Image novelty detection based on mean-shift and typical set size[⋆]

Matthias Hermann[1], Bastian Goldlücke[2], and Matthias O. Franz[1]

[1] HTWG Konstanz, Konstanz, Germany
[2] Universität Konstanz, Konstanz, Germany

Abstract. The detection of anomalous or novel images given a training dataset of only clean reference data (inliers) is an important task in computer vision. We propose a new shallow approach that represents both inlier and outlier images as ensembles of patches, which allows us to effectively detect novelties as mean shifts between reference data and outliers with the Hotelling T^2 test. Since mean-shift can only be detected when the outlier ensemble is sufficiently separate from the typical set of the inlier distribution, this typical set acts as a *blind spot* for novelty detection. We therefore minimize its estimated size as our selection rule for critical hyperparameters, such as, e.g., the size of the patches is crucial. To showcase the capabilities of our approach, we compare results with classical and deep learning methods on the popular datasets MNIST and CIFAR-10, and demonstrate its real-world applicability in a large-scale industrial inspection scenario.

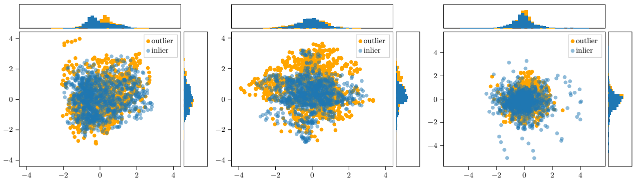
Keywords: Image novelty detection · independent component analysis · mean-shift.

1 Introduction

Novelty detection is a semi-supervised approach to anomaly detection where all available training data belongs to a single class. The task is to learn the class boundary of the reference class such that the model can classify test data into known (inliers) and novel examples (outliers). Such models output a score based on a single input example that can be used for classification. As a consequence, the decision process is not robust against noise contamination or overlapping distributions between inliers and outliers.

This motivates the main idea of our approach to novelty detection: representing both training and test images as ensembles of image patches. Instead of establishing a relation of a single test data point to an inlier distribution, this allows us to compare the training and test ensembles against each other which is inherently more robust. The literature provides a broad range of statistics for testing whether two datasets originate from the same distribution. Here, we use the Hotelling T^2 test [6] for this purpose. It is based on computing the mean shift between the training and test ensembles which is particularly simple to compute and therefore suitable for large datasets.

[⋆] Supported by Bundesministerium für Bildung und Forschung (BMBF, 01IS19083A)



(a) Horses (b) Ships (c) Birds

Fig. 1: Patch distribution of the two largest principal components of CIFAR-10 showcasing different mean shifts between reference data (blue) and anomalies (yellow). Note the considerable overlap of both distributions.

In our experiments, we found that the success of this approach critically depends on the details of how the patch ensemble is extracted from the input images. The most important parameters are the number and the size of the patches and the image basis in which the patches are represented. These parameters depend on the specific dataset and thus have to be found by an automatic model selection procedure. Our approach is based on the observation that a mean shift can only be detected when the outlier ensemble is sufficiently separate from the *typical set* of the inlier distribution. The typical set has a total probability close to one which is a consequence of the asymptotic equipartition property [AEP; 3]. Any outlier which falls inside the volume occupied by the typical set has no chance of being detected (see Fig. 1). Thus, the typical set acts as a *blind spot* for novelty detection which needs to be kept as small as possible. By estimating the volume of the typical set, an optimal parameter set can be found for each dataset.

In this work, we contribute a shallow algorithm based on the Hotelling T^2 test and independent component analysis (ICA) for image novelty detection with overlapping distributions. Notably, the hyperparameters of our model are selected using typical set theory for finding a patch ensemble which optimally represents the input images. We show in extensive experiments on raw pixel data that our approach not only achieves comparable results to Deep Learning approaches on MNIST and CIFAR-10, but is also applicable to a large-scale industrial inspection scenario, due to its simple architecture and fast predictions[3].

2 Related work

We refer the reader to [14] for a good overview over the general techniques for novelty detection and focus on images in the following. In terms of image novelty detection, current approaches rely on compression [1], generative models [4], feature engineering [18], or known statistics of images [7]. Well-known instances of the compression class are variational Autoencoders [9] or probabilistic PCA [17]. Besides, there exist autoencoders that are specialized to novelty detection, such

[3] `https://github.com/matherm/mean-shift`

as Latent Space Autoregression [1]. Density-based approaches model the training data with a statistical parametric model $\mathbf{x} \sim \mathbf{p}_\theta(\mathbf{x})$, where the parameters θ are learned by maximizing the likelihood function [4]. Here, the densities of input examples $\mathbf{p}_\theta(\mathbf{x})$ are directly available, and anomalies can be classified by using this density as a scoring rule. Kernel space methods project the data into a feature space $\mathcal{F}$ by computing a nonlinear mapping $\phi(\mathbf{x})$. Anomaly detection is conducted in feature space, where typically the norm of $||\phi(\mathbf{x})||$ is used as scoring function [15]. Natural image statistics provide image priors that can be used to derive suitable feature spaces for novelty detection [7]. Our approach is related in the sense that we also optimize within the independent components (ICA) framework, although our starting point is quite different. In contrast to existing methods, we propose to transform the input images into patches and to compute statistics of patch ensembles. A similar approach was proposed by [11] in an out-of-distribution scenario, where they used a hypothesis testing framework to test for typicality. However, they analyzed ensembles of multiple input images, instead of ensembles of patches of a single input image, and they did not optimize the typical set size.

3 Data preparation

Our method first transforms a given data set of reference input images $\mathbf{I}_0, \cdots, \mathbf{I}_N$ into an ensemble $\mathbf{X}$ of patches. We describe these preparatory steps in this section. For a given test image $\mathbf{I}^*$, we apply the same preprocessing and extract an ensemble $\mathbf{x}^*$ of patches. We then measure the mean-shift of the ensemble mean $\boldsymbol{\mu}(\mathbf{x}^*)$ with respect to the mean of all given training patches $\boldsymbol{\mu}(\mathbf{X})$. The anomaly score $\mu\text{shift}(\mathbf{x})$ is based on the Hotelling T^2 test [6] and derived in section 4. Finally, we describe how to automatically select optimal hyperparameters for the method in section 5.

We process the input examples $\mathbf{I}_i(x, y)$ on a common scale and apply contrast normalization as preprocessing [7]. The normalization step centers and projects all given input examples onto the unit sphere with respect to the L_2-norm. This is achieved by first removing the pixel-wise mean and rescaling afterwards,

$$\mathbf{I}' = \frac{\mathbf{I} - \frac{1}{D}\sum_{x,y}\mathbf{I}(x,y)}{\left\|\mathbf{I} - \frac{1}{D}\sum_{x,y}\mathbf{I}(x,y)\right\|_{L_2}}, \tag{1}$$

where D is the number of pixels - treating colors as additional pixels - in the image. To ensure the numerical stability of the algorithm, we globally divide all examples by the standard deviation $\text{std}(\mathbf{I}')$ over all of the N training examples.

Having a set of preprocessed images $\mathcal{I} = \{\mathbf{I}'_0/s, \ldots, \mathbf{I}'_N/s\}$, the important part of our algorithm is to generate patch ensembles, instead of processing the full image. There are several possible strategies for cropping square image patches from an image and we tested different sampling strategies without noticing significant differences. Therefore, we propose to simply crop the patches in a sliding window fashion and extract all *valid* patches inside the image without crossing the border. The horizontal and vertical stride τ of the sliding window helps controlling the total number of cropped patches. We did not notice a performance-critical impact of this parameter and keep it fixed to $\tau = 2$. This means that the maximum number S of distinct image patches per input image is only limited by the size D of the image and the patch size P, i.e., the larger the patch size, the fewer patches can be extracted. To distinguish between an input data point and ensembles of patches, we denote a single patch of the i-th example by $\mathbf{x}_i$ and use $\mathbf{x}_i(s)$ for indexing the ensemble where necessary. We flatten the extracted patches with c color channels and organize the vectors of all computed reference patches in a long design matrix $\mathbf{X} \in \mathbb{R}^{NS \times M}$, where $M = cP^2$.

4 Mean-shift detection

We perform mean-shift detection with the Hotelling T^2 test [6]. This is a multivariate generalization of Student's t-test and allows for computing the significance of mean-shifts between two populations. First, we present the test in its classical form. In the second part, we derive a feature space interpretation that reveals relevant hyperparameters that are needed for model selection in section 5.

In our case, we compare the pixel-wise mean

$$\boldsymbol{\mu} = \frac{1}{NS} \sum_{i}^{N} \sum_{s}^{S} \mathbf{x}_i(s) \tag{2}$$

computed over all training patches with the pixel-wise mean

$$\boldsymbol{\mu}^* = \frac{1}{S} \sum_{s}^{S} \mathbf{x}^*(s). \tag{3}$$

of the patches $\mathbf{x}^*$ extracted from a single test example. The unnormalized Hotelling T^2 test statistic for a dependent test sample is given by

$$\tilde{T}^2 = (\boldsymbol{\mu}^* - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{\mathbf{-1}} (\boldsymbol{\mu}^* - \boldsymbol{\mu}), \tag{4}$$

where

$$\boldsymbol{\Sigma} = \frac{1}{NS - 1} (\mathbf{X} - \boldsymbol{\mu})^T (\mathbf{X} - \boldsymbol{\mu}) \tag{5}$$

is the covariance matrix of the training dataset $\mathbf{X}$. In other words, the Hotelling $\tilde{T}^2$ statistic is the Mahalanobis distance between the two mean vectors. Note that because our samples are equally sized, we neglect the constant normalization factor $\frac{NS^2}{NS+S}$ for simplicity.

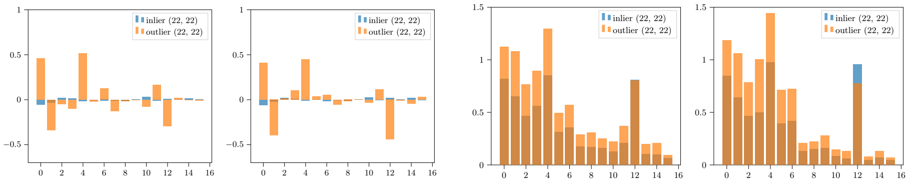
To obtain a feature space interpretation, the Mahalanobis distance can be computed by first whitening the data with a whitening transformation $\mathbf{A}$ and then computing the standard L_2 distance of the whitened mean vectors. This allows us to reveal relevant hyperparameters, such as the noise floor and the rotation freedom. Due to the linearity of $\mathbf{A}$, this is equivalent to applying $\mathbf{A}$ to the mean difference vector in the original pixel space:

$$\tilde{T}^2 = \left\| \mathbf{A}(\boldsymbol{\mu}^* - \boldsymbol{\mu}) \right\|_{L_2} \tag{6}$$

The whitening transformation $\mathbf{A}$ can be decomposed into an orthogonal matrix $\mathbf{W}$ containing the Eigenvectors of the covariance matrix $\boldsymbol{\Sigma}$ as columns, a diagonal scaling matrix $\mathbf{S}^{-1/2}$, and an arbitrary rotation matrix $\mathbf{R}$, such that

$$\mathbf{A} = \mathbf{R}\mathbf{S}^{-1/2}\mathbf{W}, \tag{7}$$

with $\boldsymbol{\Sigma} = \mathbf{W}^{\mathbf{T}}\mathbf{S}\mathbf{W}$, $\mathbf{W}\mathbf{W}^{\mathbf{T}} = \mathbf{I}$, and $\mathbf{R} \in \mathrm{SO}(M)$. The matrix $\mathbf{S}$ consists of the variances $s_i, \cdots, s_M$ along the components in $\mathbf{W}$.



(a) $\mathbb{E}[\phi]$

(b) $\mathrm{Var}[\phi]$

Fig. 2: Mean-shifts and variances of the first 16 principal components of ϕ for the deer class of CIFAR-10 with patch size $P = 22$. The left half of (a) and (b) shows the statistics with unoptimized rotation matrix, the right half after optimization.

In this transformation, we also reduce the data dimension M to k by removing the noise floor. This is done by truncating the matrices $\mathbf{W}$ and $\mathbf{S}$ by removing the dimensions with smallest variance. Across experiments, we found it helpful to control the number of informative variables k by a rule, instead of fixing the number of retained features. We retain all components up to a fixed threshold of explained variance [7], in our case 90%. We denote this number by $k = k_{90}$. Furthermore, it is known that whitened data stay whitened under rotation, so we can apply an arbitrary rotation matrix $\mathbf{R}$ without changing the $\tilde{T}^2$ statistic. We will cover the choice of this rotation freedom in more detail in section 5 and show why it is crucial for model selection.

For visualization, it is useful to decompose the $\tilde{T}^2$ statistic into the feature vector

$$\phi(\mathbf{x}) = \mathbf{A}\boldsymbol{\mu}(\mathbf{x}) \tag{8}$$

and the derived anomaly score

$$\mu\mathrm{shift}(\mathbf{x}) = \left\| \phi(\mathbf{x}) - \phi(\mathbf{X}) \right\|_{L_2}. \tag{9}$$

Fig. 1 depicts the two largest components of $\phi(\mathbf{x})$ and $\phi(\mathbf{x}^*)$ of the CIFAR-10 dataset showing the large overlap between both distribution. Nevertheless, a mean shift is still detectable in some components despite of the large variance of the feature vectors $\phi(\mathbf{x})$, see Fig. 2.

5 Typical set size minimization

For learning the model, we need to find a suitable whitening matrix $\mathbf{A}$ of the image patches. The matrices $\mathbf{S}^{-1/2}$ and $\mathbf{W}$ can be computed by standard Eigenvector decomposition of the covariance matrix $\mathbf{\Sigma}$. However, during this procedure, also the remaining hyperparameters $\Theta = \{P, \mathbf{R}\}$ need to be set. Again, P is the patch size, and $\mathbf{R}$ the arbitrary rotation. Fig. 3 shows the visual impact of the patch size P on the ensemble statistics. Depending on the size, the appearance of the classes changes drastically, particularly in terms of dissimilarity between neighboring image patches. This observation motivates the use of an entropy-related measure of dissimilarity or disorder for hyperparameter selection. Since we do not observe the outliers, we can only manipulate the statistics of the transformed reference data. In the introduction, we argued that a good strategy for model selection is to keep the size $|\mathcal{A}(\cdot)|$ of the typical set as small as possible as this limits the blind spot of the mean shift detection mechanism. A central relationship between the size of the typical set and the entropy of the feature distribution [3, 11] is

$$\log|\mathcal{A}(\phi)| \leq f(\mathcal{H}[\phi]), \tag{10}$$

where $f(\cdot)$ is a monotonically increasing function which satisfies certain constraints. This means that in order to keep the *blind spot* small, we need to minimize the entropy of ϕ .

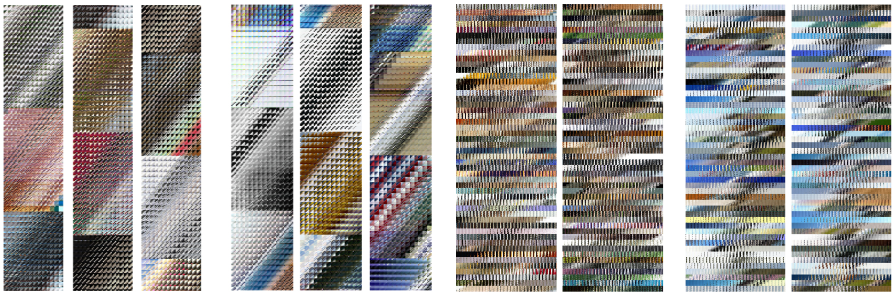
Directly minimizing entropy of stochastic variables is heavily studied in the field of sparse coding and Independent Component Analysis [ICA; 2]. A central measure in that field is the so-called negentropy, which is the negative of entropy. Negentropy has an appealing feature that arises from the maximum entropy principle, i.e., given a fixed variance the maximum entropy distribution is a Gaussian [3]. This relation can be utilized by the construction of a negentropy approximation [2] that uses the Gaussian distribution as contrast

$$\mathcal{J}[\phi] \propto \sum_{i}^{k} (g(\phi_i) - g(\gamma))^2, \tag{11}$$

where $g = \log\cosh(\cdot)$, $\gamma \sim \mathcal{N}(0,1)$, and ϕ is centered. As a consequence, the model selection rule simplifies to a linear search over the patch size P and a non-convex optimization of the rotation matrix $\mathbf{R}$,

$$\arg\max_{P,\mathbf{R}} \mathcal{J}[\phi], \tag{12}$$

with $P \in [14, \sqrt{D/c} - \tau]$ and $\mathbf{R} \in \mathrm{SO}(k)$. We chose 14 as minimum patch size to avoid the pathological case of selecting too small patches containing zero image content, such as black spots in MNIST. Again, the rotation matrix $\mathbf{R}$ needs only to be optimized as this model freedom highly impacts the negentropy measure, but does not change the Mahalanobis distance (Eq. 4). While P is found by a grid search, optimizing the rotation matrix $\mathbf{R}$ is a non-convex problem, in



(a) Cats $(14, 14)$, $\mathcal{J}[\phi] = 0.019$ (b) Planes $(14, 14)$, $\mathcal{J}[\phi] = 0.030$ (c) Cats $(30, 30)$, $\mathcal{J}[\phi] = 0.017$ (d) Planes $(30, 30)$, $\mathcal{J}[\phi] = 0.031$

Fig. 3: Patch ensembles from two classes of CIFAR-10, cats and planes, with two different patch sizes. The plane class has a more homogeneous appearance (e.g. blue sky), while the cat class is more chaotic (e.g. cat pose) yielding a smaller negentropy $\mathcal{J}$.

particular, the solution is constrained to be an orthogonal matrix. As a first step, we decompose the gradient via the chain rule,

$$\nabla_{\mathbf{R}}\mathcal{J} = \nabla_{\phi}\mathcal{J}\nabla_{\mathbf{R}}\phi, \tag{13}$$

where $\nabla_{\phi}\mathcal{J}$ is a $1 \times k$ row vector and $\nabla_{\mathbf{R}}\phi$ is a $k \times k^2$ matrix. We organize the resulting $1 \times k^2$ gradient $\nabla_{\mathbf{R}}\mathcal{J}$ as $k \times k$ matrix for further processing. Note that our problem is different to standard ICA, as we are computing the gradient w.r.t. to the average across multiple patches (cf. Eq. 8), instead of a single patch.

The orthogonality constraint can be enforced by performing gradient ascent only inside the Lie group $\mathrm{SO}(k)$ [*gradient flow*; 12]. Hence, the gradient $\nabla_{\mathbf{R}}\mathcal{J}$ of the loss function represents an infinitesimal rotation which has the form of a skew-symmetric matrix. The set of all skew-symmetric matrices is called the Lie algebra $\mathfrak{so}(k)$ associated with the Lie group $\mathrm{SO}(k)$. Every skew-symmetric matrix $\mathbf{\Theta}$ can be uniquely parameterized by a vector $\mathbf{r}$ of dimension $k(k-1)/2$, denoted as the Plücker coordinates. The rotation matrix $\mathbf{R}$ associated to its generating skew-symmetric matrix $\mathbf{\Theta}$ is $\mathbf{R} = \exp(\mathbf{\Theta})$.

In order to compute a valid gradient step beyond the neighborhood of $\mathbf{R}$, the gradient direction needs to be expressed by the Lie bracket. We refer to [12] for the full derivation of the relation between the two gradient expressions using the commutator

$$\nabla_{\mathbf{\Theta}}\mathcal{J} = \left(\nabla_{\mathbf{R}}\mathcal{J}\right)^{T}\mathbf{R} - \mathbf{R}^{T}\left(\nabla_{\mathbf{R}}\mathcal{J}\right). \tag{14}$$

We initially set $\mathbf{R}_0 \sim \mathcal{N}(0, \mathbf{I})$ and compute the feature vectors ϕ_i using Eq. 8. Afterwards we compute the negentropy $\mathcal{J}$ using Eq. 11 and the gradient w.r.t. $\mathbf{R}$ using Eq. 14. From there, we compute the corresponding parameter vector $\mathbf{r}$ for gradient ascent by taking the upper triangular matrix of $\nabla_{\mathbf{\Theta}}\mathcal{J}$ corresponding to the Plücker coordinates. Unfortunately, rotation matrices for $k > 2$ are not commutative. Hence we cannot make additive steps of ascent in $\mathfrak{so}(k)$ and need

	SVM	**KDE**	**VAE**	**LSA**	**DSVD**	**μshift**
plane	0.630	0.658	0.688	**0.735**	0.617	0.731
car	0.440	0.520	0.403	0.580	0.659	**0.711**
bird	0.649	0.657	0.679	**0.690**	0.508	0.498
cat	0.487	0.497	0.528	0.542	0.591	**0.609**
deer	0.735	0.727	0.748	**0.761**	0.609	0.582
dog	0.500	0.496	0.519	0.546	**0.657**	0.620
frog	0.725	**0.758**	0.695	0.751	0.677	0.724
horse	0.533	0.564	0.500	0.535	0.673	**0.718**
ship	0.649	0.680	0.700	0.717	0.759	**0.805**
truck	0.508	0.540	0.398	0.548	0.730	**0.751**
	0.586	0.610	0.586	0.641	0.648	**0.675**

Table 1: AUC on CIFAR-10.

	SVM	**KDE**	**VAE**	**LSA**	**DSVD**	**μshift**
0	0.988	0.885	**0.998**	0.993	0.980	0.997
1	0.999	0.996	**0.999**	0.999	0.997	0.993
2	0.902	0.710	0.962	0.959	0.917	**0.986**
3	0.950	0.693	0.947	0.966	0.919	**0.979**
4	0.955	0.844	0.965	0.956	0.949	**0.971**
5	0.968	0.776	0.963	0.964	0.885	**0.981**
6	0.978	0.861	0.995	0.994	0.983	**0.995**
7	0.965	0.884	0.974	**0.980**	0.946	0.973
8	0.853	0.669	0.905	0.953	0.939	**0.969**
9	0.955	0.825	0.978	**0.981**	0.965	0.977
	0.951	0.814	0.969	0.975	0.948	**0.982**

Table 2: AUC on MNIST.

to map between the Lie algebra and SO(k) in every iteration by using $\exp(\mathbf{\Theta})$. The *gradient flow* update rule is then given by

$$\mathbf{R}_{i+1}^T = \exp(\eta\, \mathbf{\Theta}_{\mathbf{r}_i})\, \mathbf{R}_i^T\,, \tag{15}$$

with step size η and the skew-symmetric matrix $\mathbf{\Theta}_{\mathbf{r}_i}$ parametrized by $\mathbf{r}_i$ at the i-th iteration step. For acceleration, we use the ADAM optimizer [8] with $\eta = 1e^{-2}$ and optimize for 20 epochs at maximum.

6 Evaluation

We compare our approach to the following standard algorithms: OC-SVM [10], Kernel Density Estimator (KDE), and Variational Autoencoder (VAE) [9]. Furthermore, we compare two algorithms especially designed for one-class classification and not rely on prior-knowledge or transfer learning: Latent Space Autoregression (LSA) [1][4] and Deep SVD (DSVD) [15][5]. We use the standard measure of area under the receiver operating characteristic curve (AUC) for evaluating performance [16].

First, we test the method on the CIFAR-10 challenge. For training, we use all examples from the training split of a single class. Inlier data is the corresponding test split of the particular class, consisting of 1000 examples. For the outlier data, the same number of examples is sampled from the other classes of the test split. Tab. 2 shows the results of the experiment. We see that on average μshift yields the best results on CIFAR-10. It is interesting to note that the bird class is difficult for DSVD and ours. We attribute this result to the almost orthogonal feature spaces of inliers and outliers in this case (see further discussion in Sect. 6.2). As a second example, we evaluate the methods on the MNIST challenge. Again, for training, we use all examples from the training split of a single class, consisting of 6000 examples. Inlier data is the corresponding test split from the same class, consisting of 1000 examples. For the outlier data the same number of examples is sampled from the other classes of the test split. Tab.

[4] `https://github.com/ChangYungHua/Latent-space-AR/`

[5] `https://github.com/lukasruff/Deep-SVDD-PyTorch`

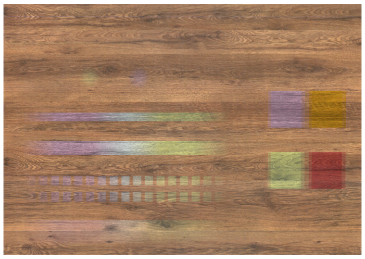


Fig. 5: Example of a scanned wooden texture with nozzle faults and color drifts from the digital printing dataset. Note that the nozzle faults are only 2px and therefore difficult to see.

	SVM	**KDE**	**VAE**	**LSA**	**DSVD**	**μshift**
0	0.812	0.785	0.802	0.717	0.910	**0.924**
1	0.513	0.542	0.484	0.614	**0.676**	0.652
2	0.591	0.501	0.606	0.499	0.648	**0.678**
3	0.534	**0.820**	0.584	0.700	0.775	0.609
4	0.615	0.563	0.487	0.523	**0.692**	0.616
5	0.646	0.545	0.645	0.541	0.594	**0.686**
6	0.573	**0.703**	0.573	0.551	0.671	0.646
7	0.441	0.518	0.572	0.529	**0.673**	0.571
8	0.838	0.559	0.802	0.433	0.545	**0.844**
9	0.609	0.676	0.743	0.647	0.736	**0.758**
	0.617	0.621	0.629	0.575	0.692	**0.698**

Table 3: AUC on the industrial digital printing dataset.

2 summarizes the results. The image statistics of MNIST are completely different from CIFAR-10. As a result, we see that a different set of algorithms performs better on this dataset. However, μshift consistently achieves the best results on the average.

Finally, we also test performance on an industrial digital printing dataset. The dataset consists of ten different wooden textures that are printed and scanned by an industrial inspection system at a resolution of 300 dpi. Fig. 5 shows the clean reference, an error-free scan, and a scan with printing anomalies, such as nozzle faults and color drifts. For training, the reference scan was divided into 7100 square non-overlapping 96×96 RGB images. The 3308 inlier examples were gathered in the same way from the additional error-free scan, the 3308 outlier examples from the anomalous scan. The results in Tab. 3 show a similar trend to CIFAR-10 and MNIST, confirming the comparatively good performance of μshift in a large-scale industrial inspection scenario with large patch sizes.

6.1 Runtime

Due to the minimalistic design of the algorithm, the algorithm can operate very efficiently on unseen test data. This is of particular importance for industrial use cases. The computation of the test scores for the digital printing dataset took about 800 ms (8270 FPS) on average using a single GPU 1080 GTX. The most expensive operations are the extraction of the S patches and the single $k \times M$ matrix multiplication. Note, that both operations can be implemented efficiently with a single 2D convolution plus a spatial averaging operation.

6.2 Limitations

The way we construct our features before applying the Hotelling T^2 test assumes that the feature spaces of the reference and outlier examples are largely overlapping and that there is a single mode. If this is not the case, it might well happen that many outliers are located in orthogonal directions with respect to the feature

	Cats vs. 3 ($\mathcal{J}[\phi]$)	3 vs. Cats ($\mathcal{J}[\phi]$)
$P = 14$	0.36 (0.020)	1.00 (0.005)
$P = 18$	0.15 (0.016)	0.99 (0.005)
$P = 22$	0.12 (0.024)	1.00 (0.008)
$P = 26$	0.11 (0.031)	1.00 (0.008)

Table 4: AUC of mean-shift detection for non-overlapping feature spaces.

space of the inliers. As a result, many features of these examples are mapped to the null space of the whitening transform such that they cannot contribute to a detectable mean shift. We already suspected that this effect might be responsible for the poor performance of μshift on the bird class example in Tab. 2 since the bird class in CIFAR-10 is very different from the other classes. In order to confirm this intuition, we created an artificial test problem where we took our references and inliers from the cats class of CIFAR-10 whereas the outliers came from the three class of MNIST. Here, the outliers occupy a very low-dimensional feature subspace that accounts for only a small proportion of the variance in the reference class. Tab. 4 clearly shows that a mean-shift detection does not work in this case. Interestingly, when reversing directions and using MNIST as the reference class, mean-shift detection becomes possible again, because the relatively rich feature space of the cats class has enough variance also in the small subspace occupied by the MNIST features.

7 Conclusion

In this work, we propose a new algorithm for the task of image novelty detection based on mean-shifts between the reference and the overlapping outlier patch distributions. The decision function is based on comparing patch ensembles instead of entire images which leads to an increased robustness and sensitivity of the algorithm. The chosen patch size and representation turns out to be the critical parameters for the success of this approach. A choice of these parameters based on minimizing the size of the critical set leads to satisfactory results that can even surpass deep learning methods. Our experiments show consistent applicability over multiple datasets, both in standard novelty detection tasks and in an industrial application scenario. Moreover, the computational load of the proposed algorithm is relatively small which recommends its application to large scale problems. A limitation of our method arises when the feature spaces of inliers and outliers do not overlap or multiple modes appear. This causes the outlier features to fall into the null-space of the whitening transform which makes mean-shift detection impossible. For practical applications, domain knowledge needs to be used to decide whether this special case prevails in the problem at hand. If so, mean-shift detection on raw pixels is not desirable and utilizing prior knowledge, feature extractors (e.g., [5, 13]) or models based on other loss functions, such as reconstruction loss, are better suited.

References

[1] Abati, D., Porrello, A., Calderara, S., Cucchiara, R.: Latent space autoregression for novelty detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 481–490 (2019)

[2] Comon, P.: Independent component analysis, a new concept? Signal processing **36**(3), 287–314 (1994)

[3] Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)

[4] Dinh, L., Sohl-Dickstein, J., Bengio, S.: Density estimation using real nvp. In: ICLR (Poster) (2016)

[5] Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. arXiv preprint arXiv:1912.02781 (2019)

[6] Hotelling, H.: The generalization of student's ratio. In: Breakthroughs in statistics, pp. 54–65. Springer (1992)

[7] Hyvärinen, A., Hurri, J., Hoyer, P.O.: Natural image statistics: A probabilistic approach to early computational vision., vol. 39. Springer Science & Business Media (2009)

[8] Kingma, D.P., Ba, J.L.: Adam: A method for stochastic optimization. In: ICLR 2015 : International Conference on Learning Representations 2015 (2015)

[9] Kingma, D.P., Welling, M.: Stochastic gradient vb and the variational autoencoder. In: Second International Conference on Learning Representations, ICLR. vol. 19 (2014)

[10] Manevitz, L.M., Yousef, M.: One-class svms for document classification. Journal of machine Learning research **2**(Dec), 139–154 (2001)

[11] Nalisnick, E., Matsukawa, A., Teh, Y.W., Lakshminarayanan, B.: Detecting out-of-distribution inputs to deep generative models using a test for typicality. arXiv preprint arXiv:1906.02994 **5**, 5 (2019)

[12] Plumbley, M.D.: Geometrical methods for non-negative ica: Manifolds, lie groups and toral subalgebras. Neurocomputing **67**, 161–197 (2005)

[13] Rippel, O., Mertens, P., Merhof, D.: Modeling the distribution of normal data in pre-trained deep features for anomaly detection. In: 2020 25th International Conference on Pattern Recognition (ICPR). pp. 6726–6733. IEEE (2021)

[14] Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Müller, K.R.: A unifying review of deep and shallow anomaly detection. Proceedings of the IEEE (2021)

[15] Ruff, L., Vandermeulen, R., Goernitz, N., Deecke, L., Siddiqui, S.A., Binder, A., Müller, E., Kloft, M.: Deep one-class classification. In: International conference on machine learning. pp. 4393–4402. PMLR (2018)

[16] Schubert, E., Wojdanowski, R., Zimek, A., Kriegel, H.P.: On evaluation of outlier rankings and outlier scores. In: Proceedings of the 2012 SIAM International Conference on Data Mining. pp. 1047–1058. SIAM (2012)

[17] Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **61**(3), 611–622 (1999)

[18] Xie, X.: A review of recent advances in surface defect detection using texture analysis techniques. ELCVIA: electronic letters on computer vision and image analysis pp. 1–22 (2008)