Yerevan State University

FACULTY OF MATHEMATICS AND MECHANICS

Department of Probability Theory and Statistics

APPLIED STATISTICS AND DATA SCIENCE EDUCATIONAL PROGRAM

Gasoyan Hripsime

MASTER'S THESIS

Anomaly detection with self-supervised learning

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of Master of Statistics

YEREVAN 2024

Abstract

Anomaly detection has gained significant importance due to the increasing complexity and volume of data across various domains. This research enhances anomaly detection by leveraging self-supervised learning techniques, specifically DINO's self-distillation and MAE's masked reconstruction. DINO facilitates robust feature representation learning without the need for labeled data, while MAE reconstructs masked image parts to identify anomalies through reconstruction errors. To evaluate the effectiveness of the learned representations, K-means clustering was applied to understand how anomalous clusters differ from non-anomalous ones, assessing the separability and compactness of the clusters. Additionally, Gaussian Mixture Models (GMM) were utilized to model the distribution of normal data and identify outliers, comparing the results before and after implementing self-supervised learning techniques. This comparison demonstrated some improvement in anomaly detection performance. Further analysis of the changes in K-means clustering results after removing anomalies with GMM provided additional insights into the effectiveness of the approach in enhancing anomaly detection.

Anomaly detection with self-supervised learning

Contents

Introduction

Anomaly detection has emerged as a critical area of research and application in recent years, driven by the increasing complexity and volume of data across various domains. This process involves the identification of patterns or behaviors that deviate significantly from the normal behavior within a dataset. Anomaly detection plays a crucial role in various fields such as cybersecurity, finance, healthcare, and industrial monitoring, where detecting outliers or anomalies can provide valuable insights into potential threats, fraud, faults, or unusual events. Recent advancements in this field owe much to the rise of deep learning models, especially self-supervised learning, which has helped create better algorithms that are more effective than before. Self-supervised learning can leverage large amounts of unlabeled data to learn robust representations of normal behavior, making it a scalable and cost-effective solution for anomaly detection. Compared to typical deep learning tasks, anomaly detection poses unique challenges due to the characteristics of the data involved. Anomalies are typically rare occurrences or costly events in the real world. Consequently, the training data for anomaly detection is imbalanced, with a majority of normal data and only a small number of anomalies. Moreover, these anomalous samples can be contaminated with noise, further complicating the detection task. However, self-supervised learning techniques show promising potential in addressing these challenges by learning representations that capture the underlying structure of the data without the need for explicit labels.

Our work leverages DINO's self-distillation and MAE's masked reconstruction to enhance anomaly detection. DINO learns robust feature representations without labels, helping to identify deviations that signal anomalies. MAE focuses on reconstructing masked parts of images, offering a detailed understanding of data and identifying anomalies through reconstruction errors. Together, these methods improve the detection of subtle and overt anomalies. We applied K-means clustering to the learned representations to understand the clustering behavior of normal and anomalous data points, assessing the separability and compactness of the clusters. To further refine the anomaly detection process, we used Gaussian Mixture Models (GMM) to model the

distribution of the normal data and identify outliers. By comparing the results of GMM before and after applying our self-supervised learning techniques, we evaluated the improvement in anomaly detection performance. Additionally, we analyzed how the K-means clustering results changed after using GMM to remove anomalies, providing insights into the effectiveness of our approach in enhancing the detection and isolation of anomalies. This study focuses on evaluating the effectiveness of self-supervised learning techniques in the context of image-based anomaly detection by integrating self-supervised learning techniques, such as DINO and MAE, to create more effective and scalable detection methods.

2 Related Work

Detecting anomalies in images is probably the most researched task by the deep learning anomaly detection community. In various domains, including anomaly detection, deep learning models are classified into supervised, semi-supervised, and unsupervised methods. Each of these approaches offers unique benefits and trade-offs in anomaly detection tasks, catering to different data availability and labeling requirements. Supervised methods, which rely on labeled data, often achieve high performance.

However, annotated data is not commonly available for anomaly detection tasks, making semi-supervised, unsupervised models the only practical options. Unfortunately, these algorithms generally do not perform as well 2 as their supervised counterparts. Despite the challenges, recent advancements in self-supervised learning have emerged as a promising direction for anomaly detection. Self-supervised learning, similar to unsupervised learning, the model learns from unlabelled data without external annotation. It learns a generalizable representation from data by solving a supervised proxy task which is often unrelated to the target task but can help the network to learn a better embedding space. Depending on the nature of the data, a diverse set of tasks, such as colorization (Larsson et al., 2016), mutual information maximization (Hjelm et al., 2019), and predicting geometric transformations (Gidaris et al., 2018) can be used as the supervised proxy task. The proxy task helps the model learn a specialized

representation for anomaly detection, instead of the more general representation learned by unsupervised models.

In previous studies, the formulation of the anomaly detection task varied based on the dataset's characteristics and the availability of data labels. One of the most prevalent formulations is known as one-class anomaly detection (also referred to as LPUE) (Golan and El-Yaniv, 2018; Sabokrou et al., 2019; Chen et al., 2020). In this formulation, one class from the dataset is designated as normal, while all other classes are treated as abnormal. For instance, an illustration of this task involves considering a specific class from CIFAR-10, as normal, and categorizing all other classes as anomalies. Conversely, in multi-class anomaly detection, several classes within the same dataset are regarded as normal during the training phase, while one or more remaining classes are identified as anomalous (Zhang et al., 2022a; Tack et al., 2020).

Self-supervised learning techniques, such as self-predictive and contrastive methods, can play a crucial role in learning robust data embeddings for anomaly detection tasks.

2.1 Self-Predictive Methods

Self-predictive methods learn data embeddings by defining supervised proxy tasks on single samples, focusing on the inherent relationships between a sample and its own contents or its augmented views. These approaches are useful for anomaly detection because they can help in learning representations that highlight the differences between normal and anomalous data points. These tasks often involve predicting the transformation applied to an image. For instance, predicting the degree of rotation of an image can serve as a useful task for learning representations. The Softmax probabilities from the supervised classifier can then be used to define the anomaly score. Another approach is to reconstruct the original input from its transformed version, such as solving jigsaw puzzles or using denoising autoencoders. There are several approaches based on both methods:

1. Geometric transformations were some of the earliest methods used for visual representation learning. For example, Doersch et al. (2015) showed that predicting the relative position of image patches improves object detection representations. Gidaris et al. (2018) used rotation prediction for better

representation learning. Geometric transformation models create a self-labeled dataset by applying different transformations to normal samples, treating the applied transformation as the label. A multi-class network is trained to detect these transformations, and the distribution of the Softmax output is used for anomaly detection.

2. The GEOM Golan and El-Yaniv (2018) method uses geometric transformation learning for anomaly detection and has significantly outperformed state-of-the-art methods on certain datasets. The anomaly score is calculated by combining the log-likelihoods of the conditional probabilities of the applied transformations, approximated by a Dirichlet distribution. However, this method has limitations, such as high variance in predictions for samples not seen during training. To address this, Outlier Exposure (OE) uses some anomalous samples during training to ensure the classifier's output is uniform for anomalies, though this supervised approach may not be practical for all real-world applications.
3. Puzzle-AE Salehi et al. (2020) is a method that trains a U-Net autoencoder to reconstruct puzzled inputs, effective for pixel-level anomaly detection and capturing both low-level and high-level semantic information. This method is enhanced by adversarial training to improve performance.
4. CutPaste Li et al. (2021) improve defect detection by randomly cropping a local patch of an image and pasting it back in a different location. This augmentation creates a more realistic representation of anomalies, and the model is trained to identify these irregularities. A binary classifier used can be parameterized by deep networks, and methods like KDE or GDE are used to calculate the anomaly score from the representation. This approach also facilitates localizing the defective area by computing the anomaly score of an image patch.
5. Schlüter et al. (2021) introduced NSA to create realistic synthetic anomalies by seamlessly cloning patches from source images into destination images. This approach dynamically produces a wide range of anomalies, outperforming state-of-the-art algorithms on real-world datasets like MVTecAD.

Self-predictive methods and various transformation-based approaches can show promising results in anomaly detection. By learning robust representations through self-supervised tasks, these methods can effectively identify anomalies in different types

of data, from images to more general data types. Combining these approaches with supervised methods can further enhance model robustness, making them versatile tools for real-world anomaly detection applications.

2.2 Contrastive Methods in Self-Supervised Learning

Contrastive self-supervised learning aims to learn a feature space where positive samples (similar instances) are clustered closely, while negative samples (dissimilar instances) are pushed further apart. This technique has gained prominence due to its effectiveness in learning robust representations for various tasks, particularly in computer vision. Models like SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) have set the benchmark in this domain by demonstrating significant improvements in image recognition tasks.

2.2.1 SimCLR algorithm

SimCLR, one of the most well-known contrastive learning algorithms, learns representations by maximizing the agreement between different augmented versions of the same image while repelling them from other samples in the batch. The algorithm operates as follows:

1. Randomly sampling a batch and augmenting each image twice, creating pairs of views. These augmented pairs are then passed through an encoder and a projection head to generate latent vectors.
2. SimCLR minimizes a loss function designed to make the latent vectors of augmented pairs similar while repelling others in the batch. This loss function is based on the cosine similarity between latent vectors, adjusted by a temperature hyperparameter:

$$\ell_{i,j} = -\log \frac{\exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\text{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

3. The final objective is to minimize the contrastive loss of overall positive pairs in a mini-batch, averaging the loss contributions from each positive pair:

$$\mathcal{L}_{SimCLR} = \frac{1}{2N}\sum_{i=1}^{N}\Big[l(2i-1,2i) + l(2i,2i-1)\Big]$$

Despite the success of contrastive learning models, they face significant challenges when applied to anomaly detection. Notably, methods like SimCLR and MoCo require negative samples to function correctly. However, in many anomaly detection tasks, we only have access to samples from one class, or the class distribution is highly imbalanced. Additionally, the representation learned may not be directly suitable for anomaly detection, necessitating the definition of an appropriate anomaly score.

2.2.2 The Contrastive Shifted Instance Method

The Contrastive Shifted Instance (CSI) method, proposed by Tack et al. (2020), was one of the first attempts to use contrastive learning for anomaly detection. CSI operates on the principle of instance discrimination, where each data point is treated as its own class and is considered negative relative to other samples. The method introduces specific transformations that generate negative samples from a given point, thereby improving the learned representation for anomaly detection. The CSI loss function is defined similarly to the SimCLR loss but treats augmented samples as negative if drawn from a set of distribution-shifting transformation S:

$$\mathcal{L}_{con-SI} := \mathcal{L}_{SimCLR}\Big(\bigcup_{S\in\mathcal{S}} \mathcal{B}_S; \mathcal{T}\Big)$$

, where $B_S := \{S(x_i)\}_{i=1}^{N}$.

In addition to discriminating each shifted instance, an auxiliary task is added with a Softmax classifier that predicts which transformation was applied, that predicts which shifting transformation is applied for a given input. The classifying shifted instances (cls-SI) loss is defined as below:

$$\mathcal{L}_{cls-SI} := \frac{1}{2B}\frac{1}{K}\sum_{S\in\mathcal{S}}\sum_{\hat{x}_S\in\hat{\mathcal{B}}_S} -\log p_{cls-SI}(y^S = S|\hat{x}_S)$$

The final CSI loss combines these components:

$$\mathcal{L}_{CSI} := \mathcal{L}_{con-SI} + \lambda\mathcal{L}_{cls-SI}$$

The authors of the CSI empirically showed that the norm of the representation $\|z(x)\|$ is indeed a good anomaly score, where z is the representation vector and $\|\cdot\|$ denotes the second norm. This can be explained intuitively by considering that the contrastive loss increases the norm of the in-distribution samples to maximize the cosine similarity of samples generated from the same anchor. Consequently, during the test time, in-distribution samples are mapped further from the origin of the z space, while the representation of other data points, i.e. anomalies, have a smaller norm and hence are closer to the origin. This is an important observation as it helps to solve the problem of defining the anomaly score on a representation that is learned in an unsupervised fashion.

2.2.3 SSD Method

Sehwag et al. (2021) proposed the SSD method, applying SimCLR for out-of-distribution (OOD) and anomaly detection. They extended their algorithm to work with labeled data, using the Supervised Contrastive (SupCon) loss when labels are available. The SupCon loss treats samples from the same class as positives:

$$\frac{1}{2N}\sum_{i=1}^{2N} -\log \frac{\frac{1}{2N_{y_m}-1}\sum_{i=1}^{2N}\mathbf{1}(i\neq m)\mathbf{1}(y_i=y_m)e^{u_m^T u_i/\tau}}{\sum_{i=1}^{2N}\mathbf{1}(i\neq m)e^{u_m^T u_i/\tau}}$$

where u_i is the normalized representation vector, with a projection head h(·) and an encoder f(·). This approach improves performance in both labeled and unlabeled settings for OOD detection.

2.2.4 MSCL Method

Reiss and Hoshen (2021) addressed the hypersphere collapse problem in one-class classifiers with Mean-Shifted Contrastive Loss (MSCL). This loss measures the angular distance relative to the normalized center of the features, instead of the origin. The mean-shifted representation is defined as:

$$\theta(x) = \frac{\phi(x) - c}{\|\phi(x) - c\|}$$

where ϕ(x) is the feature representation and c is the center. The MSCL is then:

$$\mathcal{L}_{MSCL}(x', x'') = \mathcal{L}_{CONS}(\theta(x'), \theta(x'')) = -\log \frac{exp((\theta(x').\theta(x''))/\tau)}{\sum_{i=1}^{2N}\mathbf{1}[x_i \neq x'].exp((\theta(x').\theta(x_i))/\tau)}$$

To prevent pushing normal data too far from the center, they combined MSCL with an angular center loss, enhancing training stability and anomaly detection accuracy. Recent advancements in contrastive self-supervised learning highlight its potential for anomaly detection. Adaptations like CSI, SSD, and MSCL have overcome some of the inherent challenges by introducing new loss functions, auxiliary tasks, and scoring mechanisms. These methods demonstrate that representations learned through contrastive learning can be effectively tailored for anomaly detection, providing robust solutions for identifying outliers in data.

3 Methodology

In this work, we explore the use of advanced self-supervised learning techniques, particularly DINO (Self-Supervised Vision Transformer with Knowledge Distillation) and MAE (Masked Autoencoders), to extract meaningful features from image data, crucial for tackling anomaly detection tasks. Both DINO (Distillation with No Labels) and MAE (Masked Autoencoders) excel in downstream tasks due to their robust feature extraction capabilities. DINO leverages self-distillation without the need for labels, capturing high-level semantic features that enhance performance in tasks like image classification and object detection. MAE, by reconstructing masked portions of images, learns efficient and scalable visual representations, leading to improved accuracy and generalization in various applications such as image recognition and segmentation. Both methods demonstrate strong transfer learning abilities, outperforming traditional supervised approaches in multiple benchmarks. Building on their strong performance in downstream tasks, we investigated how well these methods perform in anomaly detection.

3.1 Distillation with No Labels (DINO)

Distillation with No Labels (DINO) utilizes the Vision Transformer (ViT) architecture, which is shown in Figure 1, capitalizing on the benefits of self-supervised learning

without the need for labeled datasets. The essence of DINO lies in its student-teacher architecture, where both models are initialized with the same transformer structure.

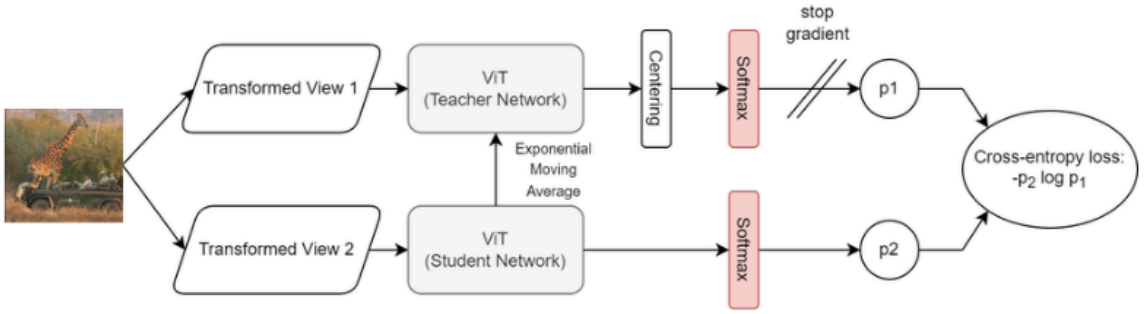


Figure 1: **Self-distillation with no labels.** The model passes two different random transformations of an input image to the student and teacher networks. Both networks have the same architecture but different parameters. The output of the teacher network is centered with a mean computed over the batch.

In this framework, the teacher network's parameters θ_t are updated using an exponential moving average (EMA) of the student network's parameters θ_S. This update mechanism, defined as $\theta_t <- \tau\,\theta_t + (1 - \tau)\,\theta_S$, where τ is the EMA decay rate, stabilizes the training process and helps the student network learn robust representations. Conversely, the student network's parameters are updated using standard gradient descent based on a self-distillation loss.

The loss function in DINO aims to align the outputs of the student network with those of the teacher network. This is achieved through a cross-entropy loss between the softmax outputs of the teacher and student networks, represented by the equation.

$L = - \sum_{i=1}^{N} p_i^t \, log \, p_i^S$, where p_i^t and p_i^S are the softmax probabilities of the teacher and student networks for the i-th image, respectively, and N is the number of images in the batch.

The Vision Transformer (ViT) serves as the backbone architecture for both the teacher and student networks in DINO. The ViT processes images by dividing them into fixed-size patches and embedding each patch into a high-dimensional space. These embeddings are then processed by a series of transformer layers. The input image is divided into non-overlapping patches, and each patch is flattened and linearly transformed into an embedding vector. Positional embeddings are added to retain the

spatial information of patches. Multiple transformer layers process the sequence of patch embeddings. Each layer consists of a multi-head self-attention mechanism and a feed-forward neural network (FFN). The self-attention mechanism computes attention scores and outputs, which are then concatenated and linearly transformed. The FFN further processes these outputs through two linear transformations with a ReLU activation in between.

DINO relies on strong data augmentations to generate multiple views of each image, including random cropping, color jittering, horizontal flipping, and Gaussian blurring. These diverse views help the model learn invariant features. The training process involves applying various augmentations to the same image to create multiple views, passing the augmented views through both the teacher and student networks, calculating the cross-entropy loss between the softmax outputs of the teacher and student networks, and updating the student network using gradient descent while updating the teacher network using the exponential moving average of the student network's parameters.

DINO differentiates itself from other self-supervised methods like BYOL and SimCLR by avoiding the need for negative pairs, which are central to contrastive learning in SimCLR. Instead, it benefits from the consistency mechanism of BYOL but enriches this approach by incorporating multiple views of the same image to refine the student model's learning process. This results in a more nuanced and comprehensive feature extraction that can adapt to a wide range of visual contexts. Figure 2. illustrates the varied applications and results of the DINO architecture in processing different image scenarios, showcasing the model's ability to handle multiple views and augmentations for robust feature extraction.

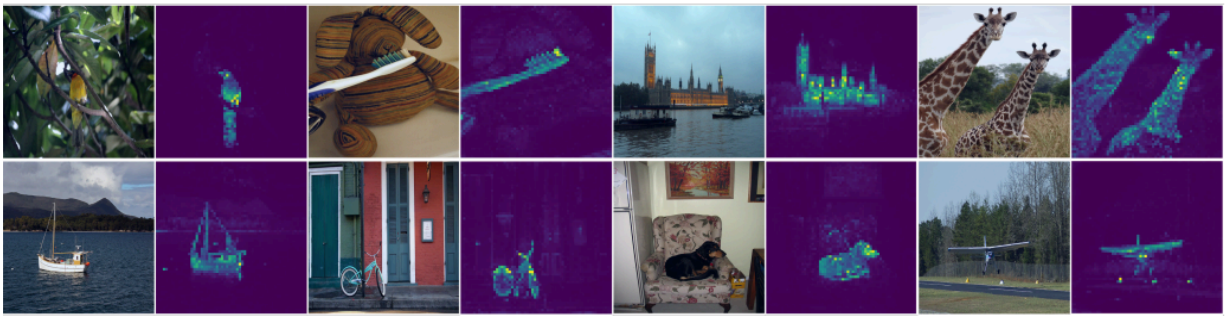


Figure 2: **Self-attention from a Vision Transformer with 8 × 8 patches trained with no supervision**. Examples of DINO architecture applied to diverse scenes, highlighting its capacity to derive stable and comprehensive visual representations without the need for negative pairs, effectively capturing distinct visual patterns across varied contexts.

3.2 Masked Autoencoders

Turning to the second method, the MAE method offers a transformer-based architecture specifically designed to capitalize on self-supervised learning principles. It employs an asymmetric encoder-decoder architecture to learn robust visual representations by reconstructing masked portions of input images, as shown in Figure 3. Unlike traditional supervised learning methods, MAE does not require labeled data, making it highly scalable and efficient for large datasets.

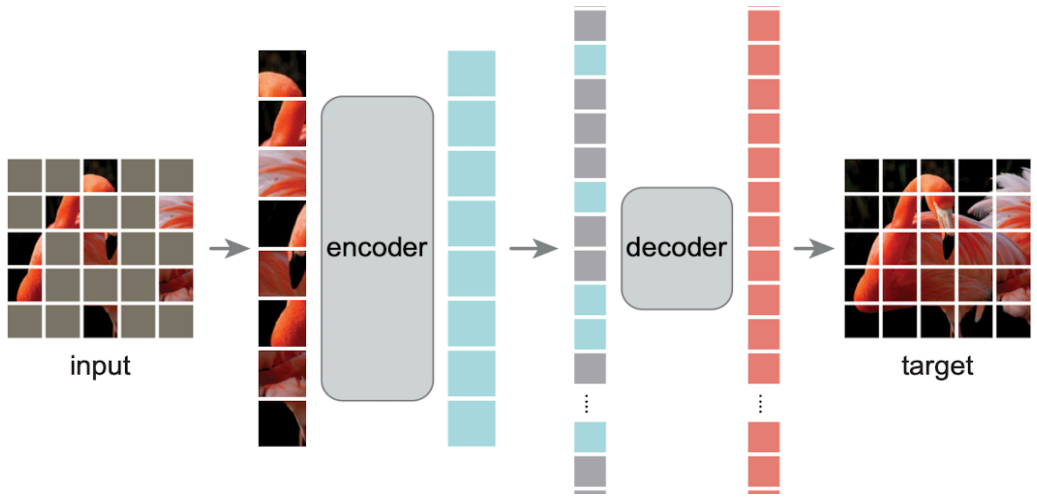


Figure 3. **MAE architecture.** The encoder is applied to the small subset of visible patches. Mask tokens are introduced after the encoder, and the full set of encoded patches and mask tokens is processed by a small decoder that reconstructs the original image in pixels. After pre-training, the decoder is discarded and the encoder is applied to uncorrupted images (full sets of patches) for recognition tasks.

In the MAE framework, an image is divided into fixed-size patches, a subset of which is randomly masked. The encoder processes only the visible patches, ignoring the masked ones, which significantly reduces computational load. This encoder produces a latent representation of the visible patches. The decoder then takes this latent representation and the mask tokens to reconstruct the original image. The reconstruction task forces the model to understand the underlying structure of the image, leading to the learning of useful representations.

The loss function in MAE is based on the reconstruction error, typically measured by the mean squared error (MSE) between the original and reconstructed images. This loss is

computed only on the masked patches, encouraging the model to generate accurate predictions for the missing content based on the context provided by the visible

patches. MAE relies on masking a high proportion of the input image to create a challenging reconstruction task that requires the model to develop a holistic understanding of the image. This strategy largely reduces redundancy and ensures that the model cannot rely on simple extrapolation from neighboring patches, thus encouraging the learning of meaningful features.
One of the main advantages of MAE is its scalability. MAE significantly reduces the computational load during training by processing only a subset of the image patches. This efficiency allows for the training of larger models on larger datasets, resulting in better performance on downstream tasks. Additionally, the representations learned by MAE generalize well across various tasks such as image classification, object detection, and segmentation.

4 Experiments

In our experiments, we constructed a multi-modal anomaly detection dataset using CIFAR-10, COCO 2017, and MVTec. The multi-modal anomaly detection method is that in each iteration, one class was designated as anomalous while the remaining classes were considered non-anomalous. This iterative process simulated an anomaly detection scenario in which the model must identify a single anomalous class among various normal classes. The final performance score for each dataset was calculated as the mean score across all iterations. High-dimensional feature representations were extracted from these images using DINO and MAE models. These feature representations were subsequently processed through various anomaly detection techniques, including dimensionality reduction, clustering, and outlier detection. This multi-step approach leveraged the strengths of both MAE and DINO methods, allowing for a comprehensive evaluation of their capabilities in different anomaly detection scenarios.

4.1 Dataset Description

Finding dedicated datasets for anomaly detection can be challenging. However, several commonly used datasets not specifically designed for anomaly detection can still be effectively utilized for this purpose. In this study, we employed three datasets:

1. CIFAR-10 Dataset: The CIFAR-10 dataset is a widely used dataset for image classification that consists of 60,000 32x32 color images in 10 different classes, with 6,000 images per class. For our anomaly detection experiments, we selected 7 classes and used 1,000 images from each class. The chosen classes were 2: 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', and 'ship'. This subset was resized to 224x224 pixels to match the input requirements of our models.

2. COCO 2017 Dataset: The COCO (Common Objects in Context) 2017 dataset is a large-scale object detection, segmentation, and captioning dataset. It contains over 200,000 labeled images in 80 different categories. For our anomaly detection task, we selected images from 4 classes: “bicycle”, “banana”, “pizza”, and “tv”. Due to the complexity and variability of the COCO dataset, we used a large number of images, approximately 12,500 in total, to capture the diverse scenarios present in this dataset. The COCO dataset is more challenging compared to CIFAR-10 because it contains images with multiple objects, diverse backgrounds, and varying lighting conditions. This complexity allows us to evaluate the model's ability to detect anomalies in a more realistic and varied setting.

3. MVTec AD Dataset: The MVTec Anomaly Detection (MVTec AD) dataset is specifically designed for industrial anomaly detection. It comprises over 5,000 high-resolution images divided into 15 different classes, such as 'bottle', 'carpet', 'pill', and 'wood'. Each class contains images of both normal and anomalous samples, with the anomalies covering a variety of defect types. Because the number of anomalous images is relatively small and includes different kinds of defects, we combined all types of anomalies within each class for our research. This approach allowed us to have a more substantial set of anomalous samples to evaluate the anomaly detection performance comprehensively.

Figure 4. Examples of images from the **COCO dataset**

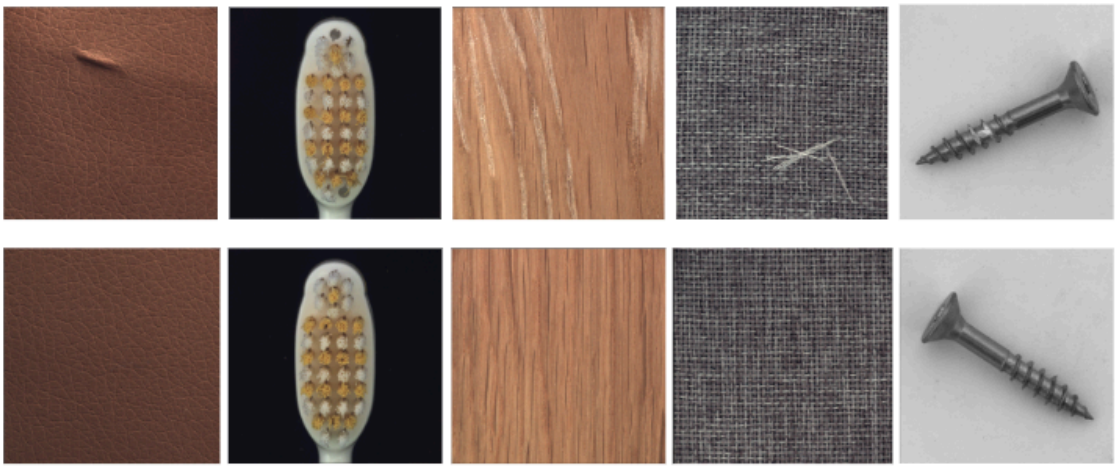


Figure 5. **MVTec AD dataset.** This example presents both normal and anomalous cases from the MVTec AD dataset.

4.2 Features Extraction

Integrating DINO and MAE methodologies into this work establishes a comprehensive framework for feature extraction and visual data analysis. Their combined self-supervised learning strategies ensure that complex visual patterns are captured effectively, laying a strong foundation for accurate anomaly detection. In this comprehensive framework, feature extraction is key to transforming raw image data into meaningful representations that DINO and MAE can utilize. This process identifies

distinctive patterns and attributes within images, providing the models with valuable insights to differentiate between normal and anomalous objects.

The vit_huge_patch14_224.mae model, trained using the Masked Autoencoders (MAE) approach, provides high-dimensional representations that capture the intricate details of input images. This model processes images by dividing them into fixed-size patches and embedding each patch into a high-dimensional space. These embeddings are then processed by a series of transformer layers, which include multi-head self-attention mechanisms and feed-forward neural networks (FFNs). This architecture helps the model capture complex patterns and relationships within the images, making it well-suited for identifying anomalies.

Due to resource constraints, we opted to use the smaller ViT model, vit_small_patch16_224.dino small model, trained using the Distillation with No Labels (DINO) approach, which leverages a self-supervised learning framework to learn robust visual representations without the need for labeled data. Like the MAE model, it processes images by dividing them into fixed-size patches and embedding each patch into a high-dimensional space. These embeddings are processed by transformer layers that include multi-head self-attention mechanisms and FFNs. The DINO model's teacher-student framework allows it to learn from augmented views of the same image, leading to robust and invariant feature representations.

To ensure compatibility between the datasets and the input requirements of DINO and MAE models, we transform the images to a standard size of 224 x 224 pixels. This transformation aligns the input dimensions with the models' training configuration, ensuring optimal feature extraction. Once resized, the images are processed through the DINO and MAE architectures, where each layer captures progressively richer and more abstract features. The extracted features from the two models were then used for anomaly detection. The extracted features shape for DINO representations were (n, 197, 384), and for MAE were (n, 257, 1280). These high-dimensional features, which capture the essential details and patterns of the input images, were subsequently processed in our anomaly detection pipeline. The Anomaly detection pipeline include various techniques such as dimensionality reduction, clustering, and outlier detection to identify anomalies in the datasets.

4.3 High-Dimensional Analysis

The first approach we employed was a simple high-dimensional analysis to determine if the anomalous data points could be separated from the non-anomalous points. For each iteration, we computed the mean of the extracted features for each class in the dataset. We then measured the Euclidian distances between all points labeled as anomalous and the computed class means. Specifically, we evaluated whether anomalous data points were closer to the anomalous class mean and farther from the non-anomalous class mean. This method helped us assess the separability of the dataset in high-dimensional space. However, our results indicated that the datasets were not easily separable using this high-dimensional analysis alone. The distances between anomalous data points and the non-anomalous class means did not provide clear separability, prompting us to explore more sophisticated methods.

4.4 Dimensionality Reduction and Visualization

To gain further insights, we applied dimensionality reduction techniques. We first reduced the dimensionality of the extracted features to 100 dimensions using Principal Component Analysis (PCA). We then used t-distributed Stochastic Neighbor Embedding (t-SNE) to visualize the data in a two-dimensional space. t-SNE is a non-linear dimensionality reduction technique particularly well-suited for visualizing high-dimensional data by minimizing the divergence between the probability distributions of the data in high-dimensional and low-dimensional spaces.

1. DINO representation results: Through our experiments with CIFAR-10 and COCO 2017, we found that the initial high-dimensional analysis did not yield clear separability between anomalous and non-anomalous data points. This prompted us to employ PCA for dimensionality reduction followed by t-SNE for visualization. The t-SNE plots for CIFAR-10 and COCO 2017 revealed that while some degree of clustering was observable, the separation was not distinct enough to reliably identify anomalous data points. The clusters formed in the t-SNE plots indicated some level of separability, but it was not sufficient to definitively categorize data points as normal or anomalous.

   In contrast, the t-SNE plots for the MVTec AD dataset showed a much clearer separability between anomalous and non-anomalous samples. Each class in the

MVTec AD dataset exhibited distinct clusters for normal and anomalous data points, indicating that this dataset is more suitable for anomaly detection tasks. The clear separation in the t-SNE plots suggests that the features extracted using DINO from the MVTec AD dataset are more discriminative, making it easier to identify anomalies.

2. MAE representation results**:** For the Masked Autoencoders (MAE) approach, we also applied dimensionality reduction techniques, reducing the dimensionality of the extracted features to 100 dimensions using PCA. We then used t-SNE for visualization. The t-SNE plots for the CIFAR-10 and COCO 2017 datasets did not reveal any clear clusters. This indicates that the MAE model did not produce features that were as separable as those from the DINO model for these datasets. The lack of distinct clustering suggests that the MAE-extracted features struggled to differentiate between normal and anomalous data points in these datasets. However, the t-SNE plots for the MVTec AD dataset showed some level of clustering, but not as distinct as those observed with DINO. While the MAE-extracted features were able to form some clusters indicating normal and anomalous data points, the separation was less pronounced compared to the DINO-extracted features. This suggests that while MAE can capture useful features for anomaly detection, it may not be as effective as DINO in creating highly discriminative feature representations for certain datasets.

By combining dimensionality reduction and visualization techniques, we were able to gain a deeper understanding of the datasets' structures and the challenges involved in anomaly detection. This approach allowed us to identify the limitations of the simple high-dimensional analysis and the need for more sophisticated methods of ineffective anomaly detection.

The DINO model provided more distinct clusters, especially for the MVTec AD dataset, which results are shown in Figure 6. indicating its strong capability in feature extraction for anomaly detection. The MAE model, while still useful, showed less pronounced clustering, particularly for CIFAR-10 and COCO 2017, which are shown in Fig.7 and Fig. 8, suggesting that additional methods or enhancements might be necessary to improve its anomaly detection performance.

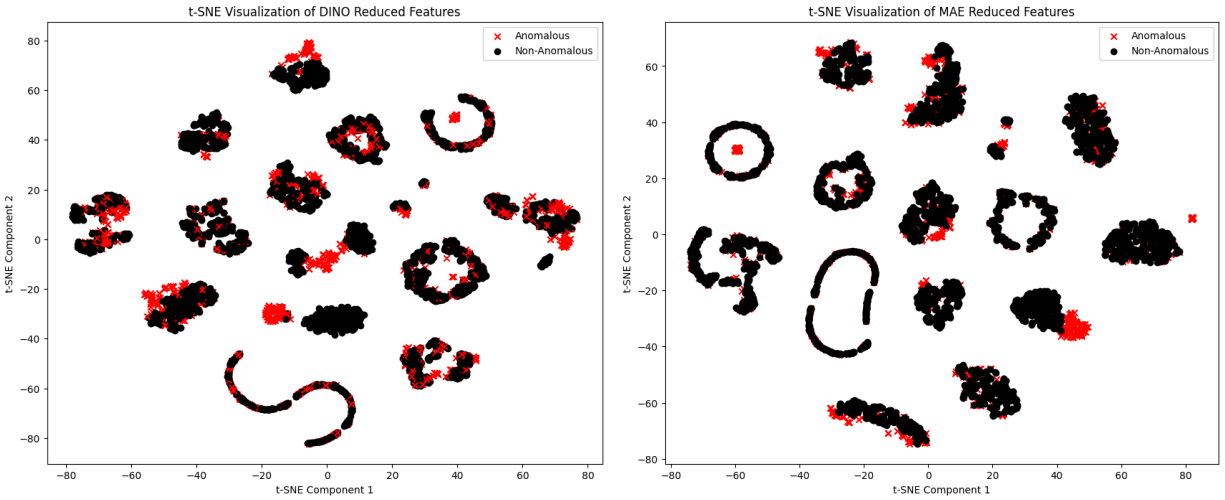


Figure 6. **MVTec dataset**: (a) t-SNE visualization of 100-dimensional DINO reduced features, (b) t-SNE visualization of 100-dimensional MAE reduced features.

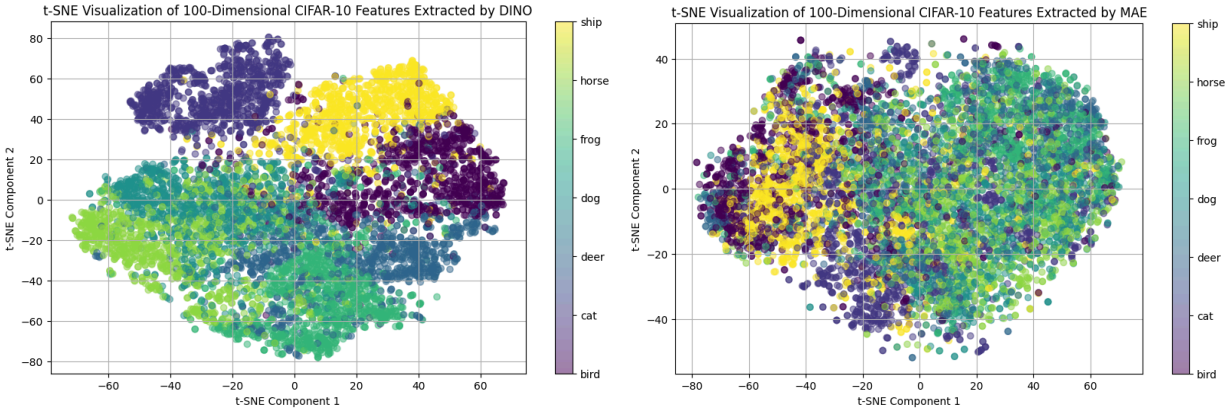


Figure 7. **CIFAR 10 dataset**: (a) t-SNE visualization of 100-dimensional DINO reduced features, (b) t-SNE visualization of 100-dimensional MAE reduced features.

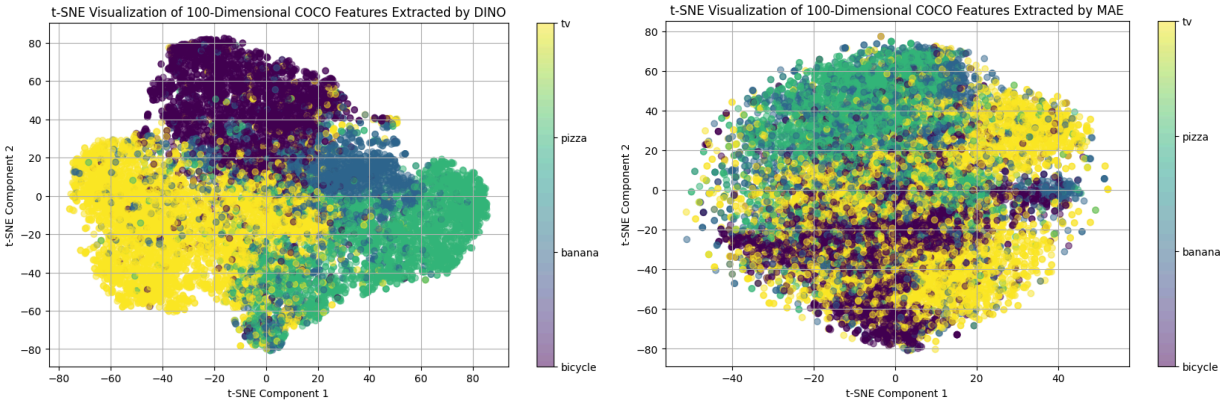


Figure 8. **COCO dataset**: (a) t-SNE visualization of 100-dimensional DINO reduced features, (b) t-SNE visualization of 100-dimensional MAE reduced features.

4.5 K-means Clustering

In this section, we explore the effectiveness of the learned representations from self-supervised learning methods, by applying K-means clustering. The aim is to understand how well these representations can separate normal and anomalous data points. We reduced the dimensionality of the extracted features to 100 dimensions using Principal Component Analysis (PCA). This step ensures that the most significant components of the data are retained, making the clustering process more efficient and effective. Thus, we applied K-means clustering to further investigate the clustering performance.

K-means is a clustering method that partitions the dataset into K clusters. The algorithm assigns each data point to the nearest cluster centroid and updates the centroids iteratively to minimize the within-cluster variance. To determine the optimal number of clusters K, we used the elbow method, which involves plotting the explained variance (or within-cluster sum of squares) against the number of clusters and selecting the point where the curve bends (the "elbow"). This is shown in Figure 9, which illustrates the elbow plot indicating the appropriate K values for the datasets.

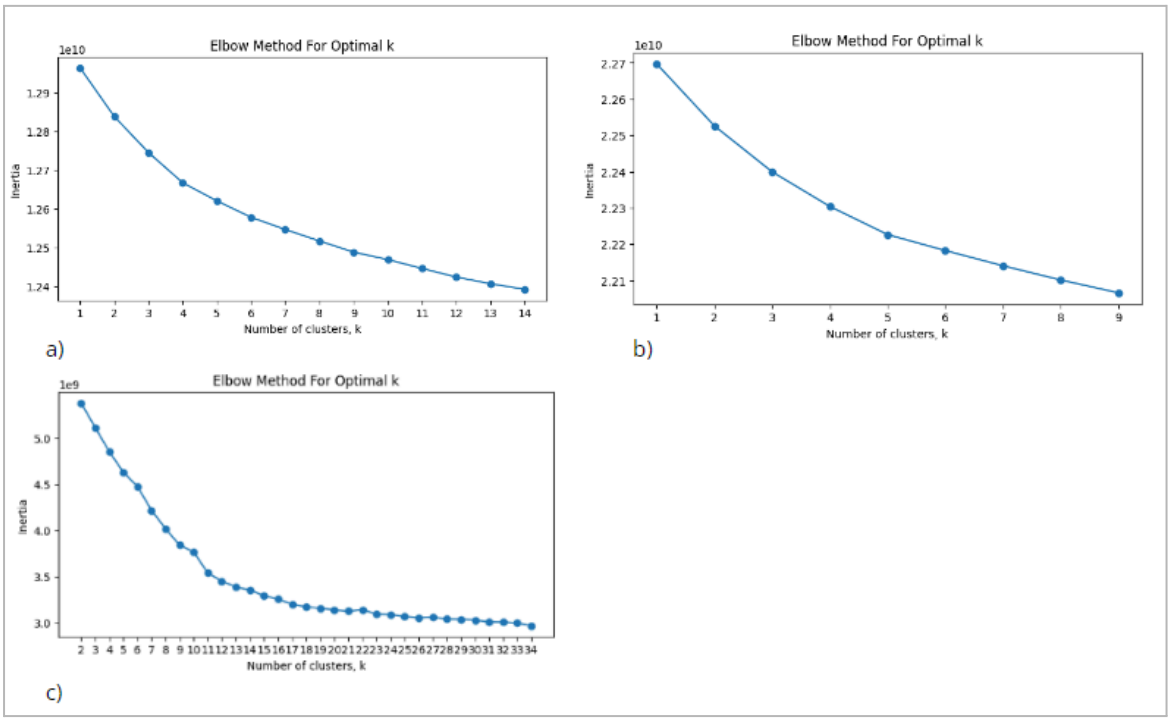


Figure 9**. Elbow point for optimal clusters in k-means:** (a) CIFAR, chosen k is 6, (b) COCO, chosen k is 5, (c) MVTec, chosen k is 9.

4.5.1 Evaluation Score

In anomaly detection, various metrics can be used to evaluate performance, including Precision-Recall curves, F1 scores, and AUC-ROC. However, for this analysis, the AUC-ROC (Area Under the Receiver Operating Characteristic Curve) metric is particularly appropriate. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings, and the AUC-ROC score is the area under this curve. This score provides a single measure of overall performance, where an AUC-ROC score of 1 indicates perfect separation, while a score of 0.5 suggests no discrimination (equivalent to random guessing).

AUC-ROC is especially useful in anomaly detection because it captures the model's ability to rank anomalies higher than normal points, which is crucial when dealing with these datasets. In our anomaly detection task, we calculated the AUC-ROC for each iteration, where in each iteration, one class was designated as anomalous and the remaining classes as non-anomalous. The final performance score for each dataset was computed as the mean AUC-ROC score across all iterations. This approach allows for a comprehensive evaluation of the model's ability to distinguish between normal and anomalous data points.

Despite these efforts, the results indicated that while the DINO model's representations were generally better than those of the MAE model, the performance was still not optimal (shown in Fig. 10). Specifically, for the COCO dataset, which consists of four different and diverse subsets, the results were not satisfactory. This prompted us to investigate the impact of outliers on the clustering performance and led to the use of Gaussian Mixture Models (GMM) for anomaly detection.

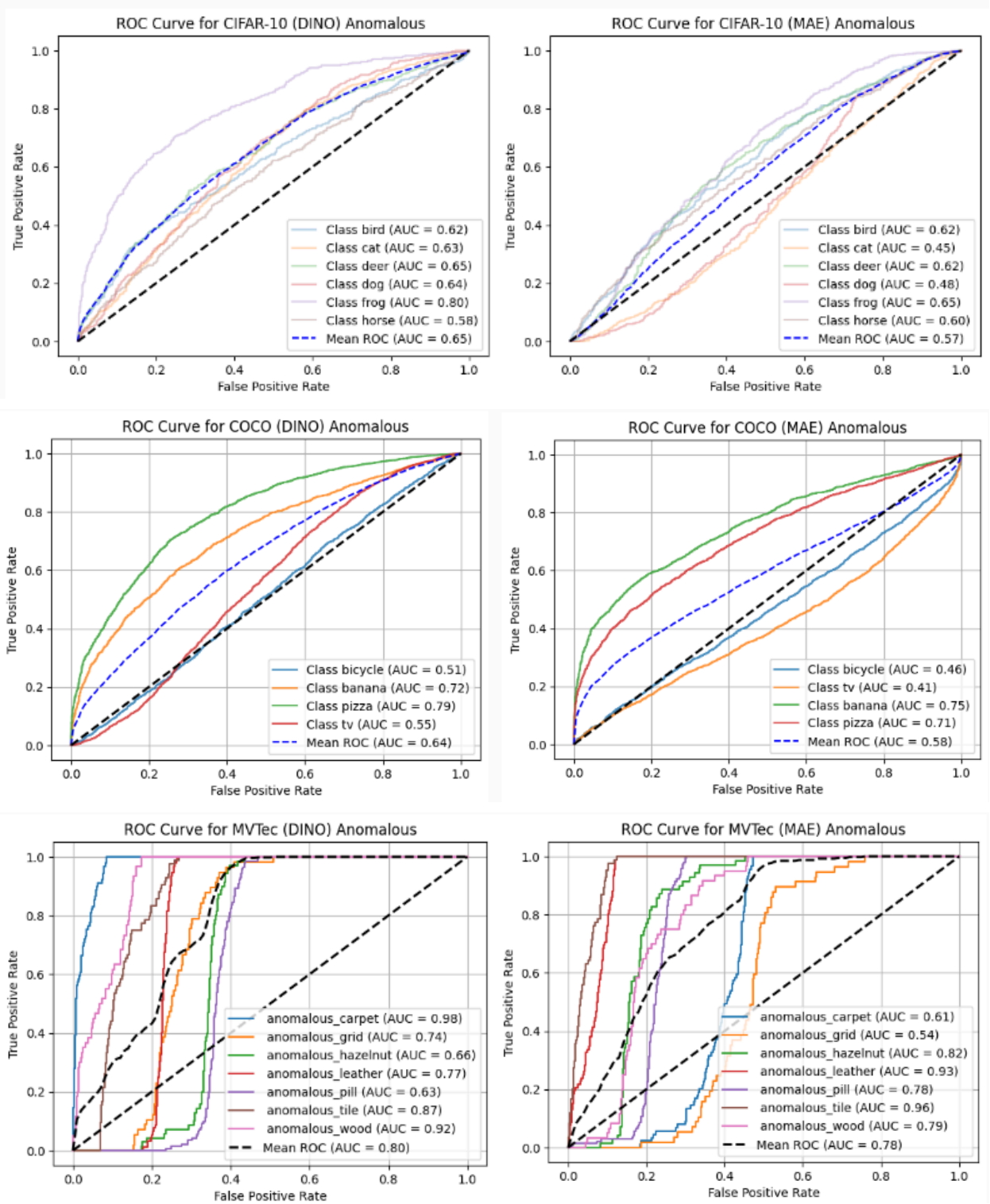


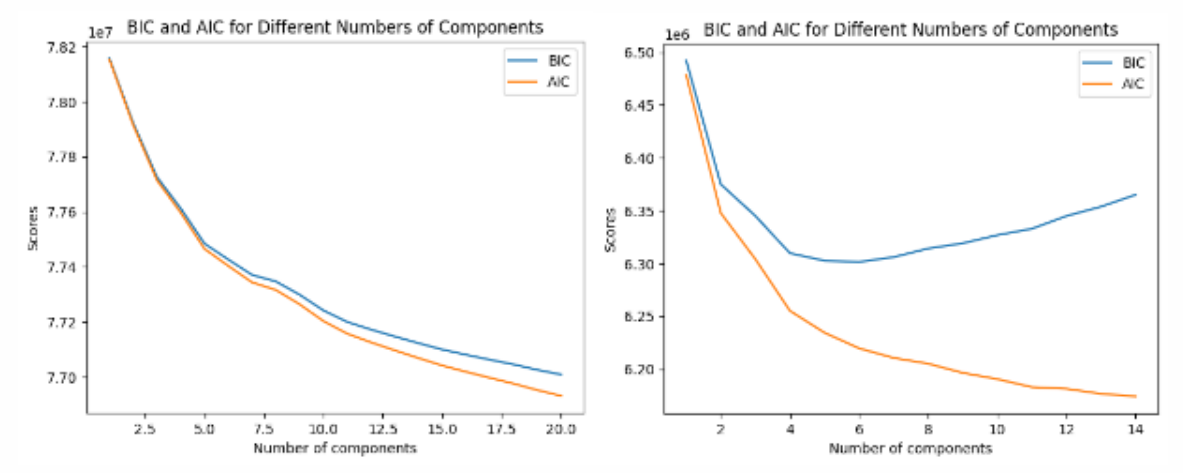
Figure 10: **ROC curves** using K-means for detecting anomalies in the CIFAR-10, COCO, and MVTec datasets using DINO (left) and MAE (right) representations. Each curve represents a different type of anomalous data with corresponding AUC scores. The mean ROC AUC is also indicated for each model.

4.6 Gaussian Mixture Models

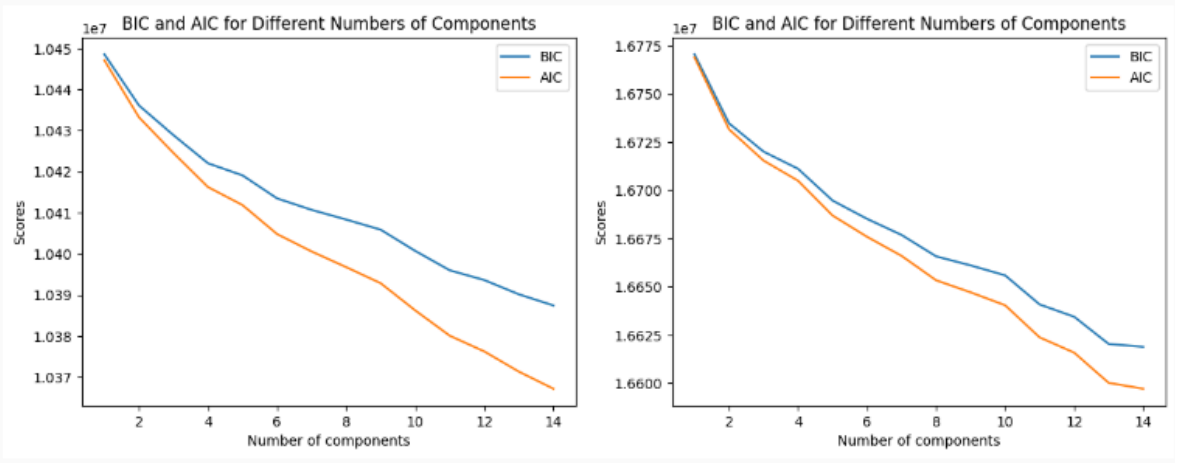
To further enhance our anomaly detection approach, we employed Gaussian Mixture Models (GMM), a probabilistic model that assumes all data points are generated from a mixture of several Gaussian distributions with unknown parameters. Unlike K-means, which assigns each data point to a single cluster, GMM allows for soft clustering where each data point has a probability of belonging to each cluster. This flexibility makes GMM more powerful in capturing the complex underlying structure of the data.

A GMM is defined as a weighted sum of K Gaussian component densities. The model parameters include the mixing coefficients, the means, and the covariances of the Gaussian components. These parameters are estimated using the Expectation-Maximization (EM) algorithm, which iteratively maximizes the likelihood of the observed data. In the E-step, the algorithm calculates the expected value of the log-likelihood function with respect to the current parameter estimates. In the M-step, it maximizes this expectation to update the parameter estimates.

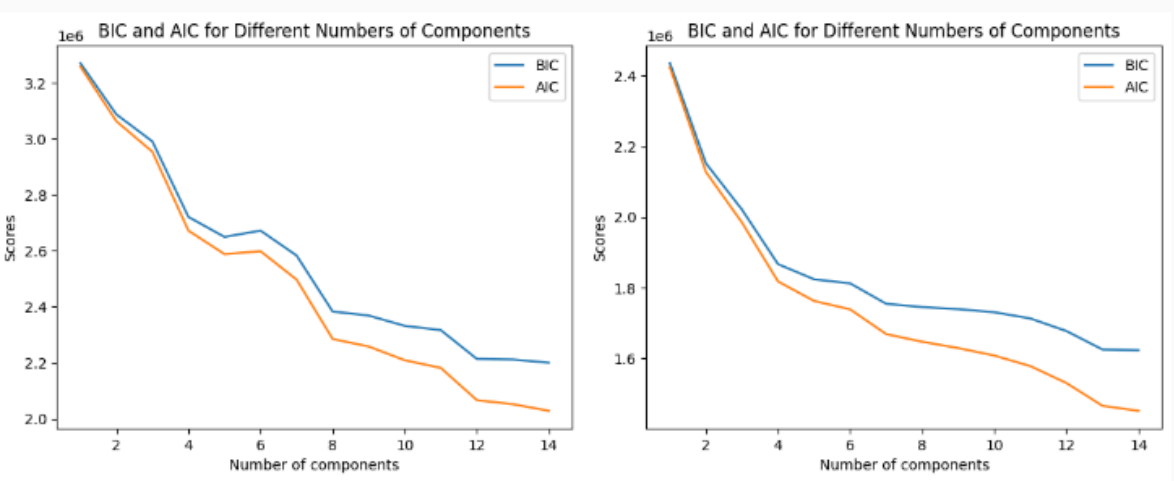
To determine the optimal number of components K in GMM, we used the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both criteria balance model fit and complexity but have different formulations. The AIC is calculated as $2k - 2ln(\widehat{L})$, where k is the number of parameters in the model and $\widehat{L}$ is the maximized value of the likelihood function. The BIC is calculated as $ln(n)k - 2ln(\widehat{L})$, where n is the number of data points. AIC penalizes the number of parameters to avoid overfitting but favors models with a high likelihood. BIC imposes a stronger penalty for the number of parameters, making it more conservative than AIC in selecting model complexity. These criteria helped us select the optimal model that best balances complexity and fit. After determining the optimal number of components K, we also considered different covariance types for the GMM. The covariance type determines the shape of the clusters and can significantly impact the model's performance. The choice of covariance type affects the flexibility of the model and its ability to fit the data. In our analysis, we tested Full and Spherical covariance types and selected the one that resulted in the best performance based on AIC and BIC scores. Each covariance type component has specific characteristics: for full, it has its own general covariance matrix, allowing for elliptical clusters with any orientation, and for spherical, it has a single variance, leading to spherical clusters of different sizes but the same shape.



a1) a2)



b1) b2)



c1) c2)

Figure 11. **AIC and BIC plots for each dataset with different representations:**

(a1) CIFAR using DINO representation, (a2) CIFAR using MAE representation,
(b1) COCO using DINO representation, (b2) COCO using MAE representation,
(c1) MVTec using DINO representation, (c2) MVTec using MAE representation.

4.7 Results

Our comprehensive analysis involved applying K-means clustering and Gaussian Mixture Models (GMM) for anomaly detection including CIFAR-10, COCO 2017, and MVTec. This evaluation aimed to determine the effectiveness of self-supervised learning representations from DINO and MAE models in distinguishing normal and anomalous data points. The methods applied included t-SNE visualizations, initial K-means clustering, and K-means clustering after GMM-based outlier removal. The AUC-ROC scores served as the primary metric for evaluating the clustering performance.

To gain initial insights, we applied t-SNE to the dimensionality-reduced data (100 dimensions via PCA) from both DINO and MAE models, followed by K-means clustering, which revealed that DINO representations formed more distinct clusters and had higher AUC-ROC scores compared to MAE.

To further enhance our anomaly detection approach, we employed Gaussian Mixture Models (GMM) for outlier detection, followed by the reapplication of K-means clustering. Unlike K-means, GMM allows for soft clustering where each data point has a probability of belonging to each cluster. This flexibility makes GMM more powerful in capturing the complex underlying structure of the data. We used GMM specifically for outlier detection in our datasets. After fitting the GMM, we identified outliers as data points with low probability under the fitted model. These outliers were considered anomalies. The parameters for GMM were estimated using the Expectation-Maximization (EM) algorithm. To determine the optimal number of components K in GMM, we used the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Both criteria helped us select the optimal model that best balances complexity and fit.

A notable observation emerged regarding the covariance type used in GMM. When using a full covariance matrix, the results were significantly better. The full covariance matrix allows each Gaussian component to have its own general covariance matrix, capturing the full range of correlations between variables, resulting in elliptical clusters of any orientation. This flexibility results in better adaptation to the actual data distribution. However, switching to a spherical covariance matrix led to a drastic change, with the results becoming only slightly better than random guessing. The spherical covariance matrix restricts each component to have its own single variance, leading to spherical clusters of different sizes but the same shape. This limitation results in less

flexible clustering, which can fail to capture the true structure of the data, leading to poorer performance. Interestingly, for the MVTec dataset, a spherical covariance matrix worked better than the full covariance matrix.

After removing the outliers detected by GMM, we reapplied K-means clustering to the refined dataset. This step aimed to improve clustering by eliminating noise and improving the overall performance. The results showed a significant improvement in the AUC-ROC scores after removing outliers identified by GMM (shown in Fig. 12), with a mean increase of at least 2% for each dataset. This demonstrates the importance of handling outliers in improving anomaly detection performance. Specifically, the results for CIFAR-10 were notably better than those for COCO 2017. This can be attributed to the less complex nature of CIFAR-10 compared to COCO 2017, which includes a more diverse set of images and categories, making the anomaly detection task inherently more challenging.

For the MVTec dataset, the results were good compared to CIFAR-10 and COCO 2017, even though we combined different anomalous variants for each class. This indicates the robustness of the method in handling diverse types of anomalies within the same class. Notably, MVTec is unique in that MAE representations were particularly effective, showing similar performance as DINO representations. The implementation details and source code for the analysis can be found in our GitHub repository: https://github.com/h-gasoyan/Self-supervised-learning-for-anomaly-detection/tree/main

Method	CIFAR10	COCO	MvTec
MAE	57.1	58.23	78
DINO	65.2	63.4	80

Method	CIFAR10	COCO	MvTec
MAE	60.2	63.4	80
DINO	71.5	68	85

Table 1. **Comparison of anomaly detection results using GMM for outlier detection followed by K-means clustering**. The first table shows the ROC AUC scores for CIFAR-10, COCO, and MVTec datasets using K-means clustering, while the second table displays the results after applying GMM for outlier detection followed by K-means clustering. Each method's performance is evaluated using both MAE and DINO models.

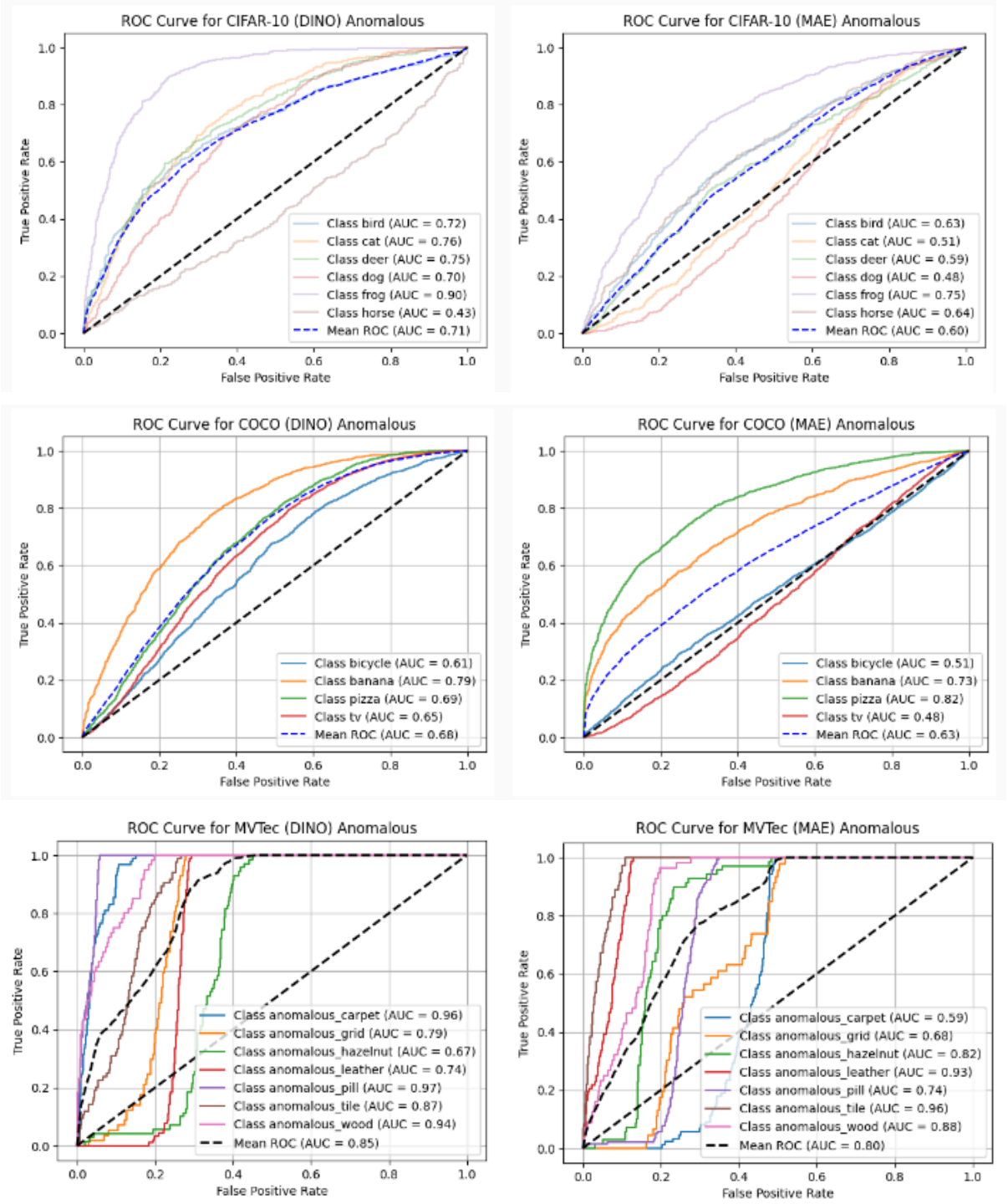


Figure 12: **ROC curves** using K-means for detecting anomalies in the CIFAR-10, COCO, and MVTec datasets after GMM-based outlier removal, using DINO (left) and MAE (right) representations. Each curve represents a different type of anomalous data with corresponding AUC scores. The mean ROC AUC is also indicated for each model.

5 Conclusion

The results of our analysis demonstrated that even though both the DINO and MAE models show good performance in downstream tasks, their effectiveness in anomaly detection was poor. Specifically, while the DINO model's representations were generally better than those of the MAE model, the overall results indicated that the learned representations that are beneficial for downstream tasks do not translate well to anomaly detection. The use of GMM for outlier detection and reapplication of K-means clustering did improve the performance to some extent, but the anomaly detection outcomes were still suboptimal. This highlights a critical gap in the ability of these models to generalize across different types of tasks, suggesting that representations optimized for general tasks may not be well-suited for detecting anomalies.

References

H. Hojjati, Thi Kieu Khanh Ho, Naregs Armanfarda, 2024. Self-Supervised Anomaly Detection: A Survey and Outlook, arXiv:2205.05173v5

Larsson, G., Maire, M., Shakhnarovich, G., 2016. Learning representations for automatic colorization, in: European Conference on Computer Vision (ECCV).

Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y., 2019. Learning deep representations by mutual information estimation and maximization, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=Bklr3j0cKX.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728.

Golan, I., El-Yaniv, R., 2018. Deep anomaly detection using geometric transformations. Advances in neural information processing systems 31.

Sabokrou, M., Khalooei, M., Adeli, E., 2019. Self-supervised representation learning via neighborhood-relational encoding, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV).

Zhang, X., Mu, J., Zhang, X., Liu, H., Zong, L., Li, Y., 2022a. Deep anomaly detection with self-supervised learning and adversarial training. Pattern Recognition 121, 108234. URL: https://www.sciencedirect.com/science/article/pii/S0031320321004155, doi:https://doi.org/10.1016/j.patcog.2021.108234.

Chen, B., Zhang, J., Zhang, X., Dong, Y., Song, J., Zhang, P., Xu, K., Kharlamov, E., Tang, J., 2022. Gccad: Graph contrastive learning for anomaly detection. IEEE Transactions on Knowledge and Data Engineering.

Zhang, X., Mu, J., Zhang, X., Liu, H., Zong, L., Li, Y., 2022a. Deep anomaly detection with self-supervised learning and adversarial training. Pattern Recognition 121, 108234. URL: https://www.sciencedirect.com/science/article/pii/S0031320321004155, doi:https://doi.org/10.1016/j.patcog.2021.108234.

Doersch, C., Gupta, A., Efros, A.A., 2015. Unsupervised visual representation learning by context prediction, in: Proceedings of the IEEE international conference on computer vision, pp. 1422–1430.

Gidaris, S., Singh, P., Komodakis, N., 2018. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:1803.07728.

Golan, I., El-Yaniv, R., 2018. Deep anomaly detection using geometric transformations. Advances in neural information processing systems 31.

Salehi, M., Eftekhar, A., Sadjadi, N., Rohban, M.H., Rabiee, H.R., 2020. Puzzle-ae: Novelty detection in images through solving puzzles. arXiv:2008.12959

Li, C.L., Sohn, K., Yoon, J., Pfister, T., 2021. Cutpaste: Self-supervised learning for anomaly detection and localization, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9664–9674.

Schl¨uter, H.M., Tan, J., Hou, B., Kainz, B., 2021. Self-supervised out-ofdistribution detection and localization with natural synthetic anomalies (nsa). arXiv preprint arXiv:2109.15222.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9729–9738.

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9729–9738.
He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., 2020. Momentum contrast for unsupervised visual representation learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9729–9738.

Chen, C., Xie, Y., Lin, S., Qiao, R., Zhou, J., Tan, X., Zhang, Y., Ma, L., 2021a. Novelty detection via contrastive learning with negative data augmentation, in: Zhou, Z.H. (Ed.), Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI21, International Joint Conferences on Artificial Intelligence Organization. pp. 606–614. URL: https://doi.org/10.24963/ijcai.2021/84, doi:10.24963/ijcai.2021/84. main Track.

Tack, J., Mo, S., Jeong, J., Shin, J., 2020. Csi: Novelty detection via contrastive learning on distributionally shifted instances. Advances in neural information processing systems 33, 11839–11852.

Sehwag, V., Chiang, M., Mittal, P., 2021. {SSD}: A unified framework for self-supervised outlier detection, in: International Conference on Learning Representations. URL: https://openreview.net/forum?id=v5gjXpmR8J.

Reiss, T., Hoshen, Y., 2021. Mean-shifted contrastive loss for anomaly detection. arXiv preprint arXiv:2106.03844.

M. Caron, H. Touvron, I. Misra, 2021. Emerging Properties in Self-Supervised Vision Transformers, arXiv:2104.14294v2, https://huggingface.co/timm/vit_small_patch16_224.dino.

Kaiming He, Xinlei Chen, Saining Xi, 2021. Masked Autoencoders Are Scalable Vision Learners, arXiv:2111.06377v3, https://huggingface.co/timm/vit_huge_patch14_224.mae.

OpenAI. (2022). ChatGPT (3.5 version) [Large language model]. https://chat.openai.com/chat)