Exploiting Convolutional Neural Networks and preprocessing techniques for HEp-2 cell classification in immunofluorescence images

Larissa Ferreira Rodrigues, Murilo Coelho Naldi, João Fernando Mari
Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Viçosa - UFV
Caixa Postal 22 - 38.810-000 - Rio Paranaíba - MG - Brasil
Email: {larissa.f.rodrigues, murilocn, joaof.mari}@ufv.br

***Abstract*—Autoimmune diseases are the third cause of mortality in the world. The identification of anti-nuclear antibody (ANA) via Immunofluorescence (IIF) test in human epithelial type-2 cells (HEp-2) is a conventional method to support the diagnosis of such diseases. In the present work, three popular Convolutional Neural Networks (CNNs) are evaluated for this task: LeNet-5, AlexNet, and GoogLeNet. We also assess the impact of six different pre-processing strategies on the performance of these CNNs. Additionally, data augmentation based on the rotation of the training set images after the pre-processing strategies was evaluated. Our work is the first to consider AlexNet and GoogLeNet models for the proposed analysis and classification of HEp-2 cells images, besides the LeNet-5. Experimental results allow to conclude that neither pre-processing strategies were essential to improve accuracy values of the CNNs. However, when data augmentation is considered, contrast enhancement followed by data centralization is significant in order to achieve good results. Additionally, our results were compared with results from other state-of-art papers. Our best results were achieved by GoogLeNet architecture trained with images with no pre-processing and no data augmentation, resulting in 98.17% of accuracy, which outperforms the results presented in other works in literature.**

***Keywords*—Convolutional neural networks; HEp-2 cells; staining patterns classification; LeNet-5; AlexNet; GoogLeNet; pre-processing; data augmentation.**

I. Introduction

In the last decades, there was a significant increase in the number of autoimmune diseases cases. They are the third most common cause of mortality, after cardiovascular diseases and cancer [1]. Autoimmune diseases are a condition in which the immunological system attacks and destroys health cells and tissues by mistake [2]. Most of autoimmune diseases have similar symptoms and difficult diagnosis. There is no cure for autoimmune diseases and the treatment consists of alleviating their symptoms by means of anti-inflammatory and immunosuppressive drugs [3].

The visual analysis of the staining patterns in immunofluorescence (IIF) images taken from human epithelial cells (HEp-2) is one of the available procedures for identification of autoimmune diseases. Manual analyses of HEp-2 cells by IIF have been done along years, but it is subjective and time consuming [4] [5] [6]. The need for automated and standard methods is known and persecuted since long time, and the development of computer aided diagnosis systems based on image processing and machine learning techniques are a fundamental part of it [7] [8].

Digital image processing and machine learning are the bases of a large number of computer aided diagnosis applications. Generally, solutions based on computerized image analysis have lower financial cost and the collected data may be easily shared and processed in other countries [9] [10], which makes such techniques attractive for countries in development [11].

The adoption of Deep Learning Techniques, Convolutional Neural Networks (CNNs) in particular, have been demonstrating good results in general visual recognition tasks, which motivates their broad application on several research fields [12] [13]. The CNNs are based in a multi-stage image processing in order to extract high level hierarchical representations of the data. Recently, Gao et al. [14] presented a method to classify HEp-2 cells based on a deep CNN inspired in the classical LeNet-5 [15]. This method was the first advance towards the state-of-the-art techniques for HEp-2 classification with CNN.

Bearing in mind the increasing demand for automatic methods of HEp-2 cells classification and the CNNs promising results, the main objective of this work is to study different approaches based on CNNs to automatically classify HEp-2 cells in microscopy images. We will compare the performance of three CNNs: LeNet-5 [15], AlexNet [12] and GoogLeNet [16], with six different pre-processing strategies. These strategies are capable of revealing several autoimmune diseases like, but not restricted to: systemic lupus erythematosus, rheumatoid arthritis, multiple sclerosis, autoimmune hepatitis and diabetes [17]. By comparing the results we intend to: a) assess the impact of different CNNs architectures in terms of accuracy and training time; b) study the influence of different pre-processing approaches on each of the CNNs architectures. To the best of our knowledge, this is the first work to apply AlexNet and GoogLeNet models, beside the LeNet-5, in the classification of HEp-2 cells images. We believe that our method can be used to help health agents to identify and manage such diseases [18].

This work is organized as: Section II presents the related work. Section III describes the material and methods. Section IV shows and discusses the results. The conclusions and future work are presented in the Section V.

II. Related Work

Perner et al. [19] was one of the first proposals of a system to automatically classify HEp-2 cells. A number of texture features are extracted from twelve quantized versions of the original images and classified with decision trees. This approach was adopted by Sack et al. [20] in order to identify positive fluorescence in a set of immunofluorescence patterns.

After that, several works were carried out with the objective of classifying HEp-2 cells with hand-crafted feature extraction. Nosaka and Fukui [21] proposed a method based on rotation invariant co-occurrence among adjacent local binary patterns (RIC-LBP) features and support vector machines (SVM) to classify HEp-2 cell images. Stoklasa et al. [22] classified HEp-2 cells using a K-NN classifier with Haralick features, Local Binary Patterns, SIFT, surface description, and a granulometry-based descriptor. Shen et al. [23] proposed a bag of visual words model (BoVW) of intensity order pooling based features with SVM classification. Manivannan et al. [8] utilized several local descriptors and incorporated multiresolution local high-order statistical features. Gragnaniello et al. [24] uses the Scale-Invariant Descriptor (SID) and the features are encoded using the BoVW model with soft assignment through Gaussian weights, to classify HEp-2 cell images uses SVM. Kastaniotis et al. [25] proposed feature encoding process by incorporating a Sparse Coding (SC), used Vector of Hierarchically Aggregated Residuals (VHAR), SIFT descriptor and different classifiers, like SVM and K-NN. Ensafi et al. [26] used SIFT and SURF features with a BoVW model to classify HEp-2 cells with a SVM.

Along with the rise of more powerful computational resources (i.e., mass parallel and distributed computing), improved optimization techniques and increasing datasets, deep CNNs models significantly overcame models based on hand-crafted (i.e., user based) feature extraction for general visual recognition tasks. As presented in [7], Malon et al. adopted a CNN to classify HEp-2 cells after pre-processing, i.e., constrast enhancement was applied over the images, standardized for 100×100 pixels with the green component only. Differently from [7], our work uses other pre-processing methods and other CNN architectures, and we achieves better results.

Gao et al. [14] propose a method for automatic classification of HEp-2 with a CNN that shares the basic architecture of LeNet-5 [15]. Differently from previously cited work, the CNN was able to learn features automatically during the training process. Experiments considering one instance of each image achieved an accuracy of 88.58%, improved to 96.76% with the increase of the data set with multiple copies of its images after rotations. On the other hand, the present work considered, in addition to the LeNet-5 architecture, the AlexNet [12] and GoogLeNet [16] for six different pre-processing strategies. Although the former architecture is considered computationally simpler, the total computational cost for training is reduced in comparison with [14], as there is no need to increase the data set during pre-processing, which results in a smaller (and faster) dataset processing for results with quality equivalent or superior to those obtained in [14].

III. Material and Methods

As presented before, the main objective of this work is to evaluate the performance of different types of CNNs in classifying microscopy images of HEp-2 cells in order to assess how different pre-processing strategies influence on their performance. The Fig. 1 illustrates the steps of the proposed method.

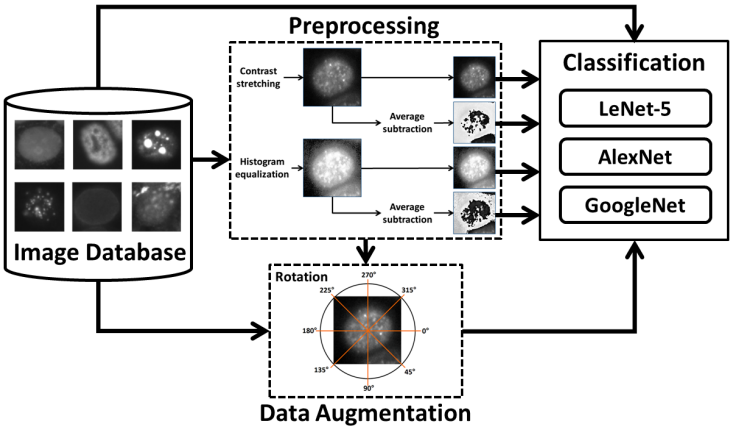

Fig. 1. Steps of proposed method, including the pre-processing, data augmentation, and training/classification step.

A. Image Dataset

We use a set of 13,596 images of HEp-2 cells; each image has one centered cell in evidence. This dataset is available for the "Contest on Performance Evaluation on Indirect Immunofluorescence Image Analysis Systems" hosted by ICPR 2014 [1] [27]. The images are classified in six different classes: centromere, golgi, homogeneous, nucleolar, nuclear membrane, and speckled. Fig. 2 show one image from each class.

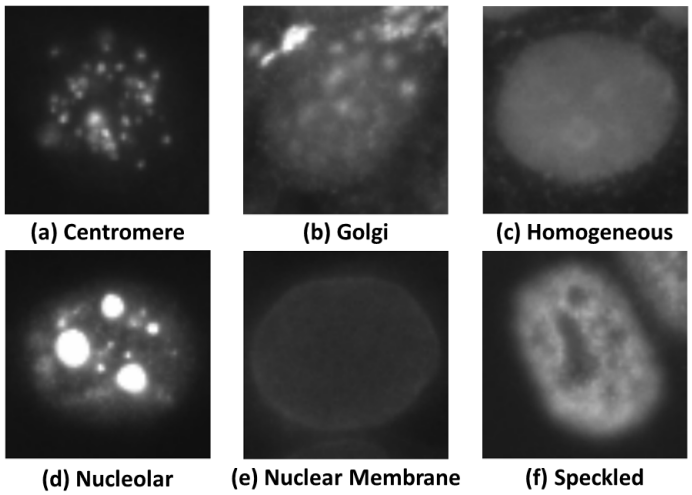

Fig. 2. Examples of images instances for each class of data set ICPR 2014.

[1] Available in: http://nerone.diem.unisa.it/hep2-benchmarking/dbtools/

B. Pre-processing

Before applying the pre-processing methods we resize all images to 78 x 78 pixels, which is the image size used in Gao et al. work [14]. These images are used to train and test the LeNet-5 and AlexNet. The GoogLeNet demands images with a minimum of 256 x 256 pixels, thus we resized the original dataset to this setting in order to enable the use of GoogLeNet.

We propose a number of image pre-processing strategies based on combinations of contrast improvement and image normalization in order to assess their impact in the classification results of each CNN model studied here. These strategies will be following described:

1) Contrast stretching: the contrast stretching is a simple linear transformation function which maps the minimum intensities in an image to 0 and the maximum values to 1, while the intermediary values are linearly mapped into interval [0,1] from their original values [28].

2) Histogram equalization: the histogram equalization builds a piecewise linear function based on the normalized image histogram, which implies an image with a uniform histogram [28].

3) Average image subtraction: we compute the average image of the training set and subtract this average image from each image in the dataset. The result is a transformed training set with zero mean and, if the training set is representative, the a dataset has its mean close to zero [29].

The combination of the described pre-processing techniques resulted into six pre-processing strategies, illustrated in Fig. 3. The leftmost image is original from the data set, which is defined as strategy (a) (i.e., no pre-processing). After that, in (b), there is a example of average subtraction strategy. Strategy (c) is contrast stretching solely and (d) is contrast stretching with average subtraction. Last, but not least, strategies (e) and (f) are histogram equalization with and without average subtraction, respectively.

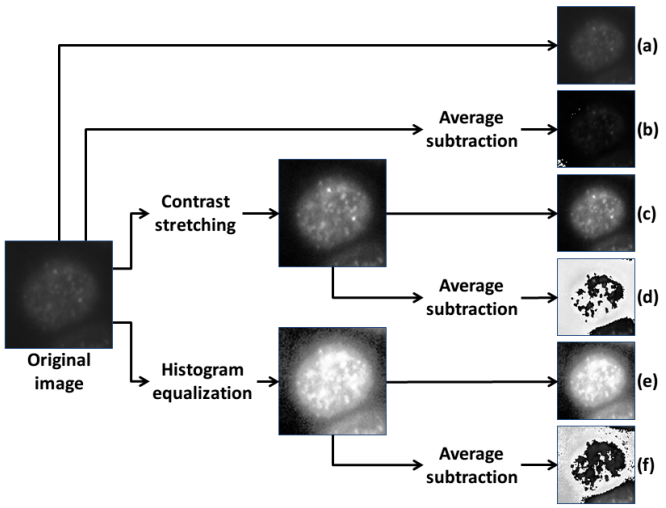

Fig. 3. Contrast improvement of HEp-2 cell image: (a) original image, (b) average subtraction, (c) contrast stretching, (d) contrast stretching with average subtraction, (e) histogram equalization and, (f) histogram equalization with average subtraction.

The average images for the ICPR 2014 data set are presented in Fig. 4. In Fig. 4(a) is presented the average image for the original set, Fig. 4(b) the mean image for the preprocessed data set with histogram equalization, and Fig. 4(c) has the average image for the preprocessed data set with contrast stretching.

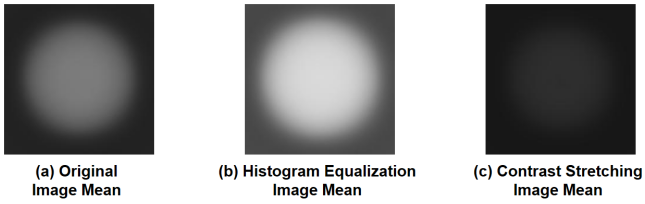

Fig. 4. Mean images of ICPR 2014 data set: (A) original, (B) after histogram equalization, and (C) after contrast stretching.

4) Data augmentation: Data augmentation is an strategies used to artificially increase the training data without introducing labeling costs [12]. Each image is rotated 360° around its center with steps of 45°, increasing the training set in 8 times.

C. Convolutional Neural Networks (CNNs)

CNNs are one of four categories of deep learning methods, along with the restricted Boltzman machines (RBMs), autoencoders, and sparse coding. A CNN consists basically of three types of neural layers: a) convolutional layers; b) pooling layers; and c) full connected layers [30]. These layers are structured in hierarchical architectures, with convolutional layers alternated with pooling layers and a fully-connected [15]. The details of these layers are following described.

Convolutional layers: the convolutional layers performs the convolution of the feature maps in the previous layer with a set of filters, in which each filter are intended to detect one kind of feature. A convolutional layer C^l is composed by a bank of K filters, W_k^l and K bias b_k^l, for which $k \in [1, ..., K]$. The size of each filter W_k^l is $(F \times D)$ in which F is the spatial stent of the filter and D is an hyper-parameter from the input volume M^{l-1}. The input volume, M^{l-1}, is a stack of D feature maps resulting from the previous layer l-1, and have size $(H \times W \times D)$. To generate the output volume M^l, the convolutional layer C^l convolve each slice $d \in [1, ..., D]$ of the input volume M^{l-1} with each of the $k \in [1, ..., K]$ filters W_k^l. The resulting convolutions from each filter are summed-up and added with the bias. The resulting K 2D feature maps are stacked in a output volume M^l. Equation 1 shows the formation of one M_k^l slice of the output volume M^l.

$$M_k^l = \sum_{d=1}^{D} M_d^{l-1} * W_d^l + b_k^l \tag{1}$$

After the convolution layer, an element wise activation function is applied. Generally a REctified Linear Unit (RELU), $f(M) = \max(0, M)$ is preferred over other common activation functions, such as sigmoid and hyperbolic tangent, because it leads to a much faster training in very deep architectures [31]. The Fig. 5 illustrates the operation in the convolution layer.

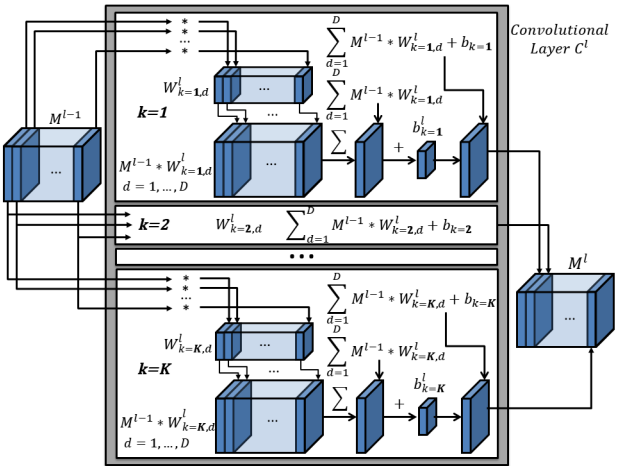

Fig. 5. Illustration of the structure of a typical convolutional layer.

Pooling layers: the pooling layers usually follow a convolution layer and are responsible to reduce the size of the feature maps in accordance to some criteria, such as maximum or average pooling in accordance with the size of the pooling region. A pooling layers with pooling size 2×2 reduces a path with 2×2 pixels to one pixel, choosing the maximum pixel or the average pixel for max-pooling or average-pooling respectively [32]. Maximum polling results in faster convergence and better generalization, in most cases [32]. The Fig. 6 illustrates of the effect of a pooling layer with size 2×2 with maximum and average criteria, respectively.

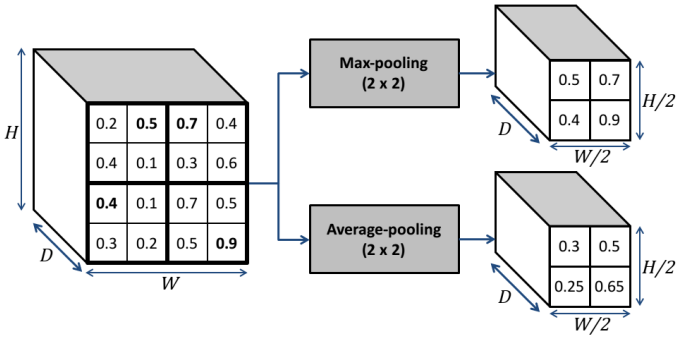

Fig. 6. Illustration of the computations in the pooling layer.

Fully connected layers: in the last layers of a CNN, the 2D feature maps are converted to a 1D feature vector. This feature vector is the input of a number of fully-connected layers, performing traditional inner products computation. The last layer, generally, consists of a softmax classifier. The softmax layer h has n neurons, in which n is the number of classes in the dataset, and the output of each neuron $\widetilde{y_j}$ is the probability that the input image has to belong to the class j [14].

The fully-connected layers contains about 90% of the total parameters in a CNN and are responsible for most of the training computational cost [30]. Bearing that in mind, GoogLeNet, one of the architectures considered for this work, adopts an strategy to reduce the number of connections in the full connected layers [16].

The training of a CNN is similar to a conventional feed-forward neural network. Each sample (image) is forwarded through the layers (weights and biases) until a loss function can be computed at the top layer. Then, the loss is back-propagated through the layers adjusting the weights in accordance to the Stochastic Gradient Descent (SGD) method. The process is repeated until a pre-defined number of training epochs is reached or the CNN "converges" (i.e., is capable of classifying all training data correctly) and is considered trained [33] [30] [34]. We used a learning rate of 0.01 for all experiments [35].

The CNNs can be either trained from scratch or fine-tuned over pre-trained models [36]. In this works we adopt the strategy to train the network from scratch, with weights randomly initialized.

D. CNN architectures

For this work, we choose to test three popular CNN architectures: LeNet-5, AlexNet and GoogLeNet. All implementations were performed using the NVIDIA DIGITS [37] framework running over Caffe Framework [38].

1) LeNet-5: Develop by Yan LeCun et al. [15], it was initially developed for character recognition. LeNet-5 is one of the first well succeed applications of CNNS. LeNet-5 is composed by three convolution layers, two pooling layers, one fully-connected and a softmax layer [15], [14].

2) AlexNet: Proposed by Krizhevsky et al. [12], the AlexNet won the ILSVC 2012 competition [39]. AlexNet is composed by five convolutional layers, three pooling layers, two fully-connected and a softmax layer [36]. AlexNet uses dropout connections to reduce overfitting in the fully-connected layers [40] and ReLU as activation function in the convolutional and fully-connected layers.

3) GoogLeNet: The GoogLeNet architecture is a particular instance of the Goggle Inception architecture. Inception architecture includes Inception modules intercalated with the convolution, polling and full connected layers. GoogLeNet is a very deep architecture with 22 layers. The Inception modules combine convolutional layers with sizes 1×1, 3×3 and 5×5 with 3×3 max-pooling layers. As AlexNet, GoogLeNet apply dropout regularization in the fully-connected layers and ReLU activation functions in the convolution layers [16].

E. Validation

In order to evaluate the CNNs and the pre-processing strategies proposed in this work we used the Mean Class Accuracy (MCA) [41], which consists in the average of the per class accuracies, as defined in Eq. 2:

$$MCA = \frac{1}{n}\sum_{k=1}^{n} CCR_k, \tag{2}$$

in which CCR_k is the accuracy classification for the k-th class and n is the number of classes. This is the same criterion used to evaluate the HEp-2 ICPR 2014 competition, from which we obtained the considered data set.

IV. Results and Discussion

The 13,596 images of the dataset were randomly partitioned among a training set, a validation set, and a testing set with the proportions 64% (8,701 images), 16% (2,175 images), and 20% (2,720 images), respectively. The same partitioning strategy was adopted by Gao et al [14] and chosen here for a fair comparison. The resulted sets are used in all experiments of this work.

After partitioning the dataset, we resized all images in order to adapt each image for the proper input size of the CNN, as described in Section III-B. The result, a partitioned data set with resized images, was preprocessed using the six different strategies (also described in Section III-B): a) no processing applied; b) subtraction of the average image; c) contrast stretching; d) contrast stretching and subtraction of the average image; e) histogram equalization and subtraction of the average image.

The results of the experiments considering all CNNs with every pre-processing strategies is presented in Table I (no data augmentation) and Table III (with data augmentation) in terms of MCA. Each line corresponds to one CNN and each column corresponds to one of the six prepossessing strategies. We reinforce that all CNN were trained from scratch, i.e, no fine-tuning strategies was used, as described at Section III-D. The time used for training each CNN with all prepossessing strategies are showed in Table II and Table IV for experiments with no data augmentation and with data augmentation, respectively.

Additionally, the computational running time for each CNN and pre-processing strategy is presented in Table II. All experiments were performed in a machine with processor Intel i5 2.67 GHz, 16 GB of memory RAM, and a GPU GeForce GTX Titan X with 12 GB memory, under a 8.0 CUDA version. Ubuntu version 16.04.2 LTS was used as an operating system.

To illustrate some aspects of the classification problem investigated in this work, Table V shows the confusion matrices for each CNN, i.e., LeNet-5, AlexNet and GoogLeNet. Due to space limitation of this publication, only the experiments without pre-processing and data augmentation are presented. As an example, one image of each cell type (class) is presented in Fig. 7, considering the confusion matrices in Table V,for LeNet-5 (Fig. 7 (a)), AlexNet (Fig. 7 (b)), and GoogLeNet (Fig. 7 (c)). Additionally, the evolution of the loss values for the training and validation sets and of the accuracy values for the validation set are showed in Fig. 8, considering the same experiment.

A. The impact of pre-processing

Based on the results presented in Table I and III, our investigation allows to conclude that the contrast stretching with average subtraction is a good choice for both the original and the augmented data sets. However, it can be observed from the data in Table I, that the original dataset, with no preprocessed and no augmented images, have one of the best results for all CNN models. This may be surprising, considering that several works in literature emphasize the importance of pre-processing HEp-2 cells before classification [21] [42] [43].

Additionally, data centralization alone, done by the subtraction of the average image, had small impact in the accuracy of the results when compared with the original data. However, it is possible to notice that the subtraction of the average image had a great positive impact over strategies submitted to contrast stretching and histogram equalization for LeNet-5 and with contrast stretching for AlexNet. GoogLeNet showed to be very robust, i.e., pre-processing strategies had little positive impact over its results of the experiments.

B. The impact of data augmentation

The datasets generated from the applying of the pre-processing strategies were submitted to a data augmentation procedure, described in Section III-B4.

As showed in the first and second columns of Table II, when applied over the original images (no pre-processing) and original images followed by average image subtraction, the data augmentation procedure made the results much worse for all CNNs, up to 76.4% worse. The data augmentation slightly improved the accuracies for LeNet-5 with contrast stretching and histogram equalization, only when followed by average image subtraction. When compared to the no augmented datasets, the improvements were 5% and 4.5% for histogram equalization + average image subtraction and contrast stretching + average image subtraction, respectively. AlexNet and GoogLeNet had their results slightly improved by the data augmentation over the datasets pre-processed with contrast stretching and histogram equalization, both with and without average image subtraction. For AlexNet the minimum improvement given by data augmentation was 3.2% for contrast stretching and the maximum improvement was 27.7% for histogram equalization, both compared with the same pre-processing strategy without data augmentation. For GoogLeNet, the improvement of data augmentation was very low, varying between 1.1% and 4.8% for contrast stretching + average image subtraction and histogram equalization, respectively.

Another important aspect of the data augmentation procedure is the impact in the training time. The computational time for training a CNNs with the dataset augmented eight times (45° rotations), as showed in Table IV, increases up to seven times, approximately, when comparing the training time of datasets with and without augmentation, as presented in Table III. This lead us to believe that the increase in the computational cost to train a CNN with data augmentation may outweighs the improvements in terms of accuracy, that were up to 5% for LeNet-5, 27.7% for AlexNet, and 4.8% for GoogLeNet. Thus, data augmentation may be an unnecessary burden for HEp-2 cells classification or different data augmentation strategies should be tested in order to achieve some improvements in therms of accuray.

C. Comparison with related works

We compared our best result, obtained by the GoogLeNet with no pre-processing and no data augmentation with other state-of-art works in the literature. All values presented in Table VI corresponds to the best results reported in the original papers. It is important to stress that all works considered the same HEp-2 dataset from ICPR 2014. We show that the

TABLE I. MEAN CLASS ACCURACY FOR EACH CNN MODEL AND PRE-PROCESSING STRATEGY

	Method and Accuracy (%)					
CNN	**Original**	**Original with average subtraction**	**Contrast stretching**	**Contrast stretching with average subtraction**	**Equalization Histogram**	**Equalization Histogram with average subtraction**
LeNet	88.24	88.06	20.82	85.80	20.82	82.91
AlexNet	91.14	93.80	89.99	91.00	61.13	88.98
GoogLeNet	98.17	97.98	93.43	94.86	93.07	95.00

TABLE II. TRAINING TIME - PRE-PROCESSING WITH NO DATA AUGMENTATION

	Training time (in minutes)					
CNN	**Original**	**Original with average subtraction**	**Contrast stretching**	**Contrast stretching with average subtraction**	**Equalization Histogram**	**Equalization Histogram with average subtraction**
LeNet	3.17	3.06	3.06	3.03	3.07	3.04
AlexNet	4.1	4.04	4.02	3.52	4.04	3.58
GoogLeNet	63.5	63.38	63.37	63.35	63.44	63.39

TABLE III. MEAN CLASS ACCURACY FOR EACH CNN MODEL AND PRE-PROCESSING STRATEGY OVER AUGMENTED DATA SET

	Method and Accuracy (%) considering data augmentation					
CNN	**Original**	**Original with average subtraction**	**Contrast stretching**	**Contrast stretching with average subtraction**	**Equalization Histogram**	**Equalization Histogram with average subtraction**
LeNet	20.82	53.77	20.82	90.17	20.82	86.63
AlexNet	52.26	47.89	92.88	96.01	78.08	93.20
GoogLeNet	77.30	57.86	96.33	95.91	97.52	97.62

TABLE IV. TRAINING TIME - PRE-PROCESSING WITH DATA AUGMENTATION

	Training time (in minutes) pre-processing with data augmentation					
CNN	**Original**	**Original with average subtraction**	**Contrast stretching**	**Contrast stretching with average subtraction**	**Equalization Histogram**	**Equalization Histogram with average subtraction**
LeNet	22.07	22.53	22.25	22.06	22.10	22.20
AlexNet	22.45	23.00	22.35	22.33	22.48	22.53
GoogLeNet	420.49	420.49	420.49	420.49	420.49	420.49

TABLE V. CONFUSION MATRIX FOR EACH CNN MODEL WITHOUT PRE-PROCESSING OR DATA AUGMENTATION

	LeNet-5						**AlexNet**						**GoogLeNet**					
	Cent.	Golgi	Homo.	Nucl.	NuMem.	Spec.	Cent.	Golgi	Homo.	Nucl.	NuMem.	Spec.	Cent.	Golgi	Homo.	Nucl.	NuMem.	Spec.
Cent.	**90.21**	0.00	0.00	2.97	0.23	6.61	**87.93**	0.00	0.23	2.74	0.23	8.89	**98.18**	0.00	0.00	1.37	0.00	0.46
Golgi	3.45	**71.56**	4.32	11.21	7.76	1.73	0.87	**84.49**	0.87	6.90	4.32	2.59	0.87	**91.38**	2.59	3.45	0.87	0.87
Homo.	0.26	0.00	**90.48**	0.76	1.01	7.52	0.00	0.51	**92.49**	0.51	1.51	5.02	0.00	0.00	**94.49**	0.00	0.00	5.52
Nucl.	3.13	0.73	0.73	**88.47**	0.97	6.01	1.93	0.73	0.00	**95.20**	1.45	0.73	0.00	0.25	0.00	**97.12**	0.00	2.65
NuMem.	0.00	1.99	2.27	0.29	**95.47**	0.00	0.00	1.70	2.55	0.00	**95.47**	0.29	0.00	0.57	5.67	0.29	**90.09**	3.40
Spec.	3.76	0.00	8.39	4.42	0.67	**82.79**	3.76	0.23	5.52	2.87	0.23	**87.42**	2.43	0.23	1.99	0.67	0.00	**94.71**

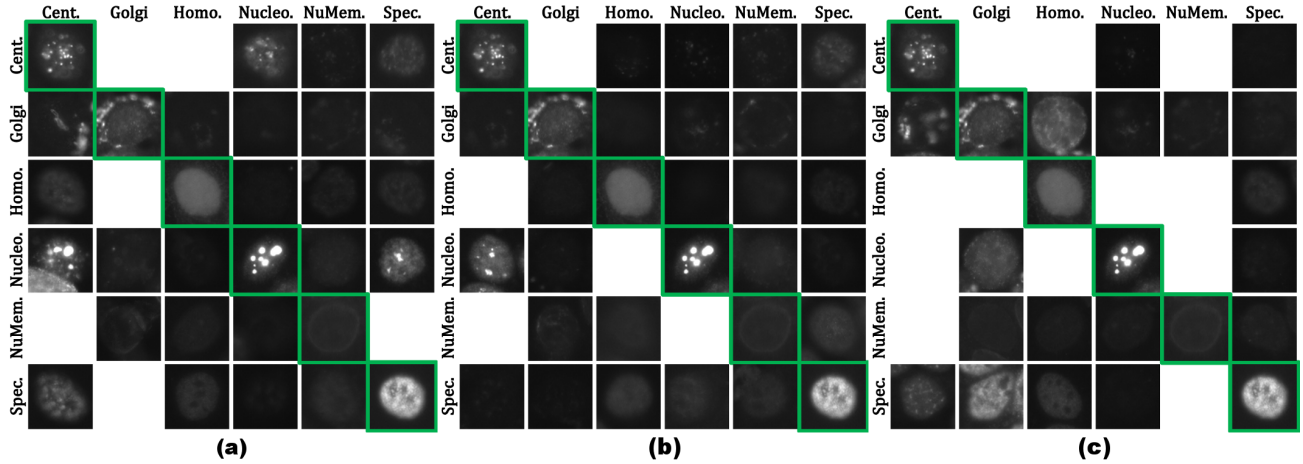

Fig. 7. Examples of correctly classified (main diagonal) and misclassified images in accordance with the confusion matrices. (a) LeNet-5; (b) AlexNet; and (c) GoogLeNet.

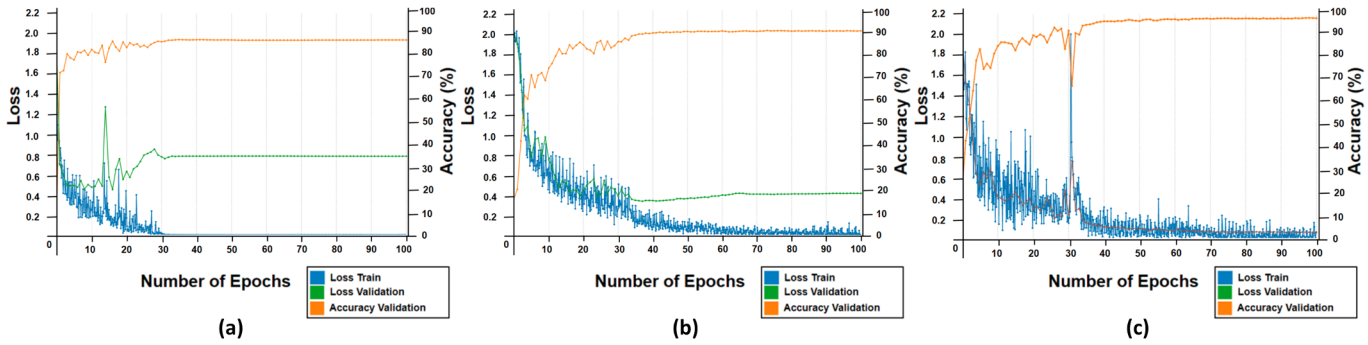

Fig. 8. Charts showing the evolution of loss values for training and validation sets and of accuracy values for the validation set. (a) LeNet-5; (b) AlexNet; and (c) GoogLeNet.

results obtained by our experiments outperforms (in terms of accuracy) the results published by the compared papers.

TABLE VI. HIGHEST ACCURACY OF THE CLASSIFICATION METHODS OVER DATASET HEP-2 FROM ICPR 2014 FOR EACH PAPER.

Method	**Accuracy (%)**
Kastaniotis et al. (2017) [25]	82.30
Gragnaniello et al. (2016) [24]	82.60
Manivannan et al. (2016) [8]	87.10
Gao et al. (2017) [14]	88.58
Gao et al. (2017) (with data augmentation) [14]	96.76
Our work (2017)	**98.17**

V. CONCLUSION

In this paper considers the automatic classification of HEp-2 cell images using three different CNNs and six pre-processing strategies based on contrast improvements, data centralization, and data augmentation. The main contribution of this work is the evaluation of three CNN models considering six different pre-processing strategies with data augmentation. Each CNN models behaved differently for each pre-processing strategy. Thus, considering our experiments, it was possible to observe which is the best pre-processing strategies for each network architecture in terms of accuracy and also training time, considering several scenarios.

The results from the performed experiments allow to conclude that the considered some of the CNN architectures may be trained without any kind of prepossessing and, still, score good results for HEp-2 image cells, which is reflected by reasonable high accuracy values. Our experimental results showed that one of our studied approaches achieved accuracy values up to 98.17%, with outperforms other state-of-art results published in literature. This result was achieved with GoogLeNet with no pre-processing strategies.

As a matter of fact, it is interesting to point it out that GoogLeNet architecture was robust to data centralization, i.e., the subtraction of average image values has few or no effect over its accuracy. As we pointed out in the paper, it should be considered that data augmentation of HEp-2 image cells should impact on the computational training time over seven folds.

However, if the training step involves data augmentation procedures, contrast stretching or histogram equalization followed by average image subtraction is advised, as these techniques showed to achieve considerable accuracy values. Moreover, the data augmentation showed to have significant impact over AlexNet only when preprocessed with histogram equalization.

As future works we intend to study another models of CNNs, such as Clarifai [44], SPP-Net [45], and VGG [46] to the problem of classifying HEp-2 cells images. It would be interesting to test the methods with other HEp-2 image datasets or datasets with other types of cells, and to test other strategies for data augmentation. Another potential future work is the study of strategies to optimize the training of the CNN models in terms of accuracy and computational cost, in order to improve the results from ensembles of different CNNs or different configurations of the same CNN architeture.

ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the TITAN Xp GPU used for this research. We would like to thanks CAPES and FAPEMIG for the financial support.

REFERENCES

[1] Jean-François Bach. The effect of infections on susceptibility to autoimmune and allergic diseases. *New England Journal of Medicine*, 347(12):911–920, 2002.

[2] I.R. Mackay and N.R. Rose. *The Autoimmune Diseases*. Elsevier Science, 2013.

[3] Lucienne Chatenoud. Precision medicine for autoimmune disease. *Nat Biotech*, 34(9):930–932, Sep 2016. News and Views.

[4] Renato Tozzoli, Chiara Bonaguri, Alessandra Melegari, Antonio Antico, Danila Bassetti, and Nicola Bizzaro. Current state of diagnostic technologies in the autoimmunology laboratory. *Clinical Chemistry and Laboratory Medicine*, 51(1):129–138, 2013.

[5] Marvin J Fritzler. The antinuclear antibody test: last or lasting gasp? *Arthritis & Rheumatism*, 63(1):19–22, 2011.

[6] Daniel H Solomon, Arthur J Kavanaugh, and Peter H Schur. Evidence-based guidelines for the use of immunologic tests: antinuclear antibody testing. *Arthritis Care & Research*, 47(4):434–444, 2002.

[7] Pasquale Foggia, Gennaro Percannella, Alessia Saggese, and Mario Vento. Pattern recognition in stained hep-2 cells: Where are we now? *Pattern Recognition*, 47(7):2305 – 2314, 2014.

[8] Siyamalan Manivannan, Wenqi Li, Shazia Akbar, Ruixuan Wang, Jianguo Zhang, and Stephen J. McKenna. An automated pattern recognition system for classifying indirect immunofluorescence images of hep-2 cells and specimens. *Pattern Recognition*, 51:12 – 26, 2016.

[9] Kunio Doi. Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, 31(4–5):198 – 211, 2007. Computer-aided Diagnosis (CAD) and Image-guided Decision Support.

[10] Geoff Dougherty. *Digital image processing for medical applications*. 2009.

[11] Qiang Wu, Fatima Merchant, and Kenneth Castleman. *Microscope image processing*. Academic press, 2010.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[13] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. Cnn features off-the-shelf: An astounding baseline for recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2014.

[14] Z. Gao, L. Wang, L. Zhou, and J. Zhang. Hep-2 cell image classification with deep convolutional neural networks. *IEEE Journal of Biomedical and Health Informatics*, 21(2):416–428, March 2017.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[16] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[17] Karl Egerer, Dirk Roggenbuck, Rico Hiemann, Max-Georg Weyer, Thomas Büttner, Boris Radau, Rosemarie Krause, Barbara Lehmann, Eugen Feist, and Gerd-Rüdiger Burmester. Automated evaluation of autoantibodies on human epithelial-2 cells as an approach to standardize cell-based immunofluorescence tests. *Arthritis Research & Therapy*, 12(2):1–9, 2010.

[18] J.M. Polak and S. Van Noorden. *Immunocytochemistry: Practical Applications in Pathology and Biology*. Elsevier Science, 2014.

[19] Petra Perner, Horst Perner, and Bernd Müller. Mining knowledge for hep-2 cell image classification. *Artificial Intelligence in Medicine*, 26(1–2):161 – 173, 2002. Medical Data Mining and Knowledge Discovery.

[20] Ulrich Sack, Stephan Knoechner, Holger Warschkau, Ullrich Pigla, Frank Emmrich, and Manja Kamprad. Computer-assisted classification of hep-2 immunofluorescence patterns in autoimmune diagnostics. *Autoimmunity Reviews*, 2(5):298 – 304, 2003.

[21] Ryusuke Nosaka and Kazuhiro Fukui. Hep-2 cell classification using rotation invariant co-occurrence among local binary patterns. *Pattern Recognition*, 47(7):2428 – 2436, 2014.

[22] Roman Stoklasa, Tomáš Majtner, and David Svoboda. Efficient k-nn based hep-2 cells classifier. *Pattern Recognition*, 47(7):2409 – 2418, 2014.

[23] Linlin Shen, Jiaming Lin, Shengyin Wu, and Shiqi Yu. Hep-2 image classification using intensity order pooling based features and bag of words. *Pattern Recognition*, 47(7):2419 – 2427, 2014.

[24] Diego Gragnaniello, Carlo Sansone, and Luisa Verdoliva. Cell image classification by a scale and rotation invariant dense local descriptor. *Pattern Recognition Letters*, 82, Part 1:72 – 78, 2016. Pattern recognition Techniques for Indirect Immunofluorescence Images Analysis.

[25] Dimitris Kastaniotis, Foteini Fotopoulou, Ilias Theodorakopoulos, George Economou, and Spiros Fotopoulos. Hep-2 cell classification with vector of hierarchically aggregated residuals. *Pattern Recognition*, 65:47 – 57, 2017.

[26] Shahab Ensafi, Shijian Lu, Ashraf A. Kassim, and Chew Lim Tan. Accurate hep-2 cell classification based on sparse bag of words coding. *Computerized Medical Imaging and Graphics*, 57:40 – 49, 2017. Recent Developments in Machine Learning for Medical Imaging Applications.

[27] P. Hobson, B. C. Lovell, G. Percannella, M. Vento, and A. Wiliem. Classifying anti-nuclear antibodies hep-2 images: A benchmarking platform. In *2014 22nd International Conference on Pattern Recognition*, pages 3233–3238, Aug 2014.

[28] Rafael C Gonzalez and Richard E Woods. Digital image processing 3rd edition, 2007.

[29] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[30] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27 – 48, 2016. Recent Developments on Deep Big Vision.

[31] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[32] Dominik Scherer, Andreas Müller, and Sven Behnke. *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*, pages 92–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[33] Léon Bottou. *Large-Scale Machine Learning with Stochastic Gradient Descent*, pages 177–186. Physica-Verlag HD, Heidelberg, 2010.

[34] Korsuk Sirinukunwattana, Shan E Ahmed Raza, Yee-wah Tsang, David R. J. Snead, Ian A. Cree, and Nasir M. Rajpoot. Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images. *IEEE Transactions on Medical Imaging*, 35(5):1196–1206, may 2016.

[35] Yoshua Bengio. *Practical Recommendations for Gradient-Based Training of Deep Architectures*, pages 437–478. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[36] Hoo-chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer Learning. *IEEE transactions on medical imaging*, 35(5):1285–98, may 2016.

[37] NVIDIA Digits. https://developer.nvidia.com/digits, 2016.

[38] 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.

[39] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[40] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[41] Brian C Lovell, Gennaro Percannella, Mario Vento, and Arnold Wiliem. Performance evaluation of indirect immunofluorescence image analysis systems. In *International Conference on Pattern Recognition (ICPR)*, 2014.

[42] Santa Di Cataldo, Andrea Bottino, Ihtesham Ul Islam, Tiago Figueiredo Vieira, and Elisa Ficarra. Subclass discriminant analysis of morphological and textural features for hep-2 staining pattern classification. *Pattern Recognition*, 47(7):2389 – 2399, 2014.

[43] Xiangfei Kong, Kuan Li, Jingjing Cao, Qingxiong Yang, and Liu Wenyin. Hep-2 cell pattern classification with discriminative dictionary learning. *Pattern Recognition*, 47(7):2379 – 2388, 2014.

[44] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*, pages 346–361. Springer International Publishing, Cham, 2014.

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.