

Received October 29, 2019, accepted November 21, 2019, date of publication November 29, 2019, date of current version December 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2956863

# CNN-Based Refactoring of Hand-Designed Filters for Texture Analysis: A Classic Revisited

FRANCESCO BIANCONI<sup>1</sup>, CLAUDIO CUSANO<sup>2</sup>, PAOLO NAPOLETANO<sup>3</sup>,  
AND RAIMONDO SCETTINI<sup>3</sup>

<sup>1</sup>Department of Engineering, University of Perugia, 06125 Perugia, Italy

<sup>2</sup>Department of Electrical, Computer and Biomedical Engineering, University of Pavia, 27100 Pavia, Italy

<sup>3</sup>Department of Computer Science, Systems and Communications, University of Milano–Bicocca, 20126 Milan, Italy

Corresponding author: Paolo Napolitano (paolo.napolitano@unimib.it)

This work was supported in part by the Italian Ministry of University and Research (MIUR) under the Individual Funding Scheme for Fundamental Research “FFABR” 2017 and in part by the Department of Engineering, Università degli Studi di Perugia, Italy, within the Fundamental Research Grants Scheme 2017.

**ABSTRACT** Filtering has been one of the main approaches to texture analysis since early on. Traditionally, the process involved designing the filters essentially by hand based on some prior knowledge (e.g. perceptual models, optimal mathematical properties, etc.) In this work we propose the use of convolutional networks for refactoring traditional, hand-designed filters. Our method consists of initialising the first convolutional layer of the network with some classic banks of filters, training the network on texture images and retrieve the modified filters. Experimenting with five classes of filters and eight datasets of texture images we show that the refactored filters can be conveniently used ‘off-the-shelf’ to achieve better performance than obtained with the original filters, but at the same computational cost.

**INDEX TERMS** Texture analysis, convolutional neural networks, image filters.

## I. INTRODUCTION

Texture description lies at the heart of many computer vision applications, as for instance content-based image retrieval [1], [2], materials categorisation [3], [4], medical image analysis [5], [6], object recognition [7], [8], image reconstruction [9], remote sensing [10], [11] and surface inspection [12], [13].

Up until not long ago texture analysis was essentially ‘hand-crafted’: the descriptors were mostly defined by hand and independently on the data to be analysed. This approach used to be the leading paradigm for many years and has given rise to a huge number of descriptors [14]–[17].

In recent years, however, research in computer vision has been moving towards data-driven models, where the visual features are no longer designed by hand but learned from the data [18]. This approach, usually referred to as Deep Learning [19], has brought dramatic improvements in many areas, such as object, scene and face recognition [20]–[23]. Nonetheless, it is still unclear whether this approach could scale well to textures. Basu *et al.* [24] for instance affirmed

The associate editor coordinating the review of this manuscript and approving it for publication was Huanqiang Zeng.

that texture recognition is an inherently higher-dimensional problem than object recognition, and therefore more difficult to treat with deep neural networks. Besides, both hand-designed descriptors and Deep Learning have strengths and weaknesses that need to be considered carefully – specifically:

- Deep Learning can be very accurate and much more resilient than hand-designed methods to texture non-stationariness and variations in the imaging conditions [17], [25], [26]; however, hand-designed methods seem still preferable when it comes to discriminating very similar textures with low intra-class variability [17], [27];
- Most hand-designed methods have a fairly intuitive and/or interpretable structure; by contrast, Deep Learning is often used as a ‘black box’ and the features generated are rather difficult to interpret;
- Deep Learning requires very large image datasets for training (not readily available for textures) and significant computational resources.

It is therefore crucial to investigate convenient ways to combine and hybridise hand-designed descriptors with Deep Learning in order to make the most of both. In this work

we propose the use of a convolutional network for refactoring traditional, hand-designed linear filters. The method consists of initialising the first convolutional (learnable) layer of the network with some classic banks of filters, training the network on texture images while letting it modify the filters' weights. Once the training is completed we obtain the refactored (optimised) filters as a result. We show that such optimised filters can be conveniently used off-the-shelf to achieve better performance than obtained with the original filters but at the same computational cost.

The remainder of the paper is organised as follows. After reviewing the related literature in Sec. II we recall some basic concepts of filtering for texture analysis (Sec. III) and describe the proposed approach in Sec. IV. Then we detail the experimental set-up in Sec. V and discuss the results in Sec. VI. We conclude the paper with some final considerations (Sec. VII) and directions for future research (Sec. VIII).

## II. RELATED RESEARCH

The use of convolutional networks for texture analysis has been attracting considerable research interest in the last few years. The two main strategies that have so far emerged are: 1) the 'off-the-shelf' use of networks trained for other tasks (e.g. object and face recognition), and 2) the design and training of networks specific for textures.

The first approach consists of plugging a pre-trained net into a standard image processing pipeline. This strategy has been investigated by Cimpoi *et al.* [25], who showed that networks trained for other tasks can be conveniently used as generic feature extractors for texture analysis, either via orderless pooling from convolutional layers or by direct use of the order-sensitive output of the fully connected layers. The viability of this approach has been confirmed in other studies [4], [17], [26], [27], though in [17], [27] the authors suggested that hand-designed features could still be better than pre-trained CNN at discriminating textures with low intra- and inter-class variability.

The second strategy involves designing and training networks specific for textures. In doing this one has to take into account two elements. The first is that texture datasets [28], [29] tend to be much smaller than those available for other recognition tasks (e.g. [30]); the second that the sensitivity to overall shape information and image layout may be inappropriate for textures, where the objective is capturing the distribution of local patterns of low complexity [31]. As a consequence, the current trend is that of adopting simple network configurations which are essentially convolutional, as for instance proposed in [31], [32].

The use of convolutional networks as a tool for learning texture-sensitive filters has received less attention in comparison. Marcos *et al.* [33] is possibly the only relevant reference in this context, though in that work the authors are mainly concerned with learning rotation-invariant filters from scratch, not the optimisation of classic, hand-designed filters.

## III. BACKGROUND: FILTERING FOR TEXTURE ANALYSIS

Texture analysis by filtering usually involves four steps: 1) filter design, 2) convolution (filtering), 3) feature extraction and 4) classification.

The aim of the first step is to define banks of filters that have some affinity for textures. This has generated considerable research interest for at least forty years, and many solutions have been proposed (see [14], [34], [35] for reviews and comparisons). Herein we considered five classes of filters as detailed in Sec III-A.

In the second step the input image  $\mathbf{I}$  is convolved with each of the filters of the bank. For a bank of  $n$  filters  $\mathbf{W} = \{\mathbf{w}_1, \dots, \mathbf{w}_n\}$  we will get as many transformed images  $\mathbf{T}_i$ ,  $i \in \{1, \dots, n\}$ :

$$\mathbf{T}_i = \mathbf{w}_i * \mathbf{I}, \quad (1)$$

where '\*' indicates convolution.

The third step consists of extracting meaningful features from the transformed images. Most often these can be as simple as the mean and standard deviation [36] (also used in this work), but more involved ones have been proposed too [37]. If we extract  $k$  such features for each transformed image we will eventually obtain a feature vector  $\mathbf{f}$  of dimension  $n \times k$ :

$$\mathbf{f} = \{f_{11}, \dots, f_{1k}, f_{21}, \dots, f_{2k}, \dots, f_{n1}, \dots, f_{nk}\}, \quad (2)$$

where  $f_{ij}$  indicates the  $j$ -th feature extracted from the  $i$ -th transformed image.

In the fourth step the feature vector is eventually fed to a standard classifier (e.g. nearest neighbour, Support Vector Machine, random forest, etc.).

### A. HAND-DESIGNED FILTERS

In this section we recall the basics of the linear filters considered in this study, and refer the reader to the given references for further details. All the filters were defined over the spatial domain using a square window of dimension  $s \times s$  (in the experiments we set  $s = 11$ px; more about this choice in Sec. IV-A).

#### 1) DISCRETE COSINE FILTERS

Discrete Cosine Filters (DCF in the remainder) are closely related to the Discrete Cosine Transform (DCT), whereby a sequence of data points is expressed in terms of a sum of cosine functions at different frequencies [38]. For a sequence of  $T$  points the  $i$ -th coefficient of the one-dimensional kernel  $\phi_{i,T}$  can be expressed as follows:

$$\phi_{i,T} = \sum_{t=1}^{T-1} \cos \left[ (2t+1) \frac{\pi i}{2T} \right] : 0 \leq i \leq T-1. \quad (3)$$

The coefficients of the two-dimensional filters are obtained via outer product of any pair of one-dimensional kernels. In the experiments we set  $T = s$  and  $i = \{1, \dots, 5\}$ , this way obtaining the bank of 25 filters shown in Fig. 1a.

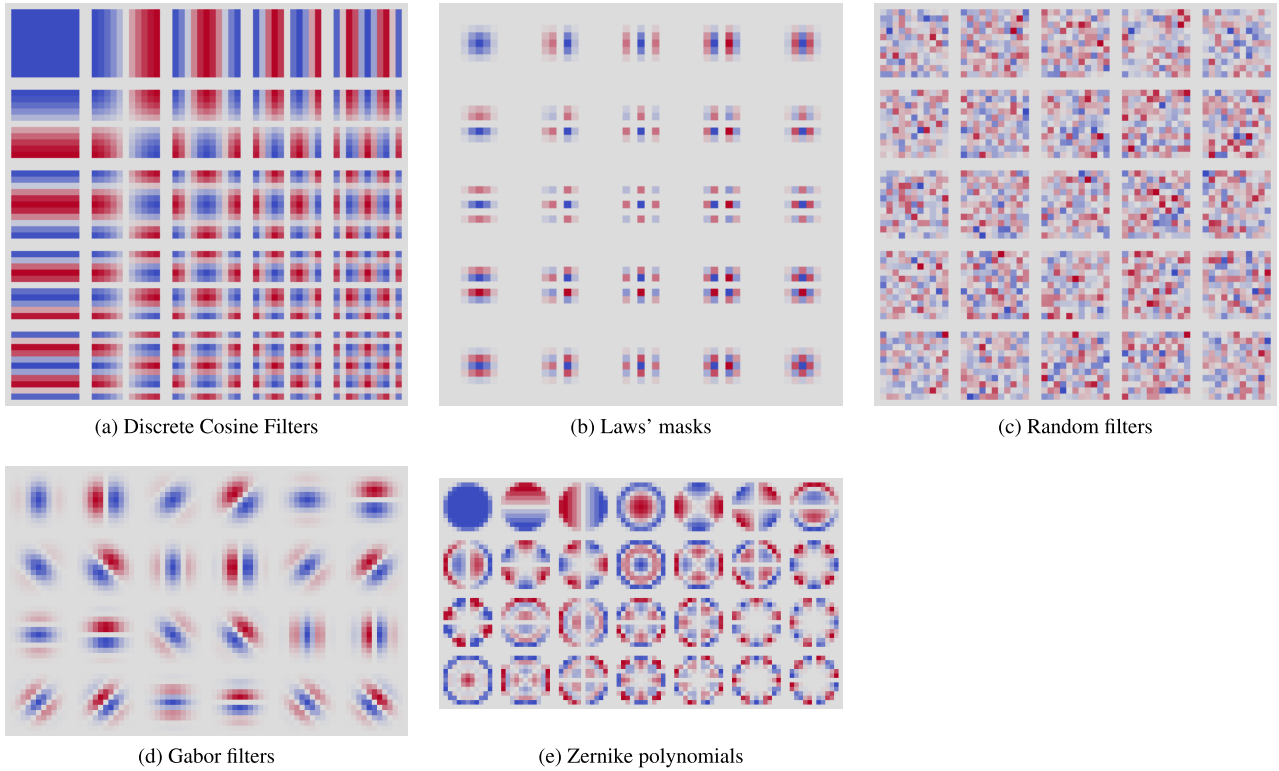


FIGURE 1. The original (non-refactored) hand-designed filters.

2) GABOR FILTERS

Gabor filters have long been used in texture analysis mainly for two reasons: 1) they are a good model for the behaviour of simple cells in the primary visual cortex [39] and 2) they achieve optimal joint resolution in the space and frequency domain [40]. Gabor filters are two-dimensional sinusoidal functions modulated by a Gaussian envelope. In the experiments we used a bank with three frequencies and four orientations; the maximum wavelength (minimum frequency) was set to  $s$ , the frequency ratio to half octave, the spread of the Gaussian window to  $(s - 1)/5$  and the shape ratio to one (i.e.: circular filters). Real and imaginary part of each filter were considered separately, giving a total of  $3 \times 4 \times 2 = 24$  filters (Fig. 1d).

3) LAWS' MASKS

Laws' masks are separable filters obtained by combining one-dimensional kernels, of which the most common (also used here) are the following:

$$\begin{aligned}
 L5 &= [1 \ 4 \ 6 \ 4 \ 1], \\
 E5 &= [-1 \ -2 \ 0 \ 2 \ 1], \\
 S5 &= [-1 \ 0 \ 2 \ 0 \ -1], \\
 W5 &= [-1 \ 2 \ 0 \ -2 \ 1], \\
 R5 &= [1 \ -4 \ 6 \ -4 \ 1],
 \end{aligned} \tag{4}$$

where the initial letters respectively indicate 'Level', 'Edge', 'Spot', 'Wave' and 'Ripple' [14], [41]. From the above we obtained 25 two-dimensional filters via outer product of all

the possible pairwise combinations of the one-dimensional kernels – e.g.:  $L5 \otimes L5$ ,  $L5 \otimes E5$ ,  $L5 \otimes S5$ , etc., where ' $\otimes$ ' indicates outer product (Fig. 1b).

4) ZERNIKE POLYNOMIALS

Zernike polynomials are sets of orthogonal polynomials that allow the expansion of an arbitrary wavefront in polar coordinates  $(\rho, \theta)$ . The Zernike polynomial  $Z_r^m$  of order  $r$  and azimuthal frequency  $m$  over the unit circle ( $0 \leq \rho \leq 1$ ) is defined as follows [42]:

$$\begin{aligned}
 Z_r^m &= R_r^m(\rho) \cos(m\theta) : m \geq 0, \\
 Z_r^{-m} &= R_r^m(\rho) \sin(m\theta) : m < 0,
 \end{aligned} \tag{5}$$

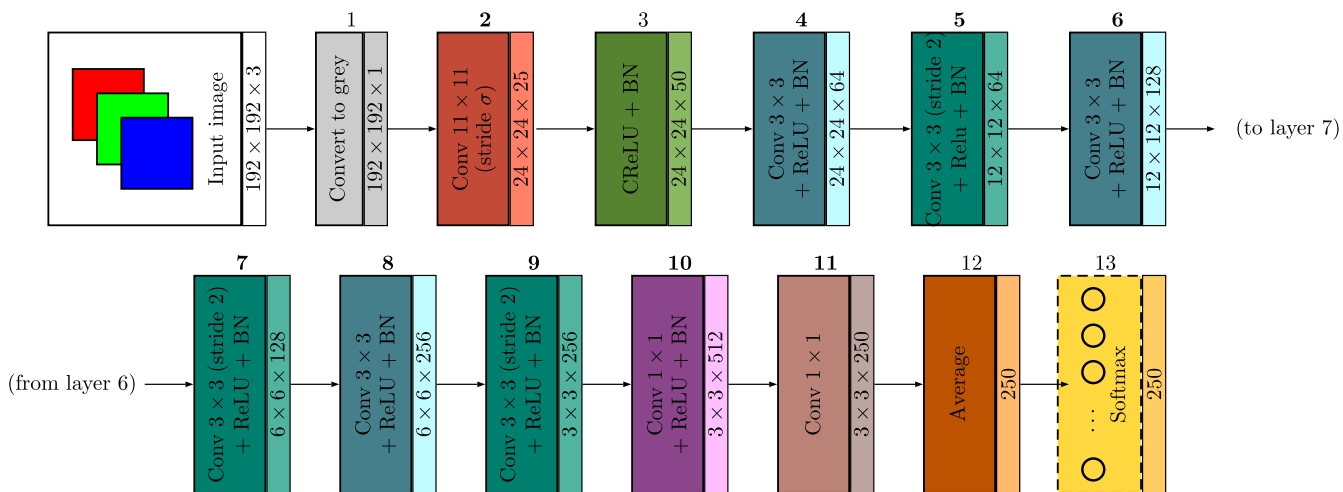
where:

$$R_r^m(\rho) = \sum_{l=0}^{\frac{n-m}{2}} \frac{(-1)^l (r-l)! \rho^{(r-2l)}}{l! \left[\frac{1}{2}(n+m-l)\right]! \left[\frac{1}{2}(r-m-l)\right]!}. \tag{6}$$

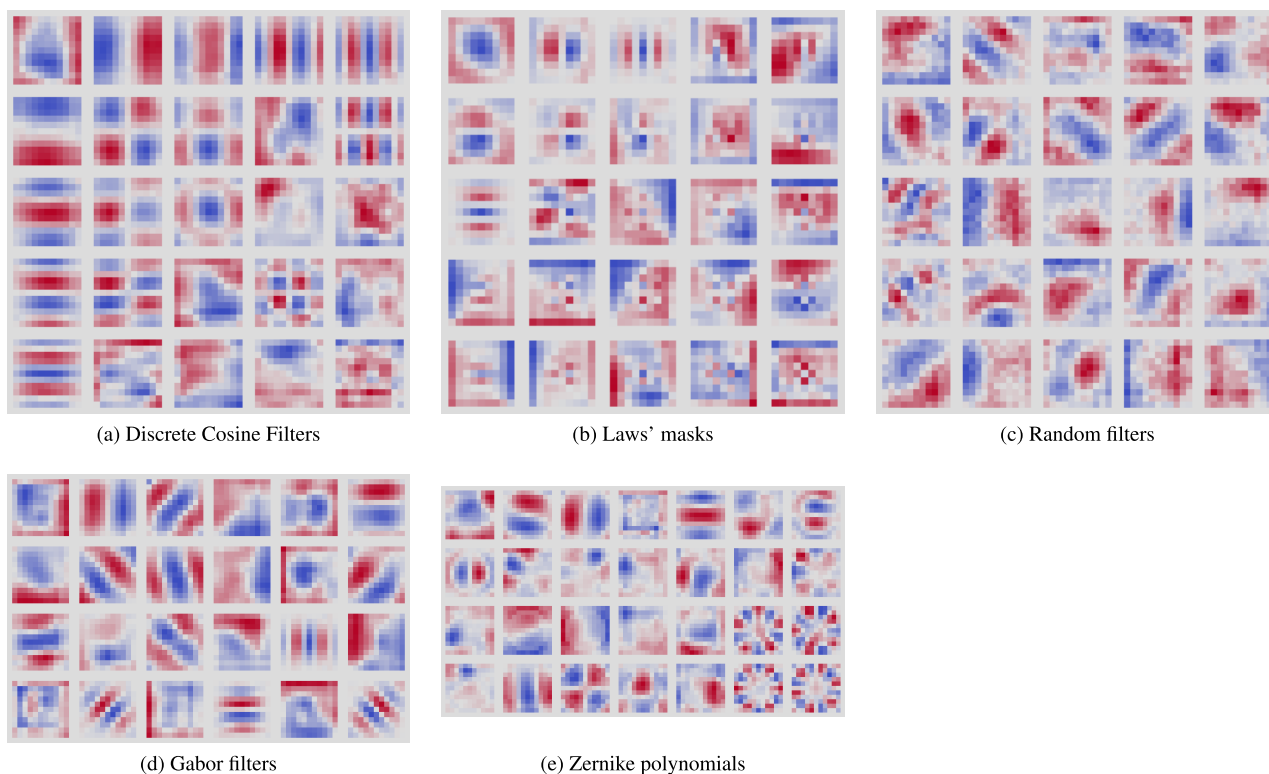
In the experiments we considered the 28 Zernike polynomials that arise from  $m = \{-6, 5, \dots, 5, 6\}$  and  $r = \{0, \dots, 6\}$ . The digital versions of the filters were obtained by resampling the polynomials over a digital circle of diameter 11px (Fig. 1e).

5) RANDOM FILTERS

Finally, we also considered a bank of 25 filters with weights sampled randomly from a univariate normal distribution of mean 0 and variance 1 (Fig. 1c).



**FIGURE 2.** Network’s architecture. Key to acronyms: ‘ReLU’ = Rectifying Linear Unit, ‘CReLU’ = Concatenated Rectifying Linear Unit, ‘Conv’ = Convolution, ‘BN’ = Batch Normalization. Layers with equal structure are the same colour. Boldface numbers indicate learnable layers.



**FIGURE 3.** The refactored filters.

**IV. PROPOSED APPROACH: REFACTORIZING FILTERS VIA CONVOLUTIONAL NETWORKS**

Our approach consists of using one layer of a convolutional network for the filter refactoring. This is a four-step procedure in which we start by designing the network first, then initialise one layer of the network with the hand-designed filters, train the network and eventually retrieve the optimise filters. We implemented the procedure using a convolutional network specifically designed for textures as described below.

**A. NETWORK’S ARCHITECTURE AND TRAINING**

We used a CNN composed of 13 layers of which nine learnable (Fig. 2). The network was designed to capture local, orderless visual features rather than global, order-sensitive ones (see also Sec. II on this point). Therefore, the learnable layers are all convolutional and there are no fully-connected layers. The architecture is also inspired on the ‘striving for simplicity – all-convolutional paradigm’ [43].

The first layers of the network were designed with the objective of simplifying the refactoring. Since we were mainly interested in evaluating the texture discrimination capability of the CNN and the refactored filters, we made the choice to operate on intensity images and discarded colour. The input images were therefore converted to grey-scale before processing. An  $11\text{px} \times 11\text{px}$  convolution is then performed on the image with the convolution kernel being initialized with one of the filter banks described in Sec. III-A. The size of the convolutional mask was the result of a trade-off between the complexity of the original, non-refactored filters (which basically depends on the number of frequencies orientations used) and the computational cost. The smaller filters (e.g. Laws') were embedded into the mask by padding with zeros the resulting gap.

The stride  $\sigma$  of this convolutional layer is of particular importance since it determines how much information the bank of filters passes on to the rest of the network. For this parameter we tested an array of four values:  $\sigma \in \{2, 5, 8, 11\}$  (see also Sec. VI on this point). Instead of the usual ReLU activation function we opted for a Concatenated Rectifying Linear Unit (CReLU) as proposed by Shang et al. [44]. Differently from the more common ReLU, this activation function preserves both positive and negative phase information, which is coherent with the filter banks used here.

The network continues with six blocks composed of a  $3\text{px} \times 3\text{px}$  convolution (with stride  $1\text{px}$  or  $2\text{px}$ ), ReLU activation and a batch normalization layer, followed by two  $1\text{px} \times 1\text{px}$  convolutional blocks yielding a set of local score vectors (one score per class). The average score vector is eventually processed in the softmax layer to produce the vector of posterior probability estimates. The total number of learnable parameters is 1435364 (half of them in layer 9).

The network was trained with the Adam optimization method on the ALOT dataset (250 texture classes, 100 image samples per class – see Sec. V-C for details). The original data were augmented via random crop, rotation (by multiples of 90 degrees) and/or flip. This way we obtained 75.000 patches, of which 73.000 were used for training and the remaining ones for validation. The other training parameters were: number of iterations = 450.000, batch size = 10, learning rate =  $10^{-4}$ , number of decay steps = 100.000, decay rate = 0.9 and  $L_2$  regularisation =  $10^{-4}$ .

### B. REFACTORED FILTERS

Once the network's training is completed we can retrieve the refactored filters and use the 'off-the-shelf' for texture classification. Note that the refactoring process modifies the original filters substantially, as shown in Fig. 3. As expected, random filters became more complex, and some of them seem to have evolved in a way that allows for sensitivity to common patterns such as edges. In this case the resulting filters do not look dissimilar from those learned by other CNNs (see for instance [45]). In the other cases we observe that some filters

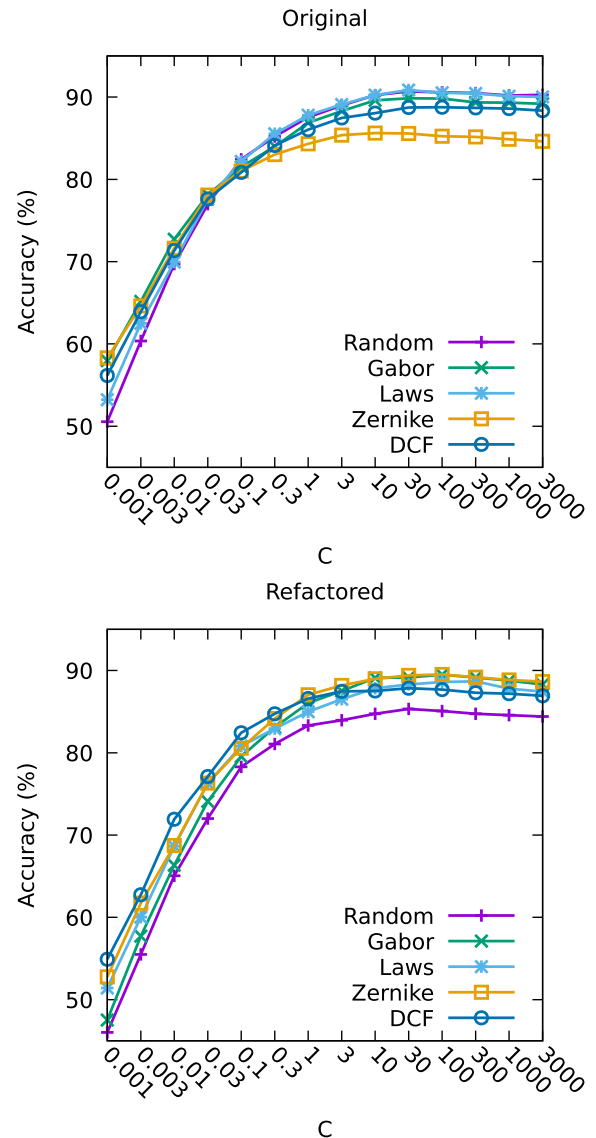



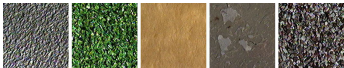
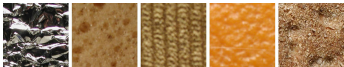

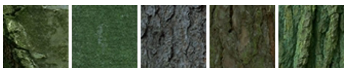


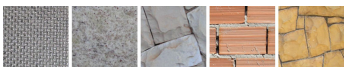
FIGURE 4. Estimated classification accuracy of the original and refactored filters as a function of C. Values obtained on KTH-TIPS2b (for the other datasets the behavior was very similar).

modified slightly, whereas others underwent more radical changes.

One common trend was for some filter to overflow their original support and spread over the whole  $11 \times 11$  domain. This was particularly evident for Laws' masks, which were originally defined over a  $5 \times 5$  mask and zero-padded to fill up the  $11 \times 11$  grid. In this case the refactoring process seems to have focused on the borders of the  $11 \times 11$  domain. The same happened with Zernike moments, which were originally defined on a disc and evolved from circular to square.

For DCF, Zernike and Gabor it appears that the simplest filters (e.g. edge detectors) underwent a tuning process, while some of the more complex ones changed more substantially – though their original shape is not lost altogether. In all the cases the low-pass filters changed to detect center-surround differences.

**TABLE 1. Texture datasets: round-up table.**

Acronym	Type of textures	No. of classes	No. of samples per class	Image size	Sample images	Refs.
ALOT	Generic materials	250	100	$1536 \times 1024$		[46], [47]
CUReT	Generic materials	61	93	$200 \times 200$		[48]
KTH-TIPS	Generic materials	10	81	$200 \times 200$		[49], [50]
KTH-TIPS2b	Generic materials	11	432	$200 \times 200$		[3], [50]
NewBarkTex	Bark	6	273	$64 \times 64$		[51], [52]
PlantLeaves	Plant leaves	20	60	$128 \times 128$		[53]
STex	Generic materials and scenes	476	16	$128 \times 128$		[54]
USPTex	Generic materials and scenes	191	12	$128 \times 128$		[55], [56]

**TABLE 2. Average accuracy difference by type of filter: refactored vs. original filters. Values are in percentage points.**

Type of filter	Average difference refactored vs. original
Random	8.6
Laws'	7.5
Gabor	6.0
DCF	5.7
Zernike	4.9

## V. EXPERIMENTS

We comparatively evaluated the accuracy of the original filters (Fig. 1) with that of the refactored ones (Fig. 3) on a set of supervised image classification experiments. The experimental settings, accuracy estimation procedure and datasets are detailed in the following subsections.

### A. FEATURE EXTRACTION AND CLASSIFICATION

Image features were, for all the filters considered, the mean and standard deviation of the magnitude of the transformed images. Classification was based on a linear Support Vector Classifier (SVC). For each classification task the optimal value for SVC penalty parameter  $C$  was determined via five-fold cross-validation on the training data of the first split of each dataset. The grid-search was carried over  $C \in \{10^{-3}, 10^{-2}, \dots, 10^3, 3 \times 10^{-3}, 3 \times 10^{-2}, \dots, 3 \times 10^3\}$ . As a result, a different optimal value of  $C$  was used for each combination image descriptor/dataset. The accuracy vs.  $C$  curve showed that performance was very stable around the optimal value of  $C$ , as can be seen in Fig. 4.

**TABLE 3. Average accuracy difference by dataset: refactored vs. original filters. Values are in percentage points.**

Dataset	Average difference refactored vs. original
STex	16.8
PlantLeaves	6.9
CUReT	6.6
USPTex	6.1
KTH-TIPS2b	3.3
NewBarkTex	3.2
KTH-TIPS	2.9

### B. ACCURACY ESTIMATION

Accuracy estimation was based on split-sample validation with stratified sampling. Half of the samples of each class (train set) were used for training the classifier, and the remaining half (test set) to estimate the accuracy. This was the fraction of samples of the test set classified correctly. For a stable estimation the results were averaged over 100 subdivisions into train and test set.

### C. DATASETS

We based the study on the following eight datasets of texture images: 1) Amsterdam Library of Textures (ALOT), 2) Columbia-Utrecht Reflectance and Texture Database (CUReT, version maintained by the Visual Geometry Group at the University of Oxford, UK), 3) KTH-TIPS, 4) KTH-TIPS2b, 5) New BarkTex, 6) Plant leaves, 7) Salzburg Texture Image Database (STex) and 8) USPTex. The main features of each dataset – e.g. type of textures,

**TABLE 4. Classification accuracy: original (hand-designed) vs. refactored filters. Boldface figures indicate the best accuracy for each combination filter/dataset; framed figures the overall best accuracy by dataset. Symbols ‘□’ and ‘♣’ respectively indicate the hand-designed and refactored version of each filter. All the refactored filters were obtained using  $\sigma = 8$ .**

Dataset	Random		Gabor		Laws’		Zernike		DCF	
	□	♣	□	♣	□	♣	□	♣	□	♣
CUReT	86.22	<b>92.25</b>	89.23	<b>92.51</b>	87.80	<b>91.61</b>	92.89	<b>93.57</b>	87.05	<b>93.26</b>
KTH-TIPS	85.72	<b>91.31</b>	90.35	<b>91.84</b>	89.62	<b>92.38</b>	93.48	<b>93.84</b>	92.00	<b>93.29</b>
KTH-TIPS2b	85.00	<b>89.12</b>	88.16	<b>89.51</b>	88.76	<b>89.80</b>	90.26	<b>91.13</b>	86.53	<b>89.71</b>
NewBarkTex	83.13	<b>87.85</b>	87.44	<b>88.67</b>	83.71	<b>87.83</b>	86.91	<b>89.35</b>	87.68	<b>89.46</b>
PlantLeaves	70.99	<b>82.30</b>	73.23	<b>81.21</b>	76.21	<b>82.24</b>	79.18	<b>83.94</b>	82.26	<b>84.45</b>
STex	70.46	<b>78.41</b>	72.40	<b>78.00</b>	68.54	<b>78.63</b>	75.02	<b>80.81</b>	74.74	<b>79.01</b>
USPTex	79.64	<b>84.63</b>	82.96	<b>84.86</b>	78.74	<b>86.82</b>	82.66	<b>87.39</b>	82.17	<b>86.61</b>

number of classes, number of samples for each class and sample images are summarised in Tab. 1; further details are available in the given references. In the experiment we used the largest dataset (ALOT) for training the network and refactoring the filters, the remaining ones for testing the accuracy. All the images were converted to grey-scale before use.

## VI. RESULTS AND DISCUSSION

Table 4 compares the accuracy obtained by the original hand-designed filters with that achieved by their refactored versions. As can be seen, the refactored filters clearly outperformed the original ones, obtaining better accuracy in all the cases. The average increase by type of filter (Tab. 2) ranged between 4.9 percentage points (for Zernike moments) and 8.6 percentage points (for random filters); the average increase by dataset (Tab. 3) between 2.9 percentage points (for KTH-TIPS) and 16.8 percentage points (for STex).

Random filters were the ones that most benefited from refactoring, as one would reasonably expect, but the increase was also noticeable with the other types – specifically Gabor filters and Laws’ masks. The gain was particularly consistent with STex and USPTex. Compared with the others, these two datasets feature a higher number of classes and less stationary texture images.

Notably, we found that the accuracy of the non-random hand-designed filters (either these be refactored or not) was generally superior to that of the random ones. This is important, since it shows that knowledge of filter design from the ‘hand-crafted era’ can be conveniently carried over into the Deep Learning domain to improve accuracy.

As for the type of filter, Zernike’s polynomials provided the best overall accuracy in five datasets out of seven, followed by DCF (two datasets).

For calibration purposes Table 5 also reports the accuracy obtained when the whole network was used as a feature extractor. The features were, in this case, the output of layer no. 10 averaged over the spatial dimensions and normalized to have zero mean and unit variance. The performance of the network was clearly superior to that of both the

**TABLE 5. Best classification accuracy by dataset achieved by the whole CNN. The results were obtained with  $\sigma = 2$ . See Tab. 4 for key to symbols.**

Dataset	Accuracy	Type of filter (layer no. 3)
CUReT	96.24	Laws’□
KTH-TIPS	97.34	Random♣
KTH-TIPS2b	96.44	Random♣
NewBarkTex	87.38	DCF□
PlantLeaves	87.17	DCF□
STex	92.37	Gabor’♣
USPTex	96.53	Laws’□

hand-designed and refactored filters, which is again in line with what one could reasonably expect.

### A. COMPARISON WITH OTHER METHODS

Table 6 compares the best accuracy achieved through the refactored filters with that obtained via other texture descriptors. As can be seen, the relative performance of the methods depends a great deal on the dataset and the method considered. The accuracy figures reported in [26], for instance, show that our best refactored filters (Zernike) performed better than pre-trained CNN models (including AlexNet, ResNet and VGG) on KTH-TIPS; however, the results available in [17] indicate a reversed trend for the NewBarkTex, Plant leaves and CUReT datasets.

A comparison with other recent hand-designed descriptors reveals that our best accuracy was slightly below than was achieved by Improved Local Quinary Patterns (ILQP [57]) on the KTH-TIPS dataset, but higher than was obtained via Hybrid Color Local Binary Patterns (HCLBP [58]) on KTH-TIPS2b. The data available in [59] also indicate that our best refactored filters (DCF in this case) did better than two high-performance Local Binary Patterns (LBP) variants (respectively Opponent-colour Local Binary Patterns, OCLBP, and their improved version, IOCLBP) on NewBarkTex, but worse on CUReT. Finally, the figures of merit reported in [60] indicate that our best refactored filters (Zernike) did better than LBP and

**TABLE 6. Comparison with other methods. Boldface figures indicate the best accuracy by row.**

Dataset	This work		Literature		
	Best accuracy	Method*	Accuracy	Method	Ref.
CUReT	93.6	Zernike	<b>94.2</b>	ResNet 101	[17, Tab. 15]
CUReT	<b>93.6</b>	Zernike	88.7	IOCLBP	[17, Tab. 15]
KTH-TIPS	93.8	Zernike	<b>94.0</b>	ILQP	[57, Tab. 4]
KTH-TIPS2b	<b>91.1</b>	Zernike	83.6	HCLBP	[58, Tab. 7]
KTH-TIPS2b	<b>91.1</b>	Zernike	78.7	ResNet 101	[26, Tab. 2]
NewBarkTex	89.5	DCF	<b>97.9</b>	ResNet 50	[17, Tab. 8]
NewBarkTex	89.5	DCF	80.2	IOCLBP	[59, Tab. 6]
PlantLeaves	84.5	DCF	<b>86.6</b>	ResNet 50	[17, Tab. 7]
PlantLeaves	<b>84.5</b>	DCF	77.7	IOCLBP	[59, Tab. 6]
STex	<b>80.8</b>	Zernike	74.2	LBP	[60, Tab. 1]
STex	<b>80.8</b>	Zernike	80.4	ILBP	[60, Tab. 1]
STex	80.8	Zernike	<b>91.5</b>	IOCLBP	[60, Tab. 1]
STex	<b>87.4</b>	Zernike	82.1	LBP	[60, Tab. 1]
STex	<b>87.4</b>	Zernike	87.0	ILBP	[60, Tab. 1]
STex	87.4	Zernike	<b>92.7</b>	IOCLBP	[60, Tab. 1]

\*Acronyms refer to the refactored version of the filters

ILBP — but worse than IOCLBP — both on the STex and USPTex datasets.

This benchmark shows that the accuracy obtained with the refactored filters was, in most cases, well within the range of values achieved by other state-of-the-art image descriptors — particularly the hand-designed ones. This result is particularly encouraging if we consider that, differently from most other descriptors, our proposed approach does not incorporate any information about color.

## VII. CONCLUSION

For many years texture analysis by filtering involved designing the filters essentially by hand. The hand-designed paradigm, as it is called, is now being challenged by Deep Learning, whereby image features are no longer computed via hand-crafted functions, but learned from the data.

In this paper we have introduced a novel technique which combines previous knowledge on filtering from the ‘hand-designed era’ with the appealing potentialities of Deep Learning. Our method consists of using a convolutional network for refactoring traditional, hand-designed filters. Starting from four classes of filters that have a long history in texture analysis, we showed how these could be conveniently refactored and afterwards used in a off-the-shelf manner at the same computational cost but with significantly improved accuracy. The gain in the discrimination accuracy measured on eight datasets of texture images ranged between 2.9 and 16.8 percentage points. The use of the refactored filters is particularly appealing in those situations where the use of the whole CNN would be inadequate, as the analysis of non-square images and/or segmentation tasks.

## VIII. LIMITATIONS AND FUTURE WORK

The main objective of this work was to investigate the use of a convolutional network for refactoring traditional,

hand-designed linear filters. Our findings are promising and could be further extended to other classes of filters and/or network architectures.

There are of course a number of limitations to this study and questions that should be addressed in future work. Among them, the effect of the filters’ size on the performance of the original and the refactored filters was not investigated in this paper. Likewise, the mechanisms underlying the changes introduced by the network into the filters’ structure are still to be understood, and the potential links to perceptual models of the human vision system to be determined. Finally, the present work is limited to grey-scale textures: extensions to the colour domain should be explored in future studies.

## REFERENCES

- [1] S. Saha, A. Das, and B. Chanda, “CBIR using perception based texture and colour measures,” in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, vol. 2, Cambridge, U.K., Aug. 2004, pp. 985–988.
- [2] Y. D. Chun, N. C. Kim, and I. H. Jang, “Content-based image retrieval using multiresolution color and texture features,” *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1073–1084, Oct. 2008.
- [3] B. Caputo, E. Hayman, and P. Mallikarjuna, “Class-specific material categorisation,” in *Proc. 10th IEEE Int. Conf. Comput. Vis. (ICCV)*, Beijing, China, vol. 2, Oct. 2005, pp. 1597–1604.
- [4] C. Cusano, P. Napoletano, and R. Schettini, “Evaluating color texture descriptors under large variations of controlled lighting conditions,” *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 33, no. 1, pp. 17–30, 2016.
- [5] G. Castellano, L. Bonilha, L. M. Li, and F. Cendes, “Texture analysis of medical images,” *Clin. Radiol.*, vol. 59, no. 12, pp. 1061–1069, Dec. 2004.
- [6] L. Nanni, A. Lumini, and S. Brahmam, “Local binary patterns variants as texture descriptors for medical image analysis,” *Artif. Intell. Med.*, vol. 49, no. 2, pp. 117–125, Jun. 2010.
- [7] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “TexonBoost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context,” *Int. J. Comput. Vis.*, vol. 81, no. 1, pp. 2–23, Jan. 2009.
- [8] A. Satpathy, X. Jiang, and H.-L. Eng, “LBP-based edge-texture features for object recognition,” *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1953–1964, May 2014.
- [9] H. He, G. Li, Z. Ye, A. Mao, C. Xian, and Y. Nie, “Data-driven 3D human head reconstruction,” *Comput. Graph.*, vol. 80, pp. 85–96, May 2019.
- [10] R. Trias-Sanz, G. Stamon, and J. Louchet, “Using colour, texture, and hierarchical segmentation for high-resolution remote sensing,” *ISPRS J. Photogram. Remote Sens.*, vol. 63, no. 2, pp. 156–168, Mar. 2008.



- [11] M. Aguilar, A. Fernández, F. Aguilar, F. Bianconi, and A. G. Lorca, "Classification of urban areas from geospatial imagery through texture features based on histograms of equivalent patterns," *Eur. J. Remote Sens.*, vol. 49, no. 1, pp. 93–120, Mar. 2016.
- [12] X. Xie, "A review of recent advances in surface defect detection using texture analysis techniques," *Electron. Lett. Comput. Vis. Image Anal.*, vol. 7, no. 3, pp. 1–22, 2008.
- [13] C. Koch, K. Georgieva, V. Kasireddy, B. Akinci, and P. Fieguth, "A review on computer vision based defect detection and condition assessment of concrete and asphalt civil infrastructure," *Adv. Eng. Inform.*, vol. 29, no. 2, pp. 196–210, 2015.
- [14] X. Xie and M. Mirmehdi, "A galaxy of texture features," in *Handbook of Texture Analysis*, M. Mirmehdi, X. Xie, and J. Suri, Eds. London, U.K.: Imperial College Press, 2008, pp. 375–406.
- [15] S. Fekri-Ershad, "Color texture classification approach based on combination of primitive pattern units and statistical features," *Int. J. Multimedia Appl.*, vol. 3, no. 3, pp. 1–13, Aug. 2011.
- [16] L. Liu, P. Fieguth, Y. Guo, X. Wang, and M. Pietikäinen, "Local binary features for texture classification: Taxonomy and experimental study," *Pattern Recognit.*, vol. 62, pp. 135–160, Feb. 2017.
- [17] R. Bello-Cerezo, F. Bianconi, F. Di Maria, P. Napoletano, and F. Smeraldi, "Comparative evaluation of hand-crafted image descriptors vs. off-the-shelf CNN-based features for colour texture classification under ideal and realistic conditions," *Appl. Sci.*, vol. 9, p. 738, Feb. 2019.
- [18] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, and M. Pietikäinen, "From BoW to CNN: Two decades of texture representation for texture classification," *Int. J. Comput. Vis.*, vol. 127, pp. 74–109, Jan. 2019.
- [19] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
- [20] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. 26th Annu. Conf. Neural Inf. Process. Syst.*, Lake Tahoe, NV, USA, Dec. 2012, pp. 84–90.
- [21] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, vol. 1, Jan. 2014, pp. 487–495.
- [22] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Columbus, OH, USA, pp. 512–519, Jun. 2014.
- [23] H. Sang and Z. Zhou, "Automatic detection of human faces in color images via convolutional neural networks," *ICIC Exp. Lett. B, Appl.*, vol. 7, no. 4, pp. 781–787, 2016.
- [24] S. Basu, S. Mukhopadhyay, M. Karki, R. DiBiano, S. Ganguly, R. Nemani, and S. Gayaka, "Deep neural networks for texture classification—A theoretical analysis," *Neural Netw.*, vol. 97, pp. 173–182, Jan. 2018.
- [25] M. Cimpoi, S. Maji, I. Kokkinos, and A. Vedaldi, "Deep filter banks for texture recognition, description, and segmentation," *Int. J. Comput. Vis.*, vol. 118, no. 1, pp. 65–94, 2016.
- [26] P. Napoletano, "Hand-crafted vs learned descriptors for color texture classification," in *Proc. 6th Comput. Color Imaging Workshop (CCIW)* in Lecture Notes in Computer Science, vol. 10213, S. Bianco, R. Schettini, S. Tominaga, and A. Tremeau, Eds. Milan, Italy: Springer, 2017, pp. 259–271.
- [27] R. Bello-Cerezo, F. Bianconi, S. Cascianelli, M. Fravolini, F. Di Maria, and F. Smeraldi, "Hand-designed local image descriptors vs. off-the-shelf CNN-based features for texture classification: An experimental comparison," in *Proc. 10th KES Int. Conf. Intell. Interact. Multimedia Syst. Services (IIMSS)*, vol. 76, Cham, Switzerland: Springer, Jun. 2018, pp. 1–10.
- [28] S. Hossain and S. Serikawa, "Texture databases—A comprehensive survey," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2007–2022, 2013.
- [29] F. Bianconi and A. Fernández, "An appendix to 'texture databases—A comprehensive survey,'" *Pattern Recognit. Lett.*, vol. 45, pp. 33–38, Aug. 2014.
- [30] *ImageNet*. Accessed: Feb. 23, 2018. [Online]. Available: <http://www.image-net.org>
- [31] V. Andrearczyk and P. F. Whelan, "Using filter banks in convolutional neural networks for texture classification," *Pattern Recognit. Lett.*, vol. 84, pp. 63–69, Dec. 2016.
- [32] T.-Y. Lin and S. Maji, "Visualizing and understanding deep texture representations," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 2791–2799.
- [33] D. Marcos, M. Volpi, and D. Tuia, "Learning rotation invariant convolutional filters for texture classification," in *Proc. Int. Conf. Pattern Recognit.*, Cancún, Mexico, Apr. 2017, pp. 2012–2017.
- [34] T. Randen and J. H. Husøy, "Filtering for texture classification: A comparative study," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 4, pp. 291–310, Apr. 1999.
- [35] K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1615–1630, Oct. 2005.
- [36] B. S. Manjunath and W. Y. Ma, "Texture features for browsing and retrieval of image data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 8, pp. 837–841, Aug. 1996.
- [37] X. Liu and D. Wang, "Texture classification using spectral histograms," *IEEE Trans. Image Process.*, vol. 12, no. 6, pp. 661–670, Jun. 2003.
- [38] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Trans. Comput.*, vol. C-23, no. 1, pp. 90–93, Jan. 1974.
- [39] M. Turner, "Texture discrimination by Gabor functions," *Biological*, vol. 55, nos. 2–3, pp. 71–82, 1986.
- [40] J. G. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A, Opt. Image Sci.*, vol. 2, no. 7, pp. 1160–1169, 1985.
- [41] K. Laws, "Rapid texture identification," in *Image Processing for Missile Guidance* (Proceedings of SPIE), vol. 0238, T. Wiener, Ed. Bellingham, WA, USA: SPIE, 1980.
- [42] V. Lakshminarayanan and A. Fleck, "Zernike polynomials: A guide," *J. Mod. Opt.*, vol. 58, pp. 545–561, Apr. 2011.
- [43] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. Int. Conf. Learn. Represent. (Workshop Track)*, San Diego, CA, USA, May 2015.
- [44] W. Shang, K. Sohn, D. Almeida, and H. Lee, "Understanding and improving convolutional neural networks via concatenated rectified linear units," in *Proc. 33rd Int. Conf. Mach. Learn. (ICML)*, vol. 5, New York, NY, USA, Jun. 2016, pp. 3276–3284.
- [45] A. Mahendran and A. Vedaldi, "Visualizing deep convolutional neural networks using natural pre-images," *Int. J. Comput. Vis.*, vol. 120, pp. 233–255, Dec. 2016.
- [46] *Amsterdam Library of Textures (ALOT)*. Accessed: Jun. 13, 2018. [Online]. Available: [http://aloi.science.uva.nl/public\\_alot/](http://aloi.science.uva.nl/public_alot/)
- [47] G. Burghouts and J.-M. Geusebroek, "Material-specific adaptation of color invariant features," *Pattern Recognit. Lett.*, vol. 3, pp. 306–313, Feb. 2009.
- [48] Visual Geometry Group. *CURvT: Columbia-Utrecht Reflectance and Texture Database*. Accessed: Jan. 26, 2017. [Online]. Available: <http://www.robots.ox.ac.uk/~vgg/research/texclass/setup.html>
- [49] E. Hayman, B. Caputo, M. Fritz, and J. Eklundh, "On the significance of real-world conditions for material classification," in *Proc. 8th Eur. Conf. Comput. Vis. (ECCV)*, in Lecture Notes in Computer Science, vol. 3024, Prague, Czech Republic, May 2004, pp. 253–266.
- [50] *The KTH-TIPS and KTH-TIPS2 Image Databases*. Accessed: Jan. 11, 2017. [Online]. Available: <http://www.nada.kth.se/cvap/databases/kth-tips/download.html>
- [51] A. Porebski, N. Vandenbroucke, L. Macaire, and D. Hamad, "A new benchmark image test suite for evaluating color texture classification schemes," *Multimedia Tools Appl. J.*, vol. 70, pp. 543–556, May 2014.
- [52] *New BarkTex Benchmark Image Test Suite for Evaluating Color Texture Classification Schemes*. Accessed: Jan. 12, 2017. [Online]. Available: [https://www.lisic.univ-littoral.fr/~porebski/BarkTex\\_image\\_test\\_suite.html](https://www.lisic.univ-littoral.fr/~porebski/BarkTex_image_test_suite.html)
- [53] D. Casanova, J. J. de Mesquita Sá, Jr., and O. M. Bruno, "Plant leaf identification using Gabor wavelets," *Int. J. Imag. Syst. Technol.*, vol. 19, pp. 236–243, Aug. 2009.
- [54] R. Kwitt and P. Meerwald. *Salzburg Texture Image Database (STex)*. Accessed: Jun. 13, 2018. [Online]. Available: <http://wavelab.at/sources/STex/>
- [55] A. R. Backes, D. Casanova, and O. M. Bruno, "Color texture analysis based on fractal descriptors," *Pattern Recognit.*, vol. 45, no. 5, pp. 1984–1992, May 2012.
- [56] *USPTex Dataset*. Accessed: Jun. 13, 2018. [Online]. Available: <http://fractal.ifsc.usp.br/dataset/USPTex.php>
- [57] L. Armi and S. Fekri-Ershad, "Texture image classification based on improved local quinary patterns," *Multimedia Tools Appl.*, vol. 78, pp. 18995–19018, Jul. 2019.

- [58] S. Fekriershad and F. Tajeripour, "Color texture classification based on proposed impulse-noise resistant color local binary patterns and significant points selection algorithm," *Sensor Rev.*, vol. 37, no. 1, pp. 33–42, 2017.
- [59] R. Bello-Cerezo, P. Fieguth, and F. Bianconi, "LBP-motivated colour texture classification," in *Proc. 2nd Int. Workshop Compact Efficient Feature Represent. Learn. Comput. Vis. (Conjunct ECCV)* in Lecture Notes in Computer Science, vol. 11132, L. Leal-Taixé and S. Roth, Eds. Munich, Germany: Springer, Sep. 2018, pp. 517–533.
- [60] F. Bianconi, R. Bello-Cerezo, and P. Napoletano, "Improved opponent color local binary patterns: An effective local image descriptor for color texture classification," *J. Electron. Imag.*, vol. 27, no. 1, 2018. Art. no. 011002.



**FRANCESCO BIANCONI** received the M.Eng. degree from the University of Perugia, Italy, and the Ph.D. degree in computer-aided design from a consortium of Italian universities. He has been a Visiting Researcher with the University of Vigo, Spain; the University of East Anglia, U.K.; the Queen Mary University of London, U.K. and City, University of London, U.K. He is currently an Associate Professor with the Department of Engineering, University of Perugia, where he conducts

research on computer vision, image processing, and pattern recognition with special focus on texture and color analysis. He is a Chartered Engineer and a Court-Appointed Expert.



**CLAUDIO CUSANO** graduated in computer science from the University of Milano–Bicocca, in 2002, and the Ph.D. degree in computer science from the University of Milano–Bicocca, in 2006. Since 2002, he has been a Fellow with the Multimedia Information Technologies Institute, Italian National Council of Research, where he started its research activity as a researcher with grant. In 2006, he became a Postdoctoral Researcher with the Imaging and Vision Laboratory, University of

Milano–Bicocca. Since 2012, he has been serving as an Assistant Professor in computer engineering with the University of Pavia, where he became an Associate Professor, in 2015. His topics of interest are focused on automatic image analysis and recognition, and include scene classification, texture analysis, color processing, face recognition, and 3D imaging.



**PAOLO NAPOLETANO** received the master’s degree in telecommunications engineering from the University of Naples Federico II, in 2003, with a thesis focused on transmission of electromagnetic fields, and the Ph.D. degree in information engineering from the University of Salerno, Italy, in 2007, with a thesis focused on computational vision and pattern recognition. He is currently an Assistant Professor of computer science (tenure track - RTDB) with the Department of Informatics,

Systems and Communication, University of Milano–Bicocca. His current research interests focus on signal, image and video analysis and understanding, multimedia information processing, and management and machine learning for multimodal data classification and understanding.



**RAIMONDO SCHETTINI** is currently a Professor with the University of Milano–Bicocca, Italy. He is also the Head of the Imaging and Vision Laboratory. He has been associated with the Italian National Research Council, since 1987, where he led the Color Imaging Laboratory, from 1990 to 2002. He has been a team leader in several research projects and published more than 300 refereed articles and six patents about color reproduction, and image processing, analysis, and classification.

He is a Fellow of the International Association of Pattern Recognition for his contributions to pattern recognition research and color image analysis.

...