Expert Systems With Applications 84 (2017) 1–11

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Convolutional Neural Network approaches to granite tiles classification

Anselmo Ferreira [a,*], Gilson Giraldi [b]

[a] Shenzhen Key Laboratory of Media Security, College of Information Engineering, Shenzhen University, Nanhai Ave 3688, Shenzhen, Guangdong 518060, PR China

[b] National Laboratory for Scientific Computing (LNCC), Av. Getúlio Vargas 333, Quitandinha, Petrópolis, Rio de Janeiro 22230000, Brazil

ARTICLE INFO

Article history:
Received 1 February 2017
Revised 26 April 2017
Accepted 27 April 2017
Available online 4 May 2017

Keywords:
Granite classification
Convolutional Neural Networks
Deep learning

ABSTRACT

The quality control process in stone industry is a challenging problem to deal with nowadays. Due to the similar visual appearance of different rocks with the same mineralogical content, economical losses can happen in industry if clients cannot recognize properly the rocks delivered as the ones initially purchased. In this paper, we go toward the automation of rock-quality assessment in different image resolutions by proposing the first data-driven technique applied to granite tiles classification. Our approach understands intrinsic patterns in small image patches through the use of Convolutional Neural Networks tailored for this problem. Experiments comparing the proposed approach to texture descriptors in a well-known dataset show the effectiveness of the proposed method and its suitability for applications in some uncontrolled conditions, such as classifying granite tiles under different image resolutions.

1. Introduction

Natural Stone is the first building material used by the humanity and it continues to be used on new structures. Among the natural stones, the granite has received a particular interest due to its beauty and strength. Granite is an intrusive igneous rock widely distributed throughout Earth's crust at a range of depths up to 31 miles (50 km) according to University of Tennessee (2008).

Since there are different colors of granite, their denomination varies depending on the country. Even with some standards being created such as the one from the European Committee for Standardization (CEN) (2009) and, more recently, from the Stone Industry SA Standards Board (2015), there are some cases on which the visual appearance of different granites with same mineralogical content may not differ significantly, making it time consuming and economically expensive to the stone industry to check, slab by slab, if the product colors to be delivered to clients are the same as the colors purchased. Most of the times, the quality assessment is done by human-skilled professionals in a subjective procedure that can fail. The international operations of stone industries require this procedure to be faster and more accurate.

Automated systems can play an important role in this scenario. Granite companies are interested in computer vision allied with machine learning procedures able to identify patterns on granite images, using these patterns to sort and grade granite products based on their visual appearance, helping in this way the product traceability and warehouse management. Also, an on-site tool that can help to identify the real type of a product can help solving possible misunderstoods between customers and manufacturers at the time of a delivery.

This challenge has received some attention by the scientific literature in the past years, and some studies of the application of well-known colors and texture descriptors were performed, such as in the works made by Kurmyshev, Sánchez-Yáñez, and Fernández (2013), Bianconi and Fernández (2006), Lepisto, Kunttu, and Visa (2005), Araújo, Martínez, Ordóñez, and Vilán (2010) and Bianconi, González, Fernández, and Saetta (2012). However, most of them are general purpose descriptors and were created for a controlled scenarios, where the slabs used for experiments have the same size and images acquired have the same resolution. Descriptors applied on granite slabs classification to date generate what is called *hand-crafted features* created by feature engineering, built based on a behavior that is expected to happen in all granite images. Finally, there is no study regarding the application of these descriptors in slabs pieces.

In this paper, we go beyond the use of hand-crafted features and propose what is, as far as we know, the first data-driven approach to identify granite patterns. The proposed technique uses deep Convolutional Neural Networks (CNNs) with different architectures, trained to classify the intrinsic patterns of granite tiles based on texture and color. Instead of designing a very deep neural network to be applied on high resolution images, a process that

* Corresponding author.
E-mail addresses: anselmo@szu.edu.br, anselmo.ferreira@gmail.com (A. Ferreira), gilson@lncc.br (G. Giraldi).

http://dx.doi.org/10.1016/j.eswa.2017.04.053
0957-4174/

requires too many layers in the architecture and also thousands of images to train the network, our approach uses lightweight neural networks on small patches of granite images, taking into account the majority voting of patches classification for the images classification, making the proposed approach able to classify slabs of different sizes, image resolutions and even slab pieces. Experiments comparing our approach against some hand-crafted descriptors and pre-trained networks show the effectiveness of the proposed technique.

In summary, the main contributions of this paper are:

1. Design and development of ad-hoc CNNs for granite tiles classification.
2. Application of CNNs on multiple image data, represented by small patches located in regions of interest from granite slab images, to learn features that lead to high recognition accuracy by using majority voting on patches classification.
3. Validation of the proposed methodology with other descriptors not used before for granite classification.

The remainder of this paper is organized as follows: Section 2 discusses related work about color and texture descriptors in the literature and also some studies applying them to the granite classification. Section 3 shows the basic concepts of CNNs, which are necessary to understand the proposed approach. Section 4 presents our approach for granite color classification and Section 5 reports all the details about the experimental methodology used for validating the proposed method against the existing counterparts in the literature. Section 6 shows the performed experiments and results and Section 7 reports our final considerations and proposals for future work.

2. Related work

The problem of classifying materials by their type can be regarded as a problem of classifying textures, colors and geometrical features. To create solutions in this regard, the scientific literature focused on applying computer-vision based approaches allied with machine learning to grade different materials.

In the specific case of granite rocks, Ershad (2011) presented a method based on Primitive Pattern Units, a morphological operator which is applied to each color channel separately. Statistical features are then used to discriminate different classes of natural stone such as granite, marble, travertine and hatchet. Kurmyshev et al. (2013) used coordinated clusters representation (CCR) for classifying granite tiles of the type “Rosa Porriño”. Bianconi and Fernández (2006) employed different Gabor filter banks for granite classification. Lepisto et al. (2005) used a similar approach, but applied in each color channel separately. Araújo et al. (2010) employed spectrophotometer to capture spectral data at different regions of interest in granite tiles and used Support Vector Machines to grade them. Fernández, Ghita, González, Bianconi, and Whelan (2011) studied how common image descriptors such as Local Binary Patterns, Coordinated Clusters Representation and Improved Local Binary Patterns acted in granite image classification after rotation conditions.

Most approaches described in literature are based on textural features alone. According to the study of Bianconi et al. (2012) this is somewhat surprising, since the visual appearance of granite tiles strongly depends on both texture and color. To this end, they perform a study using several textures and color descriptors in five different classifiers, showing that combining color and texture features and classifying them on Support Vector Machines classifiers outperforms previous methods based on textural features alone.

Finally, in a recent work, Bianconi, Bello, Fernández, and González (2015a) investigated the problem of choosing the adequate color representation for granite grading. They discussed pros and cons of different color spaces for granite classification by performing experiments using very simple color descriptors: mean, mean + standard deviation, mean + moments from 2nd to 5th, quartiles and quintiles of each color channel. They showed that, depending on the classifier used, some color spaces are better than others for classification, such as Lab and Luv spaces for the linear classifier.

As can be noticed, solutions presented for granite color classification are based only on feature engineering. In other words, these approaches are driven by patterns that are supposed to happen over all investigated images and also require different expert knowledge. Veering away from these methods, we propose a deep learning based approach, which extracts meaningful discriminative patterns straight from the data by using small image patches instead of using ordinary feature engineering in whole images. To do this, our approach exploits the advantages of back-propagation of error used in Convolutional Neural Networks to automatically learn these discriminant features present on the image textures. Before we discuss our proposed method to perform granite grading based on deep learning, it is worth discussing some basic concepts about deep neural networks in the next section.

3. Basic concepts

Convolutional Neural Networks have been attracting a considerable attention by the computer vision research community recently, mainly because of their effective results in several image classification tasks, outperforming even humans in certain situations according to He, Zhang, Ren, and Sun (2015), winning several image classification challenges, such as the ImageNet image recognition contest as described by Krizhevsky, Sutskever, and Hinton (2012). Pioneered by the work of Lecun, Bottou, Bengio, and Haffner (1998), CNNs are regarded as a deep learning application to images and, as so, they simulate the activity in layers of neurons in the neocortex, the place where most of thinking happens, according to Hof (2013). So, the network learns to recognize patterns by highlighting in its layers the edges and pixel behaviors that are commonly found in different images.

The main benefit of using CNNs with respect to traditional fully-connected neural networks is the reduced amount of parameters to be learned. Convolutional layers made of small size kernels allow an effective way of extracting high-level features that are fed to fully-connected layers. The training of a CNN is performed through back-propagation and stochastic gradient descent as Rumelhart, Hinton, and Williams (1986) describes. The misclassification error drives the weights update of both convolutional and fully-connected layers. The basic layers of a CNN are listed below:

1. *Input layer*: where data is fed to the network. Input data can be either raw image pixels or their transformations, whichever better emphasize some specific aspects of the image.
2. *Convolutional layers*: contain a series of filters with fixed size used to perform convolution on the image data, generating what is called *feature map*. These filters can highlight some patterns helpful for image characterization, such as edges, textures, etc.
3. *Pooling layers*: these layers ensure that the network focuses only on the most important patterns. They summarize the data by sliding a window across the feature maps, applying some linear or non-linear operations on the data within the window, such as local mean or max, reducing the dimensionality of the feature maps used by the following layers.
4. *Rectified Linear Unit (ReLU)*: ReLU layers are responsible for applying a non-linear function to the output x of the previous layer, such as $f(x) = max(0, x)$. According to Krizhevsky et al. (2012), they can be used for fast convergence in the training of

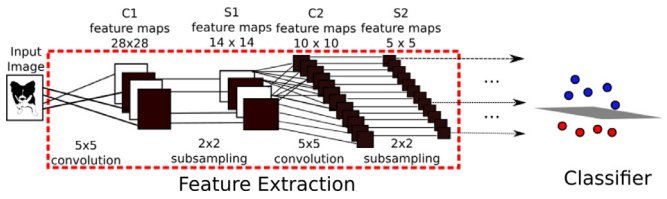

Fig. 1. Common architecture arrangement of a CNN. The input image is transformed into feature maps by the first convolution layer C1. A pooling stage S1 reduces the dimensions across the feature maps. The same process is repeated for layers C2 and S2. Finally a classifier is trained with the data generated by layer S2.

CNNs, speeding-up the training as they deal with the vanishing gradient problem by keeping the gradient more or less constant in all network layers.

5. *Fully-connected layers*: used for the understanding of patterns generated by the previous layers. Neurons in this layer have full connections to all activations in the previous layer. They are also known as the inner product layers. After trained, transfer learning approaches can extract features in these layers to train another classifier.
6. *Loss layers*: specify how the network training penalizes the deviation between the predicted and true labels and is normally the last layer in the network. Various loss functions appropriate for different tasks can be used: Softmax, Sigmoid Cross-Entropy, Euclidean loss, among others.

Fig. 1 depicts one possible CNN architecture. The type and arrangement of layers vary depending on the target application.

Although very powerful at representing patterns present in the data, the main drawback of deep learning is the fact that common CNNs normally need thousands or even millions of labeled data for training. This is an unfeasible condition in many applications due to the lack of training data and to the amount of time needed to train a model. In this work, we present an alternative approach that deals with this requirement by considering using image tiles (multiple data) of the input images in the networks, as described further in Section 4.

4. Proposed method

In this paper, we propose a methodology for granite tiles classification through the application of CNNs. The proposed pipeline relies solely on the data to learn the texture patterns therein. We solve the CNN requirement of large training sets by applying the neural network on small patches of images, this way the CNN will learn discriminative features, instead of relying on feature engineering. Fig. 2 shows the pipeline of the proposed approach. We discuss each step of this pipeline in the next paragraphs.

4.1. Conversion to grayscale

As some proposed CNNs in this paper include networks that classify grayscale images to identify textures, in this optional preprocessing step, the image is converted to gray-level, removing color information from the analysis. Given an input RGB image I, with one pixel represented by $I(i, j) = (R(i, j), G(i, j), B(i, j)$, its conversion to graylevels happens using Eq. (1) below:

$$C(i, j) = 0.2989R(i, j) + 0.5870G(i, j) + 0.1140B(i, j), \quad (1)$$

where C is the image converted to grayscale and used as input to the network and R, G and B are the color channels from the initial image.

4.2. Patching

In the first step of our proposed approach (labeled as step A in Fig. 2), the image is divided into tiles of interest. These tiles are out of image borders (no artificial pixel filling is done in the input images) and will be used as input to the CNNs. This procedure is useful for the following reasons: (i) generate more data to CNN training (ii) use majority voting in image tiles classification for testing a given input image and (iii) possibility to use different image resolutions to train and test the CNN, as only image tiles of interest will be used in the network. This is an important procedure because, normally, CNNs are designed for given resolution input images. However, this is not necessary in our approach because only small granite blocks are used as input. In this step the CNN is investigating several regions of interest (which we call multiple data) to classify the entire image. We use 28 × 28 grayscale images and 32 × 32 color images in our CNNs, described in the next subsection. (Figs. 5 and 6)

4.3. Recognition by Convolutional Neural Networks

In the second step, labeled as step B in Fig. 2, we use different Convolutional Neural Networks to recognize patterns of image patches. For this, in a training step, we apply tiles from training images through the networks to estimate the filter weights for a better classification. Then, we do what is called *transfer learning* as described by Johnson and Karpathy (2016) and Yosinski, Clune, Bengio, and Lipson (2014), using the already trained network as a feature extractor in the train and test stages. To do that we extract the last layer (the softmax layer) of the already trained network and then the new output will be feature vectors generated by the last convolutional layer. Finally, these image characterizations will be used by another classifier in the training and testing steps. The number of networks used are four and their architectures are as follows.

- *MNIST1 network:* based in a design used in the MNIST dataset digit recognition challenge (VLFEAT, 2016), it has an input layer which requires 28 × 28 grayscale input images and is composed by eight other layers: four convolutional layers, two pooling layers, one RELU layer and one fully-connected layer.
- *MNIST2 network:* we extended MNIST1 network, creating a new one composed by eleven layers: one input layer, five convolutional layers, three pooling layers, one RELU layer and a final fully connected layer.
- *MNIST3 network:* we extended even more our initial MNIST1 network, creating a new one composed by thirteen layers: one input layer, six convolutional layers, four pooling layers, one RELU layer, and a final fully connected layer.
- *CIFAR network:* based in a design used in the CIFAR image recognition challenge (VLFEAT, 2016), it has an input layer which requires 32 × 32 RGB input images and is composed by twelve other layers: five convolutional layers, three pooling layers, four RELU layers and one fully-connected layer.

These networks will generate, respectively, feature vectors with 500, 256, 256 and 64 dimensions respectively, which we aim to use in another classifier. Fig. 3 shows the learned filters in the first layer of each network. Figs. 4–7 show the output of these filters on networks MNIST1, MNIST2, MNIST3 and CIFAR respectively and their power to discriminate by convolutions some granite blocks present on the dataset built in the work of Bianconi et al. (2015a).

Additionally to the use of these networks individually, we propose the fusion of them using for that two approaches:

1. *Early fusion:* The feature vector of a block will be the fusion of descriptions (feature vectors) from the three networks. Concatenating such feature vectors will yield a final feature vector

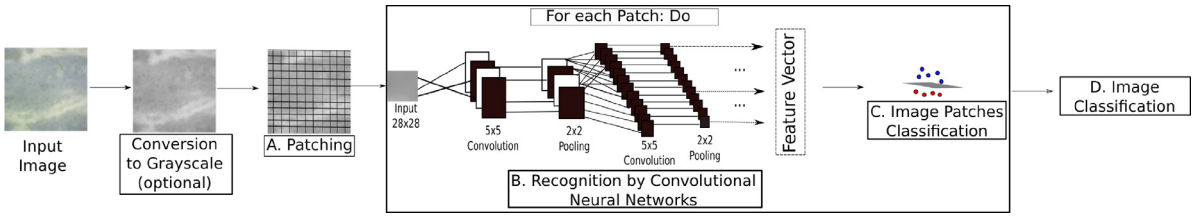

Fig. 2. Pipeline of the proposed approach based on Convolutional Neural Networks for granite classification.

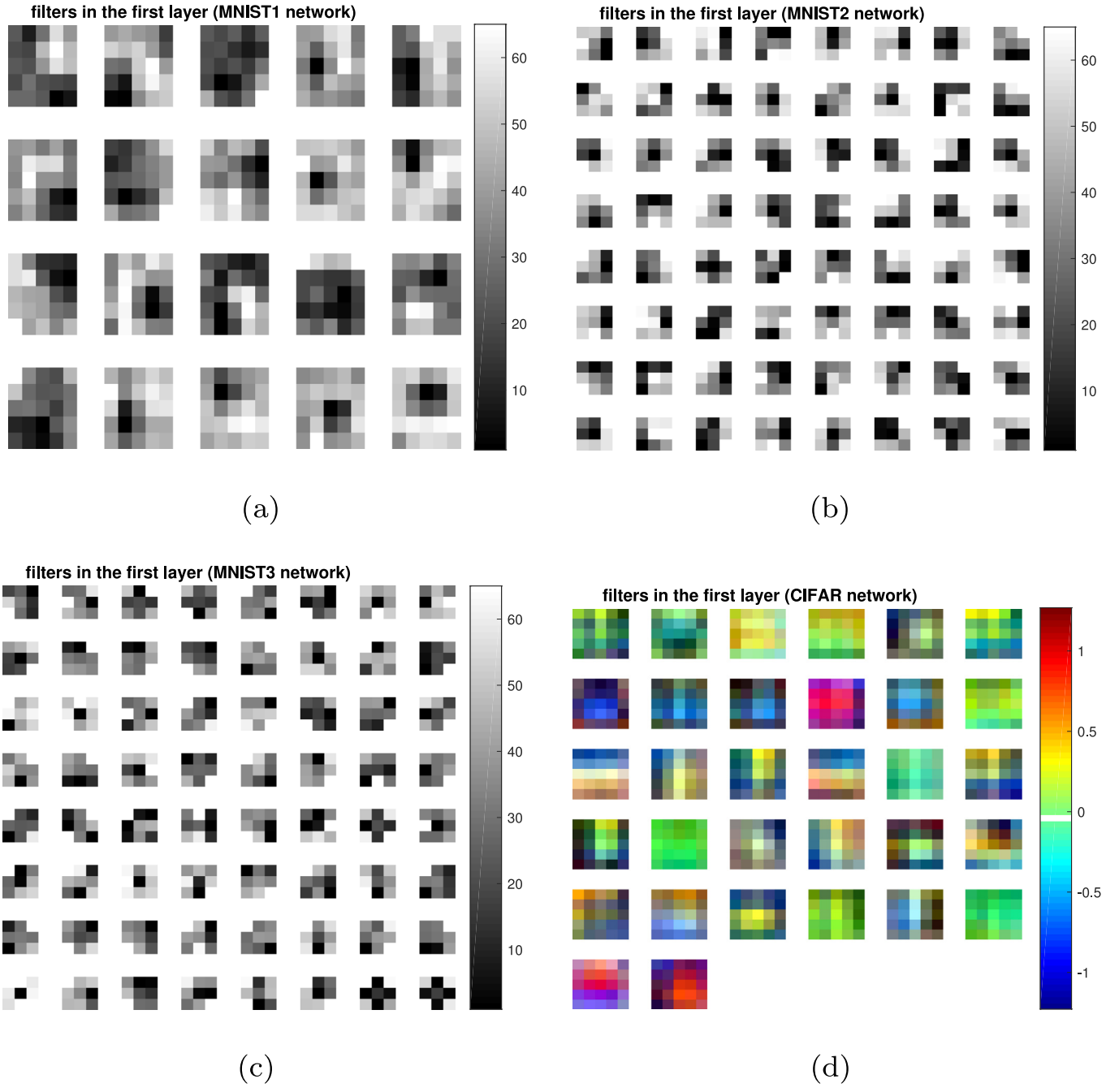

Fig. 3. Filter weights of the first layer from (a) MNIST1 network (b) MNIST2 network (c) MNIST3 network and (d) CIFAR network.

of 500 + 256 + 256 = 1012 dimensions, which will be used in a given classifier.

2. *Late fusion:* the classification of a tile will happen by majority voting of the three classifiers outputs, fed by the feature vectors generated by the three networks applied on that tile.

Figs. 8 and 9 show, respectively, the early and late fusion approaches pipelines for granite image classification using the MNIST-based CNNs considered in this paper.

If the first approach (early fusion) is chosen, the next step is normalizing (or scaling) the concatenated feature vectors. There are several approaches to do this, but the one we choose is the

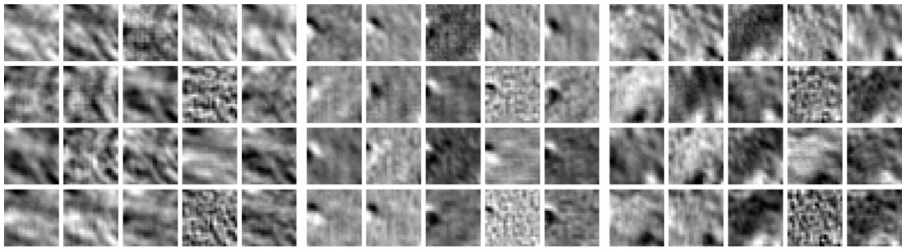

Fig. 4. MNIST1 first layer convolutions used to identify (a) "Giallo Veneziano" granite block (b) "Sky Brown" granite block and (c) "Verde Oliva" granite block.

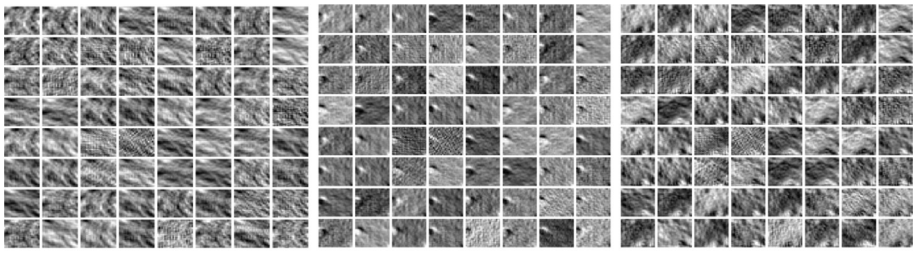

Fig. 5. MNIST2 first layer convolutions used to identify (a) "Giallo Veneziano" granite block (b) "Sky Brown" granite block and (c) "Verde Oliva" granite block.

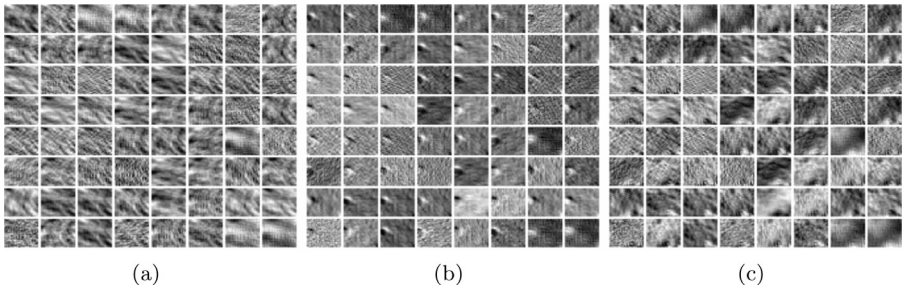

Fig. 6. MNIST3 first layer convolutons used to identify (a) "Giallo Veneziano" granite block (b) "Sky Brown" granite block and (c) "Verde Oliva" granite block.

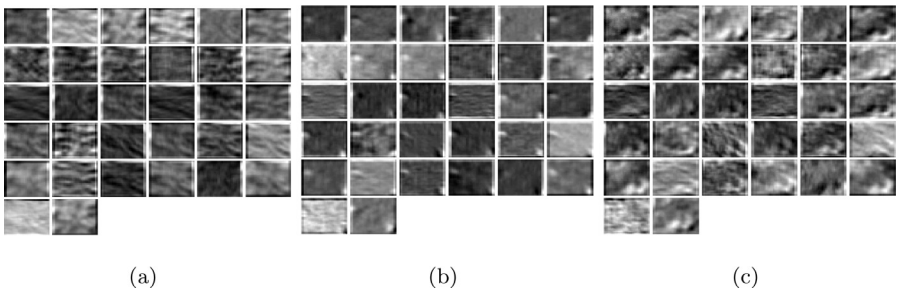

Fig. 7. CIFAR first layer convolutons used to identify (a) "Giallo Veneziano" granite block (b) "Sky Brown" granite block and (c) "Verde Oliva" granite block. Convolutions are represented in gray for a better visualization.

simplest, which is dividing each vector by its norm. Given a feature vector $\vec{V}$, its norm p is calculated as:

$$||\vec{V}||_p = \left(\sum_{i=1}^{n} |\vec{V}(i)^p| \right)^{\frac{1}{p}}, \tag{2}$$

where i is the ith vector element and n is the number of vector elements.

For our application, we choose $p = 2$ and the norm generated is called *Euclidean Norm*. The final feature vector $\vec{Vf}$ is

$$\vec{Vf} = \frac{\vec{V}}{||\vec{V}||_2}. \tag{3}$$

Using this approach the feature vectors components will be scaled in such a way that each vector magnitude is always one. This is important because, as the feature vectors to be concatenated come from different sources, the range of all features should be normalized so that each feature contributes approximately proportionately to the final feature vector.

4.4. Image patches classification

In the training and testing steps, we apply the feature vectors generated by the previously trained networks in another classifier. So, in the image patches classification step (labeled as step C in Fig. 2), we choose the 1st Nearest Neighbor (1NN) classifier to clas-

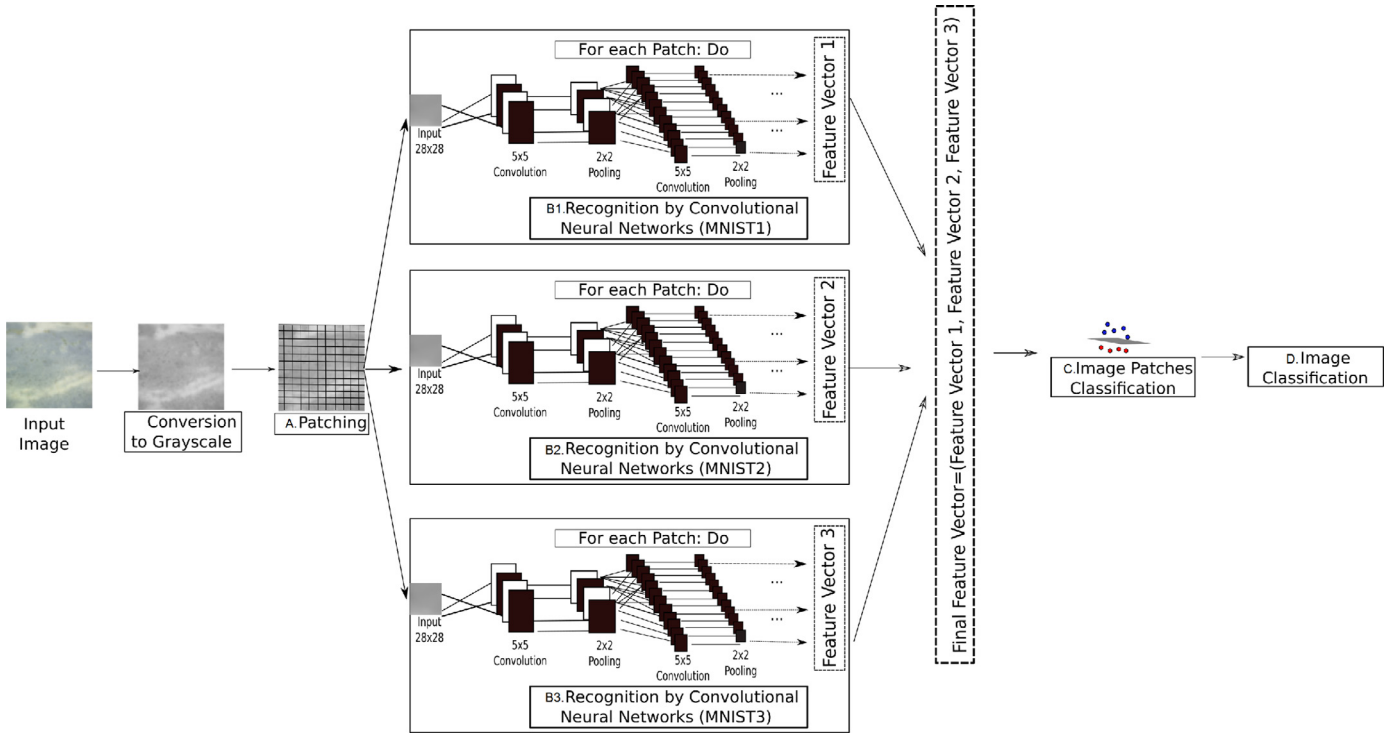

Fig. 8. Pipeline of the early fusion of Convolutional Neural Networks for granite classification. In this fusion approach, three pre-trained networks with different architectures (MNIST1, MNIST2 and MNIST3) are applied in the same tiny input data (blocks), generating feature vectors that are concatenated and normalized to classify a given 28×28 block using information from these three networks. The final classification for an image uses the majority voting of its blocks.

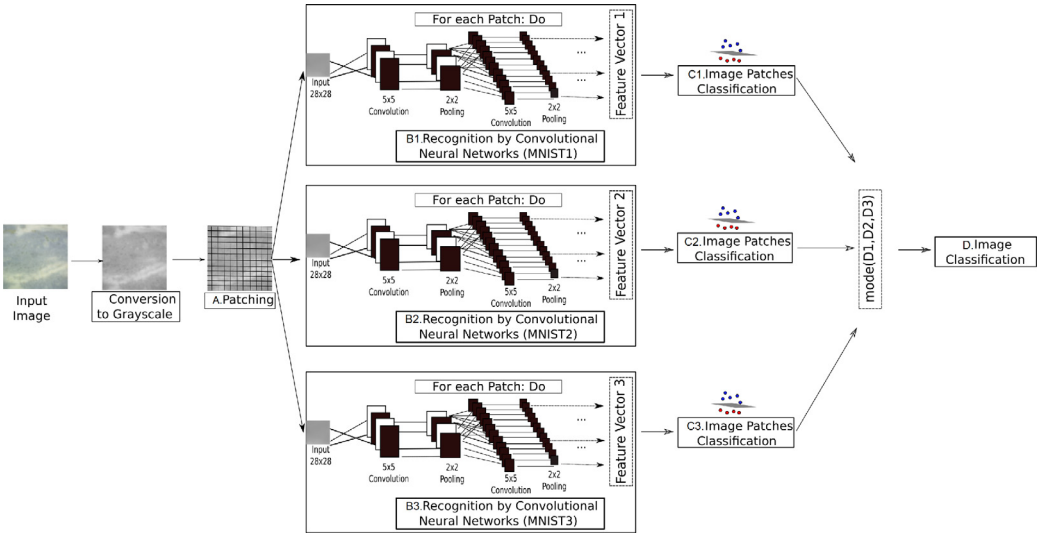

Fig. 9. Pipeline of the late fusion of Convolutional Neural Networks for granite classification. In this fusion approach, three pre-trained networks with different architectures are applied in the same tiny input data (blocks), extracting feature vectors and classifying them individually. The final classification of a block is the majority voting of labels generated by the three networks. The same happens on all the classified blocks to classify the whole image.

sify the tiny image blocks from input images. Basically, this classifier works by assigning a testing sample to the class of its nearest neighbor defined in a training step. In the end of this process, each valid block from the image is classified.

4.5. Image classification

After the individual patches classification, in step D in Fig. 2) we will classify the image by analyzing, in the blocks classification, which is the most predicted class. Given a vector $\vec{x} = \{output_1, \ldots, output_b\}$ which contains the classification of b blocks in the image, the predicted class of a granite image G is

$$class(G) = mode(\vec{x}), \tag{4}$$

where $mode(\vec{x})$ is the most predicted class in vector $\vec{x}$.

5. Experimental setup

Before we discuss our experimental results, we present in this section the materials and methods used to compare the proposed method against its counterparts in the literature. In the following subsections we show the granite tile dataset used, the methodology and metrics used in the experiments to assess the classifica-

tion performance, parameters used in our proposed approach and the state-of-the art implementations used.

5.1. Image dataset

We used the same granite tiles dataset applied in the work of Bianconi et al. (2015a). It contains 1000 RGB images with 1500 × 1500 dimensions, subdivided in 25 classes containing 40 images per class. The first 100 images were acquired naturally by a specific scanner. The other 900 were created by rotating the initial 100 images in 10°, 20°, 30°, 40°, 50°, 60°, 70°, 80° and 90°.

To be suitable for our proposed approach, we subdivide these images in 28 × 28 valid blocks (*i.e.*, outside image borders) to be applied on MNIST1, MNIST2 and MNIST3 networks, creating this way 2809 valid blocks per image. So, the dataset used in the experiments contains $2809 \times 1000 = 2809.000$ tiny grayscale images. To use the proposed CIFAR network on RGB images we subdivide the dataset images on 32 × 32 valid blocks, creating this way 2116 valid RGB blocks per image and a total of $2116 \times 1000 = 2116.000$ tiny color images.

Doing these procedures we artificially create a great amount of small images to be used in small networks, eliminating the requirement of big networks applied in a big amount of high resolution images. By subdividing the test image into these blocks, we can increase classification accuracy by doing majority voting of these blocks, classifying a big image by its small parts.

5.2. Methodology and metrics

We validate the proposed method using two experimental scenarios: assessing the performance of classifying whole images (by majority voting of blocks) and also small blocks. For the first scenario, we consider a 5 × 2 cross-validation protocol, on which we replicate the traditional 1 × 2 cross-validation protocol five times (thus 5 × 2). In each of them, we divided the set of images D randomly into equal subsets of images D_1 and D_2, on which $D_1 \cup D_2 = D$ and $D_1 \cap D_2 = \emptyset$. Then, the classifier is trained with D_1 images as input and tested on D_2 images and then the inverse is done. If the process is repeated five times, metrics can be reported after 10 rounds of experiments. The experimental results using our approaches will be in terms of high resolution images classification accuracy after majority voting of low resolution image blocks.

Using the 5 × 2 cross validation in our scenario, if our proposed approach acts on 2889 28 × 28 valid (outside borders) grayscale blocks and 2116 32 × 32 valid RGB blocks from each of the 1000 images in our dataset, we will end up with $500 \times 2809 = 1404.500$ grayscale blocks (for MNIST based networks) and $500 \times 2116 = 1058.000$ RGB blocks (for the CIFAR based network) used for training the classifier, and the same number of blocks to test the classifier. This happens in all 10 rounds of experiments. So, our training ratio will always be 50% of the data and the other 50% of data will be the testing ratio. According to a study conducted by Dietterich (1998), the 5 × 2 cross-validation is considered an optimal experimental protocol for learning algorithms.

For the second experimental scenario, we show the classification performance of image blocks. For this case, we used one combination of training and test images. We intend to show using this configuration how the proposed approaches and some state of the art behave on classifying only small image resolution images containing granite patterns. In this case results are reported on block basis (without majority voting).

We used the accuracy method to evaluate the performance of the algorithms tested. In a multi-class problem with c classes, the classification results may be represented in a $c \times c$ confusion matrix M. In this case, the main diagonal contains the true positives while the other entries contain either false positives or false negatives. So, the accuracy of an experiment round is calculated as

$$Accuracy = \frac{\sum_{i=1}^{c} M(i,i)}{\sum_{i=1}^{c}\sum_{j=1}^{c} M(i,j)}. \quad (5)$$

where i and j are line and column indexes of M, respectively.

In the 5 × 2 cross validation protocol, one confusion matrix is yielded per experiment. Therefore, we present results by averaging accuracies from these matrices.

The other set of metrics shown in the experiments are commonly applied in CNN image recognition tasks and will be used to find the number of epochs to train our proposed networks. These are called top-1 and top-5 errors and are widely used in the ImageNet Large Scale Visual Recognition Challenge and also for other tasks as reported by Russakovsky et al. (2015). Commonly, CNNs returns for one image i the j most probable classes $\{c_{i1}, \ldots, c_{ij}\}$ in the output of its last layer as the prediction for that image. If we define c_{ij} the prediction of an image i and C_i its ground truth, the prediction is considered correct if $c_{ij} = C_i$ for some j. If we define the error of a prediction $d_{ij} = d(c_{ij}, C_i)$ as 0 if $c_{ij} = C_i$ and 1 otherwise, the error of an algorithm is the fraction of test images on which the algorithm makes a mistake,:

$$err = \frac{1}{N}\sum_{i=1}^{N} min_j(d(c_{ij}, C_i)), \quad (6)$$

where N is the total of images used in the validation of a CNN.

The difference of top-1 and top-5 errors is the value used for j: in top-1 $j = 1$ and top-5 $j = 5$. In other words, top-1 error compares if the top class (the one having the highest probability, or c_{i1}) is the same as the target label C_i and top-5 scores if the target label is one of the top 5 predictions (the 5 ones with the highest probabilities, or $\{c_{i1}, \ldots, c_{i5}\}$). The top-5 error is normally used only for images with more than one object and will not be considered in our scenario (we use only top-1 error).

5.3. Implementation aspects of the proposed approaches

To implement our proposed approaches we used the CNNs library for MATLAB available at VLFEAT (2016). We used the CIFAR and MNIST1 architectures, training them from scratch. We also extended layers from network MNIST1, creating this way new networks MNIST2 and MNIST3. To create feature extractors, we remove the last layer from the trained networks and feed them again with training images, generating training feature vectors to be the input for the 1NN classifier. This same last process happens again for testing images.

For the early fusion approach, applied only to grayscale input images networks (MNIST related CNNs), we used the three trained networks from scratch to extract train and test feature vectors, concatenating the vectors from these three networks and normalizing them by Eq. 3. Finally, these vectors are the input of the 1NN classifier and, to classify the image we perform majority voting of blocks. We denominate this approach as $\{MNIST1, MNIST2, MNIST3\}\_concat$.

Finally, we test the complementarity of the three grayscale networks (MNIST related networks), using majority voting for a block. To do this we apply, in the test phase, each network individually in the same block. The three feature vectors will be then fed into three independent 1NN classifiers, previously trained with feature vectors from their corresponding CNNs. The result for a block is the majority voting of the three 1NN classifications. Then, a final majority voting of blocks will define the class of the test image. We label this approach as $\{MNIST1, MNIST2, MNIST3\}\_vote$.

The MNIST based networks were trained using batches of images containing 100 images, with learning rate fixed on 0.001. We

used the stochastic gradient descent with *momentum* equals to 0.9 and weighting decay equals to 0.0005 without dropout. The CIFAR network was trained using batches of images containing 100 images, with stochastic gradient descent and *momentum* equals to 0.9. The weighting decay is 0.0001 without dropout. Upon acceptance of this paper, all the source code of the proposed approaches will be available at GitHub[1]

5.4. Baselines

To compare our approach against the state of the art we initially chose ten texture descriptors to be used as baselines, some of them were already used for Granite Image classification and others were applied for other texture recognition applications. The first five baselines approaches can be regarded as general purpose texture descriptors used in several applications. The first three chosen are the statistics of Gray-Level Co-occurrence Matrices (GLCMs) from Haralick, Shanmugam, and Dinstein (1973) (we label this approach as *GLCM* in the experiments), the Local Binary Patterns (we label this approach as *LBP* in the experiments) from Ojala, Pietikäinen, and Harwood (1996) and the Histogram of Gradients (we label this approach as *HOG* in the experiments) from Dalal and Triggs (2005). The *GLCM* texture description approach build statistics calculated over matrices of neighborhood relations only in given directions and offsets (distances between pixels); *LBP* can be regarded as a histogram of neighborhood relations between a pixel and its eight neighbors and *HOG* is a histogram of gradient orientation on regions of interest of images. The other two descriptors are the Dominant Local Binary Patterns of Bianconi, González, and Fernández (2015b) (labeled as *DLBP* in the experiments) and the Rotation Invariant Co-Occurrences of patterns from González, Fernández, and Bianconi (2014) without feature selection (labeled as C_1^{ri} in the experiments). These two last approaches were validated in the same granite image dataset we are using, which was first presented in the paper of Bianconi et al. (2015a).

Other five descriptors used as baselines in the experiments were originally proposed for specific texture description applications and were never used for texture characterization of granite tiles. The first three approaches are based on the Convolutional Texture Gradient Filter (CTGF), proposed by Ferreira, Navarro, Pinheiro, dos Santos, and Rocha (2015) to identify texture patterns of printed letters to attribute the laser printer source of a document. This descriptor measures the texture of an image as histograms of convolved textures in low gradient areas, using for the convolution matrices of different sizes. We label approaches based on CTGF in the experiments as $CTGF_{3\times3}$, $CTGF_{5\times5}$ and $CTGF_{7\times7}$. The other two approaches came from the same work of Ferreira et al. (2015). These are multidirectional extensions of GLCMs (we call this $GLCM-MD$ in the experiments) and multi-directional and multi-scale extensions of GLCMs (called $GLCM-MD-MS$).

Finally, we also used existing pre-trained CNNs to compare against our networks trained from scratch. These are the original networks used in the CIFAR general-purpose image recognition challenge and MNIST digit recognition challenge and are available to download in the MatConvNet library from VLFEAT (2016). As we do with our approach, we removed the last layer and applied the feature vectors in the nearest neighbor classifier. We call these approaches $CIFAR_{original}$ and $MNIST_{original}$, respectively.

6. Experiments

Now we focus on showing experimental results. For this, we start applying the methodology and calculated the metrics discussed in Section 5.2 using the dataset showed in Section 5.1. All experiments were performed on a cluster node with 11 processors and 131 GB RAM and graphics card NVIDIA GeForce 210. This section is described as follows: firstly, we show how we choose the number of epochs used to train our networks. Then, we start the comparison of our proposed approaches against the state of the art techniques described in Section 5.4 in two scenarios: using big images and small blocks. These two last experiments were chosen to show the effectiveness of the proposed approaches to classify granite rocks images of different resolutions (very high and very low).

6.1. Defining epochs to train CNNs

Before start training our proposed CNNs from scratch using granite images, the number of epochs to train the network, which is the number of one forward and one backward pass of all training examples through the network, must be defined. For this experimental scenario we show the result of classification of using the first combination of train and validation using for that the classifier attached to the end of the network (*i.e.*, a soft-max classifier). In this case, we further subdivide each image on small-sized blocks and use them to train and validate the classifier. One natural solution would be using the whole image containing the granite tile as the input for CNNs, but choosing this solution would require deeper networks, which will require a larger amount of data, more computational time and more memory resources to train the networks. Using smaller areas as input do not require as many layers as using the whole image and also can lead to a faster learning of network parameters and weights.

Using our proposed procedure of classifying small blocks, we then found the number of epochs to be used in each network (MNIST1, MNIST2, MNIST3 and CIFAR) as the ones with less top-1 validation error (*valtop1e*) after 18 epochs and are, respectively for each network, 10, 15 and 15 epochs as Fig. 10 shows. For CIFAR network, we choose the number of epoch with the less top-1 validation error (*valtop1e*) after 150 epochs, which is found in the 142th epoch. As we used so many epochs in our CIFAR seek for best validation epoch, it is not so clear to see its lowest top1-error in the graph, so we decide to exclude it from Fig. 10 for the sake of clarity.

After finding the number of epochs to train our feature extractor, we find the model used to extract feature vectors by first training the networks using the epochs found in this validation experiment. Then, we can use these networks to extract feature vectors of images, by applying training and testing images in the networks and use the output of the last but one layer to feed a multiclass classifier based on the nearest neighbor classification, as described in Section 4.4. The following experiments of this paper consider this scenario and will be discussed as follows.

6.2. Comparison against baselines in high resolution images

Now we focus on the experiments comparing our proposed approaches against the state of the art considering classification of high resolution images of the dataset considered. Table 1 shows the results. For this task, our approaches (highlighted in light gray in this table) perform the classification of test images after doing the majority voting of the 2809 valid 28×28 grayscale blocks for MNIST-based networks and 2116 valid 32×32 RGB blocks in the proposed CIFAR-based network.

As can be seen from Table 1, our deeper network applied on color tiles, called *CIFAR*, is the best network to classify granite rocks, showing perfect detection in all 10 rounds of experiments. This happens because the layers in this network better decompose the color information from these images, highlighting important patterns to be used in the classification.

[1] www.github.com/anselmoferreira/granite-cnn-classification.

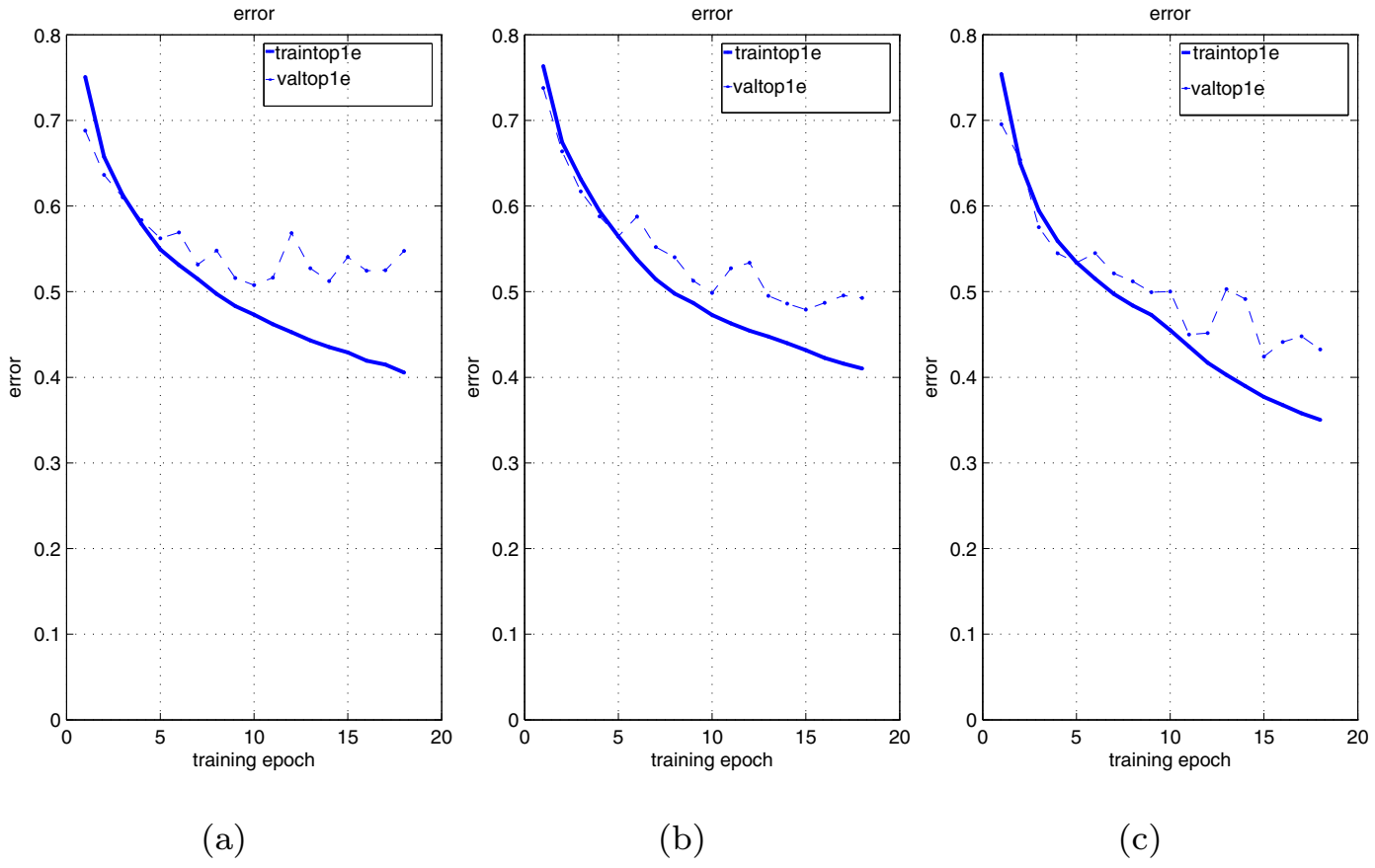

Fig. 10. Validation error results after 18 training epochs of the proposed (a) MNIST1 (b) MNIST2 and (c) MNIST3 CNNs for granite images classification. The errors in these three figures were measured for 28 × 28 granite blocks classification. From this figure the smallest validation error (*valtop1e*) can be found in the 10th, 15th and 15th epoch respectively.

Table 1
Results comparing the best configurations of the proposed method to the existing methods in the literature after 5 × 2 validation. Proposed CNNs methods and CNNs fusion approaches are highlighted in light gray.

Method	Mean ± Std. dev.	Min	Max
CIFAR	100% ± 0.00	100.00%	100.00%
CTGF_3X3 (Ferreira et al., 2015)	100% ± 0.00	100.00%	100.00%
C_1^i (González et al., 2014)	99.96% ± 0.12	99.60%	100.00%
CTGF_5X5 (Ferreira et al., 2015)	99.92% ± 0.10	99.80%	100%
CTGF_7X7 (Ferreira et al., 2015)	99.70% ± 0.14	99.60%	100.00%
DLBP (Bianconi et al., 2015b)	99.88% ± 0.16	99.60%	100.00%
MNIST2	99.32% ± 0.70	97.80%	100.00%
LBP (Ojala et al., 1996)	98.96% ± 0.64	97.80%	99.80%
MNIST3	98.16% ± 0.56	97.00%	99.00%
{*MNIST1*, *MNIST2*, *MNIST3*}*_vote*	96.50% ± 0.99	94.80%	97.60%
$CIFAR_{original}$	94.84% ± 0.52	94.00%	96.00%
GLCM − MD − MS (Ferreira et al., 2015)	92.04% ± 1.09	90.00%	93.80%
GLCM − MD (Ferreira et al., 2015)	90.18% ± 1.43	88.40%	93.20%
MNIST1	86.62% ± 1.61	84.20%	88.80%
HOG (Dalal & Triggs, 2005)	78.04% ± 1.50	76.20%	80.80%
GLCM (Haralick et al., 1973)	72.74% ± 1.97	68.80%	74.60%
$MNIST_{original}$	62.98% ± 2.12	60.80%	66.40%
{*MNIST1*, *MNIST2*, *MNIST3*}*_concat*	26.44% ± 25.89	0.00%	67.00%

One interesting point to be noticed is the good performance of other texture descriptors that were proposed for other applications, such as the ones from Ferreira et al. (2015). The best state of the art approach, *CTGF_3X3*, also showed a perfect detection in all of ten rounds of experiments. This happens because CTGF builds histogram of low pass textures that are in flat areas of the images, considering only the texture of these flat areas instead of using edges information and some abnormal imperfections that can differentiate the same class of granite slabs.

Table 1 also shows that the proposed methods MNIST1 and MNIST3 are not showing good results if compared with most state of the art. This severely impacts the proposed fusions approaches by early fusion ({*MNIST1*, *MNIST2*, *MNIST3*}*_concat*) and late fusion ({*MNIST1*, *MNIST2*, *MNIST3*}*_vote*), because feature vectors and outputs from *MNIST1* and *MNIST3* networks do not de-

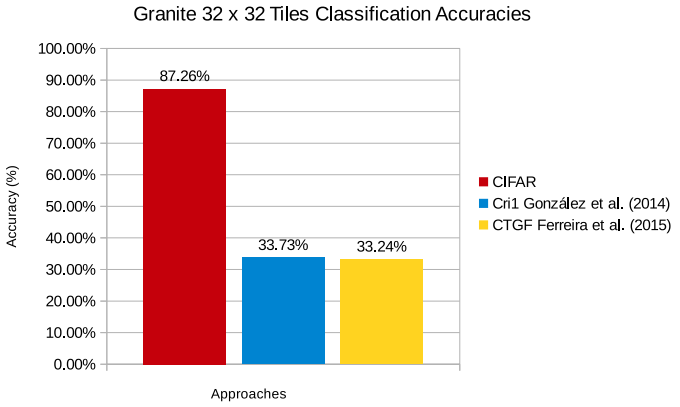

Fig. 11. Results considering the classification of 32 × 32 granite image blocks.

scribe the granite images effectively as *MNIST*2 does. One good step towards the solution could be training new and more deeper networks to be used in the fusion.

Finally, it is worth discussing the results of pre-trained models in classifying granite slabs. As seen in Table 1, the pre-trained models $CIFAR_{original}$ and $MNIST_{original}$ show poor classification results. This happens because these pre-trained models were tough to identify different patterns from the ones present on granite tiles, so their parameters (or filter weights) were not found to discriminate granite slabs patterns. This affects extremely their experiments results.

6.3. Comparison against baselines in low resolution images

For the final round of experiments considering the comparison with the state of the art, we now focus on the experiments regarding small 32 × 32 blocks classification. For this, we consider the classification result without performing the majority voting, using for this the first split of training data and test data. This will result in a total of 1000 (images) × 2116 (blocks)=2,116,000 blocks of size 32 × 32, half of them are used to train and the other half used to test the classifier. We show, in Fig. 11, the accuracy result of our best proposed approach against the best state of the art methods considered in the experiments of Section 6.2.

Results showed in Fig. 11 highlights the difficulty of classifying very tiny images. This can be explained because they contain a very small color information that can be confused with other granite classes, decreasing this way the classification accuracy. Even in this difficult scenario, our proposed CNN *CIFAR*, trained to classify these small blocks with backpropagation of error and more layers, showed a classification accuracy higher than 85% using the same nearest neighbor classifier used in the experiments of Section 6.2, classifying a total of 923,231 blocks out of 1,058,000 blocks. The high classification accuracy of low resolution blocks explains why the majority voting of individual blocks helped our proposed approach to reach the 100% accuracy on the experiments results shown in Table 1.

The poor results of the feature engineering approaches of Ferreira et al. (2015) and González et al. (2014) are due the inner characteristic of these feature engineering approaches to be global descriptors, not being thought to classify low resolution images. For example, the *CTGF*_3*X*3 approach of Ferreira et al. (2015), which classifies correctly only a total of 351,760 blocks, creates histogram of textures on convoluted areas with a 3 × 3 window. As the areas used to describe the granite tiles contain only 32 × 32 = 1024 pixels, the final histogram used as feature vector will contain several bins unused, and the accurate description of this small granite tile will be affected. The low accuracy results of literature solutions in Fig. 11 seriously decrease the possibility of using these approaches to do image classification by majority voting of image patches, as our proposed approach does. Other important aspect of the proposed approach is the time for extracting the feature vectors. While our trained from scratch network took approximately one hour to extract 1,058,000 feature vectors, the approaches from Ferreira et al. (2015) and González et al. (2014) took approximately 6 days each to extract the same number of feature vectors. This highlights the efficiency of the proposed method in this multiple data scenario.

With the results presented in this section, we show that the proposed approach can achieve perfect detection of high resolution images, showing comparable results with the state of the art and, by considering low resolution granite rocks slabs, it outperforms the state of the art comfortably. This indicates, firstly, the potential of our proposed method to be a complementary approach to others in the literature for industrial applications with controlled environment. Moreover, it can be a good starting point for helping classifying rocks on uncontrolled environments, such as using smartphones with different resolutions as acquisition devices for granite recognition, acting as a possible expert to solve misunderstoods between customers and manufacturers when a possible wrong delivery happens.

7. Conclusion

The creation of quality-control procedures for sorting natural stones such as granites is a promising task, as they avoid economical losses due to delivery of wrong granite packages to clients. The automation of such process is either important to reduce errors and eliminate the use of a subjective process involving expensive experts. However, most of the applications proposed in the literature thus far for this task rely on feature engineering approaches that investigate texture and color behaviors which are supposed to not change. Additionally, they do not validate the approaches in a difficult task of classifying natural stones of different image resolutions.

In this paper, we address these issues by proposing a deep learning based approach to granite rocks classification. Our approaches are based on Convolutional Neural Networks of different architectures applied on small image tiles, analyzing textures of each one and using majority voting of them to classify high-resolution images. We also investigate the performance of some of the networks when applied together. Experimental results showed good results of the proposed approaches for classifying high and low resolution images if compared to some other texture descriptors proposed in the literature.

Although there is a long path to go toward the classification of natural stones in real-world situations, in which different resolutions and lightning conditions of granite rocks can happen, we believe that the direction considers deep learning approaches through Convolutional Neural Networks. As our proposed approaches deal with tiny patches, they have potential to be invariant to most of acquisition resolutions, once they are bigger than the input tiny patches that fit our networks. This is an important step to make the granite slabs recognition tasks available to other acquisition devices, such as smartphones. Also, the proposed approaches can also be complimentary to other approaches in industry applications of slabs recognition, as they showed comparable results to the state of the art in classifying high resolution images.

The work started in this paper opens a set of future work to be done, and involves (i) the proposal of new and deeper network architectures for this task; (ii) use of Convolutional Neural Networks applied on other image representations; (iii) testing new ways of fusing different network architectures; (iv) studying the complementarity of these data-driven approaches with featuring engineer-

ing techniques and (v) application of experiments considering the open set scenario of Scheirer, Rocha, Sapkota, and Boult (2013), on which the classifier is designed to also consider unknown samples in the training process.

Acknowledgments

This work was supported partly by NSFC (61332012, U1636202) and the Shenzhen R&D Program (JCYJ20160328144421330). We also thank the support by the Brazilian National Council for Scientific and Technological Development (Grant #312602/2016-2) and professors Nuria Fernández, Bruno Montandon Noronha Barros and Millena Basílio da Silva for the discussions that originated this research.

References

Araújo, M., Martínez, J., Ordóñez, C., & Vilán, J. A. (2010). Identification of granite varieties from colour spectrum data. *Sensors, 10*(9), 8572–8584.

Bianconi, F., Bello, R., Fernández, A., & González, E. (2015a). On comparing colour spaces from a performance perspective: Application to automated classification of polished natural stones. In *New trends in image analysis and processing*. In *Lecture notes in computer science: Vol. 9281* (pp. 71–78). Genoa, Italy: Springer.

Bianconi, F., & Fernández, A. (2006). Granite texture classification with Gabor filters. In *Proceedings of international congress on graphical engineering (INGEGRAF), Sitges, Spain*.

Bianconi, F., González, E., & Fernández, A. (2015b). Dominant local binary patterns for texture classification: Labelled or unlabelled? *Pattern Recognition Letters, 65*, 8–14.

Bianconi, F., González, E., Fernández, A., & Saetta, S. A. (2012). Automatic classification of granite tiles through colour and texture features. *Expert Systems with Applications, 39*(12), 11212–11218.

Dalal, N., & Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proceedings of International conference on computer vision & pattern recognition, California, USA: Vol. 2* (pp. 886–893).

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*, 1895–1923.

Ershad, S. F. (2011). Color texture classification approach based on combination of primitive pattern units and statistical features. *Multimedia and its Applications, 3*(3), 1–13.

European Committee for Standardization (CEN) (2009). EN 12440:2008 natural stone: Denomination criteria. *Report*. European Committee for Standardization (CEN).

Fernández, A., Ghita, O., González, E., Bianconi, F., & Whelan, P. F. (2011). Evaluation of robustness against rotation of LBP, CCR and ILBP features in granite texture classification. *Machine Vision and Applications, 22*(6), 913–926.

Ferreira, A., Navarro, L. C., Pinheiro, G., dos Santos, J. A., & Rocha, A. (2015). Laser printer attribution: Exploring new features and beyond. *Forensic Science International, 247*, 105–125.

González, E., Fernández, A., & Bianconi, F. (2014). General framework for rotation invariant texture classification through co-occurrence of patterns. *Journal of Mathematical Imaging and Vision, 50*(3), 300–313.

Haralick, R. M., Shanmugam, K., & Dinstein, I. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-3*(6), 610–621.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of IEEE international conference on computer vision (ICCV), Santiago, Chile* (pp. 1026–1034).

Hof, R. (2013). Deep learning. With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart. https://www.technologyreview.com/s/513696/deep-learning/.

Johnson, J., & Karpathy, A. (2016). CS231n convolutional neural networks for visual recognition. http://cs231n.github.io/transfer-learning/.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of neural information processing systems (NIPS), Nevada, USA* (pp. 1106–1114).

Kurmyshev, E. V., Sánchez-Yáñez, R. E., & Fernández, A. (2013). Colour texture classification for quality control of polished granite tiles. In *Proceedings of the third IASTED international conference on visualization, imaging and image processing: Vol. 2* (pp. 603–608). ACTA Press.

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

Lepisto, L., Kunttu, I., & Visa, A. (2005). Rock image classification using color features in Gabor space. *Journal of Electronic Imaging, 14*(4), 040503-1–040503-3.

Ojala, T., Pietikäinen, M., & Harwood, D. (1996). A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition, 29*, 51–59.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*, 533–536.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., … Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision, 115*(3), 211–252.

Scheirer, W. J., Rocha, A., Sapkota, A., & Boult, T. E. (2013). Towards open set recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*, 1757–1772.

Stone Industry SA Standards Board (2015). Stone standards. *Report*. Stone Industry SA Standards Board.

University of Tennessee (2008). Granite dimensional stone quarrying and processing. *Report*. University of Tennessee – Center for Clean Products.

VLFEAT (2016). MatConvNet: CNNs for MATLAB. http://www.vlfeat.org/matconvnet/.

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? In *Advances in neural information processing systems* (pp. 3320–3328). Montreal, Canada: Curran Associates, Inc.