GLAD: A Global-to-Local Anomaly Detector

Aitor Artola[1,2], Yannis Kolodziej[2], Jean-Michel Morel[1], Thibaud Ehret[1]

[1]Université Paris-Saclay, CNRS, ENS Paris-Saclay, Centre Borelli, France
[2]Visionairy

aitor.artola@ens-paris-saclay.fr

Abstract

Learning to detect automatic anomalies in production plants remains a machine learning challenge. Since anomalies by definition cannot be learned, their detection must rely on a very accurate "normality model". To this aim, we introduce here a global-to-local Gaussian model for neural network features, learned from a set of normal images. This probabilistic model enables unsupervised anomaly detection. A global Gaussian mixture model of the features is first learned using all available features from normal data. This global Gaussian mixture model is then localized by an adaptation of the K-MLE algorithm, which learns a spatial weight map for each Gaussian. These weights are then used instead of the mixture weights to detect anomalies. This method enables precise modeling of complex data, even with limited data. Applied on WideResnet50-2 features, our approach outperforms the previous state of the art on the MVTec dataset, particularly on the object category. It is robust to perturbations that are frequent in production lines, such as imperfect alignment, and is on par in terms of memory and computation time with the previous state of the art.

1. Introduction

Anomaly detection in images consists in finding images, or image regions, that do not conform with the rest of the data. This is an important problem in many industrial, medical or biological applications. Anomaly detection is an effortless natural task for humans, who reach very good detection rates using even a single reference. This explains why quality control in production lines has historically been left to human operators. Yet, automatizing the process can accelerate, reduce the production cost, and smooth out the performance variations caused by the operators' fatigue [37, 27]. The problem remains, however, a challenge for computer vision, as there is no clear and straightforward definition of normality in arbitrary data.

Unsupervised anomaly detection for industrial applications is receiving much attention lately, especially after the publication of a new reference dataset by MvTec [3]. The "unsupervised" requirement is challenging, but might lead to a truly general solution. It is generally acknowledged that anomaly detection is not a classic classification problem [34]. Indeed, anomalies do not form well defined classes. They can be rare or have no definite patterns. New types of anomalies can actually appear later, but should be correctly handled based on past experience of normal data. This makes annotating relevant data inherently impossible. On the contrary, normal data are abundant and can be used to create accurate models. This leads to consider anomaly detection as an out-of-distribution detection problem.

In this work, we therefore focus on the modeling of normal data. We propose a global-local model that extends Padim [12] and combines the flexibility of precise local modeling with the robustness of a global Gaussian mixture model. Our model emulates a Gaussian mixture at each position without suffering from the curse of dimensionality when only few data are available. Thanks to the sparsity of the model, it is as efficient as previously proposed simpler models, and it achieves new state of the art on the MVTec dataset [3].

2. Related work

Anomaly detection has been extensively reviewed in the literature [15, 26, 33]. Methods can be classified into three main categories: methods anterior to deep learning, methods based on pre-trained networks and finally methods purely based on neural networks.

Methods anterior to deep learning. They focus on modeling normal data, also referred to as the background model. Homogeneous and stochastic methods [14, 41, 38, 39] suppose that the background model follows a known distribution. Center-surround methods [19, 18, 28] model anomalies as local events contrasting their immediate surround-

ings. Sparsity-based methods [24, 4, 7, 8] learn a sparse dictionary representing normal data. Anomalies are defined as data that don't verify the sparsity condition. Non-local methods [47, 11] assume that each image patch in normal data belongs to a dense cluster in the image's patch space. Anomalies instead occur far from their closest neighbors. Anomalies are measured by clustering image patches, which leads directly to a rarity measurement. The method by Davy *et al*. [11] bridges the gap with methods using pre-trained networks, as their method can be applied directly on images patches or on neural network features.

Using pre-trained networks. These methods use neural networks to integrate semantic information into the detection process. SPADE [10] creates a feature base from a pre-trained neural network with the reference images and uses the L_2 norm between the features of an image and its kNN of the reference base as a score. It involves features at different layers to perform a multi-scale analysis. PatchCore [30] improves on this method by a adding a coreset subsampling and a preprocessing to the feature library. MahaAD [29] models each layer as a single global Gaussian model. Anomalies are then detected by computing and thresholding the Mahalanobis distance to this Gaussian. PaDim [12] extends MahaAD [29] by learning a Gaussian model per position instead of globally.

Deep learning based. Using a variational auto-encoder (VAE), [40] learn a representation of normal data. In order to localize the anomaly, the Grad-cam attention technique [36] is used: when the image is not an anomaly the attention should be uniform over the entire image, this changes when the image contains an anomaly. A weakly supervised version is also proposed. In [20], anomalies are found based on the distribution of the gradients. Liu *et al*. [23] proposes a similar type of attention technique to detect and localize the anomaly. In [32], a deep neural network is trained by minimizing the volume of a hypersphere that encloses the network representations of the data. Yi and Yoon [42] suggest to use the patches of the image instead. The anomaly score is merely the distance between the encoded patch and its nearest neighbor. A P-style network is trained in [46]. It learns to produce a structure information from the image. The authors show that this information usually contains the anomaly and also helps during training when reconstructing the image. Li *et al*. [21] use a neural architecture search to find better deep learning architectures. DRÆM [44] learns both a reconstruction model and a discriminative model using simulated anomalies. Finally, newest methods [31, 16] are based on normalizing flows. Normalizing flows are invertible neural networks that learn a model that transforms data into a simpler, usually Gaussian, distribution.

3. Global-local Gaussian modeling of neural network features for anomaly detection

With Padim [12], Defard *et al*. proposed to learn a single multivariate Gaussian per pixel in a neural network feature space. Although they proposed to reduce the dimension of the features using PCA or random selection, the problem is that the dimension is still too large for most use cases. Indeed, the proposed dimension $d = 100$ or $d = 550$ would still require at least that many samples to estimate a non-constrained covariance matrix. This is not always possible in the MVTec dataset [3]. Moreover, precious information is arguably lost by dimension reduction. Another limitation of this model is the use of a single Gaussian at a given position. Indeed, such a model might be too restrictive since it cannot model multimodal distributions.

Inspired from image denoising, we suggest taking advantage of object redundancy to learn a more accurate and more expressive model than a single Gaussian, even under data constraints. Indeed, to mitigate the lack of data, non-local methods such as [6] take advantage of the redundancy, also called self-similarity, in natural images to estimate models of clean data. This is why we propose the global-local model summarized in Figure 1 which we introduce next.

3.1. Learning a robust global model

Zoran and Weiss [48] have shown that a Gaussian mixture can faithfully represent the entire space of patches from natural images. In this work, we develop the use of a Gaussian mixture as global model for the set of features extracted from a neural network applied to a set of sample images. In the following, $\Theta = (\pi_k, \mu_k, \Sigma_k)_{k=1,\cdots K}$ is used to refer to a Gaussian mixture model with K components such that for a given k, (μ_k, Σ_k) define a Gaussian with mean μ_k and covariance Σ_k. The weight of this Gaussian inside the model is represented by π_k.

Traditionally, Gaussian mixtures are learned using the expectation-maximization algorithm (EM). Unfortunately, this algorithm becomes very slow when applied with many Gaussians to high dimensional samples. This is why we chose the alternative algorithm K-MLE [25], which works for mixtures of exponential laws. This algorithm is analogous to K-means and generalizes by using the properties of exponential laws. It is faster because it assigns each sample u_i to a single Gaussian with index z_i and therefore does not involve all the samples in the computation of every parameter. This leads to a memory complexity of $O(N)$ for K-MLE instead of $O(KN)$ for EM. Given a set of data $(u_i)_{i=1,\cdots,N}$ in dimension d, its iterative attribution steps write

$$z_i^{t+1} = \underset{k\in[\![1,K]\!]}{\arg\min} (u_i - \mu_k^t)^T {\Sigma_k^t}^{-1} (u_i - \mu_k^t) + \log|\Sigma_k^t| + 2\log\pi_k^t. \tag{1}$$

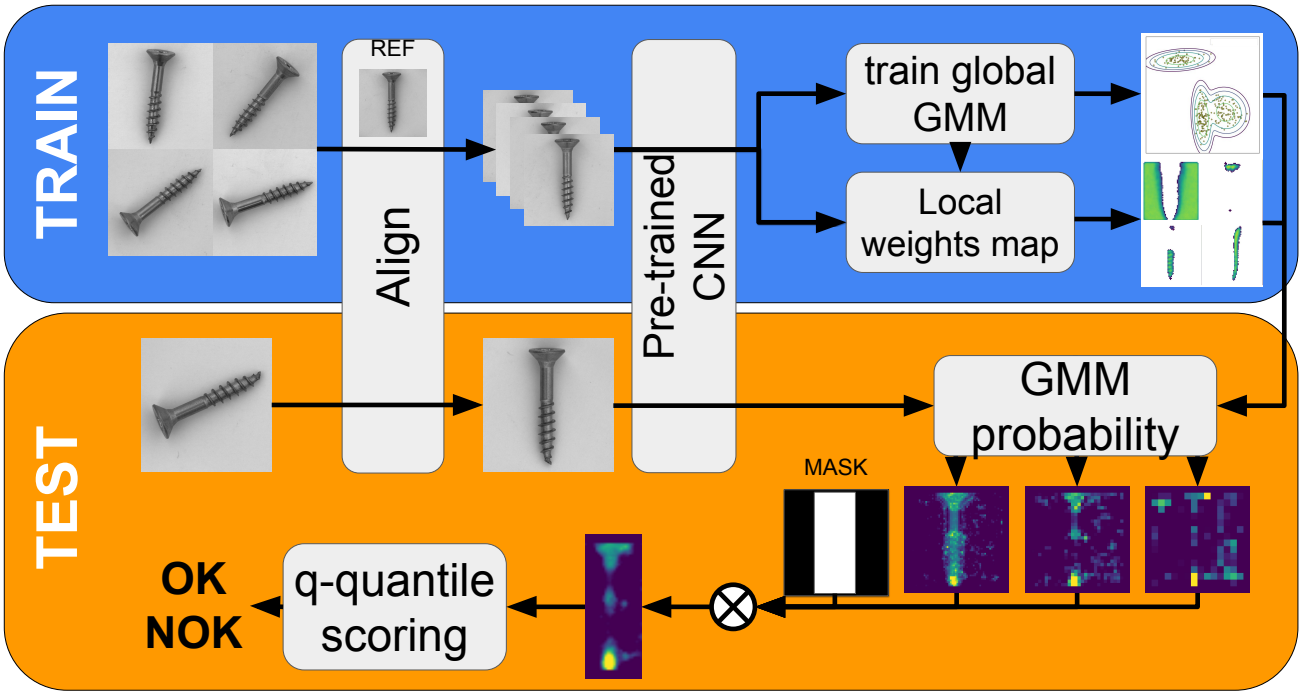

Figure 1. Proposed pipeline for unsupervised anomaly detection. After alignment on a reference, all images of the class are given to a pre-trained network where features at different level are extracted. This creates an embedding of each image. A global-local model comprised of a global Gaussian mixture and a local weight map is then trained from these features. During testing, the probability of appearance of a feature at a specific position is estimated using the global-local model. A global score and a decision are derived from the probability map.

Then, considering the sets $\mathcal{C}_k^{t+1} = \{u_i | z_i = k\}$, the algorithm separately computes the empirical parameters of each Gaussian cluster by

$$\pi_k^{t+1} = \frac{|\mathcal{C}_k^{t+1}|}{N},\ \mu_k^{t+1} = \frac{\sum_{u \in \mathcal{C}_k^{t+1}} u}{|\mathcal{C}_k|} \tag{2}$$

$$\text{and } \Sigma_k^{t+1} = \frac{\sum_{u \in \mathcal{C}_k^{t+1}} (u - \mu_k^{t+1})(u - \mu_k^{t+1})^T}{|\mathcal{C}_k^{t+1}|}. \tag{3}$$

Gaussians with no samples are dropped.

We observed that this algorithm yielded less blurry Gaussians than EM. The intuition is that, since EM performs weighted averages where all Gaussians intervene for each sample, all samples (even those further away from a specific Gaussian) contribute to the estimation. This good property *per se* is an obstacle when modeling the tail of the distribution. Indeed, there are inherently few data available to model the tail. Thus, in EM, the Gaussians, and especially their covariance matrices, are easily led to overfit the data on the tail, thus frustrating outlier detection. Traditionally, covariance matrices are estimated using the sample covariance estimator, which maximizes the likelihood. However, for a large dimension d and a small number of samples N, this estimator is very unstable and tends to overestimate large eigenvalues and to underestimate small ones. The produced covariance may also be degenerate. This traditionally leads to adding ϵI_d with $\epsilon > 0$ to the estimated covariance to ensure its positive-definiteness. This regularization, however, is not sufficient to stabilize the covariances when $N \ll d$. Hence, we opted for a shrinkage regularizer [9] which takes the form of a convex combination of the empirical covariance of the samples $\hat{S}$ and of the average of its eigenvalues multiplied by the identity $\hat{F} = \frac{\mathrm{Tr}(\hat{S})}{d} I_d$. The convex coefficient ρ is chosen to minimize the expectation of the MSE between the theoretical covariance Σ and the regularized estimator $\hat{\Sigma}$:

$$\min_{\rho} \mathbb{E}\left[\|\Sigma - \hat{\Sigma}\|_F^2\right] \text{ such that } \hat{\Sigma} = (1-\rho)\hat{S} + \rho\hat{F}. \tag{4}$$

The optimal solution of the problem is the oracle ρ_o. However, it requires the knowledge of the theoretical covariance Σ being estimated,

$$\rho_o = \frac{\mathbb{E}\left[(\Sigma - \hat{S})(\hat{F} - \hat{S})\right]}{\mathbb{E}\left[\|\hat{\Sigma} - \hat{F}\|_F^2\right]} \tag{5}$$

$$= \frac{(1 - \frac{2}{d})\,\mathrm{Tr}(\Sigma^2) + \mathrm{Tr}^2(\Sigma)}{(N + 1 - \frac{2}{d})\,\mathrm{Tr}(\Sigma^2) + (1 - \frac{N}{d})\,\mathrm{Tr}^2(\Sigma)}. \tag{6}$$

We use the Oracle Approximating Shrinkage (OAS) [9] estimator which is better in terms of MSE when $N \ll d$. This

iterative estimator is defined by

$$\rho_j = \frac{\left(1-\frac{2}{d}\right)\operatorname{Tr}(\hat{\Sigma}_j\hat{S}) + \operatorname{Tr}^2(\hat{\Sigma}_j)}{\left(N+1-\frac{2}{d}\right)\operatorname{Tr}(\hat{\Sigma}_j\hat{S}) + \left(1-\frac{N}{d}\right)\operatorname{Tr}^2(\hat{\Sigma}_j)}, \quad (7)$$

$$\hat{\Sigma}_j = (1-\rho_j)\hat{S} + \rho_j\hat{F}. \quad (8)$$

and is shown to converge to an equivalent ρ_{OAS} defined by

$$\rho_{\text{OAS}} = \min\left(\frac{\left(1-\frac{2}{d}\right)\operatorname{Tr}(\hat{S}^2) + \operatorname{Tr}^2(\hat{S})}{\left(N+1-\frac{2}{d}\right)\left[\operatorname{Tr}(\hat{S}^2) - \frac{\operatorname{Tr}^2(\hat{S})}{d}\right]}, 1\right). \quad (9)$$

3.2. Back to a local model

At this stage, the Gaussian mixture model is global, and the probability of appearance of each feature does not depend on its position. This simplification is excessive. Obviously, a feature can be normal at a position and anomalous at another. This is why we propose a method to add back localization information to the model, by taking advantage of a preliminary alignment of the tested objects.

We have already seen that it is not possible to learn a specific Gaussian mixture per position due to lack of data. This would anyway lead to exceedingly big and slow models. Another option could be to learn a local mixture in a window instead of a specific position. However, this raises the impractical issue of fixing the size of the window and a variable number K of Gaussians at each position, depending on whether the image contains varying details or is merely uniform at the considered position. To avoid an excessively local analysis, we therefore opted to keep the (global) parameters (μ_k, Σ_k) but to deduce position dependent mixing weights $\pi_k(\mathbf{x})$ from the Gaussian mixture.

We chose to adapt the K-MLE algorithm to learn the local weight map while keeping covariances matrices fixed. Let $u_m(\mathbf{x})$ denote the feature sample vector of image m at position $\mathbf{x}$, this yields

$$z_m^{t+1}(\mathbf{x}) = \underset{k\in[\![1,N]\!]}{\arg\min}(u_m(\mathbf{x})-\mu_k)^T\Sigma_k^{-1}(u_m(\mathbf{x})-\mu_k) + \log|\Sigma_k| + 2\log\pi_k^t(\mathbf{x}), \quad (10)$$

$$\mathcal{C}_k^{t+1}(\mathbf{x}) = \{u_m(\mathbf{x}) | z_m^{t+1}(\mathbf{x}) = k,\ \forall m\}, \quad (11)$$

$$\pi_k^{t+1}(\mathbf{x}) = \frac{|\mathcal{C}_k^{t+1}(\mathbf{x})|}{N}. \quad (12)$$

To obtain more samples per position and reduce the number of false alarms caused by small deformations or slight positional variations of the objects, we perform the computation of the weights at $\mathbf{x}$ by using the samples in a small circle of radius r centered at $\mathbf{x}$.

This local weight map has two interesting properties. Firstly, it models each position with a different mixture. The second one is that it turns out to be mostly sparse. This is because K-MLE uses only relevant samples to estimate the weights instead of using all available samples like with EM. The advantage of this sparsity is a faster probability inference.

3.3. Anomaly detection with a global-local model

Traditionally, samples are assigned to the closest Gaussian to detect anomalies. The Mahalanobis distance to this Gaussian is then measured. However, this attribution implies ignoring the interactions between Gaussians. Yet, multivariate Gaussians in a mixture can have strong interaction and local maxima differing from the Gaussian centers. It therefore seems more appropriate to consider the mixture as a whole when analyzing a new sample.

Using the global-local model, we compute a probability for each feature vector of an image using the mixture, thus forming a probability map. State-of-the-art methods generally score the image using the worst value (*i.e.* the smallest probability) from this map but we found that the q-quantile is more robust with respect to false alarms. The proposed method can also be made multi-scale by taking features at different layers of a network. For this, we fitted a Gaussian mixture to each selected layer. We then aggregated the probability maps by taking their product, which corresponds to assuming independence of the layers.

4. Experiments

4.1. Implementation and training details

Similarly to previous methods such as [29] and [12], we use a pre-trained network to extract the features. Classification and segmentation networks trained on ImageNet [13] or Coco [22] are indeed able to learn universal features containing relevant semantic information for the task of anomaly detection. We selected Resnet18 [17] and WideResenet50-2 [43] as backbone for the experiments shown in this Section. We use the implementation proposed by Zhang [45], as it was shown to be more robust than traditional versions. All sample images were resized to 256×256 before applying the network. The features used to learn the model were extracted from three different layers of these networks. The layers, referred to as layers 1, 2 or 3 in the following, correspond to the layers following respectively the first, second and third pooling inside the network. They extract relevant features at different scales.

The training of the global mixture model was initialized with $K = 1000$ Gaussians. However, the final model was usually comprised of fewer Gaussians. As mentioned in Section 3.1, the OAS regularization leads to the removal of many Gaussians. We stopped the training when the relative variation of log likelihood between two iterations was less than 10^{-6}, or if the number of iterations exceeded 100. We used the same stopping conditions when training the local

Backbone	Mixture training	Local weighting	Layers used 1	2	3	OAS Shrink	Memory (GB)	Computation time (s)	AUROC
ResNet18	K-MLE		✓	✓	✓	✓	0.25	1.07	96.1
ResNet18	K-MLE	✓	✓	✓	✓	✓	0.28	0.29	98.1
WideResnet50	K-MLE	✓	✓	✓		✓	0.75	0.34	97.1
WideResnet50	K-MLE	✓		✓	✓	✓	2.90	0.43	98.9
WideResnet50	K-MLE	✓	✓		✓	✓	2.50	0.44	98.5
WideResnet50	EM	✓	✓	✓	✓	✓	3.48	1.65	98.9
WideResnet50	K-MLE		✓	✓	✓	✓	3.01	3.33	97.0
WideResnet50	K-MLE	✓	✓	✓	✓		5.24	0.92	98.1
WideResnet50	K-MLE	✓	✓	✓	✓	✓	3.04	0.61	99.1

Table 1. Ablative study of the main parameters of the proposed model. The other parameters are discussed in Section 4.2. Memory consumption and computation time are averaged over all MVTec objects. Reported times do not include the alignment (about 0.2s).

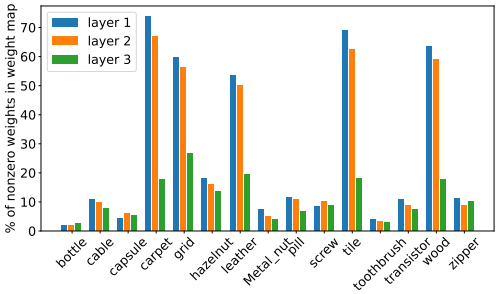

Figure 2. Sparsity of the learned local weight map on WideResNet50-2 features for layers 1, 2 and 3. On average, only a few Gaussians from the global model are necessary to model well a given position. This observation does not apply to textures.

weight map; empirically it never takes more than 10 iterations. The training of the global model generally also converged before the maximum iteration number was reached. Training the global methods takes between 20min to 2h depending on the layer and the object. We also point out that training a model with EM is more than 2.5 times slower than with K-MLE in the same conditions.

We show in Figure 3 an example of weight and local sparsity map learned on features from layer 1 of WideResnet50-2. The general structure of the object can be recognized both on the weight map and on the local sparsity map. Similar observations can be done on other objects and on some texture maps of the MVTec dataset. This confirms that each Gaussian describes a particular structural element of the image that can be found at several different positions. Conversely, the local sparsity map highlights the fact that there can be more than one relevant Gaussian per position.

For all objects of the MVTec dataset, we chose a single reference that is used to align all images from this category. As in [2], we aligned images using [5]. Contrary to the objects of the dataset, the textures were not aligned. This is because this could create boundary issues when the aligned texture stretches out of the boundary of the reference. It is also difficult to define what a proper alignment would mean for a texture anyway. We must avoid false detections outside the objects. Indeed, the background was not annotated for possible anomalies, and false detections caused by network boundary effects could occur. To that aim, we applied a crude masking to the output of the model. This means that anomalies were detected only in relevant areas. For the objects, our masks correspond roughly to the area occupied by the object (an example is shown in Figure 1). For the textures, we removed a margin of $1/16^{\text{th}}$ of the size of the image from each anomaly map. A radius $r = 1$ was used for the blur parameter of the weight map and we used the 0.5%-quantile to score anomalies. See the supplementary material for a study on the influence of the quantile.

4.2. Ablation study

We present in Table 1 an ablative study of most method parameters. We first looked at the backbone pre-trained network. Overall, we found out that Wide-ResNet50-2 performed better than ResNet18 by 1% but produced a larger model, requiring about ten times more memory and longer computation time. Indeed, WideResNet50-2 features have a larger dimension for each respective layer. Taking that into account, our model using a ResNet18 backbone brings a good compromise between performance and efficiency. We also compared with various combinations of WideResnet50-2 layers. There is no gain taking only 1+2 and 2+3 compared to ResNet18. The performances using layers 2+3 are almost the same as using 1+2+3, but the weight of the model is almost the same because the covariances of the last layer occupy the majority of the model's space. However, removing the first layer also removes its probability calculations, which decreases the inference time. Nevertheless, using all three layers (1+2+3) yielded the best results. It also shows that the model trained using K-MLE is more efficient than the one trained with EM.

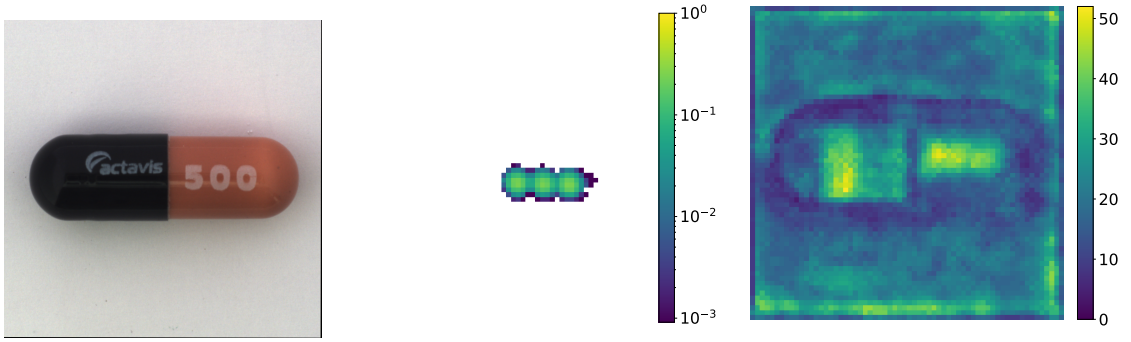

Figure 3. Example of a local weight map for capsule with WideResnet50-2 layer 1. From left to right: an image of the capsule, the local weight map corresponding to a single Gaussian and the number of non-zero weights per position for a Gaussian mixture with $K = 458$.

We also measured the impact of our localization step compared to simply using a global Gaussian mixture model. The localization has two benefits: one is in terms of performance and the other in terms of efficiency. We observed a difference of about 2% in the AUROC between the global model and our global-local model. This shows that localizing the information leads to better discrimination of anomalies from normal data. Logically indeed, a feature can be normal at a given spatial position while being anomalous at another. In terms of efficiency, the localization reduces the computation time by three to four at a very small memory cost. The small memory cost comes from the local weight maps, with a storage space proportional to the size of the image times the number of Gaussians that need to be stored. This map is actually sparse, because there are only few non-zero weights per location. This means that to evaluate the probability at a given location, it is not necessary to use all Gaussians of the model. Since estimating a probability from a Gaussian is the model's bottleneck, this reduction in the number of Gaussians being considered for each sample leads to a significant speedup. We illustrate this sparsity in Figure 2. It shows that about 10% only of the Gaussians are used for a given position on average. This observation does not apply to textures, where the same features can be found anywhere in the image with the same probability. Therefore we expect our model to slightly under-perform on these.

We then studied the stability of the training process when varying the random initialization. Doing this did not change the order of magnitude of the number K of learned Gaussians of a given object for a specific layer. The final number of Gaussians seems to match with the complexity of the object. Textures (except for grid), pill or zipper have all a low number of Gaussians, while bottle, likely due to its variability, requires a higher number. Figure 4 shows the decay of the weights π_k of the global model trained on the features of layer 1 of WideResnet50-2 for all classes of the MVTec dataset. With the exception of grid, all models converge to a smaller number of Gaussians than the initial $K = 1000$. We then verified that the number of initial Gaussians was

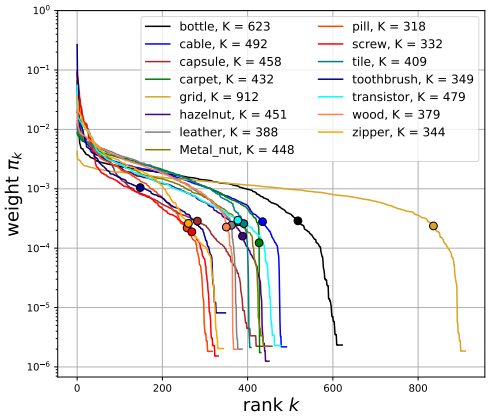

Figure 4. Decay of the global weights π_k in the global mixtures for the first layer ($d = 256$) of WideResNet50-2. The dot indicates the Gaussian from which $\pi_k < (d+1)/|\mathcal{C}_k|$. From that point the Gaussians' covariance matrices are degenerate. The exact number of Gaussians kept for each class is indicated in the top right. See the decays corresponding to the other layers in the supplementary material.

not limiting the modeling power of our method by learning a model initialized with $K = 2000$ on the grid class. This model converged to $K = 1506$ Gaussians, so more than the initial model allowed for, but this increase did not translate into better detection: the larger model yielded an AUROC of 98.8% compared the 98.7% of the original model.

We studied the impact of the OAS regularization. Without it, we observed that the number of Gaussians remained close to the initial number, thus requiring longer computation times and more memory. Moreover, as can be seen in Table 1, the performance dropped by 1%. This leads us to believe that the proposed regularization, which depends on d and $|\mathcal{C}_k^t|$, avoids overfitting and generalizes better.

We present additional ablation studies, such as more studies on the impact of the initial K, the impact of the alignment step on MVtech and the histogram of likelihood using either EM or K-MLE to train the mixture.

Backbone	EfficientNet		ResNet-18			WideResNet-50-2			Other	
	MahaAD [29]	PaDim [12]	PaDim [12]	CFlow-ad [16]	Ours	PaDim [12]	CFlow-ad [16]	Ours	DRAEM [44]	CS-Flow [31]
Carpet	93.7	-	98.4	98.2	97.6	98.4	98.7	**99.0**	97.0	**99.0**
Grid	**100**	-	89.8	99.0	98.2	89.8	99.6	98.7	99.9	**100**
Leather	**100**	-	98.8	100	**100**	98.8	**100**	**100**	**100**	**100**
Tile	99.6	-	95.9	98.4	98.8	95.9	99.9	99.6	99.6	**100**
Wood	99.3	-	99.0	98.6	97.1	99.0	99.1	98.9	99.1	**100**
Textures	98.5	99.0	96.4	98.8	98.4	96.4	99.5	99.2	99.1	**99.8**
Bottle	99.0	-	99.6	**100**	**100**	99.6	**100**	**100**	99.2	99.8
Cable	96.3	-	85.5	97.6	99.6	92.2	97.6	**99.8**	91.8	97.1
Capsule	91.4	-	87.0	93.2	95.5	91.5	97.7	97.8	98.5	**98.6**
Hazelnut	98.2	-	84.1	99.9	99.8	93.3	**100**	99.8	**100**	99.3
Metal nut	98.8	-	97.4	98.5	99.1	99.2	99.3	99.4	98.7	**99.7**
Pill	**99.1**	-	86.9	93.0	96.9	94.4	96.8	96.3	98.9	**99.1**
Screw	**100**	-	74.5	85.9	90.0	84.4	91.9	97.9	93.9	99.6
Toothbrush	97.4	-	94.7	99.9	**100**	97.2	99.7	**100**	**100**	99.1
Transistor	94.5	-	92.5	93.0	**99.8**	97.8	95.2	99.6	93.1	97.6
Zipper	94.1	-	74.1	96.2	99.1	90.9	98.5	99.9	**100**	91.9
Objects	96.9	97.2	87.6	95.7	98.0	94.1	97.7	**99.0**	97.4	98.2
All	97.4	97.9	90.5	96.7	98.1	95.5	98.3	**99.1**	98.0	98.7

Table 2. Comparison of state-of-the-art methods on MVtec using AUROC. Detailed results for PaDim with EfficientNet are not available.

	PaDim [12]	Ours	DRAEM [44]
AUROC	95.0	98.0	**99.0**

Table 3. Comparison of state-of-the-art methods on DAGM [1] using AUROC.

4.3. Comparison with the state of the art

We present the results of the comparison of our method with the state of the art on the MVTec dataset [3] in Table 2. Results are evaluated with the area under ROC curve (AUROC) metric. Using a WideResNet-50 backbone, our method improves the state of the art by 0.4%. While it trails slightly on the textures, which was to be expected as explained in Section 4.2, it performs particularly well on the object category with an improvement of 0.8% over the previous best method. The table also shows that the ResNet-18 variant of our model is very competitive while being much more lightweight. It is a good option when memory and/or computational power are limited. We also present results on DAGM [1] in Table 3.

While the goal of our model is to detect the anomaly at image level, it is possible to estimate rough heat maps from the probabilities estimated from our model. We compare their quality to other methods in Table 6. While CFlow-ad [16] remains the best method, our proposed method is still second with competitive performance. A few heat maps for the different classes of the MVTec dataset are shown in Figure 5. See the supplementary material for more heat map examples. We also compared the memory requirement and computation times with PaDim [12] in Table 5. Overall, our method, while performing a much finer modeling, is competitive both in terms of memory and computation time, especially when using ResNet18. Note that the size of our model doesn't depend on the size of the input image (with the exception of the negligible local weight map).

MVTec	MahaAD [29]	PaDim [12]	CFlow-ad [16]	DRAEM [44]	Ours
normal	92.8	94.0	99.0	97.3	98.9
Random	73.8	82.7	93.3	94.6	97.7
Diff. $\downarrow$	19.0	11.3	5.7	2.4	**1.2**

Table 4. Comparison of state-of-the-art methods on a randomized version of MVtec using the AUROC metric. The smaller the performance gap between normal and randomized versions, the better.

	PaDim	Ours	CFlow-ad	DRAEM	CS-Flow
Memory (GB)	3.8	3.04	0.64	0.36	1.03
Comp. time (s)	0.63	0.61	0.98	0.94	0.39

Table 5. Computation time and memory requirement comparison on an Intel Core i7-10700K CPU. Memory is estimated based on the number of parameters saved by each model after training. WideResNet-50-2 is used for PaDim, CFlow-ad and our method.

4.4. Robustness

A major limitation of the MVTec dataset is that almost all objects are perfectly aligned and with fixed lighting conditions, which is not realistic in a production line. In order to measure the robustness to pose perturbations, we produced a randomized version of MVTec, similar to Rd-MVTec from [12], where we added a small position jitter and a random rotation. For this experiment, we only kept objects that were not close to the boundary of their image so as to avoid boundary effects caused by the rotation-translation correction. We therefore only processed bottle, cable, capsule, hazelnut, metal nut, pill and screw. We then

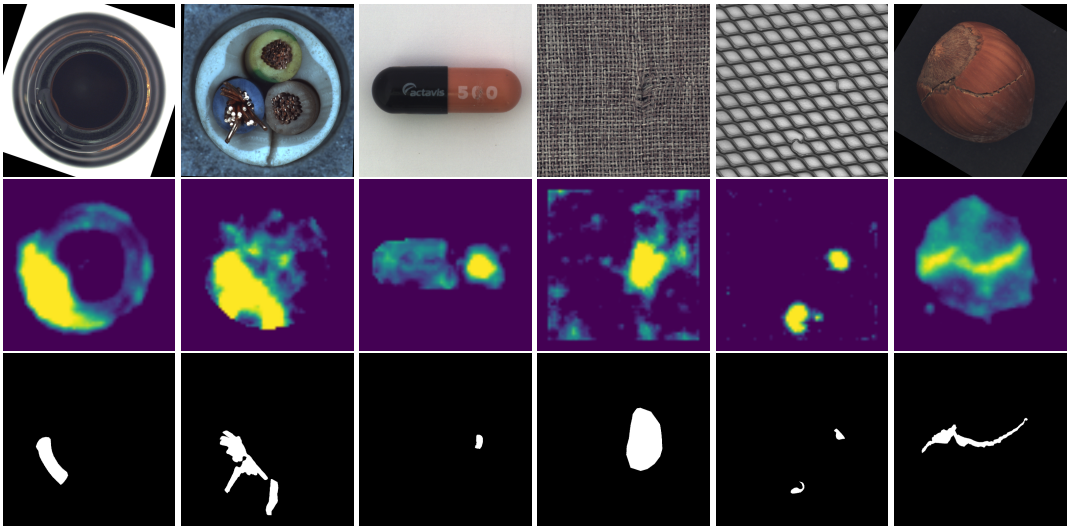

Figure 5. Examples of heat maps obtained with the multi-scale probabilities on WideResnet50-2. These heat maps give a good idea of the localization of the detected defects even though this localization is imprecise.

	PaDim [12]	CFlow-ad [16]	Ours	DRAEM [44]
Carpet	99.1	**99.2**	97.8	95.5
Grid	97.3	99.0	**99.7**	**99.7**
Leather	99.2	99.7	**99.8**	98.6
Tile	94.1	98.0	96.1	**99.2**
Wood	94.9	**96.6**	95.8	96.4
Textures	96.9	**98.6**	97.8	97.9
Bottle	98.3	**99.0**	96.9	99.1
Cable	96.7	97.6	**98.6**	94.7
Capsule	98.5	**99.0**	98.7	94.3
Hazelnut	98.2	**98.9**	98.2	99.7
Metal nut	97.2	98.6	96.2	**99.5**
Pill	95.7	**98.9**	96.2	97.6
Screw	98.5	98.6	**99.9**	97.6
Toothbrush	98.8	**98.9**	98.9	98.1
Transistor	97.5	**98.0**	96.5	90.9
Zipper	98.5	**99.1**	**99.1**	98.8
Objects	97.8	**98.7**	97.9	97.0
All	97.5	**98.6**	97.9	97.3

Table 6. Comparison of state-of-the-art methods on pixel level on MVtec using the AUROC metric.

retrained several state-of-the-art methods, namely [29, 12, 44, 16], and compare their performance with GLAD on this dataset in Table 4. We averaged the results for five different randomized versions of the dataset. While most methods suffer from these perturbations, GLAD and DRAEM [44] both show a good robustness.

5. Discussion

We have introduced GLAD, a global-local Gaussian model of neural network features for unsupervised anomaly detection. Our model is comprised of a global Gaussian mixture learned on features from a pre-trained neural network and of a local weight map. The global model is learned on all normal features, thus producing a precise and non-degenerate model of these features. A local weight map is then learned using the global model, and indicates which Gaussian is relevant to model the samples at a given position. In that way we model each position with a non-degenerate Gaussian mixture model, even when few normal data are available. The model puts in evidence a spatial specialization of each Gaussian and, conversely, the advantage of having more than one Gaussian to model the set of samples at each position. The weight map, being sparse for each object, enables faster computations. With this global-local model, we improved the state of the art on the MVTec dataset while still being competitive in terms of efficiency.

Despite improving the state of the art, our model struggles with textures. Due to our inability to align them on a reference without causing boundary effects, the proposed localization becomes useless. Thus, our model remains in that case equivalent to a mere Gaussian mixture model. Because of their repetitive pattern, we should anticipate that a localization in frequency might be preferable to the spatial localization used in our current model. We plan to study this extension in future work. Another major limitation of the proposed model is its inability to evolve once learned. For example, when a small update is made to the product tracked, it is required to learn an entirely new model even though the previous model was almost entirely correct. To amend for this limitation, we plan to train and update a Gaussian mixture model online as described in [35]. Doing so would also arguably speed up the current training step or at least enable the use of the model even before convergence. Finally, GLAD needs to be improved to produce a better segmentation of the anomaly. While a coarse localization is often enough, several methods in the literature produce more precise segmentation masks, despite their worse performance at image level.

References

[1] https://conferences.mpi-inf.mpg.de/dagm/2007/prizes.html.

[2] Aitor Artola, Yannis Kolodziej, Jean-Michel Morel, and Thibaud Ehret. Unsupervised variability normalization for anomaly detection. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 989–993. IEEE, 2021.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In *CVPR*, 2019.

[4] Giacomo Boracchi, Diego Carrera, and Brendt Wohlberg. Novelty detection in images by sparse representations. In *2014 IEEE Symposium on Intelligent Embedded Systems*, pages 47–54. IEEE, 2014.

[5] Thibaud Briand, Gabriele Facciolo, and Javier Sánchez. Improvements of the Inverse Compositional Algorithm for Parametric Motion Estimation. *Image Processing On Line*, 8:435–464, 2018.

[6] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *CVPR*. IEEE, 2005.

[7] Diego Carrera, Giacomo Boracchi, Alessandro Foi, and Brendt Wohlberg. Detecting anomalous structures by convolutional sparse models. In *2015 International Joint Conference on Neural Networks*, pages 1–8. IEEE, 2015.

[8] Diego Carrera, Giacomo Boracchi, Alessandro Foi, and Brendt Wohlberg. Scale-invariant anomaly detection with multiscale group-sparse models. In *ICIP*. IEEE, 2016.

[9] Yilun Chen, Ami Wiesel, Yonina C. Eldar, and Alfred O. Hero. Shrinkage algorithms for mmse covariance estimation. *IEEE Transactions on Signal Processing*, 58:5016–5029, 2010.

[10] Niv Cohen and Yedid Hoshen. Transformer-based anomaly segmentation. *arXiv*, 2020.

[11] Axel Davy, Thibaud Ehret, Jean-Michel Morel, and Mauricio Delbracio. Reducing anomaly detection in images to detection in noise. In *ICIP*. IEEE, 2018.

[12] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: A patch distribution modeling framework for anomaly detection and localization. In Alberto Del Bimbo, Rita Cucchiara, Stan Sclaroff, Giovanni Maria Farinella, Tao Mei, Marco Bertini, Hugo Jair Escalante, and Roberto Vezzani, editors, *Pattern Recognition. ICPR International Workshops and Challenges*, pages 475–489, Cham, 2021. Springer International Publishing.

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009. IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[14] Bo Du and Liangpei Zhang. Random-selection-based anomaly detector for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote sensing*, 49(5):1578–1589, 2011.

[15] Thibaud Ehret, Axel Davy, Jean-Michel Morel, and Mauricio Delbracio. Image anomalies: A review and synthesis of detection methods. *JMIV*, 61(5):710–743, 2019.

[16] Denis Gudovskiy, Shun Ishizaka, and Kazuki Kozuka. Cflow-ad: Real-time unsupervised anomaly detection with localization via conditional normalizing flows. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Toshifumi Honda and Shree K Nayar. Finding" anomalies" in an arbitrary image. In *ICCV*. IEEE, 2001.

[19] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *PAMI*, 20(11):1254–1259, 1998.

[20] Gukyeong Kwon, Mohit Prabhushankar, Dogancan Temel, and Ghassan AlRegib. Backpropagated gradient representations for anomaly detection. In *European Conference on Computer Vision*, pages 206–226. Springer, 2020.

[21] Yuening Li, Zhengzhang Chen, Daochen Zha, Kaixiong Zhou, Haifeng Jin, Haifeng Chen, and Xia Hu. Autood: Automated outlier detection via curiosity-guided search and self-imitation learning. *ICDE*, 2020.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[23] Wenqian Liu, Runze Li, Meng Zheng, Srikrishna Karanam, Ziyan Wu, Bir Bhanu, Richard J Radke, and Octavia Camps. Towards visually explaining variational autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8642–8651, 2020.

[24] Ran Margolin, Ayellet Tal, and Lihi Zelnik-Manor. What makes a patch distinct? In *CVPR*, 2013.

[25] Frank Nielsen. K-mle: A fast algorithm for learning statistical mixture models. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 869–872, 2012.

[26] Guansong Pang, Chunhua Shen, Longbing Cao, and Anton Van Den Hengel. Deep learning for anomaly detection: A review. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.

[27] Xiangqian Peng, Youping Chen, Wenyong Yu, Zude Zhou, and Guodong Sun. An online defects inspection method for float glass fabrication based on machine vision. *The International Journal of Advanced Manufacturing Technology*, 39(11-12):1180–1189, 2008.

[28] Irving S Reed and Xiaoli Yu. Adaptive multiple-band cfar detection of an optical pattern with unknown spectral distribution. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 38(10):1760–1770, 1990.

[29] O. Rippel, P. Mertens, and D. Merhof. Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6726–6733, Los Alamitos, CA, USA, jan 2021. IEEE Computer Society.

[30] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Schölkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14318–14328, June 2022.

[31] Marco Rudolph, Tom Wehrbein, Bodo Rosenhahn, and Bastian Wandt. Fully convolutional cross-scale-flows for image-based defect detection. In *Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022.

[32] Lukas Ruff, Nico Görnitz, Lucas Deecke, Shoaib Ahmed Siddiqui, Robert Vandermeulen, Alexander Binder, Emmanuel Müller, and Marius Kloft. Deep one-class classification. In *International Conference on Machine Learning*, pages 4390–4399, 2018.

[33] Lukas Ruff, Jacob R Kauffmann, Robert A Vandermeulen, Grégoire Montavon, Wojciech Samek, Marius Kloft, Thomas G Dietterich, and Klaus-Robert Müller. A unifying review of deep and shallow anomaly detection. *Proceedings of the IEEE*, 2021.

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *IJCV*, 115(3):211–252, 2015.

[35] Christophe Saint-Jean and Frank Nielsen. Online k-mle for mixture modeling with exponential families. pages 340–348, 01 2015.

[36] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[37] Hui-Chao Shang, You-Ping Chen, Wen-Yong Yu, and Zu-De Zhou. Online auto-detection method and system of presswork quality. *The International Journal of Advanced Manufacturing Technology*, 33(7-8):756–765, 2007.

[38] Lionel Tarassenko, Paul Hayton, Nicholas Cerneaz, and Michael Brady. Novelty detection for the identification of masses in mammograms. 1995.

[39] Du-Ming Tsai and Tse-Yun Huang. Automated surface inspection for statistical textures. *Image and Vision computing*, 21(4):307–323, 2003.

[40] Shashanka Venkataramanan, Kuan-Chuan Peng, Rajat Vikram Singh, and Abhijit Mahalanobis. Attention guided anomaly localization in images. In *ECCV*, 2020.

[41] Xianghua Xie and Majid Mirmehdi. Texems: Texture exemplars for defect detection on random textured surfaces. *IEEE PAMI*, 29(8):1454–1464, 2007.

[42] Jihun Yi and Sungroh Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation. In *ACCV*, 2020.

[43] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 87.1–87.12. BMVA Press, September 2016.

[44] Vitjan Zavrtanik, Matej Kristan, and Danijel Skocaj. Draema discriminatively trained reconstruction embedding for surface anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8330–8339, 2021.

[45] Richard Zhang. Making convolutional networks shift-invariant again. In *ICML*, 2019.

[46] Kang Zhou, Yuting Xiao, Jianlong Yang, Jun Cheng, Wen Liu, Weixin Luo, Zaiwang Gu, Jiang Liu, and Shenghua. Gao. Encoding structure-texture relation with p-net for anomaly detection in retinal images. In *ECCV*, 2020.

[47] Maria Zontak and Israel Cohen. Defect detection in patterned wafers using anisotropic kernels. *Machine Vision and Applications*, 21(2):129–141, 2010.

[48] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486. IEEE, 2011.