Classification of HEp-2 Cells Using Distributed Dictionary Learning

Sadaf Monajemi[1*] Shahab Ensafi[1†‡] Shijian Lu[‡] Ashraf A. Kassim[†] Chew Lim Tan[§] Saeid Sanei[¶] Sim-Heng Ong[†]
[*]NUS Graduate School for Integrative Sciences and Engineering, National University of Singapore, Singapore
[†]Electrical and Computer Engineering, Faculty of Engineering, National University of Singapore, Singapore
[‡]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR), Singapore
[§]School of Computing, National University of Singapore, Singapore
[¶]Faculty of Engineering and Physical Sciences, University of Surrey, Guildford, UK

***Abstract*—Automatic classification of human epithelial type-2 (HEp-2) cells can improve the diagnostic process of autoimmune diseases (ADs) in terms of lower cost, faster response, and better repeatability. However, most of the proposed methods for classification of HEp-2 cells suffer from several constraints including tedious parameter tuning, massive memory requirement, and high computational costs. We propose an adaptive distributed dictionary learning (ADDL) method where the dictionary learning problem is reformulated as a distributed learning task. With the help of this approach, we develop an automatic and robust method that effectively handles the complexity of the problem in terms of memory and computational cost and also obtains superior classification accuracy.**

I. Introduction

Autoimmune diseases (ADs) are among the top mortality causes according to the American Autoimmune Related Diseases Association (AARDA). Early diagnosis of ADs plays a crucial role in the treatment process of these diseases. To diagnose ADs, high resolution imaging of the affected organ through indirect immunofluorescence (IIF) is needed. IIF captures images of human epithelial type-2 (HEp-2) cells and antinuclear antibodies (ANA), which is a type of autoantibody binding to the contents of the cell nucleus, and is considered as a hallmark of ADs. In the cells containing ANAs, the antibodies bound to the nuclei have different patterns that can be captured and visualized via microscope imaging. Categorizing the patterns of the HEp-2 cell images can be used to distinguish the phase and severity of ADs. Computer aided diagnosis (CAD) systems for automatic classification of HEp-2 cells have attracted much interest for AD diagnosis. These systems can be used to reduce the cost and time of diagnosis and to provide repeatability across different physicians.

A publicly available dataset for HEp-2 cell classification contest was released at the 2013 International Conference on Image Processing, to be referred as the ICIP2013 dataset in this paper. Two different tasks have been set up for this benchmarking dataset [1]. The first task is cell level classification where each cell is classified independently without considering the neighboring cells in the specimen image. In the second task, the specimen images are classified by considering all the cells in the image. Here, it is assumed that most of the cells in a specimen image belong to one of the classes [1].

Sparse coding and dictionary learning (DL) methods have been rapidly increasing in recent years in different image processing domains. These methods are used by several researchers for solving the HEp-2 cell classification problem [2]–[5]. In these methods, a variety of features including local binary patterns (LBPs) [6], scale-invariant feature transform (SIFT) and morphological features are extracted from the cells. Then a Bag of Words (BoW) [7] model or DL scheme [8] is applied to represent the input feature vectors for a classifier.

One of the critical parameters in DL and BoW methods is the number of atoms in the dictionary. The higher number of atoms compared to the dimension of the feature vectors results in an *over complete* dictionary which is biologically inspired and can provide suitable sparse codes. However, learning these types of dictionaries is computationally expensive and requires a large memory. Additionally, selection of parameters (including the size and number of patches, smoothing parameters, and the number of histogram bins) can affect the performance of the method. Parameter tuning is complex and tedious, leading to massive memory and computational requirements especially when the number of training images increases. The complexities of the above methods make them impractical for realizing real time systems to be used by physicians/clinicians.

In this paper, we propose an adaptive distributed dictionary learning (ADDL) method which tackles the HEp-2 cell classification problem in a computationally efficient and less memory intensive way compared to the other methods. To the best of our knowledge, this is the first time that an adaptive distributed dictionary learning method has been successfully implemented for image classification. In this method, the dictionary matrix and the coding vector are partitioned into N blocks where each block is associated with a sub-dictionary and a sub-vector. Considering these blocks, we form a connected network of N nodes where each node updates its own sub-dictionary. We reformulate the dictionary learning problem as a distributed learning task over the network and

[1]The first two authors contributed equally to this work.

978-0-9928-6265-7/16/$31.00 ©2016 IEEE

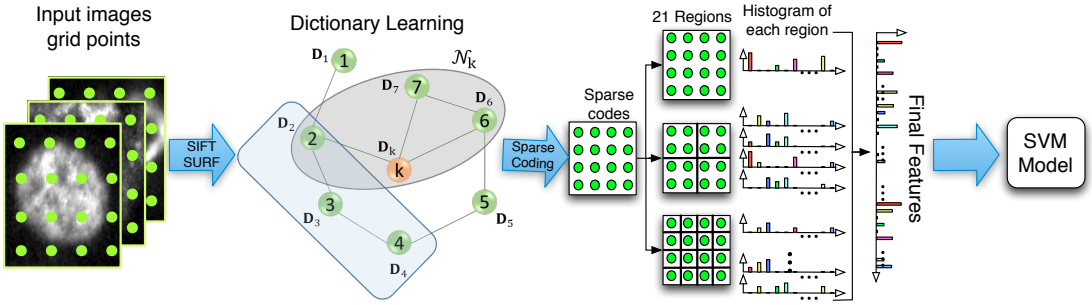

Fig. 1. The proposed ADDL framework.

use the diffusion adaptation strategy to solve this distributed problem [9]. Moreover, we propose to combine the information of neighboring nodes in an adaptive way to obtain superior performance.

II. Overview of the ADDL Method

In this paper we extract SURF (speeded-up robust features) and SIFT image features to use them as inputs for the distributed dictionary learning problem. The learned dictionary is then used for HEp-2 cell classification where the sparse codes of image patches are combined with spatial pyramid matching (SPM) [10]. In this method, each input image is divided into 1, 4 and 16 regions within three pyramid layers and max-pooling is applied on the sparse codes of each region to obtain the final feature vector. Then, a SVM is learned to classify the cell images. Fig. 1 represents the proposed ADDL method.

A. Dictionary Learning

By extracting the features of each image patch the input feature vector $\boldsymbol{F}_t$ for the DL algorithm is calculated. The DL problem can then be formulated as:

$$\min_{\boldsymbol{z}_t,\boldsymbol{D}} \left(\|\boldsymbol{F}_t - \boldsymbol{D}\boldsymbol{z}_t\|^2 + \lambda\|\boldsymbol{z}_t\|_1 + \frac{\beta}{2}\|\boldsymbol{z}_t\|_2^2\right) \tag{1}$$

where $\boldsymbol{F}_t$ is the $M\times 1$ input feature vector at time t, $\boldsymbol{D}$ is an $M\times K$ dictionary matrix, $\boldsymbol{z}_t$ is the $K\times 1$ sparse code vector, and λ and β are the adjustable penalty (regularization) terms. To solve the optimization problem in (1), we introduce the distributed learning method.

B. Distributed Dictionary Learning

To learn the dictionary in a distributed manner we adopt the recently proposed approach presented in [11]. In this method, the dictionary matrix $\boldsymbol{D}$ and the coding vector $\boldsymbol{z}$ are partitioned to block forms:

$$\boldsymbol{D} = [\boldsymbol{D}_1 \dots \boldsymbol{D}_N], \quad \boldsymbol{z} = \text{col}\{\boldsymbol{z}_1, \dots, \boldsymbol{z}_N\} \tag{2}$$

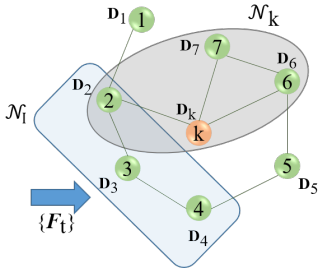

Fig. 2. Sample of a connected network where each agent k is responsible for learning a sub-dictionary $\boldsymbol{D}_k$.

where $\boldsymbol{D}_k$ is a sub-dictionary matrix of size $M\times N_k$ and $\boldsymbol{z}_k$ is a sub-vector of size $N_k\times 1$. Moreover, the summation over the sizes of sub-dictionaries is equal to the total size of dictionary:

$$N_1 + \dots + N_N = K \tag{3}$$

Now we form a connected network of N agents where each agent k in the network is responsible to update its own sub-dictionary $\boldsymbol{D}_k$ that is distributed over the network. As shown in Fig. 2, each agent in the network has a number of neighboring agents that it can interact with. Moreover, the input features $\boldsymbol{F}_t$ can be only presented to a subset of agents represented by $\mathcal{N}_\mathcal{I}$. Our experiments show that providing the input data only to a subset of agents is computationally efficient while having comparable performance with other methods. Considering (2) in the DL problem we can reformulate (1) as:

$$\min_{\boldsymbol{z}_t,\boldsymbol{D}} \left(\|\boldsymbol{F}_t - \sum_{k=1}^{N}\boldsymbol{D}_k\boldsymbol{z}_{k,t}\|^2 + \sum_{k=1}^{N}(\lambda\|\boldsymbol{z}_{k,t}\|_1 + \frac{\beta}{2}\|\boldsymbol{z}_{k,t}\|_2^2)\right) \tag{4}$$

To solve the optimization problem of (4) in a distributed manner, the cost function should have a *"sum-of-costs"* form. Specifically, in order to apply distributed methods to tackle the problem at hand, the global cost function of the optimization problem, $J^{glob}(\boldsymbol{\omega})$, should be the aggregation of individual cost functions of the agents $J_k(\boldsymbol{\omega})$:

$$J^{glob}(\boldsymbol{\omega}) = \sum_{k=1}^{N} J_k(\boldsymbol{\omega}) \tag{5}$$

It has been shown in [11] that the problem in (4) does not follow the form in (5) as it is *"cost-of-sums"* and not

"sum-of-costs". Therefore, it is not feasible to use distributed techniques for solving the problem in (4) directly. However, the dual problem of (4) has a distributed form similar to (5) and the optimal primal variables $\{\boldsymbol{D}_k\}$ and $\boldsymbol{z}_t$ can be recovered from the solution of the dual problem (see (14) and (15)). According to [11], the dual problem can be formulated as:

$$\min_{\boldsymbol{\nu}}(-g(\boldsymbol{\nu},\boldsymbol{F}_t)) = (\|\boldsymbol{\nu}\|_2^2 - \boldsymbol{\nu}^T\boldsymbol{F}_t + \sum_{k=1}^{N}\mathcal{S}_{\frac{\lambda}{\beta}}(\frac{\boldsymbol{D}_k^T\boldsymbol{\nu}}{\beta})) \quad (6)$$

where $\boldsymbol{\nu}$ is the auxiliary vector variable of size $M\times 1$ in the dual problem, and $\mathcal{S}_{\frac{\lambda}{\beta}}(x)$ is a function defined as:

$$\mathcal{S}_{\frac{\lambda}{\beta}}(x) \triangleq -\frac{\beta}{2}\cdot\|\mathcal{T}_{\frac{\lambda}{\beta}}(x)\|_2^2 - \lambda\cdot\|\mathcal{T}_{\frac{\lambda}{\beta}}(x)\|_1^2 + \beta\cdot x^T\mathcal{T}_{\frac{\lambda}{\beta}}(x) \quad (7)$$

Here $\mathcal{T}_\gamma(x)$ is the entry-wise soft-thresholding operator on vector x that can be formulated for the n_{th} element as:

$$[\mathcal{T}_\gamma(x)]_n \triangleq (|[x]_n| - \gamma)_+\text{sgn}([x]_n) \quad (8)$$

where $(x)_+ = \max(x,0)$ and $\text{sgn}(x)$ represents the signum function. We can consider the dual function in (6) as the global cost function. Therefore, the individual cost function of each node k can be defined as [11]:

$$J_k(\boldsymbol{\nu};\boldsymbol{F}_t) \triangleq \begin{cases} -\frac{\boldsymbol{\nu}^T\boldsymbol{F}_t}{|\mathcal{N}_\mathcal{I}|} + \frac{1}{N}\|\boldsymbol{\nu}\|_2^2 + \mathcal{S}_{\frac{\lambda}{\beta}}(\frac{\boldsymbol{D}_k^T\boldsymbol{\nu}}{\beta}),\ k\in\mathcal{N}_I \\ \frac{1}{N}\|\boldsymbol{\nu}\|_2^2 + \mathcal{S}_{\frac{\lambda}{\beta}}(\frac{\boldsymbol{D}_k^T\boldsymbol{\nu}}{\beta}),\ k\notin\mathcal{N}_I \end{cases} \quad (9)$$

where $|\mathcal{N}_\mathcal{I}|$ is the cardinality of $\mathcal{N}_\mathcal{I}$.

It should be noted that the summation over the individual cost functions $J_k(\boldsymbol{\nu};\boldsymbol{F}_t)$ is equal to the cost function in (6) and the dual problem for estimating the optimal solution $\boldsymbol{\nu}^o$ can be rewritten as:

$$\boldsymbol{\nu}^o = \min_{\boldsymbol{\nu}}\sum_{k=1}^{N} J_k(\boldsymbol{\nu};\boldsymbol{F}_t) \quad (10)$$

Therefore, according to (5) the dual problem can be solved using distributed learning strategies and the optimal primal variables $\{\boldsymbol{D}_k\}$ and $\boldsymbol{z}_t$ can be recovered afterwards (see (14) and (15)).

Here we use diffusion adaptation strategy as a distributed learning method [9], [12]. It has been shown that the diffusion strategy gives superior performance and stability compared to the other methods while being robust, scalable, and capable of real time adaptation and learning [9]. The details of the diffusion strategy adapted to solve the distributed optimization problem in (10) is explained in the next sub-section.

C. Diffusion Adaptation Method

In the diffusion adaptation strategy, there is a network of N nodes where each node k is connected to its neighboring nodes represented by $\mathcal{N}_k$ shown in Fig. 2. Each node can share information with and receive information from its neighbors. Each node also has an individual cost function to minimize and the global cost function of the network is the aggregation of all these individual costs similar to (5). The diffusion adaptation method consists of two steps: the adaptation step and the combination step. In the adaptation step, each node k updates its own estimate for the optimization problem via a gradient descent step. This estimate is considered as an intermediate estimate $\boldsymbol{\psi}_{k,i}$, which is further updated in the combination step. During the combination step, the neighboring nodes share their intermediate estimates. Then, each node k updates its own final estimate, $\boldsymbol{\nu}_{k,i}$, by combining the intermediate estimates received from the neighbors in the i_{th} time instant [9]. Thus, the diffusion adaptation strategy can be reformulated as:

$$\boldsymbol{\psi}_{k,i} = \boldsymbol{\nu}_{k,i-1} - \mu\nabla_{\boldsymbol{\nu}}J_k(\boldsymbol{\nu}_{k,i-1};\boldsymbol{F}_t) \quad (11)$$

$$\boldsymbol{\nu}_{k,i} = \sum_{\ell\in\mathcal{N}_k} a_{\ell k}(i)\boldsymbol{\psi}_{\ell,i} \quad (12)$$

where $\boldsymbol{\nu}_{k,i}$ is the estimate of node k of the optimal solution $\boldsymbol{\nu}_t^o$ at iteration i, $\boldsymbol{\psi}_{k,i}$ is the intermediate estimate, and $\mu>0$ is the updating step-size selected to be sufficiently small. The weights $a_{\ell k}(i)$ in (12) are called *combination weights* and each $a_{\ell k}(i)$ is the weight that node k assigns to the information received from node ℓ at time instant i. The combination weights $a_{\ell k}(i)$ must satisfy:

$$\sum_{\ell\in\mathcal{N}_k} a_{\ell k}(i) = 1,\ a_{\ell k}(i) = 0 \text{ if } \ell\notin\mathcal{N}_k \quad (13)$$

It should be noted that there are several ways to design the combination weights. It has been shown in the literature that selection of these weights can have a significant impact on the algorithm performance [13], [14]. In section II-D we discuss the role of these weights and introduce an adaptive method to learn the weights over time.

After the optimal dual variable $\boldsymbol{\nu}_t^o$ is estimated by (11) and (12), the optimal primal variables of the DL problem, including the sparse codes $\boldsymbol{z}_t^o$ and the sub-dictionaries $\boldsymbol{D}_{k,t}$, can be obtained by [11]:

$$\boldsymbol{z}_{k,t}^o = \arg\max_{\boldsymbol{z}_k}[(\boldsymbol{D}_k^T\boldsymbol{\nu}_t^o)^T\boldsymbol{z}_k - (\lambda\|\boldsymbol{z}_k\|_1 + \frac{\beta}{2}\|\boldsymbol{z}_k\|_2^2)] \quad (14)$$

$$\boldsymbol{D}_{k,t} = \Pi_{\mathcal{D}_k}(\boldsymbol{D}_{k,t-1} + \mu\cdot\boldsymbol{\nu}_t^o\,\boldsymbol{z}_{k,t}^o) \quad (15)$$

where $\Pi_{\mathcal{D}_k}[\cdot]$ is the projection operator onto the constraint set $\mathcal{D}_k$. In the next section, we continue to formulate our ADDL method by proposing an adaptive approach to design the combination weights in (12).

D. Selection of the Combination Weights

Selection of the combination weights in (12) can affect the performance of the network in solving the optimization problem. Here, we propose to use an adaptive approach for selecting the weights to address the DL task. In the previously proposed distributed dictionary learning methods, the combination weights are selected in a static manner where the nodes allocate the same weights to their neighbors without considering the reliability of the received information [11]. For

instance, where uniform weights are selected, the combination step (12) is simply an averaging over all the estimates:

$$a_{\ell k} = \frac{1}{|\mathcal{N}_k|} \text{ if } \ell \in \mathcal{N}_k \text{ (Uniform combination weights)} \quad (16)$$

By designing the combination weights in a uniform manner the nodes assign the same weight to all their neighbors without considering the reliability of the information they receive from them. It has been shown that it is important to design the weights such that the nodes can learn about the reliability of the information received from their neighbors over time [13], [14]. Therefore, the combination weights must be designed in a manner that helps the nodes to ignore misleading information and cooperate only with neighbors that share the same objective. In order to do so, we follow the same approach proposed in [15] which minimizes the instantaneous mean-square deviation (MSD) of the network defined as:

$$\text{MSD}(i) \triangleq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\|\widetilde{\boldsymbol{\nu}}_k(i)\|^2, \quad (17)$$

where $\widetilde{\boldsymbol{\nu}}_k(i) \triangleq \boldsymbol{\nu}_t^o - \boldsymbol{\nu}_k(i)$ is the error vector at node k at iteration i. Then, the combination coefficients $a_{\ell k}(i)$ can be obtained by solving the optimization problem:

$$\min_{\{a_{\ell k}(i)\}} \text{MSD}(i) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\|\widetilde{\boldsymbol{\nu}}_k(i)\|^2 \quad (18)$$

It is shown in [15] that the optimal solution can be approximated by:

$$a_{\ell k}(i) \approx \begin{cases} \frac{\|\boldsymbol{\nu}_k(i-1) - \boldsymbol{\psi}_\ell(i)\|^{-2}}{\sum_{n \in \mathcal{N}_k} \|\boldsymbol{\nu}_k(i-1) - \boldsymbol{\psi}_n(i)\|^{-2}}, & \ell \in \mathcal{N}_k \\ 0, & \text{otherwise} \end{cases} \quad (19)$$

One important observation from (19) is that the combination weights are estimated such that the nodes allocate higher weights to neighbors with similar objectives while learning to ignore misleading information. As a result, using this combination method enables the nodes to continuously learn about the objective of their neighbors so that they can distinguish between the useful and misleading information. Estimating the combination weights in this manner helps the agents to benefit from the cooperation with their neighbors. Moreover, due to promoting similarity among the nodes with similar objectives, this method results in a more discriminative dictionary which leads to better classification results (section III).

III. Experiments and Results

A. Dataset and Evaluation Methods

In this paper, a publicly available dataset namely ICIP2013 [16] is used for evaluation. This dataset contains many *cells* within each *specimen* image. This dataset contains 419 sera of patients. Approximately 100-200 cell images were extracted from each patient serum. In total, there are 68,429 cell images extracted including 13,596 cell images for training (publicly available), and 54,833 for testing (private for the organizers). Each annotated cell image contains the information about the cell pattern, intensity level (*positive* or *intermediate*), mask and the image ID (the category of the cell). Note that in the *Cell Level* we are facing a 6-class classification problem, where the classes are Centromere, Golgi, Homogeneous, Nucleolar, Nucleolar membrane (NuMem) and Speckled. But at the *Specimen Level*, we have seven classes where the Mitosis Spindle class is added.

Due to the lack of a test set, two evaluation methods are used in the literature. The first is HSM method reported in [17], where 600 cells (300 for Golgi class) from each class are used for training and the rest for test. The other method is LOSO as performed for the ICPR2012 dataset.

B. Classification Results

Table I shows the experimental results for the ICIP2013 dataset. The ADDL results are reported in two forms of adaptive and uniform weights according to (19) and (16) respectively. The proposed ADDL method with adaptive weights outperforms other methods significantly. By using the HSM evaluation method the ADDL with adaptive weights obtained 93.7% accuracy which is 2% higher than ADDL with uniform weights and other dictionary learning (DL) methods. Additionally, it outperforms non-DL methods by 5%.

With the LOSO evaluation method, the ADDL with adaptive weights obtained 81.6% accuracy on average which is 4% higher than that achieved by the ADDL with uniform weights. This performance is better than the other DL based methods and 4% higher than other classification methods. Additionally, the performance in the specimen level also outperforms other methods offering 90.4% accuracy.

C. Computational cost

The dictionary learning is a computationally expensive and time consuming task. Table II shows different dictionary learning procedures with their computation time. These measurements are accomplished in a machine with Intel Core i7 CPU and 16 GB RAM with 64-bit operating system. As shown in Table II, the proposed method takes 286.21 seconds to calculate the dictionary when the information is given to a single node to process. This is 47 seconds lower than the time needed for passing information to all nodes to process but significantly better than the results of the other dictionary methods. For example, ADDL is 9 and 20 times better than the [3] and SNPB [18] methods, respectively. Therefore, it can be seen clearly that the proposed method can enhance the performance of dictionary learning task significantly in terms of both computational cost and classification accuracy.

IV. Conclusion

We proposed an adaptive distributed dictionary learning method that benefits from lower computational cost with lower

TABLE I
THE CLASSIFICATION ACCURACIES BY USING HSM AND LOSO EVALUATION METHODS.

ICIP2013 (%)			ADDL		Other DL Methods				Others	
			Adaptive	Uniform	Ensafi [3]	SNPB [18]	Gragnaniello [19]	manivannan [20]	HSM [17]	Larsen [21]
HSM	*Cell Level*	Positive	**97.9**	95.4	95.8	96.8	-	-	95.5	-
		Intermediate	**89.4**	87.6	87.9	88.8	-	-	80.9	-
		Average	**93.7**	91.5	91.9	92.8	-	-	88.2	-
LOSO	*Cell Level*	Positive	**88.5**	84.2	83.4	83.8	-	-	-	-
		Intermediate	**74.7**	71.4	71.2	72	-	-	-	-
		Average	**81.6**	77.8	77.3	77.9	81.1	80.3	-	78.7
	Specimen Level		**90.4**	86.7	88	89.2	86.7	89.9	-	-

TABLE II
DICTIONARY LEARNING METHODS' COMPUTATIONAL TIMES (SEC).

ADDL		Ensafi [3]	SNPB [18]
All Nodes	Single Node		
333.73	**286.21**	2751.91	5742.64

number of tuning parameters which is an important advantage in solving classification problems. The ADDL method is applied to HEp-2 cell images and obtained state-of-the-art results. The proposed method enhances the accuracy of the cell classification problem compared to other methods while reducing the computational time significantly. Moreover, learning the combination weights adaptively is an important contribution for the proposed method which makes it capable of adjusting itself for different datasets.

REFERENCES

[1] P. Foggia, G. Percannella, P. Soda, and M. Vento, "Benchmarking HEp-2 cells classification methods," *IEEE Transactions on Medical Imaging*, vol. 32, no. 10, pp. 1878–1889, 2013.

[2] S. Ensafi, S. Lu, A. A. Kassim, and C. L. Tan, "Automatic cad system for HEp-2 cell image classification," in *International Conference on Pattern Recognition (ICPR), 2014*, pp. 3321–3326.

[3] ——, "A bag of words based approach for classification of HEp-2 cell images," in *Pattern Recognition Techniques for Indirect Immunofluorescence Images (I3A), 2014*, pp. 29–32.

[4] L. Shen, J. Lin, S. Wu, and S. Yu, "HEp-2 image classification using intensity order pooling based features and bag of words," *Pattern Recognition*, vol. 47, no. 7, pp. 2419–2427, 2014.

[5] A. Wiliem, C. Sanderson, Y. Wong, P. Hobson, R. F. Minchin, and B. C. Lovell, "Automatic classification of human epithelial type 2 cell indirect immunofluorescence images using cell pyramid matching," *Pattern Recognition*, vol. 47, no. 7, pp. 2315–2324, 2014.

[6] R. Nosaka and K. Fukui, "HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns," *Pattern Recognition*, vol. 47, no. 7, pp. 2428–2436, 2014.

[7] A. Sriram, S. Ensafi, S. F. Roohi, and A. A. Kassim, "Classification of human epithelial type-2 cells using hierarchical segregation," in *International Conference on Control Automation Robotics Vision (ICARCV), 2014*, pp. 323–328.

[8] S. Di Cataldo, A. Bottino, I. U. Islam, T. F. Vieira, and E. Ficarra, "Subclass discriminant analysis of morphological and textural features for HEp-2 staining pattern classification," *Pattern Recognition*, vol. 47, no. 7, pp. 2389–2399, 2014.

[9] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks," *IEEE Transactions on Signal Processing*, vol. 60, no. 12, pp. 6217–6234, 2012.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006*, vol. 2, pp. 2169–2178.

[11] J. Chen, Z. J. Towfic, and A. H. Sayed, "Dictionary learning over distributed models," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, pp. 1001–1016, 2015.

[12] S. Monajemi, S. Sanei, and S.-H. Ong, "Advances in bacteria motility modelling via diffusion adaptation," in *European Signal Processing Conference (EUSIPCO), 2014*, pp. 2335–2339.

[13] S. Monajemi, S. Sanei, S.-H. Ong, and A. H. Sayed, "Adaptive regularized diffusion adaptation over networks," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP), 2015*, pp. 1–5.

[14] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2733–2748, 2015.

[15] X. Zhao and A. H. Sayed, "Clustering via diffusion adaptation over networks," in *International Workshop on Cognitive Information Processing (CIP), 2012*, pp. 1–6.

[16] P. Foggia, G. Percannella, A. Saggese, and M. Vento, "Pattern recognition in stained HEp-2 cells: Where are we now?" *Pattern Recognition*, vol. 47, no. 7, pp. 2305–2314, 2014.

[17] X.-H. Han, J. Wang, G. Xu, and Y.-W. Chen, "High-order statistics of microtexton for HEp-2 staining pattern classification," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 8, pp. 2223–2234, 2014.

[18] S. Ensafi, S. Lu, A. A. Kassim, and C. L. Tan, "Sparse non-parametric bayesian model for HEp-2 cell image classification," in *IEEE International Symposium on Biomedical Imaging (ISBI), 2015*, pp. 679–682.

[19] D. Gragnaniello, C. Sansone, and L. Verdoliva, "Biologically-inspired dense local descriptor for indirect immunofluorescence image classification," in *Pattern Recognition Techniques for Indirect Immunofluorescence Images (I3A), 2014*, pp. 1–5.

[20] S. Manivannan, W. Li, S. Akbar, R. Wang, J. Zhang, and S. McKenna, "HEp-2 cell classification using multi-resolution local patterns and ensemble SVMs," in *Pattern Recognition Techniques for Indirect Immunofluorescence Images (I3A), 2014*, pp. 37–40.

[21] A. Larsen, J. Vestergaard, and R. Larsen, "HEp-2 cell classification using shape index histograms with donut-shaped spatial pooling," *IEEE Transactions on Medical Imaging*, vol. 33, no. 7, pp. 1573–1580, 2014.