Electronics and Electrical Engineering

AIMS Electronics and Electrical Engineering, 4(1): 47–56.
DOI: 10.3934/ElectrEng.2020.1.47
Received: 15 September 2019
Accepted: 18 November 2019
Published: 27 November 2019

http://www.aimspress.com/journal/ElectrEng

Research article

Deep learning for actinic keratosis classification

Loris Nanni[1], Michelangelo Paci[2], Gianluca Maguolo[1,*], Stefano Ghidoni[1]

[1] Information Engineering, University of Padua, Via Gradenigo 6, 35131 Padova, Italy

[2] Faculty of Medicine and Health Technology, Tampere University

*** Correspondence:** Email: gianluca.maguolo@phd.unipd.it.

Abstract: Classification of biological images plays a crucial role in many biological problems, e.g. recognition of cell phenotypes and maturation levels, localization of cell organelles and histopathological classification, and holds the potential to support early diagnosis, which is critical in disease prevention. In this paper, we tested different ensemble of canonical and deep classifiers to provide accurate identification of actinic keratosis (AK), one of the most common skin lesions that could degenerate into lethal squamous cell carcinomas.

We used a clinical image dataset to build and test different ensembles of support vector machines trained by handcrafted descriptors and convolutional neural networks (CNNs) for which we experimented different learning rates, augmentation techniques (e.g. warping) and topologies.

Our results show that the proposed ensemble obtains performance comparable to the state of the art. To reproduce the experiments reported in this paper, the MATLAB code of all the descriptors is available at https://github.com/LorisNanni.

Keywords: microscopy imaging classification; deep learning; convolutional neural networks; bioimage classifications; actinic keratosis

1. Introduction

Automated classification of biological images by means of computer vision can be applied with great success to a variety of biological problems – examples include organelle classification and location, assessment of maturation level of cells, and tissue and cancer recognition [1]. Such vision systems have a key role when dealing with preventable diseases, as they can sensibly increase the number of early diagnoses: they could be used to perform frequent screenings on large populations at a very low cost, with a very positive impact on people health.

Actinic Keratosis (AK) is one of the most common precancerous skin lesions, known also as solar keratosis that can degenerate into squamous cell carcinoma (SCC) [2]. Correlations were observed between AK risk and several factors, e.g. i) genetic factors (blue eyes and childhood freckling, albinism, xeroderma pigmentosus); ii) advancing age; iii) exposure to augmented ultraviolet radiation, and iv) immunosuppressive therapies following organ transplants. Avoiding an excessive sun exposure and using sunscreens are recommended to prevent AK [3]. The evolution of an AK lesion shows great variability: it can progress towards more invasive and also lethal diseases, remain stable or even regress [4]. Rigel et al. [4] showed that 60% to 80% SCC derived from AK lesions; however, the risk of degeneration of a single AK lesion is relatively low. Early diagnosis is fundamental, especially for subjects with high risk, since SCC is the cause of thousands of preventable deaths related to skin cancer [3].

The gold standard for diagnosing SCC is skin biopsy, but several non-invasive alternatives are also common in the clinical practice: optical coherence tomography, dermoscopy, reflectance confocal microscopy or stripping mRNA are less intrusive for the patient. The drawback for such techniques is that training and experience of the clinicians play a key role in generating the correct diagnosis [5]. Recently, a few studies [6–10] proposed the automatic identification of AK lesions using standard clinical photography [11]. Spyridonos et al. [10] proposed an approach based on late fusion of support vector machines (SVMs) fed by shallow features (color-texture features based on local binary patterns [12]) and convolutional neural networks (CNNs) (AlexNet, VGG-19, VGG-19 and GoogleNet).

In this work, we used the same dataset considered in [10] to develop a high performance AK lesion identification system. The system is based on two modules. The first is an ensemble of SVMs trained individually with one handcrafted descriptor each and then combined by sum rule. A hand-crafted method is a non-trainable algorithm that extracts features from an image and that has been created by an expert to look for specific patterns in an image. Hence, these methods are often created by experts that already know which patterns are useful for the classification problem. The second module is an ensemble of CNNs again combined by sum rule. CNNs were trained exploiting different data augmentation strategies. We were able to demonstrate that our best approach based on CNNs can be successfully used to build a bioimage system with high predictive power. Our main contributions are:

1. Exhaustive experiments with different architectures and data augmentation strategies, as well as many different handcrafted features, providing a good baseline for researchers in this field,
2. The investigation of which data augmentation strategies are the most effective for skin classification problems,
3. The proof that an ensemble of different CNNs trained with different data augmentation strategies outperforms the single networks and the handcrafted baselines.

To reproduce the method proposed in this paper, the MATLAB code is available at https://github.com/LorisNanni.

2. Methods

2.1. Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep neural networks suitable to find patterns in images. The key elements in a CNN are the convolutional and pooling layers. A

convolutional layer takes a 3D tensor as an input and computes several discrete convolutions between the tensor and a set of kernels that are learned. The RGB input image is provided to the network in the first convolutional layer as a 3D tensor. Each convolution between a kernel and the input generates a 2D image as an output. A single convolutional layer has multiple kernels and computes a convolution for each one, therefore the output of a convolutional layer is constituted by a set of multiple 2D tensors. They are in turn stacked along the third dimension: the output can then be seen as a 3D tensor. The 2D output images are called channels.

Convolutional layers are useful to spot relevant local features in the input image. The general understanding is that neural networks perform pattern recognition by gradually decreasing the dimensionality of the input by means of the hidden layers. In general, however, the output of a convolutional layer is not necessarily smaller than its input: to reduce the data size, convolutional layers are usually followed by pooling layers. A pooling layer takes a 3D tensor as an input and works on every channel independently. Every channel is divided into $m \times n$ rectangles (usually $m = n$, but this is not mandatory). Each region is then mapped into a single pixel, either taking the maximum (max-pooling) or the average (average-pooling) of the $m \times n$ pixels of the region. This operation reduces the dimension of the hidden representation by $m \times n$.

The two types of layer mentioned above gradually perform feature extraction from the input image. After a number of convolutional-pooling layer pairs, one or more fully connected layers are placed, followd by a softmax layer. Such elements implement the classification task, based on the output of the convolutional stack. The performance of CNNs clearly outperformed previous machine learning techniques in several fields like image classification [13], image segmentation [14] and object detection [15]. Training a CNN often requires a complex training phase based on a large dataset. However, a large number of training images is not always available: when this is the case, pre-trained networks can also be used as a starting point: they are then *fine-tuned* (i.e., adapted to a new problem) using smaller datasets. In this work we used three different pretrained CNNs: GoogleNet [16], ResNet50 and ResNet101 [17].

GoogleNet [16] is a light network that won the ImageNet Challenge in 2014 [18]. It only has 4 million parameters, thanks to its main feature, the so-called inception modules. These modules are made by multiple sequences of convolutional layers with different sizes, whose outputs are combined at the end of the module. Since these convolutions are small, they have a small number of parameters.

Conversely, ResNet50 and ResNet101 [17] are very deep architectures where 50 and 101 stand for the number of layers of the network. The ResNet main feature consists in the so-called skip-connections, that are modules made by a convolutional layer whose output is summed to its input. This helps the gradient flow at training time and allows the network to reach better local minima. A ResNet architecture won the ImageNet Challenge in 2015.

In each following test, for each CNN, we used two different learning rates.

2.2. Data augmentation

It has been already observed that large image datasets are needed to successfully train CNNs. However, such large amount of data may not be available for a specific problem. This is often the case when dealing with medical images: they may require expensive equipment, and the number of subjects presenting the illness or lesion is (luckily!) limited. One of the most interesting techniques to

compensate for the low number of training samples available is data augmentation: it applies small modifications to the input images (e.g., rotations, translations, flipping) to artificially produce new images. Even though the correlation between each original image and the augmented ones is clear, this approach has experimentally demonstrated to be effective.

In this paper, we heavily exploit data augmentation and test a wide range of techniques to highlight the most effective ones. We considered four different data augmentation protocols. The first one consists in the application of i) a random reflection along both x and y axes; ii) a random linear scaling in both x and y directions by two different factors uniformly sampled in [1,2]; iii) random translations in both directions randomly sampled in [0, 5] pixels; iv) a random rotation by an angle randomly sampled in [-10, 10]; v) vertical and horizontal shear by a factor uniformly sampled in [0, 30]. The transformations listed above are combined in different ways:

- Easy (EA), only reflection on x-axis;
- Standard (ST), reflection and scale on both axes;
- Long1 (L1), all the methods above described;
- Long2 (L2), as L1 but without shear.

The second protocol consists in applying the Thin-Plate-Spline Warping (TPS) [19]. TPS is an algorithm designed to modify images by randomly translating some of their pixels. The result is a similar image where every part has been locally resized. The inputs of the algorithm are an image, a list of input pixels I and a list of output pixels I^*. Every pixel of I is mapped into the corresponding pixel of I^*. The pixels of the input image that do not belong to I are mapped into the new image using inverse distance weighted interpolation.

The third data augmentation protocol consists in creating three new images with the following algorithms: contrast augmentation, sharpness augmentation and color shifting, as detailed in the following:

1. The contrast augmentation takes two values a, b as inputs. These values represent the lowest and the largest value of intensity of the image, respectively. All the input pixels whose intensity is lower than a (or larger than b) are mapped to 0 (or 255). The intensities inside the $[a, b]$ range are linearly scaled.
2. The sharpness augmentation works by subtracting a blurred version of the image to the original one. The blurred image is obtained with a Gaussian filter whose variance is set to one. Then the output image is obtained by $I^* = I + (I_b - I) \times k$, where I_b is the blurred image and k is a coefficient set to two in our case.
3. Color shifting is a simple algorithm that takes the three RGB components of an image and adds three different numbers to all the pixels of the three channels. We use different shifting parameters in every fold.

We named this approach CSC in the next sections.

The fourth protocol consists in the creation of seven new artificial images for every input image. The new images are obtained by:

1. Blurring the input image using a Gaussian filter with a variance of 0.1;
2. Adding random noise sampled from $[0, 225]$ in the blue channel;
3. Adding random noise uniformly sampled in [0,75];
4. Saturating the image by mapping it to the HSV format and augmenting the second channel by 20%, then mapping it back to RGB. We used the MATLAB implementation of the algorithms to convert images from RGB to HSV color spaces and vice-versa;

5. Adding contrast by saturating the top 1% and the bottom 1%. This means that 1% of the pixels with highest and lowest contrast are respectively mapped to 1 and 0, which are the maximum and minimum value for the contrast, respectively. The contrast of the other pixels is linearly scaled between 0 and 1;
6. Adding contrast using histogram equalization;
7. Adding contrast using Contrast-limited adaptive histogram equalization.

We named this approach FULL in the next sections.
We are aware of the great interest of the community in learning data augmentation policies such as AutoAugment [20], however our GPUs could not perform the calculation needed in a reasonable amount of time.

3. Experimental results

In this work, we used the dataset shared by Spyridonos et al. [9,10], that contains 6010 control and 16269 AK 50 × 50 pixels regions of interest (ROIs) from clinical photographs acquired from 22 patients. The data were collected by the same physician using a Nikon D610 camera with a spatial resolution of 6016x4016 pixels. The dataset consists in 157 AK and 216 healthy skin regions. The authors in [9] extracted 50x50 images moving a sliding window across all images with an overlap of 10 pixels. The patients were volunteers recruited among those patients with at least one biopsy-proven AK A variable number of ROIs is available for each patient, ranging from 19 to 4181. Notice that the shared dataset is slightly different from the one used in Spyridonos et al. [9,10]. We did not need to segment the images because the authors in [9,10] shared a segmented dataset. In Figure 1, we show one example of image along with 5 preprocessing. We used a leave-one-out testing protocol, which consists in one fold for every patient and in every fold only the images of that patient are in the test set. Since we do not use any validation set, the rest of the images are in the training set.

We tested both modules presented in the introduction. The first one is an ensemble of SVMs trained with one handcrafted descriptor each and then combined by sum rule (results reported in Table 1). We used a radial basis function as the SVMs kernels and we set $\gamma = 1$ and $C = 1000$. These are the same parameters used in [1]. Hence, we did not overfit them to this dataset. The second one is an ensemble of CNNs combined by sum rule (results in Table 2). We used two learning rates of 0.001 and 0.0001 and trained one network for every learning rate. These two networks were evaluated as an ensemble using the sum rule. We trained the networks using stochastic gradient descent with momentum for 20 epochs with a batch size of 60. The performance indicator used in the tests is the accuracy, intended as the ratio between correctly classified images and total number of images. The method named Fusion in Table 1 is the sum rule among all the handcrafted descriptors.

Concerning handcrafted features, experimental results highlight that combining different approaches does not permit to boost the performance. The best descriptor is HOG, but its performance is lower than what can be obtained using the deep learning-based approaches.

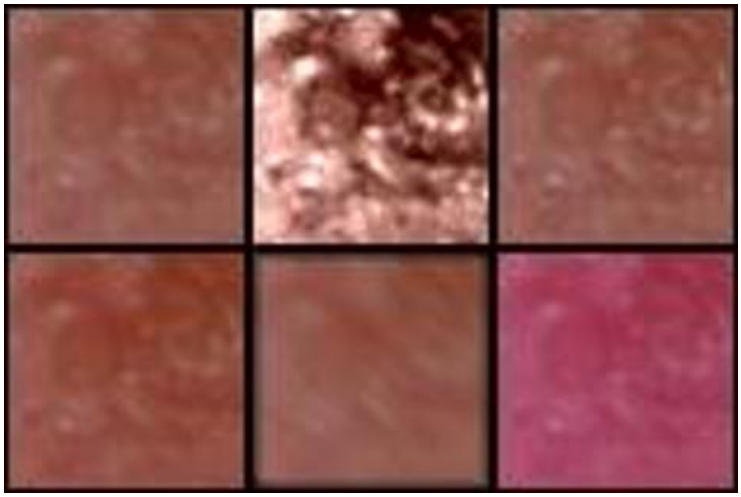

Figure 1. Preprocessing examples: original image (top-left), augmented contrast (top-center), augmented sharpness (top-right), augmented saturation (bottom-left), blurred image (bottom-center), color shifted image (bottom-right).

Table 1. Handcrafted descriptors.

Name	Parameters	Reference	Accuracy
LTP	Multiscale Uniform LTP with two (*R*,*P*) configurations: (1, 8) and (2, 16), threshold=3.	[21]	71.24
CLBP	Completed LBP with two (*R*,*P*) configurations: (1,8) and (2,16).	[22]	72.99
RIC	Multiscale Rotation Invariant Co-occurrence of Adjacent LBP with $R \in \{1, 2, 4\}$.	[23]	76.05
FBSIF	Extension of the BIF by varying the parameters of filter size (SIZE_BSIF, *size* ∈ {3, 5, 7, 9, 11}) and the threshold for binarizing (FULL_BSIF, *th* ∈ {-9, -6, -3, 0, 3, 6, 9}).	[24]	72.84
AHP	Adaptive Hybrid Pattern with quantization *level* = 5 and 2; the (*R*,*P*) configurations are (1, 8) and (2, 16).	[25]	75.11
HOG	Histogram of Oriented Gradients with 30 cells (5 by 6).	[26]	76.13
LET	Same parameters of the original paper	[27]	70.73
Fusion	Fusion of the scores using the sum rule		75.46

Considering the module based on CNN and data augmentation (Table 2), it can be seen that the best data augmentation approach is FULL. We can also see that some data augmentation techniques are more effective than others. We believe that the most effective data augmentation techniques are the ones that strongly modify the original images without changing those patterns that characterize those images. Maybe some data augmentation techniques generate images that are too similar to the original ones, while others generate images that might be too "false" to be helpful for the training.

As it was observed during the early stages of our tests, ResNet101 was tested only with the FULL augmentation method to reduce the time needed for the tests. The row named FUS_All is the sum rule among the CNNs trained using different data augmentation approaches.

Table 2. Deep learning approaches.

Data Augmentation	GoogleNet	ResNet50	ResNet101
EA	79.66	80.15	81.21
ST	74.51	77.21	78.50
L1	72.96	78.29	78.15
CSC	77.49	82.36	83.02
L2	73.16	76.31	77.10
FULL	83.08	85.17	86.17
TPS	54.74	62.10	65.08
FUS_All	80.48	82.36	83.15

We tried different fusion patterns (e.g. excluding TPS) but no ensemble outperformed the performance obtained by FULL. A very slight performance improvement is obtained by the sum rule between FULL+ResNet101 with Fusion (ensemble of handcrafted features reported in Table 1): this fusion obtains an accuracy of 86.22%. We tested the statistical validity of this improvement using the The McNemar Test for Binary Matched-Pairs Data, rejecting the null hypothesis at the 5% significance level [28].

In the Figure 2, the False Positive Rate (FPR) vs True Positive Rate (TPR) plot is reported, related to:

a) Fusion, the ensemble of handcrafted descriptors,
b) FULL+ResNet101,
c) sum rule between FULL+ResNet101 with Fusion (ensemble of handcrafted features reported in Table 1).

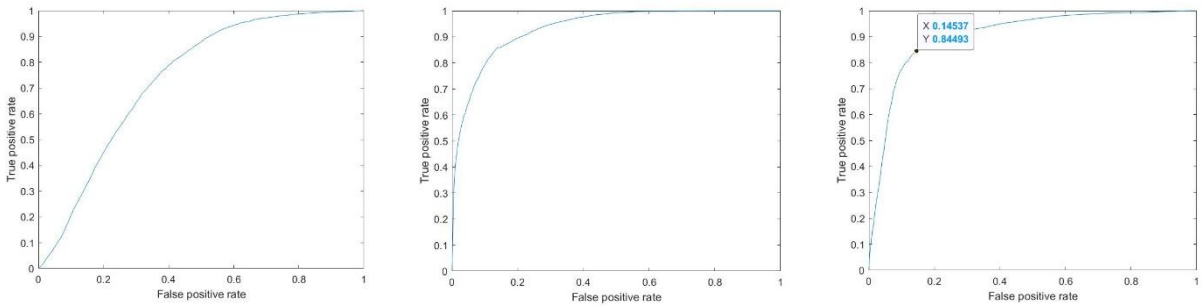

Figure 2. False positive vs True positive.

The last approach leads to a very interesting FPR of 0.145 with a TPR of 0.844, which is similar to that reported in Spyridonos et al. [9,10]. It is important to highlight, however, that we use a different testing protocol: in [10] the authors used a subset of 1000 images for each class ("by drawing random samples from all patients") for tuning the parameters of SVM.

4. Conclusion

In this paper we proposed a deep learning-based tool for the automatic classification of actinic keratosis. We created an ensemble of different convolutional networks trained with different data augmentation techniques. Besides, we created an ensemble of SVMs trained on multiple hand-crafted features extracted from the data. We proved that our ensemble can accurately classify actinic keratosis, showing a performance level that is comparable with the state of the art, but requiring a smaller dataset to achieve such results. To encourage future comparisons with the method proposed in this paper, the MATLAB code is available at https://github.com/LorisNanni.

Acknowledgments

We would like to acknowledge the support that NVIDIA provided us through the GPU Grant Program. We used a donated TitanX GPU to train the CNNs used in this work.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1 Nanni L, Paci M, Brahnam S, et al. (2017) An ensemble of visual features for Gaussians of local descriptors and non-binary coding for texture descriptors. *Expert Syst Appl* 82: 27–39.
2 Quaedvlieg PJF, Tirsi E, Thissen MRTM, et al. (2006) Actinic keratosis: how to differentiate the good from the bad ones? *Eur J Dermatol* 16: 335–339.
3 Schwartz RA (1997) The Actinic Keratosis A Perspective and Update. *Dermatol Surg* 23: 1009–1019.
4 Rigel DS, Gold LFS (2013) The importance of early diagnosis and treatment of actinic keratosis. *J Am Acad Dermatol* 68: S20–S27.
5 Wassef C, Rao BK (2013) Uses of non-invasive imaging in the diagnosis of skin cancer: An overview of the currently available modalities. *Int J Dermatol* 52: 1481–1489.
6 Ballerini L, Fisher RB, Aldridge B, et al. (2013) A Color and Texture Based Hierarchical K-NN Approach to the Classification of Non-melanoma Skin Lesions. In: M.E. Celebi, G. Schaefer (Eds.) *Color Medical Image Analysis*. Springer, Dordrecht. 6: 63–86.
7 Hames SC, Sinnya S, Tan JM, et al. (2015) Automated Detection of Actinic Keratoses in Clinical Photographs. *PLoS One* 10: e0112447.
8 Kawahara J, BenTaieb A, Hamarneh G (2016) Deep features to classify skin lesions. *2016 IEEE 13th International Symposium on Biomedical Imaging* 2016: 1397–1400.
9 Spyridonos P, Gaitanis G, Likas A, et al. (2017) Automatic discrimination of actinic keratoses from clinical photographs. *Comput Biol Med* 88: 50–59.
10 Spyridonos P, Gaitanis G, Likas A, et al. (2019) Late fusion of deep and shallow features to improve discrimination of actinic keratosis from normal skin using clinical photography. *Skin Res Technol* 25: 538–543.
11 (2016) Clinical photography. *Journal of Oral Biology and Craniofacial Research* 6: 171.

12 Pietikäinen M, Hadid A, Zhao G, et al. (2011) Local Binary Patterns for Still Images. In: *Computer Vision Using Local Binary Patterns*. Computational Imaging and Vision. Springer, London. 40: 13–47.
13 Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Commun ACM* 60: 84–90.
14 Badrinarayanan V, Kendall A, Cipolla R (2017) SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE T Pattern Anal* 39: 2481–2495.
15 Ren S, He K, Girshick RB, et al. (2017) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE T Pattern Anal* 39: 1137–1149.
16 Szegedy C, Liu W, Jia Y, et al. (2015) Going deeper with convolutions. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1–9.
17 He K, Zhang X, Ren S, et al. (2016) Deep Residual Learning for Image Recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition* 770–778.
18 Russakovsky O, Deng J, Su H, et al. (2015) Imagenet large scale visual recognition challenge. *Int J Comput Vis* 115: 211–252.
19 Bookstein FL (1989) Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE T Pattern Anal* 11: 567–585.
20 Cubuk ED, Zoph B, Mane D, et al. (2019) AutoAugment: Learning Augmentation Strategies From Data. *Proceedings of the IEEE Conference on Computer Vision Pattern Recogni*tion 113–123.
21 Tan X, Triggs W (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. *IEEE T Image Process* 19: 1635–1650.
22 Guo Z, Zhang L, Zhang D (2010) A completed modeling of local binary pattern operator for texture classification. *IEEE T Image Process* 19: 1657–1663.
23 Nosaka R, Fukui K (2014) HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns. *Pattern Recogn* 47: 2428–2436.
24 Nanni L, Paci M, Santos F, et al. (2016) Review on texture descriptors for image classification. *Comput Vis Simul Methods Appl Technol*.
25 Zhu Z, You X, Chen CLP, et al. (2015) An adaptive hybrid pattern for noise-robust texture analysis. *Pattern Recogn* 48: 2592–2608.
26 Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection.
27 Song T, Li H, Meng F, et al. (2018) LETRIST: Locally encoded transform feature histogram for rotation-invariant texture classification. *IEEE T Circ Syst Vid* 28: 1565–1579.
28 Fagerland MW, Lydersen S, Laake P (2013) The McNemar test for binary matched-pairs data: mid-p and asymptotic are better than exact conditional. *BMC Med Res Methodol* 13: 91.

AIMS Press