Biological Image Temporal Stage Classification via Multi-Layer Model Collaboration

Tao Meng and Mei-Ling Shyu
Department of Electrical and Computer Engineering
University of Miami
Coral Gables, Florida, USA
Email: t.meng@umiami.edu, shyu@miami.edu

***Abstract*—In current biological image analysis, the temporal stage information, such as the developmental stage in the *Drosophila* development *in situ* hybridization images, is important for biological knowledge discovery. Such information is usually gained through visual inspection by experts. However, as the high-throughput imaging technology becomes increasingly popular, the demand for labor effort on annotating, labeling, and organizing the images for efficient image retrieval has increased tremendously, making manual data processing infeasible. In this paper, a novel multi-layer classification framework is proposed to discover the temporal information of the biological images automatically. Rather than solving the problem directly, the proposed framework uses the idea of "divide and conquer" to create some middle level classes, which are relatively easy to annotate, and to train the proposed subspace-based classifiers on the subsets of data belonging to these categories. Next, the results from these classifiers are integrated to improve the final classification performance. In order to appropriately integrate the outputs from different classifiers, a multi-class based closed form quadratic cost function is defined as the optimization target and the parameters are estimated using the gradient descent algorithm. Our proposed framework is tested on three biological image data sets and compared with other state-of-the-art algorithms. The experimental results demonstrate that the proposed middle-level classes and the proper integration of the results from the corresponding classifiers are promising for mining the temporal stage information of the biological images.**

***Keywords*-temporal stage information; biological image mining; model fusion; biological image classification**

I. Introduction

With the fast development of the biological scientific research domain, data of different categories, such as DNA sequences generated by the second generation sequencing technology, DNA Microarray, RNA-Seq, and biological images like *in situ* hybridization (ISH) images, and microscopic images increase dramatically. Such large amounts of heterogeneous data have revolutionized biological research. Mining significant patterns from these data sets to gain new insights about biological phenomena has become an important problem. The problem becomes more challenging when it comes to mining the biological image domain because of the large volume and special characteristics of the bio-image data. The classic approach is to invite domain experts to inspect the images visually and assign labels. This approach provides qualitative results while suffering from scalability and subjectivity. Recently, the multimedia semantic analysis becomes one of the central topics in multimedia research domain [1][2][3][4]. Therefore, many researchers in multimedia and pattern recognition domains bring new techniques into the biomedical image analysis research and a relatively new research area of bio-image informatics has been formed and developed vigorously [5]. The approaches developed in this area do not only address the issue of scalability and provide quantitative and objective solutions, but also provide the capability to detect visual features which are not readily detectable by humans. Many applications tackling different problems including cell detection [6] and tumor classification [7][8], etc., have been developed to facilitate the research in both the biological and clinical fields.

A biological process usually consists of several temporal stages. The sequential events such as gene expressions and protein-protein interactions at the correct stages, are vital for normal biological activities. For example, the mitosis process consists of the stages of prophase, prometaphase, metaphase, anaphase, and telophase. In each stage, certain steps are taken to promote the progress of the mitosis. In the process of *Drosophila* embryogenesis, the correct sequence of gene expressions and interactions ensure the normal development of an individual organism from embryo to adult. Recently, owning to the rapid advances of the high-throughput microscopic imaging technology, middle-to-large scale biological image repositories have been built and the number of available images instances increases exponentially. A case in point, the BDGP database [9] which is a public database constructed by the University of California at Berkeley, currently contains 112995 ISH images spanning 6 developmental stage ranges of *Drosophila*. The total size of these images is already over 100GB [10].

Such a large volume of data sets makes the manual annotations of the temporal stage information impractical and calls for automatic computational solutions. The existing approaches generally fall into two categories: the general bio-image classification frameworks and the ad hoc solutions for a certain data set. As for the general solution, one of the most famous automatic bio-image classification and pattern recognition platforms is the Wndchrm [11]. This application extracts up to 2873 visual features including polynomial decompositions, high contrast features, pixel statistics, and textures derived from the raw images, transforms of the images, and compound transforms of the images. All features in the feature pool go through a feature selection step so that the least informative features are eliminated. The modified weighted nearest neighbor algorithm is adopted to assign a class label to a testing image. The software was tested using the IICBU [12] benchmark data sets and the performance was given as a baseline. A similar software package is Bioimagexd [13], which focuses more on the cell microscopic image analysis. However, in terms of the temporal stage classification, the general solution ignores the internal relationships among classes. For example, the instances from two consecutive stages are usually more similar and therefore more difficult to be classified than the instances from two non-consecutive stages.

The ad hoc solution is application oriented. For example, it is an essential and non-trivial problem to annotate developing stage ranges for the ISH embryonic developmental images since the comparison of different genes is the most meaningful for the embryos in the same developmental stage range. The framework proposed in [14] is the first study to address this problem. The researchers extracted Gabor texture features [15] from the sub-blocks of the raw images and the Regularized Uncorrelated Linear Discriminant Analysis (RULDA) was utilized to classify 2,705 images from the BDGP database into three developmental stage ranges (Stages 1-3, Stages 4-6, and Stages 7-8). In [16], the researchers first segmented the four blocks from one raw image by manually observing the most significant regions for classification and then extracted Gabor features to feed the multi-class Support Vector Machine (SVM) with the linear kernel for the classification purpose. Their framework was relatively successful for the specific task. However, their proposed framework relies on human inspection and is difficult to be generalized. Another work addressing the same problem is proposed in [17].

In this paper, we focus on the multi-stage classification problem in which the number of stages is greater than or equal to three. This problem is essentially a multi-class classification problem. Such a problem is usually much more challenging than the binary classification problem, which only has two classes. Some binary classification algorithms could be generalized to solve the multi-class classification problem, such as the nearest neighbor approach, Bayes net classifier, and neural network classifier. Some researchers convert the problem to a set of binary classification problems and use the methods of one-versus-all (OVA) [18], all-versus-all (AVA) [19], and error-correct output-coding (ECOC) [20] to address the multi-class classification problem. The one-versus-all (OVA) approach trains one binary classifier for each class. When a testing instance comes, the classifier which outputs the highest probability or score becomes the winner. On the other hand, the all-versus-all approach trains a binary classifier for each pair of classes and uses majority voting to make the final decision. In terms of these approaches, the biggest problem is how to integrate the outputs from different classifiers to make a decision, especially when there is a conflict. In addition, the all-versus-all approach becomes less feasible with the increase of the total number of classes. In order to resolve such issues, hierarchical based classification becomes increasingly popular. Yang [21] proposed one approach to build the classification tree and utilized the relationship between the parent class and child class to help the classification. In that work, a set of binary classifiers forms a binary tree and a testing instance is classified by traversing this tree from the root to the leaf. However, there are two remaining issues which have yet to be solved. The first problem is how to generate the non-leaf class nodes so that the relationships of the classes could be taken into account properly. The second problem is the error propagation issue that once the parent classifier makes an incorrect decision, there is no chance that the instance could be correctly classified. Consequently, there is lack of "cooperation" among different layers of classifiers.

In order to address the aforementioned issues, we develop a multi-layer classification framework based on subspace modeling, middle class creation, and multi-layer regression. In order to utilize the internal relationship of the temporal classes, some middle classes, which are not the classification targets but are closely related and relatively easy to be classified, are created, and the corresponding classifiers are trained. Next, by minimizing the total cost, a proper fusion strategy to integrate information from different classifiers is applied. It is important to point out that the proposed framework is different from the hierarchical approach since our proposed algorithm does not enforce the sequence of decisions, and all the classifiers contribute to the final decision in a proper way.

The paper is organized as follows. The proposed framework is introduced in Section II. The experiment configuration and results are described in details in Section III. Section IV concludes this paper and discusses some future work for further study.

II. The proposed framework

Our proposed framework is shown in Figure 1 and Figure 2. It consists of the training phase and the testing phase. In the training phase, the training images are first preprocessed, where visual descriptors including color, texture, edge, etc. are extracted from each image. The feature selection step selects the most significant features. Afterwards, the target layer subspace-based classifiers are trained and the middle classes are created based on the output scores from the target layer classifiers. The subspace-based classifiers are trained for each middle layer class and the final classification decision is made by fusing the scores from the middle layer

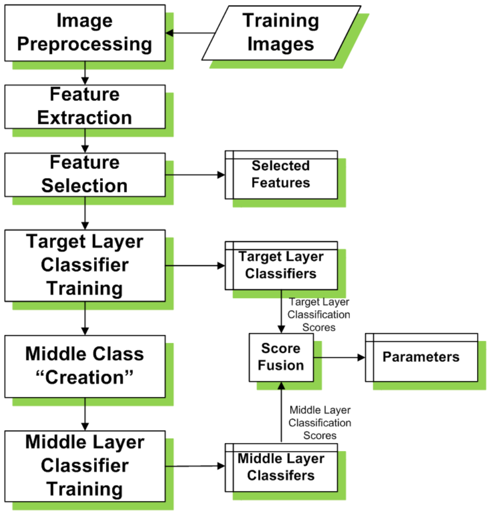


Figure 1. The training phase of the proposed framework

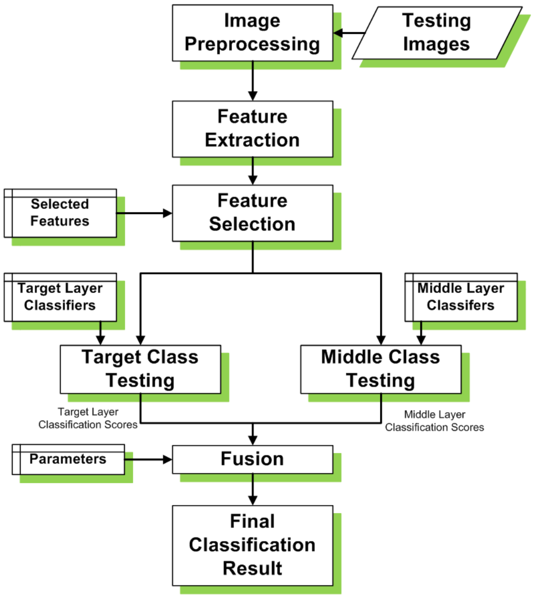


Figure 2. The testing phase of the proposed framework

classifiers and target layer classifiers. The parameters for fusion and all the classifiers are saved for the testing phase.

In the testing phase, the testing images are preprocessed in the same way as in the training procedure. The same set of features are extracted and the features corresponding to the selected features in the training stage are retained. Next, a testing image is input into the middle layer classifiers and target layer classifiers, and the generated scores are fused as a final ranking score. The details of each step and formal definitions for middle layer classifier and target layer classifier are explained in the following subsections.

A. Image Preprocessing

Since the proposed framework is geared towards the classification of a wide range of biological images, the images may contain a variety of characteristics. As a result, the input raw images are preprocessed in a sequence of steps such as histogram equalization and normalization, if necessary. Particularly, for those bio-images in which the object is clearly discernible from the background, object segmentation is performed. It is important to point out that this step is a regular image processing step but not the focus of this work.

B. Feature Extraction and Feature Selection

In order to apply the machine-learning based approach, the first step is to represent the image using a set of descriptors or features. Such a representation should cover most of the information contained in the images. In fact, a good feature representation of the object/target is an active research field itself and it never stops developing. Therefore, the proposed framework is designed to be flexible so that it can accept the inputs of any feature descriptors provided by the users. If there are no features specified by the user, we provide a default feature set consisting of 640 visual descriptors. The detailed description of these features could be found in [22]. Next, the chi-square feature selection approach is applied to select the most significant features and the number of retained features is decided in an empirical study.

C. Subspace-Based Classifier

The subspace-based model receives a lot of attention in the pattern recognition research field [23]. In this work, the CRSPM model [24] is used as the classification model. The general idea is to build an array of Principal Component Classifiers (PCCs) and each of which is trained to learn the similarities among the data instances from a particular class. The details of this algorithm were elaborated in [24] and omitted. Formally, assuming there are N classes in total, the approach builds N binary classification models denoted as PCCs and each of which outputs a posterior probability that a data instance belonging to a certain $Class_n (1 \leq n \leq N)$. The final classification decision is made by choosing the class label which corresponds to the highest posterior probability. If there is a tie, the decision will be purely determined by the prior probabilities.

D. Middle Class and Multi-Layer Model Collaboration

The preceding classification framework aims at solving the problem directly and differentiates all the classes at once. This relatively strict requirement presents challenges to the model. On the other hand, when facing a difficult problem, humans usually adopt the strategy of "divide and conquer" to break the problem into small pieces, solve them first, and integrate the solutions to solve the difficult one.

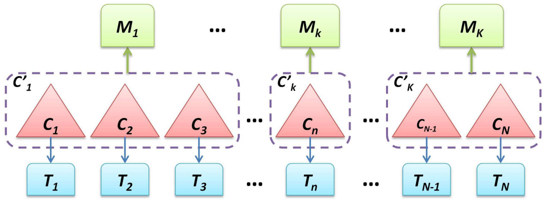


Figure 3. Architecture of the two-layer classifiers. The dashed rectangles indicate the middle classes which consist of different numbers of target classes

Inspired by this strategy, some middle level classes which are not our targets but are helpful in solving the target problem are created. Correspondingly, the proposed PCCs are built for these middle level classes. In order to eliminate ambiguity, we define the class created as a middle class and the corresponding classifier as the middle layer classifier while defining the class for the initial classification task as a target class and the corresponding classifier as the target layer classifier. Figure 3 shows an example of the structure of two-layer classifiers. In this figure, without loss of generality, C_1, C_2, ... C_n ..., C_N represent N target classes, C'_1, C'_2, ..., C'_k, ..., C'_K represent K middle classes. T_1, T_2, ..., T_n, ..., T_N represent N target layer classifiers corresponding to N target classes; while M_1, M_2, ..., M_k, ..., M_K represent K middle layer classifiers corresponding to K middle classes. In the special case, the middle class could just have one target class and the middle layer classifier is the same as the target layer classifier. For example, C'_k in the figure only contains C_n and $M_k = T_n$ in this case. Please note that the number of layers is not limited to two and it could be extended for more complicated tasks. Also, the formation of the middle level classes could be changed.

In order to deploy this multi-layer classification architecture, two essential problems need to be solved. First, how to create middle classes properly to capture the internal temporal relationship among the target classes? Second, how to combine the probabilities or scores output from different layers of classifiers without bringing the problem of error propagation? In this work, we propose the solutions described as follows.

In terms of creating the middle classes, one way is to consult the domain expert. If such information is unavailable, the data-driven determination strategy is proposed here. The main idea of this approach is that the middle classes should benefit each target class and also match the sequential relationships of the temporal stages. The whole procedure of creating the middle classes is shown in Figure 4.

The target classifiers are first trained using the training instances and the classification results are obtained on the validation data set. Based on the classification results, the confusion matrix is computed so that the row of the matrix is sorted according to the temporal sequence of the classes.

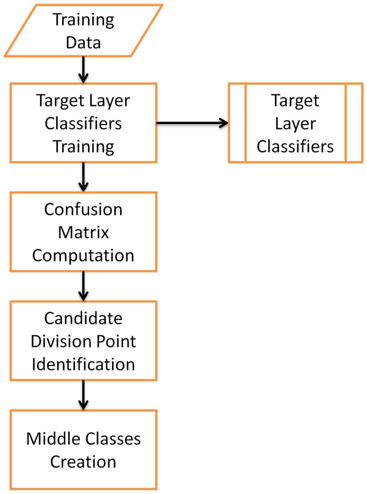
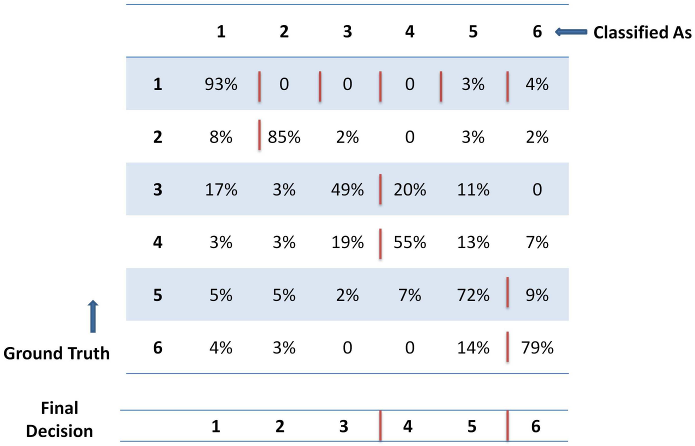

Training Data
Target Layer Classifiers Training
Target Layer Classifiers
Confusion Matrix Computation
Candidate Division Point Identification
Middle Classes Creation

Figure 4. Flowchart of the procedure to “create” middle classes


Classified As
Ground Truth

	1	2	3	4	5	6
1	93%	0	0	0	3%	4%
2	8%	85%	2%	0	3%	2%
3	17%	3%	49%	20%	11%	0
4	3%	3%	19%	55%	13%	7%
5	5%	5%	2%	7%	72%	9%
6	4%	3%	0	0	14%	79%
Final Decision	1	2	3	4	5	6

Figure 5. A numerical example of the confusion matrix and the division scheme. The candidate division points are labeled as the vertical bars in the confusion matrix. The final decision of the division points is illustrated at the bottom of the figure.

Next, the candidate division points are found for each target class by identifying the class which corresponds to the largest classification error and the final grouping strategy is decided by combining the candidate division points of each target class using the majority voting strategy with the constraint that the total number of middle classes is less than or equal to half of the total number of target classes and greater than or equal to two. Figure 5 shows the confusion matrix for the classification results based on the validation data set for a specific example with 6 target classes. In this matrix, the rows indicate the ground truth class labels and the columns indicate the predicated class labels. The classes 1, 2, ..., and 6 represent the 6 consecutive temporal stages.

Each element in the row represents the ratio of the number of instances predicted as a certain class so all the diagonal elements are correct while all the off-diagonal elements are incorrect. For instance, for all the instances which are from $Class_1$, 93% are predicted as $Class_1$, 3% are predicted as $Class_5$, and 4% are predicted as $Class_6$. Therefore, the class which corresponds to the largest error is the $Class_6$ and the candidate division points are between $Class_1$ and $Class_6$ (shown as the vertical bars in the figure). Next, the final decision is made using the majority voting strategy. In this example, given the constraint that the number of middle classes should be less than or equal to three, two division points are chosen and three middle classes are created. For the special case that overall accuracy is 100% which indicates the target classifiers are perfect for the testing data and two middle classes are formed so that the total number of positive instances for each middle class are as close as possible. It is noticed that the temporal sequence of different classes is not changed so that only the consecutive stages could be grouped together. In this way, the internal connections among classes are properly integrated into the proposed classification model.

After the middle classes are created, the PCCs described in Section II-C are trained for middle classes. When a testing instance is input into the system, each classifier of the middle classes and target classes is going to output the posterior probability. Therefore, another important problem is how to combine the outputs from different layers of classifiers to generate the final score for each target class. In this paper, a multi-layer regression model is proposed.

Formally, for an instance i $(1 \leq i \leq m)$ where m is the total number of instances, the output score from the middle layer classifier M_k is represented as $SM_k^{(i)}$ and the output score from the target layer classifier T_n is represented as $ST_n^{(i)}$. In addition, a weight matrix $\boldsymbol{\alpha}$ with a dimension of $(K+2)$ by N is defined so that each column of the matrix corresponds to the weights to combine the outputs for a target class. The final score of instance i for a target class n is represented using $SF_n^{(i)}$ which is defined using Equation (1).

$$SF_n^{(i)} = 1 \cdot \alpha_{0,n} + \sum_{k=1}^{K} SM_k^{(i)} \cdot \alpha_{k,n} + ST_n^{(i)} \cdot \alpha_{(K+1),n} \quad (1)$$

In this paper, the linear model is utilized here to save the computational cost. In order to estimate parameter matrix $\boldsymbol{\alpha}$, let $y(i)$ represent the ground truth label for instance i, where $1 \leq y(i) \leq N$. The cost function is defined in Equation (2).

$$J = \frac{1}{m} \cdot \sum_{i=1}^{m} [(SF_{y(i)}^{(i)} - 1)^2 + \sum_{n=1, n \neq y(i)}^{N} (SF_n^{(i)})^2] \quad (2)$$

From the equation above, the cost function measures the distance between the prediction and the ground truth label represented as a vector. In addition, it is a quadratic function with respect to $\boldsymbol{\alpha}$ so minimizing this cost function turns out to be a convex optimization problem. In this paper, the problem is solved using the gradient descent algorithm. The corresponding derivative with respect to the element of $\boldsymbol{\alpha}$ is defined in Equation (3), Equation (4), and Equation (5). For one instance, $SF_n^{(i)}$ will be the final output for $Class_n$, and the instance is assigned to the class which has the largest SF value.

$$\frac{\partial J}{\partial \alpha_{0,n}} = \frac{1}{m} \sum_{i=1}^{m} \delta_n^{(i)}, \quad (3)$$

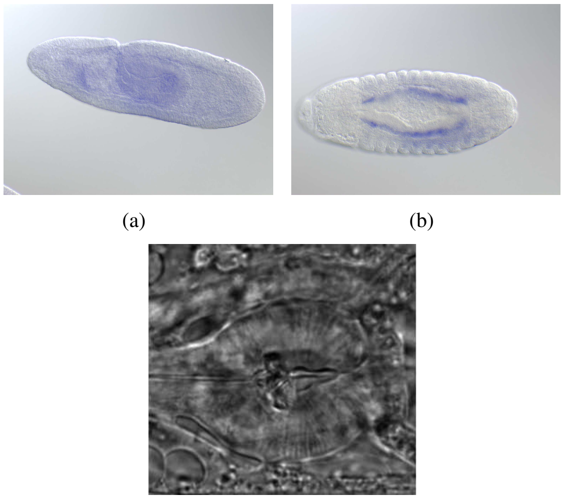
$$where \;\; \delta_n^{(i)} = \begin{cases} 2 \cdot (SF_{y(i)}^{(i)} - 1) & if \;\; n = y(i) \\ 2 \cdot SF_n^{(i)} & if \;\; n \neq y(i) \end{cases}$$

$$\frac{\partial J}{\partial \alpha_{k,n}} = \frac{1}{m} \sum_{i=1}^{m} \theta_{k,n}^{(i)} (1 \leq k \leq K), \quad (4)$$

$$where \;\; \theta_{k,n}^{(i)} = \begin{cases} 2 \cdot (SF_{y(i)}^{(i)} - 1) \cdot SM_k^{(i)} & if \;\; n = y(i) \\ 2 \cdot SF_n^{(i)} \cdot SM_k^{(i)} & if \;\; n \neq y(i) \end{cases}$$

$$\frac{\partial J}{\partial \alpha_{K+1,n}} = \frac{1}{m} \sum_{i=1}^{m} \xi_n^{(i)}, \quad (5)$$

$$where \;\; \xi_n^{(i)} = \begin{cases} 2 \cdot (SF_{y(i)}^{(i)} - 1) \cdot ST_n^{(i)} & if \;\; n = y(i) \\ 2 \cdot SF_n^{(i)} \cdot ST_n^{(i)} & if \;\; n \neq y(i) \end{cases}$$


(a)

(b)

(c)

Figure 6. Sample images from the three data sets: (a) a sample image from D1; (b) a sample image from D2; and (c) a sample image from T1.

III. Experimental results

A. Data Sets and Evaluation Criterion

In order to evaluate our proposed framework, three data sets were used in experiments. We randomly selected 1475 lateral view images and 2000 dorsal view images from the BDGP database to form the Lateral View data set (D1) and Dorsal View data set (D2). In addition, The terminal aging data set (T1) was selected from the IICBU database as it proves to be the most challenging problem among all the classification tasks [11]. The distributions of the data instances corresponding to class labels are shown in Table I and Table II. The sample images from D1, D2 and T1 data sets are shown in Figure 6.

In this study, we adopted classification accuracy, which is a common evaluation criterion in multi-class classification domain, to test the performance of different frameworks. Assuming for one instance i ($1 \leq i \leq m$) where m is the total number of instances, $C(i)$ is the output class label, $y(i)$ is the ground truth label, and $I(Condi)$ is the indicator function which outputs 1 if the $Condi$ is true and otherwise zero. The accuracy measure is defined in Equation (6). Three-fold cross validation was used in the experiment and the average accuracy is reported here.

$$Accuracy = \frac{1}{m}\sum_{i=1}^{m} I(C(i) = y(i)) \tag{6}$$

Table I
Distribution of Instances Among Classes in D1 and D2

Meaning	No. of Images (D1)	No. of Images (D2)
Stages 1-3	187	9
Stages 4-6	270	142
Stages 7-8	146	197
Stages 9-10	145	263
Stages 11-12	380	422
Stages 13-16	347	967
Total	1475	2000

Table II
Distribution of Instances Among Classes in T1

Meaning	Number of Images (D1)
day 0	106
day 2	218
day 4	159
day 6	176
day 8	195
day 10	62
day 12	54
Total	970

B. Significance of Applying the Multi-Layer Collaboration Model

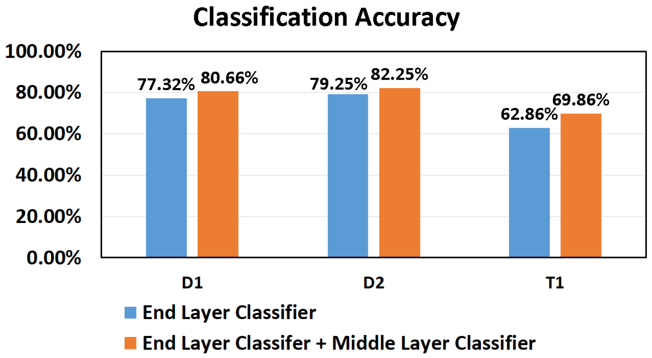


Figure 7. The effect of applying the multi-layer regression model

In order to evaluate the contribution of middle class classifiers, experiments were carried out to compare the performance of the framework with and without multi-layer regression. In Figure 7, "End Layer Classifier" indicates the performance of using target-layer classifiers only; "End Layer Classifier + Middle Layer Classifier " indicates the performance of using the multi-layer collaboration model. It could be seen that the middle class helps improve the target class classification accuracies by 2.3% to 6% for D1, D2 and T1 data sets. It proves that proposed the multi-layer collaboration model is effective in improving the overall classification accuracy.

C. Comparison with Other Classification Algorithms

In order to better evaluate the framework, the proposed framework was compared to other multi-class classification algorithms. Table III shows the performance of different classification algorithms for D1, D2 and T1. "Proposed" denotes the proposed subspace-based classification framework with multi-class regression. The other classifiers used are from Weka [25] where "J48" is the C4.5 decision tree classifier with error reduce pruning; "BN" denotes Bayes network; "NN" indicates the nearest neighbor classifier; "KNNX" stands for the X nearest neighbors; "RFX" is the Random Forest with X trees. "LSVM+L" and "LSVM+R" are LibSVM [26] with linear kernel and RBF kernel correspondingly. "ADA.+LSVMX" is the Adaboost algorithm with LibSVM and X denotes the number of iterations. "NeuroNW" indicates the neuro network algorithm. It could be seen that the proposed algorithm outperforms the other classifiers for all three data sets. For the other classifiers, The neuro network algorithm gives the second best performance, which indicates that the multi-layer classification is suitable for the classification task. However, during experiments we find the performance is highly sensitive to the parameter of the number of hidden layers and the difference could be as large as 40%-50% in terms of accuracy. Accordingly, this raises another research topic of how to choose the number of hidden layers. Besides, the back propagation approach suffers from the time complexity issue. For D1 and D2, the LibSVM with linear kernel performs relatively well. The nearest neighbor approach gives relatively inferior performance which indicates that the training data contain a significant amount of noise. In our proposed subspace-based model, the data in each class are first pruned to remove noise which is one of the reasons that the our proposed algorithm performs better. Random Forest algorithm builds a set of decision tree classifiers and is a relatively popular ensemble learning algorithm. However, it also suffers from the noisy data issue in these two data sets.

Table III
COMPARISON OF OTHER CLASSIFICATION ALGORITHMS

Classification Algorithm	D1	D2	T1
Proposed	**80.66%**	**82.25%**	**69.86%**
J48	53.22%	52.00%	43.67%
BN	46.44%	58.50%	50.61%
NN	53.56%	59.50%	53.06%
KNN10	57.29%	56.75%	59.59%
KNN20	52.88%	54.50%	57.96%
KNN30	54.24%	53.75%	59.18%
RF10	59.32%	61.50%	49.80%
RF20	64.07%	66.25%	53.88%
RF30	64.75%	66.50%	53.88%
RF40	65.08%	66.75%	53.06%
RF50	67.46%	66.75%	55.92%
RF60	66.78%	66.25%	55.92%
LSVM+L	76.98%	78.75%	40.82%
LSVM+R	58.31%	51.00%	22.45%
ADA.+LSVM20	78.30%	79.75%	41.22%
ADA.+LSVM50	78.30%	79.75%	41.22%
NeuroNW	78.33%	80.21%	60.41%

D. Comparison with Other Existing Frameworks

The previous frameworks on the similar tasks were implemented to compare with our current framework. For the data sets D1 and D2, the following two frameworks were used as comparison methods. In the first work [14], the original image was divided into 640 blocks of size 8 x 8 pixels. The Log Gabor features were extracted from each block and formed a 24-dimension vector. Therefore, one original image was represented using a $640 \times 24 = 15360$-dimension vector. The vector was then projected to a lower dimensional space to form a 1280-dimension vector using the modified version of LDA. The nearest neighbor algorithm was then utilized to perform classification. In the second work [16], the authors followed the relatively similar path while picking the significant blocks of the embryos based on human observation as the starting point for feature extraction, the Gabor features were used and the LibSVM was utilized as the classifier. The classification results are shown in Table IV. “RULDA” and “Gabor+LibSVM” represent the approach in [14] and [16], respectively. Since the significant areas in [14] were just selected for lateral view images so that the classification result is not available for D2. It could be seen that the proposed framework outperforms other existing frameworks in terms of classification accuracies on both data sets.

For the data set T1, the Wndchrm framework introduced in [11] was deployed for comparison. For this data set, the same set of features were extracted for our framework and the comparison framework. The comparison results are shown in Table V. In [11], the reported cross-validation accuracy for the same data set is 49%. In our experiments, we got 47.53% as the three fold cross-validation result, which indicates the our implementation was successful. It could be seen that the proposed framework could outperform the comparison framework by a relatively large margin.

Table IV
COMPARISON OF EXISTING FRAMEWORKS

Framework	D1	D2
The proposed	**80.66%**	**82.25%**
RULDA	75.25%	74.75%
Gabor+LibSVM	65.29%	NA

Table V
COMPARISON OF EXISTING FRAMEWORKS

Framework	T1
The proposed	**69.86%**
Wndchrm	47.53%

IV. CONCLUSION AND FUTURE WORK

In summary, a subspace-based multi-layer classification framework for annotating temporal stages for biological images is proposed in this paper. Given the internal sequential relationships among different classes, a prudent and effective grouping strategy is designed to create middle classes to help target class classification. By defining a suitable optimization function and providing a numerical solution, a novel multi-layer model integration algorithm is given to overcome the notorious error propagation issue in the multi-layer classification model. As the detailed comparison results shown in Section III-C and Section III-D, the proposed framework consistently outperforms other classic multi-class classification algorithms in Weka and several state-of-the-art frameworks. It indicates that the proposed framework provides a promising as well as robust automatic solution to the biological temporal stage classification problem and hence could be broadly applied in a research lab.

In the future, a more wide range of biological image data sets will be processed and tested. The middle class creation scheme will be further improved. In order to improve the multi-layer integration algorithm, the proper weights are going to be added to terms in the optimization criterion and a regularization parameter is going to be introduced to address the overfitting issue. Besides, the proposed work is going to be extended to handle the more general case that the classes do not have a sequential relationship.

References

[1] Y. Yang, H.-Y. Ha, F. C. Fleites, and S.-C. Chen, "A multimedia semantic retrieval mobile system based on hidden coherent feature groups," *IEEE Multimedia (in press)*, 2013.

[2] Q. Zhu, L. Lin, M.-L. Shyu, and D. Liu, "Utilizing context information to enhance content-based image classification." *International Journal of Multimedia Data Engineering and Management (IJMDEM)*, vol. 2, no. 3, pp. 34–51, July 2011.

[3] C. Zhang, X. Wu, M.-L. Shyu, and Q. Peng, "A novel web video event mining framework with the integration of correlation and co-occurrence information," *Journal of Computer Science and Technology (in press)*, 2013.

[4] L. An, M. Kafai, and B. Bhanu, "Dynamic bayesian network for unconstrained face recognition in surveillance camera networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 3, no. 2, pp. 155–164, 2013.

[5] H. Peng, "Bioimage informatics:a new area of engineering biology," *Bioinformatics*, vol. 24, pp. 1827–1836, July 2008.

[6] X. Long, W. L. Cleveland, and Y. Yao, "Multiclass detection of cells in multicontrast composite images," *Computers in Biology and Medicine*, vol. 40, pp. 168–178, February 2010.

[7] T. Meng, L. Lin, M.-L. Shyu, and S.-C. Chen, "Histology image classification using supervised classification and multimodal fusion," in *Proceedings of the 2010 IEEE International Symposium on Multimedia*, December 2010, pp. 145–152.

[8] T. Meng, M.-L. Shyu, and L. Lin, "Multimodal information integration and fusion for histology image classification," *International Journal of Multimedia Data Engineering and Management*, vol. 2, no. 2, pp. 54–70, April 2011.

[9] P. Tomancak, A. Beaton, R. Weiszmann, E. Kwan, S. Shu, S. E. Lewis, S. Richards, M. Ashburner, V. Hartenstein, S. E. Celniker, and G. M. Rubin, "Systematic determination of patterns of gene expression during drosophila embryogenesis," *Genome Biology*, vol. 3, pp. 88.1–88.14, December 2002.

[10] P. Tomancak, B. P. Berman, A. Beaton, R. Weiszmann, E. Kwan, V. Hartenstein, S. E. Celniker, and G. M. Rubin, "Global analysis of patterns of gene expression during drosophila embryogenesis," *Genome Biology*, vol. 8, pp. R145–R146, July 2007.

[11] N. Orlov, L. Shamir, T. Macura, J. Johnston, D. M. Eckley, and I. G. Goldberg, "Wnd-charm: Multi-purpose image classification using compound image transforms," *Pattern Recognition Letters*, vol. 29, pp. 1684–1693, May 2008.

[12] L. Shamir, N. Orlov, D. M. Eckley, T. J. Macura, and I. G. Goldberg, "Iicbu2008: a proposed benchmark suit for biological image analysis," *Medical and Biological Engineering and Computing*, vol. 46, no. 9, pp. 943–947, July 2008.

[13] P. Kankaanp, L. Paavolainen, S. Tiitta, M. Karjalainen, J. Pivrinne, J. Nieminen, V. Marjomki, J. Heino, and D. J. White, "Bioimagexd: an open, general-purpose and high-throughput image-processing platform," *Nature Methods*, vol. 9, no. 7, pp. 683–689, June 2012.

[14] J. Ye, J. Chen, Q. Li, and S. Kumar, "Classification of drosophila embryonic developmental stage range based on gene expression pattern images," in *Proceedings of the Computational System Bioinformatics(CSB2006)*, August 2006, pp. 293–298.

[15] B. Manjunath and W. Ma, "Texture features for browsing and retrieval of image data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 837–842, August 1996.

[16] H. Zhong, W.-B. Chen, and C. Zhang, "Classifying fruit fly early embryonic developmental stage based on embryo in situ hybridization images," in *Proceedings of the 2009 IEEE International Conference on Semantic Computing (ICSC2009)*, September 2009, pp. 145–152.

[17] T. Meng and M.-L. Shyu, "Automatic annotation of drosophila developmental stages using association classification and information integration," in *Proceedings of the 2011 IEEE International Conference on Information Reuse and Integration*, August 2011, pp. 142–147.

[18] R. Rifkin and A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, vol. 5, pp. 101–141, December 2004.

[19] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla, Eds., vol. 10. MIT Press, 1998.

[20] F. Masulli and G. Valentini, "Effectiveness of error correcting output coding methods in ensemble and monolithic learning machines," *Pattern Analysis and Applications*, vol. 6, pp. 285–300, February 2003.

[21] J.-B. Yang and I. Tsang, "Hierarchical maximum margin learning for multi-class classification," in *Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11)*. Corvallis, Oregon: AUAI Press, 2011, pp. 753–760.

[22] L. Lin, "Multimedia data mining and retrieval for multimedia databases using associations and correlations," *ACM SIGMultimedia Records*, vol. 3, no. 1, pp. 21–22, April 2011.

[23] G.-Q. Wang, Z.-Y. Ou, D.-T. Liu, and T.-M. Su, "Face recognition using neighborhood preserving discriminant embedding," *Journal of Dalian University of Technology*, vol. 3, pp. 14–15, May 2008.

[24] T. Quirino, Z. Xie, M.-L. Shyu, S.-C. Chen, and L. Chang, "Collateral representative subspace projection modeling for supervised classification," in *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, November 2006, pp. 98–105.

[25] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. Morgan Kaufmann, June 2005.

[26] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.