

Unsupervised Anomaly Detection in Industrial Image Data with Autoencoders

Tulsi Kumar¹, Gautam Malik¹ and Adriano Puglisi¹

¹Department of Computer, Control and Management Engineering, Sapienza University of Rome. Via Ariosto 25, Roma, 00185, Italy

Abstract

Traditional quality control techniques could miss small defects in manufacturing environments, reducing the quality of the final product. Using the MVTec dataset, a commonly used benchmark in industrial visual inspection, in this study we investigate two types of autoencoders, denoising autoencoders (DAE) and contractive autoencoders (CAE), to solve the problem of defect identification in industrial processes. The presence of both textured and non-textured objects allows a direct comparison between materials with different surface characteristics. The VGG16 and ResNet models pre-trained on ImageNet are used as encoders. Three variants of DAE and three of CAE are designed and evaluated. Both the loss MSE (Mean Squared Error) and the SSIM (Structural Similarity Index Measure) are used to compare the reconstruction quality and the defect detection capability. The results highlight performance differences between DAE and CAE and between different object categories, providing useful insights into the effectiveness of each approach in different industrial scenarios.

Keywords

Unsupervised Learning, Denoising Autoencoder, Contractive Autoencoder, Anomaly Detection

1. Introduction

Quality control is an essential part of many manufacturing industries. Usually, it is performed manually, but the problem with manual visual inspection is that there are possibilities for error and for this reason, vision-based inspection can be used. The deep neural network has played an important role in the automation industry. Using deep neural networks, visual inspection can also be automated. Many image processing and machine learning methods have already been used to achieve automated defect detection in production parts. However, image processing methods have limitations, as implicit engineering features are used for the application, which can be misleading for complex cases. Deep convolutional networks are a solution for automating quality control in the manufacturing industry since they have the ability to obtain the best features from images, but these methods are limited by data availability. There are two problems to consider: one is the imbalance of data between normal and defective images. The other is the annotation of the data. To overcome this problem, defect detection is treated as an anomaly detection problem.

Due to the absence of labels in the data, the problem can be addressed through unsupervised learning, by training convolutional networks on normal images and testing them on images containing defects [1, 2, 3, 4, 5]. In this work, convolutional neural network autoencoders [6, 7] are used to perform anomaly detection, in particular, the study focuses on the use of denoising autoencoders and

contractive autoencoders to improve the effectiveness of defect detection.

The autoencoder encodes the input into a lower-dimensional representation known as the latent space, from which the decoder reconstructs the output. A modification to the autoencoder called a denoising autoencoder stops the network from learning the identity function. To be more precise, if the autoencoder is too large, it can just learn the data, resulting in output equivalent to input without doing any beneficial representation learning or dimensionality reduction. Denoising autoencoders address this issue by purposefully introducing errors, noise, or masking some input values. [8, 9]

A contractive autoencoder is an unsupervised deep learning method that aids a neural network in encoding unlabeled training input. In general, autoencoders are employed to discover a representation, or encoding, for a set of unlabeled data, typically as the initial step toward dimensionality reduction or the creation of new data models. The traditional reconstruction cost function is enhanced by a penalty term in a contractive autoencoder. The Frobenius norm of the Jacobian matrix representing the activations of the encoder with respect to the input corresponds to this penalty term. This penalty term causes a localized space contraction, which in turn produces strong characteristics on the activation layer. The penalty aids in sculpting a representation that is more invariant to most directions orthogonal to the manifold while also better capturing the local directions of variation required by the data, which correspond to a lower-dimensional non-linear manifold [10].

 puglisi@diag.uniroma1.it (A. Puglisi)

 0009-0007-6307-7194 (A. Puglisi)

 © 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

2. Related Works

For the anomaly detection SIFT and SURF are used to extract the features from the images and train the model on normal image. Features of images can be misleading sometimes depending upon nature of the application. Machine learning algorithms can be used to classify anomalies from the normal images. A supervised learning approach is not good for this application, but semi-supervised and unsupervised models increase performance of model. The supervised and semi-supervised approaches are compared and the performance of the semi-supervised model is better than the supervised approach [11]. In this paper [1] they propose a model based on point features of the images. A hand-crafted point feature Harris-Laplace point detector is used in this study to detect the anomalies. The point feature uses Harris corner detector and then SIFT key points to extract the local shape around key points. Different loss functions are used for the unsupervised deep learning model. This study shows that for different types of objects the model performs differently and a specific type of loss is suitable for a specific application. To identify and categorize defects of the LED chip, Lin et al. [12] suggested the LEDNet network. Cha et al. [13] suggested utilizing Faster R-CNN, that showed promising results also in other applications [14], to identify five distinct flaws to detect structural damage.

The use of autoencoders for unsupervised anomaly identification based on reconstruction loss is examined in [15], highlighting both its strengths and weaknesses. Using an analogous situation from particle physics, it demonstrates that the standard autoencoder configuration is not a model independent anomaly tagger. In the work of Lupo et al. [16] generative models are used to detect anomalies in texts, exploring approaches ranging from machine learning to deep learning. Among the analyzed models, the variational autoencoder is the one with the most promising performances for this task. Vincent et al. [17] instead studied denoising autoencoders for the extraction of robust features from images, demonstrating that these models are able to improve the representation of visual features and, consequently, the overall image quality. Similar architectures have been employed also in the field of audio processing, for example for the automatic identification of speech disorders, exploiting unlabeled speech signals [18]. Bionda et al. [19] proposes a deep convolutional autoencoder to detect the anomalies in the textured images. MSE, or pixel wise error is not suitable for textured images as it only focuses on pixels. So, SSIM is used as a loss function to improve performance of the autoencoder. Complex wavelet SSIM performs better than MSE for the textured images. Based on the application loss function plays a great role in generative models [20]. In this paper [21] they present a novel approach

called PNI that, given neighborhood characteristics and a multi-layer perceptron network model, computes the normal distribution using conditional probability. Additionally, a histogram of typical characteristics is made for each point to use position information. The suggested technique uses an extra refining network trained on fabricated anomaly pictures in addition to the anomaly map to better interpolate and account for the shape and edge of the input image. Yang et al. [22] presents a novel method for detecting industrial image anomalies based on a self-supervised learning and self-attentive graph convolution (SLSG) network. In SLSG, pseudo-prior knowledge of anomalies is introduced by simulated abnormal samples, and the encoder is assisted in learning the embedding of normal patterns and position connections. Holly et al. [23] suggests a technique that makes use of a total reconstruction error and an autoencoder to locate system problems. In order to pinpoint the source of a problem, the signals that contribute the most to the overall reconstruction error are identified by computing the individual reconstruction error for each sensor signal.

3. Methodology

3.1. Dataset

The dataset used is the MVtec industrial images dataset. It contains images of many different industrial products. It is an industrial inspection-focused dataset for evaluating anomaly detection techniques. Over 5000 high-resolution images in fifteen different object and texture categories make up this collection. Each category includes both a test set of photos with various types of faults and images without defects as well as a set of training images with no defects [24].

3.2. Preprocessing

During the training phase, only normal images, free of defects or anomalies, are used. Images containing defects of various types, specific to each product, are instead used in the testing phase. All images provided are high-resolution RGB images with dimensions of 1024×1024 . To reduce computational complexity, they are downsampled to 256×256 and normalized by dividing the pixel values by 255.

When using a denoising autoencoder, noise is introduced into the data to artificially corrupt it. Similarly, the test images are also altered and the result of the model is compared with the original uncorrupted images. An example of the images with noise is shown in Figure 1. For the contractive autoencoder, resized but uncorrupted images are directly fed as input and the model is trained to faithfully reproduce the same images as output.

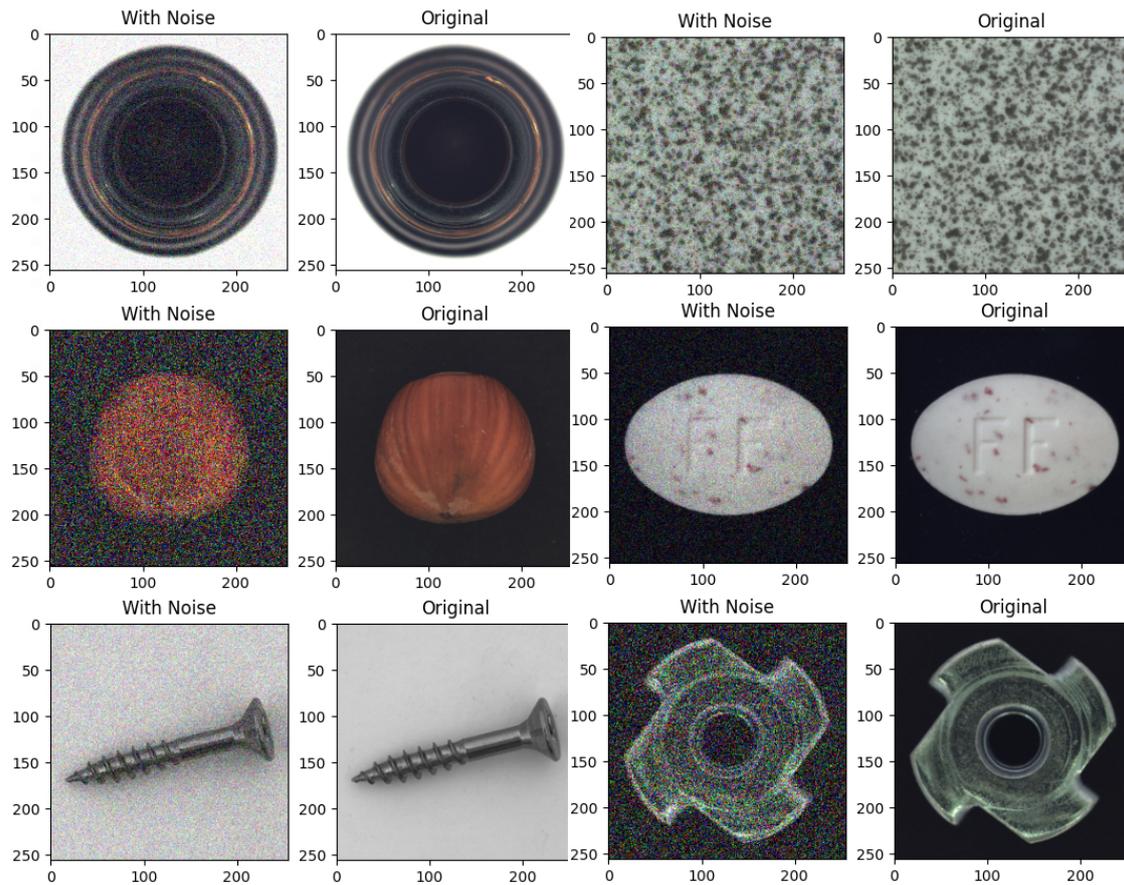


Figure 1: Preprocessed Images after adding noise

3.3. Model Architecture

The autoencoder model adopted in this work is based on a structure composed entirely of convolutional layers. Its architecture can be conceptually divided into three fundamental components, namely the encoder, the decoder and the latent space, often referred to as bottleneck. The encoder has the task of compressing the data, reducing the dimensionality of the input image until obtaining a compact representation in the latent space. This encoded representation is then transmitted to the decoder, which has the role of expanding the data again to reconstruct an image as similar as possible to the initial one.

Four convolutional layers and four max-pooling layers form the encoder, which gradually reduces the number of pixels in the image. The type of autoencoder used determines the structure of the bottleneck. The latent space in the denoising autoencoder is composed directly of the final output of the encoder. Instead, the representation passes through a thick layer that acts as a bottleneck

in the contractive autoencoder. Both architectures use the same decoder, consisting of four upsampling layers and five convolutional layers. To restore the output to its initial size, a final convolutional layer is added. The complete architectures of the two models are reported in Figure 2 and Figure 3.

To compare our model we also took into account pre-trained encoders, in particular, the VGG16 model already trained on the ImageNet dataset. This model includes sixteen layers in total, thirteen of which are convolutional and three fully connected. In our implementation, only the convolutional layers were kept, while the fully connected ones were removed [25]. The ResNet50 model pre-trained on the ImageNet dataset was also used. Although the encoder architectures are different, in both cases the decoder was kept unchanged, so as to make the comparison between the configurations more fair and meaningful.

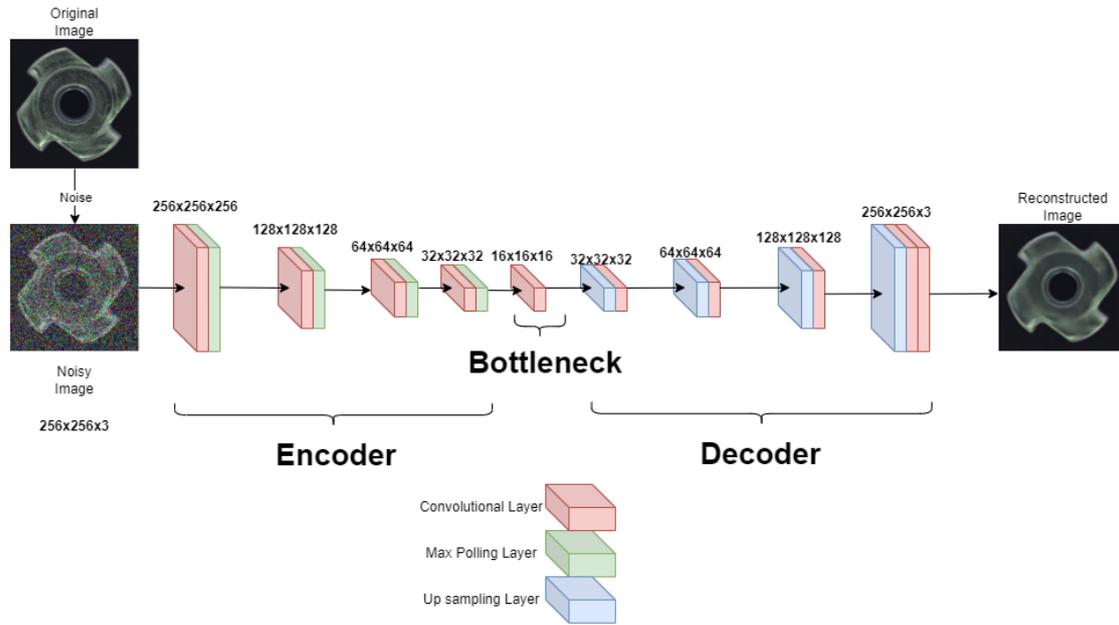


Figure 2: Denoising autoencoder architecture

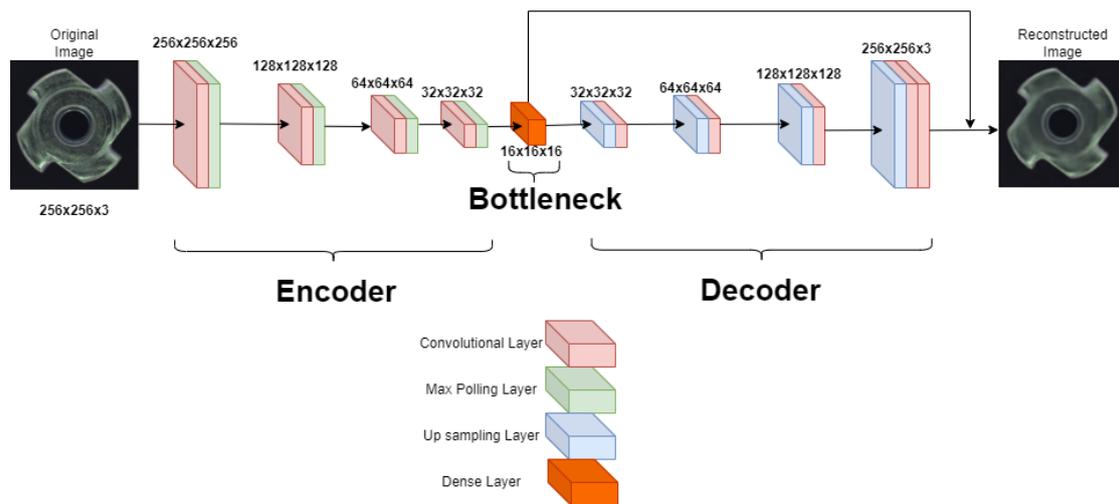


Figure 3: Contractive autoencoder architecture

3.3.1. Loss Function

For training and testing losses, denoising autoencoder uses the MSE. On measuring pixel values between two images, Mean squared error computes the average of the squared differences between corresponding pixel values. In the case of a contractive autoencoder, the bottleneck or the latent dimension of the autoencoder is used to compute the contractive loss along with MSE. Finally,

the sum of the two losses is calculated. The autoencoder is trained using contractive loss.

$$MSE(I, K) = \frac{1}{N} \sum_{i=1}^m \sum_{j=1}^n [I(i, j) - K(i, j)]^2 \quad (1)$$

Another loss function used for the texture image is the measure of resemblance between two pictures. SSIM compares the brightness, contrast, and structural elements of

Table 1

Performance results of denoising autoencoder on MVTec-AD dataset. The ratio of corrected classified normal samples (top row) and ratio of corrected classified abnormal samples (bottom row) are given. The best results are in bold.

Category	DAE(MSE)	DAE(SSIM)	DAE-VGG16(MSE)	DAE-VGG16(SSIM)	DAE-ResNet(MSE)	DAE-ResNet(SSIM)
Hazelnut	0.73	0.50	0.05	0.10	0.75	0.88
	0.86	0.88	1.00	0.98	0.70	0.66
Overall	0.81	0.76	0.64	0.66	0.72	0.74
Bottle	0.35	0.3	0.25	0.1	0.75	0.40
	0.97	0.92	1	0.98	0.33	0.56
Overall	0.82	0.77	0.81	0.77	0.43	0.53
Pill	0.42	0.53	0.38	0.35	0.48	0.35
	0.88	0.84	0.84	0.94	0.47	0.59
Overall	0.81	0.80	0.78	0.84	0.48	0.55
Screw	0.34	0.49	0.10	0.05	0	0.08
	0.83	0.79	0.92	1	0.31	0.44
Overall	0.71	0.72	0.68	0.76	0.23	0.35
Tile	0.94	0.43	0.10	0	0.46	0.27
	0.11	0.81	0.98	1	0.38	0.83
Overall	0.34	0.67	0.72	0.72	0.41	0.68

two images to determine how similar they are. It makes use of statistical metrics including pixel intensity mean, variance, and covariance. c_1 and c_2 are constants that are added to prevent denominator instability [26]. It is often adopted as a loss function for picture-based optimization tasks, for example, image denoising or super-resolution, and as a quality metric for image compression or restoration.

$$SSIM(I, K) = \frac{(2\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)} \quad (2)$$

3.3.2. Training and Optimization

To train the denoising autoencoder, input images are artificially corrupted with noise, while the corresponding clean images are used as targets. The model is optimized using Adam optimizer, with a learning rate of 0.0001 and a batch size of 32. The contractive autoencoder is trained with Adam as well, but its loss function includes a regularization term on the encoder’s Jacobian in addition to the reconstruction loss. Both loss functions are applied to each sample type, in order to analyze the model performance in different scenarios.

As for the pretrained encoders, the VGG16 and ResNet50 architectures have been considered, both optimized with the same decoder used in the other models. In the case of VGG16, the last three fully connected layers are removed, keeping the five convolutional blocks that are subsequently trained together with the decoder. After the encoder, a dense layer is inserted to act as a bottleneck. As for ResNet50, the last layer is removed and all the other layers are fine-tuned with the decoder.

In both cases, training is aimed at minimizing the loss function.

3.3.3. Thresholding and classification

For anomaly detection using autoencoders, a threshold based on the reconstruction error is adopted to distinguish between normal images and images containing defects. The error is calculated by comparing each reconstructed image with the corresponding original image, using MSE or SSIM depending on the type of sample.

The threshold is determined starting from the training images, which are all free of anomalies. For each of them, the reconstruction error is calculated, after which the threshold is obtained as the average of the errors obtained. Once established, this threshold allows to classify the test images, those with an error lower than the threshold are considered normal, while those with an error higher than the threshold are classified as anomalous.

To evaluate the effectiveness of the classification process, the accuracy and the F1 score are used. Even if in the test dataset there are different types of anomalies, in this study they are treated as belonging to a single class.

4. Results and Analysis

The model was evaluated on the hazelnut, pill, bottle, screw and tile categories of the MVTec dataset. The pill and tile classes represent textured samples, while the others have smooth surfaces. Both autoencoders, DAE and CAE, were trained using the two loss functions, MSE and SSIM, and tested on all categories. The results for DAE are reported in Table 1.

Table 2

Performance results of contractive autoencoder on MVTEC-AD dataset. The ratio of corrected classified normal samples (top row) and ratio of corrected classified abnormal samples (bottom row) are given. The best results are in bold.

Category	CAE(MSE)	CAE(SSIM)	CAE-VGG16(MSE)	CAE-VGG16(SSIM)	CAE-ResNet(MSE)	CAE-ResNet(SSIM)
Hazelnut	0.65	0.68	0.825	0.70	0.275	0.675
	0.87	0.73	0.67	0.71	0.89	0.69
Overall	0.79	0.71	0.73	0.71	0.66	0.68
Bottle	0.25	0.30	0.15	0.35	0.30	0.40
	0.92	0.97	0.98	0.98	0.84	0.79
Overall	0.76	0.81	0.78	0.83	0.71	0.70
Pill	0.50	0.58	0.73	0.50	0.50	0.65
	0.77	0.83	0.83	0.86	0.73	0.63
Overall	0.73	0.79	0.77	0.80	0.69	0.63
Screw	0.93	0.70	0.05	0.61	0.66	0.51
	0.61	0.63	0.87	0.68	0.64	0.58
Overall	0.69	0.65	0.66	0.66	0.64	0.56
Tile	0.97	0.64	0.94	0	0.77	0.94
	0.11	0.62	0.26	1	0.52	0.13
Overall	0.35	0.62	0.45	0.72	0.58	0.36

For the hazelnut and bottle classes, the classic DAE with MSE loss achieved the best performance. In the case of the screw class, the use of SSIM led to a higher accuracy, probably due to the geometric complexity of the spirals. For textured samples such as pill and tile, the DAE with SSIM consistently provided better results. The VGG16 encoder, despite slightly increasing overall accuracy, reduced the ability to detect normal images, whereas the classical DAE with SSIM maintained a better balance between normal and defective cases.

The performance of the contractive autoencoder is reported in Table 2. For the hazelnut and screw classes, classical CAE with MSE performed best. For the bottle class, CAE with VGG16 encoder and loss SSIM showed the highest performance. For textured samples, using SSIM also proved more effective. In particular, for the pill class, CAE with VGG16 achieved the best results. For the tile class, classical CAE showed solid performance, while CAE with VGG16 and loss MSE achieved the highest absolute accuracy, but failed to properly distinguish normal images.

DAE proved to be more effective in most cases, its performance drops with ResNet encoders, while CAE maintains more consistency across classes. The latter is more stable on textured samples, but less accurate in reconstructing details.

5. Conclusion

This study has shown that autoencoders are a valid and promising solution for unsupervised anomaly detection in industrial image data. In particular, the denoising autoencoder (DAE) achieved consistently better results than

the contractive autoencoder (CAE) across most object categories. This confirms that the introduction of noise during training encourages more robust feature learning and improves the generalization ability of the model when tested on unseen defective images.

The experiments demonstrated that the Structural Similarity Index Measure (SSIM) is more effective than Mean Squared Error (MSE) when dealing with textured surfaces. SSIM is sensitive to structural deformations, brightness, and contrast, and is therefore better suited for materials where texture plays a key role in defect identification. On smooth objects, instead, MSE remains competitive and sometimes preferable.

One important finding is that using pre-trained encoders such as VGG16 or ResNet50 does not always improve results. While VGG16 provided a slight improvement in some categories, it sometimes reduced the correct classification of normal samples. ResNet, in particular, underperformed in most configurations, possibly due to its architectural complexity and its limited adaptability to small or subtle defect patterns after fine-tuning. This indicates that a careful balance must be found between the use of pre-trained knowledge and the specific needs of anomaly detection tasks, where fine-grained pixel-level reconstruction is crucial.

From a methodological perspective, the combination of classical convolutional autoencoders with loss functions adapted to the type of image (MSE for regular shapes, SSIM for textures) provides a strong and flexible framework. Moreover, the use of a thresholding strategy based on reconstruction errors proved simple yet effective in binary classification between normal and defective cases.

Despite the relatively small size of the training data used, the models were able to achieve good classification

performance, confirming the potential of unsupervised learning techniques in real-world industrial inspection scenarios. These methods avoid the need for large labeled datasets and are capable of identifying a wide range of defects without explicit annotation.

For future work, it would be beneficial to explore the integration of attention mechanisms or generative adversarial networks (GANs) to enhance reconstruction quality and reduce false positives. Furthermore, expanding the set of evaluated loss functions to include perceptual losses or multi-scale SSIM might improve results on more complex textures. Finally, applying these models in real-time settings, with hardware constraints and on-the-fly decision-making, remains a key area for further development and practical validation in industrial contexts.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT, Grammarly in order to: Grammar and spelling check, Paraphrase and reword. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] A. M. Kamoona, A. K. Gostar, A. Bab-Hadiashar, R. Hoseinnezhad, Point pattern feature-based anomaly detection for manufacturing defects, in the random finite set framework, *IEEE Access* 9 (2021) 158672–158681. URL: <https://doi.org/10.1109/2Faccess.2021.3130261>. doi:10.1109/access.2021.3130261.
- [2] G. Capizzi, S. Coco, G. L. Sciuto, C. Napoli, A new iterative fir filter design approach using a gaussian approximation, *IEEE Signal Processing Letters* 25 (2018) 1615 – 1619. doi:10.1109/LSP.2018.2866926.
- [3] D. Połap, M. Woźniak, C. Napoli, E. Tramontana, Real-time cloud-based game management system via cuckoo search algorithm, *International Journal of Electronics and Telecommunications* 61 (2015) 333 – 338. doi:10.1515/eletel-2015-0043.
- [4] G. Capizzi, F. Bonanno, C. Napoli, Hybrid neural networks architectures for soc and voltage prediction of new generation batteries storage, in: 3rd International Conference on Clean Electrical Power: Renewable Energy Resources Impact, ICCEP 2011, 2011, p. 341 – 344. doi:10.1109/ICCEP.2011.6036301.
- [5] G. Lo Sciuto, G. Capizzi, R. Shikler, C. Napoli, Organic solar cells defects classification by using a new feature extraction algorithm and an ebnn with an innovative pruning algorithm, *International Journal of Intelligent Systems* 36 (2021) 2443 – 2464. doi:10.1002/int.22386.
- [6] F. Fiani, V. Ponzi, S. Russo, Keeping eyes on the road: Understanding driver attention and its role in safe driving, in: *CEUR Workshop Proceedings*, volume 3695, 2023, p. 85 – 95.
- [7] N. Boutarfaia, S. Russo, A. Tibermacine, I. E. Tibermacine, Deep learning for eeg-based motor imagery classification: Towards enhanced human-machine interaction and assistive robotics, in: *CEUR Workshop Proceedings*, volume 3695, 2023, p. 68 – 74.
- [8] V. Kumar, G. Nandi, R. Kala, Static hand gesture recognition using stacked denoising sparse autoencoders, in: 2014 Seventh International Conference on Contemporary Computing (IC3), 2014, pp. 99–104. doi:10.1109/IC3.2014.6897155.
- [9] B. A. Nowak, R. K. Nowicki, M. Woźniak, C. Napoli, Multi-class nearest neighbour classifier for incomplete data handling, in: *Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science)*, volume 9119, 2015, p. 469 – 480. doi:10.1007/978-3-319-19324-3_42.
- [10] S. Rifai, P. Vincent, X. Muller, X. Glorot, Y. Bengio, Contractive auto-encoders: Explicit invariance during feature extraction, 2011.
- [11] S. Bilik, K. Horak, Sift and surf based feature extraction for the anomaly detection, 2022. [arXiv:2203.13068](https://arxiv.org/abs/2203.13068).
- [12] H. Lin, B. Li, X. Wang, Y. Shu, S. Niu, Automated defect inspection of led chip using deep convolutional neural network, *Journal of Intelligent Manufacturing* 30 (2019) 2525–2534.
- [13] Y.-J. Cha, W. Choi, G. Suh, S. Mahmoudkhani, O. Buiyükoöztürk, Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types, *Computer-Aided Civil and Infrastructure Engineering* 33 (2018) 731–747.
- [14] F. Fiani, A. Puglisi, C. Napoli, et al., Enhancing object detection robustness for cross-depiction through neural style transfer, in: *CEUR WORKSHOP PROCEEDINGS*, volume 3684, CEUR-WS, 2023, pp. 15–20.
- [15] T. Finke, M. Krämer, A. Morandini, A. Mück, I. Oleksiyuk, Autoencoders for unsupervised anomaly detection in high energy physics, *Journal of High Energy Physics* 2021 (2021). URL: <https://doi.org/10.1007/2Fjhep06%282021%29161>. doi:10.1007/jhep06(2021)161.
- [16] F. Lupo, Variational autoencoder for unsupervised anomaly detection, Master's thesis, 2019. Corso di laurea magistrale in Ingegneria Matematica.
- [17] P. Vincent, H. Larochelle, Y. Bengio, P.-A. Manzagol, Extracting and composing robust features

- with denoising autoencoders, 2008, pp. 1096–1103. doi:10.1145/1390156.1390294.
- [18] L. Corvito, L. Faiella, C. Napoli, A. Puglisi, S. Russo, et al., Speech and language impairment detection by means of ai-driven audio-based techniques, in: *CEUR WORKSHOP PROCEEDINGS*, volume 3869, CEUR-WS, 2024, pp. 19–31.
- [19] A. Bionda, L. Frittoli, G. Boracchi, Deep autoencoders for anomaly detection in textured images using CW-SSIM, in: *Image Analysis and Processing – ICIAP 2022*, Springer International Publishing, 2022, pp. 669–680. URL: https://doi.org/10.1007/978-3-031-064302_56. doi:10.1007/978-3-031-06430-2_56.
- [20] A. Puglisi, F. Fiani, G. De Magistris, et al., Increased frame rate for crowd counting in enclosed spaces using gans., in: *ICYRIME*, 2023, pp. 39–45.
- [21] J. Bae, J.-H. Lee, S. Kim, Pni : Industrial anomaly detection using position and neighborhood information, 2023. arXiv:2211.12634.
- [22] M. Yang, J. Liu, Z. Yang, Z. Wu, Slsg: Industrial image anomaly detection by learning better feature embeddings and one-class classification, 2023. arXiv:2305.00398.
- [23] S. Holly, R. Heel, D. Katic, L. Schoeffl, A. Stiftinger, P. Holzner, T. Kaufmann, B. Haslhofer, D. Schall, C. Heitzinger, J. Kemnitz, Autoencoder based anomaly detection and explained fault localization in industrial cooling systems, 2022. arXiv:2210.08011.
- [24] P. Bergmann, M. Fauser, D. Sattlegger, C. Steger, Mvtec ad – a comprehensive real-world dataset for unsupervised anomaly detection, in: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 9584–9592. doi:10.1109/CVPR.2019.00982.
- [25] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015. arXiv:1409.1556.
- [26] J. Nilsson, T. Akenine-Möller, Understanding ssim, 2020. arXiv:2006.13846.