HEp-2 Staining Pattern Classification

Strandmark, Petter; Ulén, Johannes; Kahl, Fredrik

Published in:
Pattern Recognition (ICPR), 2012 21st International Conference on

2012

HEp-2 Staining Pattern Classification

Petter Strandmark Johannes Ulén Fredrik Kahl
Centre for Mathematical Sciences, Lund University, Sweden
{*petter,ulen,fredrik*}*@maths.lth.se*

Abstract

Classifying images of HEp-2 cells from indirect immunofluorescence has important clinical applications. We have developed an automatic method based on random forests that classifies an HEp-2 cell image into one of six classes. The method is applied to the data set of the ICPR 2012 contest. The previously obtained best accuracy is 79.3% for this data set, whereas we obtain an accuracy of 97.4%. The key to our result is due to carefully designed feature descriptors for multiple level sets of the image intensity. These features characterize both the appearance and the shape of the cell image in a robust manner.

1. Indirect immunofluorescence

Indirect immunofluorescence (IIF) is an imaging modality detecting abundance of any protein for which there is an antibody in the sample tissue. Using IIF images, it is possible to locate cells as well as to classify their type in the sample. The manual classification of these cells suffer from the usual problems in medical imaging, such as: (i) The result is dependent on the experience and expertise of the specialist and (ii) it requires a lot of manual work. Reliable automatic systems are in great demand.

This paper introduces a new classification method for mitotic cells in IIF images. Each mitotic cell is classified into one out of six categories. This classification is one step in the detection of antinuclear autoantibodies, whose presence is a symptom of several systemic autoimmune rheumatic deceases, see [12]. We compare our method to the previous state of the art in an automatic setting, where it is assumed that the segmentation of the cell boundary in the image is known. In addition, we have investigated a semi-supervised setting, where the system is allowed to abstain from classifying an example if its confidence is low.

1.1. Data set

We use the data set from the HEp-2 Cells Classification contest [11] hosted by the International Conference on Pattern Recognition 2012. The training data set consists of 721 segmented and classified images. The top row of Figure 1 shows seven random images exactly as they appear in the data set. Since the test data set of the contest is not available to us, the classification system will be evaluated using leave-one-out cross-validation.

The task of the challenge is to classify a given segmented input image of a cell as one of six classes. The six classes of the data set are:

1. Homogeneous
2. Coarse-speckled
3. Fine-speckled
4. Nucleolar
5. Centromere
6. Cytoplasmatic.

Figure 2 shows 20 examples from each class.

1.2 Previous work

A recent work on the same data set as we use is the paper by Cordelli and Soda [6]. They used the full data set with 1457 cell images, evaluated many different methods and achieved a best accuracy of 79.3% using AdaBoost. Here accuracy is simply defined as the percentage of correctly classified images. Foggia et al. [8] and Percannella et al. [12] have also performed experiments on this data set, but for a different task: mitotic cell detection.

Earlier work on other data sets include [13] with 75% accuracy, [14] and [15] with 83% and 76%, respectively. These earlier works, however, used a different data set and only four classes, making direct comparison difficult. We are not aware of any other publicly available data set for this task. We are well aware that that comparing our results to the state of the art is difficult—the ICPR contest in combination with public data sets will make this easier in the future.

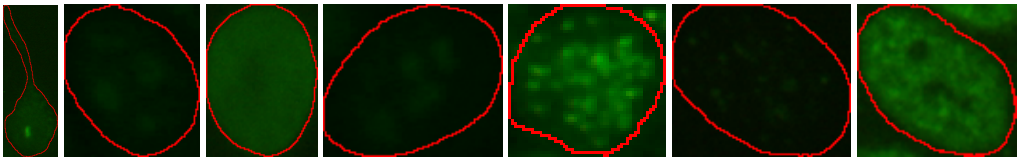

Figure 1. Seven random input images with given segmentation boundaries.

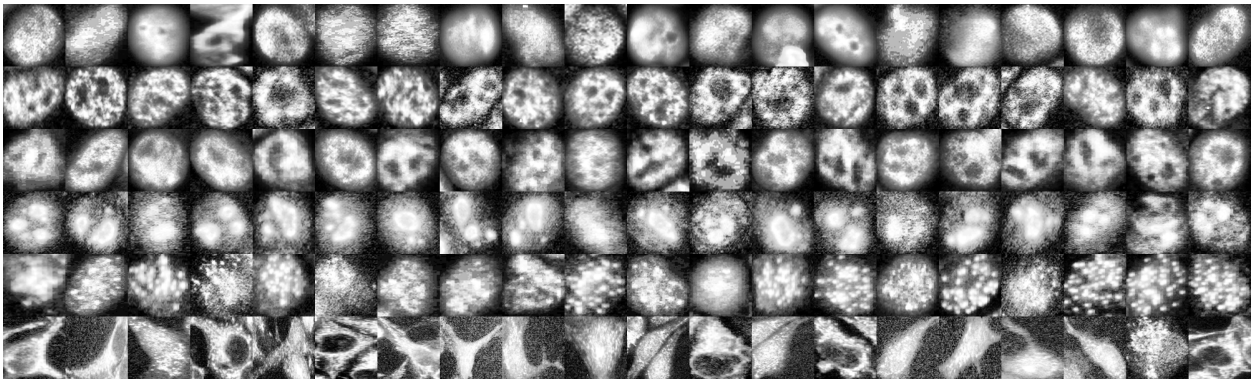

Figure 2. The rows show 20 samples from classes 1–6, respectively. The images have been resized to 100 $\times$ 100, converted to gray-scale and histogram equalized to enhance visibility.

2. Classification methods

We consider Convolutional Neural Nets to be the current state of the art for this kind of "easy" image classification tasks. For example, the work by Ciresan et al. [4] obtains a recognition rate of $99.73\% \pm 0.02\%$ on the MNIST data set of handwritten digits. On the same data set, a simple neural network trained via back-propagation [3] can also perform extremely well with an accuracy of 99.65%. Another recent result of neural nets is the German Traffic Sign Recognition Benchmark, where the winning entry [5] used a committee of convolutional neural networks and obtained an accuracy of 99.46%, significantly outperforming humans.

The method has some drawbacks, however, which makes it less useful in our context. First, neural networks have a strong tendency of overfitting. This can be alleviated by creating new training data by rotating, translating and skewing existing data. In this way, the same image is never fed to the network twice during training. In any case, the possibility of overfitting must be carefully monitored during training. Second, the training requires a vast amount of computer resources. For example, [4] used more than two weeks for training with a system with four graphics processing units.

The data set we are using have larger images than e.g. MNIST, so training an ensemble of networks is simply not computationally feasible. Instead, we use a random forest [2, 7]. A random forest computes averages over several hundreds of small decision trees, each of which is trained on a subset of the features and the training examples. It is not sensitive to overfitting and the total time required for training is 15 seconds. We also compare to a linear SVM classifier.

3. Features used for classification

In this section we describe the image features we are using as input to the random forest. Note that the classification method is not sensitive to overfitting which allows us to generate a lot of different features.

The fluorescence light captured is essentially monospectral. We therefore project the RGB color of each pixel onto the principal component of all pixel colors in the training set. Before any further processing, the background of each image is removed using the segmentation mask provided in the data set. Some images suffer from a few bright (saturated) pixels. By taking all intensities in an image and calculating a 10 binned histogram reduces this problem. We empty each bin which contains less than 0.5% of the pixels. If two consecutive bins are empty, we truncate the image intensity at this level. We then start to calculate features on the newly formed image.

The first feature is the "positive" or "intermediate" flag given in the data set. The second feature is the

aspect ratio of the image. The image is then thresholded at 20 intensities equally spaced from its minimum to its maximum intensity. Each threshold level gives a binary image on which we calculate the following features:

- Number of objects
- Area
- Area of the convex hull
- Eccentricity
- Euler number
- Perimeter.

These features are also calculated on the segmentation mask. Each binary image is also used in the same way as the mask to create a cut-out of the original image. On this cut-out we calculate the mean of the following features:

- Intensity
- Standard deviation (3-by-3 neighborhood)
- Entropy (9-by-9 neighborhood) [9]
- Range (3-by-3 neighborhood) [9].

The means of these features are also taken on a smaller mask formed by eroding the initial mask by a 5×5 kernel of ones. Another set of features are the average gradient magnitudes of the image after it has been smoothed with a Gaussian kernel with 10 different σ equally spaced in $[0.6, 10.5]$. The final features are based on gray-level co-occurrences [10] with offsets $\{-1, 1\}$. The co-occurrences are calculated on images which have been transformed to contain only $k = 2, 5, 8, 11, 14$ different intensities. Using the gray-level co-occurrences the contrast, correlation, energy and homogeneity are calculated for each k.

The described features make up a feature vector F_1 with 322 features. The same features are also calculated on the image after it has been smoothed by a gaussian kernel with $\sigma = 1$ and $\sigma = 2.5$ in order to reduce image noise. This forms two new feature vectors F_2 and F_3. The final feature vector is obtained by concatenation and subtraction as $(F_1, F_2 - F_1, F_3 - F_2)$. The total number of features is then 966. All the parameters (e.g. the amount of smoothing) were chosen by local optimization with cross-validation on a subset of the training data.

4. Results

The average time for loading a single-cell image from disk and computing all its features is 1.6 seconds. The time to classify the features is negligible (51µs on average). Combining this with a total training time of 15 seconds gives a quite efficient classification system. We used an ensemble of 500 trees, but observing the out-of-bag error rate [2] during training suggests that using less (150) trees would have worked just as well.

	1	2	3	4	5	6
1:	142	0	6	0	1	1
2:	0	108	0	1	0	0
3:	5	0	89	0	0	0
4:	0	0	0	102	0	0
5:	1	0	2	1	204	0
6:	0	0	0	1	0	57

Table 1. Confusion matrix. Rows show ground truth and columns our classification.

The overall accuracy of our system is 97.4% based on leave-one-out cross-validation. Figure 4 shows all 19 images which our system fails to correctly classify. The same information is presented in Table 1 as a confusion matrix. The accuracy of our system may be on par with the inter-lab variability [1], but additional studies are required to confirm this.

The votes from the individual trees can be used to obtain a (normalized) confidence score of each class. It is then interesting to see whether the misclassified examples have lower confidence than the correctly classified ones, see Figure 4. We allowed our system to abstain from classifying examples for which the confidence was low, in a similar manner to [15]. This feature is useful in a semi-supervised setting, where easy instances are classified automatically and the harder ones are sent to a specialist for further consideration. We can see in Figure 3 that allowing 11.1% of the cells to remain unclassified yields an accuracy of 100%.

5 Discussion

Whether our increased accuracy with respect to [6] is due to better and more discriminative features or the use of random forests is a relevant question. A simple one-against-one linear SVM classifier using our features gives an accuracy of 91.8%. Cordelli and Soda [6] tried several classifiers (kNN, SVM, AdaBoost, etc.), which strongly suggests that although random forests clearly outperforms the linear SVM, the main reason for our increased accuracy is the design of our features. By thresholding at different intensity levels and computing descriptors for each resulting level set, we incorporate more shape information of the image.

Reproducibility. We have performed our training and evaluation on a publicly available data set and we will make our source code available so that other researchers may reproduce our experimental results.

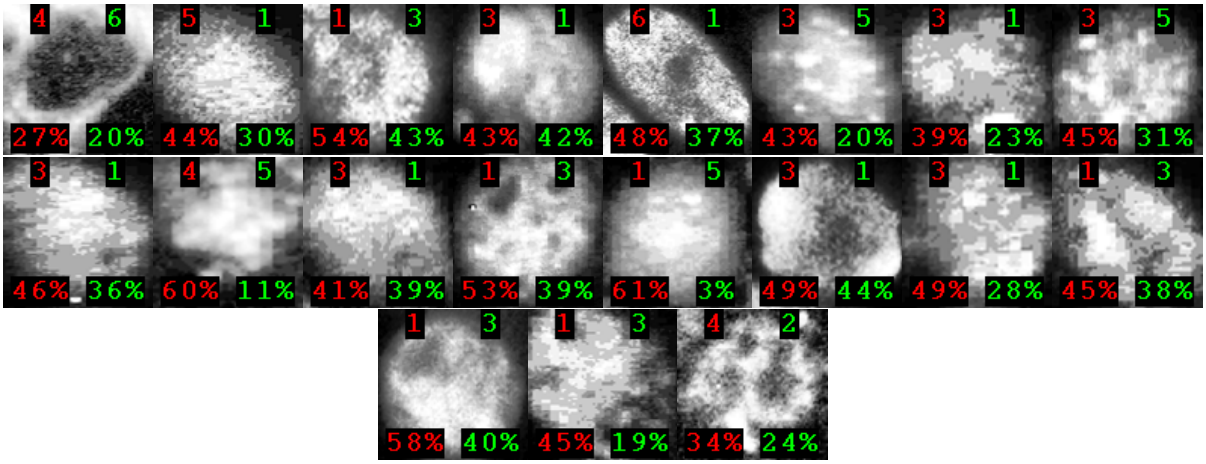

Figure 4. All 19 images for which our classification fails. The red number is our incorrect classification; the green is the ground truth. Each image also shows the confidence score.

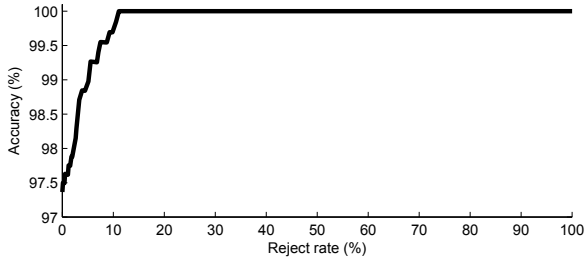

Figure 3. Accuracy as a function of the number of rejected examples. A reject rate of 11.1% gives a perfect accuracy.

References

[1] N. Bizzaro, R. Tozzoli, E. Tonutti, A. Piazza, F. Manoni, A. Ghirardello, D. Bassetti, D. Villalta, M. Pradella, and P. Rizzotti. Variability between methods to determine ANA, anti-dsDNA and anti-ENA autoantibodies: a collaborative study with the biomedical industry. *Journal of Immunological Methods*, 219(1–2):99 – 107, 1998.

[2] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[3] D. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition. Neural Computation, Volume 22, Number 12, December 2010, 2010.

[4] D. Ciresan, U. Meier, L. Gambardella, and J. Schmidhuber. Convolutional neural network committees for handwritten character classification. In *Document Analysis and Recognition (ICDAR), 2011 International Conference on*, pages 1135–1139. IEEE, 2011.

[5] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. Multi column deep neural network for traffic sign classification. *Neural Networks (to appear)*, 2012.

[6] E. Cordelli and P. Soda. Color to grayscale staining pattern representation in IIF. In *CBMS*, pages 1–6. IEEE, 2011.

[7] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests for classification, regression, density estimation, manifold learning and semi-supervised learning. Technical Report MSR-TR-2011-114, Microsoft Research, 2011.

[8] P. Foggia, G. Percannella, P. Soda, and M. Vento. Early experiences in mitotic cells recognition on HEp-2 slides. In T. S. Dillon, D. L. Rubin, W. Gallagher, A. S. Sidhu, and A. Tsymbal, editors, *CBMS*, pages 38–43. IEEE, 2010.

[9] R. Gonzalez, R. Woods, and S. Eddins. *Digital Image Processing Using MATLAB*. Prentice Hall Press, 2007.

[10] R. Haralick, K. Shanmugam, and I. Dinstein. Textural features for image classification. *Systems, Man and Cybernetics, IEEE Transactions on*, 3(6):610–621, 1973.

[11] G. Percannella, P. Foggia, and P. Soda. Contest on HEp-2 cells classification. http://mivia.unisa.it/hep2contest/organization.shtml, 2012. Read 2012-03-22.

[12] G. Percannella, P. Soda, and M. Vento. Mitotic HEp-2 cells recognition under class skew. In *Int. Conf. Image Analysis and Processing – Volume Part II*, pages 353–362. Springer-Verlag, 2011.

[13] P. Perner, H. Perner, and B. Müller. Mining knowledge for HEp-2 cell image classification. *Artificial Intelligence in Medicine*, 26:161–173, 2002.

[14] U. Sack, S. Knoechner, H. Warschkau, U. Pigla, F. Emmrich, and M. Kamprad. Computer-assisted classification of hep-2 immunofluorescence patterns in autoimmune diagnostics. *Autoimmunity Reviews*, 2(5):298–304, 2003.

[15] P. Soda and G. Iannello. Aggregation of classifiers for staining pattern recognition in antinuclear autoantibodies analysis. *Trans. Info. Tech. Biomed.*, 13:322–329, May 2009.