Quantized FCA: Efficient Zero-Shot Texture Anomaly Detection

Andrei-Timotei Ardelean[ID] and Patrick Rückbeil[ID] and Tim Weyrich[ID]

Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany

Abstract
*Zero-shot anomaly localization is a rising field in computer vision research, with important progress in recent years. This work focuses on the problem of detecting and localizing anomalies in textures, where anomalies can be defined as the regions that deviate from the overall statistics, violating the stationarity assumption. The main limitation of existing methods is their high running time, making them impractical for deployment in real-world scenarios, such as assembly line monitoring. We propose a real-time method, named QFCA, which implements a quantized version of the feature correspondence analysis (FCA) algorithm. By carefully adapting the patch statistics comparison to work on histograms of quantized values, we obtain a $10\times$ speedup with little to no loss in accuracy. Moreover, we introduce a feature preprocessing step based on principal component analysis, which enhances the contrast between normal and anomalous features, improving the detection precision on complex textures. Our method is thoroughly evaluated against prior art, comparing favorably with existing methods.
Project page: `reality.tf.fau.de/pub/ardelean2025quantized.html`*

CCS Concepts
• *Computing methodologies → Anomaly detection;*

1. Introduction

The task of finding outliers in a group of otherwise similar elements is called anomaly detection (AD). These outliers are also referred to as anomalies, discordant observations, exceptions, aberrations, surprises, peculiarities, or contaminants depending on the particular context [CBK09]. Some notable fields of application are manufacturing inspection [LXW*24a, HCH19], medical imaging [FGD*22], as a preprocessing step in data analysis [Tom76, SM11], in machine vision [Agg16] and neuroscience [BSZ*21], in weather records [WASA21], for fraud detection in the financial industry [AMI16], as well as monitoring acoustic [CZG*20] or video [KGG18] signal. In most use-cases of image AD, it is crucial to not only detect if anomalies are present, but also segment the offending regions. The task of zero-shot anomaly localization (ZSAL), targeted by this work, refers to locating such outliers without prior training or an indication as to what outliers or even non-outlier regions might look like.

Humans are generally capable of making the distinction between normal elements and irregularities, when given a set of elements without labeled outliers [TGZ*22]. Given some form of homogeneous data, like textures, anomalies would arise in the form of disturbances of the homogeneity. This assumption of homogeneity, or stationarity, supports the idea of automatically locating anomalies in a single image, based on the internal statistics.

In this paper, we approach the problem of zero-shot anomaly localization with a specific focus on textures. The method builds
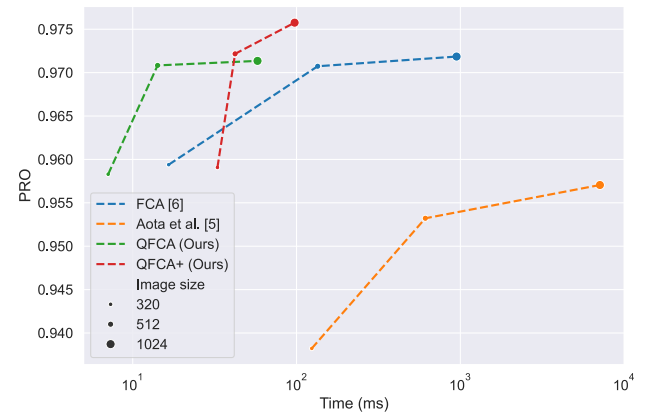

Figure 1: *Anomaly localization fidelity (PRO) vs Time, obtained using different algorithms at various resolutions. Our approach enables the best tradeoff between accuracy and runtime.*

upon feature correspondence analysis (FCA) [AW24b], the current state-of-the-art algorithm for this task, lifting its crucial limitation: running time. We identify the most time-consuming operations in FCA and replace them with a quantized version, obtaining a significant efficiency boost (one order of magnitude) with a limited decrease in accuracy. To further minimize the running time, we employ an efficient algorithm for local average pooling that runs in

constant time with respect to kernel size. Moreover, we found the k-NN version of FCA (designed to improve anomaly localization on more complex textures) prohibitively slow for images larger than 320×320. To mitigate this issue, we advance a feature preprocessing technique that yields a similar improvement in accuracy with a much lower running time.

In essence, our contributions are as follows:

- We introduce a real-time algorithm for zero-shot anomaly localization with unprecedented detection accuracy.
- We propose a feature preprocessing technique that improves detection on complex textures.
- We identify and resolve a computational bottleneck caused by an inefficient implementation of local average pooling in modern ML libraries, including PyTorch, TensorFlow, and JAX.

2. Related Work

The field of AD has a rich research history and a wide range of applications, with various methods being thereby specialized for different contexts and conditions. We refer the reader to the survey of Hodge and Austin [HA04] for a wide-scoped exposition about different approaches to outlier detection and to Liu *et al.* [LXW*24b] for a modern outlook on Visual Anomaly Detection.

Unsupervised anomaly detection. The most common setting for AD in images is unsupervised anomaly detection. More correctly referred to as one-class classification or normality-supervised [AW24a], the methods in this category use a curated set of *normal* images, guaranteed not to contain anomalies. An early popular approach for anomaly detection using neural networks (NN) includes reconstruction-based methods [TGZ*22], which try to recreate the image content from limited information or a bottleneck. This can be achieved using (variational) autoencoders [AC15, BLF*19, YRPK19, BWAN19] or generative adversarial networks (GAN) based approaches such as Skip-GANomaly [AAAB19] and SAC [YLH*22]. During inference, these methods localize anomalies using the difference between the original and the reconstruction. The underlying intuition is that the model fails to reconstruct anomalous regions since it was trained only on *normal* images.

To achieve high-fidelity detections, most recent approaches leverage neural features extracted with a CNN pretrained on ImageNet [DDS*09]. Various methods make use of these features in different ways, such as computing the k-nearest neighbors (k-NN) [BCH20, CH21], modeling multivariate Gaussians at the pixel level [DSLA21], creating a memory bank [RPZ*22], and contrastive supervision [LLS22]. A way to speed up feature extraction is to train a smaller network to predict the teacher's features [BHK24].

A significant category of unsupervised anomaly detection is represented by the **few-shot** setting. A growing number of works tackle the problem of anomaly detection with small amounts of *normal* data [HGJ*22, LLC*23, LGLF24, HGJ*24]. These methods try to maximize the information extracted from a handful of images and incorporate cross-category knowledge from other image classes. Although the few-shot setting is significantly more challenging compared to the general case, there is still a large difference to the

zero-shot setting addressed in our work, where no reference images are provided.

Zero-shot anomaly detection. Localizing anomalies from a single image of a previously unseen image class is the most challenging setting in AD, as it requires detecting anomalies using a small amount of contaminated data. In this sense, it is similar to blind anomaly detection [ZSO24, AW24a] at the level of image patches, albeit with a smaller sample size. The task was formalized in MAE-DAY [SAK*24], where a zero-shot solution was proposed, based on a masked autoencoder (MAE). The works that followed can be generally grouped into two categories: based on a visual language model (VLM) or based on internal statistics.

To contend with the lack of supervisory signal in the zero-shot setting, VLM-based methods seek to leverage the massive pretraining of these models to distinguish between normal and anomalous appearance. WinCLIP [JZK*23] is the pioneer of this approach to ZSAL; it identifies anomalies through carefully designed prompts, running the CLIP [RKH*21] model on several scales to obtain localized patch-prompt matching scores. Several methods have since been proposed that improve upon the idea of WinCLIP by bridging the domain gap (April-GAN) [CHZ23], using object-agnostic prompts (AnomalyCLIP) [ZPT*23], leveraging SAM for mask proposals (SAA) [CXS*23], improving the computation of representative text features (SDP) [CZT*24], and combining CLIP and SAM (ClipSAM) [LCY*25].

The second category is suitable for textures and relies on the information within the image to model the distribution of *normal* features. The anomalies are therefore the regions that are outliers with respect to the overall content of the texture. The method developed by Aota *et al.* [ATO23] (denoted ZvM) builds on the idea that an anomalous feature patch has fewer similar patches across the image compared to a normal one. As such, the mean distance to the k-nearest neighbors for a suitable k will be significantly higher for anomalous patches, enabling their localization.

Ardelean and Weyrich [AW24b] propose a framework for ZSAL consisting of three components: feature extraction, reference selection, and patch statistics comparison. FCA itself is an implementation of this abstraction that provides high-fidelity detections by using a fine-grained error-to-pixel association. The algorithm creates patches for each spatial location and compares them with a global representation of the texture. The comparison computes a patchwise error map using a bijective mapping derived from the 1-dimensional Wasserstein Distance [EW22]. Lastly, a score is computed using the aggregation of the pixel-wise contribution to the error maps of all nearby patches, which are finally averaged across the channel dimension (more details in Section 3.1).

3. Algorithm Design

Our method, named quantized feature correspondence analysis (QFCA), follows the framework of FCA for zero-shot localization. That is, the anomaly maps are computed according to the following formula:

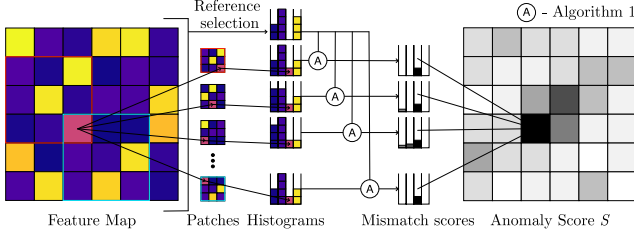$$A(x,y;F,S,R) := \sum_{F_r \in R(F(I))} S(x,y,F(I),F_r) , \qquad (1)$$

Figure 2: *Our QFCA pipeline: each patch is represented as a histogram and compared to the reference using our Algorithm 1. The final score for each pixel is given by averaging the mismatch scores from the histograms it is part of.*

where F denotes feature extraction, S compares the patch statistics, and R computes a set of global references that characterize the texture. Informally, the equation defines the anomaly score at a pixel location x, y as the sum of errors associated with that pixel when comparing the features of the surrounding patch with the set of global references. Our method differs from FCA in the implementation of these three components, which will be described in the following subsections.

3.1. Patch Statistics Comparison

The patch statistics comparison function $S(x, y, F(I), F_r)$ evaluates the degree of anomaly for each position (x, y) within its local context in the feature map $F(I) \in \mathcal{R}^{C \times H \times W}$. To do so, the function compares the local statistics around (x, y) and the statistics of a global reference F_r.

To keep the description self-contained, we begin by detailing the patch statistics comparison of FCA, and then present our approach: FCA first separates the image into multiple overlapping patches by taking each pixel and its surroundings. For each patch P, a bidirectional mapping is formed between the pixels and the reference by sorting the feature values and mapping elements of the same rank. The error associated with each pixel (x, y) is the absolute difference between the feature value and the corresponding element in the reference. Formally, this error represents the contribution of the pixel to the optimal transport between the empirical distribution of the patch and the reference. This is repeated for each feature channel independently and aggregated into a summative mismatch score $M(x, y; P)$ for each pixel of that patch P.

The most important design decision of the algorithm is that the anomaly score for a specific location is not computed based on the mismatch scores of nearby pixels, as in ZvM [ATO23]; instead, it is computed by aggregating the mismatch scores of that pixel in the context of the nearby patches (of which it is part). Formally, FCA defines it as

$$S(x, y) = \sum_{(x', y') \in P_{xy}} \mathcal{G}_{\sigma_s}(M(\cdot; P_{x'y'}))(x, y) G_{\sigma_p}(x' - x, y' - y), \quad (2)$$

where $\mathcal{G}_{\sigma_s}$ denotes Gaussian blurring with standard deviation

σ_s and $G_{\sigma_p}(x, y)$ directly retrieves the PDF of the bivariate normal distribution.

The most time-consuming operation of FCA is represented by the sorting operation that creates the bidirectional mapping between elements of the same rank. This is especially costly, as it must be performed for each patch for each channel. The main quality of our QFCA (Fig. 2) is the quantized representation of patches as histograms. As shown in our experiments (Sec. 4.5.1), by carefully adapting the error computation and association, we obtain a localization performance similar to that of the non-quantized version with as little as 16 bins.

In the following, we describe our efficient algorithm for comparing a patch to a global reference and computing the error contribution analogous to the operation in FCA. In the supplementary, we provide a proof for the correctness of the algorithm and a mathematical justification of its efficacy based on its relation to the gradient of the 2-Wasserstein distance (S4).

Firstly, we quantize the feature values into N equally-spaced values $\{Q_i\}_{i=1}^{N}$. Each patch is then represented using histograms based on the quantized feature values; we denote with $\{P_i\}_{i=1}^{N}$ and $\{R_i\}_{i=1}^{N}$ the histogram weights of the patch and reference vectors, respectively. Since we deal with empirical distributions, the histogram weights are integers; however, the following algorithm can be applied unchanged to arbitrary weights.

To find the optimal transport, we use a two-pointer (i, j) algorithm that iterates the matching quantiles in P and R. At each step, the error attributed to the current bin in P is given by $\min(P_i, R_j) \cdot |Q_i - Q_j|$, indicating a transport between bin i and bin j. Then, the index pointing to the smaller weight is moved forward, while the amount of mass that was transported, *i.e.* $\min(P_i, R_j)$, is removed from the larger bin. See Algorithm 1 for a formal description.

Algorithm 1: Quantized patch mismatch score

Input: $\{P_i\}_{i=1}^{N}$, $\{R_i\}_{i=1}^{N}$, $\{Q_i\}_{i=1}^{N}$

1   Initialize mismatch scores $E_i \leftarrow 0$, $i = 1..N$
2   Make a copy of $\{P_i\}_{i=1}^{N}$ in $\{\hat{P}_i\}_{i=1}^{N}$
3   $i \leftarrow 1, j \leftarrow 1$;
4   **while** $i \leq N$ *and* $j \leq N$ **do**
5   **if** $P_i < R_j$ **then**
6       $E_i \leftarrow E_i + P_i \cdot |Q_i - Q_j|$;
7       $R_j \leftarrow R_j - P_i$;
8       $i \leftarrow i + 1$;
9   **else**
10      $E_i \leftarrow E_i + R_j \cdot |Q_i - Q_j|$;
11      $P_i \leftarrow P_i - R_j$;
12      $j \leftarrow j + 1$;
13   **end**
14   **end**
15   $E_i \leftarrow \frac{E_i}{\hat{P}_i}$ **for** $i = 1..N$;
16   **return** $\{E_i\}_{i=1}^{N}$

After completion of the algorithm, E_i stores the contribution of the bin i to the Wasserstein distance between the patch P and the
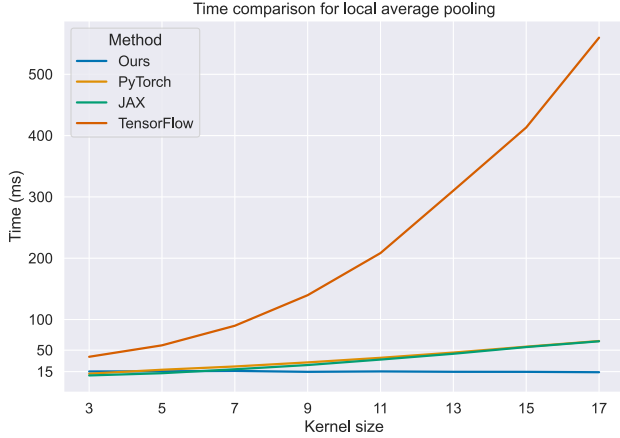
Figure 3: *Runtime for average pooling with different kernel sizes, operating on a 128×128 tensor with 8192 channels (mirroring our 16 bins per feature setting). All methods use GPU acceleration.*

reference, which is the histogram equivalent of the FCA mismatch score $M(x,y;P)$ for a certain channel. The division on line 15 ensures that the mismatch score for each bin is weighted by the number of elements that contributed to the error. It can be easily seen that the algorithm finishes after $2N-1$ steps and therefore has a linear complexity in the number of bins compared to the $O(T^2 \log(T))$ of FCA (where T is the patch size). Moreover, we compile the code to a CUDA kernel and benefit from GPU acceleration, as the computation can be performed in parallel for all patches and all channels.

The error-to-pixel association (Eq. (2)) is a crucial component of the algorithm, which differentiates it from a simple Wasserstein distance between the patches and the reference. In order to recover the same formulation as FCA, after computing the bin-level error for each patch, we distribute the error contribution to all pixels that formed each histogram bin. Similarly to the Gaussian smoothing from Eq. (2), we aggregate the errors of all histogram bins of which a certain pixel is part. We simply implement this operation as a Gaussian blurring over the bin-level errors with a standard deviation of σ_p and a kernel size equal to the patch size. Finally, the anomaly score of a pixel is given by its corresponding bin after smoothing.

As highlighted in the description of our algorithm, its time complexity does not depend on the patch size T; however, our initial experiments still revealed a linear dependence with T of the overall running time. More precisely, the patch-size dependence occurred in two places: the formation of histograms and the error-to-pixel association, even as these operations were implemented as local average pooling. To our surprise, we found that the running time of the PyTorch implementation of average pooling increases with the kernel size; upon inspection, the same observation was made for other major ML frameworks, *i.e.*, TensorFlow and JAX. To address this, we implement the 2D average pooling using summed-area tables [Cro84] (also known as integral image [VJ01]), having an $O(H \times W)$ complexity per channel. The algorithm readily benefits from parallelization, resulting in a fast runtime, as shown in Fig-

ure 3. Using our fast average pooling implementation, we reduce the overall running time of our method by 30% in the usual case (patch size of 9) and make our algorithm's complexity completely independent of the patch size.

3.2. Reference Selection

Ardelean and Weyrich [AW24b] explored various options for computing the set of references $R(F(I))$, such as: all patches, random selection, patch average, and median. The latter proved to be the best option in terms of detection fidelity and running time, which is because it uses a single reference to capture the entire global statistics $R(F(I)) = \{F_r\}$. FCA defines the feature reference F_r as:

$$\underset{F_r}{\arg\min} \sum_{(x,y)} A(x,y;\cdot,\cdot,R=F_r) \,,$$

which is the distribution that minimizes the Wasserstein distance across all patches. This minimization problem has a closed-form solution, obtained by computing the median over the patches for each rank in sorted order. A reference vector is computed for each feature channel separately.

We similarly use median-based reference selection by adapting it to our quantized representation of features. We find that the best results are obtained when using the same method as FCA to compute the median with full precision and only quantize the reference afterward. Section 4.5.3 compares different ways to compute the reference set in terms of accuracy.

Although the median reference computation works well for homogeneous images, it struggles with complex textures *e.g.*, where the texture period is larger than the patch size. An idea that was investigated in FCA is a pairwise comparison with the *k*-nearest neighbors of all patches. This showed a clear improvement in accuracy for elaborate textures, albeit with a severely slower runtime. Our paper introduces an alternative approach that increases the accuracy for more complex images, without making the method prohibitively slow. Our approach enhances the features by reducing the variance of normal, frequently-occurring features, as described in the following section.

3.3. Feature Extractor

The task of the feature extractor (F) is to create spatial features based on the input image. Virtually all zero-shot anomaly detection methods based on statics rely on medium- and high-level descriptors, which are more suitable than raw pixels to differentiate between normal and anomalous appearance. Pretrained CNNs have been shown to provide features that are better than hand-annotated features [WKW*16]. Specifically, Roth *et al*. [RPZ*22] evaluated different neural networks and found Wide ResNet-50 [ZK16] to perform especially well on the task of anomaly detection. Considering the good results and acceptable running times of this CNN, we also use it for feature extraction in our pipeline.

Inspired by the use of VAE residuals in BlindLCA [AW24a], we propose a single-image feature preprocessing technique with the aim of reducing the variance of normal features in the case of intricate images. For example, in bimodal textures, where the period of the texture is larger than the patch size, a single global reference cannot

capture the entire complexity. FCA + k-NN aims to fix this by computing all pairwise distances between patches and then discarding outliers by considering only the closest k distances. Nonetheless, this method has a high running time, as it scales quadratically with the number of patches [AW24b]. Conversely, we build on the idea behind reconstruction-based methods, and use the residuals between the original image and the reconstruction as the new features on which we apply QFCA. Differently from BlindLCA, we use a single image and employ principal components analysis (PCA) for reconstruction in order to preserve a fast running time. Intuitively, our feature preprocessing follows the idea that anomalies will not be well represented by the principal feature components of the input features. Therefore, subtracting the PCA reconstruction from the input reduces the variance of normal features, while preserving that of anomalies. This technique helps with elaborate textures, for which the reference struggles to capture the global statistics (Sec. 3.2).

Formally, our improved feature extractor is defined as:

$$F(X) = W(X) - \text{PCA}^{-1}(\text{PCA}(W(X))) \, ,$$

where W represents the Wide Resnet-50 network; *i.e.*, the feature extracted after the second convolutional block. We denote our full method, which uses this preprocessing, as QFCA+. The number of principal components used in dimensionality reduction plays an important role, which we analyze in Section 4.5.2.

4. Experiments

We evaluate our QFCA algorithm and the feature preprocessing step on three different datasets using the established metrics. The experiments show that our approach localizes anomalies faster and more accurately compared to existing zero-shot methods.

4.1. Datasets

MVTec AD [BFSS19, BBF*21] is the current industry standard dataset for anomaly localization. As our method is designed to work on textures, we use the five texture classes: carpet, grid, leather, tile, and wood; together they accumulate over 500 images and their respective segmentation masks.

DTD-Synthetic is a synthetic dataset by Aota *et al.* [ATO23], created for evaluating anomaly detection on more diverse and complex textures. It is based on DTD (Describable Texture Dataset) [CMK*14], and consists of 12 textures with 100 training images and over 100 test images each. The images are cropped out of the original DTD image with a random orientation and position, resulting in image sizes between 180×180 and 384×384 pixels.

Woven Fabric Textures (WFT) introduced by Bergmann *et al.* [BLF*19] is a dataset with 2 textures with 50 images each. The dataset includes ground-truth segmentation masks at a resolution of 512×512.

4.2. Metrics

For the evaluation of the results, we use the common metrics and the same post-processing steps as in FCA. That is, the borders are discarded and the image-level anomaly score is computed as the maximum of the pixel-level scores.

	PRO	AUROC$_s$	F$_1$	Time(s)
WinCLIP [JZK*23]	71.50	89.06	38.42	0.389
SAA+ [CXS*23]	64.79	77.82	59.19	0.270
April-GAN [CHZ23]	92.56	96.51	58.63	0.122
SDP+ [CZT*24]	92.48	96.76	52.70	0.045
AdaptClip [GZY*25]	-	97.68	55.86	0.162
CRANE [SSH*25]	95.18	97.96	57.02	-
ZvM [ATO23]	93.82	97.47	60.60	1.100
FCA [AW24b]	97.18	98.73	71.75	1.070
GRNR [YLC*24]	94.65	97.44	62.85	0.145
QFCA$_{512}$	97.08	98.77	68.91	**0.014**
QFCA (Ours)	97.13	98.72	71.88	0.057
QFCA+ (Ours)	**97.57**	**98.83**	**73.07**	0.097

Table 1: *Results on the MVTec AD dataset. QFCA$_{512}$ denotes our method ran at a smaller resolution of 512×512. For the other methods we use the image size suggested by their authors.*

Anomaly localization is customarily evaluated using the AUROC$_s$ [FBH05] and the area under the per-region overlap curve (AUPRO) [BBF*21] metrics. Both metrics are threshold independent and operate at the pixel level. The PRO score was introduced to counteract the bias toward large anomalies, which is done by weighing the anomaly predictions by the area of the anomalous region in the ground truth. As recommended by Bergmann *et al.* [BFSS19], we compute the area under the PRO curve up to a false positive rate of 30%; for brevity, we simply refer to this metric as PRO in the rest of this paper. Additionally, we quantify the anomaly localization fidelity at a fixed threshold using the F_1 metric [Chi92], *i.e.*, the harmonic mean between precision and recall calculated at the optimal threshold. As the purpose of this work is to optimize the running time and enable real-time usability, we also report the execution time for one image (s / img). Note that we measure latency, rather than throughput, keeping in mind a live-feed use-case.

As acknowledged by the community [ZWL*24], image-level AUROC$_c$ is saturated on datasets such as MVTec (scores above 99.5%). Therefore, we use the more representative pixel-level metrics in the main paper and refer the reader to the supplementary (S2) for extended numerical results.

4.3. Comparison with prior art

We compare our QFCA/QFCA+ with several zero-shot anomaly detection approaches, including methods based on VLMs: WinCLIP [JZK*23], SAA [CXS*23], AprilGAN [CHZ23], and SDP [CZT*24], as well as texture-specific methods: ZvM [ATO23], FCA [AW24b], and GRNR [YLC*24]. The results are compared quantitatively in Tab.1 and qualitatively in Fig. 4. It can be observed that QFCA performs significantly better than VLM-based methods and is on par with FCA while being more than 10 times faster. Thanks to our feature preprocessing technique, QFCA+ consistently yields the best metrics while still maintaining interactive rates.

We evaluate our method on two additional datasets in Tab. 2. The results suggest that QFCA is versatile and the improvements generalize across various texture classes. The improvement brought about by our feature preprocessing (QFCA+) is more significant
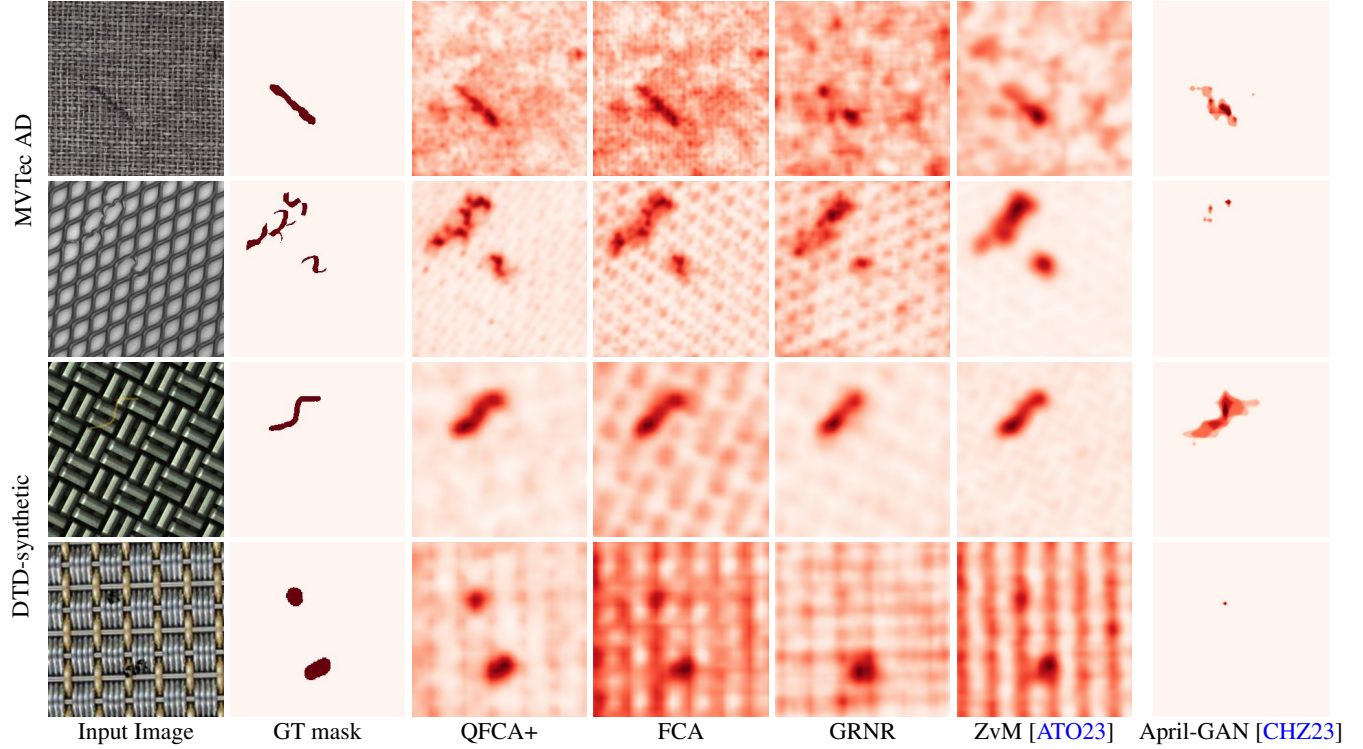
Figure 4: *Visualization of anomaly localization results on challenging examples by different methods.*

here compared to the MVTec dataset, since the textures are more elaborate. The benefit of QFCA+ can be observed in Fig. 4 as a reduction of false positives.

4.4. Image and patch size

In this section, we study the interplay between image size and patch size, which are paramount to texture anomaly detection. The size of the input has a strong effect on the resulting scores: while higher resolution images allow for a more precise localization, the receptive field of each feature becomes smaller, reducing the discriminatory potential. Moreover, as each pixel responds to a smaller area, there is a rising cost in execution time. We note that generally the patch size used by the anomaly detection method should be scaled proportionally to the image size. This relation is intuitive, since a decrease in feature receptive field can be compensated for by a larger patch size when computing local statistics.

Figure 5 shows the optimal patch size for different image sizes and their corresponding performance in milliseconds and PRO score. The largest image size allows for the highest accuracy, at the cost of increasing computational complexity. We empirically observe that the optimal patch size is about 10% of the size of the feature maps, *i.e.*, 1.25% of the size before feature extraction.

4.5. In-depth ablation study

In this section we analyze the role and behavior of the newly introduced components and their hyperparameters.

	PRO	AUROC$_s$	F$_1$	Time(s)
WFT (512×512)				
April-GAN [CHZ23]	84.97	94.90	71.51	0.122
ZvM [ATO23]	84.59	96.11	72.07	1.100
GRNR [YLC*24]	80.25	96.08	73.23	0.045
FCA [AW24b]	89.57	98.26	79.13	0.179
FCA + k-NN [AW24b]	88.77	97.73	76.30	6.345
QFCA (ours)	89.47	98.24	78.87	**0.013**
QFCA+ (ours)	**92.99**	**98.51**	**79.71**	0.041
DTD-Synthetic (320×320)				
April-GAN [CHZ23]	88.50	95.32	52.40	0.075
AdaptClip [GZY*25]	-	97.70	63.60	0.162
ZvM [ATO23]	94.32	98.00	65.96	1.100
GRNR [YLC*24]	92.07	97.29	61.14	0.042
FCA [AW24b]	94.82	98.14	68.75	0.008
FCA + k-NN [AW24b]	95.93	98.51	**71.79**	3.970
QFCA (Ours)	94.90	98.14	68.97	**0.005**
QFCA+ (Ours)	**96.64**	**98.74**	71.57	0.029

Table 2: *Results on the DTD-Synthetic and Woven Fabric Textures (WFT) datasets.*

4.5.1. Number of Bins

A key parameter of QFCA is the number of bins used in the quantization step. As described in the method section, increasing the number of bins makes the result closer to the sorting operation in FCA. A
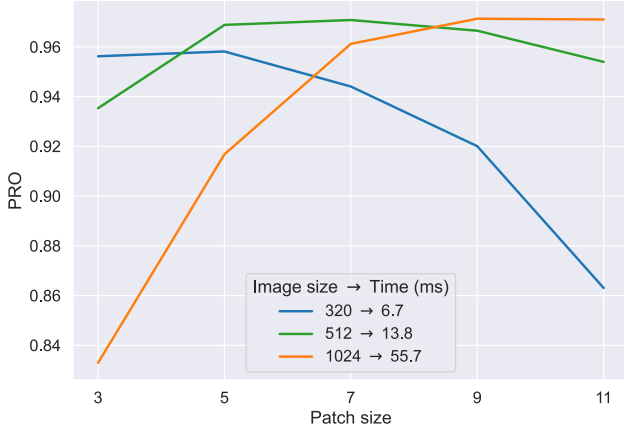
Figure 5: *Diagram showing the PRO score and execution time with patch sizes 3, 5, 7, 9, or 11. The optimum patch-size depends on the image size.*
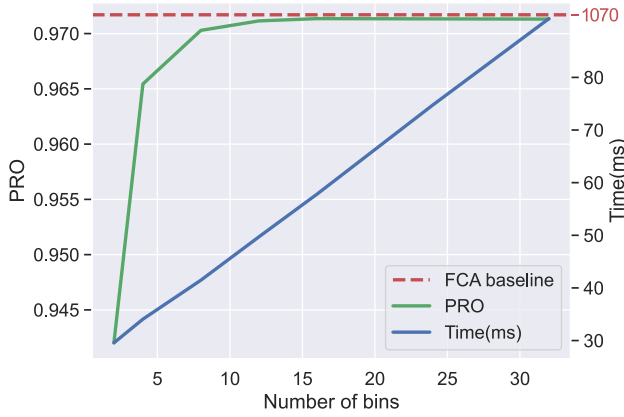

Figure 6: *Evaluating the latency and PRO metric of QFCA with increasing number of histogram bins; saturating around 16 bins.*

large number of bins would also hinder the overall performance; however, as seen in Fig. 6, 16 bins are already enough to match the result of FCA. Note that there is still a very small gap (0.05% difference) even when using a large number of bins; this is due to differences in postprocessing, as discussed in Sec. 4.5.4.

4.5.2. Feature preprocessing

We ablate our feature preprocessing in Fig. 7, including the results on different datasets. We show that our feature preprocessing can also be applied directly to FCA, similarly improving its performance (FCA+); in general, the method can be used in conjunction with arbitrary anomaly detection methods. Depending on the field of application, using our feature preprocessing can provide a substantial improvement, at the cost of a slightly increased running time; that is, about 40 ms for a 1024×1024 image.

We study the influence of the number of principal components used for feature preprocessing in Fig. 8. Naturally, when using a
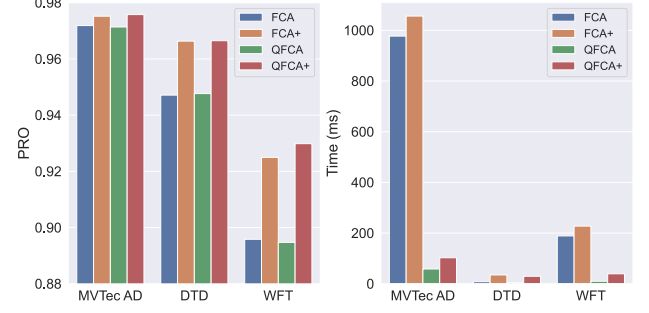

Figure 7: *Using our feature preprocessing in conjunction with FCA and QFCA consistently improves the localization performance at the cost of a slightly increased runtime.*
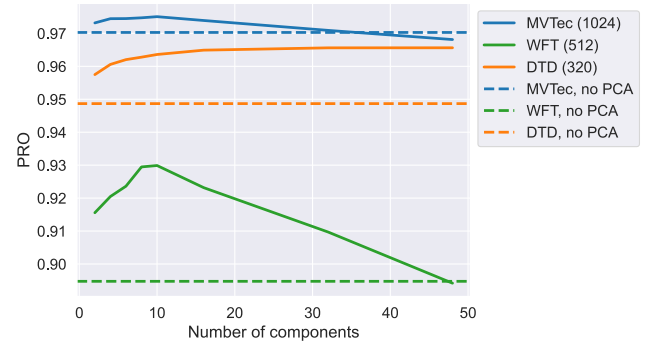

Figure 8: *Visualizing the impact of the number of principal components used in preprocessing in terms of PRO score. Dashed lines show the baseline values, without feature preprocessing.*

very small number of components, PCA can scarcely reconstruct the input features, and the residuals do not properly reflect the rarity of the features. On the other hand, too many principal components allow even infrequent features to be reconstructed well, annulling the benefit of our preprocessing. We observe that the optimum number of components depends on the complexity of the images (*i.e.*, larger size in MVTec, and intricate textures in DTD-Synthetic). Nonetheless, 10 principal components proved to be a robust off-the-shelf choice in our experiments.

4.5.3. Computing the Reference

The reference set $R(F(I))$ describes the global properties across all patches, and it is compared with all patches during error computation

Method	PRO	AUROC$_s$	F$_1$
quan → mean	96.9	98.6	71.3
quan → median	97.0	**98.7**	71.6
mean → quan	97.0	98.6	71.3
median → quan	**97.1**	**98.7**	**71.9**

Table 3: *Comparing different reference selection methods.*

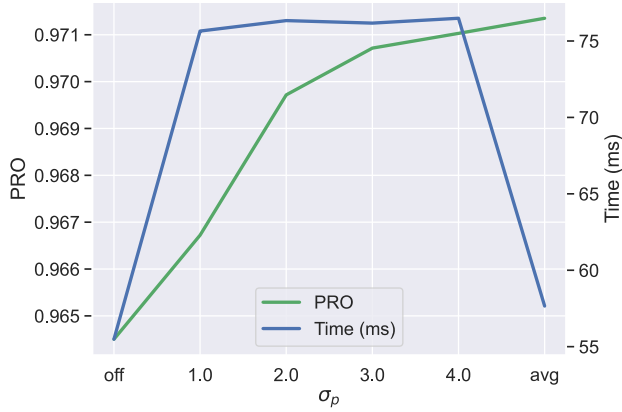**Figure 9:** *Analyzing the effect of σ_p for spatially weighing the mismatch scores. Experiment on the MVTec dataset at 1024×1024.*

Figure 10: *The effect of the final low-pass filter on the PRO score. The running time of this operation is negligible.*

(Sec. 3.1). There are multiple ways to determine this reference tensor; the FCA paper tested random selection, computing the mean, computing the median, and k-NN. The median was found to produce the best results in the case of a single reference and k-NN to work best when using multiple references. Since our work prioritizes execution time, we only consider a single reference F_r.

QFCA has a quantization step before creating the histograms that represent feature patches. The mean or median of the patch representations can be computed before or after the feature quantization. Table 3 presents the results obtained with different reference selection procedures.

4.5.4. Blurring and Averaging Steps

Similarly to FCA, we use two different Gaussian blurring operations as part of the QFCA algorithm. The first Gaussian operation, denoted $G_{\sigma_p}(x, y)$ performs the same operations as the spatial weighing function in FCA. That is, it aggregates the contribution to error in the patch statistics comparison from the corresponding histogram bin of all surrounding patches. Therefore, the kernel size of the Gaussian filter must be equal to the patch size.

Figure 9 shows a range of possible values for σ_p. The plot reports the results on the MVTec dataset for clarity; however, the DTD-Synthetic dataset and WFT follow the same trend. It can be seen that the accuracy increases with σ_p and peaks when using a simple average pooling, corresponding to $\sigma_p = \infty$, given the fixed kernel size. This suggests that all contexts (patches) a pixel is part of are equally important.

The last step of our algorithm is a 2D blurring step, with standard deviation σ_s, over the aggregated error contribution scores. We use the same name σ_s as in FCA because this parameter has the same role as the Gaussian filtering $\mathcal{G}_{\sigma_s}(\cdot)$ used there. Although not mathematically equivalent, our final blurring serves the same purpose and is significantly more efficient to compute. We show the impact of our σ_s in Fig. 10. Based on these observations, we choose $\sigma_s = 1$ as our default parameter.
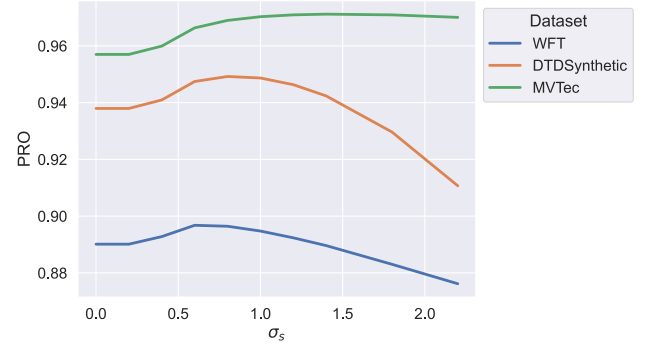
5. Limitations

As an optimized algorithm for zero-shot texture anomaly localization, our approach inherits the limitations of this class of methods. Namely, as we do not use a large VLM to inject general knowledge about anomalies, QFCA is suitable for texture-like data, and not for arbitrary objects. Nonetheless, thanks to the feature preprocessing, our method brings a significant boost even in this case, compared to FCA. Please see the supplementary (S3) for an evaluation on such non-textural images.

6. Conclusion

In this work, we introduce QFCA+, a real-time anomaly localization method that is not only faster than existing zero-shot approaches, but also detects anomalies with higher precision. More precisely, as seen in Fig. 1, our QFCA+ offers a more advantageous tradeoff between localization fidelity and runtime. By replacing the slow sorting operation in FCA with a GPU-optimized implementation on histograms, we obtain a 10× speedup. Notably, the quantization is able to match the full-precision metrics using just 16 bins. Moreover, we further speed up the method using a fast average pooling algorithm, with a time complexity invariant to kernel size, which can potentially be included in other pipelines to gain similar speedup benefits.

We additionally propose a feature preprocessing step (the + in QFCA+), which can be used to increase the accuracy of anomaly localization when a slower latency can be afforded. This addition is beneficial for complex textures and can be used in conjunction with other anomaly detection algorithms.

References

[AAAB19] AKCAY S., ATAPOUR-ABARGHOUEI A., BRECKON T. P.: Skip-ganomaly: Skip connected and adversarially trained encoder-decoder anomaly detection. *2019 International Joint Conference on Neural Networks (IJCNN)* (July 2019). doi:10.1109/ijcnn.2019.8851808. 2

[AC15] AN J., CHO S.: Variational autoencoder based anomaly detection using reconstruction probability. *Special Lecture on IE 2*, 1 (2015). 2

[Agg16] AGGARWAL C. C.: *Outlier Analysis*, 2 ed. Springer International Publishing, Cham, Switzerland, 12 2016. 1

[AMI16] AHMED M., MAHMOOD A. N., ISLAM M. R.: A survey of anomaly detection techniques in financial domain. *Future Gener. Comput. Syst. 55* (2 2016), 278–288. 1

[ATO23] AOTA T., TONG L. T. T., OKATANI T.: Zero-shot versus many-shot: Unsupervised texture anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Jan. 2023), pp. 5564–5572. 2, 3, 5, 6

[AW24a] ARDELEAN A.-T., WEYRICH T.: Blind localization and clustering of anomalies in textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops* (June 2024), pp. 3900–3909. 2, 4

[AW24b] ARDELEAN A.-T., WEYRICH T.: High-fidelity zero-shot texture anomaly localization using feature correspondence analysis. In *Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (Jan. 2024). 1, 2, 4, 5, 6, 3

[BBF*21] BERGMANN P., BATZNER K., FAUSER M., SATTLEGGER D., STEGER C.: The mvtec anomaly detection dataset: A comprehensive real-world dataset for unsupervised anomaly detection. *International Journal of Computer Vision 129*, 4 (4 2021), 1038–1059. 5

[BCH20] BERGMAN L., COHEN N., HOSHEN Y.: Deep nearest neighbor anomaly detection. *CoRR abs/2002.10445* (2020). 2

[BFSS19] BERGMANN P., FAUSER M., SATTLEGGER D., STEGER C.: Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 9584–9592. doi:10.1109/CVPR.2019.00982. 5

[BHK24] BATZNER K., HECKLER L., KÖNIG R.: Efficientad: Accurate visual anomaly detection at millisecond-level latencies. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)* (January 2024), pp. 128–138. 2

[BLF*19] BERGMANN P., LÖWE S., FAUSER M., SATTLEGGER D., STEGER C.: Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In *VISIGRAPP (5: VISAPP)* (2019), Trémeau A., Farinella G. M., Braz J., (Eds.), SciTePress, pp. 372–380. 2, 5

[BSZ*21] BIBI R., SAEED Y., ZEB A., GHAZAL T. M., RAHMAN T., SAID R. A., ABBAS S., AHMAD M., KHAN M. A.: Edge AI-based automated detection and classification of road anomalies in VANET using deep learning. *Comput. Intell. Neurosci. 2021*, 1 (Sept. 2021), 6262194. 1

[BWAN19] BAUR C., WIESTLER B., ALBARQOUNI S., NAVAB N.: Deep autoencoding models for unsupervised anomaly segmentation in brain mr images. In *Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part I 4* (2019), Springer, pp. 161–169. 2

[CBK09] CHANDOLA V., BANERJEE A., KUMAR V.: Anomaly detection: A survey. *ACM Comput. Surv. 41*, 3 (7 2009). doi:10.1145/1541880.1541882. 1

[CH21] COHEN N., HOSHEN Y.: Sub-image anomaly detection with deep pyramid correspondences, 2021. URL: https://arxiv.org/abs/2005.02357, arXiv:2005.02357. 2

[Chi92] CHINCHOR N.: Muc-4 evaluation metrics. In *Fourth Message Understanding Conference 22–29* (1992), Morgan Kaufmann. 5

[CHZ23] CHEN X., HAN Y., ZHANG J.: A zero-/few-shot anomaly classification and segmentation method for cvpr 2023 vand workshop challenge tracks 1&2: 1st place on zero-shot ad and 4th place on few-shot ad. *arXiv preprint arXiv:2305.17382* (2023). 2, 5, 6, 3

[CMK*14] CIMPOI M., MAJI S., KOKKINOS I., MOHAMED S., VEDALDI A.: Describing textures in the wild. In *2014 IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 3606–3613. doi:10.1109/CVPR.2014.461. 5

[Cro84] CROW F. C.: Summed-area tables for texture mapping. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques* (1984), pp. 207–212. 4

[CXS*23] CAO Y., XU X., SUN C., CHENG Y., DU Z., GAO L., SHEN W.: Segment any anomaly without training via hybrid prompt regularization. *arXiv preprint arXiv:2305.10724* (2023). 2, 5, 3

[CZG*20] COOPER C., ZHANG J., GAO R. X., WANG P., RAGAI I.: Anomaly detection in milling tools using acoustic signals and generative adversarial networks. *Procedia Manufacturing 48* (2020), 372–378. 48th SME North American Manufacturing Research Conference, NAMRC 48. doi:https://doi.org/10.1016/j.promfg.2020.05.059. 1

[CZT*24] CHEN X., ZHANG J., TIAN G., HE H., ZHANG W., WANG Y., WANG C., LIU Y.: Clip-ad: A language-guided staged dual-path model for zero-shot anomaly detection. In *International Joint Conference on Artificial Intelligence* (2024), Springer, pp. 17–33. 2, 5, 3

[DDS*09] DENG J., DONG W., SOCHER R., LI L.-J., LI K., FEI-FEI L.: Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (2009), pp. 248–255. doi:10.1109/CVPR.2009.5206848. 2

[DSLA21] DEFARD T., SETKOV A., LOESCH A., AUDIGIER R.: Padim: a patch distribution modeling framework for anomaly detection and localization. In *International conference on pattern recognition* (2021), Springer, pp. 475–489. 2

[DZBL23] DENG H., ZHANG Z., BAO J., LI X.: Anovl: Adapting vision-language models for unified zero-shot anomaly localization. *arXiv preprint arXiv:2308.15939 2*, 5 (2023). 3

[EW22] ELNEKAVE A., WEISS Y.: Generating natural images with direct patch distributions matching. In *European Conference on Computer Vision* (2022), Springer, pp. 544–560. 2

[FBH05] FOGARTY J., BAKER R. S., HUDSON S. E.: Case studies in the use of roc curve analysis for sensor-based estimates in human computer interaction. In *Proceedings of Graphics Interface 2005* (Waterloo, CAN, 2005), GI '05, Canadian Human-Computer Communications Society, p. 129–136. 5

[FGD*22] FERNANDO T., GAMMULLE H., DENMAN S., SRIDHARAN S., FOOKES C.: Deep learning for medical anomaly detection-a survey. *ACM Comput. Surv. 54*, 7 (2022), 141–1. 1

[GZY*25] GAO B.-B., ZHU Y., YAN J., CAI Y., ZHANG W., WANG M., LIU J., LIU Y., WANG L., WANG C.: Adaptclip: Adapting clip for universal visual anomaly detection. *arXiv preprint arXiv:2505.09926* (2025). 5, 6

[HA04] HODGE V. J., AUSTIN J.: A survey of outlier detection methodologies. *Artif. Intell. Rev. 22*, 2 (Oct. 2004), 85–126. 2

[HCH19] HSIEH R.-J., CHOU J., HO C.-H.: Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing. In *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)* (11 2019), IEEE. 1

[HGJ*22] HUANG C., GUAN H., JIANG A., ZHANG Y., SPRATLING M., WANG Y.-F.: Registration based few-shot anomaly detection. In *European conference on computer vision* (2022), Springer, pp. 303–319. 2

[HGJ*24] HUANG C., GUAN H., JIANG A., ZHANG Y., SPRATLING M., WANG X., WANG Y.: Few-shot anomaly detection via category-agnostic registration learning. *IEEE Transactions on Neural Networks and Learning Systems* (2024). 2

[JZK*23] JEONG J., ZOU Y., KIM T., ZHANG D., RAVICHANDRAN A., DABEER O.: Winclip: Zero-/few-shot anomaly classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (June 2023), pp. 19606–19616. 2, 5, 1, 3

[KGG18] KAUR P., GANGADHARAPPA M., GAUTAM S.: An overview of anomaly detection in video surveillance. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)* (10 2018), IEEE. 1

[LCY*25] LI S., CAO J., YE P., DING Y., TU C., CHEN T.: Clipsam: Clip and sam collaboration for zero-shot anomaly segmentation. *Neurocomputing 618* (2025), 129122. 2, 3

[LGLF24] LI Y., GOODGE A., LIU F., FOO C.-S.: Promptad: Zero-shot anomaly detection using text prompts. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (2024), pp. 1093–1102. 2

[LLC*23] LEE J. Y., LEE W., CHOI J., LEE Y., YOON Y. S.: Few-shot anomaly detection with adversarial loss for robust feature representations. In *BMVC* (2023). 2

[LLS22] LEE S., LEE S., SONG B. C.: Cfa: Coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. *IEEE Access 10* (2022), 78446–78454. 2

[LXW*24a] LIU J., XIE G., WANG J., LI S., WANG C., ZHENG F., JIN Y.: Deep industrial image anomaly detection: A survey. *Machine Intelligence Research 21*, 1 (1 2024), 104–135. doi:10.1007/s11633-023-1459-z. 1

[LXW*24b] LIU J., XIE G., WANG J., LI S., WANG C., ZHENG F., JIN Y.: Deep industrial image anomaly detection: A survey. *Machine Intelligence Research 21*, 1 (2024), 104–135. 2

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language supervision. In *International conference on machine learning* (2021), PMLR, pp. 8748–8763. 2

[RPZ*22] ROTH K., PEMULA L., ZEPEDA J., SCHÖLKOPF B., BROX T., GEHLER P.: Towards total recall in industrial anomaly detection. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), pp. 14298–14308. doi:10.1109/CVPR52688.2022.01392. 2, 4

[SAK*24] SCHWARTZ E., ARBELLE A., KARLINSKY L., HARARY S., SCHEIDEGGER F., DOVEH S., GIRYES R.: Maeday: Mae for few-and zero-shot anomaly-detection. *Computer Vision and Image Understanding 241* (2024), 103958. 2

[SM11] SMITH M. R., MARTINEZ T.: Improving classification accuracy by identifying and removing instances that should be misclassified. In *The 2011 International Joint Conference on Neural Networks* (July 2011), IEEE. 1

[SSH*25] SALEHI A., SALEHI M., HOSSEINI R., SNOEK C. G., YAMADA M., SABOKROU M.: Crane: Context-guided prompt learning and attention refinement for zero-shot anomaly detections. *arXiv preprint arXiv:2504.11055* (2025). 5

[TGZ*22] TAO X., GONG X., ZHANG X., YAN S., ADAK C.: Deep learning for unsupervised anomaly localization in industrial images: A survey. *IEEE Transactions on Instrumentation and Measurement 71* (2022), 1–21. doi:10.1109/tim.2022.3196436. 1, 2

[Tom76] TOMEK I.: An experiment with the edited nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics SMC-6*, 6 (1976), 448–452. doi:10.1109/TSMC.1976.4309523. 1

[VJ01] VIOLA P., JONES M.: Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001* (2001), vol. 1, Ieee, pp. I–I. 4

[WASA21] WIBISONO S., ANWAR M. T., SUPRIYANTO A., AMIN I. H. A.: Multivariate weather anomaly detection using dbscan clustering algorithm. *J. Phys. Conf. Ser. 1869*, 1 (4 2021), 012077. 1

[WKW*16] WU G., KIM M., WANG Q., MUNSELL B. C., SHEN D.: Scalable high-performance image registration framework by unsupervised deep feature representations learning. *IEEE Transactions on Biomedical Engineering 63*, 7 (2016), 1505–1516. doi:10.1109/TBME.2015.2496253. 4

[YLC*24] YAO H., LUO W., CAO Y., ZHANG Y., YU W., SHEN W.: Global-regularized neighborhood regression for efficient zero-shot texture anomaly detection. *arXiv preprint arXiv:2406.07333* (2024). 5, 6, 3

[YLH*22] YANG C., LAN S., HUANG W., WANG W., LIU G., YANG H., MA W., LI P.: A transformer-based gan for anomaly detection. In *International Conference on Artificial Neural Networks* (2022), Springer, pp. 345–357. 2

[YRPK19] YOUKACHEN S., RUCHANURUCKS M., PHATRAPOMNANT T., KANEKO H.: Defect segmentation of hot-rolled steel strip surface by using convolutional auto-encoder and conventional image processing. In *2019 10th International Conference of Information and Communication Technology for Embedded Systems (IC-ICTES)* (2019), pp. 1–5. doi:10.1109/ICTEmSys.2019.8695928. 2

[ZK16] ZAGORUYKO S., KOMODAKIS N.: Wide residual networks. *arXiv preprint arXiv:1605.07146* (2016). 4

[ZPT*23] ZHOU Q., PANG G., TIAN Y., HE S., CHEN J.: Anomalyclip: Object-agnostic prompt learning for zero-shot anomaly detection. In *The Twelfth International Conference on Learning Representations* (2023). 2

[ZSO24] ZHANG J., SUGANUMA M., OKATANI T.: That's bad: blind anomaly detection by implicit local feature clustering. *Machine Vision and Applications 35*, 2 (2024), 31. 2

[ZWL*24] ZHANG J., WANG C., LI X., TIAN G., XUE Z., LIU Y., PANG G., TAO D.: Learning feature inversion for multi-class anomaly detection under general-purpose coco-ad benchmark. *arXiv preprint arXiv:2404.10760* (2024). 5