Individual feature selection in each One-versus-One classifier improves multi-class SVM performance

Phoenix X. Huang
School of Informatics
University of Edinburgh
10 Crichton street, Edinburgh
Xuan.Huang@ed.ac.uk

Robert B. Fisher
School of Informatics
University of Edinburgh
10 Crichton street, Edinburgh
rbf@inf.ed.ac.uk

***Abstract*—Multiclass One-versus-One (OvO) SVM, which is constructed by assembling a group of binary classifiers, is usually treated as a black-box. The usual Multiclass Feature Selection (MFS) algorithm chooses an identical subset of features for every OvO SVM. We question whether the standard process of applying feature selection and then constructing the multiclass classifier is best. We propose that Individual Feature Selection (IFS) can be directly applied to each binary OvO SVM. More specifically, the proposed method selects different subsets of features for each OvO SVM inside the multiclass classifier so that each vote is optimised to discriminate between the two specific classes. This paper shows that this small change to the normal multiclass SVM improves performance and can also reduce the computing time of feature selection. The proposed IFS method is tested on four different datasets for comparing the performance and time cost. Experimental results demonstrate significant improvements compared to the normal MFS method on all four datasets.**

I. Introduction

Multiclass classifiers (that categorize objects into specific classes) are important tools since they are widely applied to machine vision and pattern recognition applications. Over the last decade, SVM has shown impressive accuracy in resolving both linear and nonlinear problems by maximizing the margin between classes [1]. Although SVM was originally designed for a binary task, additional mechanisms can create a multi-class SVM by decomposing it into several binary problems such as One-vs-Rest (OvR) and One-vs-One (OvO) [2].

Multiclass SVM is often treated as a black-box within more complicated applications, such as object recognition ([3], [4]) and bio-informatics ([5], [6]) and text classification ([7], [8]), which hides the process that the multiclass SVM generates results by using a group of assembled binary classifiers. In practice, feature selection is necessary for applications that have an abundant number of features. It not only eliminates redundant features to reduce computation and storage requirements, but also chooses appropriate feature subsets that improve the prediction accuracy. [9] categorizes the feature selection methods into three types: filter, wrapper and embedded. The filtering method evaluates the correlation of every feature and ranks them by their coefficients, so the selection algorithm chooses new features that have lower correlations to the existing features. The wrapper method, which tests the prediction power of single feature, investigates the independent usefulness of features and the selecting strategy is according to the order of power. The embedded method integrates both feature selection and training. It selects features while building the model. Figure 1 illustrates a typical example of the feature selection result of a multiclass application. Firstly, the classification performance increases as more features are selected, because more features provide more discriminative ability in the feature space. After the number of selected feature reaches 10, the accuracy score fluctuates near a specific level. Then the score starts to drop due to redundancy and over-fitting when more than 30 features are selected.

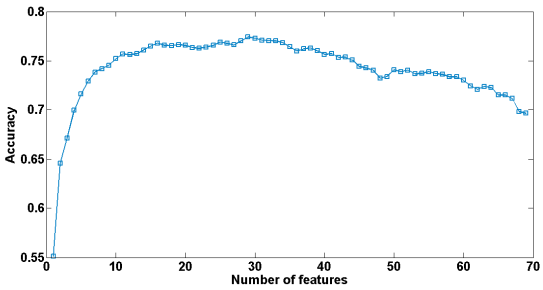

Fig. 1: An example of the feature selection result in a multiclass application. The accuracy score increases in the beginning but it drops after 30 feature are selected. This example indicates that feature selection reduces the size of the feature space and also improves the accuracy by choosing an appropriate feature subset, instead of using all features.

Normally, the Multiclass Feature Selection (MFS) procedure is applied to the black box of multiclass SVM, and it selects the same feature subset for every binary classifier to maximize the average accuracy over all classes [10], [11], [12]. Here we investigate the sequence of feature selection and constructing a multi-class SVM. We propose that an Individual Feature Selection (IFS) procedure can be directly exploited to the binary OvO SVMs before assembling the full multiclass SVM. Given samples from every pair of classes, the selected subset of features maximizes the accuracy of classifying these classes. After then, we use these optimised OvO SVMs to construct a multi-class classifier. One can hypothesise that the classification performance would be better under the second scheme because each vote is now optimised to discriminate between two specific classes. The experimental result shows that this small change to the normal multiclass SVM significantly improves performance with a decreased

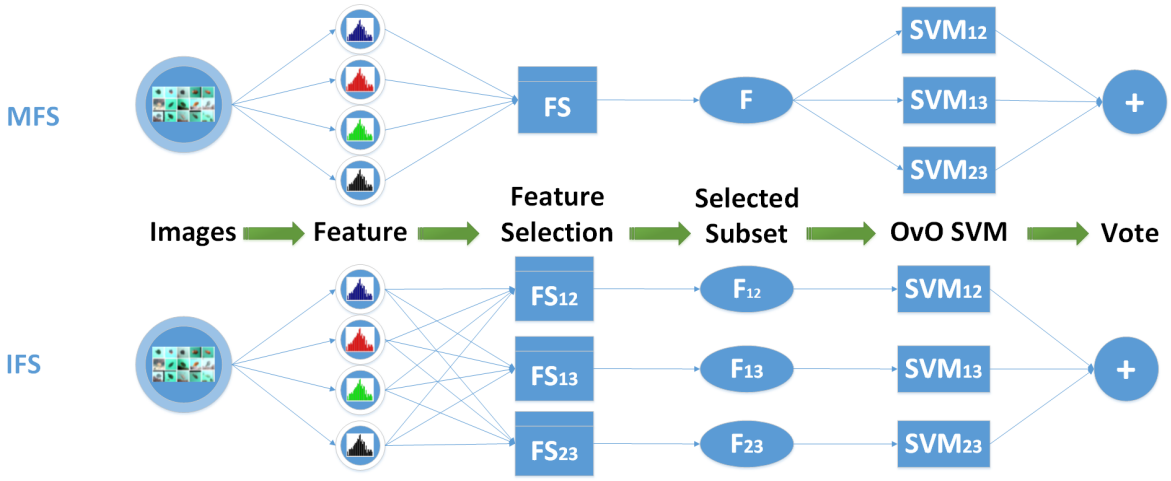

Fig. 2: Comparing the workflows of MFS and IFS. We choose an example that classifies three classes so the final prediction is calculated by voting from three OvO SVMs. In the first row, the MFS method selects the same subset of features for all binary OvO SVMs while the IFS method chooses an individual feature subset for each OvO classifier.

computing cost.

The main contribution of this paper is a novel practical mechanism that applies individual feature selection to the binary OvO SVM, called IFS-SVM. After forward sequential feature selection and learning each SVM model, IFS-SVM classifies each test sample by counting votes that are optimised for every two specific classes. The proposed method is evaluated on four different datasets for comparing the performance and computing time. We note that other feature selection and vote combination methods could be used. This paper only addresses the issue of when to do the feature selection. The rest of the paper is organized as follows: Section 2 introduces the multiclass SVM with OvO strategy. Section 3 describes individual feature selection for multiclass SVM. Section 4 shows experimental results of four datasets: two underwater fish image datasets, the Oxford flower dataset and a skin lesion image dataset.

II. Multiclass SVM with OvO strategy

Given a training set $\mathcal{D}$ from p classes, which is a set of n sample points of the form:

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^m,\ y_i \in \{1, ..., p\}\}_{i=1}^{n} \quad (1)$$

y_i indicates the class label of m-dimensional feature vector $\mathbf{x}_i$. Considering the two-class task (p = 2), the maximum margin classifier, a Support Vector Machine (SVM) [13], is optimized to find a hyperplane, called maximum-margin hyperplane, which maximizes the margin between the two classes.

A multiclass task can be decomposed into a set of two-class problems where the binary SVMs are applicable. One strategy is to train p One-versus-Rest (OvR) classifiers and they are used to classify one class from all the other classes. The final classification is determined by the highest score (winner-takes-all). The second strategy pairs each two of the classes and trains an SVM classifier for each pair, named as One-versus-One (OvO) strategy. Each binary classifier is trained on only two classes, thus the method constructs $p * (p-1)/2$ binary OvO SVMs. These binary classifiers process the test sample, and the winning class is added a vote. The class with the most votes determines the final prediction. Both strategies are widely used and have their own pros and cons. OvR uses fewer binary classifiers and the training cost is linear with p but it is criticized for no bound on the generalization error [2] and resolving potentially asymmetric problems using a symmetric approach [14]. OvO is easy to train because each classifier only resolves a binary classification problem with two classes, but the computation cost is bigger since the number of binary classifiers grows as $p * (p-1)/2$.

III. Individual feature selection for binary OvO-SVMs

After constructing the multiclass SVM using the OvO strategy, the Multiclass Feature Selection (MFS) method chooses a subset of features by either filtering features according to their correlation coefficients or wrapping them in proportion to their usefulness to a given SVM predictor [9]. In contrast to the MFS criteria that treats the multi-class SVM as a black-box and selects features such that all binary classifiers use the same subset of features, our proposed work investigates applying feature selection to each binary classifier individually so that each OvO vote is optimized. An example of comparing the different workflows of MFS and IFS is shown in Figure 2. Both methods use the same forward sequential feature selection algorithm. The complete proposed training procedure is described as follows:

(1) For every two classes i, j ($i, j \in \{1, ..., p\}$ and $i \neq j$), start with an empty feature set $F_{ij} = []$ and m features $\{f_t\} = F$. The evaluation function is named as E.
(2) Repeat until all features are evaluated, step $s \in \{1, ..., m\}$:

- select every $\{f_t\} \in F$ and evaluate $e_{s,t} = E([\tilde{F}_{ij}, f_t])$
- choose the maximum of all evaluations $\tilde{e}_s = \arg\max_t e_{s,t}$, record $\tilde{e}_s$.

- add the corresponding feature $\tilde{f}_s$ to the feature set $\tilde{F}_{ij}$ as the selected feature of step s: $\tilde{F}_{ij}$ = $E([\tilde{F}_{ij}, \tilde{f}_s])$.
- remove the feature $\tilde{f}_s$ from the feature pool F: F = F - $[\tilde{f}_s]$.

(3) Choose the feature subset $F_{ij} = [\tilde{f}_1, ..., \tilde{f}_{\tilde{s}}]$ that produce the highest evaluation score for each i, j, where $\tilde{s} = \arg\max_s \tilde{e}_s$. Note: other stopping criteria could be used.

After feature selection, these binary SVMs are trained using their corresponding feature subsets $\tilde{F}_{ij}$ of the training samples. In the evaluation step, binary SVMs also extract the $\tilde{F}_{ij}$ features of the test samples, and they vote for the final prediction. It is reasonable to assume that each vote is optimized so the prediction is more accurate.

One concern is the computational complexity. But given the assumption that the computing time of classification only depends on the number of features, we can show that our proposed method (IFS-SVM) has no more computing time in feature selection than the common MFS method (both discussed by the forward sequential feature selection algorithm):

Assumption 1: The computation time of a binary classifier only depends on the number of input features, *i.e.* $f(D_{m*n}) = f(m, n)$ where function f is the computation time, D_{m*n} is the input features, m is the number of samples, n is the number of features.

This assumption eliminates nonessential details so we can focus on comparing the time cost itself. The computation time of feature selection using MFS is:*explain why (3) & (4) are true*

$$T_{MFS} = \sum_{n=1}^{\tilde{N}}[(N-n+1)*(T_v(c)+\sum_{i\neq j\&i,j\leq c} f(M_i+M_j, n))] \tag{2}$$

where M_i is the number of samples from class i, $i \in \{1, 2, ..., c\}$, c is the number of classes, N is the number of input feature F and $\tilde{N}$ is the number of features to select, T_v is the computing time of voting. The computation time of feature selection of our proposed IFS method is:

$$\begin{aligned} T_{IFS} &= \sum_{i\neq j\&i,j\leq c} \sum_{n=1}^{\tilde{N}}[(N-n+1)*f(M_i+M_j, n)] \\ &= \sum_{n=1}^{\tilde{N}}[(N-n+1)*\sum_{i\neq j\&i,j\leq c} f(M_i+M_j, n)] \leq T_{MFS} \end{aligned} \tag{3}$$

Although the IFS-SVM method conducts p^2 times individual feature selections, the size of samples in each individual one is decreased to $2/p$ (two out of p classes). Thus the computing complexity is still $O(p^2)$. On the other hand, equations 2 and 3 describe that the IFS-SVM method saves the voting procedure when selecting features. We have conducted experiments on four datasets to compare the consumed time of both methods, as shown in the last column of Figure 5. This experiment changes the number of classes p and records the computing time of feature selection as describe in Section II. Both curves fluctuate since the number of selected features may vary in different number of classes. The general trend indicates that the proposed method (IFS-SVM) spends less time in training than the MFS method. See experimental section for more details.

IV. EXPERIMENTS

We test both feature selection mechanisms on four datasets using cross validation. The binary OvO SVM classifier is implemented by LIBSVM [15]. We use the same forward sequential feature selection for all tests so the results are comparable. All experiments are programming in Matlab. The code is compiled and deployed on a cluster of machines. The performance is evaluated by Average Recall (AR), Average Precision (AP) and Accuracy over Count (AC). AR and AP describe the recall/precision that are averaged over all classes so the minority classes have equal importance to the major ones. AC is the accuracy over all samples, and it is defined as the proportion of correct classified samples among all samples. These scores illustrate a comprehensive analysis of the experimental results regardless of whether the dataset is balanced or not. In each experiment, we compare AR/AP/AC scores of three methods: multiclass SVM without feature selection (M-SVM), multiclass feature selection for SVM (MFS-SVM), individual feature selection for multiclass SVM (IFS-SVM).

A. Underwater fish image dataset

The fish images are acquired from underwater cameras placed in the Taiwan sea with 24150 fish images (Fish24K dataset) of the top 15 most common species [16]. We use the same method of feature extraction as in [17]. These features are combinations of 69 types (2626 dimensions) including color, shape and texture properties in different parts of the fish such as tail/head/top/bottom, as well as the whole fish. All features are normalized by subtracting the mean and dividing by the standard deviation (z-score normalized after 5% outlier removal). Figure 3 shows this fish dataset, with the number of images in each species.

The classification results after feature selection with 5 fold cross-validation are shown in Table I. This dataset is very imbalanced, thus the averaged recall and precision are lower than the accuracy over all samples. The first row shows the result of multiclass SVM using all features, where the averaged recall (AR) is increased after the feature selection with the cost of reduced AP and AC (the second row). In the third row, individual feature selection (IFS-SVM) improves the classification performance in all three measures.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	76.9 ± 4.0	88.5 ± 3.6	95.7 ± 0.5
MFS-SVM	79.0 ± 3.6	86.4 ± 5.3	95.3 ± 0.3
IFS-SVM	**81.6 ± 4.7**	**90.9 ± 5.0**	**96.4 ± 0.5***

TABLE I: Experiment results on the whole fish image dataset, all results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

The Fish24K dataset is so imbalanced that the image number of the most major species is 500 times larger than the

Fig. 3: Fish data: 15 species, 24150 fish detections. The images shown here are ideal images as many of the others in the database are a bit blurry, and have fish at different distances, and orientations or are against coral or ocean floor backgrounds.

number of the least species. We conduct another experiment on a similar dataset of 6874 fish images (Fish7K dataset) to evaluate the performance when dataset is less imbalanced. The result is shown in Table II. The MFS method reduces the feature dimensions with the cost of slightly decreasing the performance, while the proposed IFS method significantly improves the performance.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	72.6 ± 6.1	77.7 ± 3.3	93.2 ± 0.9
MFS-SVM	72.3 ± 8.8	77.5 ± 7.4	92.9 ± 1.1
IFS-SVM	**80.2 ± 3.0**	**89.8 ± 5.4***	**94.9 ± 1.3***

TABLE II: Experiment results on more balanced fish dataset of 6874 images, all results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

B. Oxford flower dataset

The Oxford flower dataset [18] consists of 13 categories (753 segmented flower images) of common flowers in the UK (Figure 4). We exploit the segmentation results and use the same features as described in the previous section. The whole dataset is split into three parts for cross-validation. Half of the images are used for training while validation and test set divide the remaining images equally.

As shown in Table III, feature selection improves the classification accuracy, while the proposed method (IFS-SVM) achieves the highest performance. In this experiment, AR, AP and AC scores are close since this dataset is more balanced. Other features and machine learning methods might achieve better results. However, we only introduced the improvement

Fig. 4: Flower dataset of 13 common categories in the UK. This task is difficult because the images have large scale, pose and light variations. Some classes are quite similar to others and they both have enormous variations.

of using forward sequential method with a linear SVM, so the result focuses on the variations introduced by different feature selection methods.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	76.6 ± 3.7	78.0 ± 3.5	77.7 ± 3.6
MFS-SVM	81.4 ± 2.2	83.5 ± 2.9	83.3 ± 1.9
IFSSVM	**82.8 ± 1.4**	**85.5 ± 0.2**	**83.8 ± 1.6**

TABLE III: Experiment results on flower dataset, all results are averaged by 3-fold cross-validation.

C. Medical image dataset

The third dataset is composed by 1300 medical images of skin lesions, belonging to 10 classes [19]. 17079 dimensions of color and texture features are extracted and normalized to zero mean and unit variance. PCA is used for feature reduction which preserves the top 98% energy of components' coefficients. It reduces the dimension of features to 197 but loses about 9% accuracy (from 76% to 67%). The result in Table IV demonstrates improvements for both feature selection methods (MFS and IFS). The proposed IFS method is significantly better than the other two methods for all three evaluation criteria with 5-fold cross-validation.

method	Aver. Recall (%)	Aver. Precision (%)	Accuracy by count (%)
M-SVM	58.8 ± 2.5	66.2 ± 3.3	66.9 ± 2.9
MFS-SVM	61.8 ± 4.0	64.4 ± 5.1	70.2 ± 2.9
IFS-SVM	**73.0 ± 5.0***	**76.3 ± 4.0***	**77.0 ± 3.2***

TABLE IV: Experiment results on skin image dataset. All results are averaged by 5-fold cross-validation. * means significant improvement with 95% confidence.

D. Experiment overview

In Figure 5, we give an overview of the performance of the three methods when the number of classes changes. The first row shows the results of the Fish24K dataset. AR, AP and AC

(first three columns) are all decreasing as the number of classes increases. The MFS method (red line) is sometimes worse than the baseline M-SVM method (black line) due to over-fitting. It achieves significant improvement in the validation set, but the performance drops when it is generalized to the test set. The same trend is also observed in the following experiments: the Fish7K dataset, the Oxford flower dataset, the skin lesions dataset. Our proposed IFS method (blue line) outperforms the other two methods and achieves higher performance in all experiments. The last column shows the computing time of feature selection, which illustrates that the IFS method reduces the time cost while keeping the superiority in accuracy.

E. Optimization in computing time

LIBSVM provides its own implementation of multi-class SVM that also uses the OvO strategy. In our experiment here, we use the multiclass LIBSVM, instead of using its binary SVM utility and wrapping to a multiclass SVM in Matlab (MFS-SVM), to process the same forward sequential feature selection method on the datasets. The results are listed in Table V, comparing to the computing time of MFS and IFS methods.

method	Fish24K	Fish7K	Flower	Skin
MFS-SVM	14.34	2.48	0.19	0.92
LIBSVM	5.57	0.24	2.73e-3	0.18
IFS-SVM	3.90	0.48	0.01	0.39

TABLE V: Computing time comparison. The experiment used the datasets described above. The LIBSVM method uses the same OvO strategy as MFS-SVM but is optimized. Thus it provides an estimate of the potential optimization of our proposed method.

IFS-SVM is faster in the Fish24K dataset because it only selects features for two classes so the size of the feature subset is smaller, while the other two methods have to choose more features to balance the accuracy over all classes. This factor becomes more significant when the dataset is large. The LIBSVM method spends less computing time than IFS-SVM in other three experiments. The LIBSVM uses the same procedure as MFS-SVM, but it is more efficient since it implements the multiclass SVM in C++. This experiment also provides an estimate of the potential optimization (2-50x improvement) of the IFS method.

V. Conclusion

In this paper, we propose that individual feature selection in each one-versus-one classifier improves the performance of multiclass SVM. This method could be adapted into any multiclass classifier that is constructed by assembling binary classifiers. We test the proposed method on four different datasets, comparing to the multiclass SVM with forward sequential feature selection. The results demonstrate a significant improvement on all experiments. We also compare the computing time and show the proposed method is more efficient than the normal feature selection mechanism.

Acknowledgment

This work is supported by the Fish4Knowledge project, which is funded by the European Union 7th Framework Programme [FP7/2007-2013] and by EPSRC [EP/P504902/1]. This work has made use of the resources provided by the Edinburgh Compute and Data Facility (ECDF) (http://www.ecdf.ed.ac.uk/). The ECDF is partially supported by the eDIKT initiative (http://www.edikt.org.uk).

References

[1] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural processing letters*, vol. 9, no. 3, pp. 293–300, 1999.

[2] J. C. Platt, N. Cristianini, and J. Shawe-Taylor, "Large margin dags for multiclass classification," in *Advances in Neural Information Processing Systems 12*, 2000, pp. 547–553.

[3] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[4] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 221–228.

[5] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene selection for cancer classification using support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, Jan. 2002.

[6] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler, "Support vector machine classification and validation of cancer tissue samples using microarray expression data," *Bioinformatics*, vol. 16, no. 10, pp. 906–914, 2000.

[7] G. Forman, "An extensive empirical study of feature selection metrics for text classification," *J. Mach. Learn. Res.*, vol. 3, pp. 1289–1305, Mar. 2003.

[8] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," *The Journal of Machine Learning Research*, vol. 2, pp. 45–66, 2002.

[9] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003.

[10] M.-D. Shieh and C.-C. Yang, "Multiclass SVM-RFE for product form feature selection," *Expert Systems with Applications*, vol. 35, no. 1-2, pp. 531–541, Jul. 2008.

[11] Y. Saeys, I. n. Inza, and P. Larrañaga, "A review of feature selection techniques in bioinformatics," *Bioinformatics*, vol. 23, no. 19, pp. 2507–2517, Oct. 2007.

[12] X.-W. Chen, X. Zeng, and D. van Alphen, "Multi-class feature selection for texture classification," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1685–1691, Oct. 2006.

[13] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[14] T. Li, C. Zhang, and M. Ogihara, "A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression," *Bioinformatics*, vol. 20, no. 15, pp. 2429–2437, Oct. 2004.

[15] C. Chih-Chung and L. Chih-Jen, "LIBSVM: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[16] B. Boom, P. Huang, J. He, and R. Fisher, "Supporting ground-truth annotation of image datasets using clustering," in *21st International Conference on Pattern Recognition (ICPR)*, 2012, pp. 1542–1545.

[17] P. X. Huang, B. J. Boom, J. He, and R. Fisher, "Underwater live fish recognition using balance-guaranteed optimized tree," in *ACCV*, vol. 7724, pp. 422–433.

[18] M.-E. Nilsback and A. Zisserman, "Delving into the whorl of flower segmentation," in *Proceedings of the BMVC*, vol. 1, 2007, pp. 570–579.

[19] L. Ballerini, R. Fisher, B. Aldridge, and J. Rees, "Non-melanoma skin lesion classification using colour image data in a hierarchical k-NN classifier," in *2012 9th IEEE International Symposium on Biomedical Imaging*, 2012, pp. 358–361.

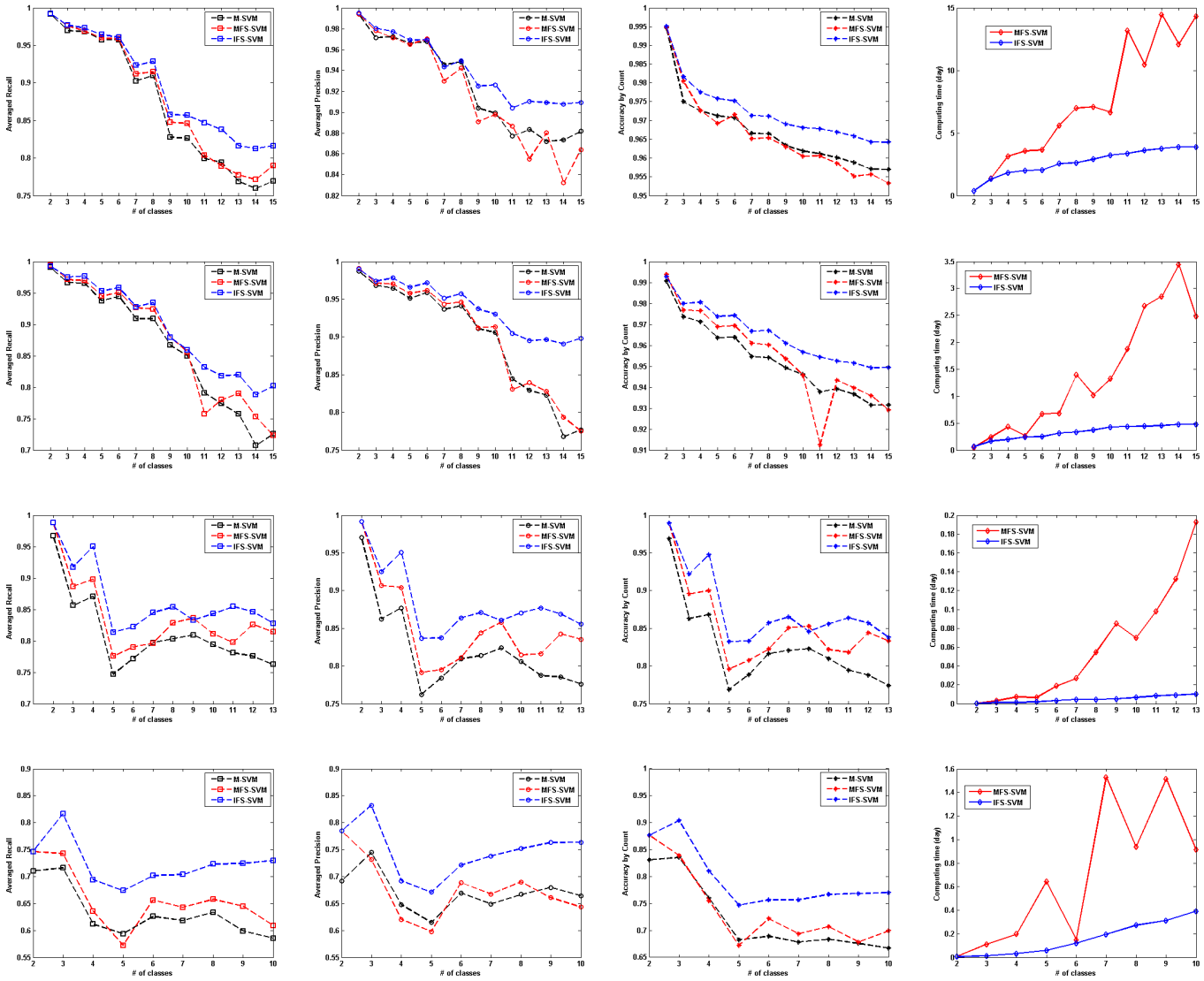

Fig. 5: Performance overview of comparing three methods as the number of classes increases. From left to right: Averaged Recall, Averaged Precision, Accuracy by Count, Computing time. From top to bottom: the Fish24K dataset (24150 images), the Fish7K dataset (6874 images), the Oxford flower dataset (753 images), and the skin lesions dataset (1300 images).