1-1-2014

HEp-2 cell image classification with multiple linear descriptors

Lingqiao Liu
Australian National University

Lei Wang
University of Wollongong, leiw@uow.edu.au

HEp-2 cell image classification with multiple linear descriptors

Abstract

The automatic classification of the HEp-2 cell stain patterns from indirect immunofluorescence images has attracted much attention recently. As an image classification problem, it can be well solved by the state-of-the-art bag-of-features (BoF) model as long as a suitable local descriptor is known. Unfortunately, for this special task, we have very limited knowledge of such a descriptor. In this paper, we explore the possibility of automatically learning the descriptor from the image data itself. Specifically, we assume that a local patch can be well described by a set of linear projections performed on its pixel values. Based on this assumption, both unsupervised and supervised approaches are explored for learning the projections. More importantly, we propose a multi-projection-multi-codebook scheme which creates multiple linear projection descriptors and multiple image representation channels with each channel corresponding to one descriptor. Through our analysis, we show that the image representation obtained by combining these different channels can be more discriminative than that obtained from a single-projection scheme. This analysis is further verified by our experimental study. We evaluate the proposed approach by strictly following the protocol suggested by the organizer of the 2012 HEp-2 cell classification contest which is hosted to compare the state-of-the-art methods for HEp-2 cell classification. In this paper, our system achieves 66.6% cell level classification accuracy which is just slightly lower than the best performance achieved in the HEp-2 cell classification contest. This result is impressive and promising considering that we only utilize a single type of feature (namely, linear projection coefficients of patch pixel values) which is learned from the image data. 2014 Elsevier Ltd. All rights reserved.

Keywords

linear, descriptors, cell, hep, image, 2, classification, multiple

Disciplines

Engineering | Science and Technology Studies

Publication Details

Liu, L. & Wang, L. (2014). HEp-2 cell image classification with multiple linear descriptors. Pattern Recognition, 47 (7), 2400-2408.

HEp-2 Cell Image Classification with Multiple Linear Descriptors

Lingqiao Liu[a,*], Lei Wang[b,*]

[a] *CECS(College of Engineering and Computer Science), Australian National University, ACT 0200, Australia*

[b] *School of Computer Science and Software Engineering, Faculty of Informatics, University of Wollongong, NSW 2500, Australia*

Abstract

The automatic classification of the HEp-2 cell stain patterns from indirect immunofluorescence images has attracted much attention recently. As an image classification problem, it could be well solved by the state-of-the-art Bag-of-Features (BoF) model if a suitable local descriptor is known. Unfortunately, for this special task, we have very limited knowledge on such a descriptor. Moreover, due to the subtle category differences, the choice of right descriptor becomes crucial for the classification performance.

In this paper, we explore the possibility of automatically learning the descriptors from the image data itself. Specifically, we describe a local patch by a set of linear projections of its raw pixels and both unsupervised and supervised approaches for learning these projections are explored. More importantly, we proposed a multi-projection-multi-codebook scheme which learns multiple descriptors for representing a same patch and builds multiple BoF

[*] corresponding author. Tel & Fax: (61) 2 4221-3771

Email addresses: lingqiao.liu@cecs.anu.edu.au (Lingqiao Liu), leiw@uow.edu.au (Lei Wang)

models for each image. Through our analysis, we show that the image representation obtained by combining these models can be more discriminative than a single-projection scheme. This analysis is further verified by our experimental study.

We evaluate the proposed approach by strictly following the protocol suggested by the HEp-2 contest organizer. In this paper, our system finally achieves 66.6% cell level classification accuracy which is just slightly lower than the best performance achieved in the HEp-2 Cell classification contest. This result is impressive and promising considering that we only utilize a single type of feature learned from the image data – linear projection coefficients of raw pixel value.

Keywords: HEp-2 cell, stain pattern, image classification, feature learning, partial least square, multiple codebooks

1. Introduction

Indirect immunofluorescence (IIF) with HEp-2 cell is a powerful test for analyzing antinuclear autoantibodies (ANA), which is considered as one of the most effective and widely-used diagnostic screening assay. It is able to detect in a timely manner some pathologies whose incidence has been constantly growing in the last few years [1]. To perform the analysis and diagnosis, the identification of staining pattern of samples is required. Among the many staining patterns which can be observed, six of them are relevant to diagnostic purposes, including: centromereseen, homogeneous, nucleolar, coarse speckled, fine speckled and cytoplasmatic, as shown in Figure **??**.

However, their pattern identification is often subjective and low standardized. Hence, it is beneficial to develop the automatic stain pattern identification algoirthms which can serve as a Computer-Aided Diagnosis (CAD) system. Due to its potential applications, the computer vision based stain patterns identification has attracted much attention [2, 3, 4, 5, 6, 7, 8, 9]. In the year 2012, an HEp-2 cells classification contest is hosted to compare the state-of-the-art methods [21].

Essentially, this problem is an image classification problem and it could be solved by employing the state-of-the-art Bag-of-Features model (BoF) based image classification methods [10, 11, 12, 13]. To apply the BoF model, one key premise is to choose a suitable local descriptor for the local patch but unfortunately for this special task we have very limited knowledge on such a choice. Moreover, due to the uniqueness of the cell stain pattern and their subtle category differences, the importance of choosing a right descriptor is more pronounced in HEp-2 cell classification since the classification performance heavily relies on the quality of descriptors. In fact, the searching for a appropriate features is always a key research focus in the related literature.

To design suitable descriptors, most previous works adopt the hand-coded methodology which creates descriptors based on human observation and domain knowledge. Different from the previous approaches, in this paper we explore the possibility of learning the descriptors from the image data itself. Comparing with the methodology of designing hand-coded feature, our approach can be more efficient in adapting to a new task since no domain knowledge but only training data is needed. Also, by carefully designing the learning algorithm, we can relate the descriptor more closely to a given objective e.g. discriminative power. Finally, when necessary, the learned descriptor can also be applied as the compensation of the traditional descriptors.

In this paper, we model the descriptors as a set of linear projection coefficients performed on the raw patch pixels. Though simple in its form, the linear projection descriptor has demonstrated surprisingly good performance in many applications, e.g. action recognition [14], image matching [15]. To learn the linear projections, we explore both the traditional unsupervised PCA approach and supervised Partial Least Square (PLS) regression approach. To our knowledge, this is the first work to employ PLS in the application of descriptor learning. More importantly, we point out that straightforwardly learning a single projection matrix will be problematic due to the presence of common local patches shared by all classes. To overcome its drawback, we proposed a multi-projection-multi-codebook scheme which learns multiple projections from k local regions in the feature space and applies them to a single patch to obtain k descriptors. We build k separate BoF models for each descriptor channel and combine them into the final image representation. Through our analysis, we argue that the image representation obtained in this way can be more discriminative and thus it is able to attain better classification performance.

Through intensive experiment, we demonstrated the advantage of our system and the importance of its building blocks. Evaluated by the protocol suggested by the organizer, in this paper our system finally achieves 66.6% cell level classification accuracy which is just slightly lower than the best performance achieved in the HEp-2 Cell classification contest. This result is impressive and promising considering that we only utilize a data-driven feature – linear projection coefficients of raw pixel value. A simpler version of our approach has won the 4th place in the competition, which outperforms many systems with hand-coded features.

2. Related Work

The focus of the existing works on HEp-2 cell stain patterns classification can be summarized as two aspects: (1) design good features and (2) design good classifier. For the first aspect, the majority of works [2, 3, 4, 5, 6, 7, 8, 9] adopt the global feature to describe the image pattern. The commonly used global features are: statistic features such as the standard deviation [2, 3, 4, 5, 6, 7, 8, 9], entropy [2, 3], intensity [6], etc; the morphological features such as the number of objects or local maximum [2], area [9], length of contour [4] etc; and the texture feature such as Local Binary Pattern (LBP) [2] which has shown promising performance in many texture classification tasks. The local feature and Bag-of-features model have been applied to this problem very recently. In [8], the author adopts a BoF framework with DCT coefficient as the local descriptor. For the second aspect, various classifiers are also tested for this classification tasks, including SVM [6], SMO network [9], Nearest Neighbor Classifier [8] and Adaboost [2]. Also, combining multiple features and multiple classifiers [5] is always a wining trick for good performance. In this paper we focus on the first aspect. However, our methodology is different from the previous ones. The key idea is that instead of manually designing a suitable descriptor we directly learn the descriptor from the image data itself. To our knowledge, this is the first work addressing the HEp-2 stain patterns classification with the automatic feature learning methodology.

Our method can be viewed as the extension of the works using linear projection as descriptor which has long history in computer vision. The most famous early attempt is the eigenface [16] for face classification, where the image feature is learned from the PCA on a number of face images. The examples of employing linear projection to present local features can be found in local feature matching [15] and action recognition[14]. These two examples are also based on PCA but perform on different raw features, one is based on the gradient map of local patch and the other is on the 3-D spatial-temporal cuboid. All of the aforementioned methods are based on single linear projection and the main contribution of our work is to point out the limitation of the single linear projection representation in patch-level descriptor learning and propose a way to overcome the drawback.

Also, since our focus is to explore the performance of feature learning, we restrict ourselves to the usage of data-driven feature only. Certainly, combining with other hand-coded features will be very promising in the regards of improving performance. We will explore this direction in our future work.

3. Our Method

3.1. System overview

Our system is built upon the BoF model (detailed in section 3.2) in which each image is represented as a set of local patches. The novelty of our system is the method of extracting features from local patches and the way to assemble them into the final image representation. The detailed work flow is shown in Figure **??**. First, the brightness normalization is performed for the input image and the local patches are extracted from each image on a dense sampling grid. In the training stage, these patches are firstly sampled and clustered into k groups, which is equivalent to partitioning the feature space into k local regions. Then a PCA/PLS is performed in each group to obtain d projections. The learned projections will be treated as k types of descriptors and they are employed to describe a single image patch. Then we build k BoF models for each type of feature, that is, k separated codebooks are built and each type of feature is encoded/pooled with its own codebook. This will result in k pooled coding vectors and they are concatenated into the final image representation which are fed into a SVM to train a classifier. In the test stage, the image representation generation process will be repeated but we do not need to partition the local patches into k groups since we do not need to learn the projections again. We will simply use the stored k groups of projections and codebooks to perform the descriptor extraction and coding.

In the following, we will discuss each module of our system in more details.

3.2. BoF model for HEp-2 cell classification and preprocessing

By observing the 6 types of HEp-2 cell stain patterns, we find they have the following three visual characteristics:

(1) The brightness of the cell images changes dramatically.

(2) The global shape pattern of different cells shows considerable variations.

(3) The difference between cells are mainly the texture difference.

To handle (1), we perform a brightness normalization scheme as follows: After extracting the luminance value from the color input image, We sort the intensity of all the pixels and find the n^{th} largest intensity v_0. Then we normalize the intensity of all the pixels by dividing them by v_0. The normalized intensity values which are greater than 1 are set to 1. The reason of choosing the n^{th} largest intensity rather than simply selecting the largest intensity is that the former is more robust to the abrupt bright noisy pixels.

The second and the third characteristics motivate us to adopt the Bag-of-Features (BoF) model to represent the cell image. In BoF model, each image is represented as a set of local features $\{\mathbf{x}_i\}$ which are extracted from the local patches. Thus, we essentially use local features to capture the texture differences between cells. Once the local feature are extracted, they are then encoded by a coding method which transforms a local feature to a sparse high dimensional coding vector $\mathbf{v}_i$. Various coding methods have been proposed recently [13, 12, 11]. In this paper, we adopt Local Soft-assignment Coding [13] since it has shown good trade-off between the classification performance and encoding complexity. To apply LSC, a codebook is needed and it can be obtained by performing k-means clustering on a set of sampled features. Finally, the coding vectors for all the local features are pooled into the final image representation. In the literature, two pooling methods, sum-pooling and max-pooling are usually employed. Sum-pooling calculates the pooled image representation via $\bar{\mathbf{v}} = \sum_i \mathbf{v}_i$ while max-pooling is obtained by $\bar{\mathbf{v}} = \max_i \mathbf{v}_i$, where max is the maximum operator performs on each dimension of the coding vectors.

As mentioned in the introduction section, we have very limited knowledge on how to describe the special texture of cell images and we resort to methodology of learning the appropriate descriptor from the image data itself. In our system, the descriptor is in the form of linear projection response. The linear projection is performed on the raw patch pixels. For the image patch, we densely extract them from the cell images and each s by s sized patch is flatten into a s^2 dimensional vector.

3.3. Projection learning and Partial Least Square Regression review

The most conventional way of learning the projection is Principal Component Analysis (PCA) which is an unsupervised method. As mentioned above, this simple strategy has been successfully applied in various areas in computer vision. Its good performance comes from two folds: (1) PCA reduces the feature correlation (2) by discarding the eignvectors corresponding to the small eigen values, the PCA projection can extract the principal structure in the data and reduce the noise.

The PCA projection does not consider the supervised information, thus it may be sub-optimal for the task of discrimination. One supervised counterpart of PCA is Partial Least Square (PLS) regression which has demonstrated impressive performance on supervised dimension reduction in computer vision literature [17]. It is promising to apply it in the descriptor learning step. We choose Partial Least Square to create the discriminative directions because it has many nice properties which make it very suitable for our application. Unlike other popular discriminative subspace learning methods such as LDA, the number of components learned from PLS can be larger than the number of image classes. Thus it allows us to create a large number of projections and enriches the diversity of the resulted descriptors. Moreover, the scalability of PLS is very good: it can easily handle a huge number of training samples and calculate the components with little computational cost.

In the following, we briefly introduce the concept and the calculation of PLS and we leave more detailed theoretical analysis and applications to the related literature [18, 19, 20].

Let $\mathbf{X} \in \mathbb{R}^{n \times d}$ be the data matrix in which each row $\mathbf{x}_i$ represents a d dimensional feature vector and let $\mathbf{Y} \in \mathbb{R}^{n \times c}$ be another data matrix with each row $\mathbf{y}_i \in \mathbb{R}^c$ indicating the c response variables which we want to predict from $\mathbf{x}_i$. In the case of supervised projection learning, the response is the class label indicator, e.g. suppose the class label for the ith sample is j, the indicator vector $\mathbf{y}_i$ will have its j th entry equal to "1" and the others equal to "0". After centralizing $\mathbf{X}$ and $\mathbf{Y}$ by subtracting their mean respectively, PLS models the relationship between $\mathbf{X}$ and $\mathbf{Y}$ as:

$$\begin{aligned} \mathbf{X} &= \mathbf{T}\mathbf{P}^T + \mathbf{E}, \\ \mathbf{Y} &= \mathbf{U}\mathbf{Q}^T + \mathbf{F} \end{aligned} \tag{1}$$

Where, $\mathbf{T} \in \mathbb{R}^{n \times p}$ and $\mathbf{U} \in \mathbb{R}^{n \times p}$ are seen as the latent components while $\mathbf{P}$ and $\mathbf{Q}$ are seen as the coefficients. $\mathbf{E}$ and $\mathbf{F}$ are the residuals. In general, PLS decomposes $\mathbf{X}$,$\mathbf{Y}$ into their latent components $\mathbf{T}$,$\mathbf{U}$ and maximizes the covariance between $\mathbf{T}$ and $\mathbf{U}$. The latent components given by $\mathbf{T}$ can be calculated by a linear transformation of $\mathbf{X}$:

$$\mathbf{T} = \mathbf{X}\mathbf{W} \tag{2}$$

In our application, $\mathbf{W}$ is the discriminative projection we want to learn and $\mathbf{T}$ is the projection response. There are two popular implementations of PLS: SIMPLS [19] and NIPALS algorithms [20]. We adopt NIPALS algorithm to calculate $\mathbf{W}$ in our system and list the algorithm outlined in Algorithm 1.

Algorithm 1 NIPALS Algorithm

```
Require: Covariance Matrices $\mathbf{A}_0 = \mathbf{X}^T\mathbf{Y}$, $\mathbf{M}_0 = \mathbf{X}^T\mathbf{X}$, $\mathbf{C}_0 = I$
for $t = 0$ to $p - 1$ do
    Compute $\mathbf{q}_t$ the dominant eigenvector of $\mathbf{A}_t^T\mathbf{A}_t$
    $\mathbf{w}_t = \mathbf{C}_t\mathbf{A}_t\mathbf{q}_t$, $\mathbf{w}_t = \mathbf{w}_t/\|\mathbf{w}_t\|$ and store $\mathbf{w}_t$ into $\mathbf{W}$ as a column
    $\mathbf{r}_t = \mathbf{M}_t\mathbf{w}_h$, $c_t = \mathbf{w}_t^T\mathbf{M}_t\mathbf{w}_t$, $\mathbf{r}_t = \mathbf{r}_t/c_t$
    $\mathbf{A}_{t+1} = \mathbf{A}_t - c_t\mathbf{r}_t\mathbf{r}_t^T$ and $\mathbf{M}_{t+1} = \mathbf{M}_t - c_t\mathbf{r}_t\mathbf{r}_t^T$
    $\mathbf{C}_{t+1} = \mathbf{C}_t - \mathbf{w}_t\mathbf{r}_t^T$
end for
```

Note that NIPALS algorithm only takes the covariance matrix between $\mathbf{x}$ and $\mathbf{y}$, the covariance matrix of $\mathbf{x}$ as the input. These two matrices can be efficiently calculated by accumulating $\mathbf{x}_i^T\mathbf{y}_i$ and $\mathbf{x}_i^T\mathbf{x}_i$, that is, *we only need to scan all the local patches once to obtain these two matrices.* Thus, NIPALS can be easily adopted to handle a large number of training samples.

3.4. Multiple projection learning and multiple-codebook

Once the projection learning method is established, one can apply it to the raw patch pixels in the training set to obtain the linear projection descriptor. Traditionally, a single global projection matrix is learned and utilized. However, this scheme may be problematic due to the following reason: generally speaking, the local patch in the image set can be divided into two categories, the informative patch and the common patch. The former contains the informative visual pattern which can be utilized to distinguish one class from the other. The later represents the patches with similar/same appearance shared by many classes, e.g. homogeneous region. Ideally, it is favorable to learn the projection only on the informative patches because by doing so the learned projection can be more meaningful for the informative patches. Note that in the codeword assignment step the learned projection will affect the distance evaluation between a patch and a visual word. Thus, maintaining a good projection for informative patches will help to reduce the inappropriate assignment for informative patches and consequently improve the discriminative power the resulted image representation.

However, If we learn a single global projection from both informative patches and common patches, the common patches will act as outliers and confuse the PCA/PLS. In particular, for the case of PLS, we often assume that the local patches share the class label of the image from which they are extracted. So a common patch may be assigned to multiple class labels. To illustrate this situation, let's consider a toy experiment shown in Figure **??**. The red and blue dots in this figure are samples coming from two categories. In this experiment, each sample is a 2 dimensional vector. We intentionally make the sample distribution of these two classes have a large portion of overlapping. Thus it mimics the case that common patches are assigned to different class labels, where the common patches are denoted as the dots in the overlapping regions. We then calculate the principal partial least square projection $\mathbf{p}$. Together with the sample mean $\bar{\mathbf{x}}$, it constructs a hyperplane $\mathbf{p}(\mathbf{x} - \bar{\mathbf{x}})$ cutting through the feature space. In Figure **??** (a), we show the obtained hyperplane by performing the PLS on all samples. It is clear that, the PLS is confused by the common features and the learned discriminative direction (the direction perpendicular to the hyperplane) is not that discriminative. If we project the samples onto the direction, the samples coming from the two categories will be mixed together.

To overcome this issue, in this paper we instead adopt a very simple approach: we pre-cluster the local patches into several groups and perform the projection learning within each group individually. Then we treat the projection learned from each group as a type of local descriptor and apply the BoF model for each of them. In other words, for each group, we apply its associated projections to a local patch to extract its own version of local descriptor; then we sample these local features to build a codebook and use it to perform coding, pooling. Suppose there are k groups, it will result in k pooled coding vectors, We then concatenate the k vectors to obtain the final image representation.

The idea of this scheme is that after pre-clustering the common patches and the informative patches will very likely to be assigned to different groups since their appearances are often quite different. Then for the group to which the informative patches are assigned, the density of the common patches is reduced and thus better projections are expected to be learned. Certainly, for the group which has many common features the learned projections may be less informative. However, since we will apply SVM (linear or additive kernel) to the resulted the pooled vector, the pooled vectors coming from the less informative groups will be automatically down-weighted since they are less relevant for the classification. Applying this idea to the toy experiment shown in Figure **??** (b) by clustering the samples into 3 groups and performing PLS within each group, we can obtain some very good projections, such as hyperplane 1 and 2. It can be seen that they well separate the samples coming from the two classes in a local region. In the same time, it may also result in some less discriminative projections, such as the one denoted by hyperplane 3. However, since we will apply a second stage classifier on top of the projections, we can expect that the coding result obtained by hyperplane 3 will be less weighted.

Note that pre-grouping is only applied to learn the projections. Once the projections are learned, we do not perform the group assignment anymore. Also, the sampled patches which are utilized to construct a group specific codebook do not necessarily belong to the same group. They are just projected by the linear projection learned for that group. An alternative way is to only use the patches belonging to the same group to construct a group specific codebook and apply this codebook to encode the patches which are assigned to same group. This scheme is more computational efficient. However, this scheme tends to introduce much more quantization error. In fact, our experiment suggests that the performance obtained by using this scheme is much inferior to the first one.

4. Experiment

In this section, we conduct a series of experiments to evaluate the performance of the proposed system on 2012 HEp-2 cells classification contest dataset [21]. This dataset contains 28 images with different number of cells per image. For more information about the image acquitision and preprocessing, please refer to [21] for more details. These cells are pre-segmented and the major task is to classify these cells into the six stain patterns. Two different evaluation criteria are suggested by the organizer: the first one is same as the protocol utilized in the contest in which the 28 images are partition into training and test groups with 14 images per group; the second one follows the leave-one-out strategy which recursively chooses one image as the test set and the remaining images as the training set.

The proposed system contains a few free parameters, e.g. the patch size, the codebook size etc. The choice of these parameters will directly affect the final classification performance. To tune these parameters, we adopt a cross-validation approach: we randomly partition the released training images in the contest setting into a training set and a validation set for 5 times. Then we test the average performance of various parameter settings on the validation set and pick up the best setting for the evaluation on the released test set. Thus, our experiments strictly follow the scenario in the contest stage. Note that better performance may be obtained by tuning the parameters directly on the test set. But this is not fair since in practical setting, we do not have access to the test data and label in the training stage.

In the following, we organize the experiments into three parts with different objectives. In the first part, we perform the cross-validation to choose the best parameters. The middle results in this parameter-tuning process is also reported to show the impact of different parameter settings. In the second part, we report the performance of the proposed system by strictly following the evaluation criteria required by the organizer. In the third part, we compare our system with several comparable alternatives and validate the importance of the building blocks of our system.

4.1. Part I: Parameter setting test

The main parameters involved in our systems are as follows:

(1) Feature extraction step: The patch size.

(2) Descriptor generation step: The dimension of the learned descriptor; The number of groups. The projection method, PCA or PLS.

(3) Coding step: coding and pooling method. size of codebook. The usage of Spatial Pyramid [22].

(4) Classification step: classifier.

To make the parameter tuning tractable and for the sake of clear presentation, we divide the test parameters into two groups. The first group includes the choice of the coding, pooling method, spatial pyramid and the classifier, which are considered as the "structural" parameters. The remaining parameters comprise the second group, which are "numerical" parameters. Two projection methods, PCA and PLS are separately evaluated in the tests of both parameter groups.

We will first test the structural parameter by setting the numerical parameters to some empirical values: the patch size is 9x9 pixels. The dimension of the learned descriptor is set to 40 and 5 groups are used. The codebook size equals to 1000. These parameters are used as the default values throughout this section.

Structural parameter test: For the coding and pooling method, we test three combinations: (1) hard coding with sum-pooling, which is the conventional histogram representation. (2) Local soft-assignment coding (LSC) [13] with max-pooling. LSC has shown good trade-off between the efficiency and performance. The usage of max-pooling has demonstrated the state-of-the-art performance in generic image classification. (3) Local soft-assignment coding with sum-pooling.

For the Spatial Pyramid, we partition the image into 1x1, 1x3 and 2x2 grids. This partition is adopted in PASCAL image classification competition [23].

For the classifier, we employ SVM with three different types of kernels: (1) linear kernel (2) Hellinger's kernel, which is defined as $K(\mathbf{x},\mathbf{y}) = \sum_k \sqrt{x_k y_k}$. (3) χ^2 kernel, which is defined as $K(\mathbf{x},\mathbf{y}) = \exp(-\frac{1}{2A}\sum_k \frac{(x_k-y_k)^2}{x_k+y_k})$. The parameter A is set as the average pairwise χ^2 distance $(\sum_k \frac{(x_k-y_k)^2}{x_k+y_k})$ in the training set.

The classification accuracy obtained from various structural parameters are shown in Table 1. From it we can draw the following conclusions: (1) For the method of calculating projections, PLS consistently outperforms PCA when sum-pooling is used and achieves the best performance. This is not surprising since PLS leverages the discriminative information while PCA does not. (2) For the coding and pooling method, interestingly the best performance is obtained by using LSC with sum-pooling, this is contrary to the conclusion in generic object classification in which the soft-assignment coding with max-pooling tends to perform better. This is probably because in stain pattern classification the occurrence frequency information is helpful for distinguishing different categories and sum-pooling can better capture this information than max-pooling in which multiple occurrences only count once. (3) No significant improvement is observed by using SPM except the case of max-pooling. The improvement in the max-pooling case is probably because when SPM is used, the occurrence information can be implicitly inferred from the pooled coding vector in different spatial grids. e.g. if we find the occurrence of the k^{th} visual word in all 8 spatial grids, we can infer that it occurs at least 8 times.

In the following, we investigate the impact of each numerical parameters by fixing the other numerical parameters as the default values. For these experiments, we simply use LSC with sum-pooling as the coding/pooling method and Hellinger's kernel SVM as the classifier due to their good performance. Spatial Pyramid is not utilized for the these experiments. This is because the usage of SPM will produce very high dimensional image representation in some parameter settings and this will cosume a lot memory. According to the last experiment, for LSC with sum-pooling the performance obtained from the SPM is similar to that attained without SPM. Hence, we believe this is a reasonable experimental protocol.

Parameter test for codebook size and group size: In this experiment, we test the classification performance of the proposed system with different number of groups and various codebook sizes. The result is shown in Figure **??**. Clearly, we can see that the advantage of using multiple groups over a single group is quite prominent. Around 6%-10% improvement can be observed depending on the codebook size. This phenomenon well supports our motivation of using multiple groups to learn the projection. However, once the number of groups is greater than 1, the performance becomes quite similar if same-sized codebook is employed. This may be due to that the visual patterns of the common patches in the cell image are quite uniform such that only by clustering the patches into few groups we can well separate them from the discriminative ones. Again, we observed that PLS produces better performance than PCA with same number of groups and codebook size. This well justifies the advantage of discriminative projection learning. For the codebook size, we observed that once the size is greater than 2000, the performance becomes quite similar.

Parameter test for patch size and dimension of the learned descriptor In this experiment, we test the impact of image patch size and the number of projections per group (the learned descriptor dimension) on the classification performance. More specifically, we experiment with various image patch sizes $(7 \times 7, 9 \times 9, 11 \times 11, 13 \times 13, 15 \times 15)$ and reduced dimensions (20, 40, 60, 150). The result is presented in Figure **??**. It can be seen that the optimal patch size is 11×11 and the performance becomes similar with different number of projections once it exceeds 20.

4.2. Part II: Performance Evaluation

Mimicking the scenario in the contest, in this section we "submit" a system with the best parameter setting discovered in the previous section and evaluate its performance by strictly following the two required experiment protocols – the contest setting and the leave-one-out setting. The best parameters in our "submitted system" are as follows: number of groups: 5 groups; codebook size: 2000 words; patch size:11×11pixels; reduced dimensionality: 40; projection learning method: Partial Least Square; Usage of Spatial Pyramid: Yes; Classifier: SVM with Hellinger's kernel.

4.2.1. the first evaluation criterion

We achieved the overall classification accuracy 66.6%. Note that this result is better than the one obtained by our submitted system in the contest which is just around 62.5%. This is because in our contest version, we did not employ the LSC coding and SPM. By comparing with the performance of other systems listed in the contest report, we found our new result is only slight lower than the best result in the contest (around 67%).

The confusion matrix of our system is demonstrated in Figure **??**. It can be seen that the centromere and cytoplasmatic categories can be well separated from the other patterns. Both of them achieve over 90% classification accuracy. However, the other four categories are very difficult to classify. For example, almost half of samples in nucleolar are mis-classified as homogeneous. By looking at the original image of these confused categories, we found their differences are subtle and there are substantially distribution drift from the training set and test set. In other words, the image of a category defined in the training set could be visually different to the one in the test set. This distribution drift probably explains why the performance evaluated in the training-validation set is much higher than the training-test set in the contest setting.

By using the majority voting scheme suggested by the organizer, we achieves the image-level classification accuracy 78.57 %. The confusion matrix of the image level classification is shown in Figure **??**. We can see that the pattern in the confusion matrix is quite similar to the cell-level confusion matrix. The centromere and cytoplasmatic categories are well classified. Also, fine speckled pattern tends to be well classified in this experiment. However, due to the limited number of test samples (only 14 in this case) the accuracy can change dramatically by correctly or incorrectly classifying one image. Hence, this observation may not be statistically stable.

4.2.2. the second evaluation criterion

For the second evaluation criterion, we achieve 58.92% average cell-level classification accuracy. The detailed cell classification accuracy for each image is presented in Table 3. We can see that the classification accuracy changes dramatically from image to image: for some images, e.g. the 28th image, the classification accuracy can be as high as 100%, but for some images, e.g. the 17th image, the classification accuracy is extremely low. Again, this phenomenon can be well understood by the distribution drift of training/test set. Evidently, for some images, the visual appearance is quite different from the other images labeled as the same category. The confusion matrix of this classification setting is shown in Figure **??**. Again, the confusion matrix is similar to the cell-level classification in the contest setting but the homogeneous pattern is better identified than the case in contest setting. Similar pattern is observed in the image-level classification matrix, as shown in Figure **??**.

4.3. Comparison and Discussion

In this section, we compare the proposed system with three comparable methods. We first compare our method with the BoF model with LBP feature – a commonly used texture descriptor. In our implementation of this method, all other parameters, e.g. code book size, patch size are fixed to the optimal setting obtained in the previous section. The second and the third comparing methods are two alternatives of our system. The first alternative employs single projection learned from PLS as the descriptor but utilizes multiple codebooks. We refer this one as "Single-Projection-Multiple-Codebook" (SPMC) method. Again, in this method all the parameters are fixed to the optimal values chosen in the parameter test experiments. Thus, the only difference between this system and the proposed one is the multiple-projection learning scheme. Hence, we could estimate the importance of using multiple projections. Another alternative is the one mentioned in section 3.4. In this alternative, we only use the patches belonging to the same group to construct a group specific codebook and apply this codebook to encode the patches which are assigned into the same group. We denote this alternative as “Hard Partition” scheme.

Conducting the experiment with the contest evaluation protocol, we obtain the performance comparison shown in Table 4. It is clear that the proposed method significantly outperforms the other three methods in comparison. We can see that LBP with BoF model performs worse than the other methods which employs data-driven feature. This validates the advantage of employing descriptor learning strategy in our work. Also, we see that the performance of SPMC is inferior to the one with multiple projections. Hence, we can confirm that introducing multiple-projections really improves the classification. Finally, it is clear that our proposed system significantly outperforms the hard partition scheme with around 6% improvement. This supports our system structure discussed in section 3.4.

5. Conclusion

In this paper, we proposed a HEp-2 cell stain pattern classification system. Unlike many traditional systems which are built on hand-coded feature, our system adopts the methodology of feature learning to learn the appropriate descriptor from the image data itself. More specifically, our system is based on the bag-of-features model and we utilize the linear projections of the raw image patch as the descriptors. We further explore the supervised and unsupervised ways to learn these projections. As our main contribution, we point out the potential drawback of learning a single projection matrix as the descriptor for the local patches. To overcome the drawback, we proposed a multi-projection-multi-codebook strategy which builds multiple descriptors for a local patch and creates multiple codebooks to generate multiple pooled vectors for an image. By concatenating the pooled vectors we obtained the final image representation. Through our detailed analysis and intensive experiments we demonstrate the advantage of our system. In conclusion, we highlight two findings of this work: (1) the descriptor learning scheme shows promising result in the task of HEp-2 cell classification. Our best performance is just slightly lower than the best performance achieved in the contest. It is quite impressive since only a single type data-driven feature is utilized. (2) the key success factor of achieving our good performance is the usage of our simple but effective multiple descriptor strategy.

[1] S. P. e. a. Rigon, A., Indirect immunofluorescence in autoimmune diseases: Assessment of digital images for diagnostic purpose., Cytometry B (Clinical Cytometry) 72 (2007) 472 – 477.

[2] E. Cordelli, P. Soda, Color to grayscale staining pattern representation in iif, in: Proceedings of the 2011 24th International Symposium on Computer-Based Medical Systems, CBMS '11, 2011.

[3] G. Percannella, P. Soda, M. Vento, Mitotic hep-2 cells recognition under class skew, in: Proceedings of the 16th international conference on Image analysis and processing - Volume Part II, ICIAP'11, 2011.

[4] P. Perner, H. Perner, B. Müller, Mining knowledge for hep-2 cell image classification, Artif. Intell. Med. 26 (1-2) (2002) 161–173.

[5] P. Soda, G. Iannello, Aggregation of classifiers for staining pattern recognition in antinuclear autoantibodies analysis, Trans. Info. Tech. Biomed. 13 (3) (2009) 322–329.

[6] F. K. Petter Strandmark, Johannes Ulën, Hep-2 staining pattern classification, in: ICPR, 2012.

[7] G. E. S. F. Ilias Theodorakopoulos, Dimitris Kastaniotis, Hep-2 cells classification via fusion of morphological and texture feature, in: BIBE, 2012.

[8] A. Wiliem, Y. Wong, C. Sanderson, P. Hobson, S. Chen, B. Lovell, Classification of human epithelial type 2 cell indirect immunofluoresence images via codebook based descriptors, in: IEEE Workshop on Applications of Computer Vision (WACV), 2013.

[9] Y.-C. Huang, T.-Y. Hsieh, C.-Y. Chang, W.-T. Cheng, Y.-C. Lin, Y.-L. Huang, Hep-2 cell images classification based on textural and statistic features using self-organizing map., in: ACIIDS (2), 2012.

[10] J. C. V. Gemert, J. mark Geusebroek, C. J. Veenman, A. W. M. Smeulders, Kernel codebooks for scene categorization, in: ECCV, 2008, pp. 696–709.

[11] J. Yang, K. Yu, Y. Gong, T. S. Huang, Linear spatial pyramid matching using sparse coding for image classification., in: CVPR, 2009.

[12] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: CVPR, Los Alamitos, CA, USA, 2010.

[13] L. Liu, L. Wang, X. Liu, In defence of soft-assignment coding, in: ICCV, 2011.

[14] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: VS-PETS, 2005.

[15] Y. Ke, R. Sukthankar, Pca-sift: A more distinctive representation for local image descriptors, in: CVPR (2), 2004, pp. 506–513.

[16] M. A. Turk, A. P. Pentland, Face recognition using eigenfaces, in: Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991, pp. 586–591.

[17] W. R. Schwartz, A. Kembhavi, D. Harwood, L. S. Davis, Human detection using partial least squares analysis, in: ICCV, 2009.

[18] M. Barker, W. Rayens, Partial least squares for discrimination, Journal of Chemometrics 17 (3) (2003) 166 – 173.

[19] S. de Jong, Simpls: an alternative approach to partial least squares regression, Chemometrics and Intelligent Laboratory Systems 18 (3) (1993) 251 – 263.

[20] B. Geladi, Paul; Kowalski, Partial least squares regression:a tutorial, Analytica Chimica Acta 185 (1986) 1 – 17.

[21] G. S. P. V. M. Foggia, P.; Percannella, Benchmarking hep-2 cells classification methods, IEEE Transactions on Medical Imaging PP (99) (2010) 1 – 1.

[22] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, Los Alamitos, CA, USA, 2006.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html (2007).

Tables

Table 1: The classification with different choices of structural parameters. SPM: spatial pyramid.

Settings			Result with different kernels		
Coding & Pooling	SPM	Projection	linear	Hellinger's	χ^2
hard-coding + sum-pooling	NO	PCA	86.1%	90.2%	90.1%
	NO	PLS	87.8%	91.2%	90.5%
	YES	PCA	86.5%	90.3%	90.5%
	YES	PLS	87.5%	91.3%	90.7%
LSC + max-pooling	NO	PCA	91.6%	91.2%	91.5%
	NO	PLS	91.4%	91.2%	91.0%
	YES	PCA	93.1%	93.0%	92.4%
	YES	PLS	92.7%	93.1%	92.5%
LSC + sum-pooling	NO	PCA	89.4%	92.7%	92.1%
	NO	PLS	91.6%	94.2%	93.9%
	YES	PCA	89.9%	93.3%	93.0%
	YES	PLS	91.9%	94.1%	94.4%

Table 2: Each row represents an image and all the cells in each image shares the same groundtruth class label. These cells are classified by the classifier learned from the rest images. For each class, the number (precentage) of cells in an image that have been assigned to it is reported. The results are splitted into two tables. This is the first one showing the results of the first 14 images. To save the space, we use the following abbreviations: cent. for centromere; homo. for homogeneous; coarse. for coarse speckled and fine. for fine speckled.

Image	Groundtruth	Class Name					
ID	class label	cent.	homo.	nucleolar	coarse.	fine.	cyto.
1	homo.	0 (0%)	60 (98%)	0 (0%)	0 (0%)	1 (2%)	0 (0%)
2	fine.	0 (0%)	3 (6%)	0 (0%)	16 (33%)	29 (60%)	0 (0%)
3	cent.	88 (99%)	0 (0%)	1 (1%)	0 (0%)	0 (0%)	0 (0%)
4	nucleolar	5 (8%)	21 (32%)	29 (44%)	1 (2%)	9 (14%)	1 (2%)
5	homo.	0 (0%)	35 (74%)	0 (0%)	0 (0%)	12 (26%)	0 (0%)
6	coarse.	51 (75%)	4 (6%)	0 (0%)	10 (15%)	3 (4%)	0 (0%)
7	cent.	49 (88%)	0 (0%)	7 (13%)	0 (0%)	0 (0%)	0 (0%)
8	nucleolar	56 (100%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)
9	fine.	0 (0%)	31 (67%)	0 (0%)	0 (0%)	15 (33%)	0 (0%)
10	coarse.	0 (0%)	0 (0%)	3 (9%)	5 (15%)	23 (70%)	2 (6%)
11	coarse.	0 (0%)	0 (0%)	0 (0%)	34 (83%)	7 (17%)	0 (0%)
12	coarse.	13 (27%)	0 (0%)	0 (0%)	35 (71%)	1 (2%)	0 (0%)
13	cent.	34 (74%)	0 (0%)	1 (2%)	11 (24%)	0 (0%)	0 (0%)
14	cent.	4 (6%)	11 (17%)	1 (2%)	39 (62%)	8 (13%)	0 (0%)

Table 3: Each row represents an image and all the cells in each image shares the same groundtruth class label. These cells are classified by the classifier learned from the rest images. For each class, the number (precentage) of cells in an image that have been assigned to it is reported. The results are splitted into two tables. This is the second one showing the results of the last 14 images. To save the space, we use the following abbreviations: cent. for centromere; homo. for homogeneous; coarse. for coarse speckled and fine. for fine speckled.

Image	Groundtruth	Class Name					
ID	class label	cent.	homo.	nucleolar	coarse.	fine.	cyto.
15	fine.	4 (6%)	13 (21%)	1 (2%)	24 (38%)	21 (33%)	0 (0%)
16	cent.	35 (92%)	0 (0%)	3 (8%)	0 (0%)	0 (0%)	0 (0%)
17	coarse.	1 (5%)	3 (16%)	0 (0%)	0 (0%)	15 (79%)	0 (0%)
18	homo.	0 (0%)	26 (62%)	2 (5%)	0 (0%)	14 (33%)	0 (0%)
19	cent.	64 (98%)	0 (0%)	1 (2%)	0 (0%)	0 (0%)	0 (0%)
20	nucleolar	31 (67%)	2 (4%)	7 (15%)	5 (11%)	1 (2%)	0 (0%)
21	homo.	2 (3%)	23 (38%)	0 (0%)	1 (2%)	35 (57%)	0 (0%)
22	homo.	0 (0%)	102 (86%)	2 (2%)	1 (1%)	14 (12%)	0 (0%)
23	fine.	0 (0%)	15 (29%)	0 (0%)	1 (2%)	35 (69%)	0 (0%)
24	nucleolar	0 (0%)	7 (10%)	66 (90%)	0 (0%)	0 (0%)	0 (0%)
25	cyto.	1 (4%)	0 (0%)	2 (8%)	13 (54%)	4 (17%)	4 (17%)
26	cyto.	1 (3%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	33 (97%)
27	cyto.	1 (3%)	0 (0%)	0 (0%)	2 (5%)	0 (0%)	35 (92%)
28	cyto.	0 (0%)	0 (0%)	0 (0%)	0 (0%)	0 (0%)	13 (100%)

Table 4: The comparison with three comparable methods. Evaluated by the cell-level contest setting.

Method	Classification Accuracy
The proposed	**66.6%**
LBP BoF	58.0%
SPMC	59.9%
Hard Partition	60.8%

Figure Captions

Figure 1. The examples of six types of HEp-2 cell patterns. (a) homogeneous (b) nucleolar (c) cytoplasmatic (d) centromere (e) fine speckled (f) coarse speckled.

Figure 2. The work flow of our classification system: (a) training stage (b) test stage

Figure 3. A toy experiment to illustrate the reason of applying multiple-PLS to learn the projection.

Figure 4. The impact of the codebook size and number of groups on the classification performance. (a) PCA (b) PLS

Figure 5. The impact of the patch size and the descriptor dimension. (a) PCA (b) PLS

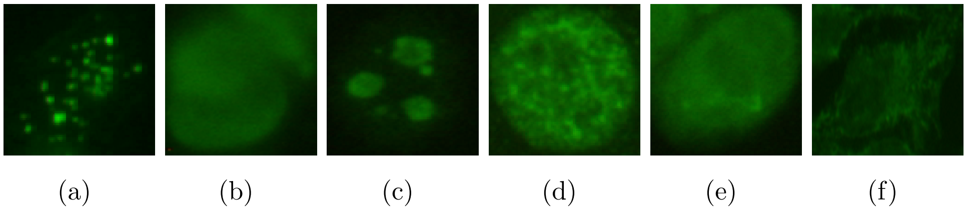
Figure 6. The confusion matrix of the proposed system for cell-level classification. Evaluated by the contest protocol. On average, we achieve 66.6% classification accuracy.

Figure 7. The confusion matrix of the proposed system for image-level classification. Evaluated by the contest protocol. On average, we achieve 78.57% classification accuracy.

Figure 8. The confusion matrix of the proposed system for cell-level classification. Evaluated by the leave-one-out protocol. On average, we achieve 58.92% classification accuracy.

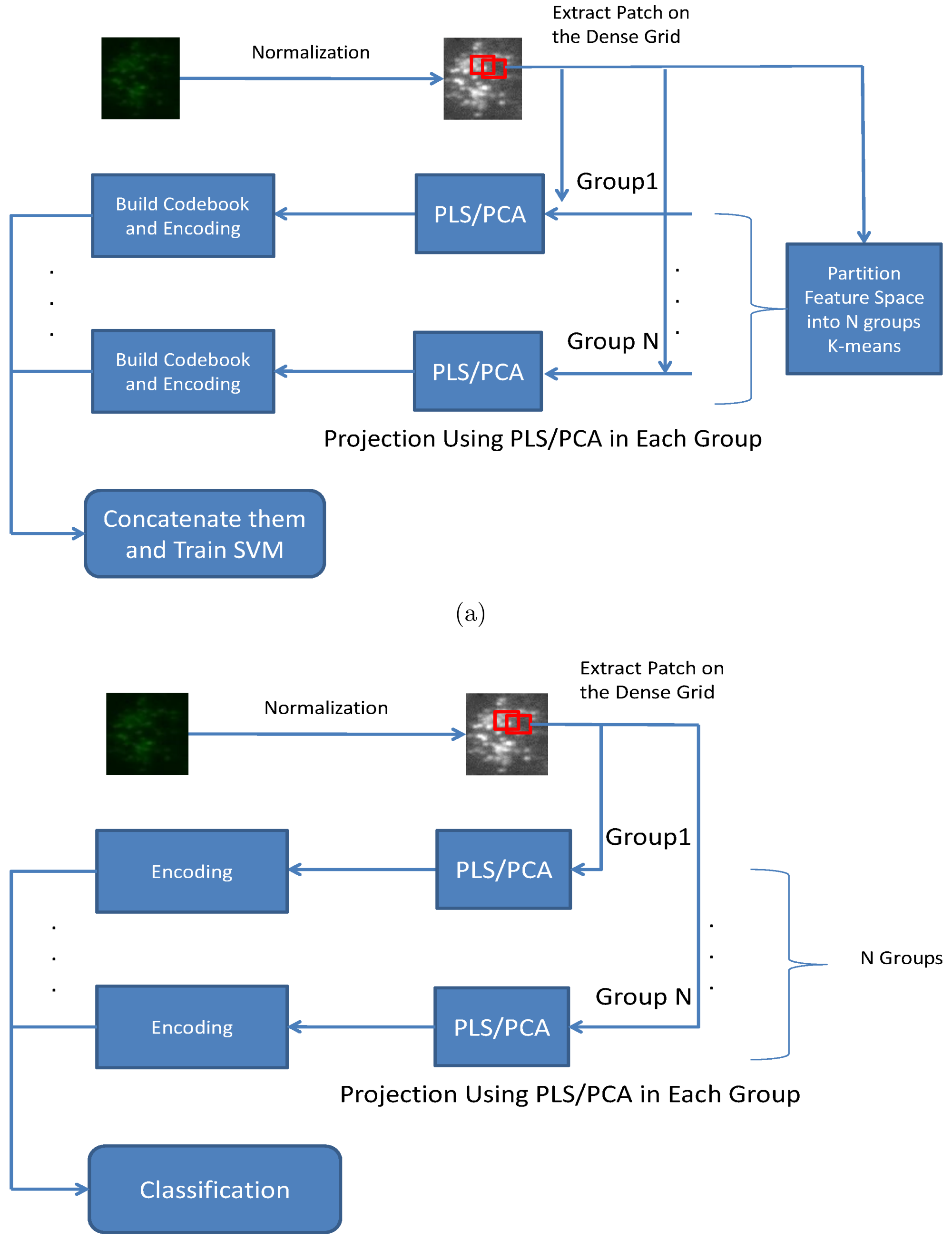
Figure 9. The confusion matrix of the proposed system for image-level classification. Evaluated by the leave-one-out protocol. On average, we achieve 64.29% classification accuracy.

Figures



(a) (b) (c) (d) (e) (f)

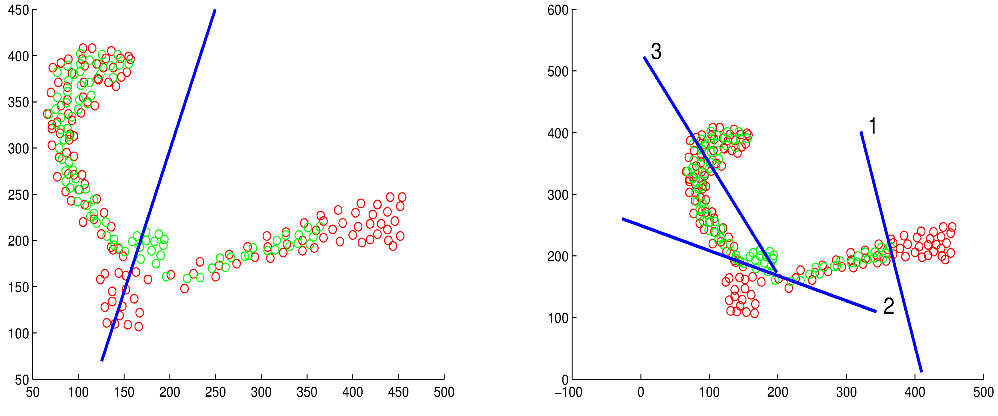
Figure 1: The examples of six types of HEp-2 cell patterns. (a) homogeneous (b) nucleolar (c) cytoplasmatic (d) centromere (e) fine speckled (f) coarse speckled.


Normalization
Extract Patch on the Dense Grid
Group1
Group N
PLS/PCA
PLS/PCA
Build Codebook and Encoding
Build Codebook and Encoding
Partition Feature Space into N groups K-means
Projection Using PLS/PCA in Each Group
Concatenate them and Train SVM
Normalization
Extract Patch on the Dense Grid
Group1
Group N
N Groups
PLS/PCA
PLS/PCA
Encoding
Encoding
Projection Using PLS/PCA in Each Group
Classification

(a)

(b)

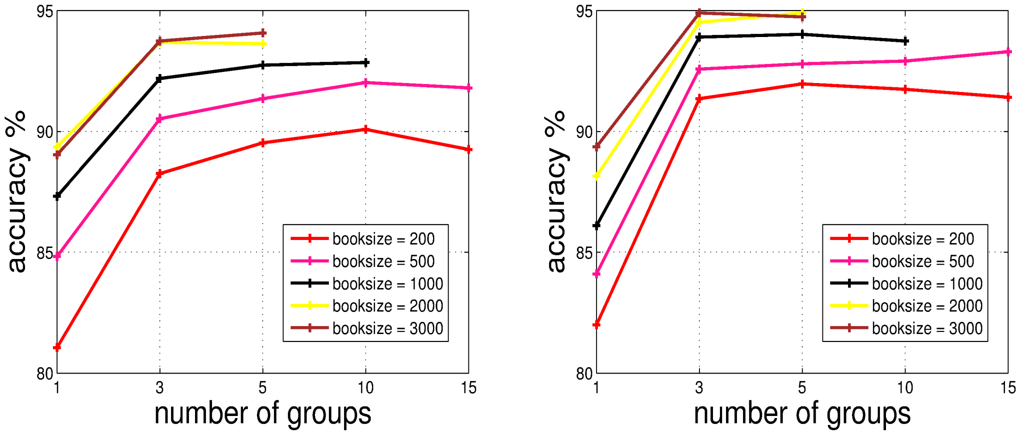
Figure 2: The work flow of our classification system: (a) training stage (b) test stage



(a)

(b)

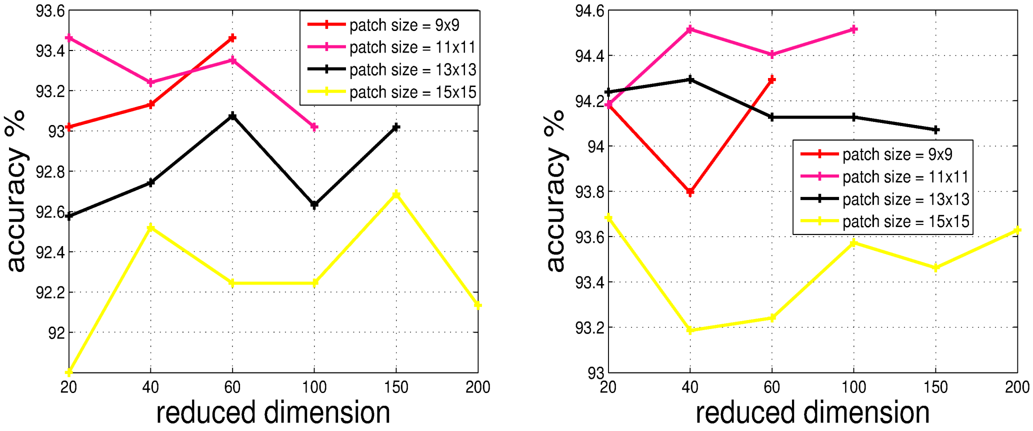
Figure 3: A toy experiment to illustrate the reason of applying multiple-PLS to learn the projection.



(a)

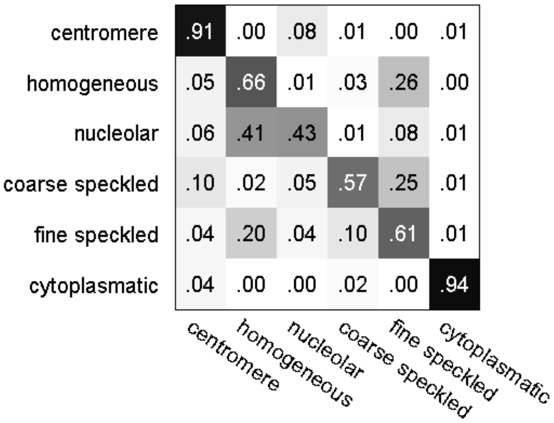
(b)

Figure 4: The impact of the codebook size and number of groups on the classification performance. (a) PCA (b) PLS



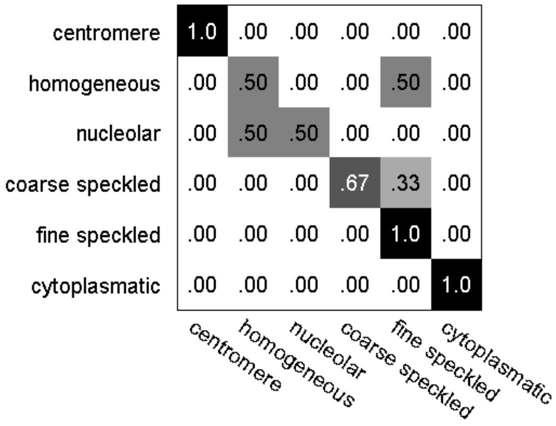
(a) (b)

Figure 5: The impact of the patch size and the descriptor dimension. (a) PCA (b) PLS



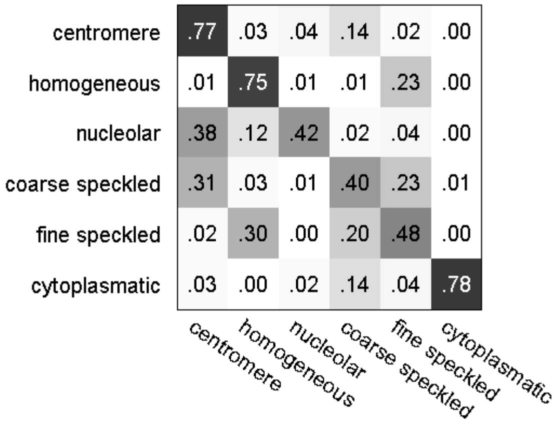
	centromere	homogeneous	nucleolar	coarse speckled	fine speckled	cytoplasmatic
centromere	.91	.00	.08	.01	.00	.01
homogeneous	.05	.66	.01	.03	.26	.00
nucleolar	.06	.41	.43	.01	.08	.01
coarse speckled	.10	.02	.05	.57	.25	.01
fine speckled	.04	.20	.04	.10	.61	.01
cytoplasmatic	.04	.00	.00	.02	.00	.94

Figure 6: The confusion matrix of the proposed system for cell-level classification. Evaluated by the contest protocol. On average, we achieve 66.6% classification accuracy.



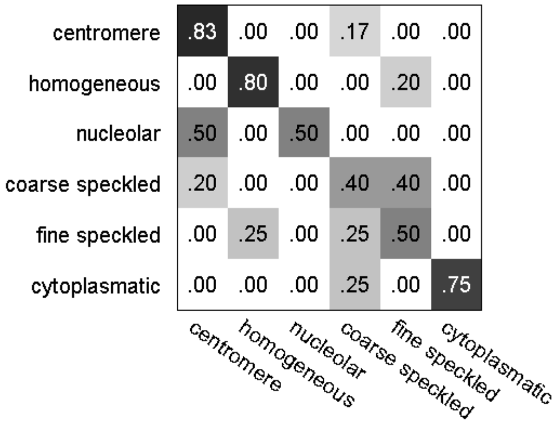
	centromere	homogeneous	nucleolar	coarse speckled	fine speckled	cytoplasmatic
centromere	1.0	.00	.00	.00	.00	.00
homogeneous	.00	.50	.00	.00	.50	.00
nucleolar	.00	.50	.50	.00	.00	.00
coarse speckled	.00	.00	.00	.67	.33	.00
fine speckled	.00	.00	.00	.00	1.0	.00
cytoplasmatic	.00	.00	.00	.00	.00	1.0

Figure 7: The confusion matrix of the proposed system for image-level classification. Evaluated by the contest protocol. On average, we achieve 78.57% classification accuracy.



	centromere	homogeneous	nucleolar	coarse speckled	fine speckled	cytoplasmatic
centromere	.77	.03	.04	.14	.02	.00
homogeneous	.01	.75	.01	.01	.23	.00
nucleolar	.38	.12	.42	.02	.04	.00
coarse speckled	.31	.03	.01	.40	.23	.01
fine speckled	.02	.30	.00	.20	.48	.00
cytoplasmatic	.03	.00	.02	.14	.04	.78

Figure 8: The confusion matrix of the proposed system for cell-level classification. Evaluated by the leave-one-out protocol. On average, we achieve 58.92% classification accuracy.



	centromere	homogeneous	nucleolar	coarse speckled	fine speckled	cytoplasmatic
centromere	.83	.00	.00	.17	.00	.00
homogeneous	.00	.80	.00	.00	.20	.00
nucleolar	.50	.00	.50	.00	.00	.00
coarse speckled	.20	.00	.00	.40	.40	.00
fine speckled	.00	.25	.00	.25	.50	.00
cytoplasmatic	.00	.00	.00	.25	.00	.75

Figure 9: The confusion matrix of the proposed system for image-level classification. Evaluated by the leave-one-out protocol. On average, we achieve 64.29% classification accuracy.