Normalizing Flows with Task-specific Pre-training for Unsupervised Anomaly Detection on Engineering Structures

Brice Marc, Philippe Foucher, Florence Forbes, Pierre Charbonnier

Normalizing Flows with Task-specific Pre-training for Unsupervised Anomaly Detection on Engineering Structures

Brice MARC
ENDSUM / STATIFY
CEREMA / INRIA
Strasbourg / Grenoble, France
brice.marc@cerema.fr

Philippe FOUCHER
ENDSUM
CEREMA
Strasbourg, France
philippe.foucher@cerema.fr

Florence FORBES
STATIFY
INRIA
Grenoble, France
florence.forbes@inria.fr

Pierre CHARBONNIER
ENDSUM
CEREMA
Strasbourg, France
pierre.charbonnier@cerema.fr

***Abstract*—Automatic anomaly detection on engineering structures is often carried out using supervised models, raising the issue of anomalous images acquisition and annotation. Unsupervised methods like normalizing flows achieve excellent results while trained with defect-free images only. However, normalizing flows methods, such as MSFlow, are generally applied on features extracted by an encoder pre-trained on datasets that may not be related to engineering structures images. Therefore, we investigate the possibility to derive more discriminative features with an additional fine-tuning of the feature extractor on images with synthetic anomalies. We consider two types of such anomalies and demonstrate their efficiency with MSFlow on the MVTec (Wood/Tile) and Crack500 datasets, with significantly improved predictions. Interestingly, both tasks produce similar results suggesting that pre-training is mainly improved by the healthy part of images and not very sensitive to anomaly realism. Additionally, when comparing our fine-tuned MSFlow with a reference supervised model, CT-CrackSeg, on the Crack500 dataset, we observe similar qualitative behaviours. This open a promising direction towards annotation-free, more scalable alternatives, in particular for anomaly detection in engineering structure applications.**

***Index Terms*—Anomaly detection, unsupervised learning, normalizing flows**

I. Introduction

Structure monitoring is a crucial topic in civil engineering to prevent damages emerging and to guarantee safety and long-term durability. Examining roads, bridges or tunnels to locate disorders is a tedious task. Therefore, developing automatic anomaly detection methods is an active research field. Many works propose models to inspect structure images [1], [2]. However, most of them are based on supervised learning, requiring segmented images with anomalies. In addition to the cost of producing such annotations, defects on structures can take various forms. They are present only in a small proportion and can differ a lot from a structure to an other. Developing a supervised model for a specific structure is thus arduous.

In contrast, normal images, i.e. images without anomalies, can be acquired in larger number on any structures and do not necessitate annotations. Unsupervised anomaly detection (UAD) methods may then offer interesting alternatives to deal with the difficulty to gather annotated anomalies.

Brice Marc's PhD research is funded by Cerema (Centre for Studies and Expertise on Risks, the Environment, Mobility and Urban Planning) as part of the Cerema-Inria ROAD-AI project.

Unsupervised anomaly detection gathers methods which only need anomaly-free images for training. As proposed in [3], UAD approaches can be grouped into four families: methods based on Reconstruction, Data augmentation, Representation, and Normalizing Flows (NF).

Reconstruction-based methods use generative models to project an image into a latent space and reconstruct this projection [4], [5]. UAD is performed by analysing the residuals between an input image and its reconstruction. As the models are trained only with normal images, it is expected that they fail to reconstruct anomalies. However, these models have often enough generalisation power to also correctly reconstruct anomalous patterns preventing then their detection.

Data augmentation-based methods are built on supervised models trained to detect synthetic anomalies placed on normal images. Synthetic anomalies can be generated with various methods [6], [7], [8]. While these methods show encouraging results on anomalies which are similar to the synthetic ones, simulating realistic anomalies is difficult in general and models struggle in detecting real world defects.

Representation-based methods train feature extractors to generate compact representations of normal images [9], [10]. During inference, a distance between the image extracted embedded vectors and the class center determined during training is computed to estimate the anomaly score.

NF models aim to transform a complex probability distribution into a simpler one [11]. NF approaches, integrated as modules in deep learning architectures, have recently given excellent results in UAD with only few drawbacks. NF methods can also be considered as a subfamily of representation based methods. A previous study reported in [12] showed that NF outperform other unsupervised methods on tunnel images. Therefore, in this paper, we focus on this family of approaches.

Most recent NF models designed for UAD like [13] or [14] perform data projection from features extracted by a pre-

trained encoder. However, while these encoders succeed in extracting common features, they may not be perfectly suited to disparate datasets, such as industrial objects or engineering structures images. The work in [15] implements a NF model named CL-Flow. It also compares several pre-training feature extraction tasks and shows that a fine-tuned extractor improves overall model performances. Building on this idea, we propose two tasks using two types of synthetic anomalies to enhance MSFlow feature extraction.

To summarize, our main contributions are:

- To generate synthetic anomalies for the feature extractor fine-tuning tasks, we apply CutPaste [6] method and propose our realistic anomaly generation method based on Poisson interpolation [16].
- We propose a U-Net [17] like structure, incorporating MSFlow [14] feature extractor, to perform pre-training tasks and refine feature extraction, with a view to improve model overall performances.
- We demonstrate the positive impact of both pre-training tasks on MSFlow performances on MVTec benchmark [18] and on Crack500 [19] dataset.
- We compare our MSFlow pipeline and CT-CrackSeg [20] detection performances on Crack500 to show that NF unsupervised methods are a promising alternative to supervised methods on engineering structure images.

II. Proposed Method

A. MSFlow anomaly detection model

MSFlow [14], one of the latest state-of-the-art normalizing flow model, is employed to perform unsupervised anomaly detection. Figure 1 depicts its architecture. In a nutshell, MSFlow first extracts features using WideResnet50 encoder [21] at three different stages. Each stage of features is then separately transformed by asymmetrical parallel flows, using Affine Coupling Layers (ACL) [22] integrating 2D convolutions. The three flow outputs are finally aggregated by a fusion module to provide the resulting distribution while benefiting from multi-scale information. ACL are reversible structures with efficient forward and reverse computation process allowing to apply any complex functions. After the image feature encoding step, the flow modules learn a bijective transformation from the complex feature distribution to a simple Gaussian distribution. The model is trained with only anomaly-free images to learn their feature distribution. During the inference step, pixels with low likelihood are considered as anomalies. It may be noticed that the extractor is frozen and is not updated during training.

B. Self-supervised tasks for feature extraction fine-tuning

We propose two methods that both involve a supervised synthetic anomaly segmentation task to fine-tune the MSFlow WideResNet50 feature extractor, before actual training. As depicted in figure 2, the extractor is integrated in a three stage U-Net [17] structure as the encoder part. The decoder part is composed of 2D transposed convolutions (indicated as *convtranspose* layers in the figure). Connections between the encoder and the decoder are formed from the three stages

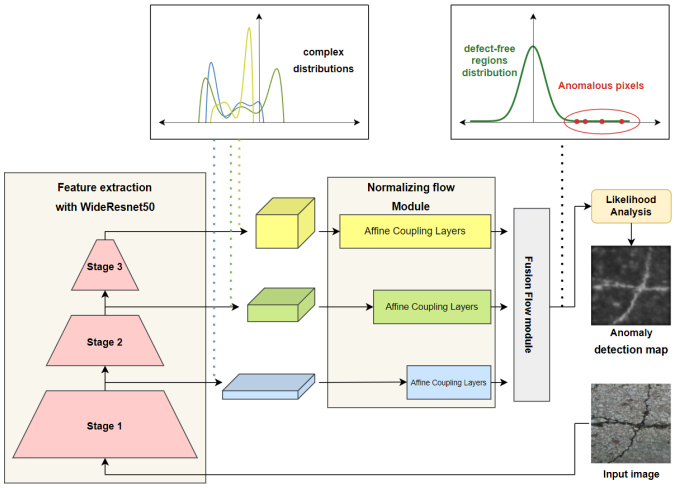


Fig. 1: Synoptic MSFlow architecture used for unsupervised anomaly detection (for more details, see [14]). The WideResNet50 [21] pre-trained feature extractor is frozen while the flow modules are updated during training.

feature maps extracted. The model is trained by minimizing the binary cross entropy loss between U-Net predictions and binary masks indicating anomalies.

Two types of synthetic anomalies are generated to perform the two pre-training tasks.

1) CutPaste-generated anomalies: To develop a self-supervised model, CutPaste algorithm [6] propose to generate local irregular patterns and train the network identifying these local anomalies. The model is then expected to generalize the detection process to real anomalies during real test inference. To synthesise an abnormal pattern on an image, a small rectangle selected at random from this normal image is copied and pasted at a random position on the same image.

The process is applied to the training images to generate synthetic anomalies. The pre-training task consists in having the U-Net architecture to detect the rectangular patches that have been pasted into the images.

2) Poisson Interpolated anomalies: While CutPaste anomalies introduce irregularities in the normal images patterns, which might be satisfactory for a pre-training task, their appearance and shape are still distant from real world anomalies. Moreover, the boundaries of the patches might cause the network to learn shortcuts. Some other self-supervised models like DRAEM [7] developed more realistic anomaly synthesis processes to improve defect detection. To assess the potential enhancements of using realistic synthetic anomalies in extractor fine-tuning, we propose a method to generate real-looking defects.

Realistic anomalies are synthesised with defects from another dataset using seamless cloning Poisson interpolation [16]. Seamless cloning is an interpolation method allowing to interpolate a part of a source image into a target image without any boundary discontinuity. Considering a source image and a target image given by $f*$, seamless cloning involves finding

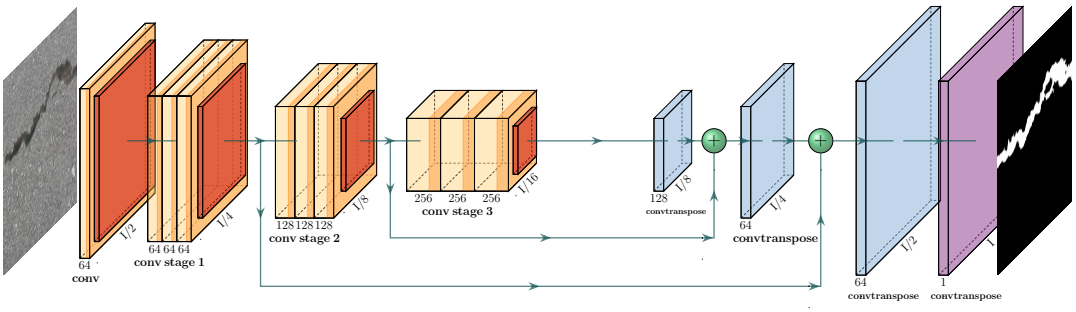


Fig. 2: Structure used to pre-train the feature extractor with synthetic anomalies. The yellow part is the WideResNet50 [21] feature extractor first stages and the blue part is the U-Net [17] decoder.

an interpolant f on a defined region Ω. The interpolant is the unique solution of the problem formulated by both equations (1) and (2).

$$f = \underset{f}{argmin} \iint_{\Omega} |\nabla f - v|^2 \quad \text{with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{1}$$

$$\Delta f = div\, v \quad \text{over } \Omega \tag{2}$$

where v, the guidance field, can be chosen among the source image gradient or a mix between the source and target gradients. As the first option is better for opaque source regions, it will be the option retained to synthesise anomalies. To find the interpolant, a discrete algorithm is described in [16].

Poisson interpolation has already been employed by UAD models to synthesise anomalies [8], [23]. However, these models take training images as the source image, generating anomalies with the superposition of another normal region on the target image, which can still be far from real anomalies. To synthesise realistic anomalies on a training image, a randomly chosen test image from another dataset fulfill the role of the source image and the regions interpolated are its anomalous regions. Code from https://github.com/bchao1/fast-poisson-image-editing is used to fulfill seamless cloning.

C. Supervised detection model

To compare the performance of MSFlow in detecting anomalies in engineering structure images with supervised models, CT-CrackSeg [20] is trained and tested. CT-CrackSeg is a convolutional-transformer crack detection model based on an encoder-decoder structure which is similar to U-Net model [17]. The model is composed of three novel components. Dilated Residual Blocks with hybrid dilated convolutions replace usual convolutional layers to improve receptive field and thin object detection. Boundary Awareness Module employs deformable convolutions to locate accurately crack boundaries. Mobile ViT Block, a lightweight transformer encoder, encodes global information and generates additional feature maps to improve model's global awareness.

III. Experiments

A. Datasets

1) MVTec: MVTec [18] is a benchmark dataset built for unsupervised anomaly detection on industrial images. Training subsets are exclusively composed of normal images without anomalies. Testing subsets contain both normal images and labelled images with anomalies. The images are split into 15 categories depending on the object or the texture represented. 5 categories correspond to texture images, which can be considered as similar to engineering structure images. Among these textures, the model is only used on the *wood* and *tile* categories, which are the closest to our field of application, due to their irregular patterns.

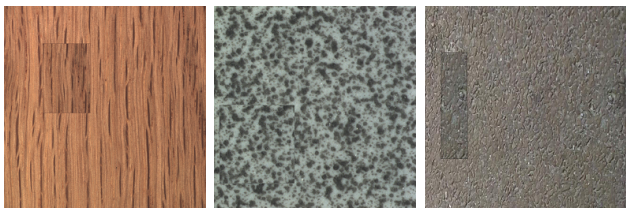
To synthesise Poisson anomalies for the pre-training task, defects from the hazelnut category are interpolated on wood training images and anomalous regions from the wood category are used to generate anomalies on tile images.

2) Crack500: To measure the efficiency of the model on real images from our application field, experiments are carried out on Crack500 [19], a dataset gathering 500 labelled images of pavement cracks. In our experiments, only the 250 images from Crack500 training subset are used. Note that the dataset only contains images with anomalies, as it was designed for supervised detection. Therefore, subsets of anomaly-free images need to be generated in order to implement our unsupervised approach. To do this, every image is divided into 9 sub-images that contain cracks or not. From the 250 images in the training subset, the first 150 are divided into 613 normal sub-images and 744 crack sub-images. The unsupervised model is trained with the normal images while the supervised model is trained with all these images. To synthesise Poisson anomalies for the pre-training task, anomalies from Deepcrack [24], another road crack dataset, are interpolated. The 100 remaining images are divided into 321 normal images and 527 crack images to generate our test set.

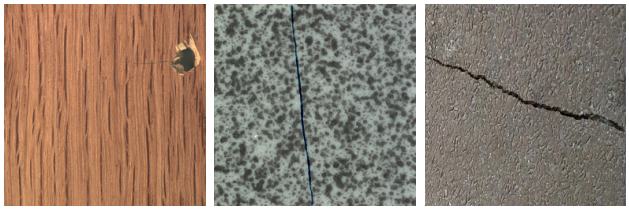
Examples of both synthetic anomalies for MVTec and Crack500 datasets are shown in figure 3.

B. Experimental settings

1) Implementation details: In the unsupervised process, all images are resized to 512×512 pixels at network input.



CutPaste anomalies [6]



Poisson interpolated anomalies [16]

Fig. 3: Examples of anomalies used for pre-training tasks with MVTec (textures *Wood* and *Tile*) and Crack500 datasets.

Following MSFlow original implementation [14], the flow module is trained with Adam optimizer and an initial learning rate of $1e^{-4}$ (resp. $1e^{-6}$) on MVTec (resp. Crack500) dataset. The learning rate is reduced by a factor of 3 at 70% and 90% of the training process. The model is trained during 100 epochs with a batch size of 8. The same implementation incorporating the U-Net like structure is used during 400 epochs for the feature extractor pre-training tasks. The full model is then regularly trained with the pre-trained extractor. The learning rate is initially $1e^{-6}$ and then reduced in the same way as for the implementation without the pre-training task.

The crack detection supervised model CT-CrackSeg [20] is trained using the official code while changing only the datasets. The network is trained for 100 epochs using Adam optimizer with a batch size of 2 and an initial learning rate of $1e^{-4}$. Images are resized to 256×256 pixels.

The experiments are run on a Tesla V100 GPU card.

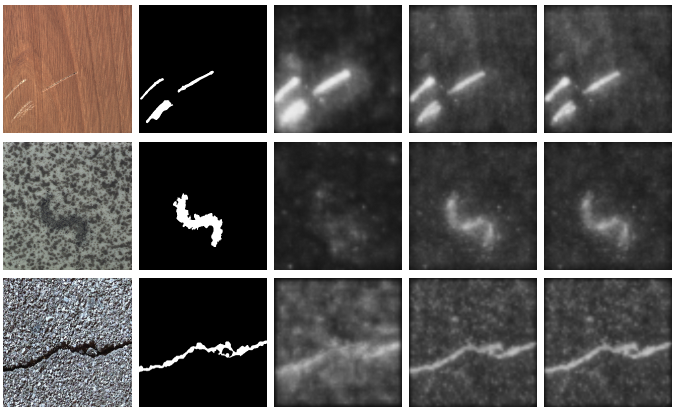
2) Evaluation metrics: For every model, the pixel-wise anomaly detection performance is measured on each dataset using the area under the receiver operator curve (AUROC) and the area under the per-region-overlap curve (AUPRO). AUPRO gives the same importance to every connected component of an anomaly, whereas the AUROC metric is more impacted by wide anomalies.

C. Influence of our proposed pre-training tasks

Quantitative performance results of the different pre-training tasks are gathered in table I. Different performance gains, with and without a pre-training task, can be observed on the three test datasets. For the MVtech wood database, the increase in the AUROC score is fairly small (from 97.08 to 97.40% at best). Performance are even slightly worse with a pre-training task if the AUPRO score is considered. This means that small anomalies are better detected without the pre-training task. In contrast, for the experiments on the MVTech Tile dataset, the use of a pre-training task has a significant impact on

Dataset	Without any task	CutPaste anomaly task (Ours)	Poisson anomaly task (Ours)
Wood	97.08 / **98.40**	97.29 / 98.12	**97.40** / 98.34
Tile	96.25 / 95.36	**98.75** / **98.81**	98.72 / 98.77
Crack500	75.68 / 60.39	**88.22** / **70.81**	88.17 / 70.49

TABLE I: MSFlow [14] pixel-wise anomaly detection performance (AUROC / AUPRO in %) of with and without pre-training tasks, on MVTec Wood, Tile and Crack500 datasets.



Input | Label | Without any task | CutPaste task (Ours) | Poisson task (Ours)

Fig. 4: Visualization of the impact of our MSFlow pre-training tasks on MVTec Wood, Tile and Crack500 images.

performance for both the AUROC (up from 96.25 to 98.75% at best) and AUPRO scores (up from 95.36 to 98.81% at best). Very significant performance enhancements can be observed using pre-training tasks on Crack500 datasets with gains of more than 10 points for the AUROC and AUPRO scores.

This quantitative analysis can be supported by the examples of predicted anomalies depicted in figure 4. Using pre-training tasks allow the model to detect anomalous areas that were not detected without pre-training task. It appears clearer in the tile image example (fig. 4, 2nd row) where the s-shaped anomaly is very poorly detected without pre-training task (3rd column) whereas detection is better by using models with pre-training tasks (4th and 5th columns). The same effects of pre-training can be observed on the road crack image (fig. 4, 3rd row), with a rather blurred detection of the crack using the model without pre-training (3rd column), compared with the noticeably more distinct detections in the images of the last two columns (i.e. with pre-training tasks). The model performs better on MVTec than on Crack500, as illustrated by a ten points gap between AUROC scores. This difference might be linked with the image acquisition settings disparity.

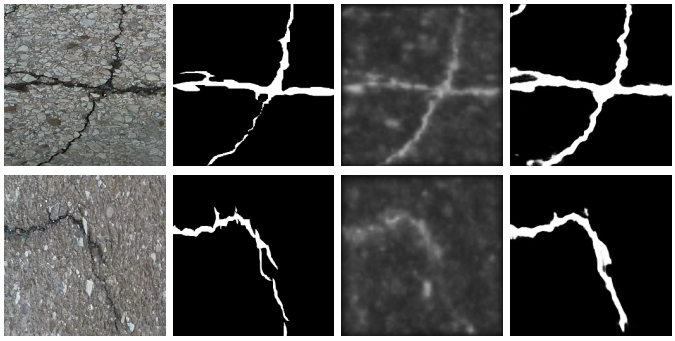
Qualitative and quantitative results obtained with the two pre-training tasks are very similar on all datasets. Therefore, the type of anomaly used during the task seem not to be relevant and only the presence of normal regions matters.

D. Comparison between supervised and unsupervised models

Quantitative performance comparison between CT-CrackSeg supervised model and MSFlow pre-trained with

Metrics	MSFlow with Poisson anomaly pre-training (Ours)	CT-CrackSeg
AUROC	88.17	**96.36**
AUPRO	70.49	**83.84**

TABLE II: Comparison of pixel wise anomaly detection performance (AUROC / AUPRO in %) between supervised CT-CrackSeg [20] and unsupervised MSFlow [14] model with our Poisson pre-training task on Crack500 [19] dataset.



Input Ground-truth Unsupervised Supervised

Fig. 5: Visualization of the comparison between supervised CT-CrackSeg [20] and unsupervised MSFlow [14] model with our Poisson pre-training task on Crack500 [19] dataset.

Poisson anomalies are presented in table II. Examples of predicted anomaly maps are depicted in figure 5.

As expected, the supervised model performs better than the unsupervised model. The anomaly detection maps computed by CT-CrackSeg are close to binary predictions and capture every parts of the cracks. The unsupervised model detects most of the large cracks but fails to identify thin cracks, as shown by the considerable difference between its AUROC and AUPRO scores. It also suffers from false positive element predictions, like the large gravels which can be observed in the examples.

IV. Conclusion

In this study, we proposed two feature extractor fine-tuning tasks and showed their positive impact on the performance of the unsupervised MSFlow model, on both MVTec and Crack500 datasets. Two types of anomalies were used to perform pre-training: CutPaste-generated anomalies and more realistic Poisson interpolated anomalies. Both tasks lead to similar results, raising questions about the necessity of the use of anomalies for these tasks. The feature distribution variations induced by the tasks could be studied to better understand their impact. The tasks should also be deployed on other NF methods to confirm their benefits.

Moreover, we compared MSFlow with CT-CrackSeg, a supervised crack detection model, on Crack500 dataset. Despite failing to detect thin cracks and being less precise than CT-CrackSeg, the unsupervised model succeeded in detecting most of the large cracks and provided encouraging results. Many tricky non-defect elements of the images are classified as anomalies by MSFlow model. Therefore, as these unusual features are often present in engineering structures, it is worth investigating methods to reduce false positive detections in order to improve model performance.

As Crack500 only represents road images, a further work would be to apply the model on tunnel or bridge lining images.

References

[1] K. Luo, X. Kong, J. Zhang, J. Hu, J. Li, and H. Tang. Computer vision-based bridge inspection and monitoring: A review. *Sensors*, 2023.

[2] H. Munawar, A. Hammad, A. Haddad, C. Soares, and S. Waller. Image-Based Crack Detection Methods: A Review. *Infrastructures*, 2021.

[3] Y. Cui, Z. Liu, and S. Lian. A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images. *IEEE Access*, 2023.

[4] A. Kascenas, N. Pugeault, and A. Q. O'Neil. Denoising Autoencoders for Unsupervised Anomaly Detection in Brain MRI. In *International Conf. on Medical Imaging with Deep Learning*, 2022.

[5] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth. f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. *Medical Image Analysis*, 2019.

[6] CL. Li, K. Sohn, J. Yoon, and T. Pfister. CutPaste: Self-supervised learning for anomaly detection and localization. In *Proc. IEEE/CVF conf. on computer vision and pattern recognition*, 2021.

[7] V. Zavrtanik, M. Kristan, and D. Skočaj. DRAEM-a discriminatively trained reconstruction embedding for surface anomaly detection. In *Proc. IEEE/CVF International Conf. on Computer Vision*, 2021.

[8] H. M. Schlüter, J. Tan, B. Hou, and B. Kainz. Natural synthetic anomalies for self-supervised anomaly detection and localization. In *European Conf. on Computer Vision*, 2022.

[9] T. Defard, A. Setkov, A. Loesch, and R. Audigier. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In *International Conf. on Pattern Recognition*, 2021.

[10] K. Roth, L. Pemula, J. Zepeda, B. Schölkopf, T. Brox, and P. Gehler. Towards total recall in industrial anomaly detection. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, 2022.

[11] D. Rezende and S. Mohamed. Variational inference with normalizing flows. In *International conf. on machine learning*, 2015.

[12] B. Marc, P. Foucher, F. Forbes, and P. Charbonnier. Évaluation de méthodes de détection d'anomalies non supervisée pour l'auscultation des ouvrages d'art. In *Proc. RFIAP*, 2024.

[13] J. Yu, Y. Zheng, X. Wang, W. Li, Y. Wu, R. Zhao, and L. Wu. FastFlow: Unsupervised anomaly detection and localization via 2d normalizing flows. *arXiv preprint*, 2021.

[14] Y. Zhou, X. Xu, J. Song, F. Shen, and HT. Shen. MSFlow: Multiscale Flow-Based Framework for Unsupervised Anomaly Detection. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[15] S. Wang, Y. Li, H. Luo, and C. Bi. CL-Flow: Strengthening the Normalizing Flows by Contrastive Learning for Better Anomaly Detection. *arXiv preprint*, 2023.

[16] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 2003.

[17] O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015.

[18] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger. MVTec AD — A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection. In *IEEE/CVF Conf. on CVPR*, 2019.

[19] F. Yang, L. Zhang, S. Yu, D. Prokhorov, X. Mei, and H. Ling. Feature pyramid and hierarchical boosting network for pavement crack detection. *IEEE Trans. on Intelligent Transportation Systems*, 2019.

[20] H. Tao, B. Liu, J. Cui, and H. Zhang. A Convolutional-Transformer Network for Crack Segmentation with Boundary Awareness. In *2023 IEEE International Conf. on Image Processing*, 2023.

[21] S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint*, 2016.

[22] L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

[23] J. Tan, B. Hou, T. Day, J. Simpson, D. Rueckert, and B. Kainz. Detecting outliers with poisson image interpolation. In *Medical Image Computing and Computer Assisted Intervention*, 2021.

[24] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, and S. Wang. DeepCrack: Learning Hierarchical Convolutional Features for Crack Detection. *IEEE Transactions on Image Processing*, 2018.