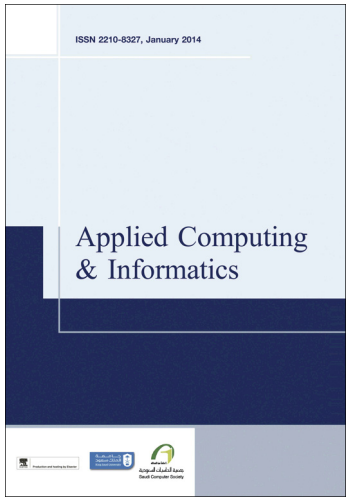
Accepted Manuscript



Ensemble of Convolutional Neural Networks for Bioimage Classification

Loris Nanni, Stefano Ghidoni, Sheryl Brahnam

PII: S2210-8327(18)30138-8
DOI: https://doi.org/10.1016/j.aci.2018.06.002
Reference: ACI 120

To appear in: *Applied Computing and Informatics*

Received Date: 22 April 2018
Revised Date: 30 May 2018
Accepted Date: 6 June 2018

Please cite this article as: Nanni, L., Ghidoni, S., Brahnam, S., Ensemble of Convolutional Neural Networks for Bioimage Classification, *Applied Computing and Informatics* (2018), doi: https://doi.org/10.1016/j.aci.2018.06.002

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

Ensemble of Convolutional Neural Networks for Bioimage Classification

Loris Nanni [1*], Stefano Ghidoni[1], Sheryl Brahnam[2]

[1] Department of Information Engineering, University of Padua, via Gradenigo 6/B, 35131 Padova, Italy.

[2] Computer Information Systems, Missouri State University, 901 S. National, Springfield, MO 65804, USA

[*]loris.nanni@unipd.it

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Ensemble of Convolutional Neural Networks for Bioimage Classification

Abstract: This work presents a system based on an ensemble of Convolutional Neural Networks (CNNs) and descriptors for bioimage classification that has been validated on different datasets of color images. The proposed system represents a very simple yet effective way of boosting the performance of trained CNNs by composing multiple CNNs into an ensemble and combining scores by sum rule. Several types of ensembles are considered, with different CNN topologies along with different learning parameter sets. The proposed system not only exhibits strong discriminative power but also generalizes well over multiple datasets thanks to the combination of multiple descriptors based on different feature types, both learned and handcrafted. Separate classifiers are trained for each descriptor, and the entire set of classifiers is combined by sum rule. Results show that the proposed system obtains state-of-the-art performance across four different bioimage and medical datasets. The MATLAB code of the descriptors will be available at https://github.com/LorisNanni.

ACCEPTED MANUSCRIPT

1. Introduction

Despite strong advances in automatic image analysis in recent years, in the field of medicine, expert clinicians remain the ones who typically make the final diagnostic determination of medical images. Automatic and semi-automatic analysis is gaining in importance, however, due to the massive growth in medical imaging technologies and thanks to some giant strides in the fields of image processing, pattern

ACCEPTED MANUSCRIPT

recognition, and image classification, all of which have made automatic analysis of medical images a viable alternative [1-3].

In general, bioimage processing often relies on approaches based on feature extraction from images that contain important information for a particular diagnostic task. Some of the best feature extraction methods for biological tissue analysis consider local textured patterns. A large variety of textural features have been employed in biomedical imaging classification systems, with some of these features combined together in ensembles under the assumption that different textural features extract different types of information from the same image [4, 5]. Some typical methods for extracting textural features include Gabor filters and Haralick's co-occurrence matrix [6]. Other feature extraction methods commonly used today are the Scale-Invariant Feature Transform (SIFT) and Local Binary Patterns (LBP) along with its many variants [7, 8]. These feature extraction methods belong to what is often referred to as the class of *handcrafted* descriptors, so named because the algorithms are designed by researchers to detect specific characteristics considered important in the analysis of images.

Besides handcrafted features, some machine learning techniques have been developed that learn features automatically. This class of so-called *learned features* are also widely used in bioimage processing [9, 10], but they tend to be limited in power because they rely heavily on the dataset used for training. This problem can be overcome by training on a very large dataset (or an ensemble of datasets) containing a broad set of images so that the system learns a wide variety of different patterns. In this way, the learned features become independent of any specific dataset and can be considered as general feature extractors. Like the handcrafted features mentioned above, these learned features can be used alone or in combination with other sets of features, both handcrafted and learned, to analyze new problems. Some examples along these lines include [9], where learned features are used for the detection of ovarian carcinomas, and [10], where learned features are combined with handcrafted features for histopathology image representation.

A powerful class of learned descriptors has recently been proposed that are based on the deep learning paradigm [11]. Deep learning has proven to be extremely effective in several image classification

ACCEPTED MANUSCRIPT

tasks, including medical image analysis [12]. Some examples include the detection/counting of mitotic events, the segmentation of nuclei, and many cancerous vs. noncancerous tissue evaluations [13].

A deep learning architecture that has been studied extensively is the Convolutional Neural Network (CNN) [14], which is a multi-layered image classification technique that incorporates spatial context and weight sharing between pixels. A CNN learns the optimal image features for a specific image classification problem by adopting an effective representation of the original image. Inspired by the process of visual perception in human beings, it requires little to no preprocessing. The basic components of a CNN are stacks of different types of specialized layers (convolutional, activation, pooling, fully-connected, softmax, etc.) that are interconnected and whose weights are trained using the backpropagation algorithm. The deepest layers of the network function as low-level feature extractors. The training phase of a CNN requires huge numbers of labelled data to avoid the problem of over-fitting; however, once trained, CNNs are capable of producing accurate and generalizable models that achieve state-of-the-art performance in general pattern recognition tasks. Some examples include LeNet [15], the first CNN proposed to classify handwritten digits; AlexNet [16], a deep network designed for image classification; ZFNet [17], a newer model that outperforms AlexNet; VGGNet [18], which increases depth using 3×3 convolution filters; GoogLeNet [19], which includes inception modules (which is a new organizational structure); and ResNet [20], a residual network that is much easier to optimize than VGGNets. The CNN architecture and the cited examples are discussed in more detail in Section 2.

When deep neural networks are trained on large datasets of images, the first convolutional filters learned by the network often resemble either Gabor filters or color blobs that are easily transferable to many other image tasks and datasets [21]. Pre-trained models can thus be used to extract learned features from novel sets of images, and these features can then be fed into other classifiers, similar to the way handcrafted features are used. Conversely, features computed in the last layer of a pretrained network are strongly dependent on the dataset used to train the deep learner and thus on the specific classification problem represented by a given dataset. Nonetheless, the outputs of these layers can be used for other tasks if CNN fine-tuning is exploited.

ACCEPTED MANUSCRIPT

All three deep learning methods described above are used in medical and bioimage classification [22]. To summarize the possibilities mentioned so far: a) deep learners can be trained on images from scratch (as in [23]); b) pre-trained CNNs can function as additional feature extractors that can be combined with existing handcrafted image features (as in [24] and [25]); and c) the outputs of pre-trained CNNs can be fine-tuned by another simpler classifier, such as SVM, on novel target images (as in [26] and [27]). Yet another class of approaches combines different CNN architectures to exploit the strengths and offset the weaknesses of a given architecture [27].

In this work, we investigate methods for building ensembles of CNNs by leveraging pre-trained CNNs. We consider several different training patterns and experiments using different learning rates, batch sizes, and topologies. What is interesting is that this simple approach produces a very high performing system, one that strongly outperforms the single best CNN trained specifically on a given dataset. Of course, there are both pros and cons involved in combining different CNNs. Although ensembles of CNNs perform exceptionally well, training such models requires high computational power (in this work we used three TitanX GPUs). Moreover, the total size of the network set is quite large, requiring considerable computational power for input classification. Hence, this approach is suitable only for problems where computation time is not critical.

Aside from exploring different ensembles of CNNs, we also consider combining heterogeneous handcrafted descriptors for bioimage classification. The best system proposed in this work combines both learned and handcrafted descriptors. For each descriptor, a different classifier is trained, and the set of classifiers along with the classification results from the deep learners are combined by sum rule. The handcrafted descriptors tested in this paper are summarized in section 3, and the power of this approach is validated on four different biomedical color datasets.

We wish to stress that the main goal of the proposed system is to produce a powerful general-purpose image classification system able to work out-of-the-box (i.e. requiring little to no parameter tuning) on any bioimage classification problem. We strive to produce a general-purpose system that performs competitively against less flexible systems that have been optimized for very specific image problems and

ACCEPTED MANUSCRIPT

datasets. Experimental results demonstrate that the proposed system obtains state-of-the-art performance in every tested bioimage problem. Yet the same set of descriptors is used in all the tested datasets, demonstrating the generalizability of the proposed approach.

2. Deep Learned Features

CNNs are a class of deep feed-forward neural networks. Like most neural networks, CNNs are composed of interconnected neurons that have inputs with learnable weights, biases, and activation functions.

CNN layers have neurons arranged in three dimensions: width, height and depth. This means that every layer in a CNN transforms a 3D input volume into a 3D output volume of neuron activations. CNNs are built with five classes of layers: *conv*olutional (CONV), *act*ivation (ACT), *pool*ing (POOL), followed by a last stage, including Fully-Connected (FC), and *class*ification (CLASS).

The CONV layer is the core building block of a CNN and is also what makes CNNs so computationally expensive. These layers compute the outputs of neurons that are connected to local regions by applying a convolution operation to the input. The spatial extent of connectivity of these local regions is a hyperparameter called the *receptive field*, and a parameter sharing scheme is used in CONV Layers to control the number of parameters. This means that the parameters of CONV layers are shared sets of weights (also called kernels or filters) that have relatively small receptive fields.

POOL layers perform non-linear downsampling operations. Max pooling is the most common non-linear operation: it partitions the input into a set of non-overlapping rectangles and outputs the maximum for each group. In this way POOL reduces the spatial size of the representation while simultaneously reducing 1) the number of parameters, 2) the possibility of overfitting, and 3) the computational complexity of the network. It is common practice to insert a POOL layer between CONV layers.

ACT layers apply some activation function, such as the non-saturating ReLU (Rectified Linear Unit) function $f(x) = \max(0, x)$ or the saturating hyperbolic tangent $f(x) = \tanh(x)$, $f(x) = |\tanh(x)|$, or the sigmoid function $f(x) = (1 + e^{-x})^{-1}$.

ACCEPTED MANUSCRIPT

FC layers have neurons that are fully connected to all the activations in the previous layer and are applied after CONV and POOL layers.

In this work, we test and combine the following CNN architectures:

- AlexNet [16]: this is the 2012 winner of the ImageNet ILSVRC challenge. AlexNet is a popular CNN that is composed of both stacked and connected layers. It includes five CONV layers followed by three FC layers, with some max-POOL layers inserted in the middle. A rectified linear unit nonlinearity is applied to each convolutional along with a fully connected layer to enable faster training.
- GoogleNet [19]: this is the 2014 winner of the ImageNet ILSVRC challenge. The main novelty of this CNN is the introduction of an inception module (INC), i.e. a subnetwork consisting of parallel convolutional filters whose outputs are concatenated. INC greatly reduces the number of parameters required (much lower than AlexNet). GoogleNet is composed of 22 layers that require training (27 layers in total, counting the POOL layers).
- VGGNet [18]: this is a CNN that placed second in ILSVRC 2014. The two best-performing VGG models (VGG-16 and VGG-19), with 16 and 19 weight layers, respectively, are available as pretrained models. Both models are very deep and include 16 CONV/FC layers. The CONV layers are extremely homogeneous and use very small (3×3) convolution filters. A POOL layer is inserted after two or three CONV layers (instead after each CONV layer as is the case with AlexNet).
- ResNet [20]: this is the winner of ILSVRC 2015. This network is approximately twenty times deeper than AlexNet and eight times deeper than VGGNet. The main novelty of this CNN is the introduction of residual (RES) layers, making it a "network-in-network" architecture. ResNet uses special skip connections and batch normalization, and the FC layers at the end of the network are substituted by global average pooling. Instead of learning unreferenced functions, ResNet explicitly reformulates layers as learning residual functions with reference to the layer inputs. As a

ACCEPTED MANUSCRIPT

result, ResNet is much deeper than VGGNet, although the model size is smaller and thus easier to optimize than VGGNet.

- Inception [19]: InceptionV3 is a variant of GoogleNet based on the factorization of 7×7 convolutions into two or three consecutive layers of 3×3 convolutions.
- IncResv2 [28]: Inception-ResNet-v2 is an Inception style networks that utilize residual connections instead of filter concatenation.

As noted in the introduction, the learning effectiveness of a CNN depends on the availability of large training data. Data augmentation is one effective way to expand training data when necessary and to reduce overfitting during CNN training by artificially expanding the training set using perturbations of individual images [16]. Data augmentation applies transformations and deformations to the labeled data, thus producing new samples as additional training data. A key attribute of the data augmentation process is that the labels remain unchanged after applying the transformations. In this work we perform random data augmentation with horizontal and vertical flipping, rotation in a range of 10°, translation of a maximum of five pixels, and scaling in a range of [1, 2].

Fine-tuning a CNN is a procedure that essentially restarts the retraining process of a pretrained network so that it learns a different classification problem. We adopt the Two-Round Tuning for fine-tuning a CNN. With Two-Round Tuning, the first round of tuning is performed by training a CNN using a leave-one-out dataset strategy, e.g. by including in the training set all the images from the dataset summarized in Table 2 except for the target dataset. The final number of classes becomes the sum of all the classes from each classification problem. The second round of tuning is the same as in One Round Tuning and involves only the training set of the target problem.

In keeping with the rationale of the Data Augmentation step, we use the following datasets in the first round of tuning:

ACCEPTED MANUSCRIPT

- PAP: the PAP SMEAR dataset [29], which contains 917 images acquired during Pap tests to identify cervical cancer diagnosis (available at http://labs.fme.aegean.gr/decision/downloads);
- LG: the "Liver gender" [30] dataset, which includes 265 images of liver tissue sections from 6-month male and female mice on a caloric restriction diet (the classes are the 2 genders);
- LA: the "Liver aging" [30] dataset, which includes 529 images of liver tissue sections from female mice of 4 ages on an ad-libitum diet;
- BR: the BREAST CANCER dataset [31], which contains 1394 images divided into the control, malignant cancer, and benign cancer classes;
- HI: the HISTOPATHOLOGY dataset [32], which contains 2828 images of connective, epithelial, muscular, and nervous tissue classes.
- RPE: a dataset composed of 195 human stem cell-derived retinal pigmented epithelium images that were divided into 16 subwindows with each subwindow divided into four classes by two trained operators (available at https://figshare.com/articles/BioMediTech_RPE_dataset/2070109).

We fine-tune the weights of the pretrained CNNs by fixing the deep CONV layers of the network and by fine-tuning only the higher-level FC layers since these layers are specific to the details of the classes contained in the target dataset. The last FC layer is designed to be the same size as the number of classes in the new dataset. All the FC layers are initialized with random values and trained from scratch using the Stochastic Gradient Descent (SGD) algorithm with data from the target training set.

3. Handcrafted features

In Table 1 we summarize the handcrafted descriptors used in our tests, along with the parameter sets used to extract each descriptor. Each descriptor is trained on an SVM, and only the training data is used to fix its parameters. Since we are working with RGB color datasets, each texture descriptor is

ACCEPTED MANUSCRIPT

applied separately to each RGB channel, with the final score given by the sum rule of the three classifiers trained with the three set of features.

Name	Parameters	Source	Section
LTP	Multiscale Uniform LTP with two (*R*,*P*) configurations: (1, 8) and (2, 16), threshold=3.	[33]	3.1
MLPQ	Ensemble of LPQ descriptors obtained by varying the filter sizes, the scalar frequency, and the correlation coefficient between adjacent pixel values.	[34]	3.2
CLBP	Completed LBP with two (*R*,*P*) configurations: (1,8) and (2,16).	[35]	3.3
RIC	Multiscale Rotation Invariant Co-occurrence of Adjacent LBP with $R \in \{1, 2, 4\}$.	[36]	3.4
FBSIF	Extension of the BIF by varying the parameters of filter size (SIZE_BSIF, *size* $\in$ {3, 5, 7, 9, 11}) and the threshold for binarizing (FULL_BSIF, *th* $\in$ {-9, -6, -3, 0, 3, 6, 9}).	[37]	3.5
AHP	Adaptive Hybrid Pattern with quantization *level* = 5 and 2; the (*R*,*P*) configurations are (1, 8) and (2, 16).	[38]	3.6
GOLD	Ensemble of Gaussians of LOcal Descriptors extracted using the spatial pyramid decomposition.	[39]	3.7
HOG	Histogram of Oriented Gradients with 30 cells (5 by 6).	[40]	3.8
MOR	A set of MORphological features.	[41]	3.9
CLM	CodebookLess Model. We use the ensemble named CLoVo_3 in [15] based on e-SFT, PCA for dimensionality	[42]	3.10

ACCEPTED MANUSCRIPT

	reduction, and one-vs-all SVM for the training phase.		
LET	Same parameters used in the source code of [43]	[43]	3.11

Table 1 Summary Handcrafted Descriptors

As it can be observed in Table 1, many of the handcrafted texture descriptors are based on Local Binary Patterns (LBP), a descriptor that has achieved great success due to its computational efficiency and discriminative power. The traditional LBP [44] is expressed as

$$LBP_{P,R} = \sum_{P=0}^{P-1} s(x)2^P, \quad (1)$$

where $x = q_p - q_c$ is the difference between the intensity levels of a central pixel (q_c) and a set of neighbouring pixels (q_p). A neighbourhood is defined by a circular region of radius R and P neighbouring points. The function $s(x)$ in Eq. 1 is defined as:

$$s(x) = \begin{cases} 1, x \geq 0 \\ 0, otherwise \end{cases} \quad (2)$$

LBP descriptors are the histograms of these binary numbers.

3.1 The Local Ternary Pattern (LTP)

LTP [33] is a ternary variant of LBP and is designed to reduce the noise in the feature vector when uniform regions are analyzed. LTP proposes a three-value coding scheme that includes a threshold around zero for the evaluation of the local gray-scale difference by adding to Eq. (2) the threshold τ:

$$s(x) = \begin{cases} 1, & x \geq \tau \\ 0, & |x| \leq \tau \\ -1, & x \leq -\tau \end{cases} \quad (3)$$

3.2 Multithreshold Local Phase Quantization (MLPQ)

MLPQ [34] extends the multi-threshold approach described for LBP to the LPQ feature [45, 46] that is based on the phase of the Short-Term Fourier Transform (STFT) evaluated on a rectangular neighborhood of size *R*. The MLPQ features used in our experiments are computed using parameter belonging to the

ACCEPTED MANUSCRIPT

following ranges: $\tau \in \{0.2, 0.4, 0.6, 0.8, 1\}$, $R \in \{1, 3, 5\}$, $a \in \{0.8, 1, 1.2, 1.4, 1.6\}$ and $\rho \in \{0.75, 0.95, 1.15, 1.35, 1.55, 1.75, 1.95\}$. Such sets were proposed in [47].

3.3 Completed LBP (CLBP)

CLBP, proposed in [35], encodes a texture by means of two components: the difference sign and the different magnitude that is computed between a reference pixel and all the pixels belonging to a given neighborhood. CLBP represents a local region by its centre pixel (CLBP-C) and a local difference sign-magnitude transform (LDSMT). This is what produces the difference signs and the difference magnitudes.

Two operators, CLBP-Sign (CLBP_S) and CLBP-Magnitude (CLBP_M), are defined for the difference signs and the difference magnitudes. Since all three descriptors (CLBP_C, CLBP_S and CLBP_M) are in binary format, they can be combined to form the final CLBP histogram.

Given a central pixel g_c and its P evenly spaced circular neighbors g_c , g_p ,p,1,…,P-1, the difference between g_c and g_p can be calculated as $d_p = g_p - g_c$ and be decomposed into two components defining the LDSMT transform:

$$d_p = S_p * m_p \text{ and } \begin{cases} S_p = sign\,(d_p) \\ m_p = |d_p| \end{cases},$$

$$S_p = \begin{cases} 1, d_p \geq 0 \\ -1, d_p < 0 \end{cases} \quad (4)$$

where S_p is the sign of d_p, and m_p is the magnitude of d_p. Thus, the LDSMT transforms the vector $[d_0, \dots, d_{P-1}]$ into a sign vector$[s_0, \dots, s_{P-1}]$ and a magnitude vector $[m_0, \dots, m_{P-1}]$.

The CLBP_S operator is the traditional LBP operator defined in Eq. (1). The CLBP_M is defined as:

$$LBP_{M_{P,R}} = \sum_{p=0}^{P-1} t(m_p, c)2^P,$$
$$t(x, c) = \begin{cases} 1, x \geq c \\ 0, x < c \end{cases} \quad (5)$$

where c is the mean value of m.

The center pixels represent the image gray level and thus contains discriminant information. These values are converted into a binary code by global thresholding, which makes them consistent with

ACCEPTED MANUSCRIPT

CLBP_S and CLBP_M as $CLBP\_C_{P,R} = t(g_c, c_1)$, where t is the threshold defined in Eq. (5), and c_1 is the average gray level of the white image.

Combining CLBP_S, CLBP_M, and CLBP_C features into joint or hybrid distributions results in significant improvement for rotation invariant texture classification. The CLBP_S, CLBP_M, and CLBP_C histograms are concatenated to obtain the CLBP descriptor.

3.4 Multiscale Rotation Invariant Co-occurrence of Adjacent LBP (RIC)

RIC [36] considers the co-occurrence in the context of LBP features, or the spatial relations among pixels. This feature adds rotational invariance for angles that are multiples of 45°. RIC depends on two parameters, namely, LBP radius and the displacement among the LBPs. The values used in our experiments are: (1, 2), (2, 4) and (4, 8).

3.5 Full BSIF (FBSIF)

FBSIF [37] is an extension of the Binarized Statistical Image Feature (BSIF) [48], that assigns each pixel of the input image a n-bit label obtained by means of a set of n linear filters. Each filter operates on a neighborhood of $l \times l$ pixels around the element they should give the label. This n-bit label can be formalized as:

$$s = \boldsymbol{W}\boldsymbol{X} \tag{6}$$

where $\boldsymbol{X}$ is a vector of length $l^2 \times 1$ obtained from the neighborhood, while W is a $n \times l^2$ matrix including the filters vector notations. FBSIF operates by evaluating BSIF using several values of the filter size (SIZE_BSIF) and a binarization threshold (FULL_BSIF). Values considered in this work are: SIZE_BSIF $\in$ {3, 5, 7, 9, 11}, FULL_BSIF $\in$ {-9, -6, -3, 0, 3, 6, 9}. Each combination of size and threshold is fed to a separate SVM: the SVMs are then combined by sum rule.

3.6 Adaptive Hybrid Pattern (AHP)

ACCEPTED MANUSCRIPT

AHP [38] descriptors were created to overcome two main drawbacks of the LBP feature: 1) its noisy behavior in quasi-uniform regions and 2) its reactivity, that is, the strong variations in the descriptor that are possibly induced by small variation in the input image, which is caused by the use of quantization thresholds.

AHP overcomes both problems by using a Hybrid Texture Model (HTD) composed of local primitive features and global spatial structure and then by applying an adaptive quantization algorithm (AQA) to improve the noise robustness of the angular space quantization. In this way, the vector quantization thresholds are adaptive to the content of the local patch.

AQA extracts the discriminative texture information provided by primitive microfeatures. T_{HTD} is defined as:

$$T_{HTD} \approx T_{global} + T_{local} \tag{7}$$

where T_{HTD} represents the texture, T_{global} the global texture information, and T_{local} the local texture information. T_{global} is the joint distribution of the global difference between gray values of the circular symmetric neighborhoods and the mean value from the whole texture image. T_{local} is the joint distribution of the local differences between the gray value of the center pixel and the gray values of the circularly symmetric neighborhoods.

The length of the feature histogram of the whole image is reduced by splitting the global pattern and the local pattern into multiple binary patterns using the threshold calculations in [49] and [50].

3.7 Gaussian Of Local Descriptors (GOLD)

GOLD [39] is based on a four-step algorithm: *i)* evaluation of SIFT features; *ii)* spatial pyramid decomposition; *iii)* parametric probability density estimation; iv) the covariance matrix is projected onto the tangent Euclidean space in order to vectorize the feature. In other words, GOLD descriptors are obtained by extracting some descriptors from an image to obtain $\mathbb{D}=\{D_1,.. D_N\}$, where $D_i \in \Re^n$, by collecting and weighting them in a spatial pyramid, and then by describing each subregion by the

ACCEPTED MANUSCRIPT

estimated parameters of a multivariate Gaussian distribution. To vectorize the descriptors, the covariance matrix is projected onto a Euclidean space and concatenated to the mean vector to obtain the final descriptor of size $(n^2+3n)/2$. Finally, the feature vector is fed into an SVM with a histogram kernel.

3.8 Histogram of Oriented Gradients (HOG)

HOG [40] groups pixels into small windows and measures intensity gradients in each of them. It is possible to view HOG as a simplified version of SIFT. HOG calculates intensity gradients, pixel by pixel; and the selection of a corresponding histogram bin for each pixel is based on the gradient direction. A histogram is then evaluated for each window, leading to the final descriptor. Windows of size 5×6 are used in our experiments.

3.9 Color descriptor (COL)

COL, proposed in [51], is a simple and compact descriptor, acquired combining statistical measures extracted from each color channel in the RGB space. The final descriptor is obtained as the concatenation of several measures: the mean, the standard deviation, the 3rd and 5th moments of each color channel and the marginal Histograms (8 bins per channel) [51].

3.10 Morphological descriptor (MOR)

MOR, proposed in [41], is a set of measures extracted from a segmented version of the image, including the aspect ratio, number of objects, area, perimeter, eccentricity, and other measures.

3.11 CodebookLess Model (CLM)

CLM [42] is based on an image modeling method that can represent an image by means of a single Gaussian. This is obtained by first evaluating SIFT features on a regular grid placed on the image. Thus, CLM is a dense sampling features model, and fitting them using a Gaussian model. The main difference

ACCEPTED MANUSCRIPT

between CLM and the other widely used dense sampling method, such as the BoF approach [52], is the absence of a codebook.

According to the experiments reported in [24], we select for CLM the ensemble named CLoVo_3 in [24] based on e-SFT, PCA for dimensionality reduction and one-vs-one SVM for the training phase.

3.12 LETRIST descriptor (LET)

LET, proposed in [43], is simple but effective representation that encodes the joint information within an image across feature and scale spaces. We use the default values available in the MATLAB toolbox.

4. Materials

Several medical datasets were used to test our system and demonstrate the generalizability of our approach. Each dataset contains different types of medical images. For the sake of easy comparisons, the datasets used in our experiments were selected because they are publicly available:

- **LY:** the LYMPHOMA dataset [53], which includes 375 images of malignant lymphoma subdivided in three classes: CLL (chronic lymphocytic leukemia), FL (follicular lymphoma), and MCL (mantle cell lymphoma).
- **BGR:** the BREAST GRADING CARCINOMA [54], which is a medium size dataset containing 300 images (Grade 1: 107, Grade 2: 102, and Grade 3: 91 images) of resolution 1280×960 corresponding to 21 different patients with invasive ductal carcinoma of the breast.
- **LAR:** the LARYNGEAL dataset [55], which contains a well-balanced set of 1320 patches extracted from the endoscopic videos of 33 patients affected by laryngeal squamous cell carcinoma (SCC). The patches are relative to four laryngeal tissue classes. LAR contains color images. In our experiments with this dataset each descriptor is separately extracted from each color channel.

ACCEPTED MANUSCRIPT

- **CO:** the COLORECTAL dataset [56], which is a collection of textures obtained by manual annotation and tessellation of histological images of human colorectal cancer.

Table 2 summarizes some important characteristics of each dataset including the number of classes (#C), the number of samples (#S) (i.e. the number of images), the image size, and the URL for downloading the dataset. The testing protocol used in our experiments is the five-fold cross-validation method except in those case where the database is specifics its own protocol.

Dataset	#C	#S	Size	URL for Download
BGR	3	300	1280×960	https://zenodo.org/record/834910#.Wp1bQ-jOWUl
LY	3	375	1388×1040	ome.grc.nia.nih.gov/iicbu2008
LAR	3	1320	100×100	https://zenodo.org/record/1003200#.WdeQcnBx0nQ
CO	8	5000	150×150	zenodo.org/record/53169#.WaXjW8hJaUm

Table 2 Descriptive Summary of the Datasets

5. Experimental results

The experimental evaluations reported in this section are intended first to compare the performance of handcrafted descriptors to deep learned descriptors on several cancer data analysis classification tasks and second to evaluate the performance of several ensembles based on the fusion of classifiers. Our main objective is to design a method that is both robust and effective on different classification problems. To assess the generalizability and robustness of our system, our best performing method is finally compared with several state-of-the-art results published by different researches on the same datasets. Note: Before each fusion, the scores of the classifiers of each descriptor are normalized to mean 0 and standard deviation 1. Experiments reported below were statistically validated using the Wilcoxon signed rank test.

ACCEPTED MANUSCRIPT

In the first experiment, reported in Table 3, we evaluate the performance (using accuracy as the performance indicator) of the baseline handcrafted descriptors described in section 3. Moreover, the performance obtained by the following ensembles of handcrafted methods are compared:

- FH: the fusion by sum rule of the following handcrafted methods LTP, CLBP, RIC, LET, MOR, AHP, COL, MLPQ and FullBSIF. We have not use GOLD and CLM in FH since they are computational expensive. Note: the scores of each method are normalized to mean zero and standard deviation 1 so that the importance of MLPQ and FullBSIF (that are methods based on ensemble combined by sum rule) is equal to the other approaches;
- FH+CLM: sum rule among the methods belonging to FH and CLM, i.e. the sum rule among the nine methods of FH and CLM;
- FH+CLM+GOLD: sum rule among the methods belonging to FH, GOLD and CLM;
- PREV: the ensemble of handcrafted features proposed in [24];
- PREV1: ensemble of handcrafted features proposed in [57].

ACCEPTED MANUSCRIPT

	LY	CO	BGR	LAR
LTP	85.33	90.40	87.54	71.97
MLPQ	92.27	93.58	90.54	82.27
CLBP	86.67	92.04	89.54	72.27
RIC	85.87	91.56	91.87	90.68
LET	92.53	93.18	93.54	90.76
MOR	84.53	93.30	91.54	79.85
AHP	93.87	94.16	91.37	85.30
COL	91.47	92.30	90.71	85.30
FBSIF	92.53	93.42	88.00	88.56
GOLD	53.07	83.58	75.33	90.61
CLM	74.40	89.60	86.33	87.58
FH	**95.20**	**95.18**	91.67	91.29
FH+CLM	94.93	95.08	91.67	92.12
FH+CLM+GOLD	93.60	94.92	**92.00**	**93.26**
PREV	92.00	93.74	87.00	92.05
PREV1	92.00	94.68	88.67	92.58

Table 3 Handcrafted descriptors.

Clearly, the fusion approaches FH and FH+CLM+GOLD works better (Wilcoxon signed rank test - p-value of 0.05) than the stand-alone methods and the previous handcrafted ensembles PREV and PREV1.

In the second experiment, see Tables 4 and 5, we test the feasibility of building an ensemble of convolutional neural networks[1] as follows:

- Training CNNs using different learning rates (LR), i.e. 0.001 & 0.0001;
- Training CNNs using different batch sizes (BS), i.e. 10, 30, 50 and 70;
- Training CNNs using different topologies.

In Table 4 we report experiments using standard tuning. In Table 5 we report the performance of the Two-Round tuning detailed in section 2.

[1] All the CNN are implemented using the MathWorks Neural Network Toolbox

ACCEPTED MANUSCRIPT

The following methods are also reported in Tables 4-5:

- SB: the single best CNN configuration in that dataset. This method is clearly overfitted since we report the best result on the testing set after running different parameter configurations and choosing the best one. It is important to report SB as baseline performance for the proposed ensemble.
- AB: best average CNN configuration in all the datasets.
- Fus: fusion among all the different CNNs trained varying the parameter configuration. If the CNN does not converge (i.e. it produces random results on the training data, which usually happens with AlexNet and VggNet with LR=0.001), the CNN is excluded from the ensemble. Note that it is not always feasible to train a CNN with a large batch size. In other words, if for a given BS we obtain a "GPU out of memory" error message, we discard that CNN configuration.
- FCN-st: fusion among the methods Fus of the all CNN topologies trained using standard tuning (column All) or all the topologies trained using standard tuning except AlexNet (column NoAlex). The scores of each given CNN topology are normalized considering how many CNNs of that topology are effectively used in the fusion Fus (i.e. by excluding all CNNs that produce random results on the training set or an out of memory error message).
- FCN-Two: fusion among the methods Fus of the all CNN topologies trained using Two-round tuning (column All) or all the topologies trained using Two-round tuning except AlexNet (column NoAlex). The scores of each given CNN topology are normalized considering how many CNNs of that topology are effectively used in the fusion Fus (i.e. by excluding all CNNs that produce random results on the training set or an out of memory error message).

Notice that Two-Round Tuning is applied on a reduced number of topologies due to computational issues.

	GoogleNet			ResNet50			ResNet101			Inception		
	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus
LY	82.93	82.93	82.93	86.40	86.40	90.67	86.40	86.40	86.13	87.47	87.47	86.93
CO	95.60	95.60	96.30	95.42	95.42	96.40	92.92	92.92	94.68	95.02	92.82	96.40

ACCEPTED MANUSCRIPT

BGR	93.00	92.67	94.33	91.33	90.33	94.00	93.33	93.33	93.00	93.67	93.67	95.00
LAR	92.35	90.83	91.97	92.20	92.05	93.41	93.64	93.64	93.79	92.73	89.77	93.56

	AlexNet			VGG16			VGG19			IncResv2			FCN-st	
	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus	All	NoAlex
LY	82.40	82.40	80.00	80.80	80.80	85.07	82.40	82.40	86.13	84.80	84.80	85.87	93.87	**93.60**
CO	94.22	94.22	95.14	96.14	96.14	96.88	95.94	95.26	96.76	93.58	93.58	95.16	97.26	**97.32**
BGR	92.00	91.00	91.33	93.00	93.00	95.00	93.67	93.67	91.67	91.00	91.00	90.67	96.00	**96.00**
LAR	90.68	89.39	90.08	93.33	91.52	91.82	94.24	93.26	**95.38**	94.62	94.62	94.39	94.70	94.85

Table 4 Standard tuning

	AlexNet			GoogleNet			VGG16			VGG19			FCN-Two	
	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus	SB	AB	Fus	All	NoAlex
LY	86.93	86.93	86.93	85.87	85.87	87.47	85.33	85.33	86.13	89.33	89.33	90.67	**95.47**	94.67
CO	94.70	92.48	95.48	95.48	95.28	96.50	96.16	95.88	97.04	96.62	95.62	97.44	97.20	**97.34**
BGR	91.67	90.33	92.00	92.33	92.33	93.00	92.23	92.00	94.33	92.00	92.00	94.00	94.33	**95.33**
LAR	92.42	90.23	92.05	91.29	90.00	92.12	94.02	94.02	93.26	93.48	93.48	**95.08**	94.24	94.70

Table 5 Two-round tuning

The following conclusions can be drawn from the results reported in Tables 4 and 5:

- For each topology, Fus outperforms AB (Wilcoxon signed rank test - p-value of 0.05);
- FCN outperforms each Fus (Wilcoxon signed rank test - p-value of 0.05);
- FCN-two obtains performance similar to FCN-st.

In Table 6 the ensemble of CNNs is combined with other methods. The ensembles evaluated in Table 6 are the following:

- FCN+: sum rule among the methods that belong to FCN-st and FCN-Two;
- Here1: sum rule between (FCN+ NoAlex) and FH; before fusion the scores of (FCN+ NoAlex) and FH are normalized to mean 0 and standard deviation 1;
- Here2: sum rule between (FCN+ NoAlex) and (FH+CLM+GOLD); before fusion the scores of (FCN+ NoAlex) and FH are normalized to mean 0 and standard deviation 1;

ACCEPTED MANUSCRIPT

	FCN+		Here1	Here2
	All	NoAlex		
LY	94.67	94.93	**97.33**	96.53
CO	97.23	97.50	**97.26**	97.20
BGR	**96.33**	**96.00**	95.33	95.33
LAR	94.77	94.85	95.38	**95.45**

Table 6 Ensemble proposed here.

In Table 7 we compare our ensemble Here1 with the literature, for a fair comparison we have reported methods based on the same testing protocol used to assess the performance of our approaches.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Methods	LY	CO	BGR	LAR
Here1	**97.33**	**97.60**	**95.00**	**95.45**
[57]	92.00	96.84	91.67	95.18
[24]	90.67	93.98		
[58]	96.80			
[1]	70.9			
[4]	66.0			
[56]		87.4		
[59]	90.93			

Table 7 Comparison with other state-of-the-art approaches; accuracy is used as the performance indicator.

The following conclusions can be drawn from the results reported in Tables 6 and 7:

- Here1 and Here2 outperform FCN+; since Here1 is simpler then Here2 our suggestion is to use Here1;
- Here1 obtains state-of-the-art-performance; e.g. in [55] a median F-measure of 92 is obtained in the LAR dataset, while our ensemble obtains an F-measure of 95.2.

ACCEPTED MANUSCRIPT

ACCEPTED MANUSCRIPT

Finally, in Table 8, we report the performance obtained by some ensemble proposed in this paper using the Kappa statistic [60] to measure the agreement between true and predicted class labels.

	FH	FCN+		Here1	Here2
		All	NoAlex		
LY	0.927	0.919	0.923	**0.959**	0.947
CO	0.944	0.968	**0.970**	**0.970**	0.969
BGR	0.873	**0.940**	0.939	0.929	0.929
LAR	0.883	0.930	0.931	0.942	**0.944**

Table 8. Ensemble tested here, k-statistic as performance indicator.

The conclusions that can be drawn by the results reported in Table 8 are similar to those that can be drawn by the performance reported in Table 6.

To better motivate the reason of the good performance of the ensemble of CNNs we calculate the Yule's Q-statistic [61] among the methods that build the ensemble. The Q-statistic is used to provide information about the correlation among the output of different classifiers. The average Q-statistic among the different CNNs that build FCN-st is 0.7098, hence the different CNNs brings different information and their combination permits to boost the performance of the stand-alone CNN.

6. Conclusion

In this work an ensemble of CNNs is proposed for cancer related color datasets. The ensemble is built in a very simple way by training and comparing the performance of CNNs using different learning rates, batch sizes, and topologies. The set of CNNs is simply combined with the sum rule. The most important finding of this work is that this simple ensemble outperforms the best stand-alone CNN. When the ensemble of CNNs is combined with other features based on handcrafted features, the final ensemble obtains state-of-the-art performance on all the four tested datasets. For each handcrafted features a

ACCEPTED MANUSCRIPT

different support vector machine is trained, than the set of SVMs is combined by sum rule; also, the fusion between deep learning ensemble and handcrafted features ensemble is performed by sum rule. Notice that, before the fusion, the set of scores of each ensemble is normalized to mean 0 and standard deviation 1.

In the future, we plan to develop and test different approaches for representing images using CNNs. Features extracted from these CNNs will then be used to train SVM classifiers. To reproduce our experiments, MATLAB source code will be available at https://github.com/LorisNanni.

7. Acknowledgment

We gratefully acknowledge the support of NVIDIA Corporation for the "NVIDIA Hardware Donation Grant" of a Titan X used in this research.

References

1. Zhou, J., et al., *BIOCAT: a pattern recognition platform for customizable biological image classification and annotation.* BMC Bioinformatics., 2013. **14**: p. 291.
2. Misselwitz, B., et al., *Enhanced CellClassifier: a multi-class classification tool for microscopy images.* BMC Bioinformatics, 2010. **11**(30).
3. Pau, G., et al., *EBImage - an R package for image processing with applications to cellular phenotypes.* Bioinformatics, 2010. **26**(7).
4. Uhlmann, V., S. Singh, and A.E. Carpenter, *CP-CHARM: segmentation-free image classification made accessible.* BMC Bioinformatics, 2016. **17**: p. 51.
5. Vailaya, A., et al., *Image classification for content-based indexing.* IEEE Transactions on Image Processing, 2001. **10**(1): p. 117-30.
6. Gonzalez, R.C. and R.E. Woods, *Digital Image Processing, 2nd Edition*. 2nd ed. 2001, Boston: Addison-Wesley Longman Publishing Co., Inc.
7. Nanni, L., A. Lumini, and S. Brahnam, *Survey on LBP based texture descriptors for image classification.* Expert Systems with Applications, 2012. **39**(3): p. 3634-3641.
8. Nanni, L., S. Brahnam, and A. Lumini, *Combining different Local Binary Pattern variants to boost performance.* Expert systems with applications, 2011. **38**(5): p. 6209-6216.
9. Vu, T.H., et al., *Histopathological image classification using discriminative feature-oriented dictionary learning.* IEEE Transactions on Medical Imaging, 2016. **35**(3): p. 738–51.
10. Otalora, S., et al., *Combining unsupervised feature learning and Riesz wavelets for histopathology image representation: application to identifying anaplastic medulloblastoma*, in *International Conference on Medical Image Computing and Computer Assisted Intervention*. 2015: Munich. p. 581–588.
11. Schmidhuber, J., *Deep learning in neural networks: An overview.* Neural Networks, 2015. **61**: p. 85-117.

ACCEPTED MANUSCRIPT

12. Greenspan, H., B. van Ginneken, and R.M. Summers, *Deep learning in medical imaging: overview and future promise of an exciting new technique.* IEEE Transactions on Medical Imaging, 2016. **35**: p. 1153-1159.
13. Janowczyk, A. and A. Madabhushi, *Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases.* Journal of Pathology Informatics, 2016. **7**(29).
14. Gua, J., et al., *Recent advances in convolutional neural networks.* Pattern Recognition, 2018. **77**: p. 354-377.
15. LeCun, Y., et al., *Gradient-based learning applied to document recognition.* Proceeding of the IEEE, 1998. **86**(11): p. 2278-2323.
16. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, in *Advances In Neural Information Processing Systems*, F. Pereira, et al., Editors. 2012, Curran Associates, Inc.: Red Hook, NY. p. 1097-1105.
17. Zeiler, M.D. and R. Fergus, *Visualizing and understanding convolutional networks*, in *Computer Vision – ECCV 2014. ECCV 2014, Lecture Notes in Computer Science*, D. Fleet, et al., Editors. 2014, Springer, Cham: Berlin.
18. Simonyan, K. and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*. 2014, Cornell University: arXiv:1409.1556v6
19. Szegedy, C., et al., *Going deeper with convolutions*, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 2015. p. 1-9.
20. He, K., et al., *Deep residual learning for image recognition*, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, IEEE: Las Vegas, NV. p. 770-778.
21. Yosinski, J., et al., *How transferable are features in deep neural networks?* 2014, Cornell University: arXiv:1411.1792.
22. Shin, H.-C., et al., *Deep Convolutional Neural Networks For Computer-Aided Detection: Cnn Architectures, Dataset Characteristics And Transfer Learning.* IEEE Transactions on Medical Imaging, 2016. **35**(5): p. 1285-1298.
23. Pan, Y., et al., *Brain tumor grading based on neural networks and convolutional neural networks*, in *37th IEEE Engineering in Medicine and Biology Society (EMBC)* 2015, IEEE. p. 699–702.
24. Nanni, L., et al., *Bioimage Classification with Handcrafted and Learned Features.* IEEE/ACM transactions on computational biology and bioinformatics, In Press.
25. van Ginneken, B., et al., *Off-the-shelf convolutional neural network features for pulmonary nodule detection in computed tomography scans*, in *IEEE 12th International Symposium on Biomedical Imaging (ISBI)*. 2015, IEEE.
26. Li, R., et al., *Deep learning based imaging data completion for improved brain disease diagnosis*, in *Medical Image Computing and Computer-Assisted Intervention*. 2014. p. 305-312.
27. Kumar, A., et al., *An ensemble of fine-tuned convolutional neural networks for medical image classification.* IEEE Journal of Biomedical and Health Informatics, 2017. **21**(1): p. 31-40.
28. Szegedy, C., et al., *Inception-v4, inception-resnet and the impact of residual connections on learning*, in *arxiv.org*. 2016, Cornell University: https://arxiv.org/pdf/1602.07261.pdf. p. 1-12.
29. Jantzen, J., et al., *Pap-smear benchmark data for pattern classification*, in *Nature inspired Smart Information Systems (NiSIS)*. 2005: Albufeira, Portugal. p. 1–9.
30. Shamir, L., et al., *IICBU 2008: a proposed benchmark suite for biological image analysis.* Medical & Biological Engineering & Computing, 2008. **46**(9): p. 943–947.
31. Junior, G.B., et al., *Classification of breast tissues using Moran's index and Geary's coefficient as texture signatures and SVM.* Computers in Biology and Medicine, 2009. **39**(12): p. 1063-1072.
32. Cruz-Roa, A., J.C. Caicedo, and F.A. González, *Visual pattern mining in histology image collections using bag of features.* Artificial Intelligence in Medicine, 2011(52): p. 91-106.
33. Tan, X. and B. Triggs, *Enhanced local texture feature sets for face recognition under difficult lighting conditions.* Analysis and Modelling of Faces and Gestures, 2007. **LNCS 4778**: p. 168-182.
34. Nanni, L., S. Brahnam, and A. Lumini, *A very high performing system to discriminate tissues in mammograms as benign and malignant.* Expert Systems with Applications, 2012. **39**(2): p. 1968-1971.

ACCEPTED MANUSCRIPT

35. Guo, Z., L. Zhang, and D. Zhang, *A completed modeling of local binary pattern operator for texture classification.* IEEE Transactions on Image Processing, 2010. **19**(6): p. 1657-1663
36. Nosaka, R. and K. Fukui, *HEp-2 cell classification using rotation invariant co-occurrence among local binary patterns.* Pattern Recognition in Bioinformatics, 2014. **47**(7): p. 2428-2436.
37. Nanni, L., et al., *Review on texture descriptors for image classification*, in *Computer Vision and Simulation: Methods, Applications and Technology*, S. Alexander, Editor. 2016, Nova Publications: Hauppauge, NY.
38. Zhu, Z., et al., *An adaptive hybrid pattern for noise-robust texture analysis.* Pattern Recognition, 2015. **48**: p. 2592-2608.
39. Serra, G., et al., *Gold: Gaussians of local descriptors for image representation.* Computer Vision and Image Understanding, 2015. **134**(May): p. 22–32.
40. Dalal, N. and B. Triggs, *Histograms of oriented gradients for human detection*, in *9th European Conference on Computer Vision*. 2005: San Diego, CA.
41. Strandmark, P., J. Ulén, and F. Kahl, *HEp-2 Staining Pattern Classification*, in *International Conference on Pattern Recognition (ICPR2012)*. 2012.
42. Wang, Q., et al., *Towards effective codebookless model for image classification.* Pattern Recognition, 2016. **59**: p. 63-71.
43. Song, T. and F. Meng, *Letrist: locally encoded transform feature histogram for rotation-invariant texture classification.* IEEE Transactions on circuits and systems for video technology, 2017. **PP**(99).
44. Ojala, T., M. Pietikainen, and T. Maeenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002. **24**(7): p. 971-987.
45. Ojansivu, V. and J. Heikkila, *Blur insensitive texture classification using local phase quantization*, in *ICISP*. 2008. p. 236–243.
46. Chan, C., et al., *Multiscale local phase quantisation for robust component-based face recognition using kernel fusion of multiple descriptors.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013. **35**(5): p. 1164-1177.
47. Nanni, L., et al., *Ensemble of local phase quantization variants with ternary encoding*, in *Local Binary Patterns: New Variants and Applications*, S. Brahnam, et al., Editors. 2014, Springer-Verlag: Berlin. p. 177-188.
48. Kannala, J. and E. Rahtu, *Bsif: Binarized statistical image features.*, in *21st International Conference on Pattern Recognition (ICPR 2012)*. 2012: Tsukuba, Japan. p. 1363-1366.
49. Nanni, L., S. Brahnam, and A. Lumini, *A local approach based on a Local Binary Patterns variant texture descriptor for classifying pain states.* Expert Systems with Applications, 2010. **37**(12): p. 7888-7894.
50. Zhu, C. and R. Wang, *Local multiple patterns based multiresolution gray-scal and tortation invariant texture classification.* Information Sciences, 2012. **187**: p. 93-108.
51. Bianconi, F., et al., *Performance analysis of colour descriptors for parquet sorting.* Expert System with Applications, 2013. **40**(5): p. 1636-1644.
52. Nowak, E., F. Jurie, and B. Triggs, *Sampling Strategies for Bag-of-Features Image Classification*, in *European Conference on Computer Vision (ECCV)*, A. Leonardis, H. Bischof, and A. Prinz, Editors. 2006. p. 490–503.
53. Boland, M.V. and R.F. Murphy, *A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells.* BioInformatics, 2001. **17**(12): p. 1213-223.
54. Dimitropoulos, K., et al., *Grading of invasive breast carcinoma through Grassmannian VLAD encoding.* PLoS ONE, 2017. **12**: p. 1–18.
55. Moccia, S., et al., *Confident texture-based laryngeal tissue classification for early stage diagnosis support.* Journal of Medical Imaging (Bellingham), 2017. **4**(3): p. 34502.
56. Kather, J.N., et al., *Multi-class texture analysis in colorectal cancer histology.* Scientific Reports, 2016. **6**: p. 27988.

ACCEPTED MANUSCRIPT

57. Nanni, L., et al., *Ensemble of Handcrafted and Deep Learned Features for Cancer Data Analysis.* In review.
58. Song, Y., et al., *Bioimage classification with subcategory discriminant transform of high dimensional visual descriptors.* BMC Bioinformatics, 2016. **17**: p. 465.
59. Nanni, L., S. Ghidoni, and S. Brahnam, *Handcrafted vs non-handcrafted features for computer vision classification* Pattern Recognition 2017. **71**: p. 158-172.
60. Smeeton, N.C., *Early History of the Kappa Statistic.* 1985. Biometrics. 41: 795. JSTOR 2531300.
61. L. Kuncheva and C. J. Whitaker, Measures of Diversity in Classifier Ensembles and their Relationship with the ensemble accuracy, *Machine Learning, vol. 51, no. 2, pp. 181-207, 2003*

ACCEPTED MANUSCRIPT