How to Reduce Anomaly Detection in Images to Anomaly Detection in Noise

Thibaud Ehret[1], Axel Davy[1], Mauricio Delbracio[2], Jean-Michel Morel[1]

[1] CMLA, ENS Cachan, CNRS, Université Paris-Saclay, 94235 Cachan, France
(`{thibaud.ehret, axel.davy, morel}@cmla.ens-cachan.fr`)
[2] IIE, Facultad de Ingeniería, Universidad de la República, Montevideo, Uruguay
(`mdelbra@fing.edu.uy`)

Communicated by Yann Gousseau   *Demo edited by* Thibaud Ehret

Abstract

Anomaly detectors address the difficult problem of detecting automatically exceptions in a background image, that can be as diverse as a fabric or a mammography. Detection methods have been proposed by the thousands because each problem requires a different background model. By analyzing the existing approaches, we show that the problem can be reduced to detecting anomalies in residual images (extracted from the target image) in which noise and anomalies prevail. Hence, the general and impossible background modeling problem is replaced by a simple noise model, and allows the calculation of rigorous detection thresholds. Our approach is therefore unsupervised and works on arbitrary images. The residual images can favorably be computed on dense features of neural networks. Our detector is powered by the a contrario detection theory, which avoids over-detection by fixing detection thresholds taking into account the multiple tests.

Source Code

The reviewed source code and documentation for this algorithm are available from the web page of this article[1]. Compilation and usage instruction are included in the `README.txt` file of the archive.

Keywords: Anomaly detection; saliency; multiscale; background modeling; background subtraction; clustering; K-nearest-neighbors; self-similarity; nonlocal means; k-means; PCA; autoencoders; Fourier transform; PHase Only Transform (PHOT); feature histogram; wavelet; center-surround; SVM; neural networks; sparse dictionary; information measure; H0 hypothesis; hypothesis testing; p-value; Mahalanobis distance; a contrario assumption; number of false alarms; NFA

---

[1]https://doi.org/10.5201/ipol.2019.263

THIBAUD EHRET, AXEL DAVY, MAURICIO DELBRACIO, JEAN-MICHEL MOREL

1 Introduction

The automatic detection of anomalous structure in arbitrary images is concerned with the problem of delineating image regions not conforming with the rest of the image. This is a challenging computer vision problem, as there seems to be no straightforward definition of what is (ab)normal for a given image.

Anomalous structure in images can be generally described as being either caused by high-level or low-level outliers. High-level anomalies are related to the semantic information presented in the scene. For example, human observers immediately detect a person inappropriately dressed for a given social event. In this work, we focus on the problem of detecting anomalies due to low- or mid-level rare events (e.g., patterns) present in images. This is an important problem in many industrial or biomedical applications where a fast and reliable way of detecting rare patterns is needed.
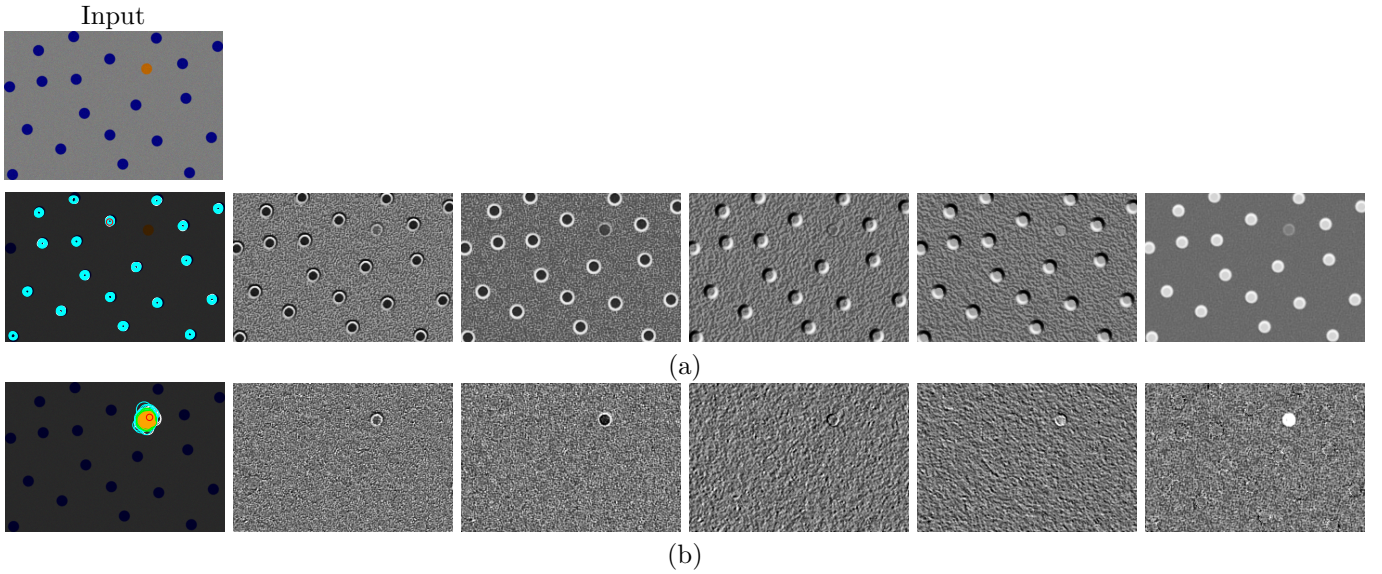

Figure 1: Image anomalies are successfully detected by removing all self-similar content and then looking for structure in the residual noise, which parameters are easy to estimate. First row: an image with a color anomaly (the red dot); (a): the detections (first column) obtained from principal components of CNN feature maps (2nd-5th columns) (b): the detection map on the same features obtained after removing the self-similar content. In the detection maps, cyan means good detection and orange extremely salient detection.

We propose here an unsupervised method for detecting anomalies in an arbitrary image. The method doesn't rely on a training dataset of normal or abnormal images, neither on any other prior knowledge about the image statistics. It directly detects anomalies with respect to residual images estimated solely from the image itself. We only use a generic, qualitative background image model: we assume that anything that repeats in an image is *not* an anomaly. Hence we extract from the input image a residual image containing everything that does not repeat. This unstructured image is akin to noise, but still contains the anomalies, if any.

Detecting anomalies in noise is far easier and can be made rigorous and unsupervised by the a contrario theory [18] which is a probabilistic formalization of the non-accidentalness principle [39]. The a contrario framework has produced impressive results in many different detection or estimation computer vision tasks, such as, segment detection [29], ellipse detection [49], spots detection [30], vanishing points detection [38], fundamental matrix estimation [45], mirror-symmetry detection [48], among others. The fundamental property of the a contrario theory is that it provides a way for automatically computing detection thresholds that yield a control on the number of false alarms (NFA). It follows that not only one can detect anomalies in arbitrary images without complex modeling,

but in addition the anomalies are associated an NFA which is often very small and therefore offers a strong guarantee of detection.

In a nutshell, our method removes from the image its self-similar content (considered as being normal). The residual can be modeled after a simple equalization as Gaussian noise, but still contains the anomalies according to their definition: they do not repeat. We shall show detections performed directly on the residual, or alternatively on residuals extracted from dense low- and mid-level features of VGG [57] the popular pre-trained deep neural network. Our method is general and works equally well on these different image representations.

A preliminary short version of this work was published in a conference [16]. The anomaly detection method developed here is also described briefly in our review paper [24]. Our Section 2 below summarizes some of the conclusions about the literature contained in this last paper. The present version incorporates a more detailed analysis of the detection method and its implementation.

The remainder of the paper is organized as follows. Section 2 discusses in detail previous work and the substantial differences to what we propose. Section 3 explains the method and its implementation. In Section 4 we present results of the proposed method on both real and synthetic data, comparing the algorithm to other state-of-the-art anomaly detection methods. We finally close in Section 5.

2 An Analysis of the Literature

The 2009 review [14] examining some 400 papers on anomaly detection considered allegedly all existing techniques and all application fields. It is fairly well completed by the more recent [52] review. These reviews agree that classification techniques like SVM can be discarded, because anomalies are generally not observed in sufficient number and lack statistical coherence. There are exceptions like the recent method [19] which defines anomalies as exceptional events that cannot be learned, but after estimating a background density model, the right detection thresholds are nevertheless learned from anomalies. A broad-related literature exists on saliency measures, for which learning from average fixation maps by humans is possible. For example [59] trained on average human fixation maps to learn both the anomalies and their surround vectors as Gaussian vectors. This reduces the problem to a two class Bayesian classification problem. The goal of saliency detectors is only to deliver a fuzzy saliency map. Anomaly detectors instead signal anomalous regions. The saliency detectors try to mimic the human visual perception and in general introduce semantic prior knowledge related to the perceptual system (e.g., face detectors). This approach works particularly well with neural networks because attention maps obtained by gaze trackers can be used as a ground truth for the training step. SALICON [33] from Huang et al. is one of these deep neural networks architecture achieving state of the art performance. In [37] a neural network is trained on a base of defect/non-defect, thus again performing two classes classification. But the anomaly detection problem has been generally handled as a "one class" learning problem. The 2003 very complete review by Markou and Singh [42] concluded that most research on anomaly detection was driven by modeling background data distributions, to estimate the probability that test data do not belong to such distributions. Hence the mainstream methods can be classified by their approach to background modeling.

2.1 Probabilistic Background Models

Their principle is that anomalies occur in the low probability regions of the background model. This stochastic model can be parametric (Gaussian, Gaussian mixture, regression), or nonparametric. For example in "spectral anomaly detection", an anomaly is defined by having deviant coordinates with respect to normal PCA coordinates. In [2] a Gaussian background Fourier model of the image phase is followed by a Mahalanobis threshold. In [20] a Gaussian background model from random pixels is

similarly followed by a Mahalanobis threshold. In [28] the background is characterized in a feature space of principal components, and hypothesis testing is used for the detection of anomalous pixels. In [58] the assumption is made that abnormalities are uniformly distributed outside the boundaries of normality, defined as the probability density estimation of the training data. In [32] Honda and Nayar introduced a generic method which works on all type of images. The main idea is to estimate the probability of a region conditioned on the surroundings. The method employs independent component analysis to find a compact representation of the region space and its surroundings. The method in [30] models the background image as a Gaussian stationary process. This is rather restrictive, but this precise model allows computing an accurate Number of False Alarms [18] for anomalies.

Thus, in methods relying on a probabilistic background model, outliers are detected as incoherent with a probability distribution estimated from the input image(s). The anomaly threshold is a statistical likelihood test on the learned background model. We now pass to non-stochastic background models.

NN-based background reconstruction. "Replicator" neural networks [31] can model a background. These are multi-layer feed forward neural networks with same number of input and output neurons. The training involves compressing data into hidden layers. The testing phase reconstructs each data sample. Its reconstruction error for the test instance is used as an anomaly score. The work in [55] is also equivalent to using an autoencoder and looking at the norm between the original and the output. A GAN is trained (generator + discriminator) via gradient descent, a representation in latent space is computed, and the output is compared to the input. The discriminator cost is then used alongside the representation on the input by the network to find the anomalies. This paper is related to [3] that computes a reconstruction probability from a variational autoencoder.

Fourier Background subtraction. Perhaps the most successful background-based method is the detection of anomalies in periodic patterns of textile [62, 63, 50]. This can be done naturally by cutting specific frequencies in the Fourier domain and thresholding the residual to find the defects.

2.2   Non-Constructive Background Models

These methods present the big advantage that they no longer require the construction of a background model, which for most images is anyway impossible. Hence, they simply make structural assumptions on the background image that would be violated by anomalies.

Center-surround enhancement. These methods are mainly used for creating saliency maps. Their rationale is that anomalies pop up as local events contrasted with their surroundings. In [35], center surround detectors based on color, orientation and intensity filters are combined to produce a final saliency map. Detection is then done on a simple winner-takes-all scheme on the maximum of the response maps. In [46] a saliency map is obtained from center-surround contrast coefficients for wavelet filters. An image wavelet pyramid reconstruction with these coefficients enhances local anomalies. Detection in images and videos is also done in [26] with center-surround saliency detectors which stem from [34] adopting similar image features. In [32], the main idea of a fast and general anomaly detection method is to estimate the probability of a region conditioned on the surroundings. The method employs ICA and KLT to find a compact (with elements as independent as possible) of the region space and its surroundings.

The sparsity model. A more recent nonparametric trend is to learn a sparse dictionary representing the background (i.e., *normality*) and to characterize outliers by their non-sparsity. In [41] the patch background model is simply its PCA and the patch saliency is computed as the L^1 norm of the patch coefficients in PCA. Aggregating these values gives a pixel saliency. The saliency formation in [64] builds a Gaussian mixture model for patches, and probability thresholds are learned on images without anomalies. The final result is a saliency map. In [7] the background model is a learned patch dictionary from a database of anomaly-free data. The abnormality of a patch is measured as its Mahalanobis distance to a 2D Gaussian learned on the parameter pairs composed by the L^1 norm of the coefficients and of their reconstruction error. An extension [13] learns a convolutional sparse dictionary. Similarly in [25] a patch is anomalous when the L^1 norm of its sparse decomposition on a learned dictionary is too large.

Defining anomaly detection as a variational problem where anomalies are detected as non-sparse is also the core of the method proposed in [1]. The L^1 norm of the coefficients on the learned background dictionary is used as an anomaly measure.

The self-similarity model. The self-similarity principle has been successfully used in many different applications. In particular in image denoising such as the bilateral filter [61] or non-local means [11] for example; but also for texture synthesis in the pioneering work by Efros and Leung [21]. The basic assumption of this generic background model, applicable to most images, is that in normal data, features are densely clustered. Anomalies instead occur far from their closest neighbors. This idea can be implemented by clustering (anomalies being detected as far away from the centroid of their own cluster), or by nearest neighbor search (NN). NN search leads to simple direct rarity measurements. For example in [56] the saliency measure is $S_i = \left(\sum_{j=1}^{N} \exp \left(\frac{-1+\rho(F_i,F_j)}{\sigma^2} \right) \right)^{-1}$ where F_i are local features, and F_j the closest features to F_i. If all F_j are far away from F_i, the saliency is high. The algorithm in [66] is inspired from NL-means [11]: a) fix a similarity threshold learned in a reference image without anomalies and b) compare each patch of the source image to the patches of the reference; if the distance is higher than the similarity threshold, then the patch is an anomaly. A similar idea can be found back in [60]: "The distance of the new object and its nearest neighbour in the training set is found and the distance of this nearest neighbour and its nearest neighbour in the training set is also found. The quotient between the first and the second distance is taken as indication of the novelty of the object." The self-similarity measurement in [27] finds for each patch p_i its 64 most similar patches q_k in a spatial neighborhood and computes its saliency as $S_i = 1 - \exp \left(-\frac{1}{64} \sum_{k=1}^{64} d(p_i, q_k) \right)$.

A similar method and saliency detector is proposed in [44]. This method performs a dimension reduction thanks to a low dimensional embedding of a nearest neighbor graph using the coordinates of points on the eigenvectors of the graph Laplacian. The anomaly score is then given by the distance to the first nearest neighbors with the diffusion distances. In [6], image regions are matched (with deformation allowed) to others in the same image or video. The probability of the deformation is estimated and gives a saliency map.

Global rarity measurement. Generally histogram based, these methods assign an anomaly score to each tested feature based on the inverse of the height of the bin to which it belongs. Similarly in [54] a saliency map is obtained by summing up the rarity of 32 multiscale oriented features, computed for each pixel as a weight inversely proportional to its rarity in the wavelet histogram. Patches are represented in [9] by their coefficients on a patch dictionary learned on natural images. The final saliency is the inverse of a patch probability of happening. The method is finally combined with a local center-surround saliency measure.

2.3   Discussion and our Proposition

We now discuss briefly the above method classification. Background probabilistic modeling is very powerful when images belong to a restricted class of homogeneous objects, like textiles. Indeed, it furnishes rigorous detection thresholds based on the estimated probability density function. But, regrettably, this method is nearly impossible to apply on generic images. For the same reason, background reconstruction models based on CNNs are restrictive and do not rely on provable detection thresholds. Center-surround contrast methods are successful for saliency enhancement, but lack a detection mechanism. Hence, they only furnish a saliency image, not a binary anomaly decision. The sparsity and the self-similarity models are tempting and thriving. Their big advantage is their universality: they can be applied to most images. But again, they lack a rigorous detection mechanism, because they work on a feature space that is not easily modeled.

Our proposition is to benefit of the advances of the above methods while avoiding their mentioned limitations. To this aim, we do construct a probabilistic background model, but it is applied to a new feature image that we call the residual. This residual is obtained by reconstructing a self-similar version of the target image. The difference between the target and its self-similar version is called the *residual* and becomes our new background. Being not self-similar, this background is akin to a colored noise. Hence hypothesis testing can be applied to it, and more precisely multiple hypothesis testing (also called a contrario method), as proposed in [30].

In that way, a general and simple method can be built that works on all images and detects anomalies by a rigorous threshold. It does not require learning, and it is easily made multiscale. Our underlying model can therefore be considered as fully generic in the sense that it decomposes any image into a self-similar part and its "residual", which contains only noise and the anomalies.

3   Method

The anomaly detection method that we propose is therefore built on two main blocks: a removal of the self-similar part of the image, and a simple statistical detection test based on the a contrario framework on the residual. The pipeline is summarized in Figure 2 and in Algorithm 1.
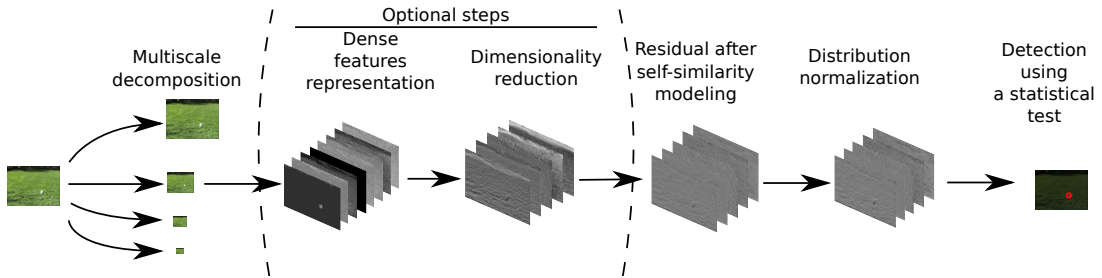

Figure 2: Our method applies the same framework at all the different scales. The framework computes the difference between the image (possibly using a dense representation using a neural network) and its self-similar model before normalizing this difference output distribution. The detection is then done on these normalized distributions. The corresponding pseudocode is presented in Algorithm 1.

3.1   Self-Similarity Background Modeling

The proposed self-similarity based background subtraction is inspired from patch-based denoising algorithms, but with a crucial difference. All self-similarity based denoising algorithms share a similar procedure. First, a set of similar patches is computed. This search is generally performed locally around each patch [15, 11] to keep computational cost low and to avoid noise overfitting. A

Algorithm 1: Anomaly detection algorithm

    input   : Input image u, reference image v, size of patch sp, number of nearest neighbors n, h
                 the patch similarity parameter, a list R of disk radii to test, the NFA threshold ε, a
                 neural network nn (optional)

    output: List of detections l (contains size and position)

    $(u_i) \leftarrow multiscaleDecomposition(u, nscales)$ *See Section 3.2*
    $(v_i) \leftarrow multiscaleDecomposition(v, nscales)$ *and Algorithm 7*

    for *scale $i = 0$ to 3* **do**

        if *nn is provided* **then**

            $\hat{u}_i \leftarrow extractFeatures(u_i, nn)$ *Use neural network features to represent u_i*
            $\hat{v}_v \leftarrow extractFeatures(v_v, nn)$ *See Section 3.3*

        else

            $\hat{u}_i \leftarrow u_i$
            $\hat{v}_i \leftarrow v_i$

        end

        $r_i \leftarrow computeResidual(\hat{u}_i, \hat{v}_i, sp, n)$ *See Algorithm 5*
        $r_i \leftarrow fitResidual(r_i)$ *See Algorithm 6*
        $l_i \leftarrow detect(r_i, height(\hat{u}_0), width(\hat{u}_0), channels(\hat{u}_0), R, \varepsilon)$ *See Algorithm 8*
        Add l_i to l

    end

---

model of the query patch is then built, based on the nearest neighbors found at the previous step. These nearest neighbors will be picked with a different rule for anomaly detection than for denoising, as we shall see next.

Self-similarity from *exclusively* non-local patches. Image denoising tries to maximize the quality of the result and must therefore perform acceptably even with the non self-similar parts of the image. For anomaly detection instead, the algorithm should perform well only if a clear structure emerges. Furthermore, we want our self-similarity search to be made on the whole image. But the main difference with classical denoising is that we *forbid* local comparisons. Thus, contrary to classic denoising algorithms, the search is performed *outside a square region surrounding each query patch* only. Otherwise any anomaly with some internal structure might be considered a structure. What matters is that the event represented by the anomaly is unique, and this is checked away from it.

Searching for similar patches. In image patch-based methods, searching for similar patches is usually done in a small squared region surrounding the query patch. In our case this can't be done because we *forbid* a local squared region around the query patch. Moreover for denoising the small region is used as a regularizer so to avoid noise overfitting which is not necessary in our case. We'd also like to take advantage of the whole image; repetitions could appear anywhere in the image and not necessarily locally. This is why we used the global search presented by Ehret et al. in [23]. The idea is to take advantage of a binary tree structure to search quickly in the entire image. The VP-trees is a modification of Ball-tree [47] in the sense that it is also based on hyperballs but the VP-tree is again a binary tree. VP comes from "vantage point" which is used to define the tree. Each vantage point is representative of a region of the space. In this case, a vantage point is simply the center of a hyperball and will represents a node in the tree. The construction of such trees is summarized in Algorithms 2 and 3. Computing the optimal VP-tree would require to actually compute $O(|\mathcal{S}|^2)$ distances; this is quite costly for the sizes of $\mathcal{S}$ that are usually considered when working with the

set of patches in an image. A solution to this problem is to consider non-optimal vantage points (for example sampled uniformly) and instead of using one tree, use multiple trees and combine the results. The global search adds on top of that a small refinement step *à la* PatchMatch [5] by searching around best candidates to find even better ones. The VPLR search proposed in [23] is summarized in Algorithm 4 with each step shown in Figure 3. The advantage of this search is that it can be done in another image, potentially multiple other images, as an external search and therefore use a different reference image (see Figure 4). In this case there's no need to forbid a region of the image.

In this work we suggest using VP-trees for the global search since it has been shown to work well for a similar problem [23]. However, other global patch search methods, such as PatchMatch [5], can also be used. While PatchMatch has more convergence guarantees, see [22], it is also less computationally efficient. PatchMatch is very efficient when looking for very few (namely one or two) nearest neighbors but struggles for a large number of nearest neighbors. PatchMatch also requires the computation of similar patches for the entire database while the current approach only does it for the image that is being processed. Moreover this approach is easier to parallelize than PatchMatch.

Algorithm 2: Construction of a VP-tree

input : A set of points in a metric space $\mathcal{S}$, b the number of elements to be used for the computation of the vantage point, m the maximum size of the bins

output: The VP-tree $\mathcal{T}_{\mathcal{S}}$

if $|\mathcal{S}| \leq m$ **then**

    No splitting is needed

    return an empty tree

end

$s \leftarrow chooseVantage(\mathcal{S}, b)$ *See Algorithm 3*

Compute the median distance p from s to the other elements of $\mathcal{S}$

Construct $\mathcal{T}_1$ using the elements of $\mathcal{S}$ which are closer than p from s

Construct $\mathcal{T}_2$ using the elements of $\mathcal{S}$ which are further than p from s

return $((s,p), \mathcal{T}_1, \mathcal{T}_2)$

Algorithm 3: $chooseVantage$: Choose a vantage point

input : A set of points in a metric space $\mathcal{S}$, b the number of elements to be used for the computation of the vantage point

output: A vantage point s

Draw a set of b points P from $\mathcal{S}$

for $s \in P$ **do**

    Compute the median distance p from s to the elements of $\mathcal{S}$

    Compute the second moment of $\{\|s - u\| - p \mid u \in \mathcal{S}\}$

end

return $s \in P$ with the largest second moment

Background model computation. Once taken into account this important difference, the background removal step follows the steps of the Non-local means algorithm [11]. For a similarity parameter and for each patch P in the image the n most similar patches denoted by P_i are searched and

---

Algorithm 4: VPLR search algorithm

input : v an image, $\mathcal{F}$ a VP-tree forest constructed with the patches from v, p a query patch, $\kappa \times \kappa$ the size of local search regions, k the number of patches required, a forbidden region (possibly empty)

output: A list of matches for p

Initialize an empty list l

for *each tree $\mathcal{T}$ in $\mathcal{F}$* **do**

    Initialize t to $\mathcal{T}$

    while *t is not a bin* **do**

        $(s, d), \mathcal{T}_1, \mathcal{T}_2 \leftarrow t$

        if *the distance from p to s is smaller than d* **then**

            $t \leftarrow \mathcal{T}_1$

        else

            $t \leftarrow \mathcal{T}_2$

        end

    end

    Retrieve the list $\{\varphi_1, \ldots, \varphi_k\}$ of k best matches from t that are outside the forbidden region and add them to l

end

Keep the k best matches from l

for *each φ_i in l* **do**

    Search in the region of size $\kappa \times \kappa$ centered on φ_i in v for better matches (consider only patches outside the forbidden region)

    Keep the k best matches in l

end

return the list $l = \{\varphi'_1, \ldots, \varphi'_k\}$ of k best matches after the update using the VPLR search

---

Step 1: the search of the figures/bin of the query patch in the VP-tree

Step 2: the local search based on the elements found with the VP-tree

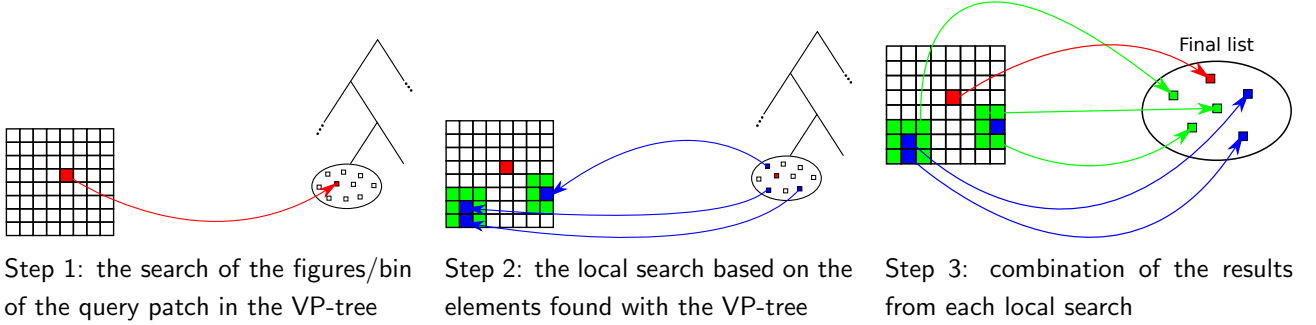Step 3: combination of the results from each local search

Figure 3: Steps of the VPLR search

averaged to get a self-similar estimate $\hat{P}$ for the patch,

$$\hat{P} = \frac{1}{\sum_{i=1}^{n} \exp\left(-\frac{\|P - P_i\|_2^2}{h^2}\right)} \sum_{i=1}^{n} \exp\left(-\frac{\|P - P_i\|_2^2}{h^2}\right) P_i. \tag{1}$$

Since each pixel belongs to several different patches, they will therefore receive several distinct estimates that can be averaged. Algorithm 5 gives a generic pseudocode for this process, which ends with the generation of a residual image $r(u)$ allegedly containing only noise and the anomalies. See Figure 1 for an illustration of this separation.
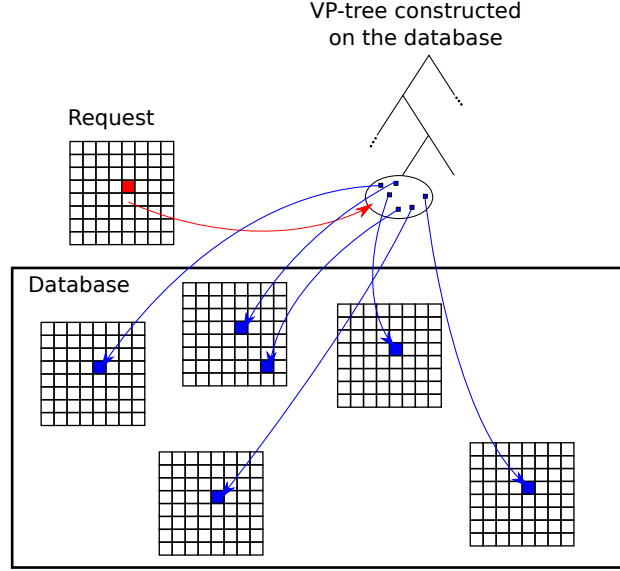
Figure 4: Using the global search to search in a database

Algorithm 5: *computeResidual*: Computation of the unstructured residual

---

input : A multichannel image u, a multichannel reference image v, n the number of nearest

neighbors, h the similarity parameter, patch size sp *v is potentially equal to u*

output: Residual $r(u) = \hat{u} - u$ where $\hat{u}$ is the model of u.

for *all multichannel patch P of u* **do**

> Estimate the n nearest neighbors $(P_1, \ldots, P_n)$ of P in v **deprived from a square region**
> **around** P if v is u.   *See Algorithm 4*
> Reconstruct the patch (using (1))

end

for *pixels j in u* **do**

> $\hat{u}(j) = \frac{\sum_{i \in \{s|j \in W_s\}} \hat{P}_i(j)}{\#\{s|j \in W_s\}}$ *Aggregate the different estimates*

end

Notation convention: W_s is the set of pixels in the patch centered at s. $\hat{P}_i(j)$ is the value at pixel j of the reconstructed patch centered at i.

Residual distribution. The distribution of $r(u)$ is not necessarily Gaussian. In [65] it is hinted that $r(u)$ might follow a Laplace distribution for natural images for some denoising algorithm. The fact that non-local means transforms white Gaussian noise into white Gaussian noise is substantiated in [12]. In our experiments, we fit a few distributions on the residual (see Algorithm 6). The distribution depends on the image and on the choice of the input features, depending on whether we work on the raw image, or with a Neural Network as mentioned in Section 3.3. Zeros are ignored for the distribution estimation (When using CNNs, RELUs cause zero to be over-represented). To equalize the residuals and simplify the detection model, we use a non-linear transform and rescale $r(u)$ in order to fit a centered Gaussian distribution with unit variance.

3.2 Statistical Detection by the A Contrario Approach

The a contrario model. Our goal is to detect structure in the feature image residual $r(u) = \hat{u} - u$. We are in a much better situation modeling $r(u)$ than u. Indeed, contrarily to u, $r(u)$ is by construction unstructured and akin to a colored noise. We treat it as a stationary probabilistic

---

Algorithm 6: *fitResidual*: Fit the residual distribution and convert it to a Gaussian

---

    input : r a set of residual values obtained from Algorithm 5
    output: r' the resulting modified residual
    Ignore all zeros for the distribution estimation.
    for *all α in* $[0.5, 0.6, \ldots, 1.4]$ **do**
       | Test if r^α is Laplacian
       | Test if r^α is Gaussian
    end

    The best distribution D and α are kept.   *Best distribution in terms of minimizing the l^2 distance from the uniform distribution of the cumulative histogram of $cdf_D(r^\alpha)$, where D is the tested distribution.*

    $r' \leftarrow inv\_cdf_{Gaussian(0,1)}(cdf_D(r^\alpha))$ *Applied element-wise*

---

spatial process and follow [30], who proposed automatic detection thresholds in any colored Gaussian noise.

Their a contrario framework computes a *Number of False Alarms (NFA)* for each value in excess in a colored noise image, under the null hypothesis that the residual is centered Gaussian noise of unit variance. The null hypothesis doesn't require the noise to be uncorrelated. Since anomalies are expected to deviate from this background model, this amounts to checking the tails of the Gaussian and to retain high values as significant if their tail has a very small area. Computing an NFA instead of a p-value is justified by the intensive multiple testing involved in the detection, which is made at every pixel and in every residual channel.

More precisely, given a set of random variables $(X_i)_{i \in [|1,N|]}$, understood as a set of threshold tests, a function f is called an NFA if it guarantees a bound on the expectation of its number of false alarms under the null-hypothesis, namely

$$\forall \varepsilon > 0, \mathbb{E}[\#\{i, f(i, X_i) \leq \varepsilon\}] \leq \varepsilon. \tag{2}$$

A common way to build an NFA is to take, for each feature channel image, $f(i, \mathbf{x}) = N\mathbb{P}(X_i \geq \mathbf{x}_i)$, or

$$f(i, \mathbf{x}) = N\mathbb{P}(|X_i| \geq |\mathbf{x}_i|), \tag{3}$$

where X_i denotes a residual image. N is the number of tests, generally the overall number of pixels of all tested images.

Detection of anomalies of different sizes at a given scale. When working with colored noise, Grosjean and Moisan [30] recommend to convolve the noise with a measure kernel to detect spots of a certain size. This corresponds to the generation of new channels $\bar{r}(u) = r(u) * K$ where K is the normalized support of a disk of a given radius. But since we apply the detection at all dyadic scales, the tested radii are limited to a small set of N_{conv} values (1 to 3) at each scale. The disks are normalized to keep unit variance. Because the residual is assumed to be a stationary Gaussian field, the results after the filtering are also Gaussian. The combination of the detections on the different versions of the residual is explicitly handled by the statistical test.

Thus, the inputs to the detection phase are multichannel images of different scales, where each one is assumed to be a centered Gaussian field having unit variance at each pixel.

Multiscaling. The problem of anomaly detection is fundamentally multiscale. One would like to find both anomalies at a fine scale, for example small anomalies inside a texture, or at a coarser scale, for example a hole inside a textile. If one wants to be thorough, all these anomalies need to be detected whatever the scale they appear in. Thus, if the detection algorithm is not inherently multiscale, it can become so by applying it at all scales. The idea to generate the different scales, process them and combine the result is inspired by multiscale denoising such as [51]. The scaled images can be computed by Gaussian subsampling. A Gaussian blur of parameter λ is applied to the image before a subsampling of 2 in each direction (so 4 in total). The standard deviation of the Gaussian λ is chosen so that the image has the same blur as the original one after subsampling. This blur is assumed to be of the order of 0.8 for natural well sampled images [40]. Since the variances of convolved Gaussians add up, this amounts to imposing a blur such that

$$0.8^2 + \lambda^2 = (2 \times 0.8)^2 \tag{4}$$

and therefore

$$\lambda = \sqrt{3} \times 0.8 \approx 1.39. \tag{5}$$

The process can then be iterated to compute all the dyadic scales. The corresponding pseudocode is in Algorithm 7.

Algorithm 7: *multiscaleDecomposition*: decompose an image in its different scales

input : An image u
output: A set (u_i) of images representing the different scales
$u_0 \leftarrow u$ *The first scale corresponds to the original image*
for $i = 1$ *to* 3 **do**
    $u_i \leftarrow G_{1.39} * u_{i-1}$ *$G_{1.39}$ is a Gaussian kernel of standard deviation 1.39, the convolution is computed using a discrete filter of size $2\lfloor 4\sigma + 0.5 \rfloor + 1$*
    $u_i \leftarrow subsample(u_i)$ *Subsample by a factor 4 (2 in each direction)*
end

The statistical test. To detect anomalies for both sides of the tails, we use the NFA given in (3). It remains to compute the number of tests N which is significantly larger than the number of image pixels. Indeed, the detection occurs on N_{scales} different scales of the residuals, computed by Gaussian subsampling. The first scale is of the size of the original image, while each other scale is reduced by a factor of two. The produced residuals are of the size of their input images (in the case of the pixel method), or smaller (in the case of CNN features. See Section 3.3). Furthermore, the images have several channels and each test is replicated on all channels.

Denoting by Ω_s the set of pixels for the residual at scale s, N_{conv} the number of convolution filters and N_{chan} the number of channels, the number of tests is

$$N = N_{conv} \cdot N_{chan} \cdot \sum_{i=0}^{N_{scales}-1} |\Omega_s|. \tag{6}$$

The detection process is summarized in Algorithm 8.

3.3 Choice of the Image Features

We now describe the image features, or channels, on which anomalies are detected. This is often considered an important step in anomaly detection. We found that our detection in the residual,

---

Algorithm 8: *detect*: Detecting anomalies in a residual image

input : A residual image r, (u_w, u_h, u_c) the size of original image u (before any downsampling), the list R of disk radii to test, the NFA threshold ε, sp the patch size used for Algorithm 5

output: A list of detections l

for R_i *in* R **do**

    Compute the disk kernel function f of radius R_i

    $\hat{r} \leftarrow r * f_i$ *Convolve the image with the kernel (exact convolution done in Fourier)*

    Estimate σ the standard deviation of $\hat{r}$

    for *each pixel $\hat{r}_k$ of $\hat{r}$ outside of the border of size $sp/2$*   *Detections are unreliable on the border*

    do

        if $\log\left(\frac{4u_h u_w u_c}{6} \text{length}(R) \operatorname{erfc}\left(\frac{|\hat{r}_k|}{\sigma}\right)\right) < \log(\varepsilon)$ **then**

            Add the position of the pixel and R_i to l

        end

    end

end

---

which is unstructured noise, is fairly independent of the channel choice. We used with equal success the raw RGB colors as channels, or the intermediate feature channels of a pre-trained neural network. To that last purpose we used the VGG network [57], a CNN trained on the ImageNet database [17].

We removed the network padding to guarantee spatial invariance. Indeed if the network padding is kept, the feature distributions differ at the borders of the feature maps [53]. We took the normalized version of the network and found slightly better results by working on the square root of the raw network features. This slight improvement might be attributed to the noise equalization property of the Anscombe transform [4]. Before the residual computation on NN features, we reduce their dimensionality with a PCA filter trained independently for each input image. This reduction is a compromise between the expressive power of the complete set of NN features and a compact image representation where visually almost-similar objects have similar representations. This transformation is represented in Figure 2.

3.4 Algorithm Parameters

The main method parameter of the statistical test is the number of allowed false alarms. In all of our experiments it was set to $1/100$. Hence an anomaly is detected at pixel $\mathbf{x}$ in channel i if and only if the NFA function $f(i, \mathbf{x})$ is below $\varepsilon = 10^{-2}$. This means a theoretical expectation of less than 10^{-2} "casual" detections per image under the null hypothesis. We shall anyway observe much smaller NFAs for the real anomalies.

Another setting is the size of the disks for the convolutions. We got better results with disks of radius one and two for the basic method working on pixels, and radius one, two and three with Neural Network features (See section 3.3). Neural Networks need bigger disks to remedy the fact that medium-sized spots for the residual at a given scale may disappear at the following scale because Neural Networks features tend to ignore very small elements.

The other parameters are fixed as follows. We set the number of scales N_{scale} to 4 in all of our tests. The patch size for Algorithm 5 is set to $8 \times 8 \times 3$ for the basic version, while when using Neural Network features, we take the first five components with PCA and use a patch size of $5 \times 5 \times 5$. In both cases, the number n of patches for the search is set to 16. The similarity parameter h is set to 10. The results displayed in this article use the outputs from layers `conv1_1`, `conv2_1` and `conv3_1`

of VGG. These layers contain respectively 64, 128 or 256 feature maps.

4 Experiments

We analyzed the proposed anomaly detector using two different input image representations. The basic one, `pixels`, directly applies the anomaly detection procedure to the residuals obtained from the color channels. We also evaluated the results using as input three sets of feature channels extracted at different levels from the VGG network [57], namely, very low level (`conv1_1`), low level (`conv2_1`), and medium level (`conv3_1`) features.

We compared our method on two different sets of images. The first one is a group of sanity checks shown in Figure 5. These are synthetic images representing different types of anomaly (color, shape and density). We added a pure white noise image devoid of any structure where no anomaly (or saliency) should be detected. The second dataset, shown in Figure 6, consists of real images coming from various sources of challenging images in the literature. It highlights the performance and versatility of the proposed method for different problems and applications (saliency datasets, textile defects, radar imaging).
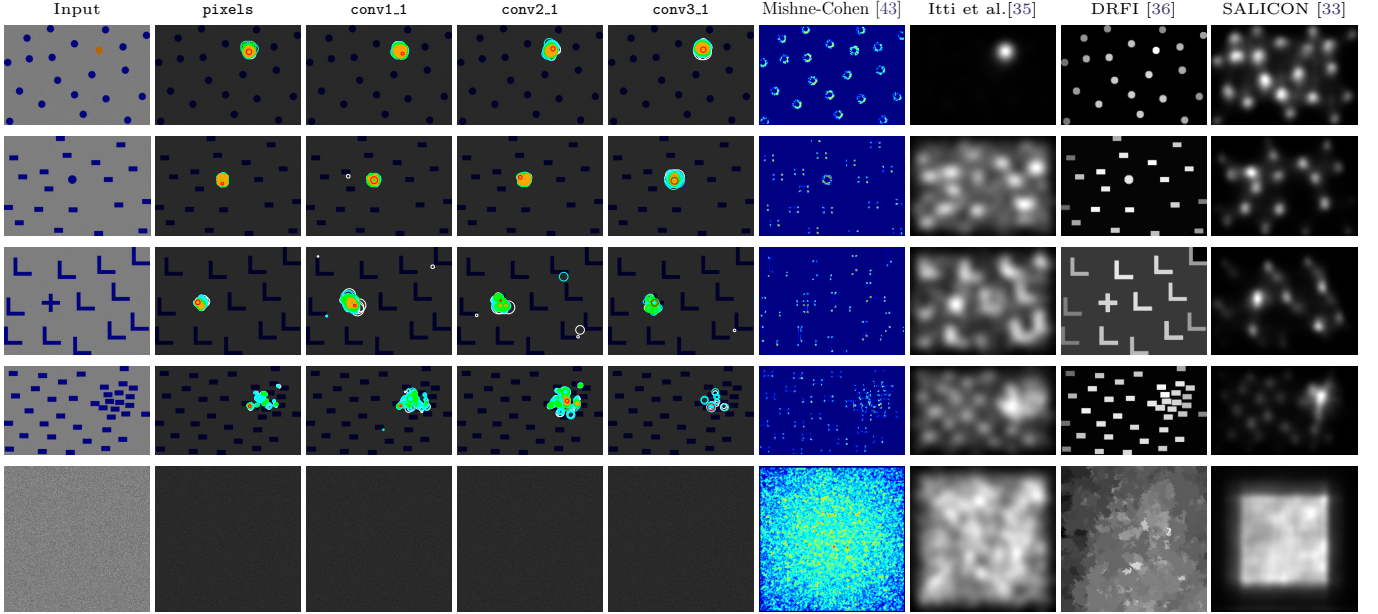

Figure 5: Detection results of our method when using as image representation directly the image pixels or the activation maps of the VGG neural Network at different layers (conv1_1, conv2_1, conv3_1) and a comparison to [43], [35], [36] and [33] on synthetic examples. Each detection is represented by a circle, where the circle radius represents the detection scale and the color the strength of the detection (NFA). White corresponds to a weak detection (NFA test value between 2×10^{-3} and 10^{-2}), cyan to a good detection (NFA between 4×10^{-8} and 2×10^{-3}), green to a very strong detection (NFA between 8.10^{-21} and 4×10^{-8}) and orange to an extremely salient detection (NFA smaller than 8×10^{-21}). Red corresponds to the detection with lowest NFA.

Anomaly or saliency detectors found in the literature are often tuned for specific applications, which probably explains the poor code availability. Moreover, only a portion of these detectors work without a training set of images. We compared to Mishne and Cohen [43], a state-of-the-art anomaly detector with available code. To complete our comparison, we compared to the salient object detector DRFI [36] (which is state-of-the-art according to [8]), and to the state-of-the-art human gaze predictor SALICON [33]. We also compared to the Itti et al. salient object detector [35], which we found to work reasonably well for anomaly detection. All methods produce saliency maps
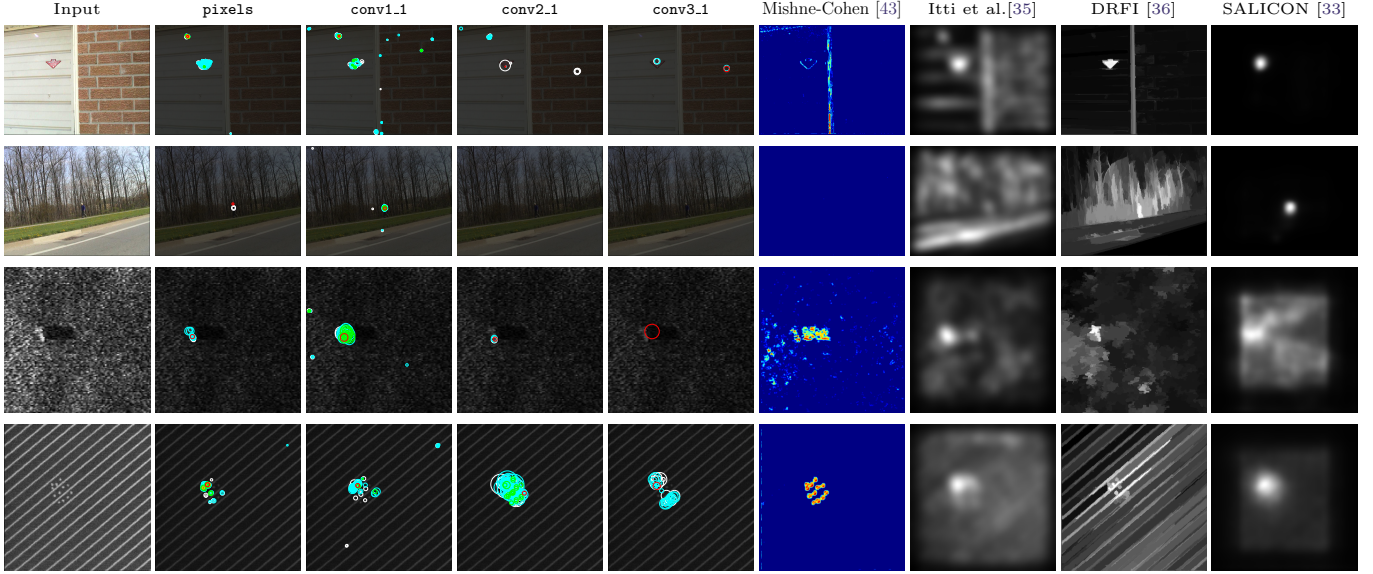
Figure 6: Detection results of our method when using as image representation directly the image pixels or the activation maps of the VGG neural Network at different layers (conv1_1, conv2_1, conv3_1) and a comparison to [43], [35], [36] and [33] on real examples. The first two images (top and second rows) are part of the Toronto dataset [10], while the third and fourth rows are from [43] and [62] respectively.

where anomalies have the highest score. Anomalies for Mishne and Cohen are red-colored, while the other methods don't have a threshold for anomalies. It is worth reminding that salient object detectors and human gaze saliency maps are not specifically designed to detect anomalies, which limits the comparison.

Synthetic images. Synthetic examples, shown in Figure 5, are used as sanity check. Contrary to real examples, it is easy to know where their anomaly lies. Our method performs well on all of them. Some weak false detections are found when using as input the different image representations at different layers of the VGG neural network. All the other compared methods miss some detections. SALICON successfully detects the anomalous density on the fourth example but in some cases misses anomalies or in others introduces numerous wrong saliencies. The Itti et al. method successfully detects the anomalous color in the first example, but fails to detect in the other ones. The Mishne and Cohen and DRFI methods do not perform well on any of the five synthetic examples.

Real images. On real images, as shown in Figure 6, the comparison is more intricate. Sometimes it requires looking in detail the images to find out whether detections made by the algorithm make sense. In the garage door example (top row), there are two detections that stand out (the lens flare and the red sign), some others – less visible – can be found (such as, scratches on the door or holes in the brick wall). For our method, the main detections are present with `pixels` and confirmed in `conv`s. There are also interesting anomalies that can be seen only at a given layer of the neural network. For example, `conv1_1` detects the holes in the brick wall and the gap between the garage door and the wall in addition to the usual ones (the ones detected with `pixels`). The variants `conv2_1` and `conv3_1` detect a missing part of a brick in the wall. Saliency methods detect the red sign but not the lens flare. The Mishne and Cohen method only detects the garage door gap at the bottom. The second example is a man walking in front of some trees. Our method detects the man with `pixels` and `conv1_1`. DRFI and SALICON detect this man while Mishne and Cohen and Itti et al. fail. The third example is a radar image showing a mine and the last example is a defect in a periodic textile. All methods detect the anomalies, with more or less precision. Note that the

detection in the top right corner for both `pixels` and `conv1_1` (and only these) seems to correspond to a defect inside the periodic pattern. More examples are available in [24].

Comparison to the a contrario method of Grosjean and Moisan [30]. This a contrario method is designed to detect spots in colored noise textures, and was applied to the detection of tumors in mammographies. This detection algorithm is the only other one computing NFAs, and we can directly compare them to ours. The detection results on a real mammography (having a tumor) are shown in Figure 7. With our method the tumor is detected with a very strong NFA of 10^{-12} whereas in [30] it has an NFA of 0.15 and is therefore weakly significant. Our self-similar anomaly detection method shows fewer false detections, actually corresponding to rare events like the crossings of arteries.
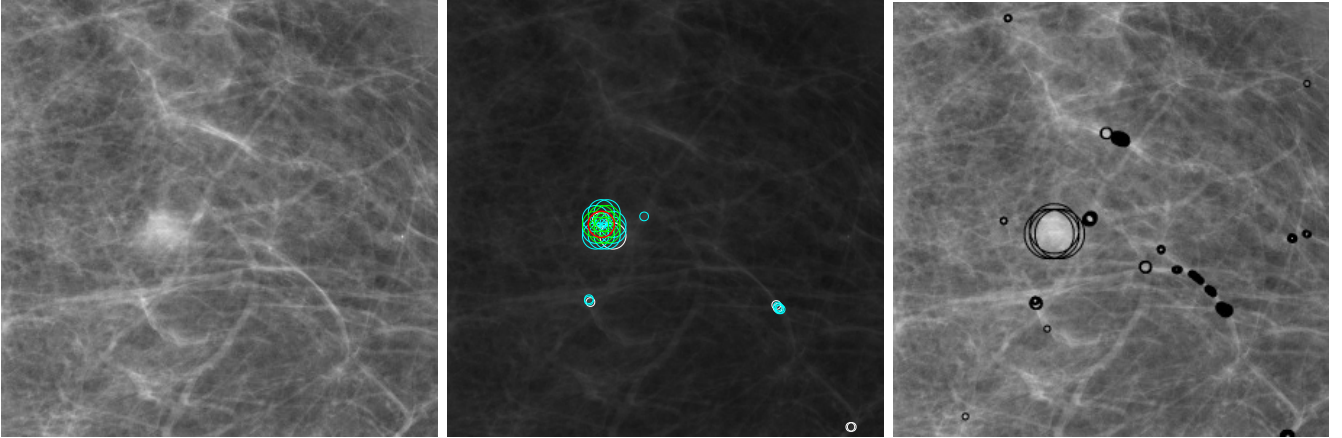

Figure 7: The region represented by the large white spot in the left image is a tumor. The proposed self-similarity anomaly detector successfully detects the tumor with a much significant NFA than the one from Grosjean and Moisan [30] (an NFA of 10^{-12} versus their reported NFA of 0.15), while making fewer false detections.

Online demo. The online demo uses the parameters presented in Section 3.4. Nevertheless it is possible to choose a different patch size, number of patches used or NFA threshold. It can also be used with a different reference even though all experiments shown in this article use the same image as reference. Since the method is quite slow the neural network part is not available for the online. Moreover if the images used are too big, they are automatically rescaled to fit in a 512×512 image.

5   Conclusions

We shall now address the objections that come to mind.

Is the method new? We have listed several saliency or anomaly detection methods based on rarity (sparsity, lack of similar patches, etc.) Thus involving sparsity or self-similarity is not new. The novelty of the method seems to be that it builds a new image, the residual, where the self-similar structure has been eliminated. As we have seen, using the a contrario framework is not new, but its use was restricted to detection in Gaussian colored noise.

Is the comparison method fair/correct? Only a very few implementations are available for anomaly detectors, that are mainly specific. Hence we compared with the few available and with saliency detectors. Even if the saliency map is *not* a detector, it can be considered as the last step

before applying a meaningfulness threshold. Also, this is why we used sanity check images that can play the role of mires that any method should resolve. (But in fact many don't).

Is the residual really stationary noise? There is a guarantee that the residual is noise, in the sense that it has lost all self-similarity. But this noise might not be stationary, which would lead to detection misses. Indeed, as a toy example assume that an image is composed of two textures, one very contrasted, and the other not. Then, the residual will have higher variance in the contrasted part. If the anomaly lies in the non-contrasted region, it might be missed because the (global) noise variance is overestimated (for this region). This can only be solved by localizing the detection, namely estimating the noise variance more locally, or equalizing its variance to make it stationary. This remains to be investigated.

Redundant detections? We have shown the detector performance either on the color channels or on VGG feature channels. Observe that these detections can be fused by a mere union, as they are all meaningful. They are, as we saw, redundant and were shown to illustrate the independence of the method from the chosen channel.

Extensions. An extension to video is highly desirable but requires a computationally intensive implementation to perform nonlocal space-time patch comparison.

Acknowledgment

Image Credits

Figures 1, 2, 3, 4, 5 are from the authors. Figure 6 is from Toronto dataset [10] (first two), [43] (third) and [62] (fourth). Figure 7 is from [30].

References

[1] A. ADLER, M. ELAD, Y. HEL-OR, AND E. RIVLIN, *Sparse coding with anomaly detection*, Journal of Signal Processing Systems, 79 (2015), pp. 179–188. https://doi.org/10.1007/s11265-014-0913-0.

[2] D. AIGER AND H. TALBOT, *The phase only transform for unsupervised surface defect detection*, in IEEE Conference on Computer Vision and Pattern Recognition, 2010. https://doi.org/10.1109/CVPR.2010.5540198.

[3] J. AN, *Variational Autoencoder based Anomaly Detection using Reconstruction Probability*, CoRR, (2016).

[4] F.J. ANSCOMBE, *The transformation of Poisson, binomial and negative-binomial data*, Biometrika, 35 (1948), pp. 246–254. https://doi.org/10.2307/2332343.

[5] C. Barnes, E. Shechtman, A. Finkelstein, and D.B. Goldman, *PatchMatch: A randomized correspondence algorithm for structural image editing*, ACM Transactions on Graphics (ToG), 28 (2009), p. 24. https://doi.org/10.1145/1531326.1531330.

[6] O. Boiman and M. Irani, *Detecting irregularities in images and in video*, International Journal of Computer Vision, 74 (2007), pp. 17–31. https://doi.org/10.1007/s11263-006-0009-9.

[7] G. Boracchi, D. Carrera, and B. Wohlberg, *Novelty detection in images by sparse representations*, in Intelligent Embedded Systems (IES), 2014. https://doi.org/10.1109/INTELES.2014.7008985.

[8] A. Borji, M-M. Cheng, H. Jiang, and J. Li, *Salient object detection: A benchmark*, IEEE Transactions on Image Processing, 24 (2015), pp. 5706–5722. https://doi.org/10.1109/TIP.2015.2487833.

[9] A. Borji and L. Itti, *Exploiting local and global patch rarities for saliency detection*, in IEEE Conference on Computer Vision and Pattern Recognition, 2012. https://doi.org/10.1109/CVPR.2012.6247711.

[10] N. Bruce and J. Tsotsos, *Saliency based on information maximization*, in Advances in Neural Information Processing Systems, 2006.

[11] A. Buades, B. Coll, and J-M. Morel, *A non-local algorithm for image denoising*, in IEEE Conference on Computer Vision and Pattern Recognition, 2005. https://doi.org/10.1109/CVPR.2005.38.

[12] ——, *Nonlocal image and movie denoising*, International Journal of Computer Vision, 76 (2008), pp. 123–139. https://doi.org/10.1007/s11263-007-0052-1.

[13] D. Carrera, G. Boracchi, A. Foi, and B. Wohlberg, *Detecting anomalous structures by convolutional sparse models*, in International Joint Conference on Neural Networks, 2015. https://doi.org/10.1109/IJCNN.2015.7280790.

[14] V. Chandola, A. Banerjee, and V. Kumar, *Anomaly detection: A survey*, ACM computing surveys (CSUR), 41 (2009), p. 15. https://doi.org/10.1145/1541880.1541882.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse 3-D transform-domain collaborative filtering*, IEEE Transactions on Image Processing, 16 (2007), pp. 2080–2095. https://doi.org/10.1109/TIP.2007.901238.

[16] A. Davy, T. Ehret, J-M. Morel, and M. Delbracio, *Reducing anomaly detection in images to detection in noise*, in IEEE International Conference on Image Processing (ICIP), IEEE, 2018, pp. 1058–1062. https://doi.org/10.1109/ICIP.2018.8451059.

[17] J. Deng, W. Dong, R. Socher, L-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, in IEEE Conference on Computer Vision and Pattern Recognition, 2009. https://doi.org/10.1109/CVPR.2009.5206848.

[18] A. Desolneux, L. Moisan, and J-M. Morel, *From gestalt theory to image analysis: a probabilistic approach*, vol. 34, Springer Science & Business Media, 2007. https://doi.org/10.1007/978-0-387-74378-3.

[19] X. Ding, Y. Li, A. Belatreche, and L.P. Maguire, *An experimental evaluation of novelty detection methods*, Neurocomputing, 135 (2014), pp. 313–327. https://doi.org/10.1016/j.neucom.2013.12.002.

[20] B. Du and L. Zhang, *Random-selection-based anomaly detector for hyperspectral imagery*, IEEE Transactions on Geoscience and Remote sensing, 49 (2011), pp. 1578–1589. https://doi.org/10.1109/TGRS.2010.2081677.

[21] A.A. Efros and T.K. Leung, *Texture synthesis by non-parametric sampling*, in International Conference on Computer Vision, 1999. https://doi.org/10.1109/ICCV.1999.790383.

[22] T. Ehret and P. Arias, *On the convergence of PatchMatch and its variants*, in EEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1121–1129. https://doi.org/10.1109/CVPR.2018.00123.

[23] T. Ehret, P. Arias, and J-M. Morel, *Global patch search boosts video denoising*, in International Conference on Computer Vision Theory and Applications, vol. 5, SCITEPRESS, 2017, pp. 124–134. https://doi.org/10.5220/0006175601240134.

[24] T. Ehret, A. Davy, J-M. Morel, and M. Delbracio, *Image Anomalies: A Review and Synthesis of Detection Methods*, Journal of Mathematical Imaging and Vision, (2019). https://doi.org/10.1007/s10851-019-00885-0.

[25] E. Elhamifar, G. Sapiro, and R. Vidal, *See all by looking at a few: Sparse modeling for finding representative objects*, in IEEE Conference on Computer Vision and Pattern Recognition, 2012. https://doi.org/10.1109/CVPR.2012.6247852.

[26] D. Gao, V. Mahadevan, and N. Vasconcelos, *The discriminant center-surround hypothesis for bottom-up saliency*, in Advances in neural information processing systems, 2008, pp. 497–504.

[27] S. Goferman, L. Zelnik-Manor, and A. Tal, *Context-aware saliency detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 34 (2012), pp. 1915–1926. https://doi.org/10.1109/TPAMI.2011.272.

[28] A. Goldman and I. Cohen, *Anomaly detection based on an iterative local statistics approach*, Signal Processing, 84 (2004), pp. 1225–1229. https://doi.org/10.1016/j.sigpro.2004.04.004.

[29] R. Grompone Von Gioi, J. Jakubowicz, J-M. Morel, and G. Randall, *LSD: A fast line segment detector with a false detection control*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 32 (2010), pp. 722–732. https://doi.org/10.1109/TPAMI.2008.300.

[30] B. Grosjean and L. Moisan, *A-contrario detectability of spots in textured backgrounds*, Journal of Mathematical Imaging and Vision, 33 (2009), pp. 313–337. https://doi.org/10.1007/s10851-008-0111-4.

[31] S. Hawkins, H. He, G. Williams, and R. Baxter, *Outlier detection using replicator neural networks*, in DaWaK, 2002. https://doi.org/10.1007/3-540-46145-0_17.

[32] T. Honda and S.K. Nayar, *Finding "anomalies" in an arbitrary image*, in International Conference on Computer Vision, 2001. https://doi.org/10.1109/ICCV.2001.937669.

[33] X. Huang, C. Shen, X. Boix, and Q. Zhao, *Salicon: Reducing the semantic gap in saliency prediction by adapting deep neural networks*, in International Conference on Computer Vision, 2015. https://doi.org/10.1109/ICCV.2015.38.

[34] L. Itti and C. Koch, *A saliency-based search mechanism for overt and covert shifts of visual attention*, Vision Research, 40 (2000), pp. 1489–1506. https://doi.org/10.1016/S0042-6989(99)00163-7.

[35] L. Itti, C. Koch, and E. Niebur, *A model of saliency-based visual attention for rapid scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 20 (1998), pp. 1254–1259. https://doi.org/10.1109/34.730558.

[36] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, *Salient object detection: A discriminative regional feature integration approach*, in IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 2083–2090. https://doi.org/10.1109/CVPR.2013.271.

[37] A. Kumar, *Neural network based detection of local textile defects*, Pattern Recognition, 36 (2003), pp. 1645–1659. https://doi.org/10.1016/S0031-3203(03)00005-0.

[38] J. Lezama, R. Grompone von Gioi, G. Randall, and J-M. Morel, *Finding vanishing points via point alignments in image primal and dual domains*, in IEEE Conference on Computer Vision and Pattern Recognition, 2014. https://doi.org/10.1109/CVPR.2014.72.

[39] D. Lowe, *Perceptual organization and visual recognition*, Kluwer Academic Publishers, 1985. https://doi.org/10.1007/978-1-4613-2551-2.

[40] D.G. Lowe, *Distinctive image features from scale-invariant keypoints*, International Journal of Computer Vision, 60 (2004), pp. 91–110. https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[41] R. Margolin, A. Tal, and L. Zelnik-Manor, *What makes a patch distinct?*, in IEEE Conference on Computer Vision and Pattern Recognition, 2013. https://doi.org/10.1109/CVPR.2013.151.

[42] M. Markou and S. Singh, *Novelty detection: a review –part 1: statistical approaches*, Signal Processing, 83 (2003), pp. 2481–2497. https://doi.org/10.1016/j.sigpro.2003.07.018.

[43] G. Mishne and I. Cohen, *Multiscale anomaly detection using diffusion maps*, IEEE Journal of Selected topics in Signal Processing, 7 (2013), pp. 111–123. https://doi.org/10.1109/JSTSP.2012.2232279.

[44] ——, *Multiscale anomaly detection using diffusion maps and saliency score*, in IEEE International Conference on Acoustics, Speech and Signal Processing, 2014. https://doi.org/10.1109/ICASSP.2014.6854115.

[45] L. Moisan and B. Stival, *A probabilistic criterion to detect rigid point matches between two images and estimate the fundamental matrix*, International Journal of Computer Vision, 57 (2004), pp. 201–218. https://doi.org/10.1023/B:VISI.0000013094.38752.54.

[46] N. Murray, M. Vanrell, X. Otazu, and C A. Parraga, *Saliency estimation using a non-parametric low-level vision model*, in IEEE Conference on Computer Vision and Pattern Recognition, 2011. https://doi.org/10.1109/CVPR.2011.5995506.

[47] S.M. OMOHUNDRO, *Five balltree construction algorithms*, International Computer Science Institute Berkeley, 1989.

[48] V. PATRAUCEAN, R. GROMPONE VON GIOI, AND M. OVSJANIKOV, *Detection of mirror-symmetric image patches*, in IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2013. https://doi.org/10.1109/CVPRW.2013.38.

[49] V. PATRAUCEAN, P. GURDJOS, AND R. GROMPONE VON GIOI, *A parameterless ellipse and line segment detector with enhanced ellipse fitting*, in Europeana Conference on Computer Vision, 2012. https://doi.org/10.1007/978-3-642-33709-3_41.

[50] D-B. PERNG, S-H. CHEN, AND Y-S. CHANG, *A novel internal thread defect auto-inspection system*, International Journal of Advanced Manufacturing Technology, 47 (2010), pp. 731–743. https://doi.org/10.1007/s00170-009-2211-8.

[51] N. PIERAZZO, J-M. MOREL, AND G. FACCIOLO, *Multi-Scale DCT Denoising*, Image Processing On Line, 7 (2017), pp. 288–308. https://doi.org/10.5201/ipol.2017.201.

[52] M.A.F. PIMENTEL, D.A. CLIFTON, L. CLIFTON, AND L. TARASSENKO, *A review of novelty detection*, Signal Processing, 99 (2014), pp. 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026.

[53] L. RAAD, A. DAVY, A. DESOLNEUX, AND J-M. MOREL, *A survey of exemplar-based texture synthesis*, Annals of Mathematical Sciences and Applications, 3 (2018), pp. 89–148. https://doi.org/10.4310/AMSA.2018.v3.n1.a4.

[54] N. RICHE, M. MANCAS, M. DUVINAGE, M. MIBULUMUKINI, B. GOSSELIN, AND T. DUTOIT, *RARE2012: A multi-scale rarity-based saliency detection with its comparative statistical analysis*, Signal Processing: Image Communication, 28 (2013), pp. 642–658. https://doi.org/10.1016/j.image.2013.03.009.

[55] T. SCHLEGL, P. SEEBÖCK, S.M. WALDSTEIN, U. SCHMIDT-ERFURTH, AND G. LANGS, *Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery*, in International Conference on Information Processing in Medical Imaging, 2017. https://doi.org/10.1007/978-3-319-59050-9_12.

[56] H.J. SEO AND P. MILANFAR, *Static and space-time visual saliency detection by self-resemblance*, Journal of vision, 9 (2009), pp. 15–15. https://doi.org/10.1167/9.12.15.

[57] K. SIMONYAN AND A. ZISSERMAN, *Very deep convolutional networks for large-scale image recognition*, 2014.

[58] L. TARASSENKO, P. HAYTON, N. CERNEAZ, AND M. BRADY, *Novelty detection for the identification of masses in mammograms*, in International Conference on Artificial Neural Networks, 1995. https://doi.org/10.1049/cp:19950597.

[59] H.R. TAVAKOLI, E. RAHTU, AND J. HEIKKILÄ, *Fast and efficient saliency detection using sparse sampling and kernel density estimation*, in Scandinavian Conference on Image Analysis, 2011. https://doi.org/10.1007/978-3-642-21227-7_62.

[60] D.M.J. TAX AND R.P.W. DUIN, *Outlier detection using classifier instability*, in Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR), 1998. https://doi.org/10.1007/BFb0033283.

[61] C. TOMASI AND R. MANDUCHI, *Bilateral filtering for gray and color images*, in International Conference on Computer Vision, 1998. https://doi.org/10.1109/ICCV.1998.710815.

[62] D-M. TSAI AND C-Y. HSIEH, *Automated surface inspection for directional textures*, Image and Vision Computing, 18 (1999), pp. 49–62. https://doi.org/10.1016/S0262-8856(99)00009-8.

[63] D-M. TSAI AND T-Y. HUANG, *Automated surface inspection for statistical textures*, Image and Vision computing, 21 (2003), pp. 307–323. https://doi.org/10.1016/S0262-8856(03)00007-6.

[64] X. XIE AND M. MIRMEHDI, *TEXEMS: Texture exemplars for defect detection on random textured surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 29 (2007), pp. 1454–1464. https://doi.org/10.1109/TPAMI.2007.1038.

[65] Z. ZHA, X. LIU, Z. ZHOU, X. HUANG, J. SHI, Z. SHANG, L. TANG, Y. BAI, Q. WANG, AND X. ZHANG, *Image denoising via group sparsity residual constraint*, in IEEE International Conference on Acoustics, Speech and Signal Processing, 2017. https://doi.org/10.1109/ICASSP.2017.7952464.

[66] M. ZONTAK AND I. COHEN, *Defect detection in patterned wafers using anisotropic kernels*, Machine Vision and Applications, 21 (2010), pp. 129–141. https://doi.org/10.1007/s00138-008-0146-y.