

Product defect detection based on convolutional autoencoder and one-class classification

Meryem Chaabi¹, Mohamed Hamlich¹, Moncef Garouani^{1,2}

¹Complex Cyber Physical System (CCPS) Laboratory, ENSAM, University of Hassan II, Casablanca, Morocco

²University Littoral Cote d'Opale, UR 4491, LISIC, Laboratoire d'Informatique Signal et Image de la Cote d'Opale, Calais, France

Article Info

Article history:

Received Sep 5, 2021

Revised Sep 16, 2022

Accepted Oct 16, 2022

Keywords:

Defect detection

Imbalanced data

MVTec anomaly detection

One-class classification

Support vector data description

ABSTRACT

To meet customer expectations and remain competitive, industrials try constantly to improve their quality control systems. There is hence increasing demand for adopting automatic defect detection solutions. However, the biggest issue in addressing such systems is the imbalanced aspect of industrial datasets. Often, defect-free samples far exceed the defected ones, due to continuous improvement approaches adopted by manufacturing companies. In this sense, we propose an automatic defect detection system based on one-class classification (OCC) since it involves only normal samples during training. It consists of three sub-models, first, a convolutional autoencoder serves as latent features extractor, the extracted features vectors are subsequently fed into the dimensionality reduction process by performing principal component analysis (PCA), then the reduced-dimensional data are used to train the one-class classifier support vector data description (SVDD). During the test phase, both normal and defected images are used. The first two stages of the trained model generate a low-dimensional features vector, whereas the SVDD classifies the new input, whether it is defect-free or defected. This approach is evaluated on the carpet images from the industrial inspection dataset MVTec anomaly detection (MVTec AD). During training, only normal images were used. The results showed that the proposed method outperforms the state-of-the-art methods.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Meryem Chaabi

Complex Cyber Physical System (CCPS) Laboratory, ENSAM, University of Hassan II

Casablanca, Morocco

Email: meryem.chaabi-etu@etu.univh2c.ma

1. INTRODUCTION

Efficient quality inspection is one of the cornerstones of successful manufacturing companies. Since human visual inspection is error-prone and subjective, there has been a movement towards automatic defect detection systems [1], more than that, recently we witness the transition to the era of quality 4.0, which we can define as a mature quality system that sought to leverage industry 4.0 technologies. In fact artificial intelligence and machine learning [2] had proven their abilities to perform quality inspection [3]. A plethora of defect detection methods have shown to be promising and efficient [4]. However most of them are based on supervised learning [5], requiring considerable amount of normal and abnormal (i.e. defected) samples to train the model efficiently, which is not always available in real applications. Actually, manufacturing companies adopt increasingly improvement approaches [6] as lean management and six sigma [7], with the attention of managing the fabrication process and reducing defective products by targeting zero defect

manufacturing. Defects hence rarely occur in production, and meanwhile there is enormous need to build automatic defect detection system.

One-class classification (OCC) algorithms [8] offer the potential to create automatic inspection system in early stage without having to wait for more defected samples to be collected. Consequently industrials could exploit the important amount of data in their possession, despite highly skewed class distribution. The main idea of OCC methods is to distinguish between normal and abnormal class, draw on the knowledge gained of normal samples during training.

The term OCC was first introduced by [9] to denote a category of classification algorithms that address cases where few to none defect samples are available for training; the normal class is well-defined while abnormal one is under-sampled [10] which is quite common in industrial areas [11], and with that, defects are seen as a deviation from defect-free class. The OCC concept encompasses several approaches, such as methods based on density [12], distance [13], neural networks [14], [15], and boundary approaches [16] that aims to encircle normal samples by a decision boundary. The work [17] developed adversarially trained deep neural networks, the first component works as image reconstructor, while the second represents the classifier. Autoencoders also were used to address OCC problems [18], [19]. In general, these researchers assumed that autoencoders generates higher reconstruction error for defected samples. Going through research works that address OCC, it is clearly apparent that support vector data description (SVDD) [20] is one of the extensively used algorithms in OCC applications for its satisfactory results [21]. Luo *et al.* [22] introduced a cost-sensitive SVDD, which is a method that provides different costs to classification errors. The paper [23] developed a new SVDD approach to deal with uncertain data. They trained the SVDD by using the similarity scores between examples and normal class. It is reported to outperforms regular SVDD in terms of sensitivity to noise. Shi *et al.* [24] introduced improved SVDD by combining relative density weight with SVDD. However, those SVDD based methods have difficulty dealing with large and high-dimensional datasets [25] mainly because of optimization complexity.

2. METHOD

The intent of this work is to create an automatic defect detection system. We propose a model that could manage imbalanced data and, at the same time yield to interesting results in high-dimensional spaces using SVDD algorithm. In this section, we describe the proposed defect detection system in detail, and we highlight the three components of our model: convolutional autoencoder, principal component analysis algorithm (PCA) and SVDD algorithm.

2.1. Overview

Product surface images are used as input for our algorithm and only images of normal class are present during training. The proposed system consists of three main phases. Firstly, we use a convolutional autoencoder that allows extracting image's abstract features. Once this submodel is trained, the decoder part is discarded. Then the bottleneck features vector is later fed into PCA in order to perform dimensionality reduction by inducing efficient and discriminating features representation. So that it serves as training input of the SVDD classifier. The test images are forwarding through the trained model to determine whether the image is normal or defected. The proposed system takes advantage of the convolutional autoencoder ability to extract robust features automatically; meanwhile it alleviates the problem of SVDD of handling high-dimensional datasets. Figure 1 shows the overview of the proposed approach, and Figure 2 illustrates the flowchart of the entire system.

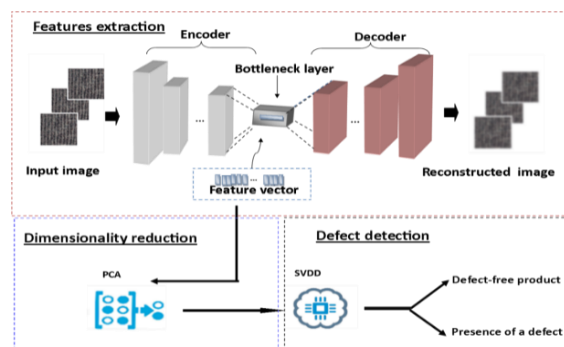


Figure 1. Shows the overview of the suggested approach

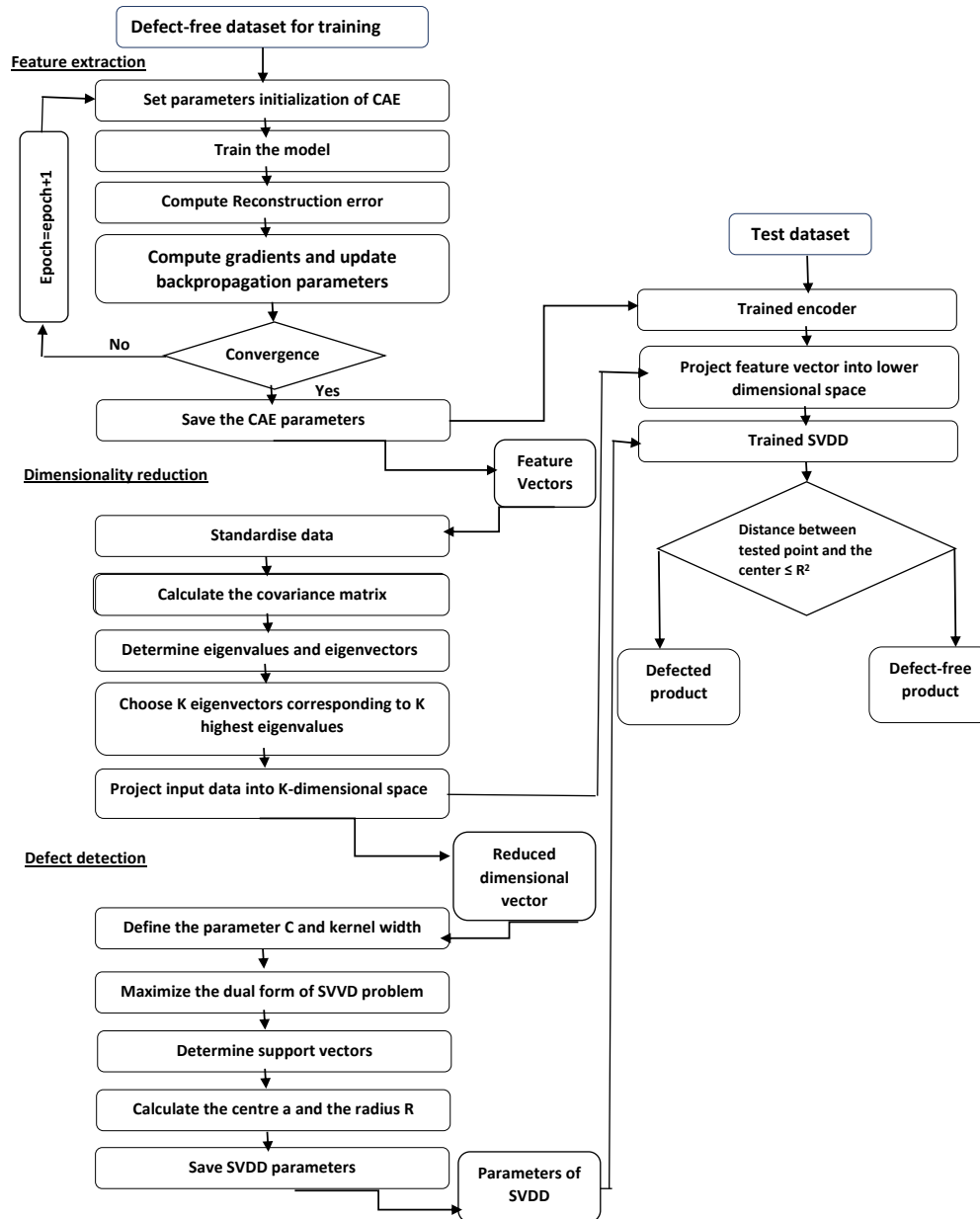


Figure 2. The flowchart of the proposed system

2.2. Features extraction

As a first stage a convolutional autoencoder (CAE) is used to extract image features. CAE can automatically provide powerful feature learning [26]. Moreover, in [27] it has been proven that the more the architecture of convolutional autoencoder becomes deeper, the more powerful features are extracted through layers, in a way that first layer provides generic features while subsequent layers extract complex features. Basically, autoencoder is a neural network that consists of two connected sub-models: an encoder and a decoder. The symmetric structure of the autoencoder allows the reconstruction of input data through the features vector provided by the bottleneck layer (i.e., encoder output) as illustrated in Figure 1. Considering X the domain of the input data samples and Z the domain of the encodings.

Given a sample $x \in X$, $z \in Z$ the encoded representation of x , and x_r the reconstructed input. The encoder-decoder process can be summarized,

$$z = g_e(x; \theta_e) \quad (1)$$

$$x_r = g_d(z; \theta d) \quad (2)$$

Where $g_e: \mathbb{X} \rightarrow \mathbb{Z}$ is the encoding function and $g_d: \mathbb{Z} \rightarrow \mathbb{X}$ the decoding function, while $\theta e, \theta d$ represent the encoder and decoder parameters respectively. A loss function is used to measure reconstruction error; we have chosen ℓ_2 loss for its simplicity and computational speed. The loss function is formulated as,

$$L(x, x_r) = \|x - x_r\|^2 \quad (3)$$

In this work, we trained the convolutional autoencoder on defect-free samples. Once the model is trained, we use the bottleneck layer as automatic features extractor. The encoder consists of five convolution layers where each layer is followed by a batch normalization layer. The max-pooling layer is used from the second convolution layer where each layers use the rectifier linear unit (ReLU) activation function,

- i. The first two convolutional layers consist of 32 filters of size 3×3 , the second one followed by a downsampling (max-pooling) layer.
- ii. The third convolutional layer consists of 64 filters of size 3×3 , followed by another downsampling layer.
- iii. The fourth convolutional layer consists of 128 filters of size 3×3 .
- iv. The fifth convolutional layer consists of 256 filters of size 3×3 .

2.3. Dimensionality reduction

Since SVDD has difficulty handling high-dimensional datasets, PCA [28] was applied to reduce the dimensionality of features vectors provided by the CAE. PCA is a statistical procedure that aims to project high-dimensional input data into a lower dimension space while retaining most of the information. Let Z be an $n \times m$ data matrix, where the rows represent the n extracted vectors while the features are represented by columns. PCA process is formulates,

- a) Standardize the m -dimensional data

$$z_{ij} \Rightarrow \frac{z_{ij} - \mu}{s_j} \quad (4)$$

Here: z_{ij} is the (i, j) th entry of Z , $i = 1, \dots, n$ and $j = 1, \dots, m$. μ and s_j are respectively the mean and the variance of j^{eme} dimension.

- b) Calculate the covariance matrix Z_c

$$Z_c = \frac{1}{n} Z^T Z \quad (5)$$

- c) Construct the n eigenvalues and n eigenvectors via covariance matrix Z_c

$$Z_c v_i = \lambda_i v_i \quad (6)$$

Where λ_i denotes the eigenvalues, v_i represents the eigenvectors and $i = 1, \dots, n$.

- a. Choose k eigenvectors corresponding to the k highest eigenvalues $k < m$.
- b. Project the input data into new k -dimensional space.

Let,

$$V = [v_1, v_2, \dots, v_k] \quad (7)$$

Then,

$$z_i' = V^T z \quad (8)$$

Where z_i' is the low-dimensional features vector.

2.4. Defect detection

The defect detection task is done using SVDD. This algorithm attempts to determine a hypersphere with the minimum volume that encircles almost all training data. The spherical boundary is characterized by a center a and a radius R , hence during test, points that fall outside the boundary are considered as abnormal as illustrated in Figure 3. The parameters R and a are defined by (9),

$$\min R^2 + C \sum_{i=1}^n \xi_i \tag{9}$$

Subject to,

$$\|z'_i - a\|^2 \leq R^2 + \xi_i \forall i \tag{10}$$

$$\xi_i \geq 0 \forall i \tag{11}$$

here ξ_i : are slack variables that allow some points in training data to be outside the sphere and C represents a penalty constant that controls the trade-off between the volume of the hypersphere and rejected points.

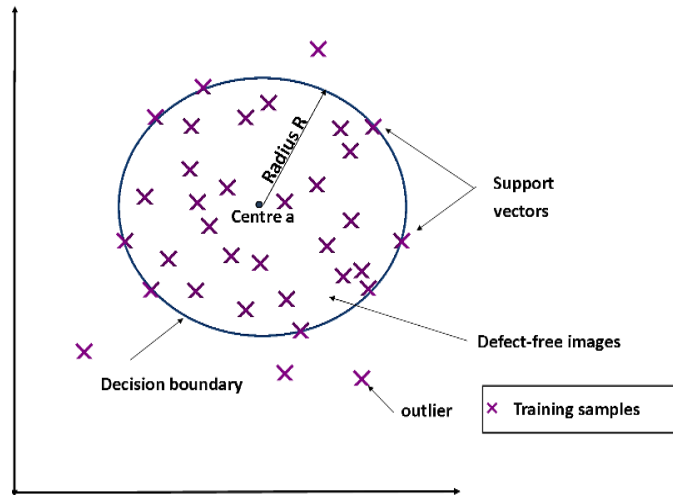


Figure 3. Illustration of SVDD

The optimization problem (9) is solved via its dual formulation,

$$\max \sum_{i=1}^n \alpha_i \langle z'_i, z'_i \rangle - \sum_{i,j} \alpha_i \alpha_j \langle z'_i, z'_j \rangle \tag{12}$$

s.t.

$$0 \leq \alpha_i \leq C \quad i=1, 2, \dots, n \tag{13}$$

$$\sum_{i=1}^n \alpha_i = 1 \tag{14}$$

Where α_i are lagrange multipliers, and $\langle z'_i, z'_j \rangle$ denotes the inner product of z'_i and z'_j . The hypersphere boundary is determined by support vectors for which $0 \leq \alpha_i \leq C$. The center 'a' is calculated as,

$$a = \sum_{i=1}^n z'_i \alpha_i \tag{15}$$

The radius R is then computed by selecting an arbitrary support vector z'_k ,

$$R^2 = \|z'_k - a\|^2 = \langle z'_k, z'_k \rangle - 2 \sum_{i=1}^n \alpha_i \langle z'_k, z'_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle z'_i, z'_j \rangle \tag{16}$$

For each new input y, the distance between this tested point and the center a is computed as,

$$\text{dist}^2(y) = \langle y, y \rangle - 2 \sum_{i=1}^n \alpha_i \langle y, z'_i \rangle + \sum_{i,j} \alpha_i \alpha_j \langle z'_i, z'_j \rangle \tag{17}$$

Thus, points with $\text{dist}^2(y) \geq R^2$ are considered as defected.

According to [20], replacing the inner product $\langle z'_i, z'_j \rangle$ in (12) with appropriate kernel function $K(z'_i, z'_j)$ provides more flexibility to define the boundary. We have chosen to use the Gaussian kernel function, given that it is the kernel reported to be yielding satisfactory results in OCC applications [29]. The Gaussian kernel is defined as,

$$K(z'_i, z'_j) = \exp\left(-\frac{\|z'_i - z'_j\|^2}{2\sigma^2}\right) \quad (18)$$

Where σ is the kernel width.

3. RESULTS AND DISCUSSION

The experimental environment is a computer with an i7-7700HQ CPU, 16GB of RAM, Nvidia RTX 2080 Ti GPU, running Windows 10, and we used the Tensorflow library to implement the proposed approach. To demonstrate the effectiveness of our suggested system for defect detection, we conducted a set of experiments on the carpet images from MVTEC anomaly detection (MVTEC AD) dataset [30], which is a benchmark of natural images dedicated to unsupervised anomaly detection that mimics real industrial inspection applications. The carpet dataset consists of 390 high-resolution images of 1024×1024 pixels, divided into 72% of normal data and 28% of abnormal data. The training set is composed of defect-free images, while the test set consists of normal images as well as images containing 5 different types of fine-grained anomalies on the carpet's surface like threads, cuts, metal contamination, color and holes. The dataset overview is shown in Figure 4. Data augmentation was needed to diversify the training data set and make the model more generalizable. To do so, random flips and rotations strategies were applied generating 1449 image. Finally, both categories were rescaled to 512×512 pixels.

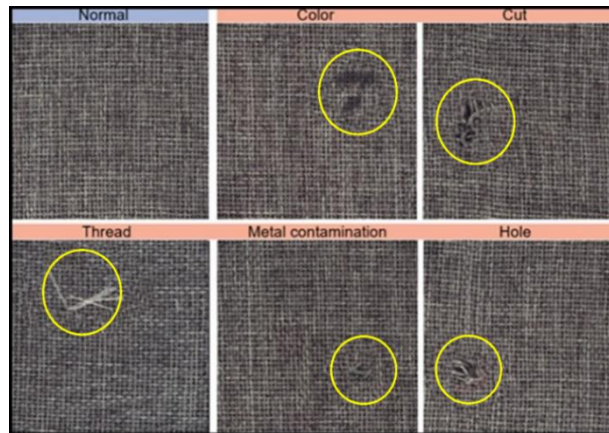


Figure 4. Images of carpet dataset

We adopted the accuracy, area under the receiver operating characteristic curve (AUROC) and F1 score to measure the performance of the proposed method. Accuracy and F1 score are computed as,

$$Precision = \frac{TP}{TP+FP} \quad (19)$$

$$Recall = \frac{TP}{TP+FN} \quad (20)$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (21)$$

$$F1score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \quad (22)$$

Where true positive (TP): is the number of images correctly classified as defective.
True negative (TN): is the number of images correctly classified as normal.

False positive (FP): is the number of images incorrectly classified as defective.

False negative (FN): is the number of images incorrectly classified as normal.

For this implementation, we assumed that there are no outliers in training data. We defined $C=1$, which implied that $\xi_{(i)}=0$ [31]. The kernel width σ is optimized through cross validation. We investigated the performance of our model on the carpet images and compare it with state-of-the-art methods. Furthermore, we conducted comparison with the two-stage model “Convolutional autoencoder-SVDD” (CAE-SVDD), to prove the importance of PCA algorithm in the proposed system.

As shown in Table 1, the proposed approach method outperforms the compared methods on all metrics at detecting the defects in carpet dataset. The AUROC result of CAE-PCA-SVDD model was better, with mean value 0.20% higher than the other algorithms. As can be observed in the comparison with CAE-SVDD algorithm, applying PCA to reduce features vector dimensionality, leads to better performance than the CAE-SVDD structure. Hence the proposed defect detection system could operate well in high dimensional spaces. However, the proposed approach has some limitations. As shown in Figure 5, the system failed to recognize a defect when the contrast of defected part is similar to the texture of the image.

Table 1. Experimental results of different approaches on the carpet dataset

Approach	AUROC	Accuracy	F1 score
Liu <i>et al.</i> [18]	0.78	-	-
Wang <i>et al.</i> [19]	0.94	0.71	0.88
Bergmann <i>et al.</i> [32]	0.87	0.67	0.87
Bergmann <i>et al.</i> [32]	0.59	0.50	0.54
CAE-SVDD	0.71	0.60	0.67
CAE-PCA-SVDD	0.98	0.94	0.97

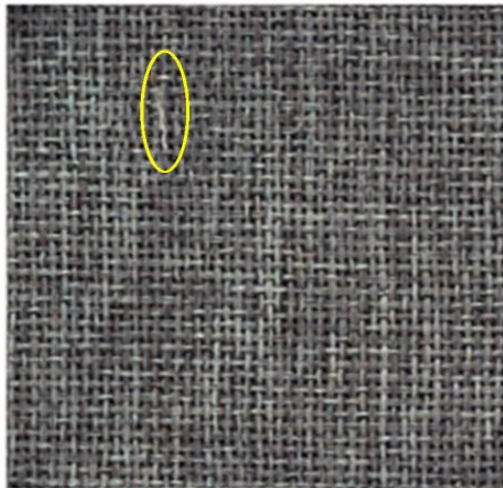


Figure 5. A false negative case





4. CONCLUSION

The current paper proposes automatic defect detection system that attempts to mitigate the lack of representative samples of defected images. Hence it can be applied in cases where significant amount of normal data is available, while the defected class is characterized by few samples. The suggested approach comprises three components: convolutional autoencoder for image features extraction, then PCA is applied to reduce dimensionality of features vector, and finally SVDD is implemented to classify images. The motivation behind this structure is to build automatic defect detection system that is able to handle effectively imbalanced data. The proposed method has proved to be efficient in experiment developed on carpet dataset from MVTec AD. It is important to note that SVDD parameters (C and the width σ) vary from one dataset to another. In this work C was defined as $C=1$, while σ was chosen through cross-validation to find best accuracy. As a future work, we plan to evaluate the proposed approach on different datasets, and also utilize an automatic method to determine the parameters: (C, σ), which will facilitate the application of our approach. Furthermore we will investigate further techniques to improve the ability of the proposed system to handle texture features.





REFERENCES

- [1] N. N. S. A. Rahman, N. M. Saad, A. R. Abdullah, M. R. M. Hassan, M. S. S. M. Basir, and N. S. M. Noor, "Automated real-time vision quality inspection monitoring system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 2, pp. 775–783, 2018, doi: 10.11591/ijeecs.v11.i2.pp775-783.
- [2] M. Hamlich and M. Ramdani, "Improved ant colony algorithms for data classification," *Proceedings of 2012 International Conference on Complex Systems, ICCS 2012*, 2012, doi: 10.1109/ICoCS.2012.6458563.
- [3] A. Abdullahi, N. A. Samsudin, M. R. Ibrahim, M. S. Aripin, S. K. A. Khalid, and Z. A. Othman, "Towards IR4.0 implementation in e-manufacturing: artificial intelligence application in steel plate fault detection," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 20, no. 1, pp. 430–436, 2020, doi: 10.11591/ijeecs.v20.i1.pp430-436.
- [4] N. M. Saad, A. R. Abdullah, W. H. W. Hasan, N. N. S. A. Rahman, N. H. Ali, and I. N. Abdullah, "Automated vision based defect detection using gray level co-occurrence matrix for beverage manufacturing industry," *IAES International Journal of Artificial Intelligence*, vol. 10, no. 4, pp. 818–829, 2021, doi: 10.11591/ijai.v10.i4.pp818-829.
- [5] Z. Ren, F. Fang, N. Yan, and Y. Wu, "State of the art in defect detection based on machine vision," *International Journal of Precision Engineering and Manufacturing - Green Technology*, vol. 9, no. 2, pp. 661–691, 2022, doi: 10.1007/s40684-021-00343-6.
- [6] G. Marodin, A. G. Frank, G. L. Tortorella, and T. Netland, "Lean product development and lean manufacturing: testing moderation effects," *International Journal of Production Economics*, vol. 203, pp. 301–310, 2018, doi: 10.1016/j.ijpe.2018.07.009.
- [7] M. Smetkowska and B. Mrugalska, "Using six sigma DMAIC to improve the quality of the production process: a case study," *Procedia - Social and Behavioral Sciences*, vol. 238, pp. 590–596, 2018, doi: 10.1016/j.sbspro.2018.04.039.
- [8] A. Mujeeb, W. Dai, M. Erdt, and A. Sourin, "One class based feature learning approach for defect detection using deep autoencoders," *Advanced Engineering Informatics*, vol. 42, 2019, doi: 10.1016/j.aei.2019.100933.
- [9] L. D. Moya, MM and Koch, MW and Hostetler, "One-class classifier networks for target recognition applications," *NASA STI/Recon Technical Report N*, vol. 93, p. 24043, 1993.
- [10] M. Chaabi and M. Hamlich, "A sight on defect detection methods for imbalanced industrial data," *ITM Web of Conferences*, vol. 43, p. 1012, 2022, doi: 10.1051/itmconf/20224301012.
- [11] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski, "Using meta-learning for automated algorithms selection and configuration: an experimental framework for industrial big data," *Journal of Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00612-4.
- [12] J. Hamidzadeh and M. Moradi, "Improved one-class classification using filled function," *Applied Intelligence*, vol. 48, no. 10, pp. 3263–3279, 2018, doi: 10.1007/s10489-018-1145-y.
- [13] J. Wang, W. Liu, K. Qiu, H. Xiong, and L. Zhao, "Dynamic hypersphere SVDD without describing boundary for one-class classification," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3295–3305, 2019, doi: 10.1007/s00521-017-3277-0.
- [14] X. E. Pantazi, D. Moshou, and A. A. Tamouridou, "Automated leaf disease detection in different crop species through image features analysis and one class classifiers," *Computers and Electronics in Agriculture*, vol. 156, pp. 96–104, 2019, doi: 10.1016/j.compag.2018.11.005.
- [15] M. Garouani, A. Ahmad, M. Bouneffa, M. Hamlich, G. Bourguin, and A. Lewandowski, "Towards big industrial data mining through explainable automated machine learning," *International Journal of Advanced Manufacturing Technology*, vol. 120, no. 1–2, pp. 1169–1188, 2022, doi: 10.1007/s00170-022-08761-9.
- [16] K. L. Li, H. K. Huang, S. F. Tian, and W. Xu, "Improving one-class SVM for anomaly detection," *International Conference on Machine Learning and Cybernetics*, vol. 5, pp. 3077–3081, 2003, doi: 10.1109/icmlc.2003.1260106.
- [17] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli, "Adversarially learned one-class classifier for novelty detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3379–3388, doi: 10.1109/CVPR.2018.00356.
- [18] W. Liu *et al.*, "Towards visually explaining variational autoencoders," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Nov. 2020, pp. 8642–8651, doi: 10.1109/CVPR42600.2020.00867.
- [19] L. Wang, D. Zhang, J. Guo, and Y. Han, "Image anomaly detection using normal data only by latent space resampling," *Applied Sciences (Switzerland)*, vol. 10, no. 23, pp. 1–19, 2020, doi: 10.3390/app10238660.
- [20] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Machine Learning*, vol. 54, no. 1, pp. 45–66, 2004, doi: 10.1023/B:MACH.0000008084.60811.49.
- [21] S. S. Khan and M. G. Madden, "One-class classification: taxonomy of study and review of techniques," *Knowledge Engineering Review*, vol. 29, no. 3, pp. 345–374, 2014, doi: 10.1017/S026988891300043X.
- [22] J. Luo, L. Ding, Z. Pan, G. Ni, and G. Hu, "Research on cost-sensitive learning in one-class anomaly detection algorithms," in *International Conference on Autonomic and Trusted Computing*, 2007, pp. 259–268, doi: 10.1007/978-3-540-73547-2_27.
- [23] B. Liu, Y. Xiao, L. Cao, Z. Hao, and F. Deng, "SVDD-based outlier detection on uncertain data," *Knowledge and Information Systems*, vol. 34, no. 3, pp. 597–618, 2013, doi: 10.1007/s10115-012-0484-y.
- [24] P. Shi, G. Li, Y. Yuan, and L. Kuang, "Outlier detection using improved support vector data description in wireless sensor networks," *Sensors (Switzerland)*, vol. 19, no. 21, 2019, doi: 10.3390/s19214712.
- [25] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016, doi: 10.1016/j.patcog.2016.03.028.
- [26] M. Maggipinto, C. Masiero, A. Beghi, and G. A. Susto, "A convolutional autoencoder approach for feature extraction in virtual metrology," *Procedia Manufacturing*, vol. 17, pp. 126–133, 2018, doi: 10.1016/j.promfg.2018.10.023.
- [27] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*, 2014, pp. 818–833, doi: 10.1007/978-3-319-10590-1_53.
- [28] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010, doi: 10.1002/wics.101.
- [29] A. Bounsiar and M. G. Madden, "Kernels for one-class support vector machines," *ICISA 2014 - 2014 5th International Conference on Information Science and Applications*, 2014, doi: 10.1109/ICISA.2014.6847419.
- [30] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "MVTEC ad-a comprehensive real-world dataset for unsupervised anomaly detection," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2019-June, pp. 9584–9592, 2019, doi: 10.1109/CVPR.2019.00982.
- [31] W.-C. Chang, C.-P. Lee, and C.-J. Lin, "A revisit to support vector data description," *Department of Computer Science, National Taiwan University, Taipei, Taiwan, Technical Report*, 2013.
- [32] P. Bergmann, S. Löwe, M. Fauser, D. Sattlegger, and C. Steger, "Improving unsupervised defect segmentation by applying structural similarity to autoencoders," in *VISIGRAPP 2019 - Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, Jul. 2019, vol. 5, pp. 372–380, doi: 10.5220/0007364503720380.





BIOGRAPHIES OF AUTHORS

Meryem Chaabi     pursued Master degree in industrial engineering from faculty of Science and Technology, Hassan First University, Morocco. She is currently pursuing PhD at CCPS Laboratory ENSAM, University of Hassan II, Morocco. She can be contacted at email: meryem.chaabi-etu@etu.univh2c.ma.



Dr. Mohamed Hamlich     is Professor at ENSAM Casablanca, since 2014. In 2013, he obtained his doctoral thesis in Computer Science from Hassan II University in Casablanca. His areas of research interests include robotics, artificial intelligence, IoT and big data. He is the president of the association of Connected Objects and Intelligent Systems and the director of “CCPS” Laboratory at ENSAM-Casablanca. <http://hamlich.sadasc.net/>. He can be contacted at email: moha.hamlich@gmail.com.



Moncef Garouani     received his Master’s degree in computer engineering from the USMBA University of Morocco in 2019. He is currently working towards his PhD degree at the ULCO university in joint supervision thesis with the UH2C university and the School of engineering's and business' sciences and technics. His main research topics are machine learning, AutoML, meta-learning, data mining and explainable AI. He can be contacted at email: mgarouani@gmail.com.