https://doi.org/10.2352/lim.2023.4.1.28

Optimizing Gabor texture features for materials recognition by convolutional neural networks

***Francesco Bianconi*[1], *Claudio Cusano*[2], *Paolo Napoletano*[3], *Raimondo Schettini*[3]**
[1] ***Department of Engineering Università degli Studi di Perugia, Italy***
{*bianco@ieee.org*}
[2] ***Department of Electrical, Computer and Biomedical Engineering, University of Pavia, Italy***
{*claudio.cusano@unipv.it*}
[3] ***Department of Informatics, Systems and Communication, University of Milano - Bicocca, Italy***
{*paolo.napoletano@unimib.it, raimondo.schettini@unimib.it*}

Abstract

In this paper, we present a novel technique that allows for customized Gabor texture features by leveraging deep learning neural networks. Our method involves using a Convolutional Neural Network to refactor traditional, hand-designed filters on specific datasets. The refactored filters can be used in an off-the-shelf manner with the same computational cost but significantly improved accuracy for material recognition. We demonstrate the effectiveness of our approach by reporting a gain in discrimination accuracy on different material datasets. Our technique is particularly appealing in situations where the use of the entire CNN would be inadequate, such as analyzing non-square images or performing segmentation tasks. Overall, our approach provides a powerful tool for improving the accuracy of material recognition tasks while retaining the advantages of handcrafted filters.

Introduction

Materials recognition involves identifying the type of material a particular object is made of based on its appearance and visual properties, such as color, reflectance, transparency, texture and gloss [12]. In this context texture features can be extremely useful for recognizing materials that have distinctive textural patterns, such as wood, fabric, textiles, natural stone, concrete and metal – to cite some. By contrast, texture analysis can be less effective with those materials (e.g. glass or plastic) for which other visual properties come into play, such as specularity, transparency and gloss. We are therefore aware that texture features alone can be effective only for materials that have distinctive textured appearance, but may be less effective for materials that have more subtle or complex properties requiring Bidirectional-Reflectance-Distribution-Function (BRDF) measurements. Even so, we must recognize that there are application contexts in which only traditional RGB imaging techniques can be deployed.

Various techniques exist for extracting texture features from images, including Gabor filters, local binary patterns, and wavelet transforms (see [8, 9, 16, 13, 5] for recent surveys). In the classic computer vision pipeline these features are typically fed to some suitable machine learning algorithm – e.g. SVM, Random Forest, etc. – to carry out material recognition. Convolutional Neural Networks (CNNs), on the other side, have shown very good performance in material recognition – and generally better performance than conventional approaches – particularly when large amounts of data are available [3, 10]. There are, however several situations in which classic computer vision could be preferred, for example when:

- Real-time performance is critical and/or hardware resources are inadequate;
- There are limited data for training;
- The task requires explainability and interpretability of the material recognition process, the so-called Explainable Artificial Intelligence (XAI).

Another limitation of many CNN-based solutions for texture and material recognition is that they are bound to operate on images of fixed shape and size, which results in two orders of problems. First, the difficulty to analyse free-form images; second, the fact that the resizing needed to make the input image fit the field of view of the newtork may wipe out important textural information. The latter, in particular, is critical when it comes to classifying textures that look very similar to one another (surface grading is one typical application).

In this paper we present a novel technique that allows for customized hand-crafted texture features by leveraging deep learning neural networks. Concretely, we propose a method that utilizes a convolutional network to optimize banks of linear filters. By initializing the first convolutional layer of the network with classic filter banks and training it on generic texture images, the network modifies the filter weights for optimal performance. As a case study we consider a classic Gabor filter bank and its CNN-refactored version for texture classification. Experimenting on seven independent datasets of texture images we show that our approach generates optimized filters which outperform the original ones (accuracy gain between 1.90 and 7.98 pp) at no additional computational cost.

Method

Gabor filters represent a widely used tool for feature extraction, especially for texture and material images. There are several explanations of their popularity, one is that Gabor filters mimic the behaviour of simple cells in the primary visual cortex [18], and they have been found to achieve optimal joint resolution in the space and frequency domain [11]. Gabor filters are defined as two-dimensional sinusoidal functions modulated by a Gaussian envelope:

$$G(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(\frac{2\pi i x'}{\lambda}\right), \tag{1}$$

where $x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$. In the experiments we used a bank of Gabor filters in the spatial domain with three wavelengths ($\lambda \in \{k, k/\sqrt{2}, k/2\}$) and four orientations ($\theta \in \{0, \pi/4, \pi/2, 3\pi/4\}$) all discretized and truncated to a grid of $k \times k$ pixels. We set the spread σ of the Gaussian window to $(k-1)/5$ and the shape ratio γ to one (i.e.: circular filters).

Figure 1. The original bank of Gabor filters before refactoring.

Real and imaginary part were considered separately, giving a total of $3 \times 4 \times 2 = 24$ filters.

For the grid size we determined that $k = 11$ px was a good trade-off between filter resolution and computational cost. Preliminary tests indeed confirmed that this value was a reasonable choice for most datasets of texture images. Furthermore, $k = 11$ px is common size for filters in the first layer of convolutional neural networks. This plays an important role in our strategy, as we shall discuss in the next section (see Figure 1).

Filter refactoring using CNN

We propose here, a method for the computation of texture features based on the refactoring of linear Gabor filters. The method uses a Convolutional Neural Network especially designed by the authors for texture classification. The network parameters are randomly initialized, except for those in the first convolutional layer: these start the training as a bank of Gabor filters, and become the new specialized feature extractors once the training is completed.

The CNN was designed to process variable size images with a stationary distribution of local features, like those typically used in texture and material recognition. It is a fully convolutional network up to the very last layers. A sequence of ten convolutions computes local features. These are processed as a set of local descriptors, and combined into a single global feature vector by a pooling layer. A final softmax yields the class posterior probabilities. All convolutions except the last one are followed by the ReLU activation function and a batch normalization layer. The architecture of the neural network is summarized in Table 1.

Architecture of the convolutional neural network. All convolutional layers are followed by a ReLU activation function and batch normalization.

Operation	Image size
Input color image	$192 \times 192 \times 3$
Grayscale conversion	$192 \times 192 \times 1$
11×11 Convolution (stride 8)	$24 \times 24 \times 24$
3×3 Convolution (stride 1)	$24 \times 24 \times 64$
3×3 Convolution (stride 2)	$12 \times 12 \times 64$
3×3 Convolution (stride 1)	$12 \times 12 \times 128$
3×3 Convolution (stride 2)	$6 \times 6 \times 128$
3×3 Convolution (stride 1)	$6 \times 6 \times 256$
3×3 Convolution (stride 2)	$3 \times 3 \times 256$
1×1 Convolution (stride 1)	$3 \times 3 \times 512$
1×1 Convolution (stride 1)	$3 \times 3 \times 512$
1×1 Convolution (stride 1)	$3 \times 3 \times 250$
3×3 Average pooling	250
Softmax	250

The first convolutional layer is the most important. As detailed in the previous section it uses a bank of 24 filters of dimension 11 px $\times$ 11 px. Its weights are initialized as a bank of Gaussian filters, and they will evolve into the final feature extractors. The rest of the network makes use of the linear features to classify texture images.

The network was trained with the Adam optimization method on the ALOT dataset [1], which includes 25 000 images in 250 different classes of texture. Random cropping and rotation was used to augment the training set. At the end of the training the refactored Gabor filters were retrieved from the first convolutional layer. The result is shown in Figure 2.

As can be seen most filters maintained their overall original “shape”; observe, however, that non-zero weights now extend to all the filter support (compare Fig. 1 and 2). In some cases, entirely new patterns emerged. These filters can be used as simple and cheap feature extractors, useful for those applications that cannot afford to use CNNs. For each filter we computed the mean and standard deviation of the magnitude of the transformed images and formed a 48 dimensional feature vector.

Experimental results

To assess the quality of the refactored filters we designed a set of experiments in which a linear Support Vector Classifier (SVC) classifies material and texture images taken from several datasets. More in detail, we used the following eight datasets of texture or material images (see also Tab. 2 and Fig. 3 for a recap): Amsterdam Library of Textures (ALOT) [1], that we use for filter refactoring, and Columbia-Utrecht Reflectance and Texture Database (CUReT) [19], KTH-TIPS [14], KTH-TIPS2b [6], New BarkTex [17], Plant leaves [7], Salzburg Texture Image Database (STex) [15], and USPTex [2], that are used for testing.

Each dataset was randomly split in a training and a test set of equal size, and this process was repeated 100 times to ensure an accurate estimate of the classification error made by the Support Vector Classifier (SVC). For each split the SVC hyperparameters were determined by grid search and five-fold cross validation on the training set.

Table 3 compares the classification accuracy obtained by using the original Gabor filters and their refactored version. For all

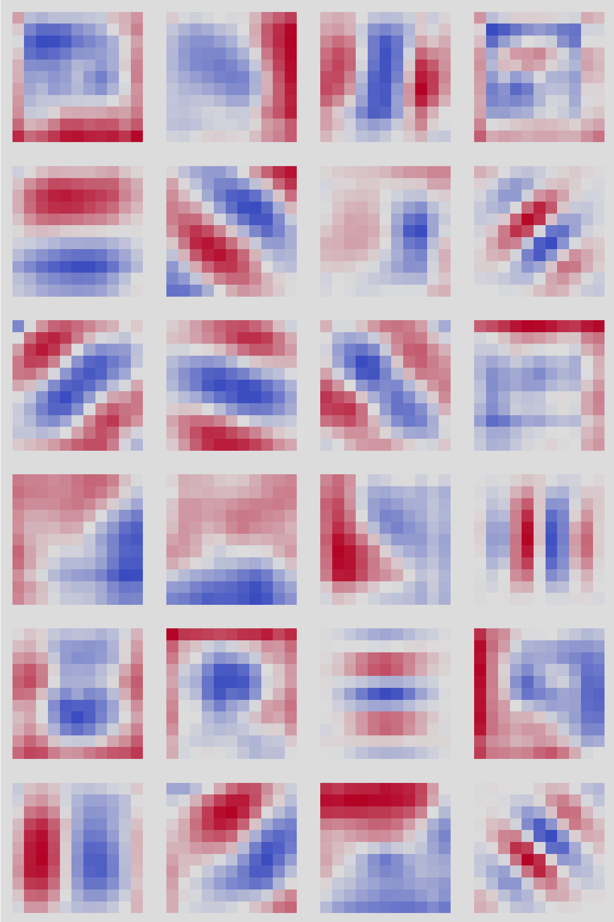

Figure 2. *The bank of refactored Gabor filters.*

Datasets used in the experiments: summary table.

Dataset	Classes	Samples per class	Image size
ALOT	250	100	1536 × 1024
CUReT	61	93	200 × 200
KTH-TIPS	10	81	200 × 200
KTH-TIPSb	11	432	200 × 200
NewBarkTex	6	273	64 × 64
PlantLeaves	20	60	128 × 128
STex	476	16	128 × 128
USPTex	191	12	128 × 128

Classification accuracy (%): original vs. refactored filters.

Dataset	Original	Refactored	Improvemet
CUReT	89.23	92.51	+3.28
KTH-TIPS	90.35	91.84	+1.49
KTH-TIPS2b	88.16	89.51	+1.35
NewBarkTex	87.44	88.67	+1.23
PlantLeaves	73.23	81.21	+7.98
STex	72.40	78.00	+5.60
USPTex	82.96	84.86	+1.90

datasets, the refactored filters clearly outperformed the original ones, obtaining better accuracy in all the cases. The improvement in accuracy ranged from a minimum of 1.23 pp (NewBarkTex) to a maximum of almost 8 pp (PlantLeaves). The average improvement across all the datasets was about 2.85 pp.

Conclusions

Materials recognition is a computer vision task in which texture analysis plays a fundamental role. In this work we have presented a technique that allows for customized handcrafted texture features by leveraging deep learning neural networks. As a case study we have shown how traditional Gabor filters can be refactored to generate a bank of linear filters with better classification accuracy and the same computational cost as the original ones. The initial Gabor filters were modified by the network using ALOT as the training dataset and tested on seven independent datasets using a simple linear SVC as the classification method. We chose Gabor filters as our proof-of-concept since they are one of the most popular approaches for texture description, and also because of their close relationship with the early visual systems of mammals [18]. Our method, however, extends seamlessly to other banks of linear filters.

There is, in our opinion, still room for improvement in the overall performance, e.g., using a different classifier or, further refining the filters on a small set of samples from the specific domain of application. However, what we wanted to show was that, not only can hand-crafted features be used to improve the performance of networks, for example, through feature injection [4], but that the reverse is also true: by using CNN networks not just as simple black boxes it is possible to improve the performance of hand-crafted features. The present investigation was limited to the analysis of grey scale textures; in future studies we plan to extend the method to colour textures.

References

[1] Amsterdam library of textures (ALOT). Available online at: `http://aloi.science.uva.nl/public_alot/`. Last accessed on Jun. 13, 2018.

[2] USPTex dataset. Available online at: `http://fractal.ifsc.usp.br/dataset/USPtex.php`. Last accessed on Jun. 13, 2018.

[3] 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. *Applied Sciences*, 9(4), Feb. 2019. Article number: 738.

[4] S. Bianco, D. Mazzini, P. Napoletano, and R. Schettini. Multitask painting categorization by deep multibranch neural network. *Expert Systems with Applications*, 135:90–101, 2019.

[5] F. Bianconi, A. Fernández, F. Smeraldi, and G. Pascoletti. Colour and texture descriptors for visual recognition: A historical overview. *Journal of Imaging*, 7(11), Nov. 2021. Art. no. 245.

[6] B. Caputo, E. Hayman, and P. Mallikarjuna. Class-specific material categorisation. In *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV'05)*, volume 2, pages 1597–1604, Beijing, China, Oct. 2005.

[7] D. Casanova, J. J. de Mesquita Sá Jr., and O. Martinez Bruno. Plant leaf identification using Gabor wavelets. *International Journal of Imaging Systems and Technology*, 19(3):236–243, Aug. 2009.

[8] C. Cusano, P. Napoletano, and R. Schettini. Combining local binary patterns and local color contrast for texture classification under varying illumination. *JOSA A*, 31(7):1453–1461, 2014.

[9] C. Cusano, P. Napoletano, and R. Schettini. Combining multiple features for color texture classification. *Journal of Electronic Imaging*, 25(6):061410–061410, 2016.

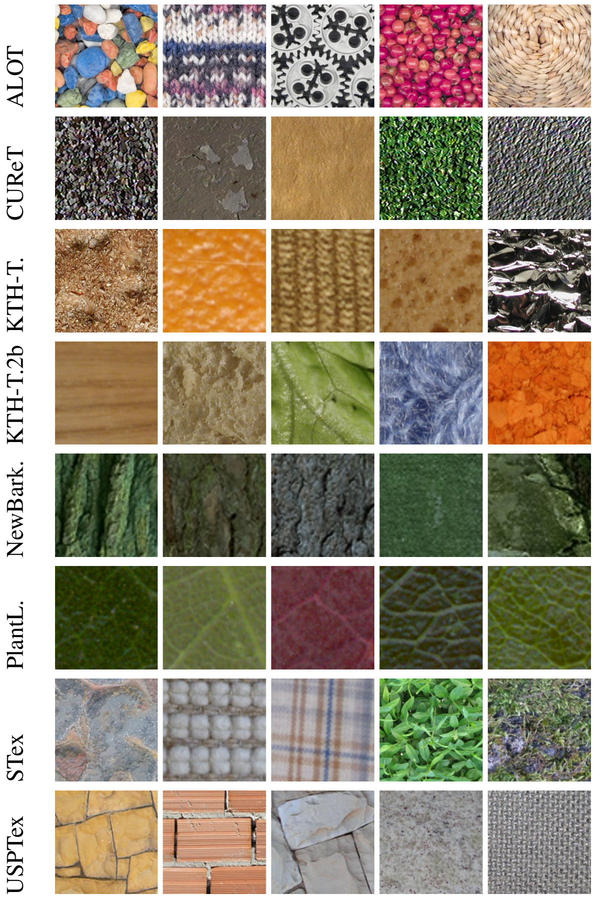

Figure 3. *Samples of the texture images used in the experiments.*

[10] C. Cusano, P. Napoletano, and R. Schettini. T1k+: A database for benchmarking color texture classification and retrieval methods. *Sensors*, 21(3):1010, 2021.

[11] G. Daugman. Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *Journal of the Optical Society of America A*, 2(7):1160–1169, 1985.

[12] C. Eugène. Measurement of “total visual appearance”: A CIE challenge of soft metrology. In *Proceedings of the 12th IMEKO TC1 Education and Training in Measurement and Instrumentation and TC7 Measurement Science Joint Symposium on Man, Science and Measurement*, Sept. 2008.

[13] A. Humeau-Heurtier. Texture feature extraction methods: a survey. *IEEE Access*, 7:8975 – 9000, 2019. Art. no 8600329.

[14] The KTH-TIPS and KTH-TIPS2 image databases. Available online at: `http://www.nada.kth.se/cvap/databases/kth-tips/download.html`. Last accessed on Jan. 11, 2017.

[15] R. Kwitt and P. Meerwald. Salzburg texture image database (STex). Available online at: `http://wavelab.at/sources/STex/`. Last accessed on Jun. 13, 2018.

[16] 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. *International Journal of Computer Vision*, 127(1):74–109, jan 2019.

[17] New [barktex] benchmark image test suite for evaluating color texture classification schemes. Available online at: `https://www-lisic.univ-littoral.fr/~porebski/BarkTex_image_test_suite.html`. Last accessed on Jan. 12, 2017.

[18] M. Turner. Texture discrimination by Gabor functions. *Biological Cybernetics*, 55(2-3):71–82, 1986.

[19] Visual Geometry Group. CUReT: Columbia-Utrecht Reflectance and Texture Database. Available online at: `http://www.robots.ox.ac.uk/~vgg/research/texclass/setup.html`. Last accessed on Jan. 26, 2017.