Applied Intelligence (2024) 54:13160–13177
https://doi.org/10.1007/s10489-024-05647-z

Voxel-wise segmentation for porosity investigation of additive manufactured parts with 3D unsupervised and (deeply) supervised neural networks

Domenico Iuso[1,6] · Soumick Chatterjee[2,3] · Sven Cornelissen[4] · Dries Verhees[5] · Jan De Beenhouwer[1,6] · Jan Sijbers[1,6]

Accepted: 25 June 2024 / Published online: 31 October 2024

Abstract

Additive Manufacturing (AM) has emerged as a manufacturing process that allows the direct production of samples from digital models. To ensure that quality standards are met in all samples of a batch, X-ray computed tomography (X-CT) is often used in combination with automated anomaly detection. For the latter, deep learning (DL) anomaly detection techniques are increasingly used, as they can be trained to be robust to the material being analysed and resilient to poor image quality. Unfortunately, most recent and popular DL models have been developed for 2D image processing, thereby disregarding valuable volumetric information. Additionally, there is a notable absence of comparisons between supervised and unsupervised models for voxel-wise pore segmentation tasks. This study revisits recent supervised (UNet, UNet++, UNet 3+, MSS-UNet, ACC-UNet) and unsupervised (VAE, ceVAE, gmVAE, vqVAE, RV-VAE) DL models for porosity analysis of AM samples from X-CT images and extends them to accept 3D input data with a 3D-patch approach for lower computational requirements, improved efficiency and generalisability. The supervised models were trained using the Focal Tversky loss to address class imbalance that arises from the low porosity in the training datasets. The output of the unsupervised models was post-processed to reduce misclassifications caused by their inability to adequately represent the object surface. The findings were cross-validated in a 5-fold fashion and include: a performance benchmark of the DL models, an evaluation of the post-processing algorithm, an evaluation of the effect of training supervised models with the output of unsupervised models. In a final performance benchmark on a test set with poor image quality, the best performing supervised model was UNet++ with an average precision of 0.751 ± 0.030, while the best unsupervised model was the post-processed ceVAE with 0.830 ± 0.003. Notably, the ceVAE model, with its post-processing technique, exhibited superior capabilities, endorsing unsupervised learning as the preferred approach for the voxel-wise pore segmentation task.

Keywords Additive manufacturing · Unsupervised models · Deeply supervised models · Voxel-wise segmentation · Anomaly detection · X-ray CT

1 Introduction

Additive manufacturing is gaining interest since it is a low-waste production technique that can conveniently produce complex objects from a given CAD file [1]. The latest developments in 3D printing technology allow to print metallic alloys effectively, as in the case of Selective Laser Melting [2]. While this technique has many advantages, a key challenge is printing metallic alloys without defects. The mechanical behaviour of 3D printed parts, including tensile or fatigue stress behaviours, greatly depends on their overall structural integrity [3]. Defects such as common keyhole or lack-of-fusion pores [4] can seriously degrade the mechanical properties of printed parts by becoming initiation centres for crack development [5]. For non-destructive evaluation of the printing process and quality assurance of the printed part, X-CT is often employed [6, 7]. X-CT has been used to analyse the structural integrity of samples [8], the internal and external surface properties [9, 10], as well as identification and quantification of the number of defects that arise from the AM process [11, 12].

Jan De Beenhouwer and Jan Sijbers contributed equally to this work.

Extended author information available on the last page of the article

Detecting anomalies from X-CT data is a challenging task (due to, for example, inhomogeneous density of the sample, a low contrast-to-noise ratio, or beam hardening artefacts) that may lead to incorrect segmentation. For such a task, data-driven deep learning-based approaches have been shown to outperform traditional machine learning techniques because they can better handle complex and varied definitions of anomalies [7, 13–15]. Anomalies can be detected in a supervised or unsupervised fashion. While supervised methods require an annotated data set, unsupervised methods are more desirable because the training data need not be annotated. Apart from reducing the technical overhead for the user due to the gathering of annotated data, unsupervised models nullify the impact that noisy annotations have on the performance of the model. On the other hand, the general challenge that researchers of unsupervised approaches face are the high recall rate and/or low precision when these approaches are compared to their supervised counterpart [16].

The majority of studies on voxel-wise segmentation tasks with DL techniques are focused on the analysis of a stack of 2D images [17–22]. For voxel-wise segmentation of pores in AM samples, a 2D approach is sub-optimal as small pores usually span only a few voxels in the three directions in X-CT images and suffer from a low contrast-to-noise ratio. Moreover, pores may be elongated as they usually exhibit anisotropy [23], with a high risk of being ignored by 2D pixel-wise segmentation methods. Recognising this shortfall, Wong et al. introduced the concept of 3D pore detection models in their pioneering study [13]. While their initial implementation with a UNet architecture demonstrated promise, their study did not delve into the exploration of deep supervision, other neural models or training patterns. Deep supervision can yield more reliable results since the hidden layers of the models are enticed to comply with the desired output [24]. However, training supervised models directly on a dataset with reduced porosity may seriously affect detection performance due to a strong class imbalance between the number of voxels that belong to pores and those that do not [25]. Moreover, supervised models are known to be highly sensitive to training labels. Unsupervised models, especially those based on VAE architectures, suffer a well-known issue of blurry representation of input images, because the models learn a low-dimensional representation that may not capture fine details [26]. For these models, the difference between the input and output images alone, which is usually the voxel-wise anomaly score, may not be a good indication of anomaly presence. Therefore, the voxel-wise anomaly score can be enhanced with a more complex anomaly score or dedicated post-processing [27, 28].

In this work, 2D supervised and unsupervised DL models are first revisited and subsequently extended to 3D for voxel-wise segmentation of pores on X-CT samples of varying alloys. Utilising a 3D patch-based approach and integrating data augmentation, our segmentation method aims to be independent of the material and shape of AM samples. Moreover, it ensures spatial consistency by operating within the 3D image domain. Several deeply supervised models are trained (UNet++ [29], UNet 3+ [30], MSS-UNet [31] and ACC-UNet [32]) while a more traditional UNet [33] serves as a baseline to compare the remaining four architectures, which are composed of similar building blocks to easier the comparison. To address the class imbalance that arises from the low amount of defects, the models are trained with the Focal Tversky (FTL) function, which allows models to penalise anomalies more effectively [34]. Since the FTL is a parametric function, the optimal parameters were found with a parameter search. The roster of unsupervised models (VAE [35], ceVAE [27], gmVAE [36], vqVAE [37] and RV-VAE [38]) aims to compare older and novel complex model architectures. To reduce misclassifications, the anomaly score of these models is post-processed, due to the inability of these models to represent the object surface adequately. Finally, the supervised models are trained again with the post-processed output of an unsupervised model instead of (potentially) noisy annotations, effectively making the training process unsupervised, to evaluate the impact on the performance of the supervised models. On the best performing model of all the experiments, an experiment was run to assess the decrease in performance when lowering the number of X-ray projections and exposure.

Summarising, the main contributions of this paper are as follows:

- First cross-validated assessment of multiple 3D DL models for voxel-wise pore segmentation in AM samples, comparing supervised and unsupervised approaches using a (patch-based) method. The neural networks, initially designed as 2D models, were tailored for the 3D context to harness volumetric information effectively.
- A post-processing algorithm is proposed and evaluated to address the issue of blurry image representation in VAE models.
- The impact of using unsupervised model labels instead of heuristic algorithm labels for training the DL models is evaluated.

2 Materials

Various DL models for voxel-wise segmentation of pores were trained using 3D X-ray CT images of AM samples. To this end, AM samples were manufactured through the selective laser melting process, in a continuous (CLM, [39]) or pulsed laser melting (PLM, [40]) strategy. Five cylindrical

samples of different materials were 3D printed (as shown in Fig. 1): one with TiAl6V4, two with CoCr-DG1 alloy and two with stainless steel 316L. Printing the test objects in multiple materials allowed to assess the effectiveness of voxel-wise pore segmentation across different materials. In the CAD model used for the 3D printing, the cylinders had an eight of 20 mm and a diameter of 5 mm. In addition to the cylindrical samples, a stainless steel 316L cube with an edge length of 9 mm was also printed. The cube was specifically printed to provide an object with different shape and poorer X-CT image quality, which is useful for evaluating porosity in a challenging visual environment and to ensure that DL models are not learning information regarding the shape of the object. These samples were essential for this study as their X-CTs provided the digital dataset with which the neural networks could be trained to segment the porosity. Porosity was intentionally induced in all samples using controlled laser parameters, as described in [41].

Next, 3D images of the AM samples were generated by scanning them with a micro-CT X-ray system [42] and reconstructed with the FDK algorithm [43] with a 10 μm resolution. The imaging settings, such as filament power, peak kV of the anode, exposure time, source filter, etc., were selected for each cylindrical sample to ensure comparable image quality. However, the geometrical distances and the number of projections were kept constant for all cylindrical samples, with a source-to-detector distance (SDD) of 650.0 mm, a source-to-object distance (SOD) of 43.33 mm, and 4283 projections. The cubic sample was scanned with different SDD (950.0 mm) and SOD (63.33 mm) and had a lower number of X-ray projections (2878) than the other scans. The X-CT of the cubic sample was also affected by severe cone-beam artefacts and poor beam-hardening compensation. The cubic sample was particularly challenging due to its different geometry and visual environment (as noticeable in Fig. 2), making it useful for evaluating porosity.

Fig. 1 Some samples used in this study. From left to right, a stainless steel 316L (CLM), a CoCr-DG1 (PLM), and a TiAl6V4 (CLM) sample

3 Methods

Several DL models were trained to segment porosity from X-CT scans of AM samples at the voxel level. Voxel-wise annotations, necessary for both training and performance evaluation, were provided using the method described in Section 3.1. These models employed either supervised or unsupervised approaches, detailed in Section 3.2.

For the training of supervised models, the class imbalance of labels was addressed using the FTL function, which will be discussed in Section 3.3. The class imbalance arose from the low amount of pores (positive instance of labels) within the training dataset. After training, and only for the unsupervised models, the anomaly score is post-processed, as unsupervised models are known to produce blurry representations of the input. The post-processing procedure is explained in Section 3.4.

3.1 Dataset annotation

To assign a label to each voxel of the X-CTs comprising the datasets, which indicates whether it is a pore or not, a 3D processing algorithm was applied. The high-level pseudo-code in Algorithm 1 outlines the pore identification process.

Algorithm 1 Pore extraction from X-CT images.

Input
CT Volumetric X-CT image
Output
$pore_{mask}$ Volumetric binary mask representing pores

Get low-value voxels through Otsu_{thr} of CT
Get the background mask by FloodFilling the low-values from a corner of CT
Get the watertight object mask from binary inversion of the background mask
Get low-value voxels by Otsu_{thr} of CT inside the object mask
$pore_{list} \leftarrow$ Collect connected low-value voxels inside the object

for all pores in $pore_{list}$ **do**
    if size of $pore <$ minimal size **then**
        remove $pore$ from $pore_{list}$
    end if
end for
$pore_{mask} \leftarrow$ Convert $pore_{list}$ to a volumetric mask

The algorithm for extracting pores from volumetric X-CT images begins by creating a binary mask to distinguish low-value voxels using Otsu thresholding ($Otsu_{thr}$). Subsequently, a background mask is obtained through FloodFill starting from a corner of the X-CT image, isolating low-attenuating values. The binary inversion of this background

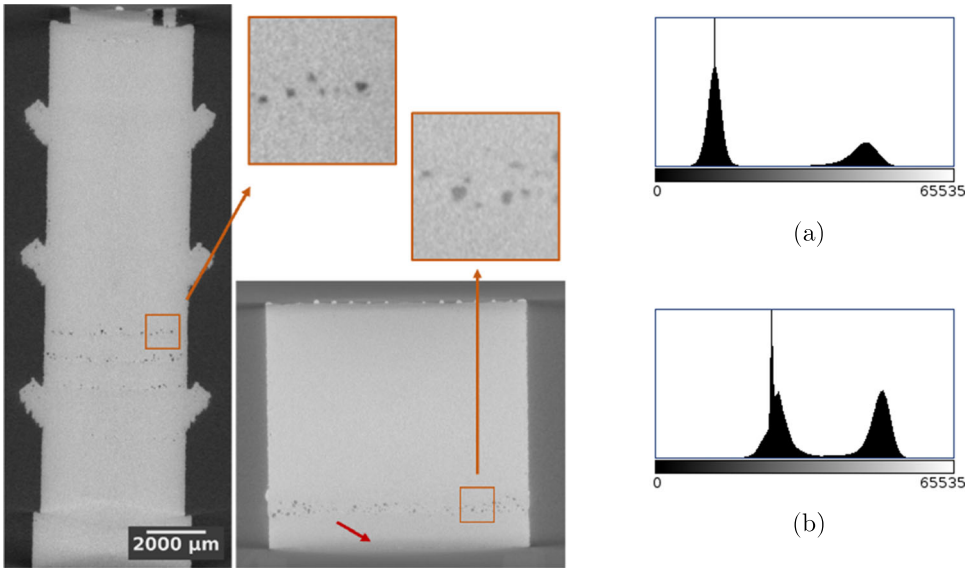


Fig. 2 A slice of the X-CT of a cylindrical sample (left) and of the cubic sample (right), with equal colour-map and scale. While all the X-CT of cylindrical samples share similar image quality, the cube has stronger artefacts (which are particularly visible at the extremities of the cube) and consequently less contrast. The histograms (a) and (b) refer to the cylindrical sample and of the cubic volume, respectively. The two peaks in each histogram are related to the background (lower) and foreground (higher) colours. The quality of each sample is defined by its distance between the peaks and the broadness of the bells, which are influenced by artefacts and noise

mask yields the watertight object mask, effectively separating the image into air and the watertight object. To identify low-value voxels corresponding to pores, a second Otsu thresholding operation is applied within the object. To address the potential misclassification of pores due to imaging noise, pores-voxels are screened based on shape criteria. Initially, pores in a 6-connected 3D neighbourhood are identified and listed. The boundary box of each pore is then examined, and the pore is excluded from the list if its boundary box is smaller than 2 in at least one dimension. This shape-based filtering is implemented to improve the reliability of the pore identification process [44, 45]. The filtered $pore_{list}$ is then converted into a volumetric binary mask ($pore_{mask}$), providing a voxel-wise representation of pore locations. It's important to note that any residual misclassification arising from partial-volume effects and imaging artefacts contributes to the overall noise of the labels.

Accurately and reliably labelling the X-CT scan of the cubic sample was a challenging task due to its poor image quality, as discussed in Section 2. Given the limitations of automated voxel-wise annotation, manual labelling was the only viable option to achieve the desired level of accuracy and dependability in the labels.

3.2 Deep learning models

The study used two types of models: VAE-based models (VAE [35], ceVAE [27], gmVAE [36], vqVAE [37] and RV-VAE [38]) and UNet-based models (UNet [33], MSS-UNet [31], UNet++ [29], UNet 3+ [30] and ACC-UNet [32]). The VAE-based models were trained in an unsupervised manner using unlabelled data, while the UNet-based models were trained in a supervised manner. Starting from their original 2D implementation, these networks were extended to accept 3D inputs of size 64^3 by substituting all 2D layers with their 3D counterparts.

3.2.1 Supervised models

UNet is a popular encoder-decoder architecture that has shown promising results in many semantic voxel-wise segmentation tasks. MSS-UNet, UNet++, and UNet 3+ are extensions of the original UNet architecture. MSS-UNet incorporates multi-scale guidance in the decoding process during training, enabling it to capture more fine-grained details and to have a more coherent processing of information in the decoding stage. UNet++ includes a nested and dense skip-connection structure to capture more multi-scale features, while UNet 3+ uses a more powerful encoder with multi-resolution inputs. To ensure consistency, UNet and MSS-UNet were built using the same encoding/decoding building blocks as for UNet++ and UNet 3+ [30]. This approach made it easier to compare the results of different architectures and understand how they impact the final outcome in voxel-wise segmentation tasks. Vision Transformers have recently addressed complexity challenges, making them

a viable and competitive solution for visual tasks, where a notable work is [46]. Building on these advancements, the core concepts of Transformers have been integrated into ResNet models, surpassing the performance of Swin Transformers. Another notable development involves incorporating essential Transformer ideas into a convolution-based neural model called ACC-UNet. This model has shown promise in segmentation tasks, motivating its use in our current study. MSS-UNet, UNet++, UNet 3+ and ACC-UNet are deeply supervised during this study, which means they are trained with a loss function calculated on multiple inner layers to supervise the learning process effectively. In contrast, the original UNet architecture is not deeply supervised.

3.2.2 Unsupervised models

The VAE-based models were trained in an unsupervised manner to learn a compressed and disentangled representation of the input data. During training, the VAE models learned to reconstruct images from the compressed representations that resemble the input images as closely as possible. The reconstruction error, which quantifies the discrepancy between the input and output of the unsupervised models, was adopted as the anomaly score. Since the introduction of the VAE model in 2014 by Kingma and Welling, it has been used in a variety of studies for voxel-wise anomaly detection (e.g. [47–49]). The ceVAE model has similar architecture as VAE but a more complex definition of the loss. During training, ceVAE uses "masked" input data where certain patches within the image are fixed to a specific value. The model uses an ad-hoc loss function to infer the missing or distorted voxels within the masked zone, which helps the network to capture the context of the image. This peculiarity of the model may have a positive impact on the score, since it can prevent the network to learn to represent the pores within the training dataset. On the contrary, the gmVAE and vqVAE models are more complex than the VAE architecture, enabling them to catch features of the input 3D images that could not be interpreted by the coarser architecture of VAE. The gmVAE model assumes that each input data point's latent representation is generated by one of several possible Gaussian distributions, each with a different mean and variance, and identifies which distribution in the mixture is most likely to have generated the latent representation of each input data point during training. The vqVAE model is based on the idea of vector quantisation, where the continuous latent space is discretised into a set of discrete codes. The model comprises an encoder network that maps the input images to a discrete code book, followed by a decoder network that maps the discrete codes to the reconstructed input images. The vqVAE model was adapted to 3D inputs without additional alterations, except for an extra encoding/decoding stage that processes larger input patches of 64^3 instead of the default 32^3. The RV-VAE model eliminates stochastic sampling, directly incorporating latent space information into decoder layers as continuous random variables. Applying the inherent mathematical prior during decoding leads to a more precise representation, making it appealing for segmentation tasks. As we use a final sigmoid activation function for all the neural models, the related RV-model for this function is provided in the Appendix C.

3.3 Focal Tversky loss function

In our pore segmentation task, the number of voxels belonging to the foreground class (pores) is much smaller than the number of voxels belonging to the background class, in the training dataset. This class imbalance results in a bias towards the background class during training, which leads to poor voxel-wise segmentation performance. In order to address the problem of class imbalance in semantic segmentation tasks, the Focal Tversky Loss (FTL) was proposed as a modification to the Tversky Loss [34], and is defined as follows:

$$\mathrm{FTL} = \left(1 - \frac{\mathrm{TP}}{\mathrm{TP} + \alpha\,\mathrm{FN} + \beta\,\mathrm{FP}}\right)^{\gamma} \tag{1}$$

The FTL depends on the number of true negatives (TN), false negatives (FN), and false positives (FP), where FN and FP are weighted by α and β, respectively. By adjusting the values of these parameters, the FTL can be fine-tuned to emphasise either precision or recall. In addition, the FTL also includes a parameter γ, which controls the degree to which the FTL prioritises correcting misclassifications by adjusting the weight given to the Tversky Loss function. If $\gamma = 1$, the FTL reduces to the standard Tversky loss and, if is also true that $\alpha = \beta = 0.5$, to the Dice-Sørensen loss. If $\gamma > 1$, the FTL function will assign a higher weight to the correction of misclassifications. This means that the loss function will be more sensitive to false negatives and false positives, and the model will prioritise the correction of misclassifications over the correct classification of the majority class. As a result, the model will be better at identifying instances of the minority class but may struggle to accurately classify instances of the majority class. The degree to which the model's sensitivity to misclassifications increases will depend on the value of γ. In case of deep supervision, the FTL is calculated at each supervised stage and averaged with geometric progression weights (1, 1/2, 1/4, etc.).

3.4 Post-processing

During the prediction or testing procedure, each of the models inferred patches belonging to the X-CT scan and then aggregated them back together to obtain an output volume with the same size as that of the input.

Only for the unsupervised models, the output was post-processed to amend the scarce quality that these models have in representing the fine details of the samples, as the surface. The surface of each of our samples has unique characteristics, due to different printing processes and polishing procedures, which can never be properly represented with an Autoencoder (AE). While AEs are designed to learn a concise representation of the input, their ability to faithfully reproduce high-fidelity images depends on factors such as the training dataset's size and diversity, the complexity of input data, and the model's architecture and hyperparameters. As this is a beneficial feature that makes the AEs potentially unable to reproduce anomalies that may be present in the training dataset, it comes with the cost of inaccuracies near the surface of the samples. To counteract this, we introduce a compensation mechanism that suppresses the anomaly score near the sample surface. The computation of the new voxel-wise anomaly score, denoted as A_{pores}, involves subtracting the spatially blurred derivative D of the inferred volume $\hat{V}$ from the original anomaly score A. As previously mentioned, the neural models struggles to faithfully represent the surface of samples, leading to pronounced derivatives of the inferred volume along the border. The elements of D are determined by the sum of the absolute voxel-wise derivatives in the x, y, and z directions of the predicted volume $\hat{V}$. These derivatives are represented as $d_{ijk} = \|\partial_x \hat{v}_{ijk}\| + \|\partial_y \hat{v}_{ijk}\| + \|\partial_z \hat{v}_{ijk}\|$, where $\hat{v}_{ijk}$ corresponds to the (i, j, k) voxel in $\hat{V}$.

The formulation for A_{pores} is expressed as:

$$A_{pores} = \max(0, A - \lambda^* G_{\sigma^*}(D)) \tag{2}$$

Values for the standard deviation σ^* of the Gaussian smearing kernel G and the scaling factor λ^* are determined through an on-the-fly optimisation process outlined in Formula (3). This optimisation process aims to minimise the disparity between the anomaly score and the Gaussian-blurred absolute sum of derivatives, utilising the mean of the L1-norm as a metric. Both λ and σ are considered to be positive parameters in this context.

The optimisation problem is formally stated as:

$$\lambda^*, \sigma^* = \arg\min_{\lambda,\sigma} \|A - \lambda G_\sigma(D)\|_1 \tag{3}$$

The results of the experiment Section 4.5 show the benefits of applying the proposed technique.

4 Experiments

The X-CT images were organized into training, validation, and testing sets, as explained in Section 4.1. All models were trained using a common training framework, detailed in Section 4.2. For the evaluations presented in this section, the labelled X-CT volumes were compared with the output of the DL models, after the output 3D patches were aggregated.

More specifically, the patch-extraction pipeline extracted overlapping patches from the input volume, each with half of their length overlapping with neighboring patches. These patches were segmented by the neural networks and then combined by computing an average value among the overlapping patches. This approach ensured a comprehensive evaluation of the model's performance on the X-CT volumes.

4.1 Dataset

The X-CT images of several AM samples composed the digital dataset for training, validation, and testing of the DL models. In a 5-fold manner, the X-CT images of the cylindrical samples were organised into 4 samples for the train-set and 1 sample for the validation-set. Noise, image artefacts, and misclassified voxel-wise labels (commonly referred to as 'noisy labels') can negatively affect training and lead to inaccurate predictions. To mitigate the influence of noisy labels during training and to expand the training sets, data augmentation was employed [50]. The data augmentation created novel spatial configurations by flipping of patches in random directions and elastic distortion while teaching the networks to be resilient against noise, specific attenuation of samples, and artefacts such as cone-beam and beam-hardening. After data augmentation was applied at every training epoch to each of the cylindrical samples, which have around 800x800x2000 voxels, 3D patches of 64x64x64 voxels were extracted and supplied to the neural networks.

4.2 Training

The deep learning framework was based on the Pytorch [51] 2.0.1, Pytorch-lightning [52] 2.0.2 and the CUDA [53] 11.6 libraries and it is publicly available (https://github.com/snipdome/nn_3D-anomaly-detection). The 3D patch extraction, aggregation and data augmentation were based on the TorchIO libraries [54] version 0.18.84. A unique main seed propagated throughout the libraries ensures that all the extraction from random distributions were reproducible. Each of the models was trained with the Adam optimiser (learning rate of 0.0001) and halted through early stopping when the loss value did not decrease by more than 0.0001 for 40 consecutive epochs.

4.3 Parameter search for the FTL function

As different values of the α, β, γ parameters sensibly affect the performance of models trained with the FTL function [55], the optimal values were identified with grid search approach. A 5-fold cross-validation strategy evaluated the

performance of the model with different parameter combinations, while the γ parameter was kept at 0.5 (as in [55]). The grid search space spanned the parameter-space uniformly from 0.1 to 0.9 for each of the variables, for a total of 4 steps. For each combination of α and β, the model was trained in a 5-fold cross-validation, resulting in a total of 16 different combinations of α and β and a total of 80 model trainings. In addition to the α and β parameters, another grid search identified the optimal γ parameter in the FTL. A higher value of γ puts more emphasis on minimising false positives and false negatives, which can be useful in tasks where the cost of misclassification is high. So, even though the author of the FTL had suggested a value of $4/3$ for the γ parameter [34], the optimal γ parameter turned out to vary for the current application of this work. The γ grid search had a total of 8 steps ranging from $1/3$ to 2, for a total of 40 trainings.

4.4 Cross-validation of performance of the DL models

All the supervised and unsupervised models have been trained in a 5-fold cross-validation, for a total of 50 trainings. In the case of supervised models, they were trained with the optimal parameters found during the experiment Section 4.3. After training, the performance has been evaluated, for each fold, on both the validation-set and the challenging test-set.

4.5 Cross-validation of performance of post-processed unsupervised models

For this experiment, the unsupervised models are compared in cross-validation before and after the application of a post-processing algorithm presented in Section 3.4. Since the post-processing happens after the aggregation of all the patches composing a X-CT volume, it is possible to compare the models before and after post-processing, without the need to re-train the models. Also in this case, the performance has been evaluated, for each fold, on both the validation-set and the challenging test-set.

4.6 Cross-validation of performance of supervised models re-trained with unsupervised models

In this experiment, the anomaly score of the (best performing) unsupervised model of experiment Section 4.5 was used as label for the training of supervised models, for each fold. Training in such a way would make the overall pipeline unsupervised, which, apart from being a favourable feature for the user, it would theoretically allow the UNet-family to reproduce the task of the unsupervised model (and its post-processing algorithm). A total of 25 trainings has been performed.

4.7 Model complexity

For this experiment, all the neural models have been compared with regards to their memory footprint and computational cost. The networks were fed with a one-element batch with size 1x64x64x64 and analysed during their complete forward and backward operation.

4.8 Cross-validation of performance of the best performing model in extreme visual scenarios

In this final experiment, the best performing model in the previous experiments has been tested when the image quality of the challenging test-set has been worsened by lowering X-ray exposure and number of projections. This test is designed to show how the performance decreases in extreme visual scenarios. The number of X-ray projections of the challenging test-set was reduced to 50% and 33.3%. The simulation of lower exposure of X-ray projections is achieved by adding Poisson distributed noise. The exposure was lowered to 75%, 50% and 25% of the original values, which corresponded in an increase in the imaging noise over the X-ray projections.

5 Results and discussions

Section 5.1 presents the cross-validation results for selecting the optimal parameters of the FTL function. These parameters were used to train all the supervised models employed in the voxel-wise segmentation task cross-validation, whose results are shown in Sections 5.2 and 5.4. Section 5.4 compares the supervised models trained with the FTL function using heuristic labels and labels generated by the post-processed output of the best performing unsupervised model. The best performing unsupervised model was established based on the performance results presented in Section 5.3.

5.1 Parameter search for the FTL function

The initial parameter search for α and β has been conducted on all the folds of the cross-validation, and the average results are shown in Table 1. As apparent from the results, the optimal values for the α and β parameters are 0.633 and 0.1, respectively. Subsequently, with these optimal parameters, the optimal γ parameter has been searched for each fold, and summary results are shown in Table 2. In this case, there is good agreement among folds that $\gamma = 1$ ensures the best performance. For the sake of completeness, the fold-wise results have been included in the appendix for both parameter searches (Appendix B).

Table 1 Average Dice-Sørensen score and standard error of the models evaluated across the related validation dataset, depending on the α/β parameters of the FTL

Dice-Sørensen score				
α \ β	0.1	0.37	0.63	0.9
0.9	0.74 ± 0.06	0.74 ± 0.04	0.75 ± 0.07	0.75 ± 0.07
0.63	0.78 ± 0.04	0.76 ± 0.06	0.76 ± 0.06	0.73 ± 0.07
0.37	0.75 ± 0.05	0.76 ± 0.07	0.74 ± 0.07	0.70 ± 0.05
0.1	0.76 ± 0.06	0.73 ± 0.07	0.72 ± 0.07	0.71 ± 0.06

5.2 Cross-validation of performance of the DL models

The segmentation results of the cross-validation technique were evaluated using two metrics: the area under the ROC curve (AUC) and the average precision (AP) of the precision-recall (PR) curve. While the AUC is a commonly used metric, it can be misleading in the presence of class imbalance [56, 57]. To address this issue, PR curves were used to evaluate the performance of algorithms, as recommended by [57]. Therefore, both PR and ROC curves were used to evaluate the models.

The voxel-wise segmentation task of the models was evaluated for each fold, whose summary ROC-AUC and AP values are shown in Fig. 3 and detailed numerically in Tables 3 and 4 for the validation dataset and the challenging test set. The cross-validated results related to the validation dataset (represented with blue colour in Fig. 3) indicate that supervised models have been generally better trained to be consistent with labels than the unsupervised methods. The results on the challenging dataset with high artefacts and manually labelled (represented with orange colour) show a clear drop of the score for all the models, as expected for the considerations in Section 3.1. Moreover, it is noticeable that the score of some of the unsupervised models is even higher than that of the supervised ones for the challenging dataset. Although these results may not seem consistent with the validation dataset, it should be noted that in both cases the labels were generated in different ways: either with a heuristic labelling algorithm or via manual annotation. Among the supervised models, there is no significant difference in performance, which suggests that deep supervision and the different architecture of the models is not inducing a significant difference in performance. On the other hand, a noticeable difference in scores is present between ceVAE and gmVAE/vqVAE on the challenging dataset, which is significant for vqVAE with a confidence of 95% (Welch's t-test, p-value $1.98*10^{-4}$ (AUC) and $1.05 * 10^{-5}$ (AP)). The higher degree of complexity of gmVAE and vqVAE is not favourable to the segmentation task by the mean of the anomaly score. These models have been capable of learning how to reproduce defects within the input samples, so the reconstruction error is not as high in the proximity of defects as it is with simpler VAEs. On another note, VAE and ceVAE are most robust with respect to the quality of the input image, since the AP/AUC scores are almost unvaried between the validation and the challenging test-set (AP/AUC differences lower than or approximately equal to a decimal point), when opposed to the other models (AP/AUC differences exceeding a decimal point). As further consideration, pores may occasionally lie on the border of a patch during inference, which may result in segmentation errors. Potential loss of accuracy caused by such segmentation errors is mitigated by averaging results from overlapping patches. Accuracy can be enhanced also by increasing the size of the 3D patches. While increasing overlap among patches will inevitably raise memory requirements, the computational complexity associated with larger patches can be managed by employing DL models with sparse operations, as demonstrated by Sparse CNNs [58]. This approach holds promise for future research (Tables 3 and 4).

5.3 Cross-validation of performance of post-processed unsupervised models

By applying post-processing to the output of the VAE models (Fig. 4), the considerations of the previous section about supervised models become more evident. When post-processing is applied to the output of the VAE and ceVAE models, which have not learned to visually represent pores, their AP scores increase by almost 2 decimal points on both datasets, while their AUC remains almost unchanged. On the other hand, post-processing adversely affected the performance of gmVAE and vqVAE, which is to be expected since the derivative of the output of these models is non-negligible near the edge of the sample as well as near the

Table 2 Average Dice-Sørensen score and standard error of the models evaluated across the related validation dataset, depending on the γ parameters of the FTL

Dice-Sørensen score								
	0.76 ±0.04	0.78 ±0.04	0.79 ±0.03	0.82 ±0.04	0.72 ±0.03	0.70 ±0.06	0.64 ±0.06	0.59 ±0.04
γ	0.33	0.5	0.67	1.0	1.33	1.5	1.67	2.0

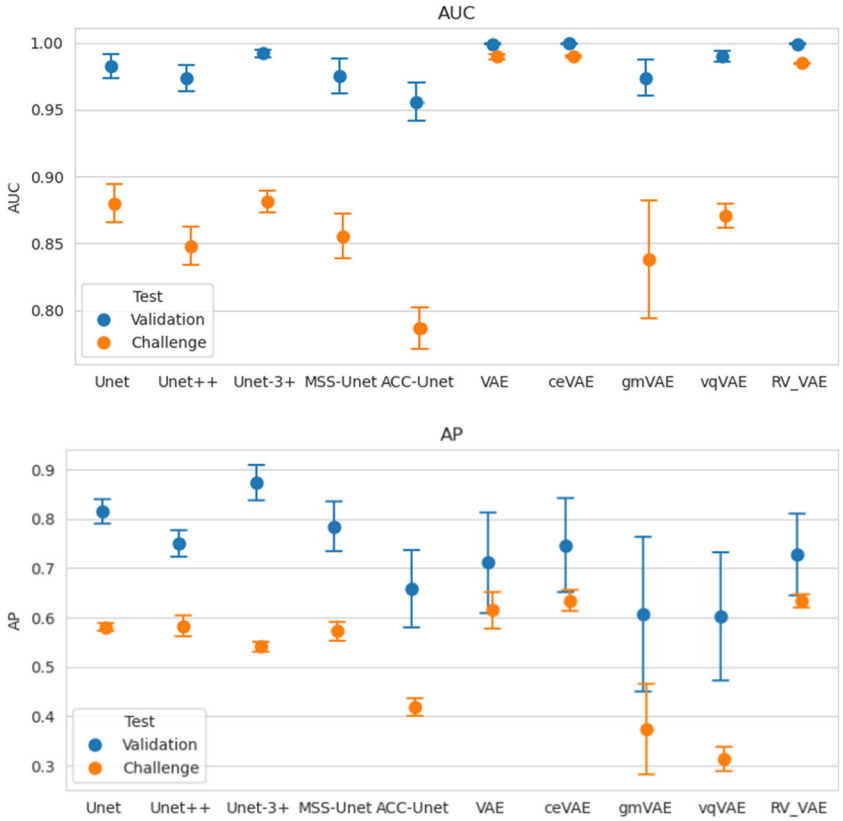


Fig. 3 Point-plots of the average ROC-AUC and AP scores (with confidence interval) of the models evaluated on the validation dataset and on the challenging dataset. The quantitative values are shown in Tables 3 and 4

pores. This behaviour is noticeable in the ROC and PR classifier curves for the challenging case as shown in Fig. 5 (other ROC and PR graphs are shown in the Appendix A). The greater complexity of gmVAE/vqVAE models enables them to replicate defects within the samples, leading to a reduction in anomaly scores and compromising performance. This effect intensifies with the application of post-processing, as illustrated in Fig. 6, where a validation sample is inferred by both ceVAE and gmVAE with and without post-processing of the anomaly scores. These results highlight that a more complex architecture is not always advantageous, particularly when anomalies exist within the training dataset. Additionally, it can be observed from Figs. 3 and 4 that the scores of VAE and ceVAE are still resilient against the poor image quality of the challenging test-set, compared to the drastic drop in performance of the supervised networks.

Table 3 Average ROC-AUC and AP scores (with confidence interval) of the supervised (△) and unsupervised (◇) models evaluated on the validation dataset

Model	AP	AUC
MSS-UNet$^{\triangle}$	0.784 ± 0.050	0.975 ± 0.013
UNet$^{\triangle}$	0.815 ± 0.025	0.982 ± 0.009
UNet++$^{\triangle}$	0.750 ± 0.026	0.974 ± 0.009
UNet-3+$^{\triangle}$	**0.873 ± 0.036**	0.992 ± 0.003
ACC-UNet$^{\triangle}$	0.658 ± 0.078	0.955 ± 0.014
VAE$^{\diamond}$	0.711 ± 0.101	0.999 ± 0.001
ceVAE$^{\diamond}$	**0.746 ± 0.094**	0.999 ± 0.001
gmVAE$^{\diamond}$	0.607 ± 0.156	0.974 ± 0.014
vqVAE$^{\diamond}$	0.602 ± 0.129	0.990 ± 0.004
RV-VAE$^{\diamond}$	0.728 ± 0.082	0.999 ± 0.001

The bold is highlighting the best score

Table 4 Average ROC-AUC and AP (with confidence interval) of the supervised (△) and unsupervised (◇) models evaluated on the challenging test-set

Model	AP	AUC
MSS-UNet$^{\triangle}$	0.572 ± 0.019	0.856 ± 0.017
UNet$^{\triangle}$	0.581 ± 0.008	0.880 ± 0.014
UNet++$^{\triangle}$	**0.583 ± 0.021**	0.848 ± 0.014
UNet-3+$^{\triangle}$	0.541 ± 0.010	0.882 ± 0.008
ACC-UNet$^{\triangle}$	0.418 ± 0.018	0.786 ± 0.016
VAE$^{\diamond}$	0.615 ± 0.038	0.990 ± 0.002
ceVAE$^{\diamond}$	**0.635 ± 0.021**	0.990 ± 0.001
gmVAE$^{\diamond}$	0.374 ± 0.092	0.838 ± 0.044
vqVAE$^{\diamond}$	0.313 ± 0.025	0.871 ± 0.009
RV-VAE$^{\diamond}$	0.634 ± 0.014	0.985 ± 0.001

The bold is highlighting the best score

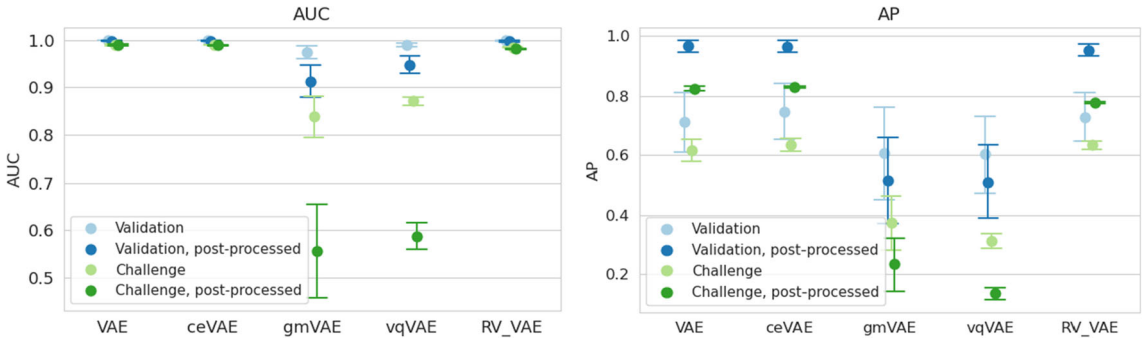


Fig. 4 Point-plots of the average ROC-AUC and AP of the models (with confidence interval) evaluated on the validation dataset and on the challenging dataset, with and without post-processing. Solely the performance of unsupervised models is shown, since the post-processing of the output is defined for them only. The values in textual form are shown in Tables 5 and 6

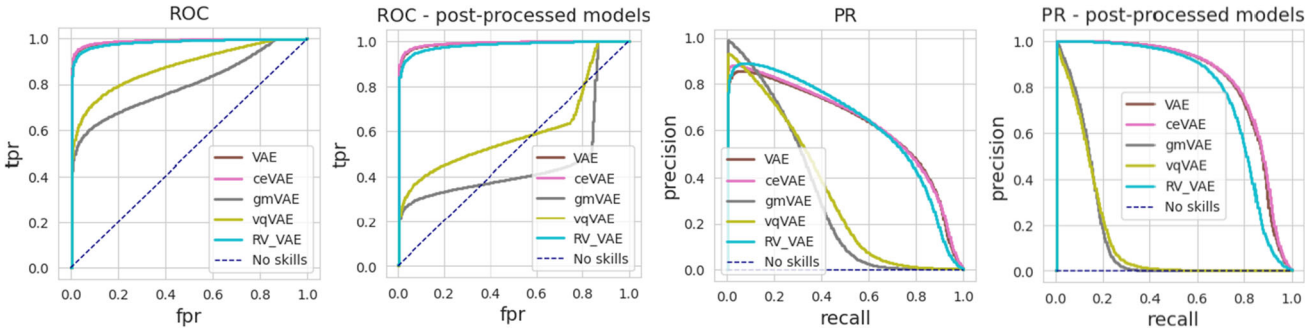
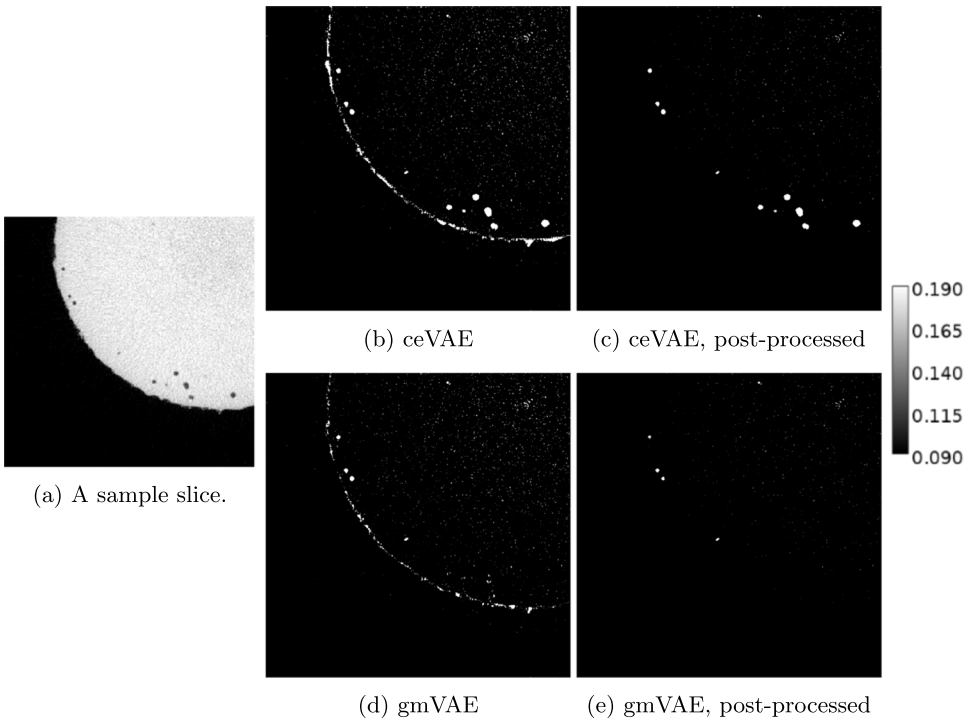


Fig. 5 Graph of the ROC and PR curves of cross-validated performance for all models. The graphs represent the median trend of the fold-wise performance on the challenging dataset without (left) and with post-processing (right) of the aggregated output



(a) A sample slice.

(b) ceVAE

(c) ceVAE, post-processed

(d) gmVAE

(e) gmVAE, post-processed

Fig. 6 A slice took from a validation dataset (a) and its voxel-wise anomaly score accordingly to ceVAE (b) and gmVAE (d). Post-processing the anomaly scores (c, e) reveals a beneficial impact, particularly for models that unequivocally classify pores as anomalies. The color-scale represents the intensity levels in the anomaly score images

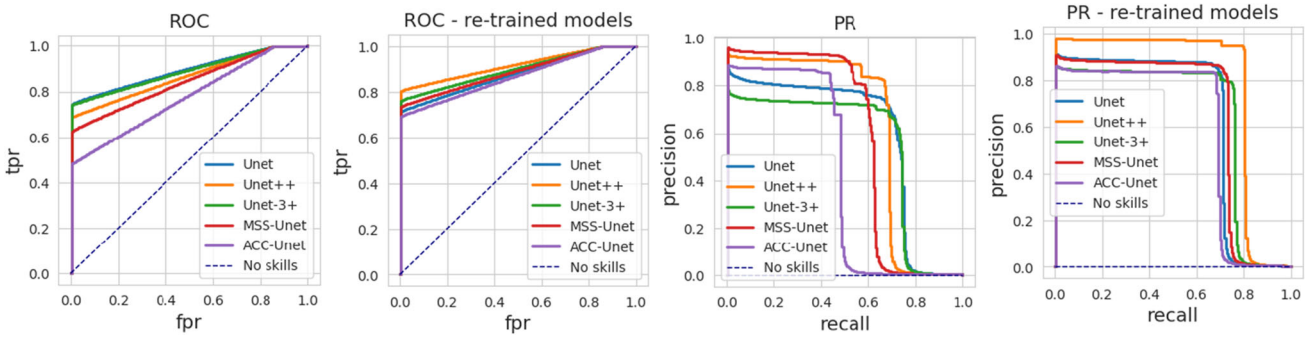


Fig. 7 Graph of the ROC and PR curves of cross-validated performance for all models. The graphs represent the median trend of the fold-wise performance on the challenging dataset, with Otsu-based labels (left) and post-processed ceVAE-generated labels (right)

5.4 Cross-validation of supervised models trained with labels generated by an unsupervised model

By using ceVAE (the best performing model) to generate labels for the samples, the supervised models could be trained from scratch to detect pores. The necessary steps for the production of these labels by ceVAE were the post-processing (with the algorithm described in Section 3.1) and the suppression of smaller pores. The results are shown in Figs. 7 and 8, and detailed numerically in Tables 5 and 6. Higher performance is achieved by using the unsupervised labels, confirmed by both AUC and AP for all the models. These results confirm the observations in Section 5.2 that the different architectures of the models are not significantly affecting the scores for this voxel-wise segmentation task (Tables 5 and 6).

5.5 Model complexity

Tables 7 and 8 presents key metrics related to the model complexity of each neural model, including the number of parameters, peak memory usage, and Multiply-Accumulate Operations (MACs). The number of parameters indicates the quantity of floating-point numbers that need to be stored in video memory, reflecting the minimal memory occupancy required to store the model. Conversely, the forward/backward peak memory highlights the memory needed to process an input with a batch size of 1. Lower memory requirements lead to larger permissible batch sizes, consequently reducing training times. The MACs value encapsulates information about the speed of the neural models to process a single 3D patch. In the case of X-CT volumes sized at 800x800x2000, comprised of numerous overlapped patches by half of their patch-length, the forward operation during the inference phase necessitates multiple repetitions to process the entire volume. The cumulative MACs operations, represented as "Total MACs" in the table, quantify the overall computational workload.

It is noteworthy that the memory usage of the UNet-family generally exceeds that of the VAE-family in forward/backward passes, with the exceptions of ceVAE and gmVAE. Specifically, the high memory requirements of ceVAE are visible only during the training procedure, as it is related solely to the backward pass. Nevertheless, ceVAE has shown good performance during the previous experiments (Sections 5.3 and 5.4). Conversely, the huge memory

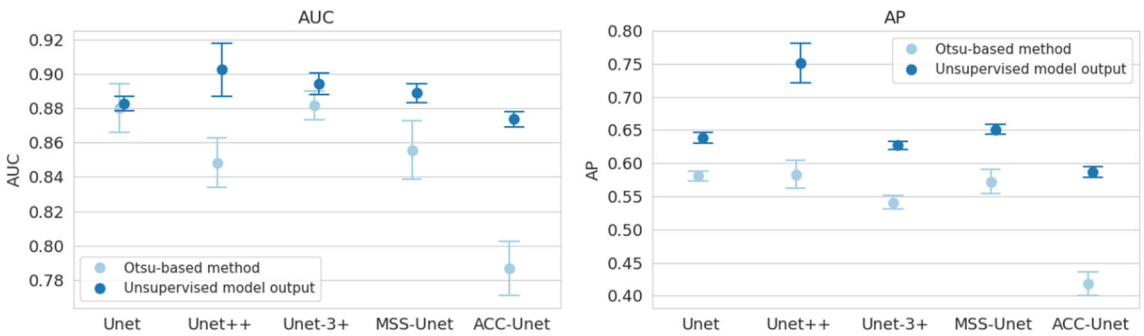


Fig. 8 Point-plots of the average ROC-AUC and AP (with confidence interval) of the supervised models evaluated on the challenging dataset. The graphs highlight the different performance when these models were supervised by the Otsu-based method and with the labels provided by the unsupervised models. The values in textual form are shown in Table 7

Table 5 Average ROC-AUC and AP (with confidence interval) of the unsupervised models evaluated on the validation dataset, with post-processing of the output

Model	AP	AUC
VAE	0.964 ± 0.020	0.998 ± 0.001
ceVAE	0.964 ± 0.021	0.999 ± 0.001
gmVAE	0.516 ± 0.144	0.913 ± 0.033
vqVAE	0.512 ± 0.122	0.948 ± 0.019
RV-VAE	0.951 ± 0.020	0.998 ± 0.001

Solely the performance of unsupervised models is shown, since the post-processing of the output is defined for them only

requirement of gmVAE and MACs do not directly translate in outstanding performance for the prior experiments.

5.6 Cross-validation of performance of the best performing model in extreme visual scenarios

By reducing the number of X-ray projections of the challenging X-CT scan and reducing the exposure of each X-ray projection, the quality of the reconstructed X-CT scan decreased. The best performing model, which was shown to be the post-processed ceVAE, was applied to these X-CT scans. An exemplary visual representation of the voxel-wise segmentation is shown in Fig. 10, related to the post-processed output of the ceVAE model, trained on the 1st fold. In this figure, a small portion of a slice of the cube is shown, in which pores are visible that were induced with off-nominal parameters of the melting laser during the printing. The degradation of the segmentation performance is noticeable due to the increasing number of voxels classified as pores (as shown in Fig. 9). Interestingly, while reducing the number of X-ray projections from 4283 (the dataset used for training/validation) to 2878 (the original challenging test-set) did not significantly affect the performance (Fig. 4), further reductions in the number of projections had a significant impact on the performance scores (Fig. 9). Another point to note is the trend exhibited by the AP scores at low exposure levels ranging from 50-25%. Specifically, reducing the number of projections from 50% to 33.3% led to a slight increase in the AP scores. When data is highly noisy and the number of projections is relatively low, adding some more X-ray projections may not always lead to better image quality of the reconstructed X-CT scans (Fig. 10). This is because the additional (noisy) projections can also introduce more noise into the reconstructed images. This can be observed from the fact that the trend gradually disappears as the exposure level increases from 25% to 100% (Fig. 10).

Table 6 Average ROC-AUC and AP (with confidence interval) of the unsupervised models evaluated on the challenging test set, with post-processing of the output

Model	AP	AUC
VAE	0.824 ± 0.007	0.989 ± 0.002
ceVAE	**0.830 ± 0.003**	0.989 ± 0.001
gmVAE	0.234 ± 0.089	0.555 ± 0.099
vqVAE	0.138 ± 0.020	0.587 ± 0.028
RV-VAE	0.777 ± 0.004	0.981 ± 0.001

Solely the performance of unsupervised models is shown, since the post-processing of the output is defined for them only
The bold is highlighting the best score

Table 7 Average ROC-AUC and AP (with confidence interval) of the supervised models re-trained with the labels generated by ceVAE and evaluated on the challenging dataset

Model	AP	AUC
MSS-UNet	0.651 ± 0.008	0.889 ± 0.005
UNet	0.639 ± 0.008	0.882 ± 0.004
UNet++	**0.751 ± 0.030**	0.902 ± 0.015
UNet-3+	0.627 ± 0.006	0.894 ± 0.006
ACC-UNet	0.586 ± 0.008	0.874 ± 0.004

The bold is highlighting the best score

6 Conclusions

This study explores recent Deep Learning techniques for voxel-wise pore segmentation in X-CT images of AM samples. Employing Tversky focal loss, deep supervision, and 3D patch-based training, we adapt various 2D neural models (UNet, UNet++, UNet 3+, MSS-UNet, ACC-UNet, VAE, ceVAE, gmVAE, vqVAE, RV-VAE) to 3D, with both supervised and unsupervised training strategies. Post-processing of unsupervised models and training supervised models with unsupervised inferred labels are also investigated.

The comprehensive comparison of all neural models reveals that supervised models (UNet-3+, AP 0.873 ± 0.036) outperform unsupervised models (ceVAE, AP 0.746 ± 0.094), a trend not upheld when tested on a challenging X-CT test set. In this scenario, ceVAE (AP 0.635 ± 0.021) outperforms supervised neural models (UNet++, AP 0.583 ± 0.021). The application of additional post-processing, beneficial for VAE and ceVAE (AP 0.830 ± 0.003 on the challenging test set), proves counterproductive for gmVAE and vqVAE due to the more complex architecture of these models. This complexity lead the models to be able to replicate defects within the training samples, thereby impairing the voxel-wise anomaly score. Although using an ideal pore-free training dataset might improve the scores of gmVAE

Table 8 Model complexity metrics for each neural model, including forward/backward peak memory usage and MACs, are specified for batch-size 1

Model	# Parameters	Forward/Backward Peak Memory	MACs	Total MACs
MSS-UNet	1.328 M	383.740 / 424.840 MB	14.270 G	69.678 T
UNet	1.325 M	353.924 / 390.925 MB	14.124 G	68.967 T
UNet++	1.503 M	933.490 / 1005.831 MB	34.821 G	170.024 T
UNet-3+	1.672 M	1571.642 / 1720.766 MB	84.881 G	414.460 T
ACC-UNet	5.062 M	6897.734 / 7269.893 MB	39.724 G	193.966 T
VAE	29.024 M	44.703 MB / 189.918 MB	3.698 G	18.058 T
ceVAE	140.650 M	33.765 MB / 778.901 MB	8.344 G	40.742 T
gmVAE	383.650 M	774.129 MB / 1842.710 MB	207.48 G	1013.096 T
vqVAE	2.511 M	17.688 MB / 32.701 MB	8.471 G	41.361 T
RV-VAE	29.024 M	223.288 MB / 230.196 MB	0.456 G	2.176 T

Total MACs represent operations for processing an 800x800x2000 voxel volume, with a 3D patch overlap of half the patch-length

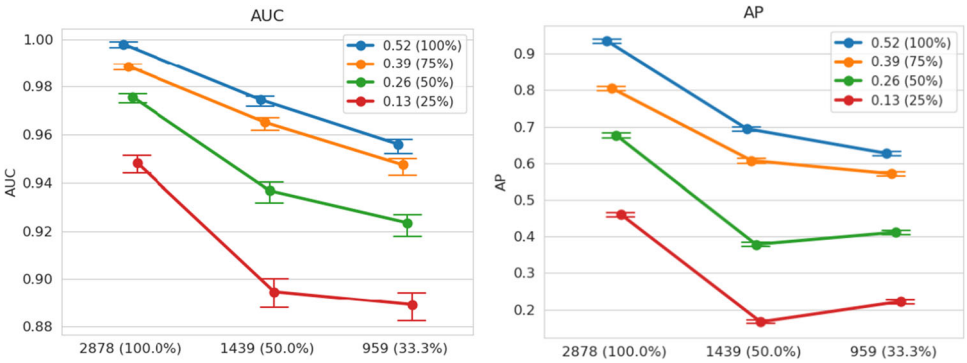


Fig. 9 Point-plots of the average ROC-AUC and AP (with confidence interval) of the anomaly score of ceVAE evaluated on the challenging test-set when the image quality is lowered by reducing the number of X-ray projections or exposure

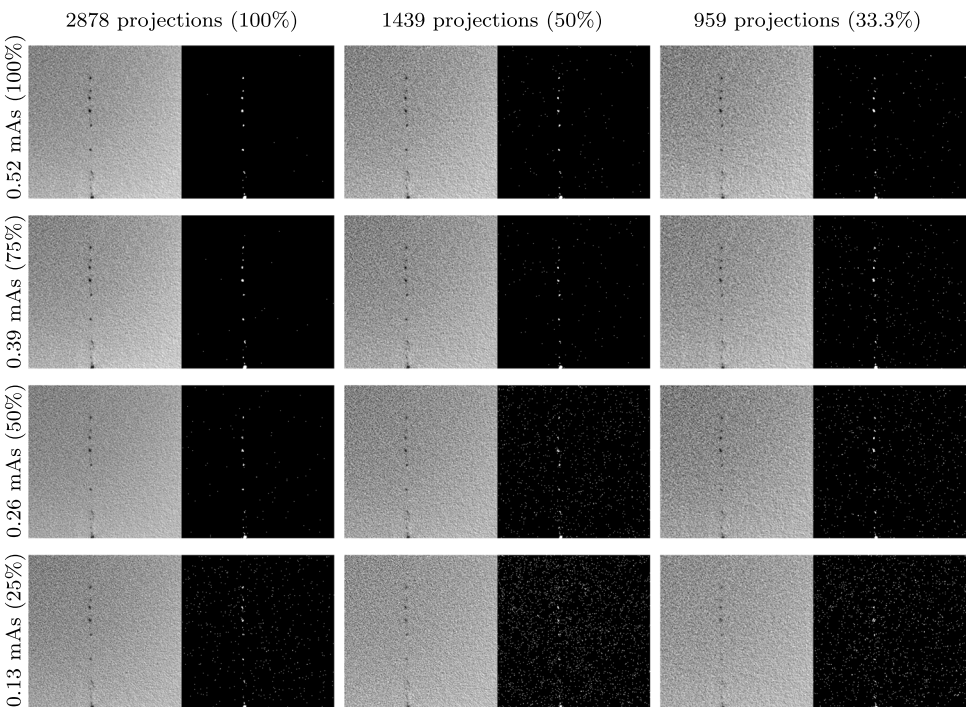


Fig. 10 A portion of a X-CT slice is shown in each row and column by modifying the number of X-ray projections and exposure of each X-ray projection. Each input slice is shown together with the label mask predicted by ceVAE (trained on the 1st fold). The degradation of the segmentation performance is noticeable from the raising number of voxels that are classified as pores (white colour in the predicted mask)

and vqVAE models, it would hinder supervised models' performance due to the absence of pores. Overall, the resulting VAE/ceVAE models exhibit resilience to lower image quality, unlike supervised models.

Training supervised models with labels derived from the best unsupervised model (ceVAE) enhances their performance (UNet++, AP from 0.583 ± 0.021 to 0.751 ± 0.030 on the challenging testset) but does not surpass that of the unsupervised model. The study confirms that unsupervised ceVAE, robustly captures the statistical properties of 3D patches compared to the supervised UNet family. This finding aligns with analogous results in anomaly detection in MRI images [59], endorsing unsupervised learning as a viable training paradigm for addressing anomaly segmentation in AM samples without the need for labelled data.

Looking ahead, future endeavours may involve developing efficient models capable of detecting pores from X-CT scans at a faster rate, with fewer projections or shorter scan times, in coherence with the future trends foreseen by Khosravani and Reinicke [60], which will expand our experiment Section 4.8. This would facilitate the use of X-CT in streamlined evaluations of entire sample batches. Furthermore, while our research primarily focuses on porosity analysis in the AM process, it opens avenues for broader anomaly detection applications, including identifying impurities, microstructural inhomogeneities, or alloying element loss due to vaporisation.

Appendix A: Classifier graphs for the voxel-wise segmentation task

The ROC and PR graphs of the voxel-wise segmentation results that were not shown in previous sections are reported here. In Fig. 11, there are the performance graphs of supervised and unsupervised models evaluated on the related validation dataset. The graphs are aligned with the findings discussed in Sections 5.2 and 5.4. In Fig. 12 are shown the performance of the unsupervised models only, since they show the segmentation scores of the post-processed output. The scores were obtained from the fold-wise performance on the related validation dataset, where is noticeable an increase of performance for VAE/ceVAE and a decrease for gmVAE/vqVAE if compared with Fig. 11 (right), in accordance with the findings in Section 5.4.

Appendix B: Cross-validation graphs for the FTL parameter search per each fold

For each of the 5 folds of the cross-validation, there is a total of 16 trainings for the α/β parameter, which are presented in Table 9. For the γ parameter, there is a total of 8 trainings per fold and the values of the Dice-Sørensen are shown in Table 10.

Appendix C: Random variable module: sigmoid activation function

In this section, we extend the discussion on random variables applied after the encoding layer of Autoencoder-based neural models, as presented in a previous article [38]. We maintain the assumptions established in that work, which include the absence of correlations between random variables. Furthermore, we leverage the ability to represent arbitrary probability distributions of real numbers through an expected value and a variance, a condition supported by the validity of the central limit theorem resulting from the summation of unrelated random variables.

Our focus here is to provide a means of obtaining the first two moments (expected value and variance) of a random vari-

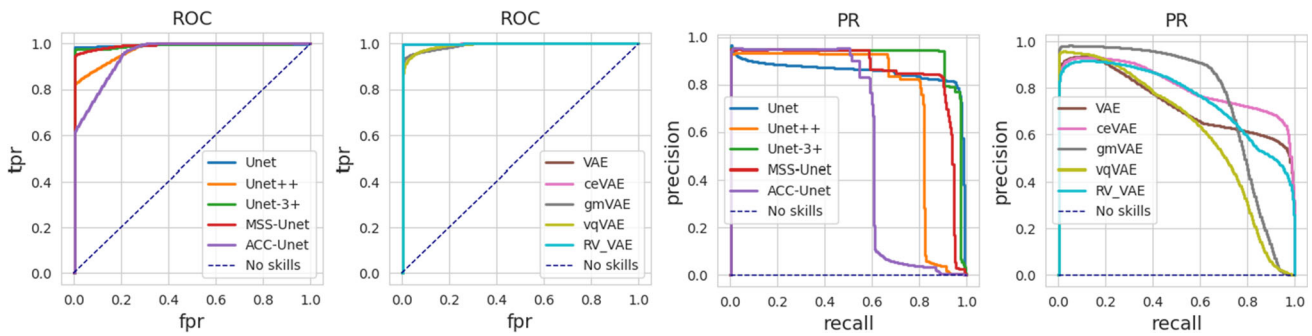


Fig. 11 Graph of the ROC and PR curves of cross-validated performance for all models. The graphs represent the median trend of the fold-wise performance on related validation dataset

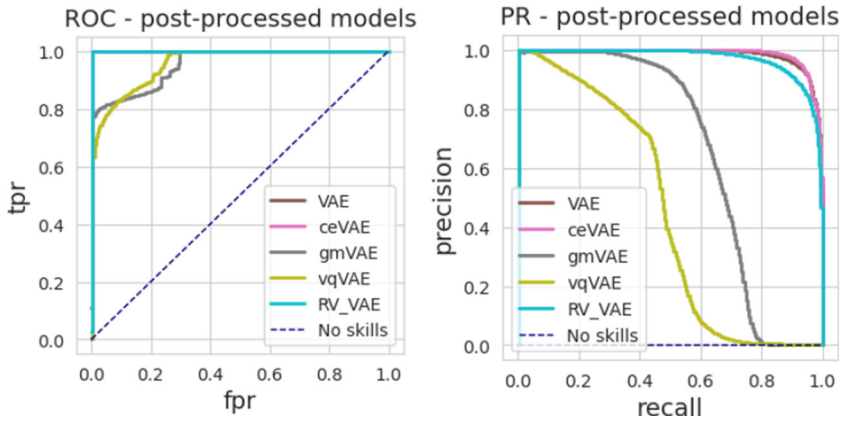


Fig. 12 Graph of the ROC and PR curves of cross-validated performance for the unsupervised models. The graphs represent the median trend of the fold-wise performance on the related validation dataset, when the output of the models is post processed

able Y resulting from the application of the sigmoid function S to its input random variable X.

Let us begin by defining the sigmoid function:

$$S(x) = \frac{1}{1+\exp(-x)} \quad , \tag{C.1}$$

alongside its first and second derivatives with respect to x

$$\dot{S}(x) = S(x)(1-S(x)) \quad \ddot{S}(x) = S(x)(1-S(x))(1-2S(x)) \, . \tag{C.2}$$

These derivatives will prove useful in deriving the expected value and variance of $Y = S(X)$, where X is considered to be a random variable.

For the calculation of $\mathbb{E}[S(X)]$, we employ a Taylor expansion centred at $X_0 = \mathbb{E}[X]$:

$$S(x) = S(\mathbb{E}[X]) + (X - \mathbb{E}[X])\dot{S}(\mathbb{E}[X]) + \frac{1}{2}(X - \mathbb{E}[X])^2 \ddot{S}(\mathbb{E}[X])^2 + \frac{1}{3!}(X - \mathbb{E}[X])^3 \dddot{S}(\mathbb{E}[X])^3 + \dots \quad . \tag{C.3}$$

Table 9 Fold-wise Dice-Sørensen score for the networks evaluated on the related validation dataset, depending on the α/β parameters of the FTL

Dice-Sørensen score - Fold 1

α \ β	0.1	0.37	0.63	0.9
0.9	0.82	0.81	0.97	0.97
0.63	0.86	0.97	0.97	0.96
0.37	0.96	0.97	0.96	0.83
0.1	0.97	0.95	0.94	0.84

Dice-Sørensen score - Fold 2

α \ β	0.1	0.37	0.63	0.9
0.9	0.76	0.63	0.61	0.60
0.63	0.69	0.61	0.60	0.59
0.37	0.64	0.60	0.58	0.57
0.1	0.63	0.54	0.53	0.57

Dice-Sørensen score - Fold 3

α \ β	0.1	0.37	0.63	0.9
0.9	0.94	0.88	0.87	0.88
0.63	0.92	0.88	0.87	0.85
0.37	0.70	0.87	0.85	0.83
0.1	0.87	0.82	0.80	0.79

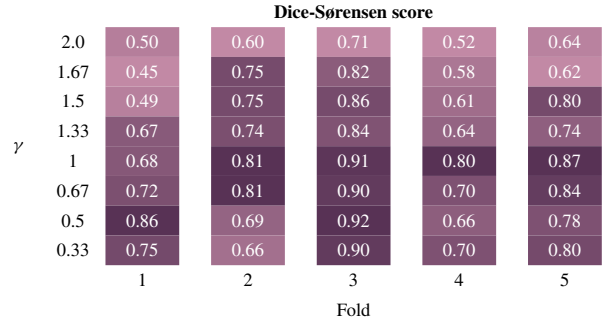
Dice-Sørensen score - Fold 4

α \ β	0.1	0.37	0.63	0.9
0.9	0.62	0.71	0.68	0.69
0.63	0.66	0.72	0.74	0.69
0.37	0.70	0.75	0.69	0.72
0.1	0.74	0.78	0.78	0.79

Dice-Sørensen score - Fold 5

α \ β	0.1	0.37	0.63	0.9
0.9	0.58	0.67	0.61	0.61
0.63	0.78	0.62	0.63	0.58
0.37	0.77	0.61	0.59	0.57
0.1	0.61	0.56	0.55	0.55

Table 10 Fold-wise Dice-Sørensen score for the networks evaluated on the related validation dataset, depending on the γ parameter of the FTL



From which we extract the expected value as

$$\mathbb{E}[S(x)] = \mathbb{E}[S(\mathbb{E}[X]) + (X - \mathbb{E}[X])\dot{S}(\mathbb{E}[X]) + \frac{1}{2}(X - \mathbb{E}[X])^2\ddot{S}(\mathbb{E}[X])^2 + \frac{1}{3!}(X - \mathbb{E}[X])^3\dddot{S}(\mathbb{E}[X])^3 + \ldots] \quad . \tag{C.4}$$

Given the assumption that the distribution of the random variable X behaves as a normal distribution, all odd central moments are expected to be null. This leads to a simplified formula for the expected value of $Y = S(X)$

$$\mathbb{E}[Y] = S(\mathbb{E}[X]) + \frac{1}{2}\ddot{S}(\mathbb{E}[X])\mathbb{V}\mathrm{ar}[X] + M_4 \quad , \tag{C.5}$$

where M_4 collects all the moments after the third and can be neglected under the assumption of smooth distribution. To calculate the expected variance of Y, we can utilise (C.3) and (C.5), so to obtain

$$\begin{aligned}\mathbb{V}\mathrm{ar}[Y] = \; & \mathbb{E}[Y^2] - \mathbb{E}[Y]^2 = \mathbb{E}[S^2(\mathbb{E}[X]) \\ & + 2(X - \mathbb{E}[X])S(\mathbb{E}[X])\dot{S}(\mathbb{E}[X]) \\ & + (X - \mathbb{E}[X])^2(\dot{S}^2(\mathbb{E}[X]) + S(\mathbb{E}[X])S''^2 \\ & \times (\mathbb{E}[X])) + (X - \mathbb{E}[X])^3(\dot{S}(\mathbb{E}[X])\ddot{S}(\mathbb{E}[X]) \\ & + \frac{2}{3!}S(\mathbb{E}[X])\dddot{S}(\mathbb{E}[X])) + R_4] \\ & - S^2(\mathbb{E}[X]) - \frac{1}{4}\ddot{S}^2(\mathbb{E}[X])\mathbb{V}\mathrm{ar}^2[X] \\ & - S(\mathbb{E}[X])\ddot{S}(\mathbb{E}[X])\mathbb{V}\mathrm{ar}[X] - \tilde{M}_4\,,\end{aligned} \tag{C.6}$$

with $\tilde{M}_4$ being analogous to M_4 in (C.5) and R_4 collecting all the central differences above the third exponent. By discarding all moments above the third, a compact approximation for the variance of Y is given by

$$\mathbb{V}\mathrm{ar}[Y] \approx \dot{S}^2(\mathbb{E}[X])\mathbb{V}\mathrm{ar}[X] - \frac{1}{4}\ddot{S}^2(\mathbb{E}[X])\mathbb{V}\mathrm{ar}^2[X]\,. \tag{C.7}$$

Acknowledgements This study is financially supported by the VLAIO/imec-ICON project Multiplicity, the Research Foundation Flanders (FWO, SBO grant no. S007219N) and the Flemish Government under the "Onderzoeksprogramma Artificiele Intelligentie (AI) Vlaanderen" programme.

Author Contributions Domenico Iuso devised the project, the main conceptual ideas and worked out all of the technical details. Soumick Chatterjee contributed to the conceptual ideas and the design of the research. Sven Cornelissen and Dries Verhees contributed with the design and printing of AM samples. Jan De Beenhouwer and Jan Sijbers supervised the project. All authors discussed the results, commented on the manuscript and approved the final manuscript.

Data Availability The data presented in this study are available upon reasonable request from the corresponding author.

Declarations

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Sachs E, Cima M, Cornie J (1990) Three-Dimensional Printing: Rapid Tooling and Prototypes Directly from a CAD Model. CIRP Ann 39(1):201–204. ISSN 0007-8506
2. Kruth J-P, Levy G, Klocke F, Childs THC (2007) Consolidation phenomena in laser and powder-bed based layered manufacturing. CIRP Ann 56(2):730–759
3. Tang M, Pistorius PC (2017) Oxides, porosity and fatigue performance of AlSi10Mg parts produced by selective laser melting. Int J Fatigue 94:192–201
4. Zhang B, Li Y, Bai Q (2017) Defect Formation Mechanisms in Selective Laser Melting: A Review. Chin J Mech Eng 30(3):515–527
5. Kumar AY, Wang J, Bai Y, Huxtable ST, Williams CB (2019) Impacts of process-induced porosity on material properties of copper made by binder jetting additive manufacturing. Mater Des 182:108001
6. Ziółkowski G, Chlebus E, Szymczyk P, Kurzac J (2014) Application of X-ray CT method for discontinuity and porosity detection in 316L stainless steel parts produced with SLM technology. Arch Civ Mech Eng 14:608–614
7. Sarkon GK, Safaei B, Kenevisi MS, Arman S, Zeeshan Q (2022) State-of-the-Art Review of Machine Learning Applications in Additive Manufacturing; from Design to Manufacturing and Property Control. Arch Comput Methods Eng 29(7):5663–5721

8. Thompson A, Maskery I, Leach RK (2016) X-ray computed tomography for additive manufacturing: A review. Meas Sci Technol 27(7):072001
9. Leary M, Mazur M, Elambasseril J, McMillan M, Chirent T, Sun Y, Qian M, Easton M, Brandt M (2016) Selective laser melting (SLM) of AlSi12Mg lattice structures. Mater Des 98:344–357
10. Salarian M, Toyserkani E (2018) The use of nano-computed tomography (nano-CT) in non-destructive testing of metallic parts made by laser powder-bed fusion additive manufacturing. Int J Adv Manuf Technol 98
11. Sanaei N, Fatemi A (2021) Defects in additive manufactured metals and their effect on fatigue performance: A state-of-the-art review. Prog Mater Sci 117:100724
12. Vandecasteele M, Heylen R, Iuso D, Thanki A, Philips W, Witvrouw A, Verhees D, Booth BG (2023) Towards material and process agnostic features for the classification of pore types in metal additive manufacturing. Mater Des 111757
13. Wong VWH, Ferguson M, Law KH, Lee Y-TT, Witherell P (2021) Automatic volumetric segmentation of additive manufacturing defects with 3D U-Net. arXiv:2101.08993
14. Bihani A, Daigle H, Santos JE, Landry C, Prodanović M, Milliken K (2022) MudrockNet: Semantic segmentation of mudrock SEM images through deep learning. Comput Geosci 158:104952
15. Kim J-H, Won-Jung O, Lee C-M, Kim D-H (2022) Achieving optimal process design for minimizing porosity in additive manufacturing of Inconel 718 using a deep learning-based pore detection approach. Int J Adv Manuf Technol 121(3–4):2115–2134
16. Yang J, Ruijie X, Qi Z, Shi Y (2022) Visual Anomaly Detection for Images: A Systematic Survey. Proc Comput Sci 199:471–478
17. Bouget D, Jørgensen A, Kiss G, Leira HO, Langø T (2019) Semantic segmentation and detection of mediastinal lymph nodes and anatomical structures in CT data for lung cancer staging. Int J CARS 14:977–986
18. Rushood IA, Alqahtani N, Da Wang Y, Shabaninejad M, Armstrong R, Mostaghimi P (2020) Segmentation of X-ray images of rocks using deep learning. In: SPE annual technical conference and exhibition. OnePetro
19. Fend C, Moghiseh A, Redenbach C, Schladitz K (2021) Reconstruction of highly porous structures from FIB-SEM using a deep neural network trained on synthetic images. J Microsc 281(1):16–27
20. Wang H, Dalton L, Fan M, Guo R, McClure J, Crandall D, Chen C (2022) Deep-learning-based workflow for boundary and small target segmentation in digital rock images using UNet++ and IK-EBM. J Pet Sci Eng 215:110596
21. Mehta M, Shao C (2022) Federated learning-based semantic segmentation for pixel-wise defect detection in additive manufacturing. J Manuf Syst 64:197–210
22. Wang R, Cheung CF (2022) CenterNet-based defect detection for additive manufacturing. Expert Syst Appl 188:116000
23. Maskery I, Aboulkhair NT, Corfield MR, Tuck C, Clare AT, Leach RK, Wildman RD, Ashcroft IA, Hague RJM (2016) Quantification and characterisation of porosity in selectively laser melted Al-Si10-Mg using X-ray computed tomography. Mater Charact 111:193–204
24. Li R, Wang X, Huang G, Yang W, Zhang K, Gu X, Tran SN, Garg S, Alty J, Bai Q (2022) A comprehensive review on deep supervision: theories and applications. arXiv:2207.02376
25. Bria A, Marrocco C, Tortorella F (2020) Addressing class imbalance in deep learning for small lesion detection on medical images. Comput Biol Med 120:103735
26. Guo C, Zhou J, Chen H, Ying N, Zhang J, Zhou D (2020) Variational Autoencoder with Optimizing Gaussian Mixture Model Priors. IEEE Access 8:43992–44005
27. Zimmerer D, Kohl SAA, Petersen J, Isensee F, Maier-Hein KH (2018) Context-encoding variational autoencoder for unsupervised anomaly detection. arXiv:1812.05941
28. Baur C, Denner S, Wiestler B, Navab N, Albarqouni S (2021) Autoencoders for Unsupervised Anomaly Segmentation in Brain MR Images: A Comparative Study. Med Image Anal 69:101952
29. Zhou Z, Siddiquee Md MR, Tajbakhsh N, Liang J (2018) UNet++: A Nested U-Net Architecture for Medical Image Segmentation. In: Deep learning in medical image analysis and multimodal learning for clinical decision support, pp 3–11. Springer
30. Huang H, Lin L, Tong R, Hu H, Zhang Q, Iwamoto Y, Han X, Chen Y-W, Wu J (2020) Unet 3+: A full-scale connected UNET for medical image segmentation. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 1055–1059. IEEE
31. Zhao W, Jiang D, Queralta JP, Westerlund T (2020) MSS U-Net: 3D segmentation of kidneys and tumors from CT images with a multi-scale supervised U-Net. Inform Med Unlocked 19:100357
32. Ibtehaz N, Kihara D (2023) ACC-UNet: A completely convolutional UNet model for the 2020s. In: International conference on medical image computing and computer-assisted intervention, pp 692–702. Springer
33. Ronneberger O, Fischer P, Brox T (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: International conference on medical image computing and computer-assisted intervention, pp 234–241. Springer
34. Abraham N, Khan NM (2019) A Novel Focal Tversky loss function with improved Attention U-Net for lesion segmentation. In: 2019 IEEE 16th international symposium on biomedical imaging (ISBI 2019), pp 683–687. IEEE
35. Kingma DP, Welling M (2013) Auto-encoding variational Bayes. arXiv:1312.6114
36. Dilokthanakul N, Mediano PAM, Garnelo M, Lee MCH, Salimbeni H, Arulkumaran K, Shanahan M (2016) Deep unsupervised clustering with Gaussian mixture variational autoencoders. arXiv:1611.02648
37. Van Den Oord A, Vinyals O et al (2017) Neural Discrete Representation Learning. Advances in neural information processing systems, p 30
38. Nicodemou VC, Oikonomidis I, Argyros A (2023) RV-VAE: Integrating Random Variable Algebra into Variational Autoencoders. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 196–205
39. Meiners W, Wissenbach K, Gasser A (1998) Shaped body especially prototype or replacement part production. DE Patent, 19
40. Abe F, Osakada K, Shiomi M, Uematsu K, Matsumoto M (2001) The manufacturing of hard tools from metallic powders by selective laser melting. J Mater Process Technol 111(1–3):210–213
41. Booth BG, Heylen R, Nourazar M, Verhees D, Philips W, Bey-Temsamani A (2022) Encoding Stability into Laser Powder Bed Fusion Monitoring Using Temporal Features and Pore Density Modelling. Sensors 22(10):3740
42. De Samber B, Renders J, Elberfeld T, Maris Y, Sanctorum J, Six N, Liang Z, De Beenhouwer J, Sijbers J (2021) FleXCT: a flexible X-ray CT scanner with 10 degrees of freedom. Opt Express 29(3):3438–3457
43. Feldkamp LA, Davis LC, Kress JW (1984) Practical Cone-Beam Algorithm. Josa a 1(6):612–619
44. Du Plessis A, Sperling P, Beerlink A, Tshabalala L, Hoosain S, Mathe N, Le Roux SG (2018) Standard method for microCT-based additive manufacturing quality control 1: Porosity analysis. MethodsX 5:1102–1110
45. Kim FH, Moylan SP, Garboczi EJ, Slotwinski JA (2017) Investigation of pore structure in cobalt chrome additively manufactured parts using X-ray computed tomography and three-dimensional image analysis. Addit Manuf 17:23–38

46. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, Lin S, Guo B (2021) Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 10012–10022
47. Chen W, Xu H, Li Z, Pei D, Chen J, Qiao H, Feng Y, Wang Z (2019) Unsupervised anomaly detection for intricate KPIs via adversarial training of VAE. In: IEEE INFOCOM 2019-IEEE conference on computer communications, pp 1891–1899. IEEE
48. Lin S, Clark R, Birke R, Schönborn S, Trigoni N, Roberts S (2020) Anomaly Detection for Time Series Using VAE-LSTM Hybrid Model. In: ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP), pp 4322–4326. IEEE
49. Chatterjee S, Sciarra A, Dünnwald M, Agrawal S, Tummala P, Setlur D, Kalra A, Jauhari A, Oeltze-Jafra S, Speck O et al (2021) Unsupervised reconstruction based anomaly detection using a Variational Auto Encoder. In: 2021 ISMRM & SMRT Annual Meeting & Exhibition, p 2399
50. Song H, Kim M, Park D, Shin Y, Lee J-G (2022) Learning from Noisy Labels with Deep Neural Networks: A Survey. IEEE Trans Neural Netw Learn Syst
51. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L et al (2019) PyTorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, p 32
52. Falcon W, Borovec J, Wälchli A, Eggert N, Schock J, Jordan J, Skafte N, Bereznyuk V, Harris E, Murrell T et al (2020) PyTorchLightning/pytorch-lightning: 0.7. 6 release. Zenodo: Geneva, Switzerland
53. Vingelmann P (2020) NVIDIA and FH Fitzek. Cuda, release: 10.2. 89, 2020
54. Pérez-García F, Sparks R, Ourselin S (2021) TorchIO: a Python library for efficient loading, preprocessing, augmentation and patch-based sampling of medical images in deep learning. Comput Methods Programs Biomed 208:106236
55. Iuso D, Chatterjee S, Heylen R, Cornelissen S, De Beenhouwer J, Sijbers J (2022) Evaluation of deeply supervised neural networks for 3D pore segmentation in additive manufacturing. In: Developments in X-Ray Tomography XIV, vol. 12242, 122421K. SPIE
56. Hanczar B, Hua J, Sima C, Weinstein J, Bittner M, Dougherty ER (2010) Small-sample precision of ROC-related estimates. Bioinformatics 26(6):822–830
57. Saito T, Rehmsmeier M (2015) The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. PloS One 10(3):e0118432
58. Liu B, Wang M, Foroosh H, Tappen M, Pensky M (2015) Sparse convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, p 806–814
59. Chatterjee S, Sciarra A, Dünnwald M, Tummala P, Agrawal SK, Jauhari A, Kalra A, Oeltze-Jafra S, Speck O, Nürnberger A (2022) StRegA: Unsupervised Anomaly Detection in Brain MRIs using a Compact Context-encoding Variational Autoencoder. Comput Biol Med 149:106093
60. Khosravani Md R, Reinicke T (2020) On the Use of X-ray Computed Tomography in Assessment of 3D-Printed Components. J Nondestruct Eval 39:1–17

Authors and Affiliations

Domenico Iuso[1,6] · Soumick Chatterjee[2,3] · Sven Cornelissen[4] · Dries Verhees[5] · Jan De Beenhouwer[1,6] · Jan Sijbers[1,6]

✉ Domenico Iuso
Domenico.Iuso@uantwerpen.be; snipdomenico@gmail.com

[1] imec-Vision Lab, Department of Physics, University of Antwerp, Antwerp 2610, Belgium

[2] Faculty of Computer Science, Otto von Guericke University, Magdeburg 39106, Germany

[3] Genomics Research Centre, Human Technopole, Milan 20157, Italy

[4] Materialise NV., Technologielaan 15, Leuven 3001, Belgium

[5] Flanders Make vzw., Oude Diestersebaan 133, Lommel 3920, Belgium

[6] DynXlab: Center for 4D Quantitative X-ray Imaging and Analysis, Antwerp 2610, Belgium