Few-Shot Early Defect Detection in Industrial Manufacturing Using Explainable Siamese Networks

Ravindu Senarathna
Department of Computing
Informatics Institute of Technology
Colombo, Sri Lanka
ravindusenruwan@gmail.com

Nethmi Wijesinghe
Department of Computing
Informatics Institute of Technology
Colombo, Sri Lanka
wt.nethmi@gmail.com

2025 Moratuwa Engineering Research Conference (MERCon) | 979-8-3315-6724-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/MERCON67903.2025.11217104

Abstract— Early defect detection is essential to ensure product quality and reduce waste in industrial manufacturing. However, traditional defect detection methods rely on large labelled datasets to train models or manual inspection, both of which can be time-consuming and prone to errors. The challenge lies in developing an automated system for early anomaly detection that requires minimal labelled data, making it adaptable to various industrial environments. To address this challenge, a Siamese network, a few-shot learning technique, was utilised. The network was designed to detect defects in images of products with only a few labelled examples. A custom lightweight Convolutional Neural Network (CNN) was developed for the embedding phase of the Siamese network to reduce inference time while maintaining model performance. This architecture, coupled with Explainable AI (XAI), enabled the model to provide transparent and explainable results, which is crucial for industrial applications where quick decision-making and understanding of model behaviour are vital.

Keywords— Anomaly Detection, Siamese Network, Few-shot Learning, Explainable AI (XAI).

I. Introduction

Industrial manufacturing has long been the backbone of global economic development, evolving from steam power in the 18th century [1] to today's advanced automated systems. It spans sectors like textiles, pharmaceuticals, construction, and electronics, driven by the need for high-volume, high-quality, and cost-effective production. Maintaining product quality in such fast-paced environments is critical. Quality inspection ensures that products meet standards before reaching the market, preventing defects that could lead to recalls, waste, or safety risks. This is especially crucial in industries like pharmaceuticals and electronics, where a single defect can have serious consequences.

In recent years, machine learning (ML) has transformed quality control by automating defect detection. Anomaly detection models can identify subtle deviations from known-good samples, improving accuracy and consistency. However, most existing models require large, labelled datasets and struggle to adapt when presented with new or rare defects, making them less practical for real-time production lines where conditions constantly change [2].

Few-shot learning offers a promising alternative. It enables models to learn from very few labelled examples, addressing the issue of limited data. But despite its strengths, the current few-shot anomaly detection approaches often lack transparency. In high-stakes manufacturing, explainability is just as important as accuracy. Trust and adoption remain low without clear reasoning behind a model's decision.

This paper explores a solution that combines few-shot learning with explainable Siamese networks to detect defects early in production, using minimal data while remaining explainable and adaptable to changing environments.

II. Anomaly Detection

Anomaly detection in industrial settings has been widely studied to enhance and automate defect identification processes. With the growing integration of artificial intelligence, particularly deep learning, the ability to detect subtle irregularities in manufacturing has significantly improved. This section provides an overview of current approaches in image-based anomaly detection, focusing on the roles of computer vision, and explores different learning paradigms, including supervised, unsupervised, and few-shot learning.

A. Computer Vision and Feature Extraction

The use of computer vision in anomaly detection has rapidly advanced due to deep learning (DL) techniques, especially Convolutional Neural Networks (CNNs), which have proven effective at extracting relevant features from image data. CNNs can learn hierarchical representations directly from raw pixel inputs, making them well-suited for tasks like defect identification in industrial images [3].

A key advancement in this area is transfer learning, where pre-trained CNN models such as ResNet50, InceptionV3, DenseNet121, and Xception, originally trained on large datasets like ImageNet, are repurposed for new tasks by removing the top classification layers and using the lower layers for feature extraction [4]. This removes the need for handcrafted feature methods like edge detection, Gabor filters, or statistical texture features (e.g., Gray-Level Co-occurrence Matrix), allowing the model to focus on learning from the data itself.

B. Supervised Anomaly Detection

In supervised anomaly detection, models are trained with fully labelled datasets where both normal and defective samples are known. According to surveys by Cui, Liu, and Lian (2023)[5] and Wang et al. (2021)[6], this approach has traditionally been effective in controlled environments, where each defect class is well-documented.

However, the main limitation is data dependency. The requirement for large, labelled datasets covering all possible defect types. This is not always feasible in real-world manufacturing, where collecting and labelling data is time-consuming and expensive. Furthermore, supervised models often struggle with generalisation, meaning they may fail to detect novel defects not seen during training. A common

979-8-3315-6724-8/25/$31.00 ©2025 IEEE

algorithm used in this category is Support Vector Machines (SVM), but due to the challenges mentioned, research has gradually shifted toward unsupervised approaches for better scalability.

C. Unsupervised Anomaly Detection.

Unsupervised anomaly detection methods are typically trained using only normal data, without requiring examples of defective or anomalous samples. These models assume that anomalies will deviate significantly from the learned distribution. While this approach eliminates the need for labelled anomalies, it introduces several limitations, including the challenge of selecting an appropriate detection threshold and the inability to introduce or learn from specific anomalous examples.

1) Autoencoder (AE)

Autoencoder is one of the most widely used unsupervised methods [7]. It works by compressing input images through an encoder and reconstructing them through a decoder. The reconstruction error, calculated as the difference between input and output, is used to detect anomalies. A high reconstruction error suggests that the input image deviates from normal patterns. However, this method relies heavily on setting a proper error threshold, which typically requires extensive experimentation and manual tuning. Additionally, it cannot adapt to new anomaly types because it only learns from normal data.

2) Variational Autoencoder (VAE)

Variational Autoencoder enhance standard AEs by encoding the input into a distribution (mean and variance) rather than a single point. This probabilistic encoding helps handle more diverse inputs and provides better generalisation to unseen samples [8][9]. Despite this, VAEs are more complex to train and suffer from similar thresholding issues as AEs.

3) Generative Adversarial Network (GAN)

GAN go a step further by training two competing networks - a generator that tries to create realistic images, and a discriminator that learns to distinguish between real and generated images [10]. Anomalies are flagged when the generator fails to replicate input data convincingly. While GANs can be powerful, they are computationally intensive and harder to train effectively, especially for smaller datasets.

III. Few-shot learning

Few-shot learning is a technique designed to overcome the challenge of limited labelled data by allowing models to generalise from only a few examples per class. This makes it especially useful in situations where obtaining large annotated datasets is impractical. In anomaly detection, few-shot approaches aim to identify anomalies by learning key patterns from a small set of normal and, occasionally, anomalous samples. Although Siamese Networks were originally introduced for tasks like signature verification and face recognition [11], they have been increasingly adapted for anomaly detection despite not being initially designed for it. This research builds upon that adaptation, applying the Siamese approach to industrial defect detection.

A. Siamese Networks

Siamese Networks are a popular few-shot architecture that works by comparing input image pairs. The model consists of two identical subnetworks that extract feature embeddings from each image [11]. The distance (e.g., Euclidean or absolute) between these embeddings is then calculated to determine similarity. A small distance implies the images are similar (likely normal), while a large distance suggests a discrepancy. Siamese networks are well-suited for few-shot tasks due to their ability to learn a similarity metric instead of classifying explicitly.

B. FewSOME

FewSOME is a one-class anomaly detection method built on a Siamese-based structure [12]. It was trained using 60 normal samples from the MVTec industrial anomaly detection dataset [10], enabling the model to learn what constitutes normality in an industrial setting. However, while effective, this approach still requires a relatively large number of normal samples for a few-shot scenario. Moreover, because it is trained only on normal samples, it has no built-in mechanism to introduce or learn directly from anomalous examples, limiting its adaptability when new types of defects emerge.

IV. Explainable AI (XAI)

XAI techniques help make deep learning models more transparent and explainable by highlighting the areas the model focuses on when making decisions. This is particularly important in fields like anomaly detection, where understanding the cause of a prediction is as crucial as the prediction itself. Several XAI techniques have been developed to explain convolutional neural networks (CNNs) and their decision-making process.

A. Saliency Maps (Pixel Attribution Maps)

Saliency Map compute the gradient of the model's output with respect to each input pixel. This results in a heatmap that reveals which pixels contributed most to the model's decision. Because of their fine-grained sensitivity to image structure, saliency maps help visualise subtle pixel-level cues that influence prediction [13].

B. Grad-CAM (Gradient-weighted Class Activation Mapping)

Grad-CAM works by computing the gradients of a target output (typically the class score or feature map) with respect to the activations in the final convolutional layer. These gradients are averaged and multiplied with the activation maps to generate a weighted heatmap. This highlights spatial regions that contribute most to the prediction. In anomaly detection, especially when using CNN-based classifiers trained to detect defects, Grad-CAM can show which image regions led to a high anomaly score [14].

C. LIME (Local Interpretable Model-agnostic Explanations)

LIME works by perturbing parts of the input image (such as turning patches off) and observing how the model's output changes. It then trains a simple, interpretable model (like linear regression) on these perturbed samples to approximate the local behaviour of the complex model. In anomaly detection, LIME can help highlight which image regions are most responsible for increasing the anomaly score. For example, if removing a small region significantly drops the anomaly score, that region is likely part of the defect [15].

However, these explainability methods are primarily designed for traditional classification models that produce class-specific outputs or anomaly scores based on a single input image. In contrast, Siamese networks operate differently, they compare two input images and generate a similarity score based on the distance between their learned feature embeddings. Since there is no direct class score or heatmap tied to a single image, methods like LIME cannot be effectively applied. Recent work has explored adapting Grad-CAM to Siamese networks by applying it to the backbone network to highlight features influencing similarity scores. These approaches generally focus on higher-level feature activations within deep models [16][17]. In this work, we focus instead on a lightweight approach that directly computes pixel-level saliency by measuring the gradient of the similarity score with respect to input pixels. This allows for fine-grained, interpretable visualisations that reveal the most influential regions in the input image, offering intuitive insights into the model's decision process without relying on class-based outputs.

V. Proposed method

This method is structured around four key components: image pairing, embedding extraction, distance measurement and the XAI module. Together, these components form a complete pipeline tailored for anomaly detection using a Siamese network. The process begins with preparing and augmenting the dataset to form image pairs suitable for training. These pairs are then passed through a feature extractor to generate embeddings, which are compared using a distance metric to evaluate similarity. Based on these comparisons, the model is trained to distinguish between normal and anomalous image pairs. Finally, an explainability module is integrated to produce saliency maps that highlight regions of interest, making the model's predictions more transparent. Each component is described in detail in the following sections.

A. Data preparation for

The first step in the proposed method involves preparing the dataset for training the Siamese network. A total of 24 images are initially used, consisting of 8 anomalous images and 16 non-anomalous images. These images are categorised into three specific groups: 8 anchor images selected from the non-anomalous set, 8 non-anomalous images that differ from the anchors, and 8 anomalous images representing defective or irregular instances.

To enhance the diversity and volume of the training data, each image undergoes an augmentation process. Seven augmentation techniques are applied to each image, including rotation by 90 degrees clockwise, rotation by 180 degrees, rotation by 90 degrees counterclockwise, horizontal flipping, vertical flipping, adjustment to high brightness, and adjustment to low brightness. These transformations simulate various realistic conditions and viewpoints, thereby helping the model generalise better during training. As a result, the number of images in each class increases to 64, significantly enriching the dataset.

Once augmentation is complete, the next step involves constructing image pairs, which is a critical requirement for training Siamese networks. Unlike conventional deep learning models that operate on single images, a Siamese network learns to differentiate between pairs of images by comparing their feature embeddings. Therefore, pairs of images are created with associated similarity labels. Each non-anomalous image is paired with an anchor image and assigned a label of 1, indicating that the two images are similar. Conversely, each anomalous image is paired with an anchor image and assigned a label of 0, reflecting that the two images are dissimilar.

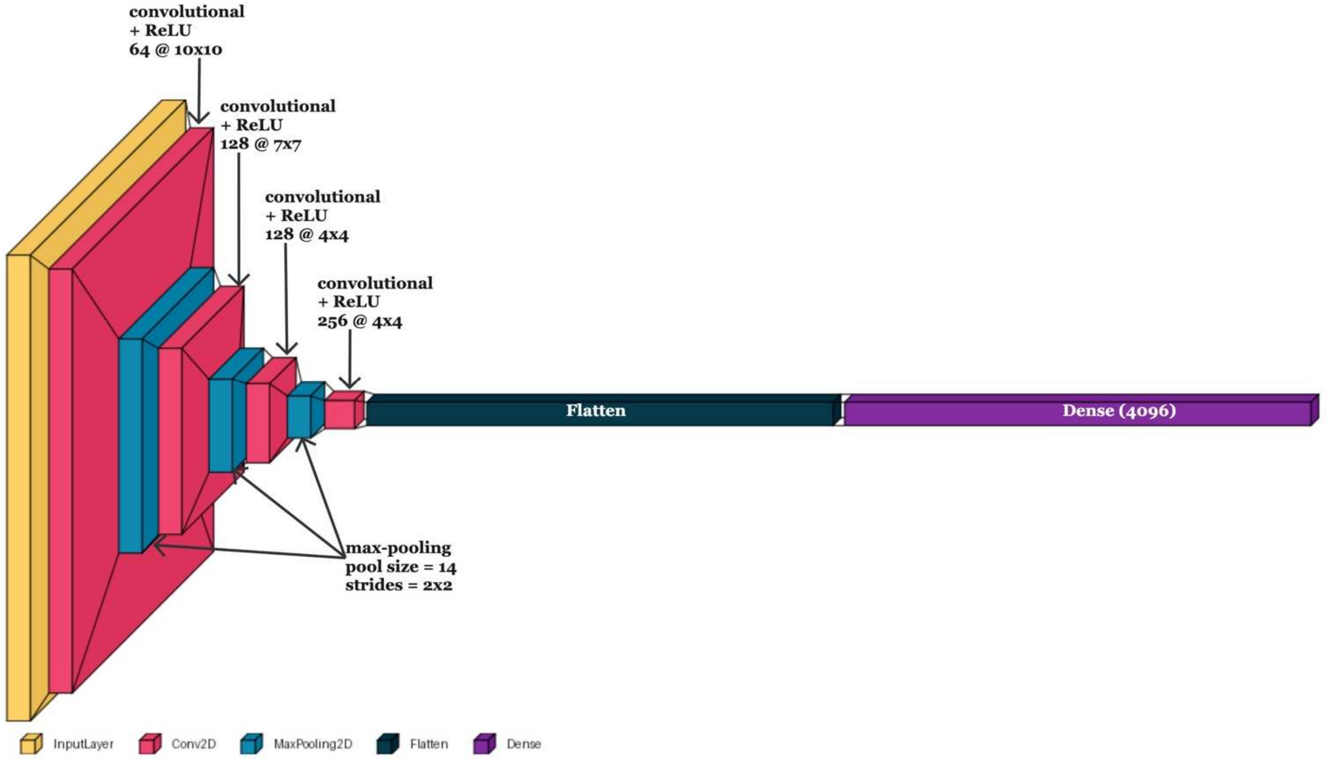

Fig. 1. Architecture of the proposed lightweight CNN model for feature embedding within the Siamese network. The network consists of 1 input layer, 4 convolutional 2d layers, 3 max pooling 2d layers, 1 flatten layer, and 1 dense (fully connected) layer. This design is optimised to extract essential discriminative features while maintaining low model complexity and fast inference times.

This pairing mechanism ensures that the Siamese network is trained to learn meaningful representations of visual similarity and dissimilarity, which is essential for effective anomaly detection. The specific architecture of the Siamese network and the underlying training process are discussed in the next section.

B. *Siamese network*

1) *Embedding Model*

. In the Siamese network architecture, the extraction of embeddings from input images plays a pivotal role in the overall performance of the model. These embeddings, which serve as condensed feature representations, are subsequently compared through a distance metric to assess the similarity between image pairs. Hence, the quality and efficiency of the feature extraction process directly impact the Siamese network's ability to discriminate between similar and dissimilar images.

Conventional approaches to feature embedding often leverage transfer learning with pre-trained convolutional neural networks (CNNs) such as VGG19, ResNet50, or InceptionV3 [4]. These models, while highly effective in capturing complex feature hierarchies, are typically computationally intensive, leading to increased inference times and elevated resource requirements. For applications where low-latency and resource efficiency are critical, such overhead becomes a significant limitation.

To overcome these challenges, this research proposes a lightweight custom CNN specifically optimised for embedding extraction within the Siamese framework. The developed network comprises only 8 layers, a substantial reduction compared to the 19 layers of VGG19 [18], the 50 layers of ResNet50 [19], and the approximately 48 layers of InceptionV3 [20]. Despite its relatively shallow depth, the proposed CNN architecture is meticulously designed to retain essential discriminative features necessary for accurate similarity learning. By minimising network complexity without compromising feature quality, the custom CNN achieves a favourable balance between inference speed and embedding effectiveness. The detailed structure of the proposed CNN is depicted in Fig. 1.

2) *DistanceLayer*

After obtaining the feature embeddings of the anchor and validation images through the embedding model, the DistanceLayer computes their absolute element-wise difference. This operation highlights the magnitude of difference between corresponding features in the two embeddings, without considering the direction of change. Using the absolute value ensures that the comparison captures only how much two features differ, simplifying the information fed to the classifier.

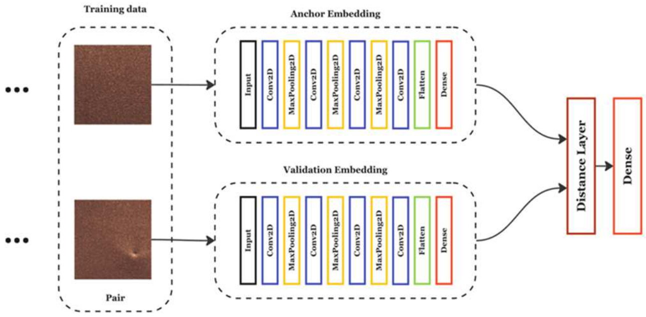

Fig. 2. Siamese model architecture

There are several ways to measure distance between embeddings in Siamese networks, such as Euclidean distance (L2 norm), cosine similarity, or even learnable distance functions. Each method captures different aspects of similarity: Euclidean distance measures overall geometric distance, while cosine similarity measures the angle between vectors, focusing on direction rather than magnitude [21][22]. However, in this case, absolute difference is used for its simplicity and computational efficiency and because it provides a straightforward, explainable representation of feature-wise dissimilarity.

The output from the DistanceLayer is then passed into a dense layer with a sigmoid activation, producing a similarity score between 0 and 1. By explicitly computing the distance, the model reduces the burden on the classifier to infer relational patterns directly from embeddings, leading to improved explainability and more efficient training convergence.

C. *XAI Module*

Considering the limitations of conventional explainability techniques in Siamese models, a custom saliency-based method was developed to provide explainability for the proposed Siamese network. This module enables the visualisation of the most influential regions within the validation images that affect the model's similarity predictions.

The method operates by calculating the gradient of the similarity score with respect to the input pixels of the validation image. When a validation image is compared with an anchor image, the model produces a similarity output based on the distance between their feature embeddings. To determine the contribution of each pixel to this similarity score, the partial derivatives of the output are computed with respect to each input pixel. The magnitude of these gradients reflects the sensitivity of the output to changes in the corresponding pixels. To construct the saliency map, the absolute values of these gradients are taken to capture the overall strength of influence regardless of direction. These values are then averaged across the colour channels to produce a single-channel intensity map. Finally, a smoothing operation is applied to the map to improve its visual explainability by reducing noise while preserving critical structures.

The resulting saliency maps highlight the regions within the validation image that have the greatest impact on the model's assessment of similarity or dissimilarity. Higher intensity areas in the map correspond to regions that the network relies upon more heavily during the comparison process. By adapting the gradient-based saliency technique to a Siamese framework, this method provides a practical and effective way to interpret model decisions in architectures where conventional class-based explanations are not applicable.

VI. Testing

The performance of the proposed custom-built CNN was evaluated against several widely used pre-trained CNNs, including VGG19, ResNet50, and InceptionV3. The evaluation was conducted using the leather class from the MVTec Anomaly Detection dataset, with precision, recall, F1-score, accuracy, and area under the receiver operating characteristic curve (AUROC) used as the primary performance metrics. Results are presented in TABLE I.

TABLE I. PERFORMANCE COMPARISON OF THE SIAMESE MODEL USING DIFFERENT CNN FEATURE EXTRACTORS ON THE MVTEC LEATHER DATASET

Embedding model	Precision	Recall	F1-score	Accuracy	AUROC
Custom CNN	93.59%	100%	0.97	96.58%	0.985
VGG19	98.63%	60%	0.75	66.44%	0.623
RestNet50	94.52%	56.56%	0.70	60.96%	0.66
InceptionV3	100%	52.21%	0.69	54.79%	0.147

The results show that the custom CNN outperformed the pre-trained models in terms of overall balance across metrics. While InceptionV3 and VGG19 showed high precision, they suffered from significantly lower recall, which is critical in anomaly detection contexts where missing a true anomaly has a high cost. The custom model achieved perfect recall, high precision, and the highest F1-score, indicating strong performance in both detecting and correctly classifying anomalies. Additionally, it demonstrated superior AUROC, suggesting robust discriminative ability across different thresholds.

Given that early anomaly detection is a key focus of this research, further evaluation was conducted to assess computational efficiency. The models were compared in terms of inference time, training time, and model size, as shown in TABLE II.

These results reinforce the practical advantages of the proposed model. The custom CNN significantly reduced inference and training time while maintaining a competitive model size. Compared to the larger and more complex pre-trained alternatives, it offers a more efficient solution suitable for real-time or resource-constrained deployment environments.

To evaluate the model's generalisation capability beyond the MVTec dataset, it was further tested on the Marble Surface AD 2 dataset [23]. TABLE III. presents the accuracy and efficiency metrics on both datasets with the custom CNN as the embedding model.

TABLE II. INFERENCE TIME, TRAINING TIME, AND MODEL SIZE OF THE CUSTOM CNN COMPARED TO PRE-TRAINED CNN MODELS

Embedding model	Inference time	Training time	Model size
Custom CNN	~ 0.03s	~ 55s	467 MB
VGG19	~ 0.069s	~ 6m	1.47 GB
RestNet50	~ 0.085s	~ 12m	5.03 GB
InceptionV3	~ 0.072s	~ 16m	363 MB

TABLE III. EVALUATION RESULTS OF THE CUSTOM SIAMESE MODEL ON MVTEC AND MARBLE SURFACE AD 2 DATASETS

Dataset	Accuracy (%)	Recall (%)	Precision (%)	F1-score	AUROC	Inference Time	Model size (MB)
MVTec	96.58	100	93.59	0.97	0.985	~ 30ms	467
Marble	92.18	94.4	90.32	0.92	0.936	~ 30ms	467

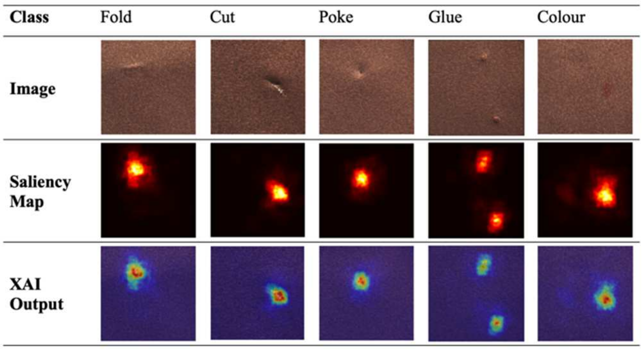

Fig. 3. Saliency map visualizations of anomalous samples from the MVTec leather dataset, showing regions that contributed most to the similarity judgment.

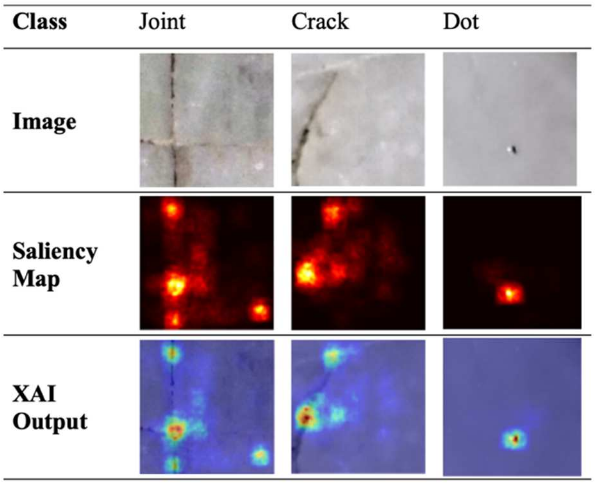

Fig. 4. Saliency map outputs for samples from the Marble Surface AD 2 dataset, highlighting key areas influencing the anomaly detection outcome.

The model maintained high accuracy, precision, and recall on the external Marble dataset, confirming its ability to generalise across domains. The inference time and model size remained consistent, indicating stable efficiency regardless of the dataset.

To complement the quantitative evaluation, visual explainability was assessed using the custom saliency-based XAI module described earlier. The generated saliency maps highlight the most influential regions within anomalous samples that guided the model's similarity judgments. Fig. 3 presents example outputs from the MVTec leather dataset, while Fig. 4 illustrates results from the Marble Surface AD 2 dataset. In both cases, the highlighted regions align well with actual defect areas, confirming the model's ability not only to detect anomalies accurately but also to provide meaningful visual explanations of its decisions.

VII. CONCLUSION

This study presented a custom-designed Siamese network for anomaly detection, focusing on the performance and efficiency of the model. The custom CNN used as the embedding model in the Siamese architecture demonstrated superior performance compared to the same Siamese network using pre-trained CNNs, including VGG19, ResNet50, and InceptionV3, as embedding models. On the leather class of the MVTec Anomaly Detection dataset, the Siamese model with the custom CNN achieved a precision of 93.59%, a recall of 100%, an F1-score of 0.97, and an AUROC of 0.985, highlighting the effectiveness of the custom CNN with the Siamese model in anomaly detection tasks.

A key contribution of this work was the development of a pixel-level explainable AI (XAI) module designed for the Siamese network. The XAI module generates saliency maps to visualise which parts of the input images influenced the model's anomaly detection decisions. This approach is essential for improving model explainability, especially in non-conventional architectures like Siamese networks, where traditional XAI techniques do not directly apply.

The performance of the custom Siamese network was also evaluated in terms of computational efficiency. Testing on a MacBook Pro with an M3 Pro chip demonstrated an impressive inference time of 30ms. To assess real-world applicability, future work could explore the model's performance in throttled, controlled environments and test its feasibility for real-time anomaly detection. Additionally, the model's performance with higher-resolution images could be evaluated to detect more subtle anomalies and expand its capabilities.

In conclusion, the custom Siamese network with an integrated XAI module offers a robust, efficient, and explainable solution for anomaly detection. The strong performance of the Siamese model using the custom CNN as its embedding model, combined with its computational efficiency and explainability, makes it a promising approach for industrial anomaly detection tasks. Future work could focus on optimising the model architecture, evaluating its generalisability across a broader range of datasets, and further refining the XAI module for application in other detection scenarios.

References

[1] Peer Vries, "The Industrial Revolution," 2008, pp. 158–161. Accessed: Jul. 27, 2024. [Online]. Available: https://www.researchgate.net/publication/282572543_The_Industrial_Revolution

[2] M. S. Minhas and J. Zelek, "Semi-supervised Anomaly Detection using AutoEncoders," Jan. 06, 2020, *arXiv*: arXiv:2001.03674. Accessed: Jul. 29, 2024. [Online]. Available: http://arxiv.org/abs/2001.03674

[3] R. Ghosal, *Anomaly Detection Using Computer Vision*. 2024. doi: 10.58445/rars.972.

[4] N. Wijesinghe, R. Perera, N. Sellahewa, and P. D. Talagala, "Early Identification of Deforestation using Anomaly Detection," in *2023 8th International Conference on Information Technology Research (ICITR)*, Dec. 2023, pp. 1–6. doi: 10.1109/ICITR61062.2023.10382919.

[5] Y. Cui, Z. Liu, and S. Lian, "A Survey on Unsupervised Anomaly Detection Algorithms for Industrial Images," *IEEE Access*, vol. 11, pp. 55297–55315, 2023, doi: 10.1109/ACCESS.2023.3282993.

[6] S. Wang, J. F. Balarezo, S. Kandeepan, A. Al-Hourani, K. G. Chavez, and B. Rubinstein, "Machine Learning in Network Anomaly Detection: A Survey," *IEEE Access*, vol. 9, pp. 152379–152396, 2021, doi: 10.1109/ACCESS.2021.3126834.

[7] W.-H. Choi and J. Kim, "Unsupervised Learning Approach for Anomaly Detection in Industrial Control Systems," *Applied System Innovation*, vol. 7, no. 2, Art. no. 2, Apr. 2024, doi: 10.3390/asi7020018.

[8] J. An and S. Cho, "Variational Autoencoder based Anomaly Detection using Reconstruction Probability," 2015. Accessed: Oct. 08, 2024. [Online]. Available: https://www.semanticscholar.org/paper/Variational-Autoencoder-based-Anomaly-Detection-An-Cho/061146b1d7938d7a8dae70e3531a00fceb3c78e8

[9] D. Zimmerer, F. Isensee, J. Petersen, S. Kohl, and K. Maier-Hein, "Unsupervised Anomaly Localization using Variational Auto-Encoders," Jul. 11, 2019, *arXiv*: arXiv:1907.02796. Accessed: Oct. 08, 2024. [Online]. Available: http://arxiv.org/abs/1907.02796

[10] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," *Int J Comput Vis*, vol. 129, no. 4, pp. 1038–1059, Apr. 2021, doi: 10.1007/s11263-020-01400-4.

[11] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese Neural Networks for One-shot Image Recognition," 2015.

[12] N. Belton, M. T. Hagos, A. Lawlor, and K. M. Curran, "FewSOME: One-Class Few Shot Anomaly Detection with Siamese Networks," arXiv.org. Accessed: Oct. 08, 2024. [Online]. Available: https://arxiv.org/abs/2301.06957v4

[13] K. Fan, C. Ma, Y. Peng, Y. Fang, and K. Ma, "Decision Rules are in the Pixels: Towards Pixel-level Evaluation of Saliency-based XAI Models," Oct. 2024, Accessed: Mar. 13, 2025. [Online]. Available: https://openreview.net/forum?id=mKGXdsq7fD

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," *Int J Comput Vis*, vol. 128, no. 2, pp. 336–359, Feb. 2020, doi: 10.1007/s11263-019-01228-7.

[15] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why Should I Trust You?': Explaining the Predictions of Any Classifier," Aug. 09, 2016, *arXiv*: arXiv:1602.04938. doi: 10.48550/arXiv.1602.04938.

[16] I. E. Livieris, E. Pintelas, N. Kiriakidou, and P. Pintelas, "Explainable Image Similarity: Integrating Siamese Networks and Grad-CAM," *J. Imaging*, vol. 9, no. 10, p. 224, Oct. 2023, doi: 10.3390/jimaging9100224.

[17] A. Fedele, R. Guidotti, and D. Pedreschi, "Explaining Siamese networks in few-shot learning," *Mach Learn*, vol. 113, no. 10, pp. 7723–7760, Oct. 2024, doi: 10.1007/s10994-024-06529-8.

[18] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Apr. 10, 2015, *arXiv*: arXiv:1409.1556. doi: 10.48550/arXiv.1409.1556.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 10, 2015, *arXiv*: arXiv:1512.03385. doi: 10.48550/arXiv.1512.03385.

[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," Dec. 11, 2015, *arXiv*: arXiv:1512.00567. doi: 10.48550/arXiv.1512.00567.

[21] T. Rosa, R. Primartha, and A. Wijaya, "Comparison of Distance Measurement Methods on K-Nearest Neighbor Algorithm For Classification," presented at the Sriwijaya International Conference on Information Technology and Its Applications (SICONIAN 2019), Atlantis Press, May 2020, pp. 358–361. doi: 10.2991/aisr.k.200424.054.

[22] V. B. S. Prasath *et al.*, "Distance and Similarity Measures Effect on the Performance of K-Nearest Neighbor Classifier -- A Review," *Big Data*, vol. 7, no. 4, pp. 221–248, Dec. 2019, doi: 10.1089/big.2018.0175.

[23] "Marble Surface Anomaly Detection - 2." Accessed: Mar. 16, 2025. [Online]. Available: https://www.kaggle.com/datasets/wardaddy24/marble-surface-anomaly-detection-2