Video Feed Anomaly - Report

Jan Gorczyński, Michał Opioła, Jan Kapusta, Marta Figurska

December 2025

1 Introduction

In this project, we investigate models for visual anomaly detection in images captured on railway platform. Specifically, we focus on Autoencoders and Generative Adversarial Networks, which are capable of learning compact latent representations of visual patterns and modeling the underlying data distribution.

The objective of this work is to compare the effectiveness of AE-based and GAN-based approaches in detecting trains on the platform. The comparison aims to identify strengths and limitations of each model type and to assess their suitability for object detection.

2 Data Preparation

To extract data, we first used yt_dlp to download raw footage of the railway platform from a youtube livestream. Then using ffmpeg, we extracted frames in intervals of 10 seconds. Finally, we saved each frame with a timestamp. We calculated each timestamp by adding multiples of 10 to each consecutive frame, with the first timestamp being the start of the downloaded footage.

3 Research

The research focuses on image vectorization techniques and anomaly detection methods based on Autoencoders and Generative Adversarial Networks.

3.1 Vectorization and PCA

To perform vectorization, we use the Img2Vec class from the img2vec_pytorch library. This tool allows us to extract feature vectors from images using pretrained convolutional neural networks. Instead of manually designing features, we reuse deep representations learned by models trained on large image datasets.

Different network architectures were tested during the experiments, such as ResNet152 or AlexNet, to analyze how the choice of feature extractor influences the quality of the representations and the anomaly detection results.

The extracted feature vectors are high-dimensional, which makes direct visualization difficult. To better understand the structure of the data, Principal Component Analysis is applied.

PCA reduces the dimensionality of the vectors while preserving the most important variance in the data. In this project, the vectors are reduced to two dimensions, which allows them to be visualized on a 2D scatter plot. This visualization helps to e.g., observe clustering of similar images.

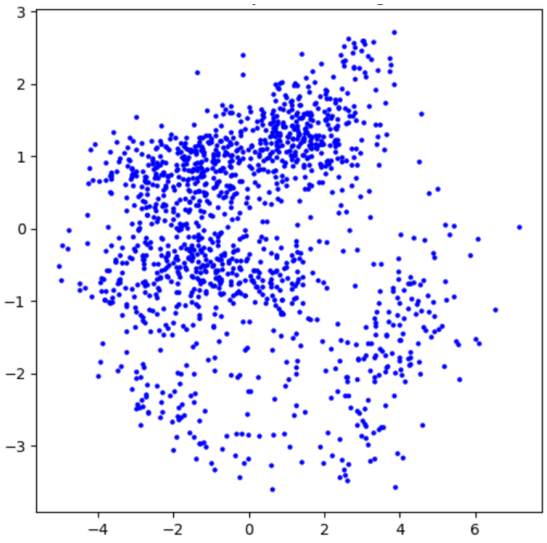

Figure 1: Resnet latent space

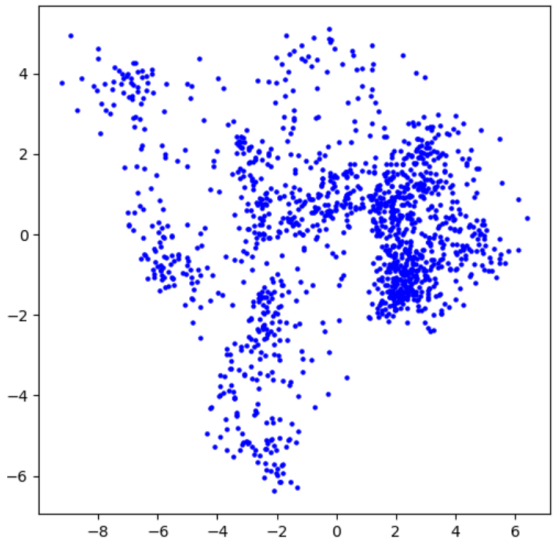

Figure 2: Densenet201 latent space

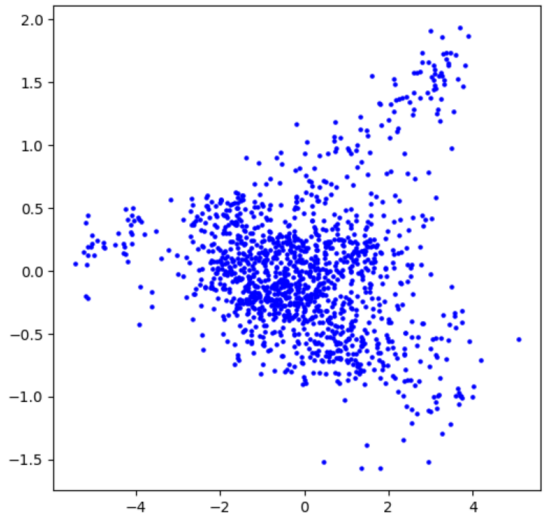

Figure 3: efficientnet_b7 latent space

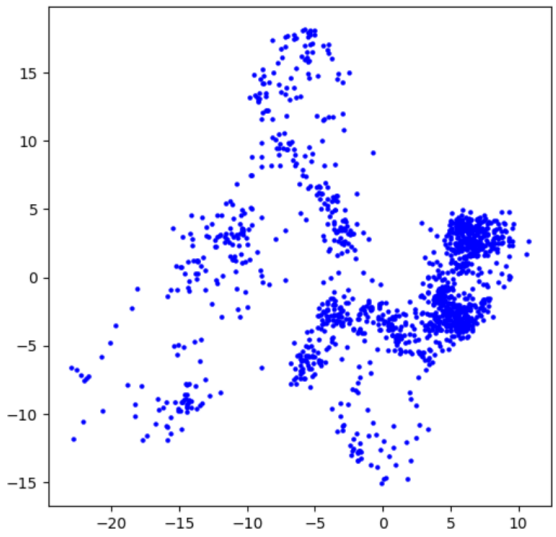

Figure 4: dino latent space

3.2 AE

First, image feature vectors are extracted using a pretrained AlexNet model.The dataset is transformed from raw images into numerical feature vectors.

A fully connected autoencoder is then defined to operate on the extracted feature vectors. The encoder gradually reduces the dimensionality input vector down to a 16-dimensional latent representation. This compact latent space captures the most important information from the original feature vectors. The decoder reconstructs the original 4096-dimensional vector from the latent representation.

The autoencoder is trained to minimize the reconstruction error between the input feature vectors and their reconstructed versions.

After training, reconstruction loss is calculated for each feature vector, and the distribution of these losses is visualized using a histogram. This visualization helps to understand the typical reconstruction error for normal data and provides a basis for defining an anomaly detection threshold.

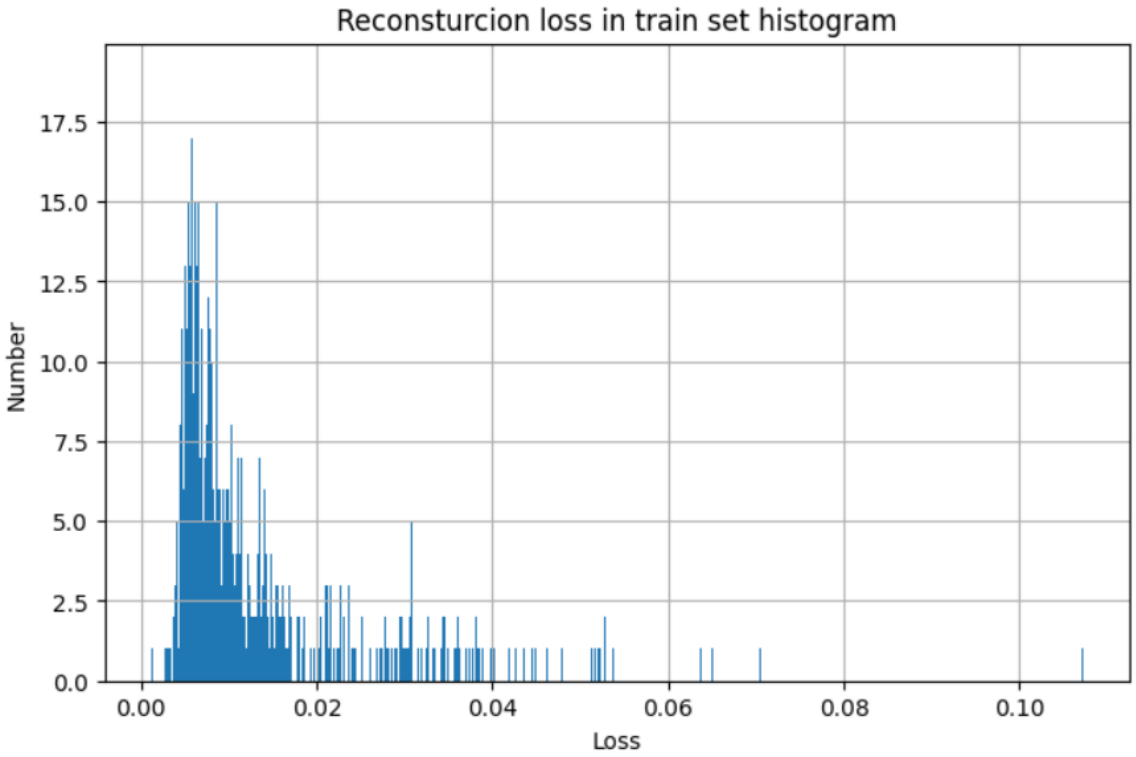

Figure 5: Reconstruction loss histogram

Then, the latent vectors obtained from the autoencoder are used for clustering. The DBSCAN clustering algorithm is applied to the latent space. DBSCAN groups samples based on their density and does not require specifying the number of clusters in advance.

After clustering, PCA is used to reduce the dimensionality of the latent vectors to two components.

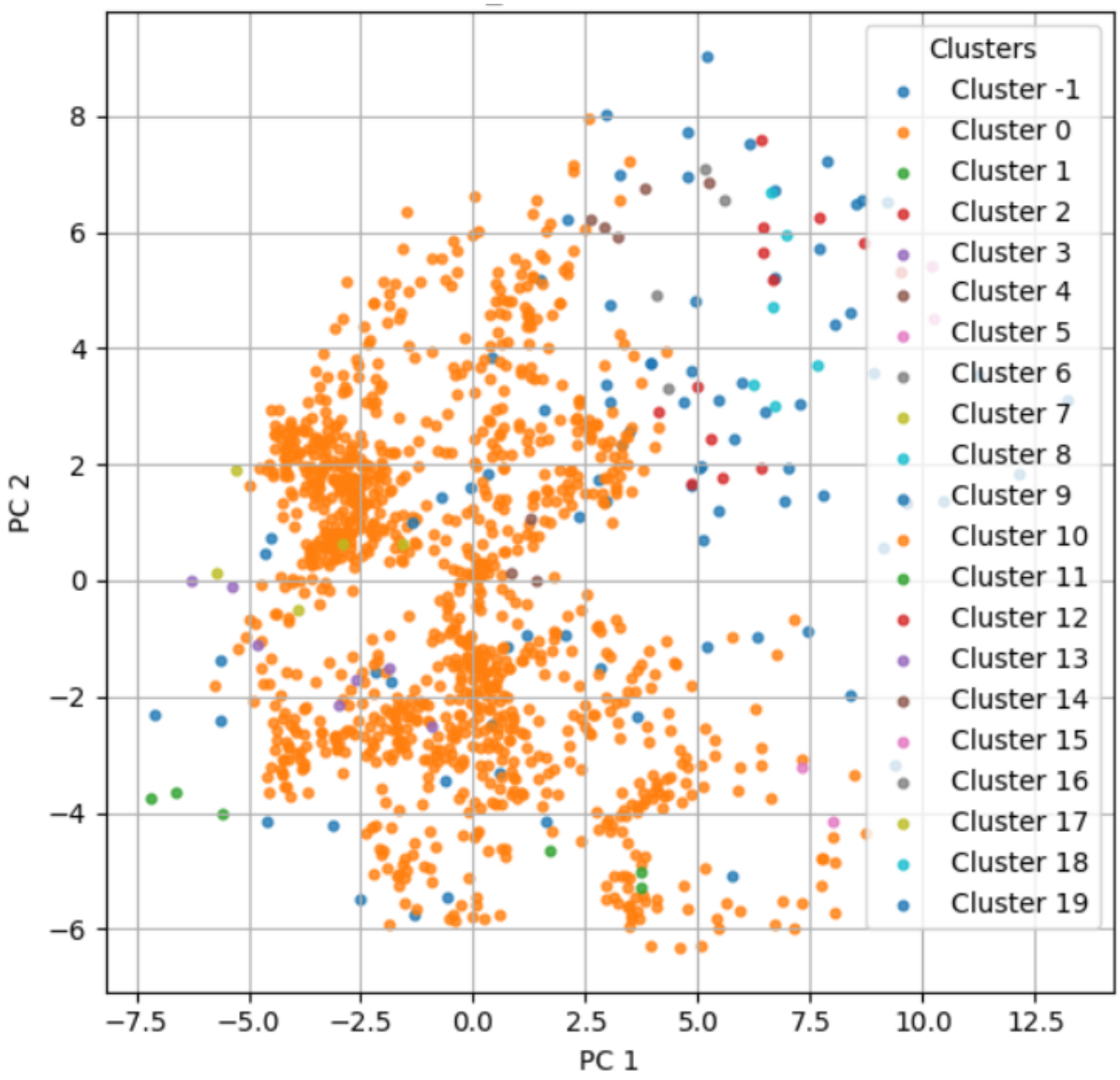

Figure 6: DBSCAN clustering

The results of clustering:

Figure 7: Cluster 0

We can see that the exemplary results on train dataset look good as the images of platform without the trains is in one class.

Figure 8: Clusters not 0

Other classes consist of the platform with trains as we can see some examples above.

The model performs well also on the test dataset which indicates correct training.

3.3 VAE

The Variational Autoencoder follows a scheme similar to the autoencoder described in the previous section. The encoder maps the input feature vectors to a probabilistic latent space by learning the mean and variance of a 16-dimensional latent distribution. After training, the latent vectors are used for clustering in the same way as for the autoencoder. DBSCAN is applied to the latent space, and PCA is used for visualization.

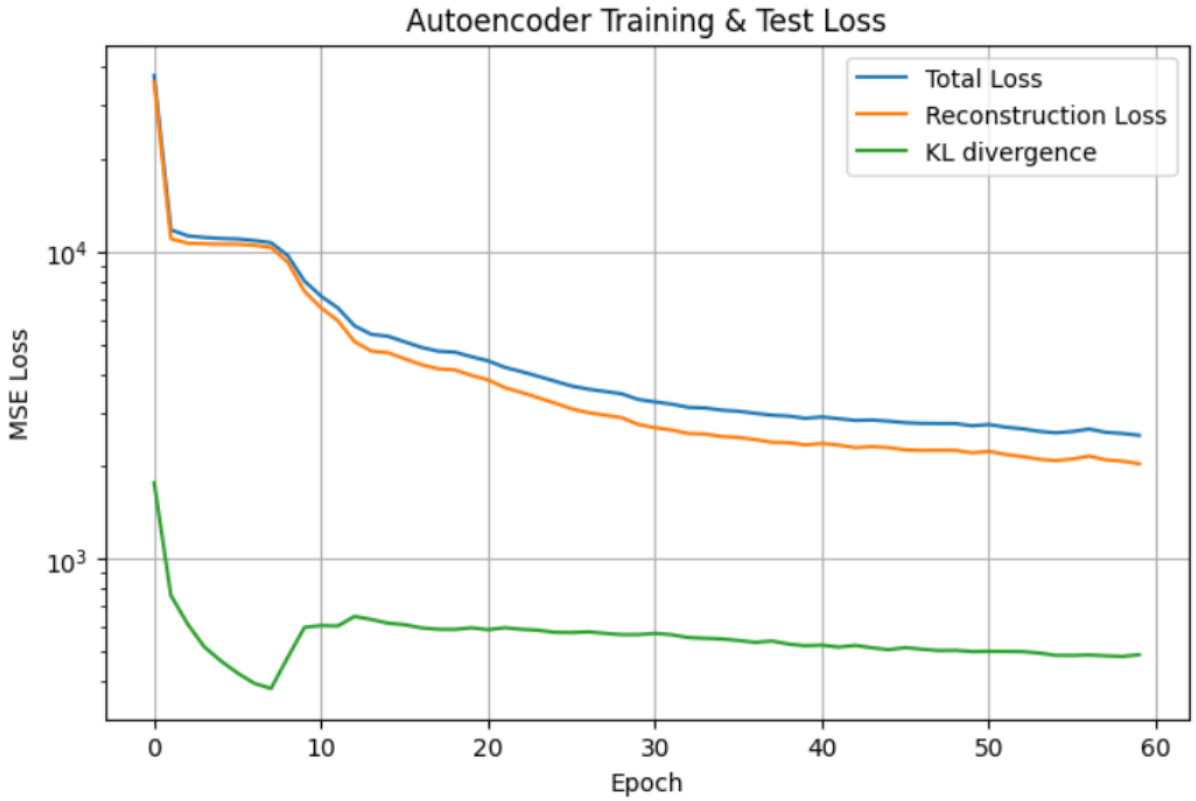

Figure 9: VAE losses and KLD

The clustering results are not very strongly separated; however, it can be observed that one class is concentrated closer to the center of the latent space, while the second class is distributed toward the outer regions.

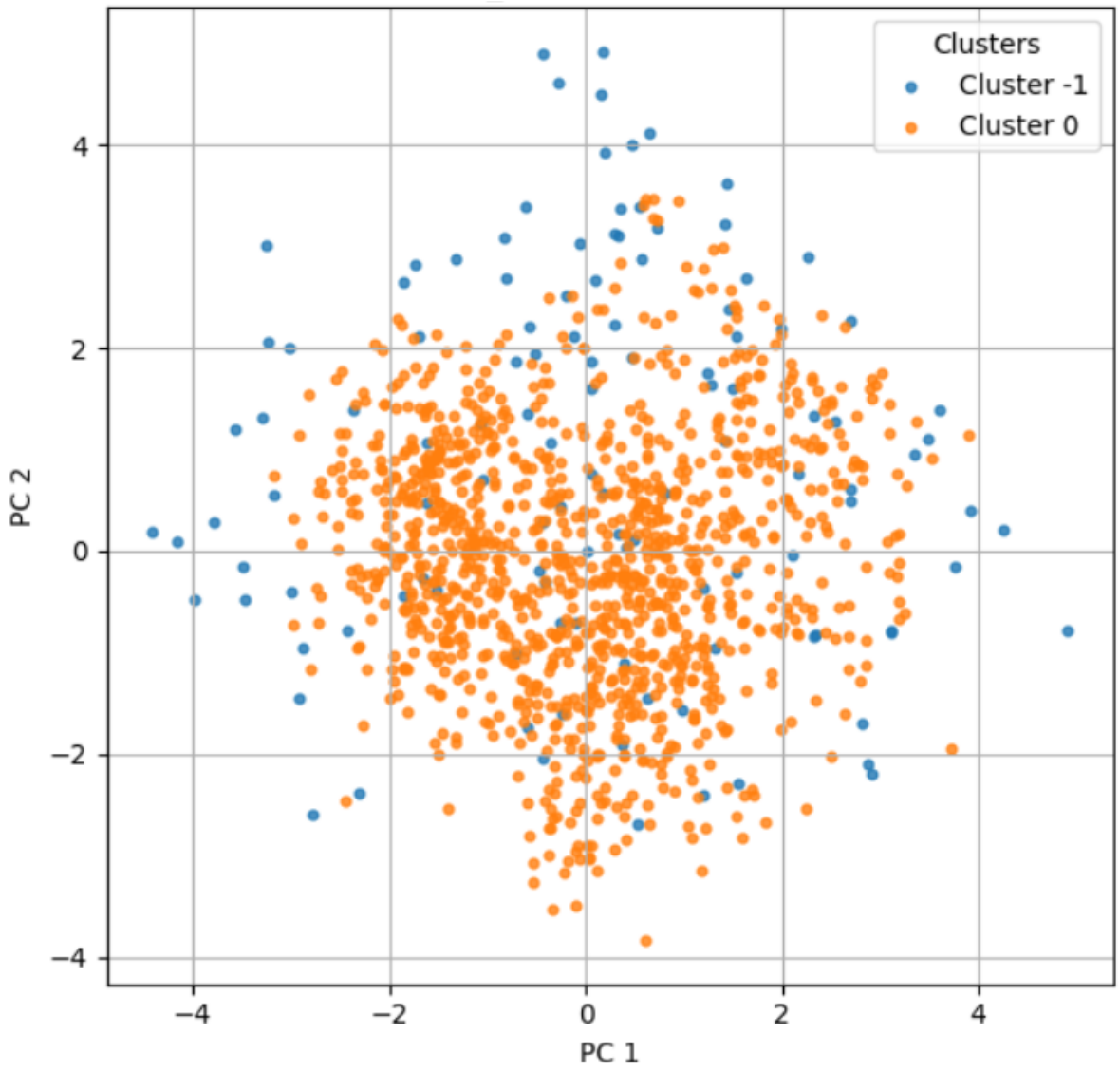

Figure 10: DBSCAN clustering for VAE

Here we can observe some cluster 0 representation (without train):

Figure 11: Cluster 0

Clusters other than 0 - training set:

Figure 12: Cluster not 0

3.4 GANs

To test another way of creating useful latent space we created a model based on Bidirectional Generative Adversarial Network. The model consists of 3 networks: encoder, decoder and discriminator. The encoder maps input images into latent space, while the decoder reconstructs imgaes from latent vectors using a deep convolutional upsampling architecture. The discriminator operates on (image, latent) pairs distinguishing between real and generated pairs.

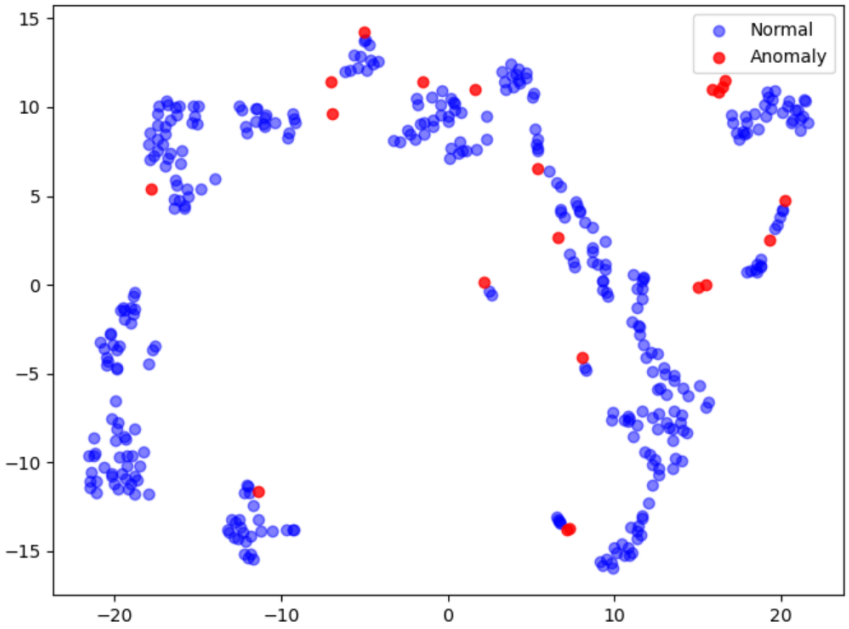

Figure 13: Custerization using DBSCAN

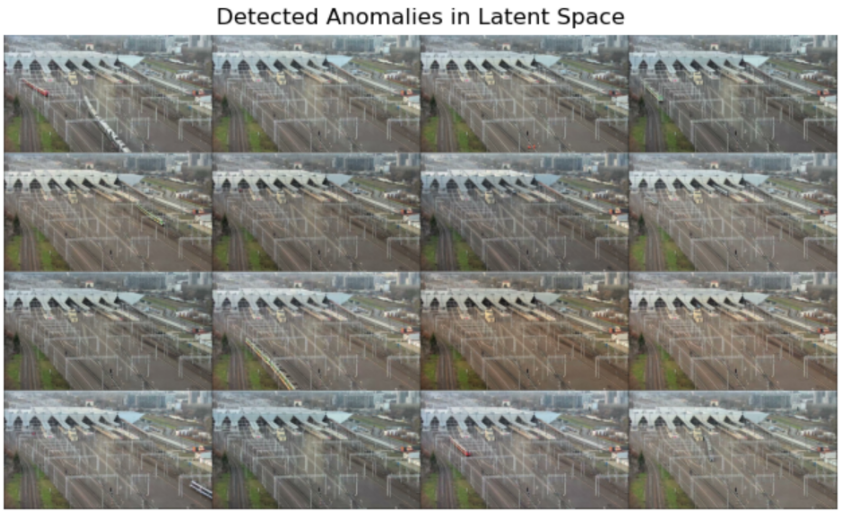

Figure 14: Photos classified as anomalies by DBSCAN in latent space

In practice, the latent scape learning by the encoder was not well-structured and clustering-based anomaly detection methods like DBSCAN struggled to reliably separate normal photos from anomalies.

In contrast, image-based anomaly detection performed significantly better. Especially when to the comparison between reconstructed images and original ones were added filters, which further amplified anomaly-related differences and minimized the impact of noise.

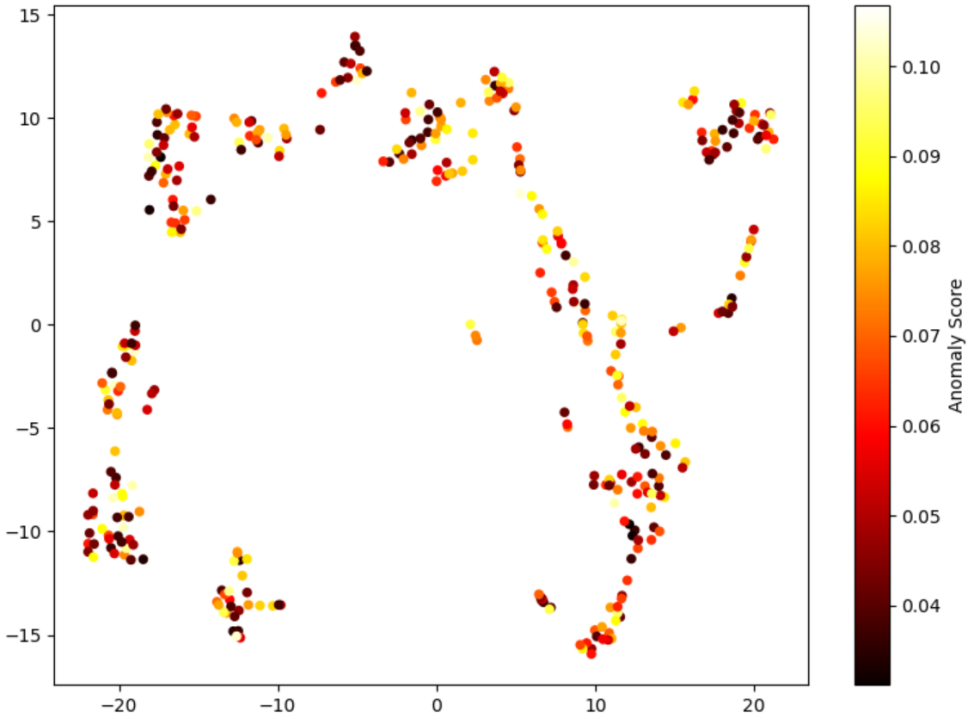

Figure 15: Visualization of reconstruction-based anomaly detection in latent space

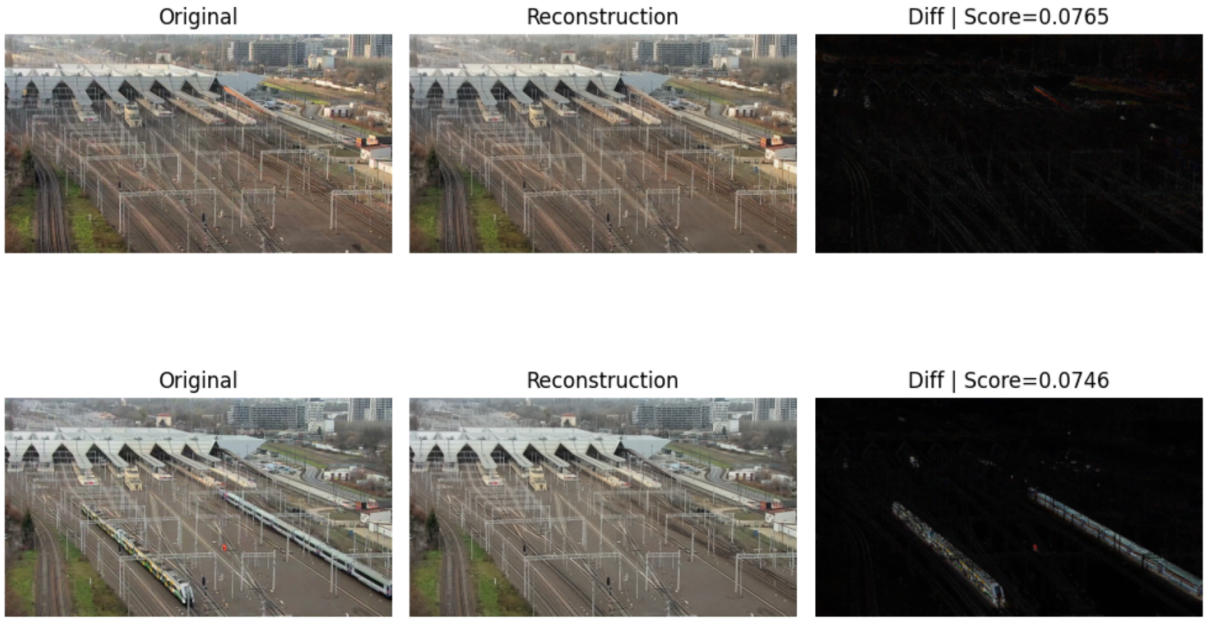

Figure 16: Results of reconstruction-based anomaly detection without filters

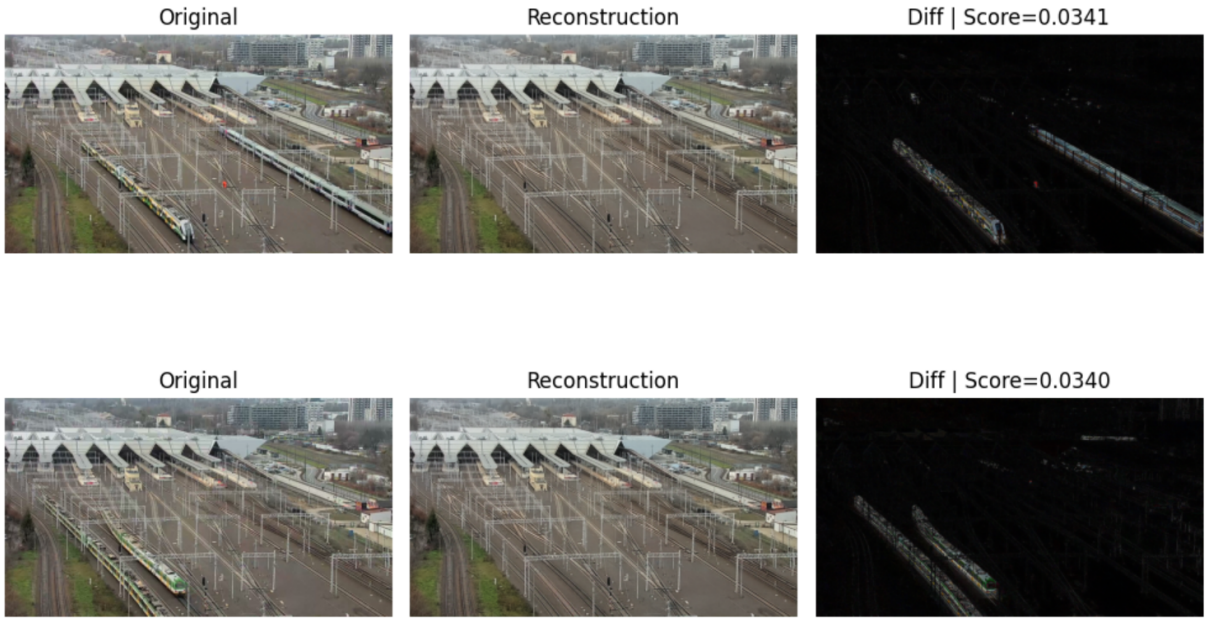

Figure 17: Results of reconstruction-based anomaly detection with filters

4 Comparison and summary

Both AE and VAE achieved effective dimensionality reduction, resulting in clearly observable structures in the latent space. In our case, quantitative benchmarking was not possible because the dataset lacked labels, and creating them manually would have been too time-consuming. The main difference between AE and VAE becomes evident when examining the distribution of points in the latent space. The standard AE produces multiple centers of distribution, leading to several distinct clusters identified by DBSCAN. In contrast, the VAE generates a smoother latent space with a single dominant center, making it easier to separate noise from the main cluster.

When comparing BiGAN based model to standard AE and VAE, both AE and VAE produced more separable and stable latent spaces, which also allowed better results in anomaly detection by using clusterization methods like DBSCAN. It may be partially attributed to the difference in input representation, AE and VAE were using vectors created via img2vec funtion rather than raw images. Also because BiGAN was operating on whole photos it had to be larger than basic AE.

Overall, while the BiGAN based model was less effective for latent-space anomaly detection, it turned out to be strong in reconstruction based anomaly detection at the image level.