Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

Semi-supervised visual anomaly detection based on convolutional autoencoder and transfer learning

Jamal Saeedi [*], Alessandro Giusti

Dalle Molle Institute for Artificial Intelligence (IDSIA USI-SUPSI), Lugano, Switzerland

ARTICLE INFO

Keywords:
Anomaly detection
Visual inspection
Convolutional autoencoder
Deep feature embedding
One-class classifier

ABSTRACT

Recent advances in deep neural networks have shown that reconstruction-based methods using autoencoders have potential for anomaly detection in visual inspection tasks. However, there are challenges when applying these methods to high-resolution images, such as the need for large network training and computation of anomaly scores. Autoencoder-based methods detect anomalies by comparing an input image to its reconstruction in pixel space, which can result in poor performance due to imperfect reconstruction. In this paper, we propose a method to address these challenges by using a conditional patch-based convolutional autoencoder and one-class deep feature classification. We train an autoencoder using only normal images and compute anomaly maps as the difference between the input and output of the autoencoder. We then embed these anomaly maps using a pretrained convolutional neural network feature extractor. Using the deep feature embeddings from the anomaly maps of training samples, we train a one-class classifier to compute an anomaly score for an unseen sample. A simple threshold-based criterion is used to determine if the unseen sample is anomalous or not. We compare our proposed algorithm to state-of-the-art methods on multiple challenging datasets, including a dataset of zipper cursors and eight datasets from the MVTec dataset collection. We find that our approach outperforms alternatives in all cases, achieving an average precision score of 94.77% for zipper cursors and 96.51% for MVTec datasets.

1. Introduction

Anomaly detection (AD) refers to the identification of items or events that deviate from an expected pattern or do not match other items in a dataset. In this study, we focus on using AD for visual inspection of various products with low-level anomaly types in high-resolution images. These anomalies can manifest in different forms, such as scratches, dents, contamination, and various structural changes. In contrast, a high-level (semantic) anomaly is one that deviates in high-level factors of variation or semantic concepts, such as a dog among a normal class of cats.

Visual inspection tasks in the manufacturing industry often present a few examples of defective samples, or it can be unclear what types of defects may appear. This makes it challenging to provide a sufficiently large dataset where each sample is labeled as either "normal" or "abnormal", which is necessary for traditional supervised classification techniques (Saeedi, Dotta, Galli, et al., 2021). To address this issue, many relevant applications must rely on semi-supervised algorithms for identifying anomalous samples. Semi-supervised techniques construct a model using only normal training samples, which represent normal behavior. Then, they test unseen samples using the learned model. The objective of the project presented in this paper is to automate the inspection process of zipper cursors in production lines using an image acquisition system (IAS) and a dedicated software based on a semi-supervised pipeline. A general block diagram for the semi-supervised pipeline is shown in Fig. 1. The first step is image registration (alignment), which is performed using a reference part, followed by an image enhancement algorithm. After training a model using normal samples, anomalous parts are selected using anomaly score calculation and a predefined threshold. In this project, it is assumed that the object for inspection has a rigid shape, and a reference image is used for image registration.

The AD in products with complex structures is challenging due to the wide range of image variations. Some changes in the image do not necessarily indicate manufacturing defects. There are various sources of image variations, such as misalignment, changes in appearance, variations within the normal specification range, and variations outside of the specification range (normal images that are not included in the training set). The main challenge in AD is to differentiate between unrelated image changes and identify true anomalies with reliable performance. In recent years, advances in deep neural networks have led to the development of reconstruction-based methods using autoencoders (AEs) that have shown great potential for AD tasks. An

[*] Corresponding author.
E-mail addresses: jamal.saeedi@idsia.ch (J. Saeedi), alessandro.giusti@idsia.ch (A. Giusti).

https://doi.org/10.1016/j.mlwa.2023.100451
Received 27 September 2022; Received in revised form 20 January 2023; Accepted 20 January 2023
Available online 24 January 2023

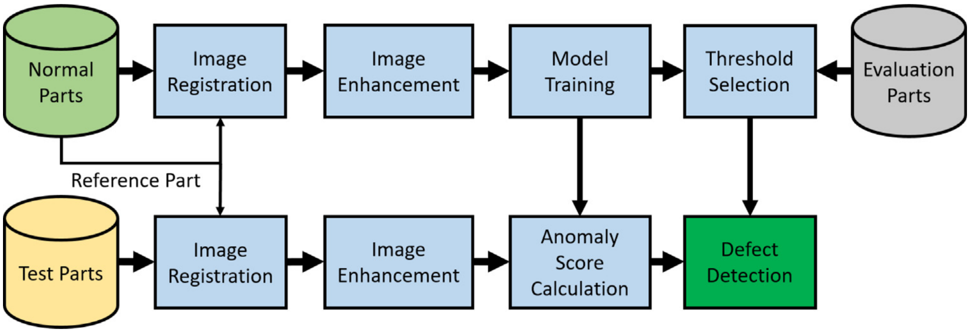


Fig. 1. Block diagram of the semi-supervised pipeline for anomaly detection. Note that the evaluation parts consist of both normal and abnormal samples, which are labeled and only used for threshold selection.

AE is a neural network that is trained to learn reconstructions that closely match its original input. These methods assume that normal and anomalous samples lead to significantly different embeddings, and thus, the corresponding reconstruction errors can be used to distinguish between normal and anomalous samples (Jinwon & Sungzoon, 2015; Kingma & Welling, 2014). By utilizing an AE-based approach for the visual inspection project, we encounter two significant challenges:

- The limitations associated with high-resolution images and low-level defects: designing an appropriate CAE with high-resolution images results in a large network size. Training such a large network is very time-consuming, and there is a risk of network overfitting due to the small number of training samples in some cases.
- The use of simple per-pixel comparison for anomaly score calculation: The AE-based method detects anomalies by comparing the input image to its reconstruction in pixel space. This can lead to poor AD performance due to imperfect reconstruction.

These challenges serve as the motivation for the framework presented in this paper, which combines the AE-based method with one-class deep feature classification. In the following sections, we provide more details of the proposed solution.

State-of-the-art deep learning methods, such as those using AE and their variations (Chao-Qing et al., 2019), typically evaluate on public datasets with small dimensions, such as MNIST (LeCun, 1998) at 28×28 pixels, Fashion-MNIST (Xiao, Rasul, & Vollgraf, 2017) at 28×28 pixels, CIFAR-10 (Krizhevsky & Hinton, 2009) at 32×32 pixels, and ImageNet (Deng et al., 2009) at 224×224 pixels. However, in visual inspection scenarios, such as the MVTec dataset ((et al., 2019a) at 1024×1024 pixels, the image dimension is much larger. Previous approaches to addressing this size issue include downsizing (Bergmann, Fauser, et al., 2019a) and patch-wise inspection (Matsubara, Hama, Tachibana, & Uehara, 2018), but these can be problematic for AD. Downsizing can lead to small defects being lost, and patch-wise inspection can miss larger defects that are larger than the patch size. In this paper, we propose a new framework using a conditional patch-based convolutional autoencoder (CPCAE) to address this size issue. Our approach uses both downsizing and patch extraction to avoid these problems. Specifically, we extract overlapping patches from downsized images to train the AE, and provide the network with the index of the patches in the image (i.e., the patches' location) as an auxiliary input. This allows each patch to remember where it came from in the image, similar to the recently developed conditional variational autoencoder (VAE) (Pol, Berger, Germain, Cerminara, & Pierini, 2019) for MNIST data AD, where class labels (from "0" to "9") were used as a condition for training the VAE. To calculate the anomaly map and score for a test image, we reverse the patch extraction and upsizing process for the AE output, and obtain an anomaly map using the difference between the input and the reconstructed images".

We propose a new approach to improve the performance of AD by incorporating transfer learning with the AE-based AD method. This is done to avoid the issue of computing anomaly scores using AE reconstruction errors, which can result in poor performance due to simple per-pixel comparison and imperfect reconstruction (Bergmann et al. 2019b; (Nalisnick, Matsukawa, Whye, Gorur, & Lakshminarayanan, 2018)). Transfer learning, which involves using discriminative embeddings from pretrained networks, has been shown to improve the performance of many supervised computer vision algorithms (Ruff et al., 2018). Recent research suggests that these feature spaces generalize well for AD tasks and that even simple baselines can outperform deep learning approaches (Kornblith, Shlens, & Le, 2019). A new trend in recent years for AD is using one-class classification with deep features extracted from a pretrained convolutional neural network (CNN) (Bergman, Cohen, & Hoshen, 2020; Oza & Patel, 2019; Perera & Patel, 2019). Our approach involves using a pretrained CNN (on the Imagenet dataset) to embed the anomaly maps obtained from a trained AE. Then, a one-class classifier, k nearest neighbor (k-NN), is trained to compute the anomaly score for unseen samples. Finally, a simple threshold-based criterion is used to determine whether the unseen sample is anomalous. In summary, we propose a hybrid framework that combines transfer learning with AE modeling to overcome issues with simple per-pixel comparisons or imperfect reconstructions in the AE-based AD method.

We extensively evaluate the proposed method on various datasets, including the zipper cursor dataset, which was specifically acquired and introduced for this study, as well as the recently introduced MVTec AD dataset, which includes different types of visual inspections (Bergmann, Lowe, Fauser, Sattlegger, & Steger, 2019b). Our results show that when combined with one-class deep feature classification using the proposed framework, our method outperforms state-of-the-art techniques.

Our main contributions include:

1. A novel approach using CPCAE for AD, addressing the challenges associated with high-resolution images in visual inspection scenarios.
2. A hybrid framework based on transfer learning, which calculates anomaly scores using AE reconstruction errors, and embeds anomaly maps computed by AE using a pretrained CNN feature extractor to train a one-class classifier.
3. Demonstrated state-of-the-art performance on multiple datasets, including the zipper cursor dataset and MVTec anomaly detection dataset.

The remainder of this paper is organized as follows: In Section 2, we will conduct a review of related work. In Section 3, we will discuss the proposed method, which is based on CPCAE and transfer learning. In Section 4, we will present the experimental results and discussion. This section will include information on the experimental setup, dataset, evaluation metrics, evaluation methods, and AD results. Finally, in Section 5, we will provide conclusions and outline future work.

2. Related work

AD methods can be broadly categorized into the following: representation-based, reconstruction-based and hybrid approaches. These categories are briefly discussed as follows:

(1) Representation-based methods extract discriminative features for normal images or normal image patches with a deep CNN, and establish a one-class classifier using one of the following approaches:

(a) Probabilistic approaches, such as Gaussian mixture models (Eskin, 2000) and kernel density estimation (Xu, Caramanis, & Sanghavi, 2012) assume that the normal data follows a statistical model. During the training, a distribution function is fitted on the features extracted from the normal samples. Then, during the test, the samples that are mapped to different statistical representations are considered anomalous.

(b) Proximity-based algorithms assume that the proximity of an anomalous object to its nearest neighbors massively deviates from its proximity to most of the other objects in the dataset. Given a set of objects in a feature space, a distance measure can be used to compute the similarity between objects. Then, objects that are far from others can be regarded as anomalies. These methods depend on the well-defined similarity measure between two data points. The basic proximity-based methods include the local outlier factor (Breunig, Kriegel, Ng, & Sander, 2000) and its variants (Tang & He, 2017).

(c) Boundary-based approaches, mainly involving one class of support vector machines (Scholkopf, Platt, Shawe-Taylor, Smola, & Williamson, 2001) and support vector data descriptions (Tax & Duin, 2004), usually try to define a boundary around the normal samples. Anomaly samples are determined by their location with respect to the boundary. A recent trend in boundary-based AD methods is to utilize transfer learning techniques using a pretrained CNN network to extract discriminative embedding vectors for classification. Andrews, Tanay, Morton, and Griffin (2016) use activations from different layers of a pretrained VGG network and model the anomaly-free training distribution with one class of support vector machines. Similar experiments have been performed by Burlina, Joshi, and Wang (2019), as they reported a superior performance of discriminative embeddings compared to feature spaces obtained from generative models. Nazaré et al. (2018) investigated the performance of different off-the-shelf feature extractors pretrained on an image classification task for anomaly segmentation in surveillance videos. Their approach trains a 1-nearest-neighbor classifier on embedding vectors extracted from a large number of anomaly-free training patches. Similarly, Napoletano, Piccoli, and Schettini (2018) extracted activations from a pretrained ResNet-18 for a large number of cropped training patches and modeled their distribution using k-means clustering after a prior dimensionality reduction with a principal component analysis.

(2) Reconstruction-based approaches assume that anomalies cannot be compressed, and therefore, cannot be efficiently reconstructed from their low-dimensional embeddings. In this category, principal component analysis (Olive, 2017) and its variations (Baklouti, Mansouri, Nounou, Nounou, & Hamida, 2016; Harrou, Kadri, Chaabane, Tahon, & Sun, 2015) are widely used. In addition, AE- and VAE-based methods also belong to this category (Jinwon & Sungzoon, 2015; Kingma & Welling, 2014).

(3) Hybrid approaches utilize both reconstruction- and representation-based methods in a hybrid framework. Specifically, these methods use AE to generate feature embedding for training a one-class classifier, in which the latent space variables act as the embedding. Kawachi, Koizumi, and Harada (2018) proposed an assumption that the anomaly prior distribution is a complementary set of the prior distribution of normal samples in latent space. Based on this assumption, the anomalous and normal data have complementary distributions, which means that they can be separated in the latent space; then, it is possible to apply a one-class classifier to detect anomalies. Similarly, Guo, Liu, Zuo, and Wu (2018) used the compressed hidden layer vector of a trained AE on normal data, which is regarded as the deep feature representation of the original data, to train a k-NN-based AD method. Amarbayasgalan, Jargalsaikhan, and Ryu (2018) proposed an approach to detect novelty by combining deep AE for low-dimensional representation and clustering techniques for novelty estimation.

In this paper, we aim to propose a better discriminative embedding compared to the latent space variables of AEs for one-class classification. The proposed method presented in this paper can be considered a hybrid approach, as we utilize both reconstruction- and representation-based approaches, which will be fully discussed in the next section.

3. Proposed anomaly detection based on autoencoder and transfer learning

This section outlines the key principles of the CPCAE method, which is illustrated in Fig. 2. We work within a semi-supervised framework, where instances of anomalous data are not available. As a result, we train a model using only normal samples that are initially provided. The proposed method is divided into two parts: the generation of an anomaly map using a CAE and the calculation of an anomaly score using deep feature one-class classification, as shown in Fig. 2. By utilizing this hybrid approach, we combine the use of AE and transfer learning with one-class classification to enhance AD results and then evaluate each method individually. In the subsequent subsections, we will discuss image pre-processing, the CPCAE method, deep feature one-class classification, the selection of an adaptive decision threshold, and anomaly detection.

3.1. Pre-processing

The image pre-processing step is applied to eliminate irrelevant image-to-image changes, such as spatial misalignment and appearance changes. Here, we apply image registration and contrast enhancement methods to improve AD results. The image registration algorithm applied here is based on parametric image alignment using correlation coefficient maximization, which finds a rigid (Euclidean) motion between two sets of images (Evangelidis & Psarakis, 2008). This approach is shortly described as follows: The alignment problem can be considered here as a mapping between the coordinate systems of two images. Once the geometric parametric transformation has been defined in first step, the alignment problem reduces itself into a parameter estimation problem. Therefore, the next step is to select an appropriate performance measure, that is, an objective function. An enhanced correlation coefficient (Psarakis & Evangelidis, 2005) is adopted here, as the objective function for the alignment problem. Finally, an iterative scheme of nonlinear optimization is applied for the optimum parameter estimation. The algorithm is available within the OpenCV library (Bradski, 2000) and captures various geometric transforms (translation, rigid, similarity, and affine). The transform parameters and optimization step sizes can be initialized as input parameters, and a region of interest can also be specified to limit the spatial region used for registration.

Image contrast enhancement is applied to improve poor image contrast in zipper cursor images, which mainly occurs because of the images' shiny surfaces and curve geometry. Considering that we have a range of objects from shiny to matte, we need an adaptive contrast enhancement approach. To achieve this, we propose an adaptive gamma correction (AGC) method that dynamically determines an intensity transformation function according to the characteristics of the input image. The first step of the AGC method is applying the intensity-hue-saturation (IHS) transform on the image. The intensity image is used to segment the object region from the background using a Canny edge detector, as well as morphological dilation and morphological image filling, as shown in Fig. 3. Because there is only one dominant region in the image, the region with the maximum area is selected as a valid

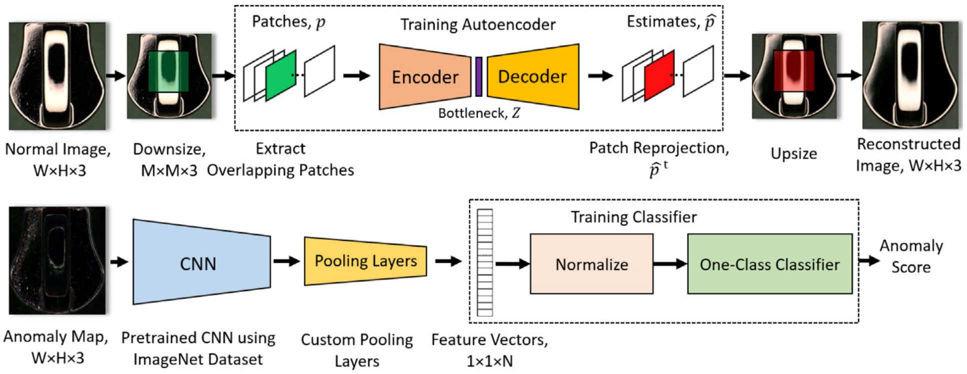


Fig. 2. Block diagram of the proposed anomaly detection method (dashed lines show the steps involved in the training step).

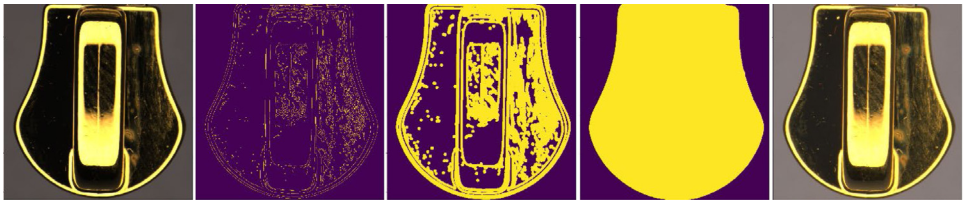


Fig. 3. Adaptive gamma correction, from left to right: the input image, the image with Canny edge detection, the image dilation, the binary image filling, and the output image after gamma correction.

region. Next, the average brightness value inside the segmented area is found and denoted by I_μ. Adaptive gamma is then obtained using a predefined threshold, C, represented as normal brightness using the following:

$$g = \log_{10} C / \log_{10} I_\mu \tag{1}$$

AGC is performed on the intensity image using the image to a power of g, and finally, the inverse IHS transform generates a contrast-enhanced image. Fig. 4 shows a few examples of the AGC applied on zipper cursor images ($C = 0.4$), which improves the image's quality to discriminate defects, including halos and scratches.

3.2. Conditional patch-based convolutional autoencoder (CPCAE)

Autoencoders attempt to reconstruct an input image $\mathrm{x} \in \mathbb{R}^{\mathrm{C\times H\times W}}$ through a bottleneck, mapping the input image into a lower-dimensional space called the latent space (Chao-Qing et al., 2019). An AE consists of an encoder, $E : \mathbb{R}^{\mathrm{C\times H\times W}} \rightarrow \mathbb{R}^{\mathrm{d}}$, and a decoder, $D : \mathbb{R}^{\mathrm{d}} \rightarrow \mathbb{R}^{\mathrm{C\times H\times W}}$ where d indicates the latent space's dimensionality and C, H, *and* W represent the number of channels, height, and width of the input image, respectively. The overall process can be written as follows:

$$\hat{x} = D(E(x)) = D(z) \tag{2}$$

where z represents the latent vector and $\hat{x}$ *denotes* the reconstruction of the input. The functions E and D are parameterized by CNNs.

For simplicity and computational speed, a per-pixel error measure such as the L_2 loss is chosen to force the AE to reconstruct its input:

$$L_2(\mathrm{x}, \hat{x}) = \sum_{c=0}^{C} \sum_{h=0}^{H} \sum_{w=0}^{W} (\mathrm{x}(c,h,w) - \hat{x}(c,h,w))^2 \tag{3}$$

where $\mathrm{x}(c,h,w)$ denotes the intensity value of image x at pixel (c,h,w). During evaluation, the per-pixel ^{2}-distance of x and $\hat{x}$ is computed to obtain a residual map $\mathrm{R}(\mathrm{x},\hat{x}) \in \mathbb{R}^{\mathrm{C\times H\times W}}$.

For the AD task, the AE is only trained on defect-free samples. AEs are data-specific, meaning they are only able to efficiently process data similar to what they have been trained on. Therefore, in semi-supervised AD, the AE trained only on normal samples will generate higher reconstruction errors for anomalies in the test set compared to normal samples. This is because anomalous samples result in different encodings compared to normal samples, and the AE is unable to reconstruct defects that it has not seen during training. The reconstruction error, L_2, of each test data point is used as the anomaly score, and data with a high anomaly score is defined as an anomaly.

There are two main challenges when deploying AE for the AD task in a visual inspection scenario: handling high-resolution images and achieving good performance by avoiding simple per-pixel comparisons and imperfect reconstructions (Bergmann, Lowe, et al., 2019b; Nalisnick et al., 2018). In this paper, we address the high-resolution image issue by applying overlapping patches and conditional learning for AE, which is discussed in this subsection. Additionally, we propose a new approach that incorporates transfer learning with AE to avoid computing anomaly scores using simple per-pixel comparisons, which is discussed in the next subsection. To resolve the high-resolution image problem for AE modeling, we use downsizing and patch extraction. We assume that by downsizing the input image to a certain extent, its normality (i.e., the image details that represent the normal class) is preserved. After downsizing the input image, overlapping patches are extracted to train the AE. The overlapping patches are extracted using a sliding window with square size and stride (the amount to skip in a particular direction) that are balanced so that the overlap is equal to half of the patch size. By default, we select 256 and 128 for patch and overlap sizes, respectively. However, these parameters can be fine-tuned for different datasets.

In addition to the patches, the network also receives information about the location of the patches within the image as a conditional variable. This allows the network to take into account both local information (i.e., the patches themselves) and global information (i.e., the location of the patches) when training. The use of conditional variables helps the network to train more effectively and reduces the likelihood of reproducing small defects when given a test image with defects. To calculate an anomaly map, the procedure is to apply patch reprojection and upsizing to the output of the AE, and then calculate the difference between the input and reconstructed images to obtain the anomaly map.

The most common architecture utilized for AE in AD is the convolutional layers followed by the pooling layers, the fully connected layers on the encoder side, the fully connected layers followed by the

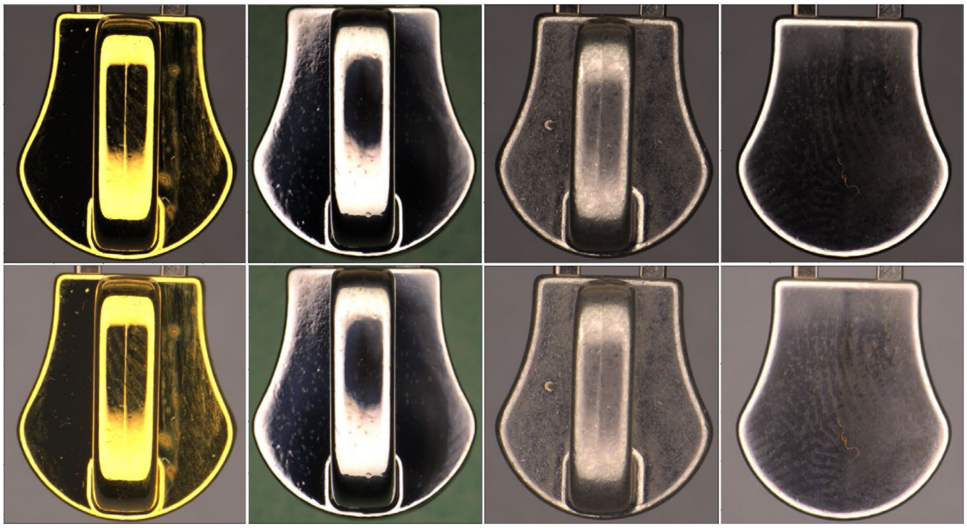


Fig. 4. Adaptive gamma correction applied to the zipper cursor dataset, top row: before and bottom row: after correction.

convolutional layers and the upsampling on the decoder side (Ribeiro, Lazzaretti, & Lopes, 2018). There are different variations of unpooling rather than upsampling in the literature (Huo, Liu, Zheand, & Yin, 2017). AE itself can be fed into another AE to form a stacked AE (Masci, Meier, Cireşan, & Schmidhuber, 2011). It is not recommended to use convolutional layers without dense layers for the AD task because this type of network is able to memorize the input's spatial information and can somehow reconstruct the defects in the given test image. AE only deploys convolutional layers because this fits better for other applications, such as image segmentation and compression, where detailed spatial information is very important for encoding (Badrinarayanan, Kendall, & Cipolla, 2017; Yildirim, Tan, & Acharya, 2018).

The proposed architecture for CPCAE is shown in Fig. 5. We use convolutional layers with strides on the encoder side instead of pooling layers, and convolutional transpose layers with strides on the decoder side. The convolutional (transpose) layers with strides allow the network to learn spatial subsampling (upsampling) from the data, resulting in higher transformability. In addition, we incorporate the new conditional variable (the index of the patches) into the encoder part of AE using concatenation. Similarly, the decoder is concatenated with the conditional vector.

The proposed CPCAE generates anomaly maps to be used for training a one-class classifier in the second step of the proposed hybrid method. However, there is a challenge in training the one-class classifier using the same training set that was already applied for AE modeling. One solution to this problem is to split the training set into two parts, one for AE modeling and the other for classifier training. However, this approach may not be feasible if the training set is small, as it could decrease the AE's ability to learn normal behavior. Another solution is to train the AE using sparse information from the training set to avoid network overfitting and to preserve the useful information in the anomaly maps for classifier training. To achieve this, we propose adding regularization to the AE's loss function to obtain a sparse AE.

For a sparse autoencoder, in most cases, the loss function is constructed by penalizing activations of hidden layers, meaning that only a few nodes can be activated when a single sample is fed into the network. L_1 and L_2 regularizations are widely used in deep learning, and the main difference between them is that L_1 regularization tends to reduce the penalty coefficients to zero, while L_2 regularization would move coefficients near zero. More details can be found in Chang, Du, and Zhang (2019). The loss function using L_1 regularization is expressed as follows:

$$Loss = L_2(\mathrm{x}, \hat{x}) + \lambda \sum_i \left| a_i^{(h)} \right| \tag{4}$$

The second term penalizes the absolute value of the vector of activations a in layer h for sample i. A hyperparameter λ is also used to control its effect on the whole loss function.

3.3. Deep feature-based one class classification

In this subsection, the second step of the proposed method, which involves feature extraction using a pretrained CNN model followed by a one-class classifier, is explained. Specifically, the anomaly maps generated using AE modeling in the first step are used to train a one-class classifier. Using binary classification in addition to the AE modeling result, we would like to leverage transfer learning through feature extraction via a pretrained CNN network and to avoid computing the anomaly score using simple AE per-pixel comparisons.

Transfer learning improves many performances of the supervised computer vision algorithms (Burlina et al., 2019; Kornblith et al., 2019) using discriminative embeddings from the pretrained networks. This is also true for semi-supervised AD tasks, as recent works suggest that combined with these feature spaces, one-class classifiers outperform AE-based approaches (Nazaré et al., 2018).

The second step of the proposed AD method takes a set of anomaly maps generated by AE modeling, $X_{train} = x_1, x_2 \ldots x_N$. It uses a pretrained feature extractor F to extract features from the entire training set:

$$f_i = F(x_i) \tag{5}$$

The training set is now summarized by a set of embeddings $F_{train} = f_1, f_2 \ldots f_N$. In this paper, we use a deep feature extractor that was pretrained on the ImageNet dataset. Several CNN models serve as solid options for extracting features, including AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), VGGNet (Simonyan & Zisserman, 2014), GoogLeNet (Szegedy, Liu, Jia, et al., 2014), ResNet (He, Zhang, Ren, & Sun, 2016), and Xception (Chollet, 2017). The deep network and its depth are data-related and should be selected experimentally. In this study, we chose the Xception network just before the global pooling layer. Xception can be considered an extreme inception architecture (Szegedy, Vanhoucke, Ioffe, et al., 2016), which introduces the idea of depth-wise separable convolution. More mathematical details can be found in (Wang et al., 2019). However, we compare different pretrained networks for feature extraction in the results section.

The global max pooling layer is usually used in addition to the last convolutional layer of the pretrained networks to generate feature embedding (Nazaré et al., 2018). Here, we apply a new pooling layer to generate the final image embedding, as shown in Fig. 6. Since we

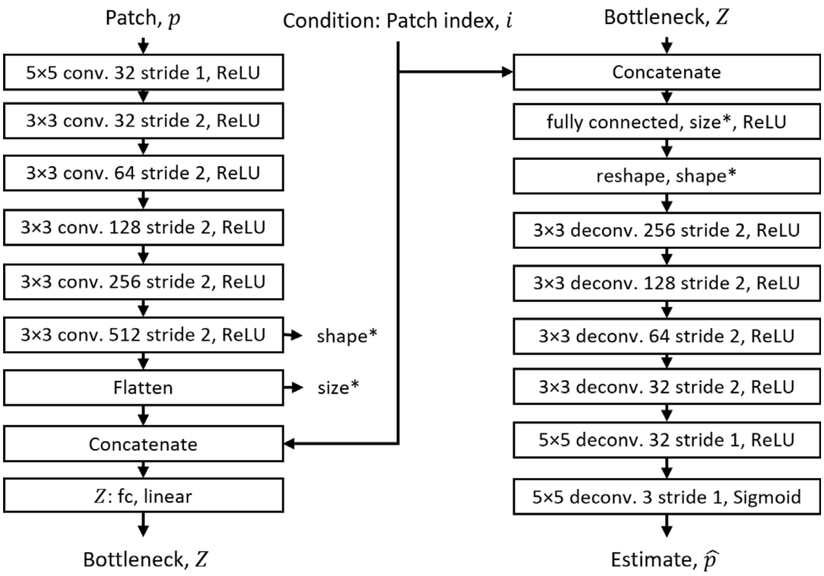

Patch, p
Condition: Patch index, i
Bottleneck, Z
5×5 conv. 32 stride 1, ReLU
3×3 conv. 32 stride 2, ReLU
3×3 conv. 64 stride 2, ReLU
3×3 conv. 128 stride 2, ReLU
3×3 conv. 256 stride 2, ReLU
3×3 conv. 512 stride 2, ReLU
shape*
Flatten
size*
Concatenate
Z: fc, linear
Bottleneck, Z
Concatenate
fully connected, size*, ReLU
reshape, shape*
3×3 deconv. 256 stride 2, ReLU
3×3 deconv. 128 stride 2, ReLU
3×3 deconv. 64 stride 2, ReLU
3×3 deconv. 32 stride 2, ReLU
5×5 deconv. 32 stride 1, ReLU
5×5 deconv. 3 stride 1, Sigmoid
Estimate, $\hat{p}$

Fig. 5. The structure of the proposed CPCAE for anomaly map generation. In the middle, there is a fully connected autoencoder. The rest are convolutional layers and convolutional transpose (deconv) layers.

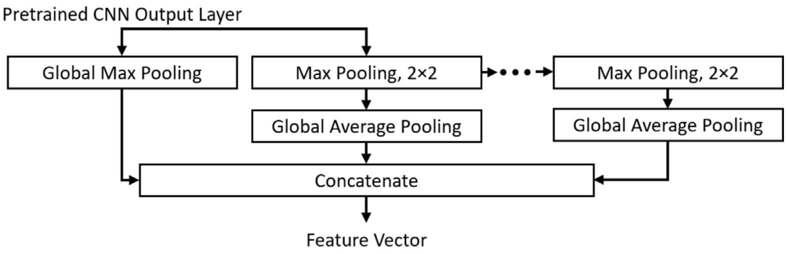


Fig. 6. Proposed pooling layer used on top of the pretrained CNN network for feature extraction.

feed the input image without downsizing into the pretrained network, the number of features after a global max pooling layer is too small to represent a high-resolution image in a visual inspection scenario. Using the new pooling layer, which consists of parallel and cascade pooling, along with concatenation, our generated final embedding has more features compared to the traditional final embedding.

After acquiring the image embedding following normalization (mean removal and variance scaling), a suitable one-class classifier, such as one-class support vector machines (OC-SVM) (Scholkopf et al., 2001), support vector data descriptions (Tax & Duin, 2004). Alternatively, k-NN (Bergman et al., 2020) can be trained on the embeddings. For instance, k-NN is explained here, as it is widely applied to AD tasks (Nazaré et al. 2018; (Guo et al., 2018)). To detect if a new sample y is anomalous, we first extract its feature embedding using (7) and normalize it. We then compute its k-NN distance and use it as the anomaly score as follows:

$$d(y) = \frac{1}{k} \sum_{f \in N_k(f_y)} \left\| f - f_y \right\|^2 \tag{6}$$

$N_k\left(f_y\right)$ denotes the k nearest embeddings to f_y in the training set F_{train}. Euclidean distance is used here and often achieves superior results on features extracted by deep networks (Bergman et al., 2020), but other distance measures can be used in a similar way. We determine if an image y is normal or anomalous by confirming whether the distance $d(y)$ is larger than a threshold. It should be mentioned that we compare different types of one-class classifiers in the experimental results section.

3.4. Adaptive decision threshold selection

Having the normal samples' anomaly scores inside training set, it is not clear what would be the values of anomaly scores for the test set. Some studies adjust the threshold by cross-validation (Jinwon & Sungzoon, 2015). However, building a sufficiently large validation set is not possible in some cases, as anomalous samples are difficult to collect. This is also challenging for the current project because cross-validation is not feasible for every type of zipper cursor when there are many different types. Therefore, we need an adaptive strategy for the threshold selection task.

In this paper, we utilize a modified approach proposed in Wang et al. (2019) for adaptive threshold selection using a probability density function (PDF). The idea is to use the kernel density estimation (KDE) method to estimate the PDF using anomaly scores obtained from the training set (Gramacki & Gramacki, 2017). After computing the PDF, the decision threshold can be obtained using the cumulative distribution function:

$$T(s) = \alpha \int_{-\infty}^{s} p(s)\, ds + \beta \tag{7}$$

where α is the significance level, β *is an offset*, s is the anomaly score, and $p(s)$ is the PDF estimated by KDE method using training dataset's anomaly scores.

To obtain the parameters of (7), we run an experiment to find some thresholds using the test sets of different dataset where false-positive and true positive rates have optimal balance. With the optimal thresholds for different datasets, T(s*), we can obtain the optimal

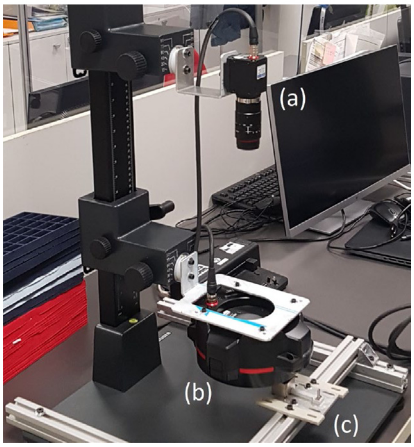


Fig. 7. Image acquisition setup, (a) CV-X series vision system from KEYENCE, (b) Lighting system, and (c) Holder and fixture.

Table 1
Statistical overview of the zipper cursor dataset.

Set	# Train	# Test (normal)	# Test (defective)
# 1	60	55	148
# 2	60	47	84
# 3	49	33	14
# 4	40	39	36
# 5	44	19	31
# 6	28	23	18

parameters of (7), α *and* β, by solving the linear equations. A higher significance level α and β leads to a lower missing alarm rate, which means that models have fewer chances to mislabel outliers as normal data. On the contrary, a lower α and β means a lower false alarm rate. Therefore, the choice of parameters is a trade-off.

3.5. Detecting anomalies

In this subsection, we summarize the steps for training the AE model using normal samples, generating anomaly maps to train the deep one-class classifier, and using the trained AE and deep one-class classifier to detect anomalies for the test set in Algorithms 1 and 2.

4. Experimental results and discussion

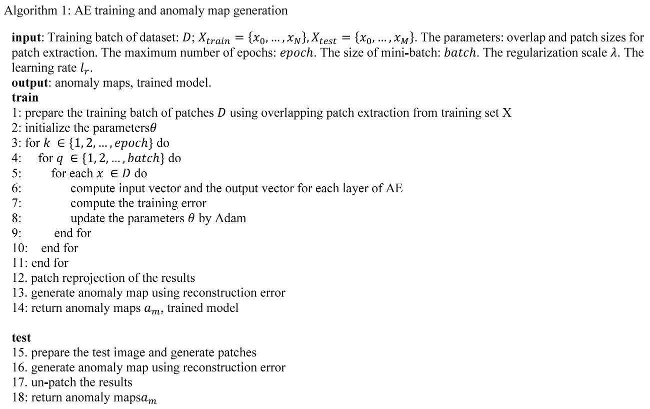
In this section, the results of the proposed method for the AD task are presented. In addition, collecting data for zipper cursors and evaluating the proposed framework, as well as several state-of-the-art approaches are discussed. In the following subsections, we also discuss the experimental setup, the dataset, the evaluation metrics, the evaluated methods and the AD results.

4.1. Experimental setup

The IAS used here is a CV-X series vision system from KEYENCE, which is a multimode IAS. The models for the camera, lens and lighting system are as follows: CA-H200MX, CA-LHR50, and CA-DRM10X (Keyence, 2022). We use a 2-megapixel camera that generates images 1600 × 1200 in size. In the current IAS system setup, we use a diffused ring light system near the object where the object is illuminated from a low angle by the uniform diffuse light through the light conduction plate. Diffuser attachment is used to lessen the problem with shiny surfaces and to simulate dome lighting. The IAS and the camera and lighting stand with the fixture and holder is shown in Fig. 7.

4.2. Datasets

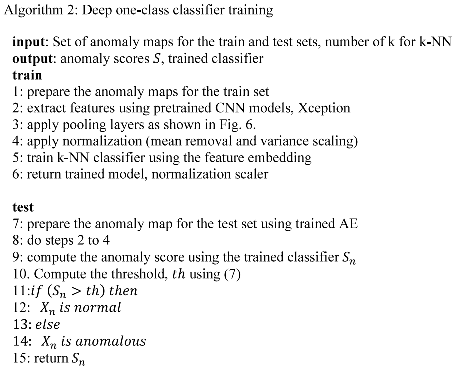
The proposed method mainly focuses on addressing the challenges related to the zipper cursor dataset. However, the method can be generalized to work with other AD datasets, specifically those for visual inspection with high-resolution images. The zipper cursor dataset includes six different types and is summarized in Table 1. The anomalies manifest themselves in the form of bubbles, residue, halos and scratches. In addition to the zipper cursor dataset, we evaluate the proposed method on the MVTec dataset, which involves different types of visual inspection (Bergmann, Fauser, et al., 2019a). The MVTec dataset is composed of 15 categories. However, we only consider 8 of the 15 categories of the MVTec dataset that have rigid shapes to be registered. Table 2 gives an overview of each object's category. The anomalies consisted of different types of defects, including scratches, dents, contaminations, and various structural changes. For all of the datasets, pixel values of all images are normalized to [0, 1], and the images are cropped to maximize the field of view. Fig. 8 shows different sets of zipper cursors and different categories of MVTec datasets used for the analysis.



```
Algorithm 1: AE training and anomaly map generation

input: Training batch of dataset: $D$; $X_{train} = \{x_0, \dots, x_N\}, X_{test} = \{x_0, \dots, x_M\}$. The parameters: overlap and patch sizes for patch extraction. The maximum number of epochs: $epoch$. The size of mini-batch: $batch$. The regularization scale $\lambda$. The learning rate $l_r$.
output: anomaly maps, trained model.
train
1: prepare the training batch of patches $D$ using overlapping patch extraction from training set X
2: initialize the parameters$\theta$
3: for $k$ $\in \{1, 2, \dots, epoch\}$ do
4:    for $q$ $\in \{1, 2, \dots, batch\}$ do
5:       for each $x$ $\in D$ do
6:           compute input vector and the output vector for each layer of AE
7:           compute the training error
8:           update the parameters $\theta$ by Adam
9:       end for
10:    end for
11: end for
12. patch reprojection of the results
13. generate anomaly map using reconstruction error
14: return anomaly maps $a_m$, trained model

test
15. prepare the test image and generate patches
16. generate anomaly map using reconstruction error
17. un-patch the results
18: return anomaly maps$a_m$
```



```
Algorithm 2: Deep one-class classifier training

input: Set of anomaly maps for the train and test sets, number of k for k-NN
output: anomaly scores $S$, trained classifier
train
1: prepare the anomaly maps for the train set
2: extract features using pretrained CNN models, Xception
3: apply pooling layers as shown in Fig. 6.
4: apply normalization (mean removal and variance scaling)
5: train k-NN classifier using the feature embedding
6: return trained model, normalization scaler

test
7: prepare the anomaly map for the test set using trained AE
8: do steps 2 to 4
9: compute the anomaly score using the trained classifier $S_n$
10. Compute the threshold, $th$ using (7)
11:$if\ (S_n > th)\ then$
12:   $X_n\ is\ normal$
13: $else$
14:   $X_n\ is\ anomalous$
15: return $S_n$
```

4.3. Evaluation metrics

The receiver operator characteristic (ROC) and precision–recall (PR) curves are common metrics for the AD task that are defined over all possible decision thresholds. It is also useful to quantitatively evaluate the model's performance using a single value rather than comparing curves. The area under the ROC curve (AUC) and average precision (AP) are the common metrics that are obtained using ROC and PR curves, respectively. AP summarizes a PR curve with a sum of that precisions at each threshold, and then multiplies them by the increase in recall, which is an approximation of the area under the PR curve. Since the AD task always has a large skew in the class distribution, AP gives a more accurate assessment of an algorithm's performance (Davis & Goadrich, 2006). In our experiments, AUC and AP were used to evaluate the model's performance.

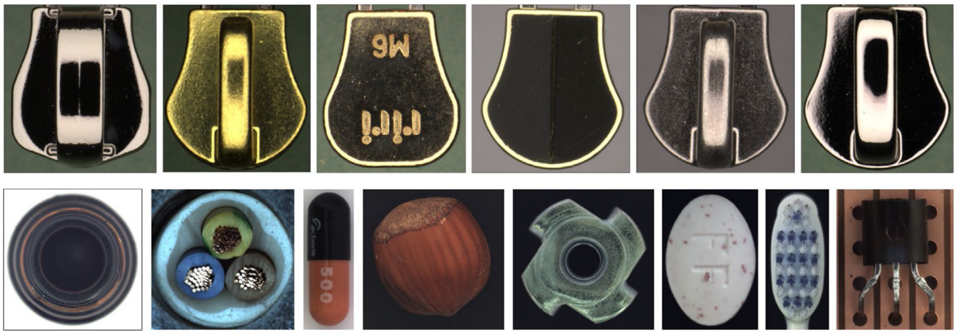


Fig. 8. Anomaly detection dataset, first row, left to right: zipper cursor dataset sets. #1 to #6. bottom row, left to right: "Bottle", "Cable", "Capsule", "Hazelnut", "Metal Nut", "Pill", "Toothbrush", and "Transistor".

Table 2
Statistical overview of the MVTec AD dataset.

Set	# Train	# Test (normal)	# Test (defective)
Bottle	209	20	63
Cable	224	58	92
Capsule	219	23	109
Hazelnut	391	40	70
Metal Nut	220	22	93
Pill	267	26	141
Toothbrush	60	12	30
Transistor	213	60	40

4.4. Evaluated methods

We compare the proposed AD method with four different approaches that are implemented here: AE (Bergmann, Fauser, et al., 2019a), deep feature one class classifier (Perera & Patel, 2019), variation (Steger, Ulrich, & Wiedemann, 2018) and nearest neighbor (NN) (Vaikundam, Hung, & Chia, 2016). To evaluate the AE method, we use the same CAE architecture described in the paper for the proposed method. For a deep feature classifier, we use the implementation proposed in Perera and Patel (2019), which applies a pretrained CNN network to the image and extracts features using global max pooling. After normalization, k-NN is used to generate anomaly scores ($k = 15$ is used for classifier). The variation is a baseline method that is based on statistics, mean and standard deviation, which is computed by the normal training set. Anomaly maps are then obtained by computing the distance of each test pixel's gray value to the computed pixel mean relative to the computed standard deviation. The anomaly score is obtained using the sum of the squares of pixels in the anomaly map. NN is another baseline method where the anomaly score is obtained by computing the distance (usually l_2) between the test sample and its most similar image inside the normal training set. It should be mentioned that parameter tuning is performed for different models included in the comparison to find the best solution. In addition, for AE and deep feature approaches, we use a similar architecture in the proposed method for a fair comparison.

Apart from the methods that have been implemented for the comparison in this paper, we also report AUC results for the MVTec dataset from recently published deep-learning-based methods, consisting of VAE (Jinwon & Sungzoon, 2015), AnoGAN (Schlegl, Seebock, Waldstein, Erfurth, & Langs, 2017), GeoTrans (Golan & El-Yaniv, 2018), GANomaly (Akcay, Atapour-Abarghouei, & Breckon, 2018), AE (SSIM) (Bergmann, Lowe, et al., 2019b), VAE-grad (Dehaene, Frigo, Combrexelle, & Eline, 2020), PatchSVDD (Yi & Yoon, 2020), UniStud (Bergmann, Fauser, Sattlegger, & Steger, 2020), DLA (Yoa, Lee, Kim, & Kim, 2021), and ViV-Ano (Choi & Jongpil, 2022).

4.5. Anomaly detection results

In the first experiment, we demonstrate the results of using various pretrained CNN architectures for feature extraction on the ImageNet dataset using the proposed AD method. The networks compared are VGG (Simonyan & Zisserman, 2014), Inception (Szegedy et al., 2016), MobileNet (Andrew et al., 2017), ResNet (He et al., 2016), DenseNet (Huang, Liu, Van Der Maaten, & Weinberger, 2017), and Xception (Chollet, 2017). As shown in Table 3, VGG has the highest number of parameters, Inception has the lowest error rate for ImageNet dataset classification, and DenseNet has the greatest depth. The results of the proposed AD method for the zipper cursor and MVTec datasets using different pretrained CNN architectures are presented in Tables 4 and 5, respectively. It can be observed from the results that, on average, the Xception network performs the best. However, for each dataset, the best performing network for feature extraction can be selected

The second experiment includes the two-dimensional t-SNE visualizations of the extracted features for the proposed method compared to the AE's latent space variables for normal and abnormal images in the test set (Van der Maaten & Hinton, 2008). AE's latent space variables have been used as image embedding for AD tasks in recent years (Amarbayasgalan et al., 2018; Guo et al., 2018; Kawachi et al., 2018). However, we propose a better discriminative embedding using anomaly maps in this paper. The t-SNE visualizations are shown in Figs. 9 and 10 for the zipper cursor and MVTec datasets, respectively. The features extracted by the proposed method facilitate a better distinction between normal and abnormal images compared to the AE's latent space variables.

The third experiment presents the AD results of the proposed method using different one-class classification approaches in the last step of our pipeline. The Xception network is used as a pretrained network for feature extraction. The one-class classifiers considered for comparison consist of k-NN (Bergman et al., 2020), OC-SVM (Scholkopf et al., 2001), isolation forest (IF) (Liu, Kai, & Zhou, 2009), Gaussian mixture models (GMM) (Kemmler, Rodner, Wacker, & Denzler, 2013), local outlier factor (LOF) (Breunig et al., 2000), and kernel density estimation (KDE) (Denkena, Dittrich, Noske, et al., 2020). K-NN adopts a formulation by utilizing the distance of a data point from its kth nearest neighbor and selects the top n points in the ranking list as the outliers. OC-SVM learns a close contour of the known normal data points by a kernel function. If observations lie outside the class contour, they would be regarded as anomalies. IF determines outliers by randomly splitting the high-dimensional data feature space with hyperplanes. GMM estimates a parametric generative model based on the training data using Gaussian distributions to represent the underlying distribution of the data. LOF measures the local variation density of a given data point to the local densities of its neighbors. If the local density of a data point is substantially lower than its neighbors', then it would be considered an anomaly. Finally, KDE estimates the density function directly from the data without an assumption about

Table 3
Comparison of the different pretrained CNN architectures used for feature extraction.

CNN Architectures	Year	Main contribution	Parameters	Error Rate ImageNet	Depth
VGG-16	2014	- Designing deeper networks (roughly twice as deep as AlexNet), which is done by stacking uniform convolutions. - Homogeneous topology - Uses small size kernels	138 M	7.3%	19
Inception	2015	- Handles the problem with a representational bottleneck - Replaces large size filters with small filters - Building networks using modules or blocks, instead of stacking convolutional layers	23.6 M	3.5%	159
MobileNet	2017	- Depth-wise separable convolutions - Introducing efficient model - Applying a width multiplier and a resolution multiplier by trading off a reasonable amount of accuracy to reduce size and latency	13 M	10.5%	28
ResNet-50	2016	- Residual learning - Identity mapping-based skip connections - Designing even deeper CNNs without compromising the model's generalization power - Among the first to use batch normalization.	25.6 M	3.6%	152
DenseNet	2017	- Cross-layer information flow - Using the residual mechanism to its maximum by making every layer densely connected to its subsequent layers. - Model's compactness makes the learned features nonredundant, as they are all shared through a collective knowledge.	25.6 M	6.12%	190
Xception	2017	- Depth-wise convolution followed by point-wise convolution - Introduced CNN based entirely on depth-wise separable convolution layers	22.8 M	5.5%	126

Table 4
Anomaly detection results for the zipper cursor dataset using different pretrained CNN architectures.

CNN Architectures	Metrics	Set #1	Set #2	Set #3	Set #4	Set #5	Set #6	Mean
VGG-16	AUC	94.67	97.74	93.90	88.55	92.36	97.59	94.13
	AP	97.82	97.08	89.17	89.68	93.31	97.61	94.11
Inception	AUC	94.82	96.48	92.57	88.36	94.66	96.60	93.91
	AP	97.73	98.04	88.75	89.35	96.09	96.76	94.45
MobileNet	AUC	95.58	97.08	93.25	87.58	90.94	93.52	92.99
	AP	98.14	98.46	89.44	88.92	93.09	94.61	93.77
ResNet-50	AUC	94.78	97.23	92.57	87.26	86.62	87.99	91.07
	AP	97.72	98.42	88.75	88.74	85.30	86.17	90.85
DenseNet	AUC	95.06	97.01	94.95	87.67	93.11	95.65	93.90
	AP	97.82	98.34	91.23	88.82	94.84	96.12	94.52
Xception	AUC	95.67	96.90	95.20	88.08	94.64	96.10	**94.43**
	AP	98.35	98.16	90.90	88.81	96.00	96.43	**94.77**

Table 5
Anomaly detection results for the MVTec dataset using different pretrained CNN architectures.

CNN Architectures	Metrics	Bottle	Cable	Capsule	Hazelnut	Metal Nut	Pill	Toothbrush	Transistor	Mean
VGG-16	AUC	98.20	83.75	75.63	99.82	68.09	79.99	99.14	88.41	84.24
	AP	99.47	89.97	92.41	99.90	90.57	95.57	99.67	85.49	94.64
Inception	AUC	97.63	86.18	78.50	96.06	74.55	88.17	100.0	92.29	86.84
	AP	99.32	91.68	94.72	97.75	92.74	97.57	100.0	91.66	95.63
MobileNet	AUC	99.22	78.46	86.32	95.99	75.12	82.25	98.60	91.17	86.22
	AP	99.77	85.39	96.72	97.38	92.54	96.04	99.46	89.81	94.64
ResNet-50	AUC	95.64	75.09	78.75	91.96	58.28	87.13	86.48	89.84	81.14
	AP	98.77	80.36	94.35	95.41	87.13	97.50	94.05	90.71	92.25
DenseNet	AUC	98.52	87.79	83.25	98.32	75.79	88.67	100.0	92.54	88.72
	AP	99.55	92.30	95.67	99.03	93.10	97.78	100.0	91.90	96.23
Xception	AUC	99.71	87.73	86.39	95.87	79.28	86.74	100.0	93.06	**91.09**
	AP	99.90	92.54	96.71	97.90	94.88	97.23	100.0	91.68	**96.35**

the underlying distribution. Samples in low-density areas are selected as anomalies.

Tables 6 and 7 show the results for zipper cursor and MVTec datasets using different one-class classifiers in the proposed AD framework. From the results, it can be seen that k-NN and LOF classifiers achieve high-quality performances for the AD task compared to other alternative classifiers. The advantage of k-NN-based approaches is that they do not need an assumption for the data distribution and can be applied to different data types. To gain a better understanding of the reasons for this high-quality performance, we consider the t-SNE plots from the second experiment. In Fig. 10, we can observe that the t-SNE plots from the test set feature embeddings extracted from the

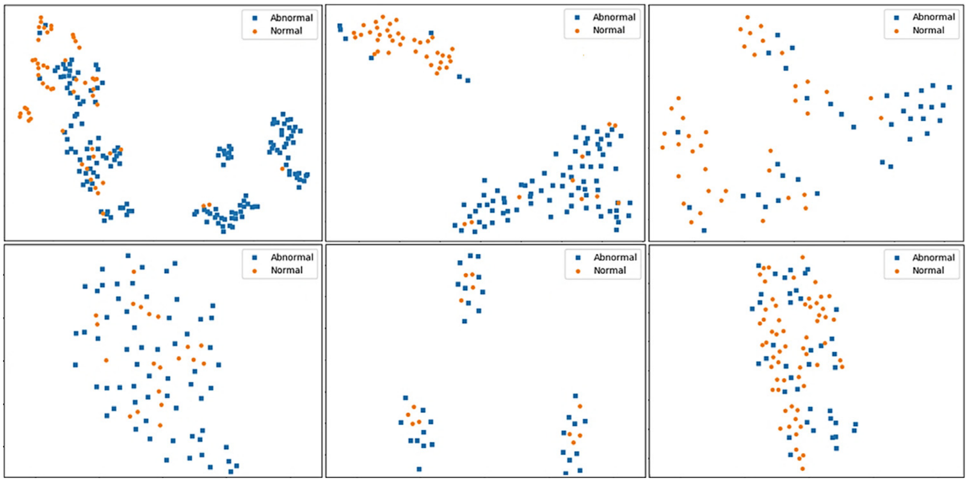


Fig. 9. 2D t-SNE plots of the feature embedding obtained using the AE's latent space representation. Top row: zipper cursor dataset, from left to right: "Set #1", "Set #2" and "Set #4", and bottom row: MVTec dataset, from left to right: "Bottle", "Toothbrush" and "Transistor".

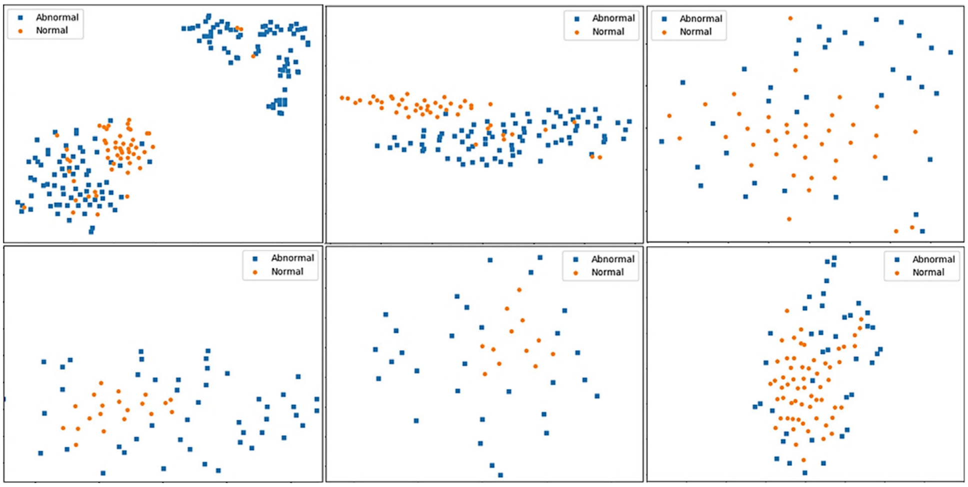


Fig. 10. 2D t-SNE plots of the feature embedding obtained using the proposed method. Top row: zipper cursor dataset, from left to right: "Set #1", "Set #2" and "Set #4", and bottom row: MVTec dataset, from left to right: "Bottle", "Toothbrush" and "Transistor". Normal and abnormal samples are relatively more separated in the proposed method.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 6
Anomaly detection results for the zipper cursor dataset using different one-class classifiers.

Classifier	Metric	Set #1	Set #2	Set #3	Set #4	Set #5	Set #6	Mean
k-NN	AUC	95.67	96.90	95.20	88.08	94.64	96.10	**94.43**
	AP	98.35	98.16	90.90	88.81	96.00	96.43	**94.77**
OC-SVM	AUC	95.35	96.79	94.77	88.89	94.46	95.40	94.27
	AP	98.13	98.26	90.64	89.17	96.06	96.11	94.72
IF	AUC	87.78	89.29	77.99	74.83	75.77	95.85	83.58
	AP	94.38	93.30	56.68	74.58	80.93	96.48	82.72
GMM	AUC	95.05	96.72	94.77	87.85	89.15	95.88	93.23
	AP	97.97	98.22	90.55	88.56	92.88	96.19	94.06
LOF	AUC	95.35	96.83	94.77	87.96	94.10	95.43	94.07
	AP	98.13	98.26	90.31	88.74	96.22	95.23	94.48
KDE	AUC	95.55	96.79	94.11	88.10	93.39	95.88	93.96
	AP	98.22	98.32	91.98	88.84	95.30	96.39	94.16

pretrained network for different datasets. The normal class is colored orange, while the anomalous data is marked in blue. It is clear from the t-SNE plots that the pretrained features embed image samples from the normal class into a fairly compact region. Therefore, it is expected that the density will be much higher around normal test samples than around anomalous test images. This could be the main reason for the success of k-NN-based methods.

The fourth experiment presents the results of the proposed method, as well as baselines and deep learning-based approaches, for the zipper cursor and MVTec datasets. Tables 8 and 9 display the AUC and AP metrics for the zipper cursor and MVTec datasets. The results indicate that the proposed hybrid framework outperforms baseline methods in terms of different metrics, specifically PR, which provides a more accurate representation of an algorithm's performance when there is

Table 7
Anomaly detection results for the MVTec dataset using different one-class classifiers.

Classifier	Metric	Bottle	Cable	Capsule	Hazelnut	Metal Nut	Pill	Toothbrush	Transistor	Mean
k-NN	AUC	99.71	87.73	86.39	95.87	79.28	86.74	100.0	93.06	91.09
	AP	99.90	92.54	96.71	97.90	94.88	97.23	100.0	91.68	96.35
OC-SVM	AUC	99.70	83.95	85.22	95.59	81.15	85.14	100.0	89.82	90.07
	AP	99.90	90.47	96.49	97.73	94.78	96.96	100.0	89.26	95.69
IF	AUC	96.57	77.56	78.86	85.50	69.83	73.23	98.89	81.54	82.74
	AP	98.95	83.03	94.96	90.81	90.89	94.27	99.58	81.03	91.69
GMM	AUC	99.93	87.41	86.92	94.26	82.00	86.54	99.16	90.18	90.80
	AP	99.98	92.13	96.85	97.01	95.13	97.19	99.68	89.75	95.96
LOF	AUC	99.54	88.27	88.99	97.40	83.52	88.09	100.0	91.26	**92.13**
	AP	99.85	92.94	97.40	98.61	95.55	97.52	100.0	90.28	**96.51**
KDE	AUC	97.45	79.14	81.87	89.94	72.73	79.66	100.0	88.03	86.10
	AP	99.35	88.46	95.81	95.57	92.79	95.91	100.0	88.92	94.60

Table 8
Anomaly detection results for the zipper cursor dataset.

Methods	Metric	Set #1	Set #2	Set #3	Set #4	Set #5	Set #6	Mean
CPCAE	AUC	95.67	96.90	95.20	88.08	94.64	96.10	**94.43**
	AP	98.35	98.16	90.90	88.81	96.00	96.43	**94.77**
AE (L_2)	AUC	87.81	95.59	92.21	77.54	78.48	91.11	87.12
	AP	95.24	97.05	87.88	83.01	82.47	90.17	89.30
Deep feature	AUC	85.32	95.77	80.81	74.97	62.15	93.77	82.13
	AP	93.78	96.99	51.10	74.75	73.47	93.31	80.56
NN	AUC	91.06	94.14	90.69	82.72	76.31	91.58	87.75
	AP	96.36	96.50	86.51	85.94	79.88	92.83	89.67
Variation	AUC	91.32	93.45	91.13	77.06	73.53	81.76	84.70
	AP	96.18	95.47	86.37	83.33	77.11	81.06	86.58

Table 9
Anomaly detection results for the MVTec dataset.

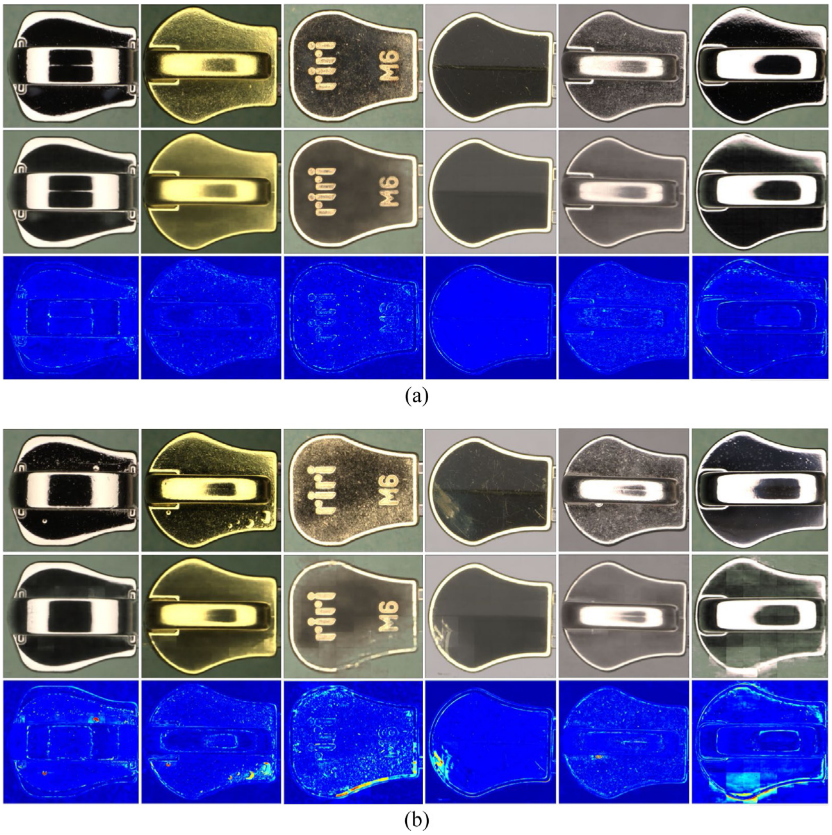
Methods	Metric	Dataset								Mean
		Bottle	Cable	Capsule	Hazelnut	Metal Nut	Pill	Toothbrush	Transistor	
CPCAE	AUC	99.54	88.27	88.99	97.40	83.52	88.09	100.0	91.26	**92.13**
	AP	99.85	92.94	97.40	98.61	95.55	97.52	100.0	90.28	**96.51**
AE (L_2)	AUC	89.77	82.70	54.89	85.23	59.45	79.85	76.68	86.35	75.31
	AP	96.83	89.70	87.12	91.74	87.36	95.59	91.03	85.72	91.39
Deep Feature	AUC	96.71	83.61	88.37	90.29	68.93	71.71	91.67	85.33	83.27
	AP	98.90	90.45	96.40	95.01	90.21	92.38	96.85	85.51	93.89
NN	AUC	81.02	81.30	69.78	52.83	60.16	63.66	95.84	81.12	68.12
	AP	93.78	88.11	91.09	74.29	87.76	88.71	98.46	78.64	87.29
Variation	AUC	79.25	68.20	46.05	49.13	42.95	62.86	86.48	73.09	58.07
	AP	93.12	77.88	83.08	70.93	79.20	87.56	94.36	75.15	81.96

a large skew in the class distribution (Davis & Goadrich, 2006). The next best method on average is AE for the zipper cursor dataset and deep feature classification for the MVTec dataset. This is because AE is able to generalize better on the zipper cursor dataset, which has a simpler appearance than the MVTec dataset. The baseline approaches, including the variation and NN methods, did not produce reliable results. For the zipper cursor dataset, where there are not enough samples for training, e.g., sets #4-6, the performances of the proposed method, along with other approaches, decline. For the MVTec dataset, a high-quality performance can be observed in bottles, toothbrushes, hazelnuts, and transistors, while the proposed method yields comparably poorer results for metal nuts, cables, and pills. This is mostly because the latter objects contain certain random variations on the objects' surfaces, which prevent the model from learning detailed information for most of the image pixels.

The proposed method generates similar and mostly better results compared to recently published deep learning-based methods in terms of the AUC metric for the MVTec dataset, as shown in Table 10. The benchmarking methods consist of VAE (Jinwon & Sungzoon, 2015), AnoGAN (Schlegl et al., 2017), GeoTrans (Golan & El-Yaniv, 2018), GANomaly (Akcay et al., 2018), AE (SSIM) (Bergmann, Lowe, et al., 2019b), VAE-grad (Dehaene et al., 2020), PatchSVDD (Yi & Yoon, 2020), UniStud (Bergmann et al., 2020), DLA (Yoa et al., 2021), and ViV-Ano (Choi & Jongpil, 2022). Our proposed method reaches 92.13% on average on the MVTec dataset. It outperforms both PatchSVDD (Yi & Yoon, 2020) and UniStud (Bergmann et al., 2020) and is only second to DLA (Yoa et al., 2021). Therefore, the proposed CPCAE offers state-of-the-art image anomaly detection performance. It should be noted that the proposed method was introduced and developed specifically for the zipper cursor dataset, and we did not tune the parameters for the MVTec dataset. However, performance can be enhanced for each individual object using hyperparameter tuning. AD in industrial scenarios is a challenging problem and it is difficult to achieve good results by relying only on image-level data reconstruction or generation. Most existing approaches take assistance from additional data used in large pre-trained models, data synthesis by self-supervised means, or designing proxy tasks. Recent deep learning AD methods have relied on three primary proxy tasks consisting of synthetic or simulated abnormal data, consider the spatial information of the neighborhood patch, and using image attributes such as color and orientation. It can be observed from the results that major methods employing additional data, such as the proposed CPCAE method (pretrained models and relative position information), PatchSVDD (pretrained models), UniStud (pretrained models), DLA (pretrained models and self-supervised

Table 10
Anomaly detection results for the MVTec dataset and state-of-the-arts methods.

Methods	Metric	Dataset								Mean
		Bottle	Cable	Capsule	Hazelnut	Metal Nut	Pill	Toothbrush	Transistor	
CPCAE	AUC	99.5	88.2	88.9	97.4	83.5	88.1	100.0	91.2	92.13
VAE (2015)	AUC	89.7	65.4	52.6	87.8	57.6	76.9	69.3	62.6	71.66
AnoGAN (2017)	AUC	62.0	38.3	30.6	69.8	32.0	77.6	74.9	54.9	51.71
GeoTrans (2018)	AUC	74.4	78.3	67.0	63.0	35.9	63.0	97.2	86.9	63.60
GANomaly (2018)	AUC	89.2	75.7	73.2	74.3	78.5	74.3	65.3	79.2	77.53
AE (SSIM) (2019)	AUC	83.4	47.8	86.0	91.6	60.3	83.0	78.4	72.5	75.35
VAE-grad (2020)	AUC	95.1	85.9	88.4	96.5	91.0	90.7	97.1	92.2	92.11
PatchSVDD (2020)	AUC	98.6	90.3	76.7	92.0	94.0	86.1	100.0	91.5	91.15
UniStud (2020)	AUC	91.8	86.5	91.6	93.7	89.5	93.5	86.3	70.1	87.87
DLA (2021)	AUC	89.4	88.3	94.5	96.2	92.6	95.4	95.8	88.3	**92.56**
ViV-Ano (2022)	AUC	93.5	88.1	86.9	88.4	91.4	89.5	92.8	87.6	89.77


(a)

(b)

Fig. 11. Anomaly detection results for (a) normal and (b) defective samples, top to bottom: input image, reconstructed image using AE, and anomaly map; left to right, sets #1 to #6 of zipper cursor dataset.

learning by synthetic abnormal data), and ViV-Ano (pretrained models and relative position information), outperform methods that do not use extra data. Since we do not have enough examples of defective samples for training a two-class classifier and fine-tuning the pre-trained CNN network, we can use the same concept applied in DLA (Yoa et al., 2021) by using self-supervised learning by synthetic or simulated abnormal data. This can further improve the proposed method by fine-tuning the pre-trained network used for feature extraction.

In the final experiment, we present the reconstructed images and anomaly maps generated using the proposed CPCAE method for samples from the zipper cursor dataset shown in Fig. 11 and the MVTec dataset illustrated in Fig. 12. The zipper cursor dataset features anomalies in the form of various defects such as bubbles, residue, scratches, and halos, as depicted in Fig. 11(b). On the other hand, the MVTec dataset contains broken, cracked, cut, colored, contaminated, and misplaced defects, as shown in Fig. 12(b). The results indicate that the proposed method is unable to accurately reconstruct the defective regions, but its generalizability allows for the reconstruction of normal, unseen images within normal specifications. Thus, the defective regions can be easily identified in the anomaly map images.

5. Conclusions and future works

A novel framework for semi-supervised anomaly detection tasks is proposed to introduce a method for zipper cursor visual inspection. The proposed method uses a conditional path-based convolutional autoencoder to tackle the challenges related to high-resolution images in visual inspection scenarios. In addition, we use a binary classification on top of the autoencoder modeling result to leverage transfer learning through feature extraction via a pretrained CNN network and to avoid computing an anomaly score using simple per-pixel comparisons of the autoencoder. We demonstrate state-of-the-art performances on different datasets, including the zipper cursor dataset and the recently introduced MVTec dataset.

The proposed methods have some limitations, including the requirement for the inspected object to have a rigid shape and the use

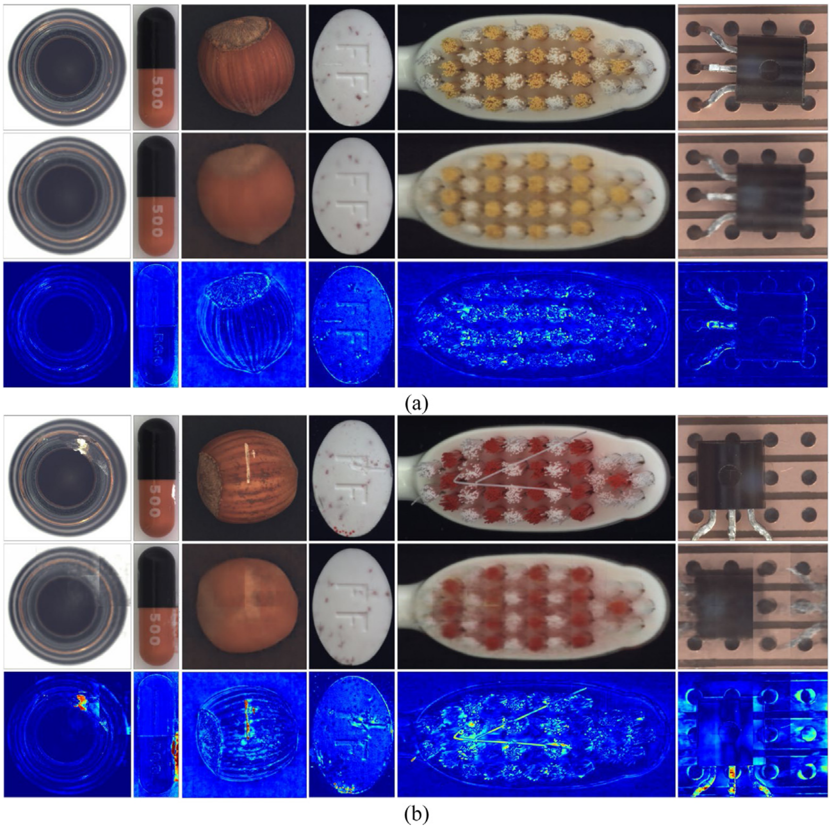


Fig. 12. Anomaly detection results for (a) normal and (b) defective samples, top to bottom: input image, reconstructed image using AE, and anomaly map; left to right, "Bottle", "Capsule", "Hazelnut", "Pill", "Toothbrush" and "Transistor" of the MVTec dataset.. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

of a reference image for image registration. Additionally, the pre-trained network used for feature extraction has not been fine-tuned for the anomaly detection task, and the defect localization is not very precise using the anomaly map generated by the autoencoder. These issues will be addressed in future work. Specifically, alternative deep learning frameworks such as variational autoencoders and generative adversarial networks will be investigated in place of the autoencoders used in the proposed method. Furthermore, regarding deep feature one-class classification, we plan to explore fine-tuning the pre-trained CNN network based on a two-class classification task for each specific anomaly detection dataset.

CRediT authorship contribution statement

Jamal Saeedi: Conceptualization, Methodology, Software, Data curation, Validation, Visualization, Writing – original draft. **Alessandro Giusti:** Conceptualization, Investigation, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work has been supported by the Swiss Innovation Agency (Innosuisse) project 27371.1 IP-ICT. The authors would like to thank RIRI SA for assistance in preparing the zipper cursor datasets and for their valuable feedback. The authors express their gratitude to the anonymous reviewers for their valuable and insightful feedback, which greatly helped enhance this paper.

References

Akcay, S., Atapour-Abarghouei, A., & Breckon, T. P. (2018). GANomaly: Semi-supervised anomaly detection via adversarial training. In *ACCV*.

Amarbayasgalan, T., Jargalsaikhan, B., & Ryu, K. H. (2018). Unsupervised novelty detection using deep autoencoders with density based clustering. *Applied Sciences*, *8*(9), 1468.

Andrew, G. H., et al. (2017). MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv abs/1704.04861.

Andrews, J. T. A., Tanay, T., Morton, E. J., & Griffin, L. D. (2016). Transfer representation learning for anomaly detection. In *Anomaly detection workshop at ICML*.

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495.

Baklouti, R., Mansouri, M., Nounou, M., Nounou, H., & Hamida, A. B. (2016). Iterated robust kernel fuzzy principal component analysis and application to fault detection. *Journal of Computational Science*, *15*, 34–49.

Bergman, L., Cohen, N., & Hoshen, Y. (2020). Deep nearest neighbor anomaly detection. arXiv, abs/2002.10445.

Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2019a). Mvtec AD - A comprehensive real-world dataset for unsupervised anomaly detection. In *IEEE conference on computer vision and pattern recognition* (pp. 9592–9600).

Bergmann, P., Fauser, M., Sattlegger, D., & Steger, C. (2020). Uninformed students: Student-teacher anomaly detection with discriminative latent embeddings. In *CVPR*.

Bergmann, P., Lowe, S., Fauser, M., Sattlegger, D., & Steger, C. (2019b). Improving unsupervised defect segmentation by applying structural similarity to autoencoders. In *Imaging and computer graphics theory and applications*: *vol. 5*, *Proceedings of the 14th international joint conference on computer vision* (pp. 372–380).

Bradski, G. (2000). The openCV library. Dr Dobb &#x27;s. *Journal of Software Tools*.

Breunig, M., Kriegel, H., Ng, R. T., & Sander, J. (2000). LOF: Identifying density-based local outliers. In *International conference on management of data* (pp. 93–104).

Burlina, P., Joshi, N., & Wang, I. (2019). Where's wally now? Deep generative and discriminative embeddings for novelty detection. In *IEEE conference on computer vision and pattern recognition* (pp. 11507–11516).

Chang, S., Du, B., & Zhang, L. (2019). A sparse autoencoder based hyperspectral anomaly detection algorithm using residual of reconstruction error. In *IEEE international geoscience and remote sensing symposium* (pp. 5488–5491).

Chao-Qing, H., et al. (2019). Inverse-transform autoencoder for anomaly detection. arXiv abs/1911.10676.

Choi, B., & Jongpil, J. (2022). ViV-Ano: Anomaly detection and localization combining vision transformer and variational autoencoder in the manufacturing process. *Electronics*, *11*(15), 2306.

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *IEEE conference on computer vision and pattern recognition* (pp. 1800–1807).

Davis, J., & Goadrich, M. (2006). The relationship between precision recall and ROC curves. In *International conference on machine learning* (pp. 233–240).

Dehaene, D., Frigo, O., Combrexelle, S., & Eline, P. (2020). Iterative energy-based projection on a normal data manifold for anomaly localization. arXiv, abs/2002.03734.

Deng, J., et al. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition* (pp. 248–255).

Denkena, B., Dittrich, M. A., Noske, H., et al. (2020). Statistical approaches for semi-supervised anomaly detection in machining. *Production Engineering, Research and Development*, *14*, 385–393.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the 17th international conference on machine learning* (pp. 255–262).

Evangelidis, G., & Psarakis, E. (2008). Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(10), 1858–1865.

Golan, I., & El-Yaniv, R. (2018). Deep anomaly detection using geometric transformations. In *NeurIPS*.

Gramacki, A., & Gramacki, J. (2017). Fft-based fast bandwidth selector for multivariate kernel density estimation. *Computational Statistics & Data Analysis*, *106*, 27–45.

Guo, J., Liu, G., Zuo, Y., & Wu, J. (2018). An anomaly detection framework based on autoencoder and nearest neighbor. In *15th International conference on service systems and service management* (pp. 1–6).

Harrou, F., Kadri, F., Chaabane, S., Tahon, C., & Sun, Y. (2015). Improved principal component analysis for anomaly detection: Application to an emergency department. *Computers & Industrial Engineering*, *88*, 63–77.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR*.

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE conf. comput. vis. pattern recognition* (pp. 2261–2269).

Huo, X., Liu, X., Zheand, E., & Yin, J. (2017). Deep clustering with convolutional auto-encoders. In *International conference on neural information processing* (pp. 373–382).

Jinwon, A., & Sungzoon, C. (2015). *Variational autoencoder based anomaly detection using reconstruction probability*: *Tech. rep. Special lecture on IE 2*, (pp. 1–18). SNU Data Mining Center.

Kawachi, Y., Koizumi, Y., & Harada, N. (2018). Complementary set variational autoencoder for supervised anomaly detection. In *IEEE international conference on acoustics, speech and signal processing* (pp. 2366–2370).

Kemmler, M., Rodner, E., Wacker, E. S., & Denzler, J. (2013). One-class classification with Gaussian processes. *Pattern Recognition*, *46*(12), 3507–3518.

Keyence 2022. https://www.keyence.com/.

Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. In *International conference on learning representations* (pp. 1–14).

Kornblith, S., Shlens, J., & Le, Q. V. (2019). Do better imagenet models transfer better? In *IEEE conference on computer vision and pattern recognition* (pp. 2661–2671).

Krizhevsky, A., & Hinton, G. (2009). *Learning multiple layers of features from tiny images*: *Technical report*, University of Toronto.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *NIPS*.

LeCun, Y. (1998). The mnist database of handwritten digits. http://yannlecun.com/exdb/mnist/.

Liu, F. T., Kai, M. T., & Zhou, Z. H. (2009). Isolation forest. In *Eighth IEEE international conference on data mining* (pp. 413–422).

Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J. (2011). Stacked convolutional auto-encoders for hierarchical feature extraction. In T. Honkela, W. Duch, M. Girolami, & S. Kaski (Eds.), *Lecture notes in computer science*: *vol. 6791*, *Artificial neural networks and machine learning – ICANN*. Berlin, Heidelberg: Springer.

Matsubara, T., Hama, K., Tachibana, R., & Uehara, K. (2018). Deep generative model using unregularized score for anomaly detection with heterogeneous complexity. arXiv preprint arXiv:1807.05800.

Nalisnick, E., Matsukawa, A., Whye, Y., Gorur, D., & Lakshminarayanan, B. (2018). Do deep generative models know what they don't know? arXiv preprint arXiv:1810.09136.

Napoletano, P., Piccoli, F., & Schettini, R. (2018). Anomaly detection in nanofibrous materials by cnn-based self-similarity. *Sensors*, *18*(1), 209.

Nazaré, S., et al. (2018). Are rained CNNs good feature extractors for anomaly detection in surveillance videos? arXiv abs/1811.08495.

Olive, D. J. (2017). Principal component analysis. In *Robust multivariate analysis* (pp. 189–217). Springer.

Oza, P., & Patel, V. M. (2019). One-class convolutional neural network. *IEEE Signal Processing Letters*, *26*(2), 277–281.

Perera, P., & Patel, V. M. (2019). Learning deep features for one-class classification. *Transactions on Image Processing*, *28*(11), 5450–5463.

Pol, A., Berger, V., Germain, C., Cerminara, G., & Pierini, M. (2019). Anomaly detection with conditional variational autoencoders. In *IEEE international conference on machine learning and applications* (pp. 1651–1657).

Psarakis, E. Z., & Evangelidis, G. D. (2005). An enhanced correlation-based method for stereo correspondence with sub-pixel accuracy. In *Tenth IEEE international conference on computer vision* (pp. 1907–912).

Ribeiro, M., Lazzaretti, A. E., & Lopes, H. S. (2018). A study of deep convolutional auto-encoders for anomaly detection in videos. *Pattern Recognition Letters*, *105*, 13–22.

Ruff, L., Görnitz, N., Deecke, L., Siddiqui, S., Vandermeulen, R. A., Binder, A., et al. (2018). Deep one-class classification. In *Proceedings of the 35th international conference on machine learning, vol. 80* (pp. 4393–4402).

Saeedi, J., Dotta, M., Galli, A., et al. (2021). Measurement and inspection of electrical discharge machined steel surfaces using deep neural networks. *Machine Vision and Applications 32*, *21*, 1–15.

Schlegl, T., Seebock, P., Waldstein, S. M., Erfurth, U. S., & Langs, G. (2017). Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International conference on information processing in medical imaging* (pp. 146–157).

Scholkopf, B., Platt, J. C., Shawe-Taylor, J. C., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural Computing*, *13*(7), 1443–1471.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Steger, C., Ulrich, M., & Wiedemann, C. (2018). *Machine vision algorithms and applications* (2nd ed.). Wiley-VCH, Weinheim.

Szegedy, C., Liu, W., Jia, Y., et al. (2014). Going deeper with convolutions. arXiv preprint arXiv:1409.4842.

Szegedy, C., Vanhoucke, V., Ioffe, S., et al. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE computer society conference on computer vision and pattern recognition* (pp. 2818–2826).

Tang, B., & He, H. A. (2017). Local density-based approach for outlier detection. *Neurocomputing*, *241*, 171–180.

Tax, D. M. J., & Duin, R. P. W. (2004). Support vector data description. *Machine Learning*, *54*(1), 45–66.

Vaikundam, S., Hung, T., & Chia, L. T. (2016). Anomaly region detection and localization in metal surface inspection. In *IEEE international conference on image processing* (pp. 759–763).

Van der Maaten, L., & Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2605.

Wang, X., Du, Y., Lin, S., Cui, P., Shen, Y., & Yang, Y. (2019). AdVAE: A self-adversarial variational autoencoder with Gaussian anomaly prior knowledge for anomaly detection. arXiv preprint arXiv:1903.00904.

Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747.

Xu, H., Caramanis, C., & Sanghavi, S. (2012). Robust PCA via outlier pursuit. *IEEE Transactions on Information Theory*, *58*(5), 3047–3064.

Yi, J., & Yoon, S. (2020). Patch SVDD: Patch-level SVDD for anomaly detection and segmentation. In *Proceedings of the Asian conference on computer vision*.

Yildirim, O., Tan, R. S., & Acharya, U. R. (2018). An efficient compression of ECG signals using deep convolutional autoencoders. *Cognitive Systems Research*, *52*, 198–211.

Yoa, S., Lee, S., Kim, C., & Kim, H. J. (2021). Self-supervised learning for anomaly detection with dynamic local augmentation. *IEEE Access*, *9*, 147201–147211.

Jamal Saeedi received his M.Sc. and Ph.D. degrees in Electronic Engineering from Amirkabir University of Tehran, Iran in 2010 and 2015, respectively. He works in the field of signal and image processing and computer vision, specializing particularly in information fusion, pattern recognition and deep learning, road traffic monitoring systems, industrial inspection, and synthetic aperture radar imaging.

Alessandro Giusti holds a Ph.D. in Computer Science from Politecnico di Milano, 2008; since then, he is with the Dalle Molle Institute for Artificial Intelligence (IDSIA), where he is now a permanent Senior Researcher and head of the robotics lab. He took part in a dozen projects in applied research focusing on Deep Learning applications to mobile and industrial robotics; he is the author of more than 80 peerreviewed publications in top conferences and journals, and the recipient of several awards, most of which for innovative applications of deep learning to various fields.