Self-supervised Algorithms for Anomaly Detection on X-Rays

Marat Saibodalov [1,2], Iakov Karandashev [1,2]

[1] Peoples' Friendship University of Russia (RUDN University), 117198, Moscow, Russia
[2] Scientific Research Institute for System Analysis of RAS, 117218, Moscow, Russia

Abstract

In this paper, we consider the problem of prohibited objects detection on X-Ray images obtained by personal inspection scanners. Such scanners are often used on objects that require increased security control. The available data has a number of problems, which are described and addressed in the text. In this paper we consider only self-supervised anomaly detection algorithms. We are using several architectures of autoencoders and comparing them with the state-of-the-art algorithm Patch SVDD, which could be designed and trained on our data from scratch. Unlike supervised learning algorithms, which are often used for such problems, these models do not require a large amount of labeled data for training.

Keywords

Deep learning, image segmentation, anomaly detection, human X-Ray images.

1. Introduction

This work considers the problem of anomaly detection on X-Ray images obtained by personal inspection scanners. Personal inspection scanners (PIS) are often used on objects that require increased security control. They allow for a quick X-Ray image of a person, on which the PIS operator can see all objects on the person's body and visually confirm or deny the presence of any prohibited items. The process has a number of disadvantages associated with the human factor: quality analysis of the image requires significant time and attention, which leads to quick operator fatigue and can negatively affect the quality of image analysis. This process can be significantly automated, making it cheaper for the organization and more comfortable for the human operator. Deep neural networks were used to solve this problem. According to paperswithcode.com [1] the main state-of-the-art unsupervised approaches on MVTEC AD dataset [2] are Patch SVDD [3] and PatchCore [4]. We could not use the PatchCore since this model uses pre-trained ResNet [5] on ImageNet dataset [6], so we could not train it on our data. In this paper we consider only self-supervised anomaly detection algorithms. We are using several architectures of autoencoders [7] with SSIM (structural similarity index measure) [8, 9] loss function and comparing them with the state-of-the-art algorithm Patch SVDD, which could be designed and trained on our data from scratch. Unlike supervised learning algorithms, such models do not require a large amount of labeled data for training.

2. Formulation of the Problem

It is required to create an automated solution for the detection of prohibited objects on images obtained by personal inspection scanners. There are four datasets provided as input, obtained by different personal inspection scanners. Figure 1 shows an example of the original image and its corresponding mask.

Commonly these problems are solved by using supervised algorithms, where the model needs a huge amount of labeled data for training. Labeling thousands of images can require a financial expense and

GraphiCon 2023: 33rd International Conference on Computer Graphics and Vision, September 19-21, 2023
V.A. Trapeznikov Institute of Control Sciences of Russian Academy of Sciences, Moscow, Russia
EMAIL: maratsaibodalov@gmail.com (M. Saibodalov); karandashev@niisi.ras.ru (I. Karandashev)
ORCID: 0009-0003-0473-2161 (M.Saibodalov); 0000-0001-8483-072X (I. Karandashev)

labeling is also a long and exhausting process. Moreover, supervised methods require a lot of samples with exact anomalies, in our case, for example, guns, but there are not so many samples with such anomalies [10]. So this is the reason why this work considers only unsupervised and self-supervised algorithms for anomaly detection. These algorithms do not require labeled data, while learning they try to identify data features on their own. The main trick is to train them on a dataset that contains only normal data that doesn't contain anomalies. In case of the considered problem these models would be trained to identify only human body features and while testing pictures with anomalies would be abnormal for the model.

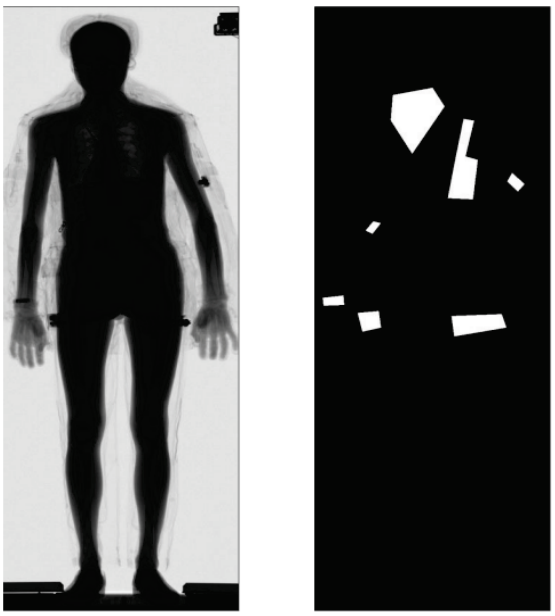

Figure 1: Example of the original image and the corresponding mask

3. Methods

3.1. AE with SSIM loss

The first considered method is an autoencoder. This model has been first introduced in [6] as a neural network that is trained to reconstruct its input. On figure 2 there is an architecture of an AE:

The network is trained to minimize the reconstruction error between the input and the output, which means that it learns to capture the essential features of the input data while discarding the noise and irrelevant details. In this implementation SSIM loss was used:

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (1)$$

where:

- μ_x - the pixel sample mean of x,
- μ_y - the pixel sample mean of y,
- σ_x^2 - the variance of x,
- σ_y^2 - the variance of y,
- σ_{xy} - the cross-correlation of x and y,
- $c_1 = (k_1 L)^2$, $c_2 = (k_2 L)^2$ - two variables to stabilize the division with weak denominator:
  - L - the dynamic range of the pixel-values (typically this is $2^{(\text{bits per pixel})} - 1$),
  - $k_1 = 0.01$, $k_2 = 0.03$ by default.

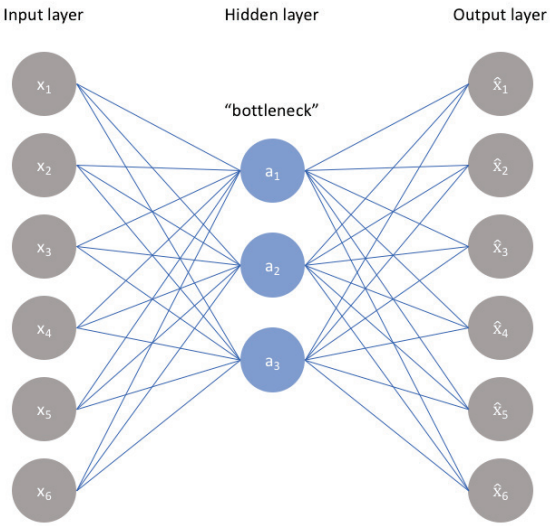

Figure 2: AE architecture

The model was trained exclusively on data without anomalies. During testing of the trained model, it is fed with images that contain anomalies. The idea is that the algorithm, trained to reconstruct images without anomalies (i.e., only the human body), will not be able to reconstruct the anomalies and, thus, an image similar to the input but without anomalies will be obtained. Then, an anomaly map can be obtained by subtracting the processed image from the original image. The results of the model's work can be seen in figures 3 and 4. It can be noticed that the model effectively highlights anomalies located outside the human body and even detects some anomalies on the body itself. However, a large number of anomalies were not detected, and there are also a significant number of false positives (evident in the algorithm highlighting parts of the floor or parts of the human body). This occurs because the data is too diverse: people are in different poses, and the images were taken with different scanners, causing variations in floor height across some images. The autoencoder processes the image quite quickly (around 3-4 seconds per image), but the results are far from ideal, as a significant number of anomalies were not detected. We will use the SSIM loss function later on.

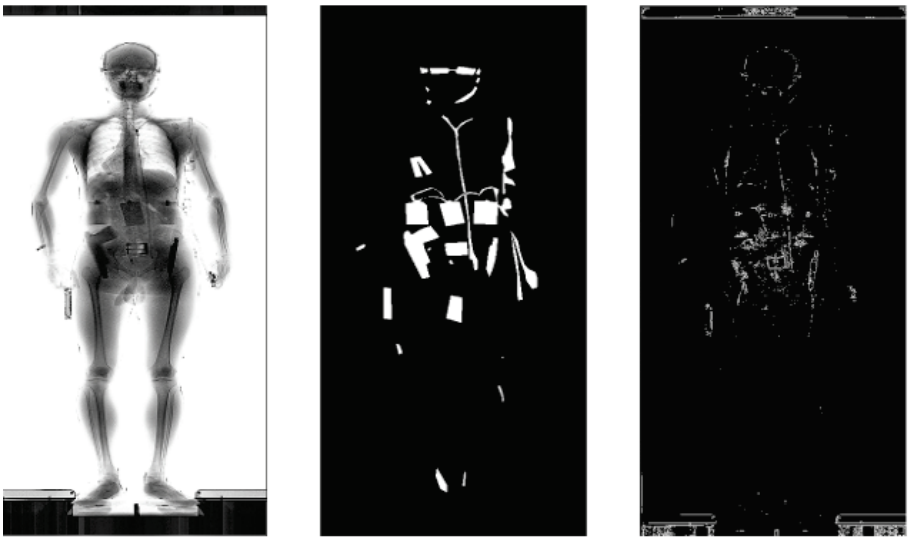

Figure 3: Result of the AE with SSIM loss. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

Figure 4: Result of the AE with SSIM loss. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

3.2. Patch VAE

The second considered method is a variational autoencoder [11] (figure 5). The difference between AE and VAE is that in VAEs the latent space is modeled as a Gaussian distribution with a mean and standard deviation. During training the network is optimized to learn the mean and standard deviation of this distribution which helps reconstruct images better in terms of color intensity. Furthermore, we will use small images (patches) in this approach since we assume that neural networks perform better on small images [3, 4, 12, 13].

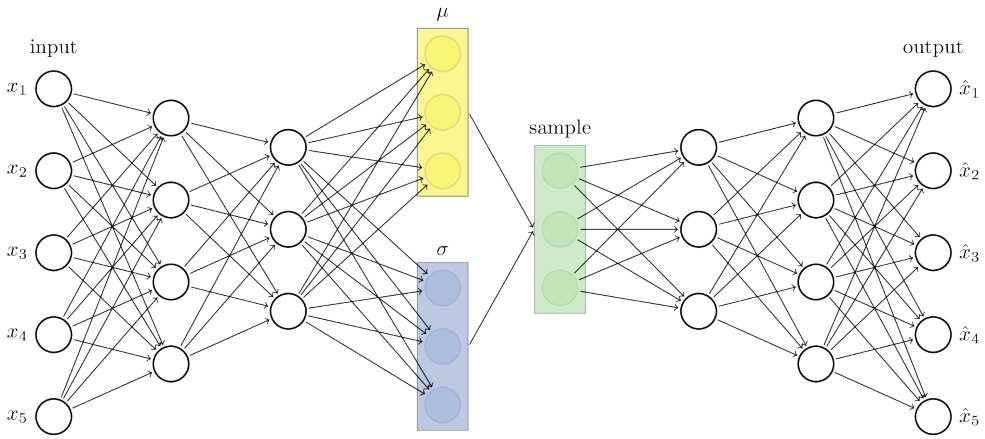

Figure 5: VAE architecture

A new dataset of patches was prepared and we trained the model on this data (figure 6). While testing we transfer the image to the neural network, it crops the image into patches, processes them individually and then recreates the image from patches. After that we calculate the difference between the original image and our neural networks output and obtain an anomaly map.

Figure 6: Example of the patches

As can be seen in figure 7 and figure 8, the result of the algorithm is significantly better than AE results. This algorithm finds more anomalies, but there is still a problem with anomaly detection on the body. Anomaly maps were obtained in the same way as with AE, i.e. the reconstruction by the model was subtracted from the similar image. It is worth mentioning that the use of the approach of dividing the original image into patches significantly reduced the number of false positive detections compared to the standard autoencoder. The processing time for one image does not actually differ from the autoencoder and is 3.5 seconds.

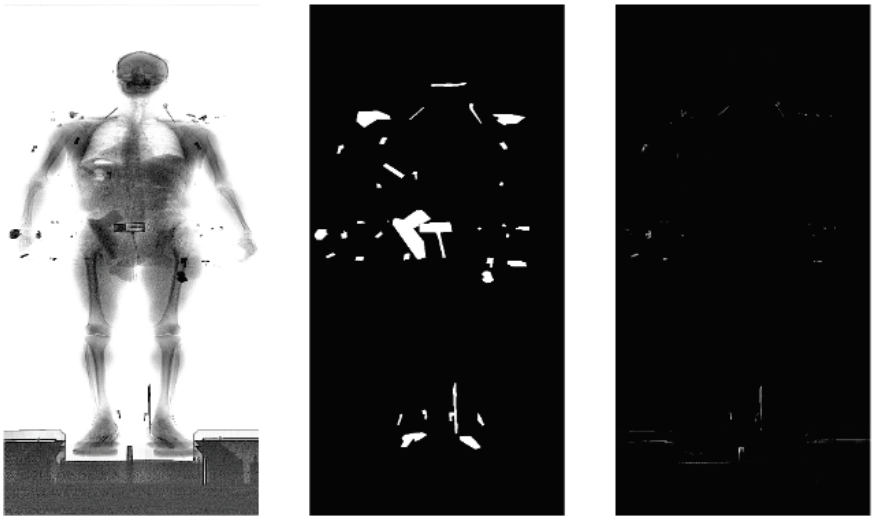

Figure 7: Result of the Patch VAE. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

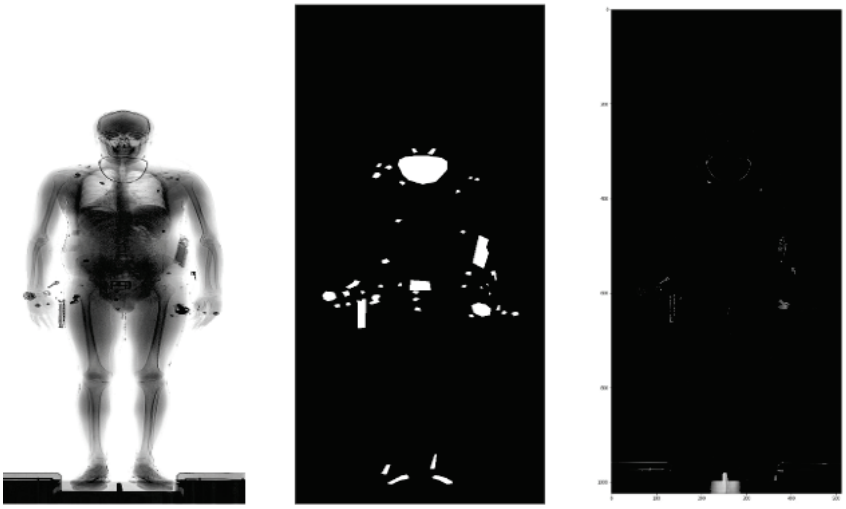

Figure 8: Result of the Patch VAE. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

3.3. Patch SVDD

The model was presented in the article Patch SVDD: Patch-level SVDD for Anomaly Detection and Segmentation [3] and was trained on MVTEC AD dataset [2].

Patch SVDD (Support Vector Data Description) is a deep learning model that combines the concept of patch decomposition and the SVDD algorithm for anomaly detection. The SVDD algorithm is a one-class classification algorithm that aims to learn a tight boundary around a set of normal data, and it is commonly used for anomaly detection.

Patch SVDD works by first decomposing the input data into a set of overlapping patches of fixed size. Each patch is then passed through a patch encoder, which maps the patch to a lower-dimensional latent space representation. The patch encoders are trained using a reconstruction loss, which encourages the network to learn a compressed and informative representation of the input patches.

The compressed patch representations are then used to train a support vector data description (SVDD) model [14]. The SVDD model learns a tight hypersphere boundary around the normal data in the latent space, and the objective is to minimize the distance between the center of the hypersphere and the normal data points while maximizing the distance between the center and any potential anomaly data points.

During testing, the input data is decomposed into patches and passed through the patch encoder to obtain the latent space representations. The latent space representations are then passed through the SVDD model, and the distance between the data point and the center of the hypersphere is calculated. If the distance is larger than a predefined threshold, the data point is flagged as an anomaly. On figure 9 Patch SVDD architecture is shown:

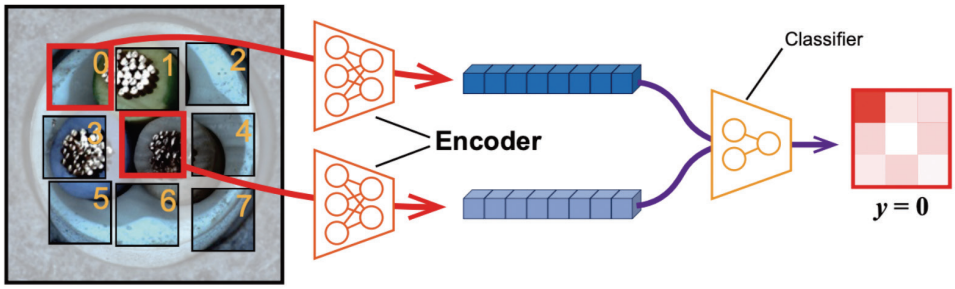

Figure 9: Patch SVDD Architecture. The image was taken from the original paper [3]

This model was reimplemented for our dataset, the results are more than satisfying (figure 10, figure 11), but the processing time of one picture is longer than testing time of other models (about 12s). Also, training time of this model is significantly higher than training time of AE and Patch VAE (table 1).

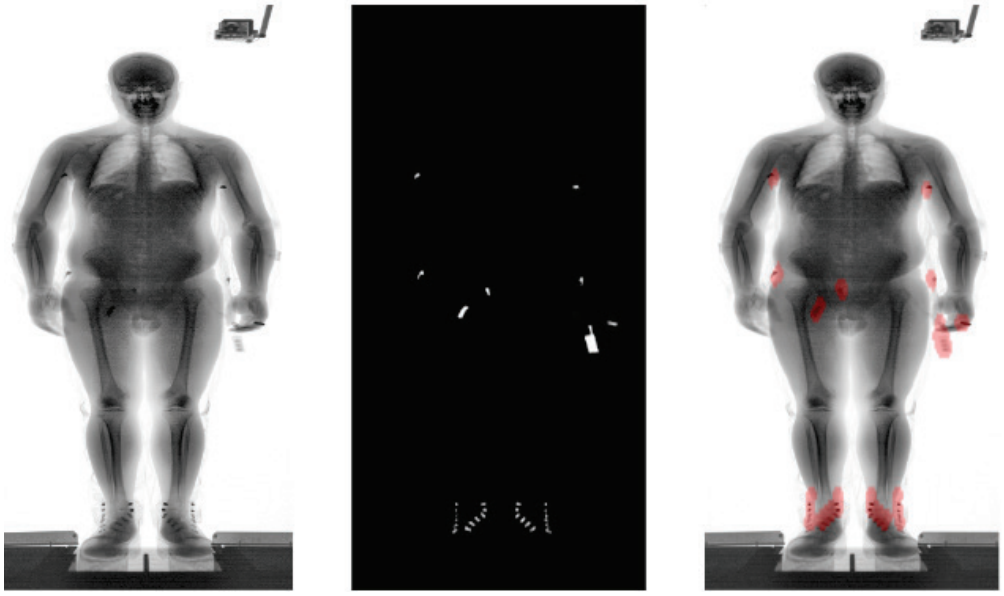

Figure 10: Patch SVDD result. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

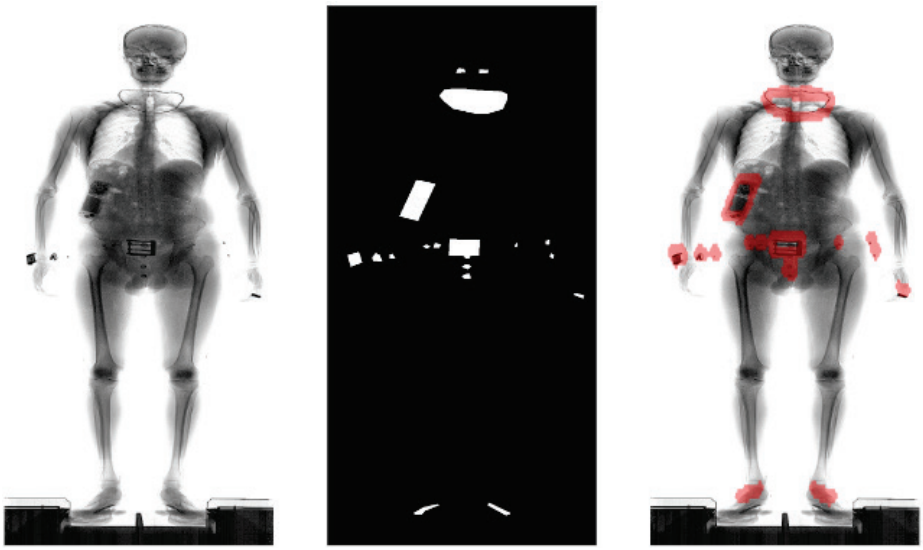

Figure 11: Patch SVDD result. The image on the left hand side is the input image, the image in the middle is the corresponding mask (ground truth), and on the right hand side is the anomaly map obtained through the model

4. Results and Conclusion

Table 1 shows that the Patch SVDD algorithm is better in terms of solving the problem of anomaly detection in comparison with other algorithms. However, the time for its training and processing of one image is significantly higher than competitors. The tests were carried out on 50 images from the test samples, where an anomaly map was compared with ground truth. The IoU (Intersection over union) metric is calculated by dividing the overlap between the predicted and ground truth annotation by the union of these:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|} \tag{2}$$

Table 1: Comparison of AE with SSIM (structure similarity) loss, Patch VAE and Patch SVDD

	AE SSIM	Patch VAE	Patch SVDD
Training time, 100 epochs	1h	1h 45m	9h 23m
One image processing time, sec	4s	5s	15s
SSIM	0.69	0.76	0.94
IoU	0.71	0.74	0.92

We have demonstrated that unsupervised and self-supervised models could handle the task of anomaly detection even on such complex data like human body X-Rays. The key is to pick up the correct architecture, number of layers and correctly preprocess data. It may take a time, but such models are much more lightweight in terms of hardware consumption, and also such models do not require labeled data for training, unlike supervised learning algorithms.

5. Acknowledgements

The work is financially supported by State Program of Federal Research Center “Scientific research institute for system analysis of the Russian Academy of Sciences” No. FNEF-2022-0003.

6. References

[1] URL: https://paperswithcode.com/sota/anomaly-detection-on-mvtec-ad.

[2] P. Bergmann, M. Fauser, D. Sattlegger and C. Steger, MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*)*, Long Beach, CA, USA, 2019, pp. 9584-9592, doi: 10.1109/CVPR.2019.00982.

[3] J. Yi and S. Yoon. Patch svdd: Patch-level svdd for anomaly detection and segmentation, in Proceedings of the Asian Conference on Computer Vision, 2020, doi:10.1002/jnm.3134.

[4] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox and P. Gehler, Towards Total Recall in Industrial Anomaly Detection, 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,* New Orleans, LA, USA, 2022, pp. 14298-14308, doi: 10.1109/CVPR52688.2022.01392.

[5] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*,* Las Vegas, NV, USA, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[6] J. Deng, W. Dong, R. Socher, L. -J. Li, Kai Li and Li Fei-Fei, ImageNet: A large-scale hierarchical image database, 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 2009, pp. 248-255, doi: 10.1109/CVPR.2009.5206848.

[7] Rumelhart, D.E., Hinton, G.E., Williams, R.J. Parallel distributed processing: Explorations in the microstructure of cognition, vol. 1. chap. Learning Internal Representations by Error Propagation, pp. 318–362. MIT Press, Cambridge, MA, USA (1986).

[8] Zhou Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, April 2004, doi: 10.1109/TIP.2003.819861.

[9] Bergmann, Paul, Löwe, Sindy, Fauser, Michael, Sattlegger, David and Steger, Carsten Improving Unsupervised Defect Segmentation by Applying Structural Similarity to Autoencoders, 2018. , doi: 10.5220/0007364503720380.

[10] A.S. Markov, E.Yu. Kotlyarov, N.P. Anosova, V. A. Popov, Ya.M. Karandashev, and D.E. Apushkinskaya. Using Neural Networks to Detect Anomalies in X-Ray Images Obtained with Full-Body Scanners. Automation and Remote Control, 2022, Vol. 83, No. 10, pp. 1507–1516, doi:10.1134/s00051179220100034.

[11] Kingma, D. P. & Welling, M. (2014). Auto-Encoding Variational Bayes. 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, Conference Track Proceedings, 2014.

[12] Yuchen Lu, Peng Xu. Anomaly detection for skin disease images using variational autoencoder. arXiv, 2018.

[13] A. Krizhevsky, G. Hinton. Learning multiple layers of features from tiny images, 2009.

[14] Tax, D.M., Duin, R.P. Support Vector Data Description. Machine Learning 54, 45–66 (2004), doi: 10.1023/B:MACH.0000008084.60811.49