Image retrieval based on texture using latent space representation of discrete Fourier transformed maps

Article *in* Neural Computing and Applications · April 2021

DOI: 10.1007/s00521-021-05955-2

CITATIONS
0

READS
125

4 authors:

Surajit Saikia
Universidad de León
6 PUBLICATIONS **49** CITATIONS
SEE PROFILE

Laura Fernández-Robles
Universidad de León
78 PUBLICATIONS **496** CITATIONS
SEE PROFILE

Enrique Alegre
Universidad de León
199 PUBLICATIONS **1,330** CITATIONS
SEE PROFILE

Eduardo Fidalgo
Universidad de León
52 PUBLICATIONS **326** CITATIONS
SEE PROFILE

Noname manuscript No.
(will be inserted by the editor)

Image Retrieval based on Texture Using Latent Space Representation of Discrete Fourier Transformed Maps

Surajit Saikia[1,2] · **Laura Fernández-Robles**[2,3] · **Enrique Alegre**[1,2] · **Eduardo Fidalgo**[1,2]

Received: date / Accepted: date

Abstract Texture-based instance retrieval is typically performed on images that present a single texture pattern and is mainly applied to the retrieval of fabrics or textiles. In this work, we apply it to indoor scene images that typically present many different texture patterns, which constitutes a more challenging problem. Such retrieval systems, together with the retrieval of faces and objects, can be used as a valuable tool for evidence matching in crime scene investigation. Even though recent deep learning-based approaches have made significant improvement in many computer vision tasks, texture retrieval remains an open problem. In this work, we introduce a Fourier based approach, in which spatial images and their discrete Fourier transform maps are combined to derive a novel texture representation. We further present a new and efficient texture-based image retrieval framework based on region proposal networks, convolutional autoencoders and transfer learning, in which we extract the features from the latent space layer of the encoder as texture descriptors. The experimental results on four datasets: TextileTube, Outex, USPtex and Stex, validated the effectiveness of our proposed method, yielding better results than the current state-of-the-art.

Keywords Texture retrieval · Convolutional autoencoders · Texture classification · Discrete Fourier transform

✉ Surajit Saikia
E-mail: ssai@unileon.es
ORCID: 0000-0001-7757-1547
· Laura Fernández-Robles
E-mail: l.fernandez@unileon.es
ORCID: 0000-0001-6573-8477
· Enrique Alegre
E-mail: ealeg@unileon.es
ORCID: 0000-0003-2081-774X
· Eduardo Fidalgo
E-mail: efidf@unileon.es
ORCID: 0000-0003-1202-5232
[1]Department of Electrical, Systems and Automation Engineering, Universidad de León, Spain.
[2]Researcher at INCIBE (Spanish National Cybersecurity Institute), León, Spain.
[3]Department of Mechanical, Informatics and Aerospace Engineering. Universidad de León, Spain.

1 Introduction

Over the past two decades, due to a substantial increase on the volume of image data acquired through multimedia devices, content-based image retrieval (CBIR) [1–3] and image search [4–6] has attracted significant attention. CBIR methods are automatic approaches for finding similar images based on a given query from a huge collection of structured and unstructured images. To create an accurate image retrieval system, finding discriminative features is essential. The texture is an important visual property perceived in objects that characterize them, and it could be a useful and discriminative feature for CBIR purposes.

Among many CBIR applications, texture retrieval can play a key role in examining crime scenes for forensic analysis [7–9]. In a crime scene, the clues derived from images may empower the investigative work of forensic departments. Image retrieval for crime scene investigation (CSI) [10, 11] can help to uncover various crimes by linking similar images or videos with a police case. In Fig. 1, we illustrate some of the images provided by Europol, the European Union's Law Enforcement Agency (LEA), for one of their activities which aims at stopping child abuse by tracing objects[1].

[1] https://www.europol.europa.eu/stopchildabuse

Fig. 1 Sample images from Europol's 'stop child abuse - trace an object' activity.

As illustrated, some of these images are texture patches that do not contain much information about the object contour. One way to trace the objects is to compare them against dense database environments to look for similarities with other suspicious images using a CBIR system. To define the characteristics of such images, the texture patterns are the prime cues for visual descriptions. Texture-based image retrieval was included as a task in the European projects ASASEC[2] and currently in GRACE[3]. These projects, in which some European LEAs are or were involved, aim to provide solutions based on computer vision and deep learning to help LEAs to fight against child abuse.

Texture description has been extensively investigated in the computer vision domain for different purposes such as object recognition, remote sensing, image retrieval and medical image analysis [12–18]. Given a texture patch as a query, searching it among a huge collection of images in a dataset is a challenging task. In general, most of CBIR systems [19] extract features that represent various shapes and patterns present in an image. However, a query patch containing just a texture pattern without contour information presents a more challenging scenario to effectively retrieve images with the same or a similar texture.

The presented work is inspired by the recent advances of deep learning algorithms [13, 20–24] in the computer vision domain. Convolutional Neural Networks (CNN) have proven to be very successful for solving computer vision tasks, and descriptors extracted from CNNs have been fruitfully applied to image retrieval [25, 26]. Also, autoencoders [27] have drawn lots of attention in the field of image processing. Autoencoders can be used in image retrieval [28, 29] because the target output of an autoencoder is the same as its input and similar images produce similar latent space representations. The most important advantage of an autoencoder based architecture is that the network can be trained with diverse classes of texture images, and still, a compact and informative feature representation can be achieved.

[2] https://ec.europa.eu/home-affairs/financing/fundings/projects/HOME_2010_ISE_AG_043_en

[3] https://cordis.europa.eu/project/id/883341

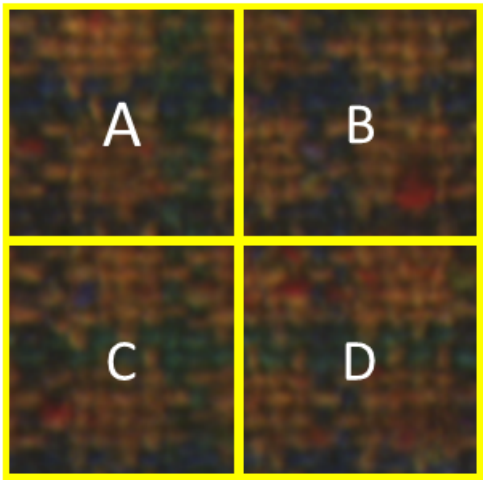

Fig. 2 Example of four texture patches, framed in yellow, containing similar repeated patterns. The four patches were cropped from the same object.

Until the advent of deep learning, in the image processing domain the Fourier transform [30, 31] was widely used to process and represent images. It is able to extract the dominant spatial frequency and orientations of structures present in an image. The texture images usually contain quasi repetitive patterns, and Fourier transform can represent those periodic functions present in the form of repetitive patterns. In Fig. 2, we show an example of four texture patches, marked by boxes A, B, C and D, containing similar repeated patterns, which illustrates the types of patches that we will consider as queries. In Fig. 3, we illustrate the motivation behind using the Discrete Fourier Transform (DFT) to represent texture patches. It can be seen that texture patches that belong to different classes present distinguishable DFT maps, whereas images of the same class have similar DFT maps. Although the frequency domain features are robust to noise [32], it is not sufficient to use only Fourier transform since many texture images might have similar frequencies. Therefore, in order to make texture representation highly discriminative, we decided to combine information from both spatial and frequency domains.

In this paper, we propose a novel texture retrieval method based on Fourier transform and deep learning techniques. Our approach uses an autoencoder, and we apply transfer learning to the encoder using a VGG-16 network pre-trained with ImageNet dataset [33]. We combine the texture representation of the DFT transformed images with the corresponding spatial images, and the resultant images are used to train the latent space layer and the decoder in order to learn their texture representation. After training the network, the VGG-encoder with its latent space layer serves as a fea-

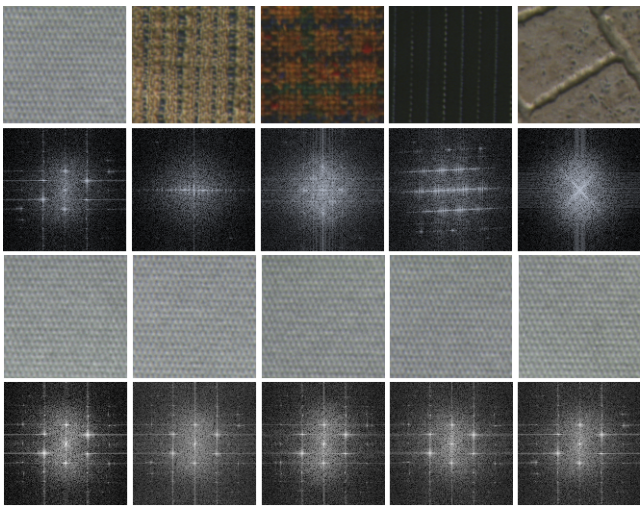

Fig. 3 DFT of texture images: Row 2 represents the DFT maps of different texture patterns from row 1. In row 3, we have texture patterns from the same class, and row 4 represents their DFT maps.

ture extractor to generate texture representation, which we name as Deep Fourier Texture Descriptor (DFTD). We further integrate the VGG-encoder and latent space layer with the region proposal network (RPN) of the R-FCN [34] to generate DFTD of multiple texture proposals concerning an image, so that a queried texture patch can be compared locally against various proposals. In Fig. 4, we outline our proposed method using two stages. Stage 1 illustrates the generation of texture descriptors and stage 2 represents the retrieval framework. To the best of our knowledge, this work presents a new approach where an autoencoder based architecture with transfer learning and the Fourier transform are used for texture classification and retrieval.

The main contributions of this study are as follows:

1. We introduce a novel texture descriptor known as Deep Fourier Texture Descriptor (DFTD). The features of the descriptor are extracted from the latent space layer of a convolutional autoencoder whose inputs are the outcomes of blending the magnitude spectrum of a DFT and the spatial information of the images.
2. We propose a CBIR framework for texture retrieval which uses an RPN to propose prospective texture regions which are fed to an autoencoder. The model was trained with transfer learning techniques.
3. We evaluate the proposed texture-based image retrieval approach in the context of a real-world application, which can be applied for image, instance and object retrieval for crime scenes investigation during forensic analysis. We also assess the proposed texture descriptor both for texture-based image retrieval and for texture classification on four public datasets, where the proposed method outperformed some recent and relevant state-of-the-art works.

The remainder of the paper is organized as follows. Section 2 discusses the related work. The proposed method is described in Section 3. In Section 4 we present the datasets, experiments are provided in Section 5, and finally, presents the main conclusions we draw doing this work.

2 Related Work

In this section, we present the most relevant previous works related to image and texture retrieval based on both handcrafted methods, Fourier and CNN-based approaches. Creating efficient texture descriptors to characterize the image is essential in works related to texture-based image retrieval and classification [35–37]. In the bulk of literature, many descriptors were recently proposed for texture analysis, for example, in [38], a rotation-invariant texture descriptor was proposed to address classification task, and *Pham et al.* [39] introduced a method for texture retrieval using multiscale feature extraction. For texture image recognition, *Tuncer et al.* [40] used a neural network for texture feature extraction, and later, introduced a novel chess based local image descriptor [41] for texture feature extraction inspired by chess game. The main objective of all such works is to meet the demands of certain applications with regard to CBIR. For example, texture-based image retrieval has been used in the textile and clothing industry [42], where clothes and textiles can be represented by texture descriptors. Furthermore, in textile stores, retrieving desired textile images from huge databases using a query is a need for both customers and retailers to suggest products [43, 44].

Existing image retrieval techniques usually use keywords annotated on images to search related similar images on a dataset. Image retrieval methods in [45, 46] used the Bag of Visual Words (BoVW) and considered texture and colour keywords to retrieve relevant images. However, such manual annotations increase time complexity, and also significantly ignore primary visual features such as textures, colours, and shapes. In order to reduce this semantic gap, some domain-independent visual-based descriptors were proposed to define objects of interests for retrieval tasks [47, 48]. Such descriptors were proposed to support automatic recognition [49, 50]; however, detail annotations are required and they are computationally expensive.

CBIR descriptors were traditionally subdivided into local and global. Handcrafted local descriptors such as

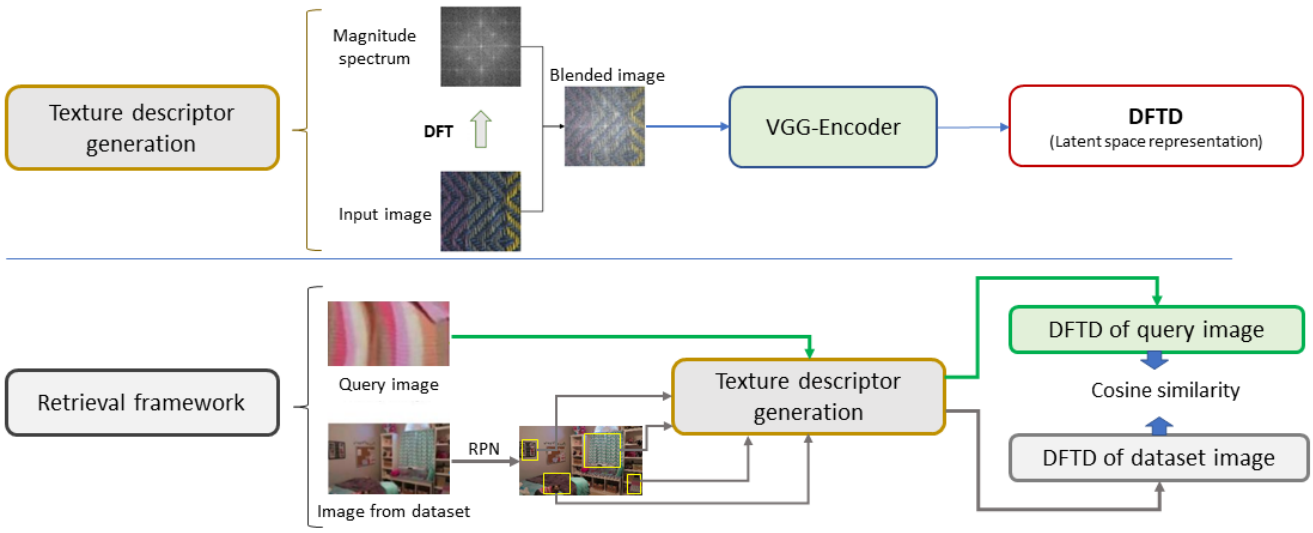

Fig. 4 Overview of our proposed query-based retrieval method. Stage 1 represents generation of texture descriptor DFTD, and stage 2 illustrates the retrieval framework.

SIFT [51], SURF [52], Histogram of Oriented Gradients (HOG) [53], Local Binary Pattern (LBP) [54] extract primary visual information from local patches of an image while the global descriptors extract meaningful information from the full image. Local descriptors detect keypoints and describe the texture of the regions of interest around them. Sometimes various descriptors are used together to provide joint information about image content, for example, LBP and Local Neighborhood Difference Pattern (LNDP) were combined for texture-based image retrieval [55]. Similarly, local texture-based colour histogram combines texture and colour descriptions for retrieval [56], and *Singh et al.* [57] introduced a new local colour descriptor named Local Binary Patterns for Color images (LBPC). *Pavithra et al.* [58] proposed a new hybrid framework using LBP and Canny edge detector for CBIR to address issues related to accuracy associated with traditional image retrieval systems.

Regarding global descriptors for retrieving similar images, the shape details of images can be represented in the form of salient features, which can be obtained using Fourier transform. *Sokic et al.* [31] proposed a method for extracting Fourier descriptor for shape-based image retrieval by preserving phase, and recently, *Yang et al.* [59] introduced a novel multi-scale Fourier descriptor based on triangular features to identify shapes. This is different from what we present because, in our work, we introduce an autoencoder based architecture with transfer learning, and train it using blended images obtained by combining spatial images and images representing the magnitude spectrum of the DFT.

Apart from the recent use of the Fourier transform with CNNs, handcrafted and Fourier descriptors dominated many computer vision applications until the advent of deep learning based approaches [21, 22, 60–62]. The deep learning based CNNs can be used as a feature extractor to generate neural codes [25, 63, 64] as global representations of images for image retrieval [65–67]. *Cimpoi et al.* [66] proposed a texture descriptor known as FV-CNN, which is obtained by Fisher vector pooling of a CNN filter bank. *Babenko et al.* [25] introduced an image retrieval method, where a pre-trained CNN model was retrained and tested in different datasets and achieved improved performance. Most recently, *Yikun et al.* [68] presented an image retrieval technique which uses a CNN-based architecture for deep representation of images.

Since global feature representation ignores the local spatial information of images, region-based CNNs are employed to extract features locally for query-based image retrieval [12, 13, 69]. For example, *Salvador et al.* [64] fine-tuned a Faster RCNN network with the type of images aimed at retrieval by taking advantage of RPN to generate object proposals and compared them against the query image.Similarly, in our work, we employ RPN to generate region proposals on the Textile-Tube dataset to compare against the query patches for retrieving texture images.

3 Method

The proposed method consists of two stages (see Fig. 4): (1) texture descriptor generation, and (2) instance

retrieval framework. In the first stage, we compute the texture descriptors of images created by a pixel-wise combination of the magnitude spectrum DFT and the spatial domain of an image. We also propose a new architecture based on autoencoders to extract the texture descriptors, which we train with the generated images. In the second stage of Fig. 4, we illustrate our complete query based retrieval framework using texture descriptors. In addition, we also present a texture classification schema to validate the efficacy of the proposed texture descriptors.

3.1 Texture descriptor generation

3.1.1 Discrete Fourier transform of images

The Fourier transform is a classical image processing tool which decomposes images into sine and cosine components. The output generated by Fourier transform represents the image in the frequency or Fourier domain, while the input, which is the user-provided image, is the spatial domain equivalent. Each pixel in a Fourier domain image represents a particular frequency contained in the spatial domain image.

To generate frequency images, we use the discrete Fourier transform (DFT), which is the sampled Fourier transform that contains enough frequencies to fully describe an image in a spatial domain. Moreover, the image generated by DFT is of the same size as that of the spatial image, and hence, we can train a network without modifying the input size parameters.

The DFT for an image of size $M \times N$ pixels denoted by $f(x, y)$, with x and y its spatial coordinates is given by:

$$F(u,v) = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(x,y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}, \tag{1}$$

where u and v are in the range $[0, 1, 2, ..., M-1]$ and $[0, 1, 2, ..., N-1]$, with $i^2 = -1$ is the complex imaginary number. One of the properties of the DFT is that the original image $f(x, y)$ can be obtained by applying inverse DFT to $F(u, v)$, which is defined as:

$$f(x,y) = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} F(u,v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})}. \tag{2}$$

Since the DFT is a bijective function, $f(x, y) \iff F(u, v)$, the DFT of an image satisfies the following two properties:

$$f(x,y) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})} \iff F(u - u_0, v - v_0) \tag{3}$$

$$F(x - x_0, y - y_0) \iff F(u,v) e^{-i2\pi(\frac{ux_0}{M} + \frac{vy_0}{N})} \tag{4}$$

In the DFT image, the zero frequency component is present in the top left corner, and to bring it to the center the DFT result is shifted by $M/2$ and $N/2$ in both directions u and v. Based on the properties defined in Eq. 3 and 4, the Fourier transform of an image can be shifted using the transformation:

$$f(x,y)(-1)^{(x+y)} \iff F(u - M/2, v - N/2), \tag{5}$$

so that $F(0, 0)$ is at the point $(u_0, v_0) = (M/2, N/2)$. Since, Fourier transform is a complex-valued function of a real-valued function, we can write it as

$$F(u,v) = R(u,v) + iI(u,v), \tag{6}$$

where R is the real part and I is the imaginary part, respectively. Finally, we generate the magnitude spectrum as

$$F_m(u,v) = 20 \times \log |\sqrt{R^2(u,v), iI^2(u,v)}|, \tag{7}$$

as the discrete Fourier representation of input image $f(x, y)$. In the paper, we refer to $F_m(u, v)$ as DFT magnitude spectrum image, and for the shake of brevity, DFT image.

Intuition of DFT and RPN framework. In Fig. 5, we illustrate the intuition behind our proposed approach to show how the DFT can be used for creating distinctive features for image retrieval and classification. Let $f(x, y)$ be the image in the spatial domain and $F_m(u, v)$ be its corresponding DFT magnitude spectrum image. The task is to find out if the image $f'(x, y)$ contains the instance $f(x, y)$ or not. We compute the DFT $F'_m(u, v)$ of the image $f'(x, y)$, which represents the Fourier spectrum of the complete image, and hence the DFT image differs from $F_m(u, v)$ due to the presence of multiple textures. The image $f'(x, y)$ posses certain frequencies, and we need to search if there are regions in it that might contain frequency similar to $f(x, y)$. The red box overlaid on image $f'(x, y)$ represents the ground truth, and that particular region contains the same frequency as that of $f(x, y)$. However, to generate possible regions in $f'(x, y)$ we need an external region proposal system. In order to address this issue, we employ an RPN to generate proposals, and then we compute the DFT magnitude spectrum images for each of the proposals. On the right part of Fig. 5, we present three examples of proposals generated from $f'(x, y)$ as $f'_1(x, y)$, $f'_2(x, y)$ and $f'_3(x, y)$, and their corresponding magnitude spectrums DFT images $F'_{m1}(u, v)$, $F'_{m2}(u, v)$ and $F'_{m3}(u, v)$, respectively. The proposal $f'_2(x, y)$ overlaid with the red bounding box represents a texture patch of the same class as the query $f(x, y)$, and also we

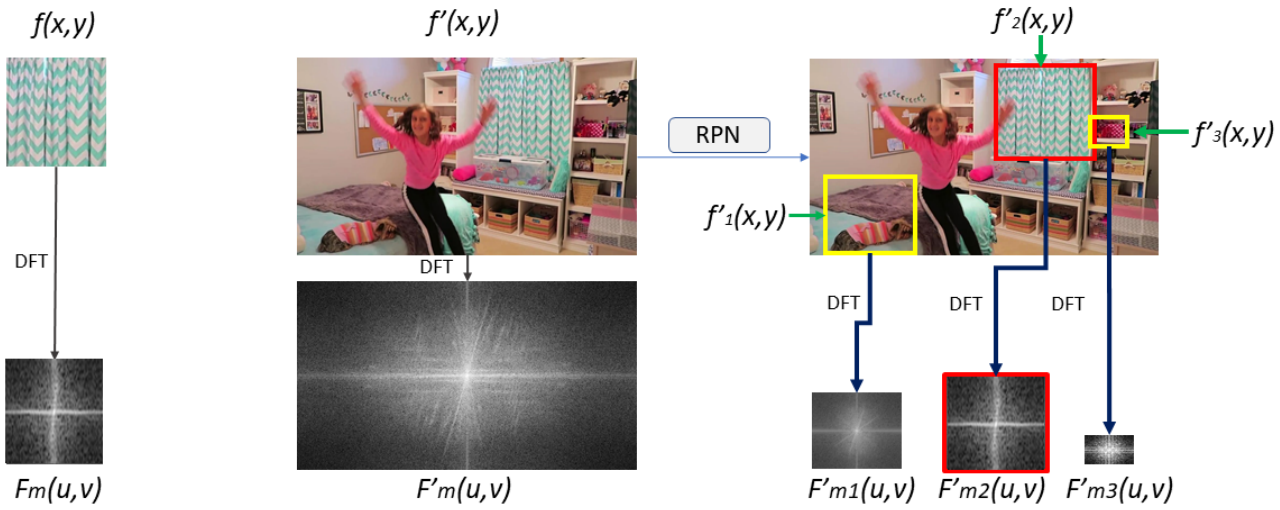

Fig. 5 Some images and their corresponding magnitude spectrum DFT maps to illustrate the motivation behind our approach. In the left, a query image. In the middle, a dataset image. The magnitude spectrum DFT map contains the frequencies of the query image but unrecognizable with other frequencies from other textures in the image. In the right, some texture patches are detected with an RPN, they are represented with overlaid yellow and red rectangles, and the magnitude spectrum DFT maps are computed. The red rectangle identifies a texture proposal of the same class as the query.

can observe that they have similar magnitude spectrum DFT images, as represented by $F_m(u, v)$ and $F'_{m2}(u, v)$. Therefore, this property of DFT can be used with spatial images to derive representations to search similar images for retrieval.

3.1.2 Linear blending of DFT magnitude spectrum and spatial images

Fourier transform has the very useful property of highlighting the dominant spatial frequencies as well as the orientations of the structures contained in an image. In our problem, the texture patterns present in the images are those structures. The texture images usually contain quasi repetitive patterns, and Fourier transform can define those periodic functions present in the form of repetitive patterns. However, while dealing with a big corpus of images, the frequency information is not sufficient to distinguish correctly different texture patterns, as some images belonging to different classes might have similar DFT magnitude spectrum representations. Hence, to address this issue, we combine both frequency and spatial information of the image by doing pixel-wise weighted addition of the DFT magnitude spectrum image $F_m(u, v)$ and the spatial image $f(x, y)$ to obtain a blended image $B(x, y)$, as defined in Eq. 8 and shown in Fig. 7.

$$B(x, y) = (1 - \alpha)f(x, y) + \alpha F_m(u, v) \tag{8}$$

The parameter α can be varied from 0 to 1 to stress $F_m(u, v)$ or $f(x, y)$. We empirically selected α equal to 0.7.

3.1.3 Proposed architecture

The proposed architecture is based on convolutional autoencoders, where the encoder part consists of the convolutional layers of the VGG-16 architecture initialised with ImageNet weights. In general, a convolutional autoencoder extends the basic structure of the simple autoencoder by changing the fully connected to convolution layers. The encoders learn to encode the input in a set of simple signals and then try to reconstruct the input from them based on the latent space representation learned by the network. However, our architecture varies from traditional autoencoders, since on the one hand, we apply transfer learning to the encoder, and on the other hand, the number of layers in the encoder and decoder are not the same. We next explain the architecture in detail as illustrated in Fig. 6.

VGG encoder. Creating a network and training it from scratch is expensive in terms of computational cost and availability of annotated data. Besides, it requires an optimisation of the network hyper-parameters to minimize classification error. This complexity can be avoided by using a pre-trained network, and henceforth, we use VGG-16 model pre-trained with ImageNet dataset as an encoder. In addition, another main reason for selecting VGG-16 is that it is a sequential network,

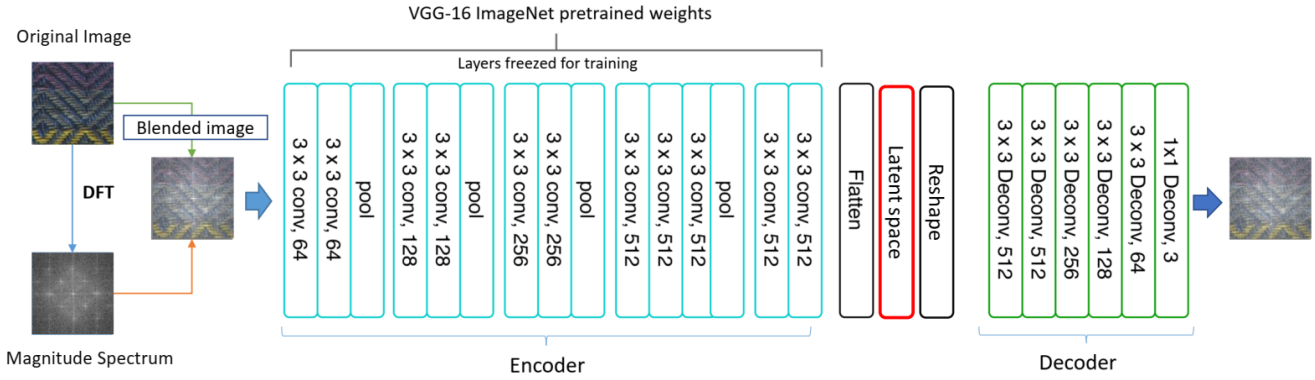

Fig. 6 Training architecture of the convolutional autoencoder with DFT based blended images.

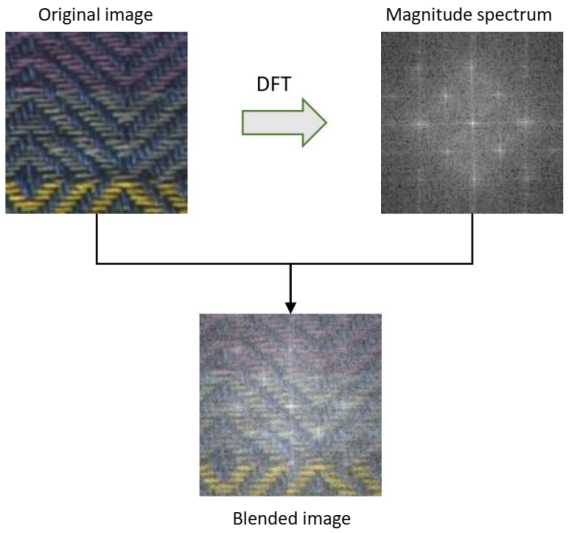

Fig. 7 Blended image $B(x, y)$ generated by the weighted addition of the original image and the DFT magnitude spectrum image

which facilitates the construction of a simplified convolutional autoencoder. It is true that other architectures such as ResNet and InceptionNet can also be used as encoders but they have some disadvantages. With such networks, we need to create a complex decoder since they are not sequential, and it will make the overall architecture computationally expensive in terms of both space and time. Hence, we choose the VGG-16 as an encoder, and we train the complete architecture with blended texture patches. In Fig. 6, a blended texture patch $B(x, y)$ is given as an input to the VGG encoder, which is followed by the latent space representation and the decoder.

Latent space representation as DFTD. In between the VGG encoder and the decoder we have the latent space layer (Fig. 6), which contains a vectorised representation of the input image of dimension 512. The 512-dimensional vector defines the proposed descriptor DFTD. The decoder takes this encoded vector and builds the output image to be as close to the input image as possible. The architecture learns to encode a DFT based blended image $B(x, y)$ in a set of simple signals and then tries to reconstruct the input from them based on the representation from the latent space layer. In Fig. 6, we represent the architecture of our proposal. The first convolution layer accepts the DFT based blended image $B(x, y)$, and the decoder learns to convert the latent space representation back to the same input image $B(x, y)$ as close as possible. In this way, the network learns specific texture representations on top of the ImageNet features so that it can generalize to different types of texture images for texture feature extraction.

3.2 Retrieval Framework

In this section, we describe the complete texture retrieval framework which is composed of three steps: (1) query feature extraction (2) dataset feature extraction (3) similar texture search. In Fig. 8 we illustrate the proposed framework.

Query Feature extraction. We first apply the DFT to the query image to obtain the DFT magnitude spectrum, and then we perform the pixel-wise weighted combination defined in Eq. 8 to the query and its DFT magnitude spectrum to obtain the DFT based blended query image $B(x, y)$. We next feed the resultant image to the trained convolutional autoencoder model and extract the latent space representation also named as DFTD, which describes the features of the supplied query image.

Detection of regions of interest using an RPN and creation of a DFTD database. The identification of a particular texture in an image is the most

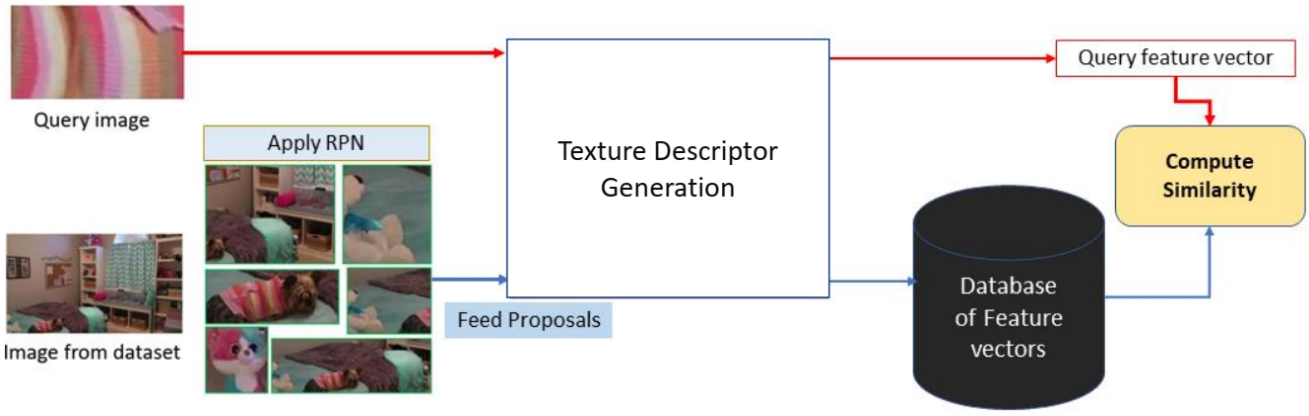

Fig. 8 Pipeline of our texture-based instance retrieval approach. On the one hand, we compute the DFTD descriptor from the query image. On the other hand, we employ an RPN to generate proposals, and then we compute the DFTD descriptors from the proposals which are stored in a database. Finally, we compare the query and database descriptors using the similarity metric.

crucial step in texture-based instance retrieval tasks. In particular, the queried texture needs to be compared with all distinctive texture patches of the image since the query can be present in different sizes, scales and orientations. In this stage, we integrate the RPN with the convolutional encoder, so that the region proposals generated by the RPN are fed to the encoder to extract the DFTD descriptors of each of the regions. The DFTD descriptors of all proposals are stored in a database together with the label of the image they belong to. However, in the case of datasets that contain a single texture pattern in each image, we can directly compute the DFTDs from the whole images without the need for an RPN.

Similar texture search. To retrieve images that contain a similar texture pattern as that of the query, we compute the cosine similarity (CS) metric, Eq. 9, of all pairs of query and database DFTD descriptors. The cosine metric indicates the similarity between two vectors, the higher it is, the more similar are the considered descriptors. We first compare the query descriptor with all the proposal descriptors present in a particular image, and we then store the proposal with the highest similarity score in a hit list. We repeat this process for all images in the dataset. Finally, we sort the hit list in descending order according to the CS scores and retrieve the top-n instances.

$$CS = \frac{H_q m_i}{||H_q||||m_i||} = \frac{\sum_{j=1}^{N} H_{q_j} m_{i_j}}{\sqrt{\sum_{j=1}^{N} H_{q_j}^2}\sqrt{\sum_{j=1}^{N} m_{i_j}^2}}, \quad (9)$$

where N is the dimension of the texture descriptor, H_q is the DFTD of the query image and m_i is the DFTD fo the i_{th} proposal of an image.

We further summarize the full texture retrieval pipeline in Algorithm 1.

Algorithm 1: Texture retrieval using DFT based latent features

Function Texture-retrieval (q, m, k);
INPUT: query image q, encoder m and number of retrievals k;
OUTPUT: top-k similar instances;
1. Apply DFT to the query image $f(x, y)$ to get $F_m(u, v)$;
2. Obtain $B(x, y)$ from $f(x, y)$ and $F_m(u, v)$;
3. Feed $B(x, y)$ to the VGG encoder m to get the latent space representation;
4. Apply DFT to all images on the dataset;
5. Compute $B^*(x, y)$ images for all images on the dataset;
6. Extract proposals from the $B^*(x, y)$ images using an RPN;
7. Feed the proposals to the VGG encoder m to get the latent space representations;
6. Compute Cosine Similarity (CS) between all pairs of query and proposal descriptors;
7. Sort images in a list L in descending order with respect to the highest CS score in relation to the query vector;
if *lenght-of-list(L)=k* **then**
  return *top-k* results;
else
  No sufficient images in the dataset for comparison
end

3.3 Texture classification

We also present a texture classification approach based on transfer learning, in which we use the same VGG-

Net architecture pre-trained with the ImageNet dataset as in Section 3.2. In addition, we added two dense layers after the convolutional layers of the VGG-16 network, and the end of the classification layer uses the softmax activation. Similarly to the retrieval framework, we train the network with $B(x, y)$ images using the same hyper-parameters. For testing, we extract the class labels based on the activation obtained from the classification layer of the network, and we compare the class labels with the ground truth labels. Unlike the retrieval framework, in this classification approach we have a dense layer instead of the latent space layer, and for convenience, we will also represent the dense features as DFTD.

4 Datasets

Our proposed method is evaluated based on the following datasets.

Coloured Brodatz texture (CBT) dataset: Coloured Brodatz Texture (CBT) dataset [70] is an extension of the Brodatz texture dataset. It contains texture images which possess a wide variety of colour content. Further, it consists of 112 textures of size 640×640 pixels, where each one is divided into 25 non-overlapping images of size 128×128 pixels. As a consequence, the final dataset consists of 2800 images in total with 25 images per class. In our work, we use this dataset only to train our network with coloured texture images, so that the network can well distinguish between similar textured patterns with different colours.

Outex dataset: The Outex TC-00013 dataset [71] is a collection of 1360 images representing heterogeneous materials such as paper, fabric, wool and stones. It comprises 68 texture classes and each one includes 20 image samples of 128×128 pixels. Out of which, 10 images are for training and the other 10 are for testing in each class.

USPtex dataset: The USPtex dataset [72] consists of 2292 images with 191 classes of both natural scenes (road, vegetation and cloud) and materials (seeds, rice and tissues). Each class consists of 12 image samples of 128×128 pixels, where six images are for testing and the rest for training.

Stex dataset: The Salzburg Texture Image Database (STex) [73] consists of 476 colour texture images which are similar to the ones present in Outex and USPtex datasets. For testing, the images are divided into 16 non-overlapping tiles of size 128×128 pixels. As a result, the final dataset consists of 7616 images with 16 images per class.

TextileTube dataset: This dataset [36] is composed of 684 images of sizes that range between 480×360 and 1280×720 pixels obtained from 15 videos of YouTube. The videos were recorded in bedrooms with different cameras. The dataset contains 67 classes of textiles such as curtains, carpets, sofas, shirts or dresses, among others. The ground truth of this dataset comprises the class labels and the bounding boxes of the texture regions. This dataset creates a similar context for texture-based image retrieval as the one that usually appears in child sexual exploitation videos recorded in indoor environments, typically bedrooms.

To compare our methods with the state-of-the-art reports, we first use the original queries as considered in [36], in which all ground truth textile regions were considered as query images. However, these queries contain several objects parts and hence present shape information. To make the queries completely texture-based, we cropped the images so that only the texture part is visible, and we named this set of queries as *New queries*. The number of queries remains the same, the task is more challenging since there is not shape information. We have made the New queries available to the research community[4]. In our work, we provide the retrieval results using both types of queries, the original and the New ones. In Fig. 9, we present some original and New queries samples from the TextileTube dataset. Since training images are not provided in the TextileTube dataset, we randomly selected 25 classes, and randomly chose one image of each class to be in the training set. Later, we applied data augmentation to generate multiple images. Besides these few training samples, we considered all other images in the dataset to test the retrieval framework.

In Fig. 10, we show some sample images from the CBT, Outex and USPtex datasets. These datasets contain only a single texture pattern per image. Fig. 11 represents a sample from the TextileTube dataset, which contains multiple textures in a single image.

5 Experimental setup and results

The experiments and results presented in this work aim to verify the assessment of the DFTD descriptors derived from the convolutional encoder trained with DFT based blended images. We carried out two different types of experiments with two types of datasets. The

[4] http://gvis.unileon.es/dataset/textiltube/

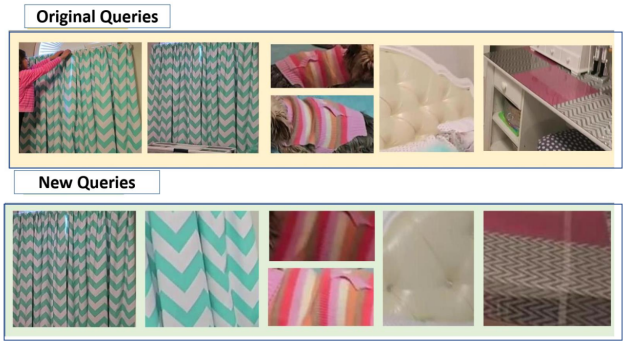

Fig. 9 Original queries: the queries as in [36] to compare with state-of-the-art methods. **New Queries**: images with only the texture portion, which were cropped out from the original queries.

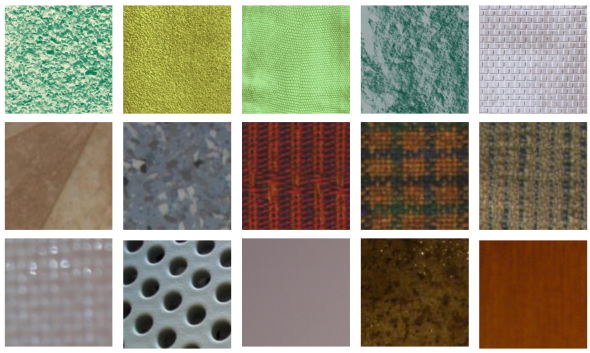

Fig. 10 From top to bottom row: samples of CBT, Outex and USPtex datasets, respectively

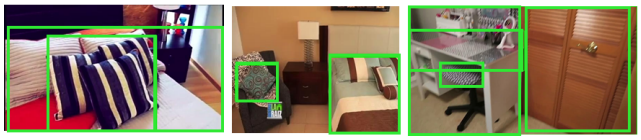

Fig. 11 Sample images from the TextileTube dataset. Overlaid green boxes represent the bounding boxes of the groundtruth.

first type of experiments was performed using Outex, USPtex and Stex datasets in which images are comprised of only a single textured pattern. Whereas in the second kind of experiment, we consider TextileTube dataset consisting of multiple objects with different textures for texture-based image retrieval. The second experiment represents a real-world scenario, where the images in the TextileTube dataset are taken from indoor scenes, and the queried texture pattern could be present anywhere in those images in different conditions of shape, scale, lighting, orientation, etc. We verified our descriptor using Outex, USPtex and Stex datasets for texture retrieval and classification in order to be able to compare the performance with the state-of-the-art works in the bulk of the literature. TextileTube comprises a more challenging dataset which is quite specific and, at the same time, very useful for the application of evidence search in Child Sexual Abuse crime cases. Nonetheless, it can as well represent other applications like clothing search for textile marketing.

5.1 Experimental setup

5.1.1 Training data preparation

To train our network architecture, we prepared training images from TextileTube, CBT, Outex and USPtex datasets. The images from TextileTube consist of texture images extracted from real-world images, whereas the images from the Outex, USPtex and CBT are well defined textured patterns with various colours. Thus, the network can learn texture features along with colours. To prepare the training set, we considered the training samples provided in the Outex and USPtex datasets, and the complete CBT dataset. In addition, from the TextileTube dataset, we randomly selected 30 images taken from different classes. Besides, since there are few training examples, we created a larger set of training images by augmenting them via several random transformations such as Gaussian distortion, rotation, skewing, tilting and flipping. As a result, we generated a larger training set consisting of $60,000$ images.

5.1.2 Retrieval framework setup

We trained the retrieval network using DFT based blended $B(x, y)$ images by applying transfer learning to the encoder. During training, we froze the first five convolutional layers of the ImageNet initialised VGG encoder, and the decoder learns to map the latent space representation back to the input training images. We chose Adam optimizer with an initial learning rate of 0.0001, and a batch size of 16 images so that the full GPU memory could be utilized. Furthermore, we stopped our training when we observed that the images generated at the end of the decoder were visually similar to the input images. This means that the network has learnt the latent space representation of the input image in order to reconstruct the original input through the decoder. However, for Outex, USPtex and Stex datasets, the RPN was not applied since all the images in these three datasets are composed of single textured patterns.

5.1.3 Classification framework setup

Likewise the training of the retrieval network, we used DFT based blended $B(x, y)$ images to train the dense layers of the classification model. We trained Outex dataset with the augmented training data and evaluated the performance on the test data. For USPtex dataset, we carried out the same procedure. We measured the performance of the proposed DFTD texture descriptors on Outex and USPtex datasets for texture classification, where We evaluated the performance in terms of accuracy. Here, we define accuracy as the percentage of test images classified as true positive.

5.1.4 Implementation

All the experiments were carried out in an Nvidia GeForce GTX 1060 GPU using TensorFlow framework. For texture retrieval in TextileTube dataset, we used the RPN of the R-FCN network to generate proposals from the images where we want to search a queried texture pattern. However, for retrieval in Outex, USPtex and Stex datasets, RPN was not necessary again because all images present only a texture pattern.

5.1.5 Evaluation metrics

To compare our approach with other state-of-the-art methods, we used two different evaluation protocols. The evaluation protocol used by the Outex, Stex and USPtex is Average retrieval rate (ARR) which was suggested in [39]. However, the state-of-the-art results concerning the TextileTube dataset are provided based on *precision*@k, as proposed in [36].

Evaluation metric for Outex, USPtex and Stex: We evaluated the performance for texture-based instance retrieval using average retrieval rate (ARR). Let N be the total number of images in the dataset and R_q be the number of relevant images for a query q. Let $m_{(q,k)}$ be the number of correctly retrieved images found within the first k retrievals for a query q. ARR in terms of the number of retrieved images is given by:

$$ARR = \frac{1}{N}\sum_{q=1}^{N}\frac{m_{(q,k)}}{R_q}. \tag{10}$$

Evaluation metric for TextileTube dataset We evaluated the retrieved top k texture images concerning a given query image according to the precision in a ranking-based criterion. The creation of the hit list was defined in Section 3.2. The precision@k is defined as:

$$Precision@k = \frac{\sum_{i=1}^{k} R(i)}{k}, \tag{11}$$

Fig. 12 The green square shows the groundtruth, and the red one presents the detected region in relation to the query image. Since there is an overlap between both regions, $R(i)$ would be set to 1.

where $R(i)$ denotes the relevance between the i^{th} ranked image in the hit list and the query. If the bounding box of the detected texture region in the retrieved image intersects the ground truth, R is set to 1; else to 0. Fig. 12 illustrates this scenario. In our experiments, we consider $k = \{1, ..., 40\}|k \in \mathbb{N}$.

5.2 Experimentation and results

In this section, we present the experiments carried out and the results obtained in comparison with the state-of-the-art approaches.

5.2.1 Experiments on Outex, USPtex and Stex datasets

Results and comparison for texture classification: We tested our approach for texture classification to evaluate the performance of the proposed DFTD descriptors against state-of-the-art works since texture classification is wider explored than texture retrieval. We followed the method and experimental set-up explained above. In Table 1, we present the results, the best accuracy reported in the literature was 89.62% by Chesspattern method [41] on Outex and 93.83% by Tuncer *et al.* [40] on USPtex. In contrast, we have achieved an accuracy of 90.20% and 95.62% on Outex and USPtex, respectively, outperforming all the reported results.

Results and comparison for texture-based instance retrieval: In Table 2, we show the performance of our approach for texture-based instance retrieval regarding ARR metric on Outex, USPtex and Stex datasets, and compare it to other state-of-the-art methods. We observed that descriptors based on learned CNN representation, i.e. CNN-VGG19 [75], yielded competitive results as compared to handcrafted features, such as DDBTC [74]. It is also noticeable that our proposed approach outperformed all the methods by obtaining

Table 1 Comparison of our proposed approach in terms of accuracy (in percentage) on Outex and USPtex datasets for texture classification.

Method	Outex	USPtex
LESTP [74]	78.00	82.41
LECTP [74]	79.06	83.10
PCANet [74]	76.04	83.65
LQP [74]	81.49	87.83
Multifractals [75]	75.07	68.76
Fourier [75]	82.21	71.16
ARCS-LBPt [75]	85.70	88.85
Chess-Pattern [41]	88.9	-
Tuncer *et al.* [40]	89.62	93.83
DFTD (Ours)	**90.20**	**95.62**

Table 2 Comparison of our proposed approach in terms of average retrieval rate (in percentage) on Outex, USPtex and Stex datasets for texture-based instance retrieval.

Method	Outex	USPtex	Stex
DDBTC [74]	66.82	74.97	44.79
CNN-AlexNet [75]	69.87	83.57	68.84
CNN-VGG16 [75]	72.91	85.03	74.92
CNN-VGG19 [75]	73.20	84.22	73.93
LED [39]	75.14	87.50	76.71
SLED [39]	75.96	88.60	77.88
MS-SLED [39]	76.15	89.74	79.87
DFTD (Ours)	**80.36**	**90.25**	**81.02**

an ARR of 80.36% on Outex, 90.25% on USPtex and 81.02% on Stex datasets, whereas the best reported result in the literature was yielded by MS-LED method [39] with an ARR of 76.15%, 89.74% and 79.87%.

5.2.2 Experiments on TextileTube dataset

Results and comparison with state-of-the-art methods: We used both types of queries, the original and the New ones, to evaluate our method. Fig. 13 summarizes the results obtained by our proposed method which outperforms the other approaches in terms of precision@k. RPN+DFTD indicates the results obtained by the original queries, whereas RPN+DFTD(new) represents results obtained using the New queries. The results using ALBP+HCLOSIB, ALBP, Faster R-CNN, HOG, HOG+CLOSIB and HOG+HCLOSIB are taken from [36].

Besides Faster R-CNN, the rest of the methods do not rely on modern deep learning techniques. In order to overcome this issue and establish a strong baseline, we also considered R-FCN and fully convolutional one-stage object detection (FCOS) [21] networks. Even though both Faster R-CNN and R-FCN are region-based detectors and use ResNet-101 for feature extraction, RFCN demonstrated to be 20× faster than Faster R-CNN. In contrast, FCOS can be exploited to generate multiple levels of intermediate proposals, where each level have proposals of different sizes, increasing in this way the scope of the query image search across a larger number of regions. For R-FCN, we took all 300 proposals as candidate regions to check the presence of the texture queries. Regarding FCOS, we used two approaches for the generation of proposals: (a) FCOS_a, where the proposals are obtained at the final classification layer, which results in around 100 proposals; and (b) FCOS_b, where we have considered all the raw proposals from the intermediate layers up to the final layer, which results in more than 2000 proposals. The best result for these state-of-the-art methods was yielded by FCOS_b because a query image descriptor can be compared against more local descriptors, but it comes at the cost of a high computational time. To evaluate our proposed descriptors and retrieval framework, we used the RPN of R-FCN to speed up the generation of texture patches.

Figure 13 illustrates the precision@k of the methods proposed in [36] (ALBP+HCLOSIB, ALBP, Faster R-CNN, HOG, HOG+CLOSIB and HOG+HCLOSIB), the considered baseline methods (R-FCN, FCOS_a and FCOS_b), and the proposed method using the original queries (RPN+SFD) and the New queries (RPN+SFD(new)). The baseline methods and our proposed method outperformed the results presented in [36]. FCOS_b yielded higher precision@k than FCOS_a and R-FCN. It can be seen that for a similar number of proposals, R-FCN outperformed FCOS_a. Our proposed method using the original queries (RPN+DFTD) achieved higher precision@k than the methods proposed in [36] and the considered baseline methods, which also utilize the original queries, for all values of $k = \{1, ..., 40\} | k \in \mathbb{N}$. Specially, the proposed method shows relevant improvement with respect to the rest of the methods for larger values of k. The results for the retrieval of the New queries (RPN+DFTD(new)) and the original queries (RPN+DFTD) using the proposed method are quite similar. We will later comment about them concerning the numeric results.

In Fig. 14, we show the top-5 retrievals with respect to two sample query images using RPN+DFTD (new) on TextileTube dataset. In each of the rows, the images framed with red bounding boxes are the query images, and the images in the left of the queries are the top-5 retrieved ones. It can be seen that the query images represent two different texture patches, which we compared with the proposals of the database images to retrieve the ones where the query patch might be present. If there is any proposal that is likely to match with the query, then it would have a high CS

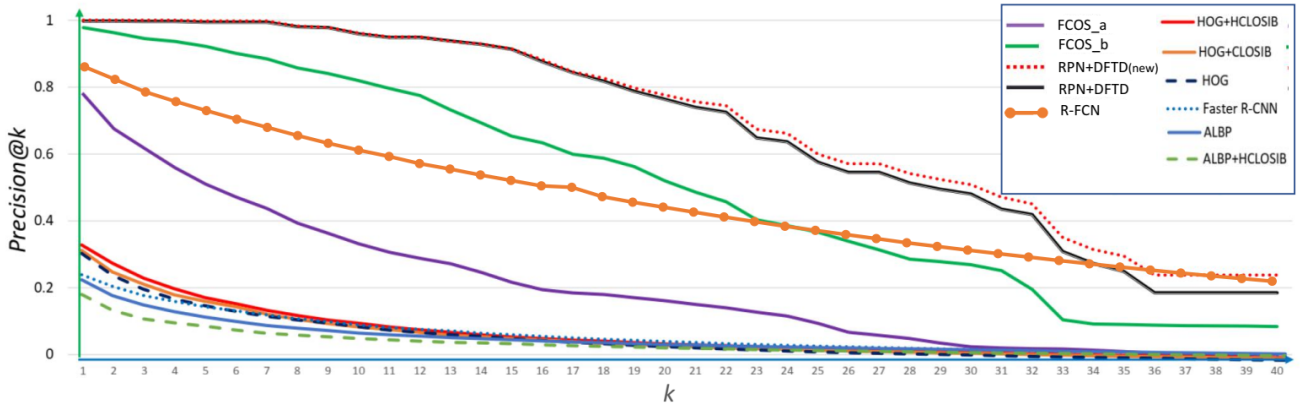

Fig. 13 Precision@k of state-of-the-art methods, recent baselines and our method (RPN+DFTD) on TextileTube dataset. RPN+DFTD represents the results with the original queries and RPN+DFTD (New) with the New queries.

score, and we retrieve the image to which that proposal corresponds. Also, it can be seen that the queried texture patterns are successfully detected in the retrieved database images, which are sorted in descending order corresponding to CS scores of the detected proposals. Moreover, the overlaid yellow boxes illustrate the detected proposals, and we can observe that the query images successfully match with them.

Table. 3 illustrates precision@k for top-k (1 to 10) retrievals. Using our method RPN+DFTD with the original queries, we obtained a precision@k of 99.7 at $k = 1$ in comparison to the highest precision of 37.5 in the literature with HOG+HCLOSIB. Furthermore, using our baseline FCOS_a and FCOS_b, we achieved a precision of 77.8 and 97.7 for $k = 1$, respectively. The results obtained by our method indicate that, for every query search, the first retrieval is a hit with a probability of 99.7%. As k increases, the proposed method achieved a higher precision with respect to the rest of the methods. For example, for $k = 10$, the proposed method yielded a 97.7% of precision, whereas FCOS_b obtained 81.9%, and HOG+HCLOSIB 18.6%, which means that our method achieved an improvement of 19.29% and 425.27% in relation to FCOS_b and HOG+HCLOSIB, respectively. For the retrieval of the New queries, the proposed method obtained 100% precision for the top-4 retrievals. For higher values of k, the precision starts slightly decreasing concerning the original queries. For large values of k, the absence of contour and surrounding information in the New queries, which contain only texture patterns, possibly provokes the fall in precision.

In Table 4, we present the arithmetic mean of precision@k for three different intervals, where k ranges from 1 to 10, 1 to 20 and 1 to 30. We achieved an arithmetic mean of 99.2%, 93.2% and 67.9% using RPN+DFTD for the three intervals, respectively, followed by FCOS_b at 90.4%, 78.0% and 50.8%, respectively. We can notice that the improvement is significantly high as compared to the reported results in [36] of 24.8%, 19.5% and 15.1% with HOG+HCLOSIB. For the new queries, the results were higher in the three intervals with arithmetic mean precision of 99.5%, 94.5% and 69.8%, respectively.

All the results clearly show that, in terms of precision, our proposed method RPN+DFTD outperformed the retrieval results obtained using HOG+HCLOSIB, from 37.2% to 99.7% for $k = 1$, and from 18.6% to 97.7% for $k = 10$. Such results prove that the activation generated by the latent space node of the VGG encoder using Fourier based images can efficiently represent the texture of a patch. Moreover, since our architecture is based on an autoencoder, we were able to train a large number of texture classes by keeping the latent space node compact with a constant dimension of 512. Furthermore, the proposals generated by the RPN cover well the regions of interest of the images on the dataset, and thus, it enables the localization of the texture query patches.

6 Conclusion

We presented a new deep Fourier texture descriptor DFTD based on the discrete Fourier transform and the latent space representation of a VGG autoencoder. Besides, we also proposed a framework for texture-based instance retrieval. We used an RPN to identify regions of interest in the image dataset which were given as an input to a VGG autoencoder. The VGG autoencoder was trained with images obtained from a weighted linear combination of DFT magnitude spectrum and spatial images. The RPN proved to be very useful to iden-

Fig. 14 Top-5 correctly retrieved images on TextileTube dataset using RPN+DFTD(New). The queries are shown on the first column and are framed with a red box, and the proposals that lead to the retrievals are framed with a yellow box.

Table 3 Precision@k, in percentage, for texture descriptors reported in [36], the baseline and the proposed method on TextileTube dataset. Results highlighted in bold mark the results obtained using RPN+DFTD(new).

Descriptor	1	2	3	4	5	6	7	8	9	10
HOG+HCLOSIB [36]	37.2	32.8	29.3	26.8	24.7	23.2	21.7	20.5	19.4	18.6
HOG+CLOSIB [36]	35.9	30.8	27.9	25.4	23.8	22.5	20.8	19.5	18.5	17.7
HOG [36]	35.2	30.0	26.7	24.4	22.8	21.5	20.2	19.4	18.7	17.8
Faster R-CNN [36]	30.1	27.4	25.2	23.8	22.5	21.6	20.6	19.7	19.0	18.2
ALBP+HCLOSIB [36]	28.9	25.2	23.0	21.4	20.1	19.0	18.1	17.5	16.9	16.3
ALBP+CLOSIB [36]	25.5	21.7	19.6	18.7	17.9	17.0	16.2	15.8	15.4	15.0
R-FCN	82.1	79.0	75.9	73.5	71.3	69.2	67.2	65.18	63.3	61.6
$FCOS_a$	77.8	67.5	61.7	55.8	50.9	47.0	43.7	39.3	36.2	33.2
$FCOS_b$	97.7	96.2	94.5	93.6	92.2	90.0	88.4	85.7	84.0	81.9
RPN+DFTD	99.7	99.7	99.6	99.6	99.4	99.4	99.4	99.4	98.0	97.7
RPN+DFTD(new)	**100.0**	**100.0**	**100.0**	**100.0**	**99.7**	**99.7**	**99.7**	**98.2**	**97.8**	**96.1**

Table 4 Arithmetic mean of precision at k (M) for intervals ranging from 1 to 10, 1 to 20 and 1 to 40 on TextileTube dataset. The results with RPN+DFTD, both the original and New queries, are shown in bold.

Descriptor	M(1-10)	M(1-20)	M(1-40)
HOG+HCLOSIB	24.8	19.5	15.1
HOG+CLOSIB	23.7	18.8	14.7
HOG	23.1	18.5	14.3
Faster R-CNN	22.5	18.8	15.3
ALBP+HCLOSIB	18.1	15.7	13.5
ALBP+CLOSIB	19.6	17.0	14.6
R-FCN	70.8	62.7	49.6
$FCOS_a$	51.3	36.7	20.7
$FCOS_b$	90.4	78.0	50.8
RPN+DFTD	**99.2**	**93.2**	**67.9**
RPN+DFTD(new)	**99.5**	**94.5**	**69.8**

tify texture regions in images with several texture patterns, such as indoor scene images.

In this work, we considered two different types of datasets to test our approach. On the one hand, we experimented using Outex, Stex and USPtex, which are similar datasets containing images with only one texture pattern per image but they are broadly used for texture retrieval and classification. And, on the other hand, we considered TextileTube dataset which comprises indoor scene images with lots of different texture patterns. We selected this dataset because it represents a useful case scenario related to Child Sexual Abuse (CSA) digital content, in which apart from faces, objects, etc., textures also may represent a clue to find pieces of evidence among already known cases of CSA.

To evaluate the performance of the proposed DFTD descriptor, we also performed experiments for texture classification since it is a problem broadly studied among computer vision researchers. We assessed the performance on Outex and USPtex datasets and compared our proposal with the recent top methods for texture classification.

DFTD is a quite compact descriptor in the form of a 512-dimensional vector, which makes the matching computationally inexpensive. Furthermore, the obtained results on the four datasets demonstrate that the proposed architecture and DFTD descriptor are effective for retrieval and classification purposes, yielding state-of-the-art performance.

As a part of future research, we will modify the proposed Fourier based descriptor to enhance rotation invariance by taking into account the Fourier coefficients. Also, we will explore other real-world datasets to evaluate and improve the retrieval framework.

Acknowledgements This work has been supported by the grant Junta de Castilla y Leon (EDU/529/2017) and the framework agreement between the University of Leon and INCIBE (Spanish National Cybersecurity Institute) under Addendum 01. We gratefully acknowledge the support of Nvidia Corporation for their kind donation of GPUs (GeForce GTX Titan X and K-40).

Conflict of interest

The authors declare that they have no conflict of interest.

References

1. Saritha RR, Paul V, Kumar PG (2019) Content based image retrieval using deep learning process. Cluster Computing 22(2):4187–4200
2. Tzelepi M, Tefas A (2018) Deep convolutional learning for content based image retrieval. Neurocomputing 275:2467–2478
3. Ahmed KT, Ummesafi S, Iqbal A (2019) Content based image retrieval using image features information fusion. Information Fusion 51:76–99
4. Alzu'bi A, Abuarqoub A (2020) Deep learning model with low-dimensional random projection for large-scale image search. Engineering Science and Technology, an International Journal
5. Forcen JI, Pagola M, Barrenechea E, Bustince H (2020) Co-occurrence of deep convolutional features for image search. Image and Vision Computing p 103909
6. Keisler R, Skillman SW, Gonnabathula S, Poehnelt J, Rudelis X, Warren MS (2019) Visual search over billions of aerial and satellite images. Computer Vision and Image Understanding 187:102790
7. Saikia S, Fidalgo E, Alegre E, Fernández-Robles L (2017) Object detection for crime scene evidence analysis using deep learning. In: International Conference on Image Analysis and Processing, Springer, pp 14–24
8. Karie NM, Kebande VR, Venter H (2019) Diverging deep learning cognitive computing techniques into cyber forensics. Forensic Science International: Synergy 1:61–67
9. Mohammad RMA, Alqahtani M (2019) A comparison of machine learning techniques for file system forensics analysis. Journal of Information Security and Applications 46:53–61
10. Liu W, Wu CY (2019) Crime scene investigation image retrieval using a hierarchical approach and rank fusion. In: 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), IEEE, pp 1974–1978
11. Liu Y, Hu D, Fan J, Wang F, Zhang D (2017) Multi-feature fusion for crime scene investigation image retrieval. In: 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), IEEE, pp 1–7
12. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
13. He K, Gkioxari G, Dollár P, Girshick R (2017) Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 2961–2969
14. Nanni L, Brahnam S, Lumini A (2010) A local approach based on a local binary patterns variant texture descriptor for classifying pain states. Expert Systems with Applications 37(12):7888–7894
15. Nanni L, Lumini A, Brahnam S (2010) Local binary patterns variants as texture descriptors for medical image analysis. Artificial intelligence in medicine 49(2):117–125
16. Srinivasan G, Shobha G (2008) Statistical texture analysis. In: Proceedings of world academy of science, engineering and technology, vol 36, pp 1264–1269
17. Van de Wouwer G, Scheunders P, Van Dyck D (1999) Statistical texture characterization from discrete wavelet representations. IEEE transactions on image processing 8(4):592–598
18. Wu Q, Wang J, Yang C, Cui G, Yang W (2016) Target recognition by texture segmentation algorithm. Expert Systems with Applications 46:394–404
19. Zheng L, Yang Y, Tian Q (2017) Sift meets cnn: A decade survey of instance retrieval. IEEE transactions on pattern analysis and machine intelligence 40(5):1224–1244
20. Singh B, Li H, Sharma A, Davis LS (2018) R-fcn-3000 at 30fps: Decoupling detection and classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 1081–1090
21. Tian Z, Shen C, Chen H, He T (2019) Fcos: Fully convolutional one-stage object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp 9627–9636
22. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2020) Deep learning for generic object detection: A survey. International Journal of Computer Vision 128(2):261–318
23. Wu X, Sahoo D, Hoi SC (2020) Recent advances in deep learning for object detection. Neurocomputing

24. Ma W, Wu Y, Cen F, Wang G (2020) Mdfn: Multi-scale deep feature learning network for object detection. Pattern Recognition 100:107149
25. Babenko A, Slesarev A, Chigorin A, Lempitsky V (2014) Neural codes for image retrieval. In: European conference on computer vision, Springer, pp 584–599
26. Ng WW, Li J, Tian X, Wang H, Kwong S, Wallace J (2020) Multi-level supervised hashing with deep features for efficient image retrieval. Neurocomputing
27. Wu Y, Wang S, Huang Q (2019) Multi-modal semantic autoencoder for cross-modal retrieval. Neurocomputing 331:165–175
28. Daoud MI, Saleh A, Hababeh I, Alazrai R (2019) Content-based image retrieval for breast ultrasound images using convolutional autoencoders: A feasibility study. In: 2019 3rd International Conference on Bio-engineering for Smart Technologies (BioSMART), IEEE, pp 1–4
29. Xu G, Fang W (2016) Shape retrieval using deep autoencoder learning representation. In: 2016 13th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP), IEEE, pp 227–230
30. Ying G, Wei Q, Shen X, Han S (2008) A two-step phase-retrieval method in fourier-transform ghost imaging. Optics communications 281(20):5130–5132
31. Sokic E, Konjicija S (2016) Phase preserving fourier descriptor for shape-based image retrieval. Signal Processing: Image Communication 40:82–96
32. Tsai DM, Tseng CF (1999) Surface roughness classification for castings. Pattern recognition 32(3):389–405
33. Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition, Ieee, pp 248–255
34. Dai J, Li Y, He K, Sun J (2016) R-fcn: Object detection via region-based fully convolutional networks. In: Advances in neural information processing systems, pp 379–387
35. Alzu'bi A, Amira A, Ramzan N (2015) Semantic content-based image retrieval: A comprehensive study. Journal of Visual Communication and Image Representation 32:20–54
36. García-Olalla O, Alegre E, Fernández-Robles L, Fidalgo E, Saikia S (2018) Textile retrieval based on image content from cdc and webcam cameras in indoor environments. Sensors 18(5):1329
37. Kwitt R, Uhl A (2008) Image similarity measurement by kullback-leibler divergences between complex wavelet subband statistics for texture retrieval. In: 2008 15th IEEE International Conference on Image Processing, IEEE, pp 933–936
38. Ouslimani F, Ouslimani A, Ameur Z (2019) Rotation-invariant features based on directional coding for texture classification. Neural Computing and Applications 31(10):6393–6400
39. Pham MT (2018) Efficient texture retrieval using multiscale local extrema descriptors and covariance embedding. In: Proceedings of the European Conference on Computer Vision (ECCV), pp 0–0
40. Tuncer T, Dogan S, Ertam F (2019) A novel neural network based image descriptor for texture classification. Physica A: Statistical Mechanics and its Applications 526:120955
41. Tuncer T, Dogan S, Ataman V (2019) A novel and accurate chess pattern for automated texture classification. Physica A: Statistical Mechanics and its Applications 536:122584
42. King I, Lau TK (1996) A feature-based image retrieval database for the fashion, textile, and clothing industry in hong kong. In: Proc. of International Symposium Multi-Technology Information Processing, vol 96, pp 233–240
43. D'Amato JP, Mercado M, Heiling A, Cifuentes V (2016) A proximal optimization method to the problem of nesting irregular pieces using parallel architectures. REVISTA IBEROAMERICANA DE AUTOMATICA E INFORMATICA INDUSTRIAL 13(2):220–227
44. Wong C (2017) Applications of computer vision in fashion and textiles. Woodhead Publishing
45. Gordo A, Almazan J, Revaud J, Larlus D (2017) End-to-end learning of deep visual representations for image retrieval. International Journal of Computer Vision 124(2):237–254
46. Dos Santos JM, De Moura ES, Da Silva AS, da Silva Torres R (2017) Color and texture applied to a signature-based bag of visual words method for image retrieval. Multimedia Tools and Applications 76(15):16855–16872
47. Mezaris V, Kompatsiaris I, Strintzis MG (2003) An ontology approach to object-based image retrieval. In: Proceedings 2003 International Conference on Image Processing (Cat. No. 03CH37429), IEEE, vol 2, pp II–511
48. Ren J, Shen Y, Guo L (2003) A novel image retrieval based on representative colors. In: Proceedings of the Image and Vision Computing, NZ, Citeseer
49. Song J, Gao L, Liu L, Zhu X, Sebe N (2018) Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern

Recognition 75:175–187
50. Maillot N, Thonnat M, Boucher A (2004) Towards ontology-based cognitive vision. Machine Vision and Applications 16(1):33–40
51. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. International journal of computer vision 60(2):91–110
52. Bay H, Ess A, Tuytelaars T, Van Gool L (2008) Speeded-up robust features (surf). Computer vision and image understanding 110(3):346–359
53. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05), IEEE, vol 1, pp 886–893
54. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence 24(7):971–987
55. Verma M, Raman B (2018) Local neighborhood difference pattern: A new feature descriptor for natural and texture image retrieval. Multimedia Tools and Applications 77(10):11843–11866
56. Nan B, Xu Y, Mu Z, Chen L (2015) Content-based image retrieval using local texture-based color histogram. In: 2015 IEEE 2nd International Conference on Cybernetics (CYBCONF), IEEE, pp 399–405
57. Singh C, Walia E, Kaur KP (2018) Color texture description with novel local binary patterns for effective image retrieval. Pattern recognition 76:50–68
58. Pavithra L, Sharmila TS (2018) An efficient framework for image retrieval using color, texture and edge features. Computers & Electrical Engineering 70:580–593
59. Yang C, Yu Q (2019) Multiscale fourier descriptor based on triangular features for shape retrieval. Signal Processing: Image Communication 71:110–119
60. Liu W, Wang Z, Liu X, Zeng N, Liu Y, Alsaadi FE (2017) A survey of deep neural network architectures and their applications. Neurocomputing 234:11–26
61. Popa CA, Cernăzanu-Glăvan C (2018) Fourier transform-based image classification using complex-valued convolutional neural networks. In: International Symposium on Neural Networks, Springer, pp 300–309
62. Chitsaz K, Hajabdollahi M, Karimi N, Samavi S, Shirani S (2020) Acceleration of convolutional neural network using fft-based split convolutions. arXiv preprint arXiv:200312621
63. Saikia S, Fidalgo E, Alegre E, Fernández-Robles L (2017) Query based object retrieval using neural codes. In: International Joint Conference SOCO'17-CISIS'17-ICEUTE'17 León, Spain, September 6–8, 2017, Proceeding, Springer, pp 513–523
64. Salvador A, Giró-i Nieto X, Marqués F, Satoh S (2016) Faster r-cnn features for instance search. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pp 9–16
65. Cimpoi M, Maji S, Kokkinos I, Mohamed S, Vedaldi A (2014) Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 3606–3613
66. Cimpoi M, Maji S, Vedaldi A (2015) Deep filter banks for texture recognition and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3828–3836
67. Cimpoi M, Maji S, Kokkinos I, Vedaldi A (2016) Deep filter banks for texture recognition, description, and segmentation. International Journal of Computer Vision 118(1):65–94
68. Yikun Y, Shengjie J, Jinrong H, Bisheng X, Jiabo L, Ru X (2020) Image retrieval via learning content-based deep quality model towards big data. Future Generation Computer Systems
69. Redmon J, Farhadi A (2018) Yolov3: An incremental improvement. arXiv preprint arXiv:180402767
70. Abdelmounaime S, Dong-Chen H (2013) New brodatz-based image databases for grayscale color and multiband texture analysis. ISRN Machine Vision 2013
71. Ojala T, Pietikainen M, Maenpaa T (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on pattern analysis and machine intelligence 24(7):971–987
72. Casanova D, Florindo JB, Falvo M, Bruno OM (2016) Texture analysis using fractal descriptors estimated by the mutual interference of color channels. Information Sciences 346:58–72
73. Kwitt R, Meerwald P (2018) Salzburg texture image database (stex)
74. Guo JM, Prasetyo H, Wang NJ (2015) Effective image retrieval system using dot-diffused block truncation coding features. IEEE Transactions on Multimedia 17(9):1576–1590
75. Napoletano P (2017) Hand-crafted vs learned descriptors for color texture classification. In: International Workshop on Computational Color Imaging, Springer, pp 259–271