HEp-2 Cell Image Classification with Convolutional Neural Networks

Zhimin Gao, Jianjia Zhang, Luping Zhou, Lei Wang
School of Computer Science and Software Engineering
Uinversity of Wollongong
Wollongong, Australia
Email: {zg126, jz163}@uowmail.edu.au, {lupingz, leiw}@uow.edu.au

***Abstract*—The diagnosis of many autoimmune diseases can be greatly facilitated by automatic staining patterns classification of Human Epithelial-2 (HEp-2) cells within indirect immunofluorescence (IIF) images. In this paper, we propose a framework to classify the HEp-2 cells by utilizing the deep convolutional neural networks (CNNs). With carefully designed network architecture and optimized parameters, our networks extract features from raw pixels of cell images in a hierarchical manner and perform classification jointly, avoiding using hand-crafted features to represent a HEp-2 cell image. We evaluate our method on the training dataset of HEp-2 cells classification competition held by ICPR 2014. Our system achieves mean class accuracy of 96.7% on the held-out test set and it also obtains competitive performance on the ICPR 2012 cell dataset.**

***Keywords*-staining patterns; classification; indirect immunofluorescence; deep CNNs; raw pixels;**

I. Introduction

Indirect immunofluorescence (IIF) on Human Epithelial-2 (HEp-2) cells is a standard methodology to diagnose autoimmune diseases in current diagnostic procedure due to its effectiveness [1]. The cell staining patterns in IIF images are indicators of the presence of antinuclear antibodies in the patient serum, and specific autoimmune disease is related to the type of the staining patterns. Manual identification of these patterns leads to some crucial limitations: the subjectivity of the results and the inter-laboratories variability which limits the repeatability of the reading results [2]. Furthermore, it becomes a time-consuming process if a large number of cell images are available. Thus, automatic classification of the staining patterns by utilizing pattern recognition techniques has been increasingly demanded.

In this paper, we propose an automatic feature extraction and classification framework for HEp-2 cell staining patterns based on deep convolutional neural networks (CNNs) [5]. The networks extract features from raw pixels of cell images in a hierarchical manner and avoid extracting the classical hand-crafted features. The internal representations for each kind of cell staining patterns are obtained after training the proposed multi-layer networks with provided cell images and class label information via back-propagation algorithm [6]. The classification layer is jointly learnt under this single architecture and predicts the probabilities of a cell image belonging to each of the six classes.

II. Related Work

A variety of methods have been proposed for staining patterns classification of HEp-2 cells, especially during the first and second HEp-2 cells classification competitions [2], [3]. Generally, the majority of these methods perform feature extraction and classification separately. For feature extraction, a variety of hand-crafted features are adopted based on expert knowledge, such as local binary pattern (LBP) or its extension [8], and scale-invariant feature transform (SIFT). They may not generalize well to other laboratory environments [4]. A combination of these features with morphological features or statistic features is often adopted. A popular framework for cells classification is the bag-of-words (BoWs) model as indicated in [3]. Methods based on this model [4], [9], [10] usually achieve promising performance. For the classification stage, various classifiers are also adopted, such as Nearest Neighbour classifier, AdaBoost, SVM and Multiple Kernel SVM [4].

Our framework is different from these methods. The discriminative features are automatically learnt from the cell data, instead of being devised specially. Moreover, the properties of CNNs enable the learnt features to be invariant to small input variations, and the spatial information is preserved through the hierarchical learning stages. Malon et al. [2] adopted a CNN to classify cell images at the first competition and obtained the 6th place. Our classification framework is different from theirs, in terms of both image preprocessing method and network architecture. Liu et al. [11] proposed a method to learn the descriptors also from the raw data, but the final image representation is obtained by adopting a BoWs model.

CNNs belong to one class of models inspired by the multi-stage processes of the visual cortex [12]. The typical architecture of CNNs [5] is composed of convolutional layers interlaced with subsampling layers to perform feature extraction and pooling respectively, followed by fully-connected layers to conduct classification. The recent outstanding classification performance achieved by deep CNNs on large scale natural image benchmarks has revived the interest in CNNs [7]. Many hand-crafted features and corresponding classification pipelines more or less share the basic building blocks with CNNs. However, for these methods, the coding or spatial integrating methods have to be designed carefully to preserve useful information. The excellent performance of CNNs motivates us to extend it to the cell image classification problem, despite

978-1-4799-4252-7/14 $31.00 © 2014 IEEE
DOI 10.1109/I3A.2014.15

the challenge that it is difficult to attain satisfying CNNs with a limited number of images.

III. Proposed Framework

The framework consists of three stages: image preprocessing, network training, and feature extraction and classification, as shown in Fig. 1. Designing and training a CNN requires choosing a variety of parameters, such as the number and types of layers, the number and size of kernels of convolutional layers, and so on. The correct selection of them is crucial to obtain a network that generalizes well [13].

A. Network Architecture

The architecture of our CNN contains eight layers with weights and biases: the first six layers are convolutional layers interlaced with max-pooling layers, and the remaining two are fully-connected layers. The output of the last fully-connected layer is passed to a 6-way softmax regression, which produces the probabilities for an input cell image over the six classes. Our network optimizes the cross-entropy between the output probabilities and the class labels.

The first layer of the CNN convolves the input image with six different kernels of size 7×7 with a stride of one pixel. A bias is added and a hyperbolic tangent function $f(x) = 1.7159 \tanh(\frac{2}{3}x)$ is applied to each unit. They are also applied to the other two convolutional layers and the first fully-connected layer. The second layer takes the output of the first layer as input, and applies max-pooling over non-overlapping regions of size 2×2 for each feature map with a stride of one. The following convolutional layers and pooling layers also adopt the same stride. The third layer has 16 kernels of size 4×4, and each output feature map is the result of a sum of convolving all the previous layer's feature maps with corresponding kernels. The forth layer applies max-pooling over non-overlapping pooling regions of size 3×3. The fifth layer includes 32 kernels of size 3×3 and operates the same convolution as in the third layer. The sixth layer employs 3×3 non-overlapping max-pooling to the output maps of the fifth layer. The resulting 32 feature maps of size 3×3 are cascaded and fed into the fully-connected layer (the seventh layer) including 150 neurons. The last layer contains six neurons corresponding to each of the six classes, and outputs the probabilities over the six classes for each input image after the softmax regression.

B. Image Preprocessing

Each cell image is normalized to enhance the contrast by first subtracting the minimum intensity value of the image. Each pixel's intensity is then divided by the difference between the maximum intensity and the minimum intensity of the image. The normalized intensity values that are larger than one are set to one. Furthermore, each image is resized to a fixed size of 78×78 to guarantee a uniform scale for all images.

In order to obtain rotation invariance and increase the number of training samples, data augmentation is performed by rotating each image with an angle step of α degree. The training set is enlarged by a factor of $n = \frac{360}{\alpha}$ as a result.

C. Network Training

The network is trained on the preprocessed training images. Each pixel of an input image is standardized to have zero mean and unit variance. The weights of the convolutional layers and fully-connected layers are initialized from the uniform distribution and the biases are initialized to zero. All these trainable parameters are updated periodically via stochastic gradient descent [5] on a mini-batch of training images after evaluating the cost function. We train our model starting with a learning rate of 0.01, in conjunction with a momentum of 0.9 to speed up the learning procedure, and weight decay of 0.0005 for regularization term. The learning rate is divided by 10 when the training error rate becomes stabilized. We stop training the network after the error rate of the validation set (which is held out from the given training images) plateaus at a specific epoch.

D. Feature Extraction and Classification

During test, the test image is first applied the same preprocessing and rotation as described above. For each test image, there will be n variants in total after rotation. Each of them is forward-propagated through the network. The spatial resolution of each feature map decreases as the features are extracted hierarchically from one layer to next, and the spatial information of input is preserved by the feature maps because of the spatial convolution and pooling operations. The features obtained are invariant to small translation, distortion or shift of input images, because the kernel weights of the convolutional layers are fixed for different regions of input maps and the max-pooling layers are robust to small variations. A 150-dimensional feature vector is obtained at the first fully-connected layer. The probabilities of the cell image for each class are given by the output layer. To further improve the robustness, we select four CNNs of different epochs to jointly classify the test image in our submission. The classification label is decided as the class having the maximum output probability by averaging over the $4n$ probabilities.

IV. Experiments

A. HEp-2 Cell Dataset

The dataset of ICPR 2014 competition contains 13,596 training cell images, and test cell images are not published. All the images are extracted from 83 specimen images, and they have been manually segmented and annotated. Each image belongs to one of the six types of staining patterns: centromere (CE), golgi (GO), homogeneous (HO), nucleolar (NUC), nuclear membrane (NUM) and speckled (SP), as shown in Fig. 2. The foreground mask and intensity information of each cell image are also provided to participants.

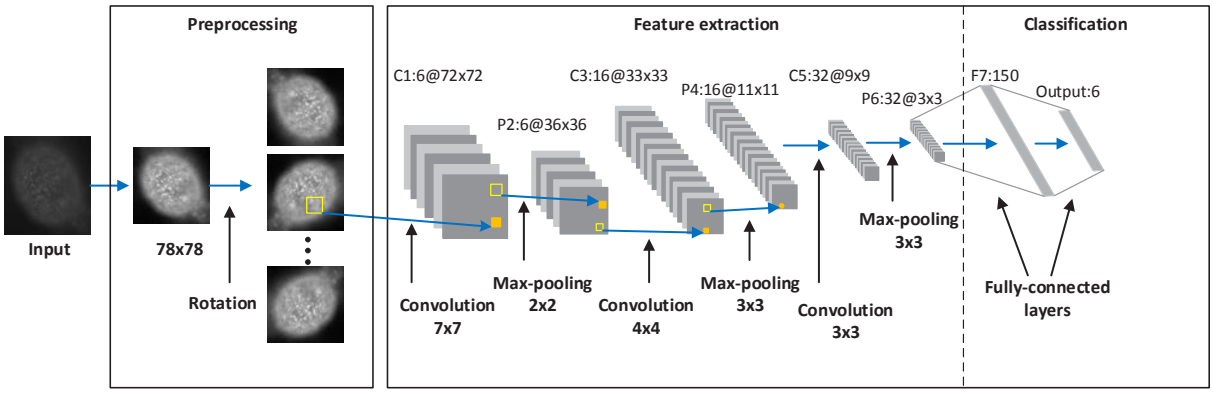

Figure 1. The architecture of our deep convolutional neural network classification system.

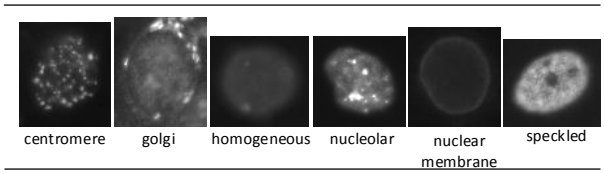

Figure 2. HEp-2 cell images.

B. Hyper-parameters Optimization

The hyper-parameters in our system can be categorized as two groups in Tables I and II. To tune these hyper-parameters, we randomly partition the 13,596 cell images into three subsets, that is, 64% for training (8701 images), 16% for validation (2175 images), and 20% for test (2720 images). This partition is utilized to all experiments. When tuning hyper-parameters, we use the original images without applying rotation. We first empirically set these parameters and then tune them based on the training set and validation set errors following [13]. The parameters are tuned until the validation set error becomes sufficiently small and stabilized. Also, overfitting should be avoided at the same time. The best model-relevant hyper-parameters and training-relevant hyper-parameters obtained are shown in Tables I and II.

With these hyper-parameters, our CNN is trained to learn the weights and biases of different layers, the total number of which are over 50,000. They are initialized in the way indicated in section III. *C*. The learning curves of the mean class accuracy (MCA) of the training, validation, and test sets are shown in Fig. 3. The result shows that overfitting does not occur during the learning process and the highest MCA of test set is 88.78%, which demonstrate the effectiveness of our system.

C. Effectiveness of Data Augmentation

Image rotation is applied to each cell image. We perform three different angle steps: 36°, 18° and 9° separately and the total number of images are increased by 10, 20 and 40 times respectively. The CNNs are trained with all the images of the training set after rotation. The MCA of test set for different angle steps is shown in Fig. 4.

Table I
MODEL-RELEVANT HYPER-PARAMETERS OBTAINED

Layer Number	Layer Type	Hyper-parameter
Layer 1	Convolution	Kernel number: 6
		Kernel size: 7×7
		Activation function: hyperbolic tangent $f(x) = 1.7159 \tanh(\frac{2}{3}x)$
Layer 2	Pooling	Pooling region size: 2×2
		Pooling method: max-pooling
Layer 3	Convolution	Kernel number: 16
		Kernel size: 4×4
		Activation function: hyperbolic tangent $f(x) = 1.7159 \tanh(\frac{2}{3}x)$
Layer 4	Pooling	Pooling region size: 3×3
		Pooling method: max-pooling
Layer 5	Convolution	Kernel number: 32
		Kernel size: 3×3
		Activation function: hyperbolic tangent $f(x) = 1.7159 \tanh(\frac{2}{3}x)$
Layer 6	Pooling	Pooling region size: 3×3
		Pooling method: max-pooling
Layer 7	Fully connection	Neurons number: 150
		Activation function: hyperbolic tangent $f(x) = 1.7159 \tanh(\frac{2}{3}x)$
Output	–	Activation function: softmax $\sigma(x_i) = \frac{e^{x_i}}{\sum_{i=1}^{6} e^{x_j}}, i, j = 1, ..., 6.$

Table II
TRAINING-RELEVANT HYPER-PARAMETERS OBTAINED

Hyper-parameter	Initial learning rate	Mini-batch size	Momentum	Weight decay
Value	0.01	113	0.9	0.0005

The result demonstrates that the classification accuracy is significantly improved after increasing the number of training images via rotation. It also shows that the accuracy increases as the angle step decreases from 36° to 18°. But from angle step 18° to 9°, the accuracy almost does not change, which may suggest that the networks trained with 18° rotation have become invariant to images rotation. Our submission adopts the angle step of 18°.

To improve the robustness of our system, we select four CNNs corresponding to the 75th, 85th, 95th and 100th epochs, to jointly classify a cell image. Table III shows the results. The 4CNNs trained with images rotated every 18° achieve MCA of 96.71%. The confusion matrix is shown in Fig. 5. It can be seen that four classes achieve over 97% accuracy, while golgi and speckled are relatively not well separately from the others.

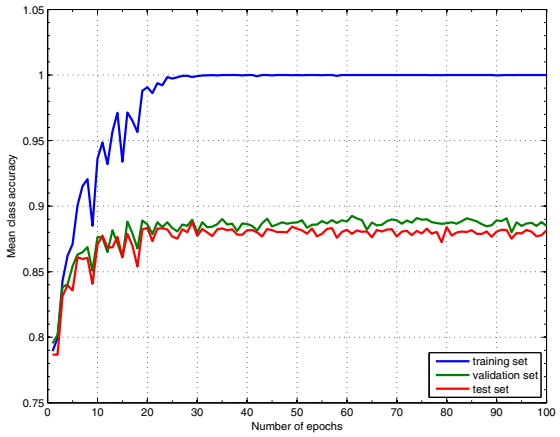

Figure 3. Mean class accuracy (MCA) of training, validation and test sets against the number of training epochs.

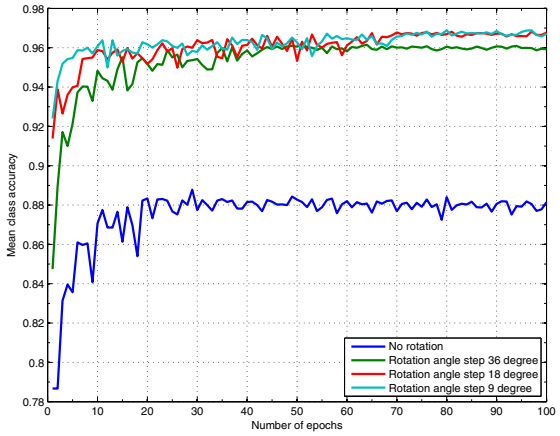

Figure 4. The impact of number of training images (rotation angle steps) on MCA of test set.

D. Performance on ICPR 2012 Cell Dataset

We also evaluate our methods on the ICPR 2012 cell dataset of the first contest [2]. It consists of 1455 cell images from 28 specimen images. The specimen images are captured differently and two of the six staining patterns are different from that of the ICPR 2014 dataset. We train the network with the training images (721) of ICPR 2012 dataset. They are applied the same preprocessing and rotation operation. We utilize the same hyper-parameters obtained on the ICPR 2014 dataset. It is observed that the network converges fast during training. We classify the test images also with four CNNs and average the output probabilities to obtain the class label. Table IV shows both the average classification accuracy (ACA), which is utilized at ICPR 2012 cells classification contest, and the MCA obtained with our system. The ACA of the best performer and the team utilizing CNN at the ICPR 2012 contest are also shown in Table IV for comparison. It can be seen that our networks outperform the CNN at the ICPR 2012 contest and is comparable to the best-performing method when the training set is enlarged via image rotation. This suggests that our network generalizes well among different cell datasets.

Table III
MCA OF OUR 4CNNS

	No rotation	Rotation angle step 36°	Rotation angle step 18° (**our submission**)	Rotation angle step 9°
4CNNs	88.10%	95.99%	96.71%	96.76%

	CE	GO	HO	NUC	NUM	SP
CE	97.95	0	0	1.03	0	1.03
GO	0	92.12	0	4.85	2.42	0.61
HO	0	0.38	98.47	0	0	1.15
NUC	0	0.19	0.38	98.87	0	0.57
NUM	0	0.45	0.23	0.23	98.87	0.23
SP	0.52	0.17	2.43	1.39	1.22	94.26

Figure 5. Confusion matrix of our system (%).

Table IV
PERFORMANCE (%) ON THE ICPR 2012 CELL DATASET

Method	2012 contest best-performing method	2012 contest CNN method	Our 4CNNs method				
				Rotation angle step (°)			
			No Rotation	72	36	18	9
ACA	68.7	59.8	63.2	69.9	68.7	69.6	69.8
MCA	–	–	62.7	71.2	70.7	**72.0**	**72.0**

V. CONLUSION

In this paper, we proposed an automatic HEp-2 cell staining patterns classification system with deep convolutional neural networks. Through evaluation experiments, we demonstrate the effectiveness of our system on this challenging classification task. The experimental results also confirm that the performance of deep neural networks can be greatly improved with data augmentation when the number of training images are limited. Moreover, competitive performance has been achieved on the ICPR 2012 cell dataset, indicating the good generalization capability of our network.

REFERENCES

[1] P. L. Meroni and P. H. Schur, "Ana screening: an old test with new recommendations," *Annals of the rheumatic diseases*, vol. 69, no. 8, pp. 1420–1422, 2010.

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

[3] 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.

[4] 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.

[5] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[6] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[7] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems(NIPS)*, vol. 1, no. 2, 2012.

[8] 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.

[9] X. Kong, K. Li, J. Cao, Q. Yang, and L. Wenyin, "Hep-2 cell pattern classification with discriminative dictionary learning," *Pattern Recognition*, vol. 47, no. 7, pp. 2379–2388, 2014.

[10] 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.

[11] L. Liu and L. Wang, "Hep-2 cell image classification with multiple linear descriptors," *Pattern Recognition*, vol. 47, no. 7, pp. 2400 – 2408, 2014.

[12] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of physiology*, vol. 160, no. 1, p. 106, 1962.

[13] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," *Neural Networks: Tricks of the Trade*, Springer Berlin Heidelberg, pp. 437-478, 2012.