biosystems Engineering 104(3), 299–307 (2009)

[26] Wolpert, D.H.: Stacked generalization. Neural Networks 5(2), 241–259 (1992)

[27] Xu, X., Frank, E.: Logistic regression and boosting for labeled bags of instances. In: Proceedings of the 8th Pacific-Asia Advances in Knowledge Discovery and Data Mining Conference (PAKDD). pp. 272–281. Springer-Verlag, Berlin (2004)

[28] Zhou, Z., Zhang, M.: Multi-instance multi-label learning with application to scene classification. Advances in Neural Information Processing Systems 19, 1609–1616 (2007)