Detecting Anomalous Structures by Convolutional Sparse Models

Diego Carrera, Giacomo Boracchi
Dipartimento di Elettronica, Informazione e Bioingegeria
Politecnico di Milano, Italy
{giacomo.boracchi, diego.carrera}@polimi.it

Alessandro Foi
Department of Signal Processing
Tampere University of Technology
Tampere, Finland
alessandro.foi@tut.fi

Brendt Wohlberg
Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM, USA
brendt@lanl.gov

***Abstract*—We address the problem of detecting anomalies in images, specifically that of detecting regions characterized by structures that do not conform those of normal images. In the proposed approach we exploit convolutional sparse models to learn a dictionary of filters from a training set of normal images. These filters capture the structure of normal images and are leveraged to quantitatively assess whether regions of a test image are normal or anomalous. Each test image is at first encoded with respect to the learned dictionary, yielding sparse coefficient maps, and then analyzed by computing indicator vectors that assess the conformance of local image regions with the learned filters. Anomalies are then detected by identifying outliers in these indicators.**

Our experiments demonstrate that a convolutional sparse model provides better anomaly-detection performance than an equivalent method based on standard patch-based sparsity. Most importantly, our results highlight that monitoring the local group sparsity, namely the spread of nonzero coefficients across different maps, is essential for detecting anomalous regions.

***Keywords*—Anomaly Detection, Convolutional Sparse Models, Deconvolutional Networks.**

I. Introduction

We address the problem of detecting *anomalous* regions in images, i.e. regions having a structure that does not conform to a reference set of *normal* images [1]. Often, anomalous structures indicate a change or an evolution of the data-generating process that has to be promptly detected to react accordingly. Consider, for instance, an industrial scenario where the production of fibers is monitored by a scanning electron microscope (SEM). In normal conditions, namely when the machinery operates properly, images should depict filaments and structures similar to those in Figure 1(a). Anomalous structures, such as those highlighted in Figure 1(b), might indicate a malfunction or defects in the raw materials used, and have to be automatically detected to activate suitable countermeasures.

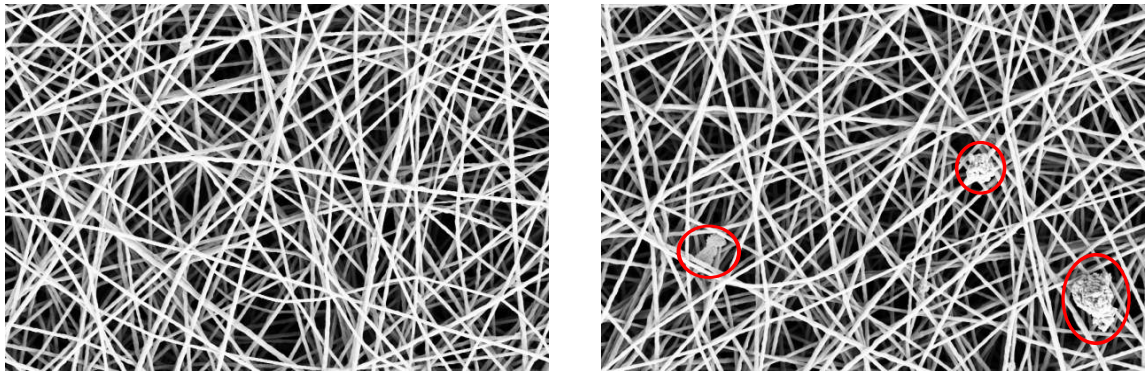
Detecting anomalies in images is a challenging problem. First of all because, often, no training data for the anomalous regions are provided and it is not feasible to forecast all the possible anomalies that might appear. Second, anomalies in images might cover arbitrarily shaped regions, which can be very small. Third, anomalies might affect only the local structures, while leaving macroscopic features such as the average pixel-intensity in the region untouched.

Our approach is based on convolutional sparse models, which in [2] were shown to effectively learn mid-level features of images. In convolutional sparse representations, the input image is approximated as the sum of M convolutions between a small filter d_m and a sparse coefficient map x_m, i.e. a spatial map having few non-zero coefficients. A convolutional sparse model is a synthesis representation [3], where the image is encoded with respect to a dictionary of filters, yielding sparse coefficient maps. The decoding consists in adding all the outputs of the convolution between the filters and corresponding coefficient maps.

Structures from normal images are modeled by learning a dictionary of M filters $\{d_m\}$ from a training set of normal images. Learned filters represent the local structure of training images, as shown in Figure 2(a). Each test image is encoded with respect to the learned filters, computing the coefficient maps that indicate which filters are activated (i.e. have nonzero coefficients) in each local region of the image. In normal regions we expect the convolutional sparse model to describe the image well, yielding sparse coefficient maps and a good approximation. This is illustrated in Figure 2(b), where the green patch in the left half belongs to a normal region and has only few filters activated: the corresponding coefficient maps are sparse. In contrast, in regions characterized by anomalous structures, we expect coefficient maps to be less sparse or to less accurately approximate the image. The red patch inside the right (anomalous) half of Figure 2(b) shows coefficient maps that are not sparse.

We detect anomalies by analyzing a test image and the corresponding coefficient maps in a patch-wise manner. Patches, i.e. small regions having a predefined shape, are thus the core objects of our analysis. For each patch we compute a low-dimensional *indicator vector* that quantitatively assesses the goodness-of-fit of the convolutional model, namely the extent to which the patch is consistent with normal ones. Indicators are then used to determine whether a given patch is normal or anomalous. Given the above considerations, the most straightforward indicator for a patch would be a vector stacking the reconstruction error and the sparsity of the coefficient maps over each patch.

However, in our experiments we show that the sparsity of the coefficient maps is too loose a criterion for discriminating anomalous regions, and that it is convenient to consider also the spread of nonzero coefficients across different maps. In particular we observed that, while normal regions can be



(a) (b)

Fig. 1: Examples of SEM images depicting a nanofibrous material produced by an electrospinning process: Fig. (a) does not contain anomalies, and is characterized by specific structures also at local-level. Fig. (b) highlights anomalies that are clearly visible among the thin fibers.

typically approximated by few filters, the representation of anomalous ones often involves many filters. Therefore, we include in the indicator vector a term measuring the a local group-sparsity of the coefficient maps, and show that this is information is effective at discriminating anomalous regions.

Summarizing, the contribution of the paper is two-fold. First, we develop a novel approach for detecting anomalies by means of convolutional sparse models describing the structures of normal images. Second, we show that the local group sparsity of the coefficient maps is a relevant prior for discriminating anomalous regions.

The remainder of the paper is organized as follows: Section II provides an overview of related works, while Section III formulates the anomaly-detection problem. The proposed approach is outlined in Section IV while details about convolutional sparse representations and the indicator vectors are provided in Sections IV-A and IV-B, respectively. Experiments are presented and discussed in Section V.

II. Related works

Anomaly detection [4], refers to the general problem of detecting unexpected patterns both in supervised scenarios (where training samples are labeled as normal, or either normal and anomalous) and in unsupervised scenarios (where training samples are provided without labels). Anomaly detection is also referred to as *novelty detection* [1], [5], [6], in particular when anomalies are intended as patterns that do not conform to a training set of normal data. In the machine learning literature, novelty detection is formulated as a one-class classification problem [7]. In this paper, we shall refer to the patterns being detected as anomalies, despite the novelty detection context. An overview of novelty-detection methods for images is reported in [1].

Convolutional sparse models were originally introduced in [2] to build architectures of multiple encoder layers, the so-called deconvolutional networks. These networks have been shown to outperform architectures relying on standard patch-based sparsity when learning mid-level features for object recognition. More sophisticated architectures involving both decoder and encoder layers showed to be effective in visual recognition tasks, such as supervised pedestrian detection [8] and unsupervised learning of object parts [9]. Deconvolutional networks are strictly related to the well-known convolutional networks [10], which in contrast are analysis representations, where the image is subject to multiple layers where it is directly convolved against filters. Convolutional sparse models have not previously been used for the anomaly-detection problem, such as that considered in this work. For simplicity, we presently develop and describe a single-layer architecture, although more layers may also be used.

Convolutional sparse models can be seen as extensions of patch-based sparse models [11], the former providing a representation of the entire image while the latter independently represent each image patch. Patch-based sparse models have been recently used for anomaly detection purposes [12], where an unconstrained optimization problem is solved to obtain the sparse representation of each patch in a test image. Then, the reconstruction error and the sparsity of the computed representation are jointly monitored to detect the anomalous structures. In [13] anomalies are detected by means of a specific sparse-coding procedure, which isolates anomalies as data that do not admit a sparse representation with respect to a learned dictionary. Sparse representations have been used for detecting unusual events in video sequences [14], [15] by monitoring the functional minimized during the sparse coding stage. The detection of structural changes – a problem closely related to anomaly detection – was addressed in [16], where sparse representations were used to sequentially monitor a stream of signals.

Convolutional sparse models offer two main advantages compared to patch-based ones: first of all, they directly support the use of multiscale dictionaries [18], whereas this is not straightforward for standard patch-based sparse representations. Second, the size of the patch that has to be analyzed in the anomaly detection can be arbitrarily increased at a negligible computational overhead when convolutional sparse models are exploited. In contrast, this requires additional training and also increases the computational burden [22] in the case of patch-based sparsity.

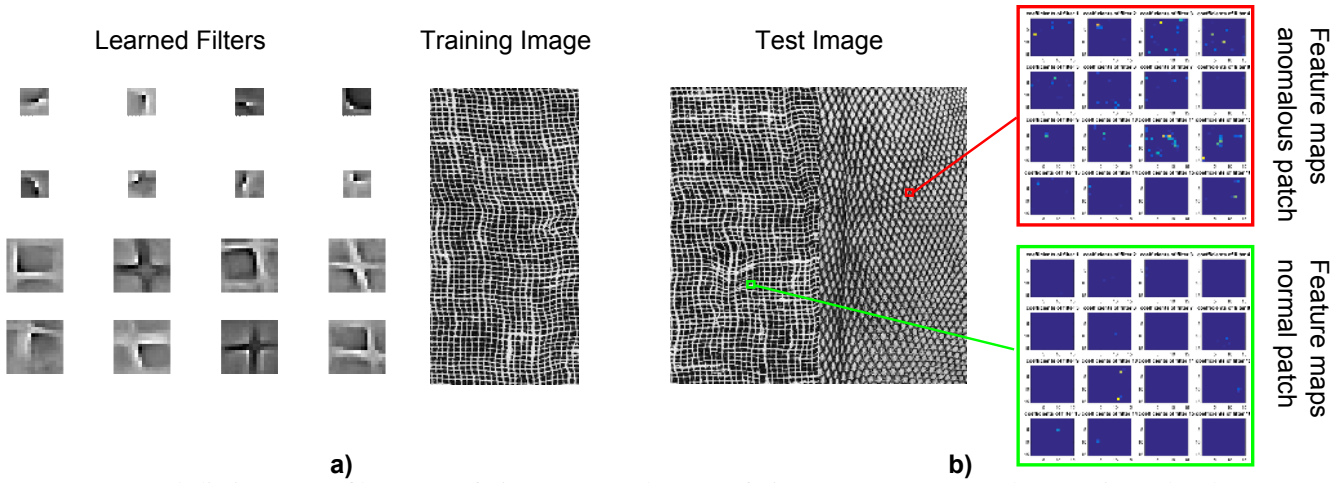


Fig. 2: (a) Learned dictionary (8 filters are of size 8×8 and 8 are of size 16×16) reports the prominent local structures of the training image. (b) A test image used in our experiments: the left half represents the normal region (filters were learned from the other half of the same texture image), while the right half represents the anomalous region. The ideal anomaly detector should mark all pixels within the right half as anomalous, and all pixels within the left half as normal. The coefficient maps corresponding to the two highlighted regions (red and green squares) have approximately the same ℓ^1 norms. However, the right-most figures show that there is a substantially different spread of nonzero elements across coefficient maps. Thus, in this example the local group sparsity of the coefficient maps is more informative than sparsity for discriminating anomalous regions. Feature maps have been rescaled for better visualization.

III. Problem Formulation

Let us denote by $s : \mathcal{X} \to \mathbb{R}^+$ a grayscale image, where $\mathcal{X} \subset \mathbb{Z}^2$ is the regular pixel grid representing the image domain having size $N_1 \times N_2$. Our goal is to detect those regions in a test image s where the structures do not conform to those of a reference training set of normal images T.

To this purpose, we analyze the image locally, and formulate the detection problem in terms of image patches. In particular, we denote

$$\mathbf{s}_c := \Pi_c s = \{s(c+u), u \in \mathcal{U}\}, \ \forall c \in \mathcal{X} \tag{1}$$

the patch centered at a specific pixel $c \in \mathcal{X}$, where $\mathcal{U}$ is a neighborhood of the origin defining the patch shape, and Π_c denotes the linear operator extracting the patch centered at pixel c. We consider $\mathcal{U}$ a square neighborhood of $\sqrt{P} \times \sqrt{P}$ pixels (indicating by P the cardinality of $\mathcal{U}$), even though patches $\mathbf{s}_c$ can be defined over arbitrary shapes.

We assume that patches in anomaly-free images are drawn from a stationary, stochastic process $\mathcal{P}_\mathcal{N}$ and we refer to these as normal patches. In contrast, anomalous patches are generated by a different process $\mathcal{P}_\mathcal{A}$, which yields unusual structures that do not conform to those generated by $\mathcal{P}_\mathcal{N}$.

The training set T is used to learn a suitable model specifically for approximating normal images. Instead, no training samples of anomalous images are provided, thus it is not possible to learn a model approximating anomalous regions. In this sense, $\mathcal{P}_\mathcal{A}$ remains completely unknown.

Anomalous structures are detected at the patch level: each patch $\mathbf{s}_c$ is tested to determine whether it *does or does not conform* to the model learned to approximate images generated by $\mathcal{P}_\mathcal{N}$. This will result in a map locating anomalous regions in a test image.

IV. Method

We treat the low and high-frequency components of images separately; we perform a *preprocessing* to express s as

$$s = s_l + s_h, \tag{2}$$

where s_l and s_h denote the low-frequency and high-frequency components of s, respectively. Typically, s_l is first computed by low-pass filtering s and then $s_h = s - s_l$.

In particular, we compute the convolutional sparse representation of s_h with respect to a dictionary of filters learned to describe the high-frequency components of normal images. Restricting the convolutional sparse representation to the high-frequency components allows the use of few small filters in the dictionary. For each patch $\mathbf{s}_c$ we compute a vector $\mathbf{g}_h(c)$ that simultaneously assesses the accuracy and the sparsity of the convolutional representation around c. The low-frequency content of s is instead monitored by computing the sample moments of $s_l(\cdot)$ over patches, yielding vectors $\mathbf{g}_l(\cdot)$. Then, for each patch, we define the indicator vector $\mathbf{g}(c)$ as the concatenation of $\mathbf{g}_h(c)$ and $\mathbf{g}_l(c)$. Anomalies are detected as patches yielding outliers in these indicator vectors.

Let us briefly summarize the proposed solution, presenting the high-level scheme in Algorithms 1 and 2; details about convolutional sparse representations and indicators are then provided in Section IV-A and IV-B, respectively.

Training: Anomalies are detected by learning a model describing normal patches from the training set T. In particular, we learn a dictionary of filters $\{d_m\}$ yielding convolutional sparse representation for high-frequency components of normal images (Algorithm 1, line 1 and Section IV-A1). The indicator vectors are then computed from all the normal patches, as described in Section IV-B, and a suitable confidence region

$\mathcal{R}_\gamma$ that encompasses most of these indicators is defined (Algorithm 1, lines 2 and 3).

Testing: During operation, each test image s is preprocessed to separate the high frequency content s_h from the low frequency content s_l (Algorithm 2, line 1). The convolutional sparse representation of s_h with respect to the dictionary $\{d_m\}$ is obtained by the sparse coding procedure described in Section IV-A2 (Algorithm 1, line 2). Then, for each pixel c, $\mathbf{g}_h(c)$ is computed by analyzing the convolutional representation of s_h in the vicinity of c, and $\mathbf{g}_l(c)$ is computed by analyzing s_l in the vicinity of c (Algorithm 1, lines 4 and 5). The indicator vector $\mathbf{g}(c)$ is then obtained by stacking $\mathbf{g}_h(c)$ and $\mathbf{g}_l(c)$, namely $\mathbf{g}(c) = [\mathbf{g}_h(c), \mathbf{g}_l(c)]'$. Any indicator $\mathbf{g}(c)$ that falls outside a confidence region $\mathcal{R}_\gamma$ estimated from normal images is considered an outlier and the corresponding patch anomalous (Section IV-B3, Algorithm 1, line 7).

Input: training set of normal images T:
1. Learn filters $\{d_m\}$ solving (4)
2. Compute $\mathbf{g}_h(\cdot)$ (9) and $\mathbf{g}_l(\cdot)$ (10) for all the normal patches, define $\mathbf{g}$ as in (11)
3. Define $\mathcal{R}_\gamma$ in (12) setting the threshold $\gamma > 0$

Algorithm 1: Training the anomaly detector using convolutional sparse models.

Input: test image s:
1. Preprocess the image $s = s_l + s_h$
2. Compute the coefficient maps $\{x_m\}$ solving (7)
3. **foreach** pixel c of $\mathbf{s}$ **do**
4. Compute $\mathbf{g}_h(c)$ as (9)
5. Compute $\mathbf{g}_l(c)$ as (10)
6. Define $\mathbf{g}(c) = [\mathbf{g}_l(c), \mathbf{g}_l(c)]'$ as (11)
7. **if** $\mathbf{g}(c) \notin \mathcal{R}_\gamma$ **then**
8. c belongs to an anomalous region
else
9. c belongs to a normal region
end
end

Algorithm 2: Detecting anomalous regions using convolutional sparse models.

A. Convolutional Sparse Representations

Convolutional sparse representations [2] express the high-frequency content of an image $s \in \mathbb{R}^{N_1 \times N_2}$ as the sum of M convolutions between filters d_m and coefficient maps x_m, i.e.

$$s_h \approx \sum_{m=1}^{M} d_m * x_m \,, \tag{3}$$

where $*$ denotes the two dimensional convolution, and the coefficient maps $x_m \in \mathbb{R}^{N_1 \times N_2}$ have the same size of the image s. Filters $\{d_m\}$[1] might have different sizes, but are typically much smaller than the image.

Coefficient maps are assumed to be sparse, namely only few elements of each x_m are nonzero, thus $\|x_m\|_0$ (the number of nonzero elements) is small. Sparsity regularizes the model and prevents trivial solutions of (3).

[1]For notational simplicity we omit $m \in \{1, \dots, M\}$ from the collection of filters and coefficient maps.

1) Dictionary learning: To detect anomalous regions we need a collection of filters $\{d_m\}$ that specifically approximate the local structures of normal images. These filters represent the *dictionary* of the convolutional sparse model, and can be learned from a normal image s provided for training. Dictionary learning is formulated as the following optimization problem

$$\begin{aligned} \underset{\{d_m\},\{x_m\}}{\arg\min}\ & \frac{1}{2}\left\| \sum_{m=1}^{M} d_m * x_m - s_h \right\|_2^2 + \lambda \sum_{m=1}^{M} \|x_m\|_1 \,, \\ \text{subject to}\ & \|d_m\|_2 = 1, \quad m \in \{1, \dots, M\} \,, \end{aligned} \tag{4}$$

where $\{d_m\}$ and $\{x_m\}$ denote the collections of M filters and coefficient maps, respectively. To simplify the notation, (4) presents dictionary learning on a single training image s, however, extending (4) to multiple training images is straightforward [11].

The first term in (4) denotes the reconstruction error, i.e. the squared ℓ^2 norm of the residuals, namely

$$\left\| \sum_m d_m * x_m - s_h \right\|_2^2 = \sum_{c \in \mathcal{X}} \left(\sum_m (d_m * x_m)(c) - s_h(c) \right)^2 , \tag{5}$$

while the ℓ^1 norm of the coefficient maps is defined as

$$\sum_m \|x_m\|_1 = \sum_m \sum_{c \in \mathcal{X}} |x_m(c)|. \tag{6}$$

In practice, the penalization term in (4) promotes the sparsity of the solution [17], namely the number of nonzero coefficients in the feature maps. Thus, the ℓ^1 norm is often used as replacement of ℓ^0 norm to make (4) computationally tractable. The constraint $\|d_m\|_2 = 1$ is necessary to resolve the scaling ambiguity between d_m and x_m (i.e. x_m can be made arbitrarily small if the corresponding d_m is made correspondingly large). We solve the dictionary learning problem using an efficient algorithm [18] that operates in Fourier domain. Learned filters typically report the prominent local structures of training images, as shown in Figure 2(a).

We observe that filters $\{d_m\}$ learned in (4) may have different sizes [18], which is a useful feature for dealing with image structures at different scales. Figure 2(a) provides an example where 8 filters of size 8×8 and 8 filters of size 16×16 were simultaneously learned from a training image.

2) Sparse Coding: The computation of coefficient maps $\{x_m\}$ of an input image s_h with respect to a dictionary $\{d_m\}$ is referred to as *sparse coding*, and consists in solving the following optimization problem [2]:

$$\underset{\{x_m\}}{\arg\min} \frac{1}{2}\left\| \sum_m d_m * x_m - s_h \right\|_2^2 + \lambda \sum_m \|x_m\|_1 \,, \tag{7}$$

where filters $\{d_m\}$ were previously learned from (4).

The sparse coding problem (7) can be solved via the Alternating Direction Method of Multipliers (ADMM) algorithm [19], exploiting an efficient formulation [11] in the Fourier domain. The dictionary-learning problem is typically solved by alternating the solution of (4) with respect to the coefficient maps $\{x_m\}$ when filters are fixed (sparse coding) and then with respect to the filters $\{d_m\}$ when coefficient maps are fixed.

B. Indicators

To determine whether each patch $\mathbf{s}_c$ is normal or anomalous we compute an *indicator vector* $\mathbf{g}(c)$ that quantitatively assesses the extent to which $\mathbf{s}_c$ is consistent and with normal patches. Indicators $\mathbf{g}(c)$ are computed from the decomposition of s in (2) to assess the extent to which the dictionary $\{d_m\}$ matches the structures of s_h around c (Section IV-B1), as well as the similarity between low frequency content of $\Pi_c s_l$ and normal patches (Section IV-B2).

1) High-Frequency Components: In anomalous regions filters are less likely to match the local structures of s_h, thus it is reasonable to expect the sparse coding (7) to be less successful, and that either the coefficient maps would be less sparse or that (3) would yield a poorer approximation. This implies that we should monitor the reconstruction error (5) and the sparsity of coefficient maps (6), locally, around c. However, we observed that monitoring the sparsity term (6) is too loose a criterion for discriminating anomalous regions.

To improve the detection performance, we take into consideration also the distribution of nonzero elements across different coefficient maps. This choice was motivated by the empirical observation that often, within normal regions – where filters are well matched with image structures – only few coefficient maps are simultaneously active; in contrast, within regions where filters and image structures do not match, more filters are typically active. Figure 2(b) compares two coefficient maps within normal and anomalous regions, and shows that in the latter case more filters are active. For this reason, we also include a term in $\mathbf{g}(h)$ to monitor the local group sparsity of the coefficient maps, namely:

$$\sum_m \|\Pi_c x_m\|_2 = \sum_{m=1}^{M} \sqrt{\sum_{u\in\mathcal{U}} \left(x_m(c+u)\right)^2}. \tag{8}$$

The indicator based on the high frequency components of the image is defined as

$$\mathbf{g}_h(c) = \begin{bmatrix} \left\|\Pi_c \left(s_h - \sum_m d_m * x_m\right)\right\|_2^2 \\ \sum_m \|\Pi_c x_m\|_1 \\ \sum_m \|\Pi_c x_m\|_2 \end{bmatrix}, \tag{9}$$

where first and second components represent the reconstruction error of $\mathbf{s}_c$ and the sparsity of the coefficient maps in the vicinity of c, respectively: these directly refer to the terms in (7), thus inherently indicate how successful the sparse coding was. The third element in (11) represents the group sparsity, and indicates the spread of nonzero elements across different coefficient maps in the vicinity of pixel c.

2) Low-Frequency Components: Anomalies affecting the low frequency components of s are in principle easy to detect as these affect, for instance, the average value of each patch. We analyze the first two sample moments over patches $\Pi_c s_l$, extracted from $\mathbf{s}_l$. More precisely, for each pixel c of $\mathbf{s}_l$ we define

$$\mathbf{g}_l(c) = \begin{bmatrix} \mu_c \\ \sigma_c^2 \end{bmatrix}, \tag{10}$$

where $\mu_c = \sum_{u\in\mathcal{U}} s_l(u+c)/P$ denotes the sample mean and $\sigma_c^2 = \sum_{u\in\mathcal{U}} (s_l(u+c) - \mu_c)^2/(P-1)$ the sample variance computed over the patch of s_l centered in c.

Both $\mathbf{g}_h$ and $\mathbf{g}_l$ can be stacked in a single vector to be jointly analyzed when testing patches, namely

$$\mathbf{g}(c) = \begin{bmatrix} \mathbf{g}_l(c) \\ \mathbf{g}_h(c) \end{bmatrix}. \tag{11}$$

This is the indicator we use to determine whether a patch is anomalous or not, analyzing both the high and the low frequency content in the vicinity of pixel c.

3) Detecting Anomalous Patches: We treat indicators as random vectors and detect as anomalous all the patches yielding indicators that can be considered outliers. Therefore, we build a confidence region $\mathcal{R}_\gamma$ around the mean vector [20] for normal patches, namely:

$$\mathcal{R}_\gamma = \left\{ \phi \in \mathbb{R}^2 \ : \ \sqrt{(\phi - \overline{\mathbf{g}})^T \Sigma^{-1} (\phi - \overline{\mathbf{g}})} \leq \gamma \right\}, \tag{12}$$

where $\overline{\mathbf{g}}$ and Σ denote the sample mean and sample covariance matrix of indicators extracted from normal images in T, and $\gamma > 0$ is a suitably chosen threshold. $\mathcal{R}_\gamma$ represents an high-density regions for indicators extracted from normal patches, since the multivariate Chebyshev's inequality ensures that, for a normal patch $\mathbf{s}_c$, holds

$$\Pr(\{\mathbf{g}(\mathbf{s}_c) \notin \mathcal{R}_\gamma\}) \leq \frac{2}{\gamma^2}, \tag{13}$$

where $\Pr(\{\mathbf{g}(\mathbf{s}_c) \notin \mathcal{R}_\gamma\})$ denotes the probability for a normal patch $\mathbf{s}_c$ to lie outside the confidence region (false-positive detection). Therefore, outliers can be simply detected as vectors falling outside a confidence region $\mathcal{R}_\gamma$, i.e.

$$\sqrt{(\mathbf{g}(c) - \mu)^T \Sigma^{-1} (\mathbf{g}(c) - \mu)} > \gamma, \tag{14}$$

and any patch $\mathbf{s}_c$ yielding an indicator $\mathbf{g}(c)$ satisfying (14) is considered anomalous.

V. Experiments

We design two experiments to assess the anomaly-detection performance of convolutional sparse representations and, in particular, the advantages of monitoring the local group sparsity of the coefficient maps. In the first experiment we detect anomalies by monitoring both the low and high frequency components of an input image, while in the second experiment we exclusively analyze the high-frequency components. The latter experiment is to assess the performance of detectors based exclusively on sparse models.

Considered Algorithms: We compare the following four algorithms built on the framework of Algorithm 1 and 2:

- **Convolutional Group**: a convolutional sparse model is used to approximate s_h. Anomalies are detected by analyzing the indicator $\mathbf{g}$ in (11) which includes also the local group sparsity of the coefficient maps (8). The indicator has five dimensions.
- **Convolutional**: the same dictionary of filters for the Convolutional Group solution is used to approximate s_h, however, the local group sparsity term of the coefficient maps (8) is not considered in $\mathbf{g}$. Thus, the indicator has four dimensions.

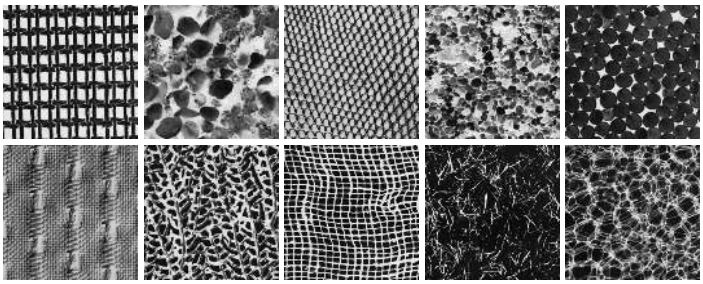


Fig. 3: Ten of the textures selected from Brodatz dataset (textures 20, 27, 36, 54, 66, 83, 87, 103, 109, 111). For visualization purposes, we show only 240×240 regions cropped from the original images, which have size 640×640.

- **Patch-based**: a standard sparse model [21] rather than a convolutional sparse model is used to describe patches extracted from s_h, as in [12]. The indicator includes the reconstruction error and the sparsity of the representation. To enable a fair comparison against convolutional sparse models, the indicator includes the sample mean over each patch from s_h and s_l, as well as the sample variance over each patch from s_l. The indicator has five dimensions.
- **Sample Moments**: no model is used to approximate the image and we compute the sample mean and variance over patches from s_l and s_h. The indicator has four dimensions.

In the second experiment, where only the high-frequency content of images is processed, the same algorithms are used. In particular, the Convolutional Group computes three-dimensional indicator ($\mathbf{g}_h$), the Convolutional computes a two-dimensional indicator (only the first two components of $\mathbf{g}_h$ are used), the Patch-based operates only on s_h computing a three-dimensional indicator and the Sample Moments also operates on s_h computing a two-dimensional indicator.

We learned dictionaries of 16 filters (8 filters of size 8×8 and 8 of size 16×16), solving (4) with $\lambda = 0.1$. The same value λ was used also in the sparse coding (7). The dictionaries in the patch-based approach are 1.5 times redundant, and learned by [22]. In all the considered algorithms, indicators are computed from 15×15 patches.

Preprocessing: We perform a low-pass filtering of each test image s to extract the low frequency components. More precisely, s_l corresponds to the solution of the following optimization problem

$$\arg\min_{s_l} \frac{1}{2} \|s_l - s\|_2^2 + \frac{\alpha}{2} \|\nabla s_l\|_2^2 , \qquad (15)$$

where ∇s_l denotes the image gradient, $\alpha > 0$ regulates the amount of high frequency components in s_l (in our tests $\alpha = 10$). The problem (15) can be solved as a linear system and admits a closed-form solution.

Dataset: To test the considered algorithms we have selected 25 textures from the Brodatz dataset [23] having a structure that can be properly captured by 15×15 filters and patches (10 of the selected textures are shown in Figure 3). A dataset of test images has been prepared by splitting each texture image in two halves: the left half was exclusively used for training, the right half for testing. The right halves of the 25 textures are pair-wise combined, creating 600 test images by horizontally concatenating two different right halves. An example of test image is reported in Figure 2(b). We consider the left half of each test image as normal and the right half as anomalous, therefore we perform anomaly detection using the model learned from the texture on the left half. Note that, since anomalies are detected in a patch-wise manner, having anomalies covering half test image does not ease the anomaly detection task with respect to having localized anomalies like those in Figure 1(a).

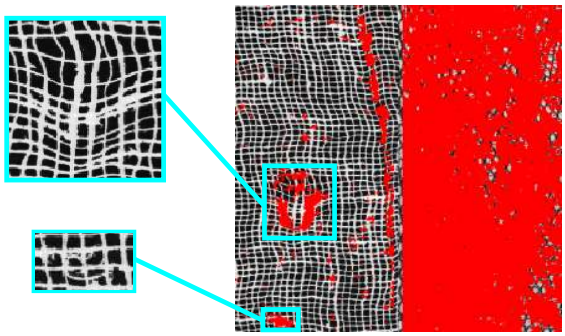


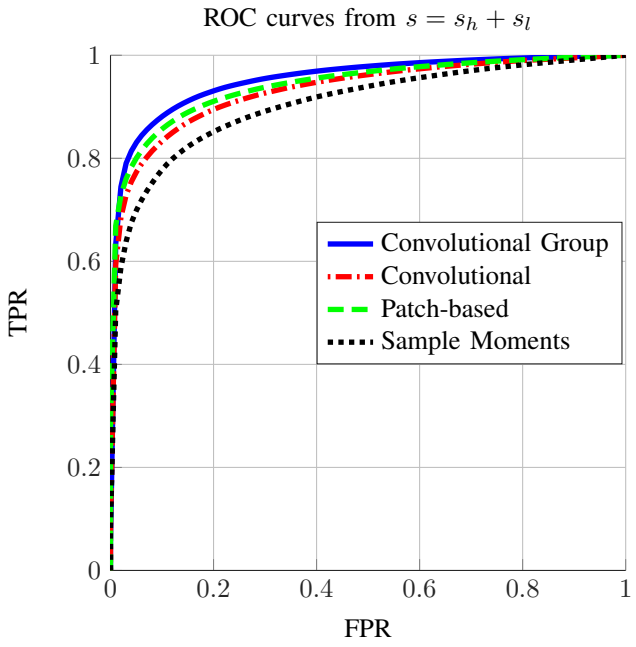
Fig. 4: Example of anomaly-detection performance for the Convolutional Group algorithm. Any detection (red pixels) on the left half represents a false positive, while any detection on the right half a true positive. The ideal anomaly detector would here detect all the points in the left half and none on the right half. Patches across the vertical boundary are not considered in the anomaly detection to avoid artifacts. As shown in the highlighted regions, most of false positives in this example are due to structure that do not conform to the normal image in Figure 2(a).

Figures of Merit: The anomaly detection performance are assessed by the following figures of merit:

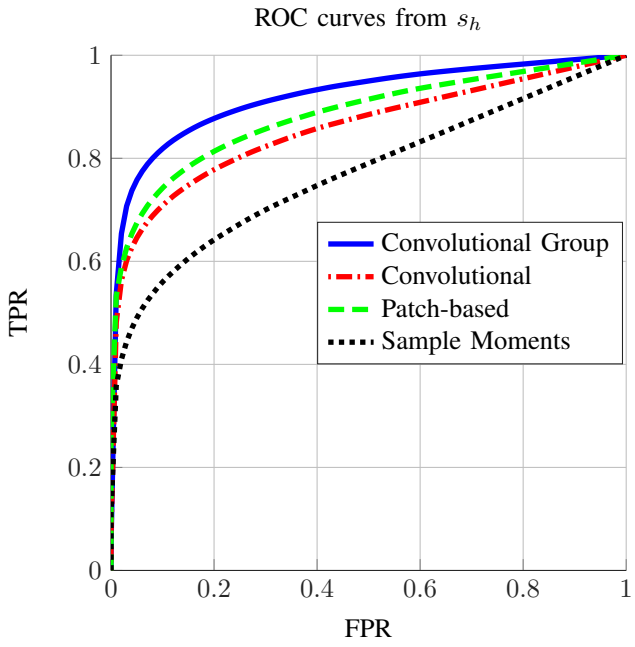
- FPR, false positive rate, i.e. the percentage of normal patches detected as anomalous.
- TPR, true positive rate, i.e. the percentage of correctly detected patches.

Figure 4 provides an example of false positives and true positives over a test image (the left half being normal, the right half anomalous).

Both FPR and TPR depend on the specific value of γ used when defining $\mathcal{R}_\gamma$. To enable a fair comparison among the considered algorithms we consider a wide range of values for γ and plot the receiver operating characteristic (ROC) curve for each method. In practice, each point of a ROC curve corresponds to the pair (FPR, TPR) achieved for a specific value of γ. When computing FPR and TPR, we exclude patches overlapping the vertical boundary between different textures.



(a) The Area Under the Curve values are: Convolutional Group: 0.9520; Convolutional: 0.9317; Patch-Based: 0.9422; Sample Moments: 0.9048



(b) The Area Under the Curve values are: Convolutional Group: 0.9198; Convolutional: 0.8578; Patch-Based: 0.8836; Sample Moments: 0.7706

Fig. 5: ROC curves for the algorithm considered in Section V, obtained by varying the parameter γ in the definition of the confidence region $\mathcal{R}_\gamma$. Figure (a) shows the performance in detecting anomalies when the whole spectrum of the images is considered, while in Figure (b) are reported the ROC curves obtained by monitoring s_h.

Figures 5(a) and 5(b) report the average ROC curves[2] when processing the whole image spectrum and high frequency components only, respectively. The Area Under the Curve (AUC) is the typical scalar metric to compare the detection performance: AUC values are indicated in the captions of the Figures and in Figure 6, which displays the AUC averaged over all the images having a specific texture as normal.

Discussions: These experiments indicate that convolutional sparsity provides an effective model for describing local image structures and detecting anomalous regions. In both the ROC curves of Figure 5, the best performance is achieved when considering the group sparsity of the coefficient maps, suggesting that the spread of nonzero coefficients across different maps is relevant information for detecting anomalies. This clearly emerges from Figure 5(a), where the Convolutional Group algorithm substantially outperforms the others, and from Figure 5(b) which shows an increased performance gap when anomalies can be only perceived from the high-frequency components.

VI. Conclusions

We have presented a novel approach for detecting anomalous structures in images by learning a convolutional sparse model that describes the local structures of normal images. Convolutional sparse models are shown to be effective at detecting anomalies in the high frequency components of images, and this is essential in applications where the anomalies are not very apparent from the low-frequencies of the image, or where anomalies affecting the low frequency content have not to be detected.

Our experiments also show that the local group sparsity of the coefficient maps is an essential information for assessing whether regions in a test image conform or not the learned convolutional sparse model. In our ongoing work we will investigate whether the local group sparsity represents a general regularization prior for convolutional sparse models, and design a specific sparse-coding algorithm that leverages a penalization term to promote this form of regularity.

[2]The curves were averaged over the whole dataset setting the same FPR values.

References

[1] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[2] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2528–2535.

[3] M. Elad, *Sparse and redundant representations: from theory to applications in signal and image processing*. Springer Science & Business Media, 2010.

[4] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[5] M. Markou and S. Singh, "Novelty detection: a review - part 1: statistical approaches," *Signal processing*, vol. 83, no. 12, pp. 2481–2497, 2003.

[6] ——, "Novelty detection: a review - part 2: neural network based approaches," *Signal processing*, vol. 83, no. 12, pp. 2499–2521, 2003.

[7] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support vector method for novelty detection." in *Advances in Neural Information Processing Systems (NIPS)*, 1999, pp. 582–588.

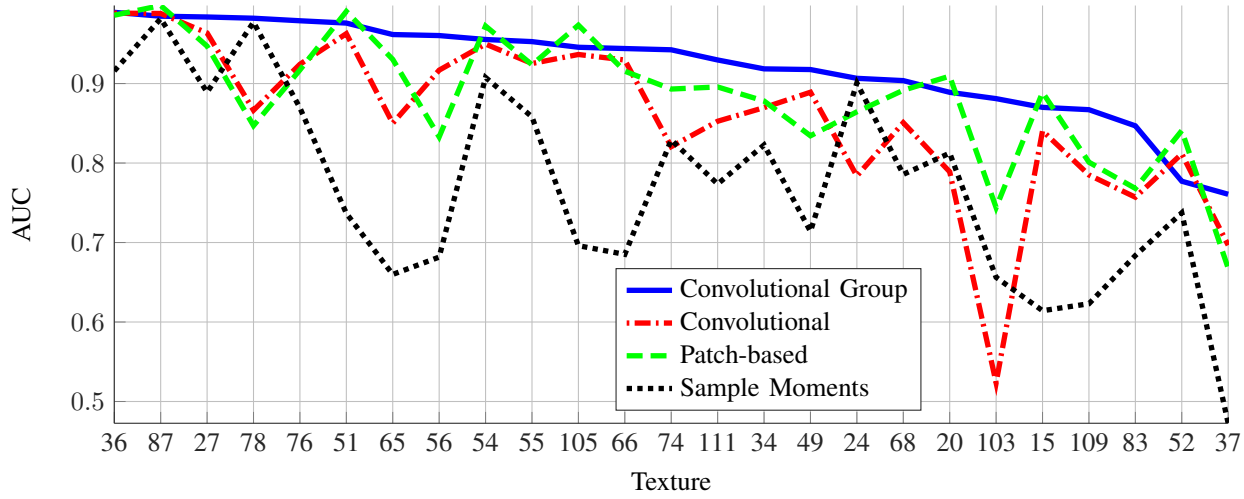


Fig. 6: For each texture it is shown the AUC value averaged over the test images where such texture is considered normal. Convolutional Group achieves the best performance in almost all cases, and in some cases it substantially outperforms other algorithms.

[8] K. Kavukcuoglu, P. Sermanet, Y.-L. Boureau, K. Gregor, M. Mathieu, and Y. L. Cun, "Learning convolutional feature hierarchies for visual recognition," in *Advances in Neural Information Processing Systems (NIPS)*, 2010, pp. 1090–1098.

[9] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proceedings of Annual International Conference on Machine Learning (ICML)*, 2009, pp. 609–616.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[11] B. Wohlberg, "Efficient convolutional sparse coding," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7173–7177.

[12] G. Boracchi, D. Carrera, and B. Wohlberg, "Novelty detection in images by sparse representations," in *Proceedings of IEEE Symposium on Intelligent Embedded Systems (IES)*, 2014, pp. 47–54.

[13] A. Adler, M. Elad, Y. Hel-Or, and E. Rivlin, "Sparse coding with anomaly detection," in *Proceedings of IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[14] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3313–3320.

[15] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 3449–3456.

[16] C. Alippi, G. Boracchi, and B. Wohlberg, "Change detection in streams of signals with sparse representations," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 5252–5256.

[17] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, pp. 267–288, 1996.

[18] B. Wohlberg, "Efficient algorithms for convolutional sparse representations," Manuscript currently under review, 2015.

[19] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[20] R. Johnson and D. Wichern, *Applied multivariate statistical analysis*. Prentice Hall, 2002.

[21] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[22] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the Annual International Conference on Machine Learning (ICML)*, 2009, pp. 689–696.

[23] P. Brodatz, *Textures: A Photographic Album for Artists and Designers*. Peter Smith Publisher, Incorporated, 1981.