CLASSIFICATION OF DERMOSCOPY PATTERNS USING DEEP CONVOLUTIONAL NEURAL NETWORKS

Sergey Demyanov[1] *Rajib Chakravorty*[1] *Mani Abedini*[1] *Alan Halpern*[2] *Rahil Garnavi*[1]

[1]IBM Research - Australia, Melbourne, VIC, Australia
[sergeyde, rachakra, mabedini, rahilgar]@au1.ibm.com

[2]Memorial Sloan-Kettering Cancer Center, New York, NY, USA
halperna@mskcc.org

ABSTRACT

Detection of dermoscopic patterns, such as typical network and regular globules, is an important step in the skin lesion analysis. This is one of the steps, required to compute the ABCD-score, commonly used for lesion type classification. In this article, we investigate the possibility of automatically detect dermoscopic patterns using deep convolutional neural networks and other image classification algorithms. For the evaluation, we employ the dataset obtained through collaboration with the International Skin Imaging Collaboration (ISIC), including 211 lesions manually annotated by domain experts, generating over 2000 samples of each class (network and globules). Experimental results demonstrates that we can correctly classify 88% of network examples, and 83% of globules example. The best results are achieved by a convolutional neural network with 8 layers.

Index Terms— Dermoscopy patterns, Image classification, Convolutional Networks, Deep Learning

1. INTRODUCTION

Even though melanoma is one of the most common and fatal types of skin cancers, early detection of the disease can significantly reduce mortality rate [1]. For this purpose, experts use various heuristic methods. The most common method is based on the set of ABCD scores [2], standing for Asymmetry, Border irregularity, Color variation and Dermoscopic patterns. Other methods, such are Menzies method or CASH rule, also use various visual lesion features. However, the correct diagnosis of a skin lesion is not trivial even for health care professionals [3], [4], and therefore is highly biased. Automated analysis of lesion images can assist domain expert to make better decision.

The score for dermoscopy is based on the presence (or absence) of abnormal dermoscopic patterns, such as "network" or "globules", within the lesion area. Each of them increases the probability of a lesion to be classified as malignant [5].

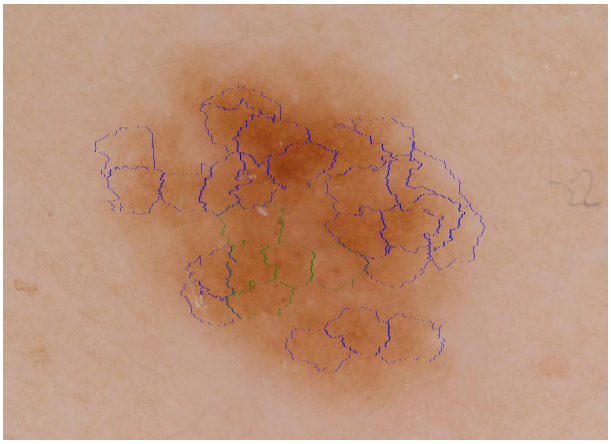

Fig. 1. An example of a skin lesion with segments of regular globules (green) and typical network (blue).

Hence, accurate identification of these patterns is an important part of the automated diagnostic system. Moreover, automated pattern detection provides the evidence that supports the diagnostic decision to a health care professional.

In this paper, we use a dataset of high-resolution skin images to build the classifiers, that are able to identify two types of dermoscopy patterns: typical network and regular globules. For this purpose, we train deep convolutional neural networks, which demonstrate state-of-the-art results in the image classification problems, and compare them with the standard algorithms, based on unsupervised feature extraction and hand-crafted features.

2. DATASET

The International Skin Imaging Collaboration (ISIC) dataset [6] is one of the largest sets of high-resolution non-polarized dermoscopic images, which were manually labeled by professional doctors according to the lesion type (malignant or benign). These images were collected from a number of sources

978-1-4799-2349-6/16/$31.00 ©2016 IEEE

Fig. 2. "Typical Network" examples

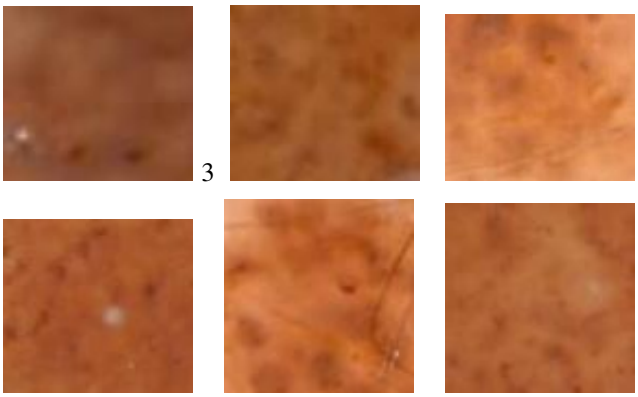

Fig. 3. "Regular Globules" examples

by ISIC. Therefore, the image acquisition process is guaranteed to be non-uniform in terms of focal length, distance from the skin, lighting conditions, etc. This dataset was used for automatic lesion type classification [7]. A subset of 211 images was also divided on segments of various shapes and sizes, and labeled according to the presence (or absence) of the particular dermoscopic patterns: typical networks or regular globules. An example of such segmentation is shown on Fig. 1. Using these labels, two datasets of segments (networks and globules) were constructed. They contain subimages extracted from the rectangles, surrounding the labeled segments. The network dataset contains 2705 positive and 12061 negative examples, and the globules dataset contains 2134 positive and 12423 negative examples. The size of subimages varies from 20×15 to 222×317 pixels. In this article, we refer to them as "Typical Network" and "Regular Globules" datasets. The examples of each class are shown on Fig. 2 - 3.

3. ALGORITHMS

3.1. Convolutional neural networks

Deep convolutional neural networks (CNN) have recently become a major tool for image classification problem. They can learn features automatically from pixel images, thus there is no need to manually handcraft features. The combination of convolutional and subsampling layers yields robustness to variations in image location, and also significantly reduces the number of trained parameters. The recent appearance of large labeled training sets (e.g., ImageNet dataset [8]), fast GPU implementations [9, 10], and the improvements in the architecture of neural networks [11], [12] has made it possible to train very deep networks without overfitting in a reasonable amount of time.

In this article, we employ neural networks to solve the problem of dermoscopy pattern classification. We start from a shallow architecture with one layer, and increase the depth until it does not improve the training set cross-validation accuracy anymore. In order to prevent overfitting, we use several methods of data augmentation, which include geometrical variations and changes in pixel intensities.

3.2. Baseline algorithms

For comparison, we use three algorithms (K-means clustering, sparse coding and Fisher Kernel encoding), which extract features in an unsupervised manner, and classify them using SVM.

The first two algorithms (K-means clustering and sparse coding) initially extract small random patches from the training images, and then learn *dictionaries*, which represent the most common types of patches. Next, the algorithms calculate features for all patches of each image using the obtained dictionaries, and average them to obtain the features for the whole image. To obtain a better representation, the features can also be averaged over several spatial blocks, and concatenated after that. For example, for $2 \times 2 = 4$ blocks the number of features is 4 times larger.

The first algorithm is described in [13]. It generates a dictionary using K-means clustering over a set of random patches extracted from both classes. The features for each patch are computed using the "soft" scheme of assignment. If the distances to the cluster centers are $d_{i \in \{1,\dots,K\}}$, then the features are $x_i = \max(m - d_i, 0)$, where $m = \sum_i^K d_i / K$ is the mean distance. Thus, for all clusters which are further than the average distance the feature value is zero.

The second algorithm (sparse coding, [14]) creates the dictionary D, solving the following minimization problem

$$(D, A) = \arg\min_{D,A} \sum_{i=1}^{N} \frac{1}{2} ||x_i - D\alpha_i||_2^2, \;\; s.t. \; ||\alpha_i||_1 \leq \lambda$$

Here $x_{i \in \{1,\dots,N\}}$ are the random patches, $A = \alpha_{i \in \{1,\dots,N\}}$ is their approximation using the elements of the dictionary D, and λ is the parameter, which controls the representation sparsity. After the dictionary is found, the features for all image patches are computed using the same minimization problem for the fixed D: $\alpha = \arg\min_\alpha \frac{1}{2} ||x - D\alpha||_2^2, \;\; s.t. \; ||\alpha||_1 \leq \lambda$.

As another baseline we use Fisher kernel-feature encoding [15, 16], a well-established advanced feature encoding technique based on learning a visual dictionary [17]. Recent evaluations [18, 19] show this encoding method achieved best results in many cases. First, local features are extracted from each image (bag of features), then Fisher kernel encodes the distribution information of the feature points which can separate the image specific information from the noisy local features. Fisher kernel-encoded features can be represented using a linear model which is computationally efficient. Then the features extracted from images are given to a classifier. In our experiments, first local visual features are extracted from each image by using Dense Scale-Invariant Feature Transform (Dense SIFT) [20] and Speeded Up Robust Features (SURF) [21] (we use OpenCV implementation of SIFT and SURF). Fisher kernel technique is then applied to convert bag of features to a single feature vector per each image.

4. EXPERIMENTS

For each algorithm we performed 20 experiments with different randomized seeds. The obtained values of the test errors were then averaged, and the mean error and the standard deviation are reported. The procedure for each seed is described below.

Firstly, we split the datasets to the training and test sets. Since we could observe only whole images, we split the datasets such that segments from the same image always belonged to the same set. It allowed to obtain an unbiased estimation of accuracy. In our experiments we used the test ratio 0.2, so we chose 169 files for training and 42 for testing.

Second, in order to generate balanced training and test sets, we bound the number of negative samples to the size of positive samples by random selection from all negative samples. This significantly reduces training time and allows to use accuracy as a measure of performance.

Third, we found the optimal hyperparameters. For this purpose we performed 5-fold cross-validation for each possible value of a hyperparameter, and chose the one which yields the lowest cross-validation error.

4.1. Convolutional neural networks

For the neural network the modified hyperparameters were: a) the size S of the input image, b) the number of pairs of convolutional and subsampling layers, c) the size and the number of convolutional layer filters, d) the size of hidden fully connected layers. We trained the network using the standard algorithm of Stochastic Gradient Descent (SGD) with the batch size equal to 32. The initial learning rate 0.01 which was decreased in 2% every epoch. The total number of epochs was 50. To make convergence more robust we used the momentum rate equal to 0.9. Before training we computed the mean pixel value across all training images and subtracted from both training and test images.

Since the neural network accepts a fixed size of input image, we extracted images of some size $S \times S$ from random locations of the original image. When data augmentation was used, we also varied scale and angle of the extracted subimage. The variation parameters were chosen such that the extracted image was always inside the original image. If one of the dimensions of the original image was smaller than S, we ignored it. In the data augmentation regime we also varied image color using the principal components of the pixel distribution, as it is described in [22]. The value of standard deviation for variation was also 0.1.

For our experiments we employed the fast GPU implementation of neural networks for Matlab [23], using the NVidia 780Ti graphics card.

4.2. Baseline algorithms

For the dictionary-based algorithms the parameters were: a) the patch size, b) the size of the dictionary, c) the stride between the extracted image patches, d) the number of blocks for pooling, e) the parameters of SVM (C and γ). In our experiments we fixed the patch size to be 6×6, which was found to be the best in [13], and used the fixed stride size 2 in both dimensions. Other parameters were tuned using cross-validation. We used the SVM with RBF kernel (1),

$$\langle x, y \rangle = \exp(-\gamma ||x - y||_2^2) \tag{1}$$

which allows to vary the classifier's flexibility changing the parameter γ. The sets of possible values were $C = 10^i, -1 \leq i \leq 2$ and $\gamma = 10^i, -5 \leq i \leq -2$ accordingly. The parameter λ for sparse coding algorithm was fixed to the standard value 0.15. Notice, that dictionary-based algorithms are by nature train a shift-invariant classifier, because the extracted patches for shifted images are mostly the same. We have also tried to augment the datasets by scaled and rotated images. It required much more training time, but did not give any significant improvement of accuracy.

For the Fisher Kernel method, we first extracted random SIFT/SURF features and build 32 clusters using Gaussian Mixture Models (GMM). Then we performed PCA for each cluster and kept new generated features that explain 95% of total variation. Finally, we represented each image as a bag of the rest SIFT/SURF features for different locations, and combined them into a single vector of size 32 using the trained Fisher kernel. Since RBF kernel is sensitive to data scaling, we performed normalization of features such that they have zero mean and unit variance. First, we normalized random patches used for clustering, and used the same parameters to normalize all other image patches. Second, we normalized the training set features, and transformed the test set features the same way. All results were computed by the "SPAMS", "Scikit-learn" and OpenCV toolboxes in Python.

5. EXPERIMENTAL RESULTS

The final classification results for both datasets are summarized in Table 1. First of all, we can see that all classifiers (except Feature kernel) are able to achieve high prediction accuracy. The best mean accuracy for the "Typical Network" and "Regular Globules" datasets is $\approx 88\%$ and $\approx 83\%$ accordingly. We suppose that further improvement is limited by the nature of data. From Fig. 2 - 3 we can notice that the examples of different classes are very similar, so even humans have difficulties to distinguish between two classes. Nevertheless, these results are high enough to use the classifiers as a part of a more complex system of lesion analysis.

Table 1. The Table represents the mean classification error and standard deviation for different algorithms. The best results are obtained by the CNN with data augmentation.

Algorithm	*Typical Network*	*Regular Globules*
CNN, with d/a	12.06 ± 2.69	16.81 ± 3.16
CNN, without d/a	14.42 ± 3.27	18.20 ± 4.13
K-means clustering	13.30 ± 1.86	18.63 ± 3.60
Sparse coding	15.88 ± 1.50	19.14 ± 3.27
Fisher kernel	24.37 ± 1.11	25.37 ± 3.80

From Table 1 we can see that the highest accuracy is achieved by the convolutional neural network. Its architecture was tuned to achieve the best performance, and the final result is described in Table 2. It has three pairs of convolutional and subsampling layers, followed by two fully-connected layers. The small amount of large 5×5 filters is followed by the large amount of small 3×3 filters on higher layers. This architecture allows to capture large visual structures, that correspond to skin abnormalities. However, such a highly flexible classifier requires data augmentation in order to prevent overfitting. Interestingly, we observe that another widely used regularization method "dropout" [12] significantly worsens the results.

Table 2. The architecture of the CNN, which yields the highest classification accuracy. All convolutional layers have padding 1. All subsampling layers have size 3×3 and stride 2, and perform max-pooling. Each convolutional and fully-connected layer is followed by a ReLU [11] function.

N	*Layer Type*	*Weights*	*Neurons*
0	Input		$44 \times 44 \times 3$
1	Convolutional	$5 \times 5 \times 3 \times 16$	$42 \times 42 \times 16$
2	Subsampling		$21 \times 21 \times 16$
3	Convolutional	$4 \times 4 \times 16 \times 32$	$20 \times 20 \times 32$
4	Subsampling		$10 \times 10 \times 32$
5	Convolutional	$3 \times 3 \times 32 \times 32$	$10 \times 10 \times 32$
6	Subsampling		$5 \times 5 \times 32$
7	Fully-Connected	$5 \times 5 \times 32 \times 128$	128
8	Output	128×2	2

Similar to the results reported in [13], the best performance for K-means clustering features is achieved by using 1000 clusters and $2 \times 2 = 4$ pooling blocks. Therefore, the number of features for these parameters was ≈ 4000. Some of the features were excluded, because they had zero variance. We have also found that the same number of clusters yields the near best performance for the sparse coding algorithm. Further increase of the number of clusters did not improve the accuracy. Actually, in our experiments after further increase of the number of blocks the accuracy degraded. This happened because the number of features becomes too large, so even a large value of the SVM regularization parameter does not prevent overfitting.

The best values of the SVM kernel parameter γ are 10^{-3} and 10^{-4} for the K-means clustering and sparse coding algorithms accordingly for both "typical network" and "regular globules" datasets. Note, that smaller value of γ corresponds to a higher kernel dimensionalily. The best values of the parameter C are also different: 10 for the K-means and 1 for the sparse coding algorithms. Therefore, the sparse coding features are more distinct, that allows to use a more flexible classifier with less strong regularization. Interestingly, that the Fisher kernel method performs significantly worse than other algorithms. We suppose it is because SIFT and SURF are not effective on small images with non discriminative key points.

While the best network architecture has 8 layers, the difference from simple architectures with just one pair of convolutional-subsampling layers and 24×24 input images was just several percent. However, the larger architecture requires data augmentation as a necessary part of training. From Table 1 we can see that without data augmentation the accuracy of CNN is close, or worse than the accuracy achieved by other classification algorithms. The p-values of the Mann-Whitney U-test (0.089 and $8 \cdot 10^{-8}$) confirm that the CNN with data augmentation is statistically better than K-means clustering algorithm, while the CNN without data augmentation is not. In general, all algorithms show similar performance results, what is common for simple classification tasks like this, when the the image objects do not have a complicated shape and high variance of appearance.

6. CONCLUSION

In this paper, we have investigated the possibility to automatically identify the type of dermoscopic patterns within the skin lesion. The experiments show that this problem can be successfully solved by using deep convolutional neural networks coupled by data augmentation. The achieved accuracy is 88% for the "typical network" pattern and 83% for the "regular globules" pattern. We believe that the proposed classification system can be used as a part of a more complex framework for skin lesion analysis.

7. REFERENCES

[1] A.C. Geller, S.M. Swetter, K. Brooks, M.F. Demierre, and A.L. Yaroch, "Screening, early detection and trends for melanoma: Current status (2000-2006) and future directions," *J. American Academy of Dermatology*, vol. 57, pp. 555–572, 2007.

[2] S.W. Riemann, A.B. Cognetta, R. Talamini, R. Corona, and F. Sara et. al, "Abcd rule of dermatoscopy: a new method for early recognition of malignant melanoma," *Eur J. Dermatology*, vol. 4, pp. 521–527, 1994.

[3] I. Maglogiannis and C. Doukas, "Overview of advanced computer vision systems for skin lesions characterisation," *IEEE Transaction on Information Technology in Biomedicine*, vol. 13, no. 5, pp. 721733, 2009.

[4] C.A. Morton and R.M. Mackie, "Clinical accuracy of the diagnosis of cutaneous malignant melanoma," *Br J Dermatol*, vol. 138, no. 2, pp. 283287, 1998.

[5] U. Weigert, W.H.C. Burgdorf, and Wilhelm Stolz, *ABCD Rule*, pp. 113–117, Second edition edition.

[6] "International Skin Imaging Collaboration (ISIC) Project," `http://isdis.net/isic-project/`, [Online; accessed 20-Oct-2015].

[7] Noel Codella, Junjie Cai, Mani Abedini, Rahil Garnavi, Alan Halpern, and John R Smith, "Deep learning, sparse coding, and svm for melanoma recognition in dermoscopy images," in *Machine Learning in Medical Imaging*, pp. 118–126. Springer, 2015.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[9] Alex Krizhevsky, "cuda-convnet2," `https://code.google.com/p/cuda-convnet2`, 2014.

[10] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[11] Vinod Nair and Geoffrey E Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, 2010, pp. 807–814.

[12] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] Adam Coates, Andrew Y Ng, and Honglak Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International conference on artificial intelligence and statistics*, 2011, pp. 215–223.

[14] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *The Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[15] Florent Perronnin and Christopher Dance, "Fisher kernels on visual vocabularies for image categorization," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE, 2007, pp. 1–8.

[16] Florent Perronnin, Jorge Sánchez, and Thomas Mensink, "Improving the fisher kernel for large-scale image classification," in *Computer Vision–ECCV 2010*, pp. 143–156. Springer, 2010.

[17] Li Fei-Fei and Pietro Perona, "A bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. IEEE, 2005, vol. 2, pp. 524–531.

[18] Ken Chatfield, Victor S Lempitsky, Andrea Vedaldi, and Andrew Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods.," in *BMVC*, 2011, vol. 2, p. 8.

[19] Qiang Chen, Mani Abedini, Rahil Garnavi, and Xi Liang, "Ibm research australia at lifeclef2014: Plant identification task," in *Working notes of CLEF 2014 conference*, 2014.

[20] Ivica Dimitrovski, Dragi Kocev, Suzana Loskovska, and Sašo Džeroski, "Hierarchical annotation of medical images," *Pattern Recognition*, vol. 44, no. 10, pp. 2436–2449, 2011.

[21] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *Computer vision–ECCV 2006*, pp. 404–417. Springer, 2006.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[23] Sergey Demyanov, "ConvNet for Matlab," `https://github.com/sdemyanov/ConvNet`, 2014, [Online; accessed 20-Oct-2015].