Research on Anomaly Detection Algorithm Based on Pixel-level Histogram Distribution Variations of Integrated Features

Wendong Geng, Fenghua Zhu, Shichao Chen, Yisheng Lv, Gang Xiong

***Abstract*—We propose an innovative methodology for the detection of defects in industrial components, leveraging deep neural networks to extract features and employing a truncated L1 loss function for the enhanced training of the feature extraction model, ensuring the model's adeptness in capturing the nuances of positive samples. Our approach further incorporates the state-of-the-art KAN network layer, substituting it within the feature extraction network, which has yielded improvements in both the velocity of the training process and the precision of the outcomes. The model, once trained, distills feature maps from a training set comprised exclusively of positive instances, followed by feature amalgamation and histogram mapping to delineate the feature distribution at each pixel locale. Akin procedures are executed on the test set, where the histograms of the images are juxtaposed with the emblematic histograms derived from the training set, facilitating the computation of histogram similarity. This metric is instrumental in assessing the affiliation of each pixel site on the test images to positive samples; an elevated score is indicative of a closer resemblance to positive samples, whereas a diminished score suggests the presence of anomalies. The utilization of an F1-adaptive algorithm to ascertain an optimal threshold K enables the efficacious identification of defective specimens. Our methodology is proficient in the detection of defect categories and the precise localization of anomalous regions, exhibiting adaptability across a spectrum of defect types. It has garnered competitive outcomes when evaluated on the open-source dataset MVTec, substantiating its viability in the realm of industrial defect detection.**

***Index Terms*—Unsupervised Learning; Defect Detection; KAN Network**

I. Introduction

In industrial image analysis, the difficulty of detecting and localizing anomalies is largely due to the limited availability and wide variety of anomalous examples, which can range from minor surface scratches to severe structural flaws. This complexity challenges the effectiveness of supervised learning approaches, which require abundant labeled anomaly data. Consequently, research has shifted towards unsupervised learning methods that utilize only normal samples for training. The primary unsupervised strategies include reconstruction-based, synthesis-based, and embedding-based approaches. Reconstruction-based methods propose that a deep network trained exclusively on normal data may fail to accurately reconstruct anomalous regions, utilizing pixel-level reconstruction errors to identify anomalies. However, this approach can lead to false positives due to the network's propensity for overgeneralization. Synthesis-based methods strive to establish the decision boundary between normal and anomalous instances by generating artificial anomalies on images devoid of actual anomalies. Nevertheless, these synthetic anomalies often exhibit discrepancies in appearance compared to real anomalies and do not encompass all potential scenarios. Embedding-based methods have recently yielded significant advancements. These methods achieve anomaly detection by mapping normal features into a compressed space, positioning anomalous features distant from the normal feature cluster. Nevertheless, employing features extracted from networks pretrained on general datasets, such as ImageNet, may introduce domain bias, which is detrimental for industrial image applications. To mitigate the aforementioned challenges and enhance the understanding of the spatial characteristics of positive samples, we introduce Gaussian noise into the feature space to generate negative samples. This method not only avoids the complexity of synthesizing anomalies at the raw image level but also delineates the boundary of the normal feature space through noise level fine-tuning. The objective is to enable the model to thoroughly learn the feature space of positive samples, translating the intricate and diverse image defects into discrepancies within feature channels, which are subsequently compared using histograms. Our computational approach calculates histogram differences with pixel-level precision, i.e., histogram differences at each individual pixel location. Furthermore, our method employs a streamlined discriminator realized through KAN network layers, simplifying the anomaly detection process and improving computational efficiency. Integrating these innovations, our proposal is not only straightforward to train and implement but also exhibits

*The work was supported by China National Railway Group Co., Ltd. Science and Technology Research and Development Program Project (L2022X002), Jiangxi Provincial Natural Science Foundation (20232ABC03A07), Chongqing Transportation Technology Project (CQJT-CZKJ2024-04)

Wendong Geng is with School of AI Technology, Chinese Academy of Sciences University, Beijing 100071, China.

Fenghua Zhu,Shichao Chen,Yisheng Lv and Gang Xiong are with State Key Laboratory of Multimodal Artificial Intelligence Systems, Institute of Automation, Chinese Academy of Sciences,Beijing 100190, China.

(gengwendong2023@ia.ac.cn, fenghua.zhu@ia.ac.cn, shichao.chen@ia.ac.cn, yisheng.lv@ia.ac.cn, gang.xiong@ia.ac.cn)

Corresponding author: Gang Xiong.

superior performance and inference capabilities, significantly reducing model training duration. This novel approach offers a practical solution for image anomaly detection and localization in industrial settings.

II. Related Work

There are three main categories of methods for anomaly detection and localization: synthetic-based methods, reconstruction-based methods, and embedding-based methods.

Synthetic-based methods aim to synthesize anomalies on normal images. For instance, DRÆM proposed a network trained to distinguish between normal images and synthetically generated anomalies. Similarly, CutPaste[10] introduced a strategy where parts of images are cut and pasted onto random positions of larger images, followed by training convolutional neural networks (CNNs) to differentiate between normal and synthetic anomaly distributions. However, synthetic anomalies may not perfectly match real anomalies, given their diverse and unpredictable nature, making it impractical to generate a comprehensive set of all possible anomalies.

Reconstruction-based methods leverage generative models such as autoencoders and generative adversarial networks [4] to encode and reconstruct normal data. Additionally, some approaches frame anomaly detection as an image restoration problem, where portions of images are randomly masked and neural networks are used to predict the erased information. Anomalies are identified based on pixel-wise differences between normal images and their reconstructed counterparts. However, anomalies may be well-reconstructed if they share common compositional patterns with normal training data, or if the decoder overly emphasizes decoding certain anomaly encodings.

Based on embedding-based methods, more advanced performance has been achieved. These methods embed normal features into a compressed space, with anomaly features located far from normal clusters in the embedding space. Typical approaches utilize networks pre-trained on ImageNet for feature extraction. For instance, PaDiM [5] employs Gaussian embedding to extract anomaly patch features, leveraging a pre-trained model. PatchCore [6] utilizes a repository of most representative nominal patch features, evaluating input features during testing using Mahalanobis distance or maximum feature distance. However, industrial images often differ in distribution from ImageNet, potentially leading to mismatch issues when directly using pre-trained features. Moreover, methods either compute the inverse covariance or perform nearest-neighbor searches using a memory bank, processes that are complex and challenging for real-time computation on edge devices, limiting performance.

Inspired by these methods, our approach adapts features using a transfer learning feature adapter trained on the target dataset to mitigate biases in pre-trained CNNs during inference. By performing pixel-wise comparisons, our method enhances final segmentation and classification accuracy at a fundamental level. Additionally, replacing complex Mahalanobis distance computations with histogram similarity improves computation speed without sacrificing accuracy, making our approach suitable for real-time deployment in edge device scenarios.

A. Our Contribution

- Inspired by SimpleNet[7] and PaDiM[5], this study integrates model training, feature extraction, and distribution analysis, enhancing anomaly detection in the MVTec dataset for industrial applications.
- During model training, the incorporation of the innovative Kolmogorov-Arnold Network layer has been validated for its effectiveness, leading to improved performance with fewer training epochs and optimizing the training process.
- Feature extraction and histogram distribution enhance the differentiation between anomalous and normal data, providing a computationally efficient alternative to PaDiM's Mahalanobis distance.

III. Method

A. Feature Extractor

We describe the entire algorithm process as follows. As depicted in Figure 1, in Step 1, we denote the training and test sets as X_{train} and X_{test} respectively. We train the feature extractor f_{ext} on the training set X_{train}, consisting entirely of positive samples. f_{ext} utilizes common backbone networks such as ResNet18 (R18) or Wide ResNet-50-2 [8]. Since pre-trained networks tend to bias towards their training datasets, training f_{ext} on the target dataset X_{train} and utilizing specific layers from the ResNet50 [8] model as the feature extractor f_{ext} is reasonable.

For any image $X_i \in R^{W*H*C}$ in both the training and test sets, where W, H, and C denote the height, width, and number of channels of the feature map, and L represents the total number of layers in the backbone network. l denotes a specific layer, where ${X_i}^{Wl*Hl*Cl} = f_{ext}^{l}(X_i)$ denotes the output feature map of the l-th layer of the feature extractor. W_i, H_i, and C_t respectively represent the height, width, and number of channels of the feature map at the l-th layer. The feature maps from multiple layers are fused using an aggregation function f_{agg} (adaptive average pooling is used here) to aggregate features. This entire process can be defined as:

$$ {X_i}^1 = f_{ext}^1(X_i) \tag{1} $$

$$ {X_i}^2 = f_{ext}^2({X_i}^1) \tag{2} $$

$$ {X_i}^3 = f_{ext}^3({X_i}^2) \tag{3} $$

$$ {O_i}^+ = f_{agg}({X_i}^1, {X_i}^2, {X_i}^3) \tag{4} $$

To train the feature extractor to better capture the common characteristics of *N* positive samples, deep learning techniques are employed. This involves iterative forward inference and

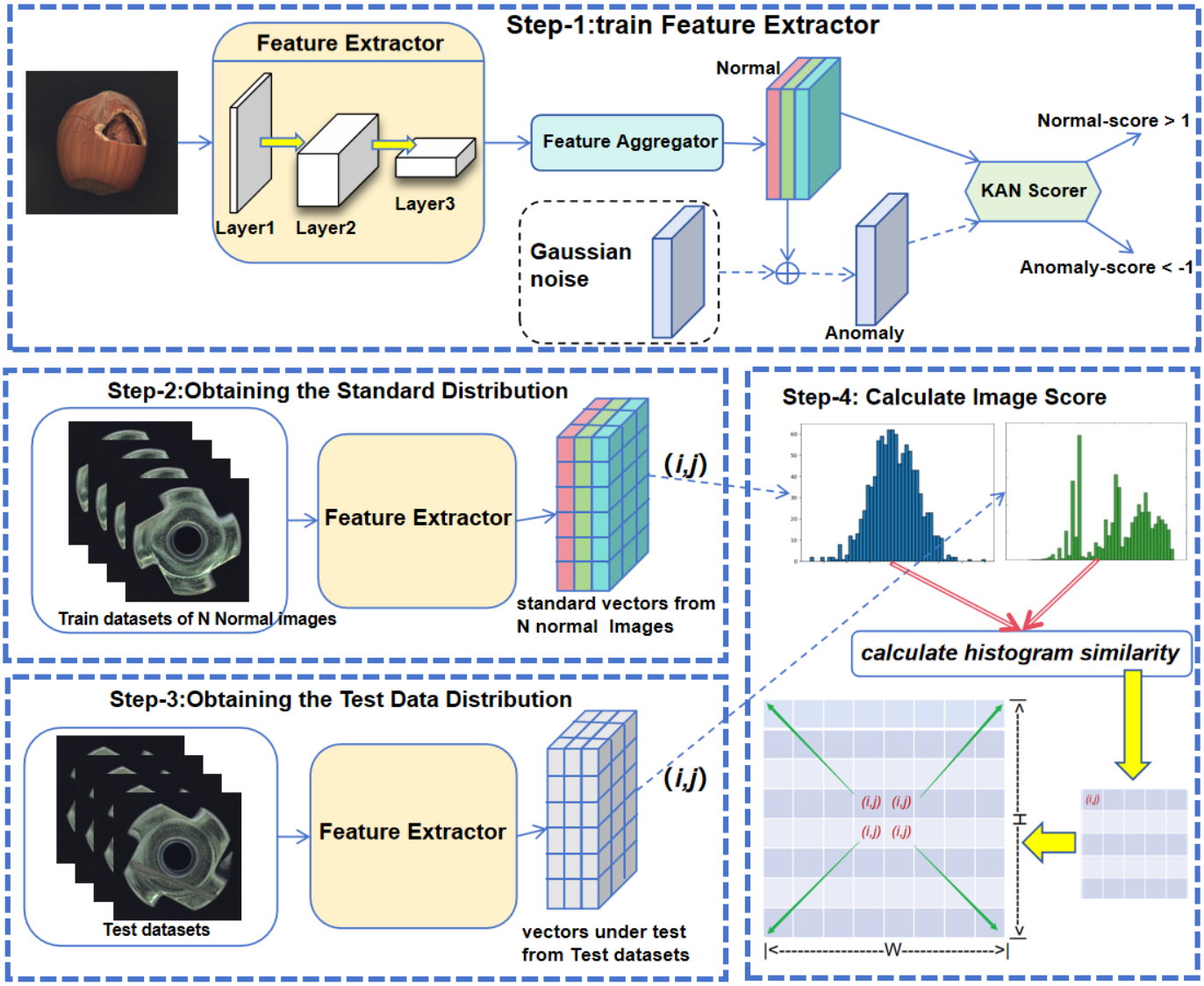

Fig. 1. Overview of the detection method. Step 1 involves further training the model's feature extractor on positive samples, building upon a pre-trained model from ImageNet. In Step 2, the trained model is used to extract features and perform feature fusion on a training set composed entirely of positive samples. The mean of the features from N feature maps is computed to obtain the final standard feature map, and the histogram distribution of each point i on the feature map is calculated. In Step 3, the trained feature extractor is used again to repeat the process from Step 2 for each image in the test set, resulting in a histogram for each point i on the feature map. Step 4 involves comparing the histograms for each corresponding point (i,j) on the feature maps, calculating histogram similarity, and finally expanding the feature maps into the original image using linear interpolation.

backpropagation to update the parameters of the feature extractor f_{agg}. In defect detection, obtaining defect samples is challenging and costly. Although studies rely on additional data to synthesize defect images, the synthesis process is relatively complex. Inspired by SimpleNet, we construct abnormal features ${O_i}^+$ by adding simple noise to the feature maps of normal samples ${O_i}^-$ in the feature space. During the training process, the noise vector does not participate in parameter updates. Each time, a new independent noise vector τ is generated, with each τ following an independent and identically distributed Gaussian distribution $N(\mu, \sigma^2)$. The abnormal features can be expressed as:

$$o_i^- = o_i^+ + \tau \tag{5}$$

B. *Loss Function and Scorer*

Defining the scorer $\mathcal{D}_\Phi$, we replaced the traditional MLP structure with KAN network layers. As shown in Table 2, experiments demonstrate that substituting parts of the MLP structure with KAN network layers (such as the fully connected layer at the end of a typical classification network) not only accelerates model convergence but also improves certain computational metrics of the model. Since negative samples are generated together with normal features, they are all fed into the discriminator during training. The discriminator is

expected to output positive values for normal features and negative values for abnormal features. The loss function adopts a simple truncated L1 loss function, derived as follows:

$$loss = max(0, t_h^+ - D_\Phi(o_i^+)) + max(0, -t_h^- + D_\Phi(o_i^-)) \quad (6)$$

The truncation terms t_h^+ and are t_h^- used to prevent overfitting, with default values set to 0.5 and -0.5. To enhance the model's ability to distinguish between positive and negative feature maps, we set these values to 1 and -1, respectively.

C. Extracting Common Features of Positive Samples

After training and backpropagation, the parameters of the feature extractor f_{ext} are updated, resulting in a feature extractor F_{ext} that is biased towards the feature distribution of the X_{train} training data. F_{ext} has the same structure as f_{ext}but with different weight parameters. At this point, we compute the feature map O_t that captures the common features of *N* normal sample images, as follows:

$$O_t = mean(\sum_t^N f_{agg}(F_{ext}^1(x_i), F_{ext}^2(x_i^1), F_{ext}^3(x_i^2))) \quad (7)$$

The dimensions of O_t are Wl,Hl and Cl (to avoid information loss during feature fusion, the dimensions of the largest shallow feature map, typically *l*=1, are used). O_t can be understood as being composed of Wl*Hl embedding vectors, each of size (1, Cl). That is:

$$v_{ij}^T \in O_t^{Wl*Hl}, \{i, j | (i, j) \in Wl * Hl\} \quad (8)$$

D. Similarity Calculation

Perform the same operation on X_{test} using F_{ext} to obtain v_{ij}^{test}. Calculate the pixel-level histogram similarity between the embedding vectors X_{test} and the standard vector v_{ij}^T for each corresponding point *(i,j)*. Here, we use the correlation coefficient method. When calculating histogram similarity using the correlation coefficient method (Correlation), the setting of bins has a relatively small impact on the result (the setting of bins mainly depends on the specific characteristics of the image and the application scenario; here we use the default value of 255). The closer the value is to 1, the more similar the histograms are, as shown in the following formula:

$$H1(i) = Histogram(i), i \in v_{ij}^{test} \quad (9)$$

$$H2(i) = Histogram(i), i \in v_{ij}^T \quad (10)$$

$$S_{(i,j)} = \frac{\sum_i (H1(i) - \mu_{H1})(H2(i) - \mu_{H2})}{\sqrt{\sum_i (H1(i) - \mu_{H1})^2 \sum_i (H2(i) - \mu_{H2})^2}} \quad (11)$$

Here, H1 and H2 are the two histograms, and μ_{H1} and μ_{H2} are their means. From this, we can compute the histogram similarity matrix for the feature map of the image to be tested. Finally, using linear interpolation, the matrix is expanded to the original image size of *(W,H)*. At this point, high-score positions *(i,j)* in the matrix that exceed the predetermined threshold *K* indicate abnormal regions.

IV. Experimental Results and Comparative Analysis

A. Evaluation Metrics

Defective samples are extremely rare. Therefore, in datasets with class imbalance, metrics such as accuracy can be misleading because they are easily influenced by the majority class. Hence, ROC AUC (Receiver Operating Characteristic Area Under Curve) is used as an evaluation metric. ROC AUC is a comprehensive metric that simultaneously considers the classifier's performance at different thresholds. By calculating the area under the ROC curve, AUC provides a single value for comparing the effectiveness of different models. The value ranges from 0 to 1, where a higher value indicates better classification performance of the model.

B. Dataset Description

We initially evaluate our model on the MVTec AD dataset [2], which is specifically designed for testing anomaly detection algorithms and single-class learning settings in industrial quality control. The dataset encompasses 15 categories, with approximately 240 images. The original image resolutions range from 700x700 to 1024x1024 pixels. It includes 10 object categories and 5 texture categories. The objects are consistently centered and aligned throughout the dataset, as illustrated in the first column of images in Figure 1.

C. Data Augmentation and Hyperparameter Tuning

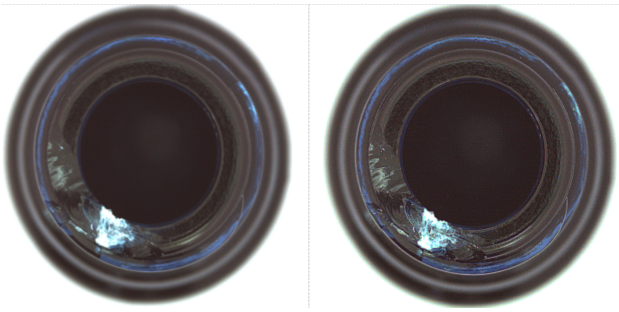

Fig. 2. The image on the left corresponds to an instance within the MVTec toothbrush class, whereas the image on the right illustrates the outcome subsequent to the application of a high-pass filter, which has significantly amplified the clarity of the textural details.

Data augmentation procedures encompass a series of operations: with a 0.5 probability, images are horizontally flipped; resized to a dimension of 512x512 pixels; randomly cropped to 288x288 pixels; and finally, centrally cropped to a size of 256x256 pixels. Training can be conducted on a per-category basis, with a default of 50 training epochs for each category. If the final scorer utilizes an MLP (Multilayer Perceptron) structure, the Adam optimizer[9] is employed. The initial learning rate is set to 10^{-4}, with a batch size of 16 images. For

TABLE I

A COMPARATIVE ANALYSIS IS CONDUCTED WITH THE STATE-OF-THE-ART STUDIES WITHIN THE LAST TRIENNIAL PERIOD ON THE MVTEC AD DATASET. THE PERFORMANCE OF EACH STUDY IS DELINEATED THROUGH THE IMAGE-LEVEL AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC (I-AUROC) AND THE PIXEL-LEVEL AREA UNDER THE RECEIVER OPERATING CHARACTERISTIC (P-AUROC) METRICS, ARTICULATED IN THE CONVENTIONAL FORMAT OF I-AUROC PERCENTAGE FOLLOWED BY A SLASH AND THE P-AUROC PERCENTAGE.

Model	**PaDiM**	**PatchCore**	**SimpleNet**	**Our-method**
Carpet	99.7 / 98.9	98.7 / 98.0	98.5 / 97.4	**99.8 / 98.9**
Grid	97.5 / 96.3	98.2 / 98.7	**99.9 / 98.8**	98.5 / 97.7
Leather	98.8 / 98.9	98.3 / 99.4	98.9 / 99.1	**99.7 / 99.3**
Tile	98.2 / 93.9	98.7 / **95.0**	**100** / 94.5	98.2 / 93.9
Wood	99.0 / 94.4	98.2 / 95.0	97.9 / **97.8**	**99.5** / 94.4
Avg.Text.	98.6 / 96.5	98.4 / 97.2	99.0 / **97.5**	**99.1** / 96.8
Bottle	99.8 / 98.1	**100 / 99.8**	100 / 98.8	98.8 / 98.1
Cable	93.1 / 96.8	99.5 / 98.4	99.9 / 97.6	**100 / 98.8**
Capsule	87.5 / 98.5	**98.9** / 98.2	97.7 / 98.9	98.7 / **99.3**
Hazelnut	93.1/ 98.0	**100 / 99.7**	99.6 / 97.5	99.1 / 95.2
Metal Nut	99.8 / 97.2	99.7 / 98.4	**100 / 98.6**	97.8 / 93.2
Pill	93.9 / 96.1	98.3 / **97.5**	98.6 / 96.3	**100** / 93.1
Screw	83.7 / 98.3	98.6 / **99.4**	98.7 / 94.7	**100** / 99.1
Toothbrush	97.5 / 98.7	**100 / 99.7**	97.2 / 97.3	98.1 / 98.7
Transistor	97.5 / 97.5	98.8 / **98.0**	96.5 / 95.3	**100** / 97.5
Zipper	94.6 / 98.4	**98.7 / 98.9**	98.6 / 98.5	98.3 / 98.4
Avg.Obj.	94.0 / 97.4	**98.8 / 98.8**	98.6 / 97.1	98.7 / 97.1
Average	95.6 / 97.3	98.9 / **98.3**	97.4 / 97.3	**99.1** / 97.1

texture categories, prior to operations such as image rotation and cropping, the images are first processed through a high-pass filter. This step enhances the clarity of the textures, allowing the feature extractor to more effectively learn the distribution of the image texture data. The high-pass filter can be mathematically represented as follows:

$$K = \begin{bmatrix} -1 & -1 & -1 \\ -1 & -9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{12}$$

$$I_{ij}^{t} = \sum_{m=-1}^{1} \sum_{n=-1}^{1} K_{m+1,n+1} * I_{m+i,n+j} \tag{13}$$

The convolution kernel *K* is a 3×3 matrix, $K_{m+1,n+1}$ with its elements representing the corresponding pixel values on the original image. The outcome of applying the high-pass filter is depicted in Figure 3.

V. EXPERIMENTAL RESULTS

Figure 3 illustrates a comparative evaluation of our approach against studies presented at the Conference on Computer Vision and Pattern Recognition (CVPR) over the preceding triennium, achieving results that are highly competitive. On the open-source MVTec dataset, which encompasses a variety of 15 defect categories, our methodology has delivered performance metrics for both the classification indicator, denoted as I-AUROC, and the defect area detection indicator, denoted as P-AUROC. Remarkably, our approach has not only matched but, in select metrics, exceeded the precedents established by previous scholarly endeavors.

As illustrated in Table 2, the network, which is centered around the WideResNet50 [8] as its foundational model,

TABLE II

THE DATA PERFORMANCE IN TERMS OF CLASSIFICATION ACCURACY AND SEGMENTATION PRECISION IS EVIDENT ACROSS VARIOUS LAYERS AFTER FEATURE FUSION.

Layer1	**Layer2**	**Layer3**	**I-AUROC**	**P-AUROC**
Y			98.1	93.3
	Y		98.2	95.2
		Y	97.7	93.8
Y	Y		97.1	**98.7**
	Y	Y	98.5	94.3
Y	Y	Y	**99.6**	98.3

integrates features extracted from disparate layers and subsequently computes the respective classification and segmentation accuracies. The amalgamation of feature maps emanating from each layer culminates in the attainment of superior performance across both accuracy metrics.

TABLE III

THE COMPARATIVE ANALYSIS BETWEEN EMPLOYING A KAN-LAYER AS THE SCORING MECHANISM AND UTILIZING A CONVENTIONAL MULTILAYER PERCEPTRON (MLP) LAYER FOR THE SAME PURPOSE.

scorer structure	**L1 truncated loss**		***Average I-AUROC***
	Epoch=50	***Epoch=100***	
MLP-Layer	0.45	0.23	93.73
Kan-Layer	0.27	0.12	94.03

VI. CONCLUSION

Our research endeavor, which stands on the shoulders of prior scholarly work, has achieved a significant reduction

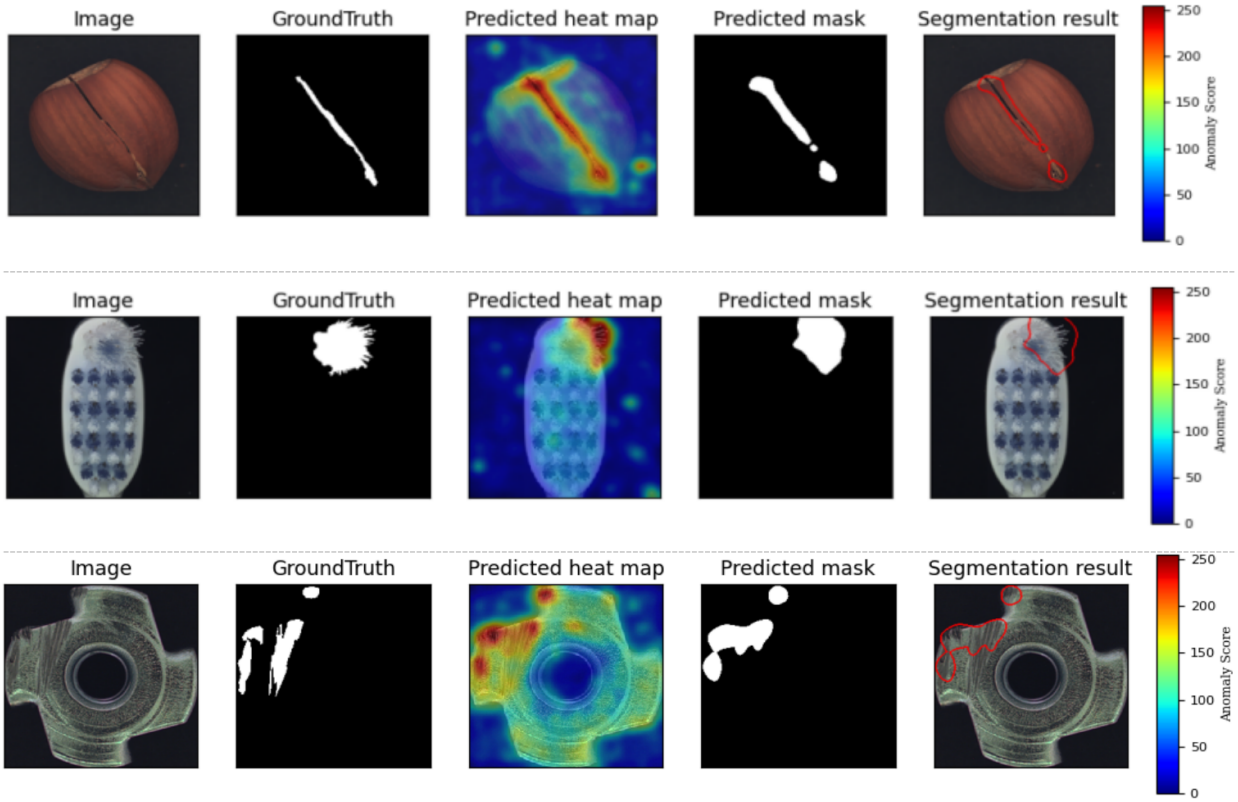

Fig. 3. In the MVTec AD dataset, the visualization of samples is meticulously arranged. The initial pair of columns exhibits the pristine imagery and the designated defect locales as furnished by the open-source repository. Subsequent columns, specifically the third and fourth, delineate the heatmaps prognosticated by the model along with the localized predictions of defect occurrences. The terminal column encapsulates the outcomes of the segmentation process.

in model training duration through the innovative integration of the Kan-Layer as the underlying framework of the scoring mechanism. Furthermore, the adoption of histogram similarity as a computational metric has obviated the need for the computationally intensive Mahalanobis distance, which is often accompanied by substantial hardware expenditure. This advancement presents a groundbreaking methodology that facilitates the swift implementation and operationalization within the context of industrial applications.

REFERENCES

[1] Ziming Liu,Yixuan Wang et al. "KAN: Kolmogorov-Arnold Networks." arXiv:2404.19756, 30 Apr. 2024, v1; last revised 16 June 2024, v4.

[2] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad–a comprehensive real-world dataset for unsupervised anomaly detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 9592–9600, 2019.

[3] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In International Conference on Pattern Recognition, pages 475–489.Springer, 2021.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.2

[5] Thomas Defard, Aleksandr Setkov, Angelique Loesch, and Romaric Audigier. Padim: a patch distribution modeling framework for anomaly detection and localization. In International Conference on Pattern Recognition, pages 475–489.Springer, 2021.

[6] Karsten Roth, Latha Pemula, Joaquin Zepeda, Bernhard Scholkopf, Thomas Brox, and Peter Gehler. Towards total recall in industrial anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14318–14328, 2022.

[7] Zhikang Liu, Yiming Zhou, Yuansheng Xu, Zilei Wang. SimpleNet: A Simple Network for Image Anomaly Detection and Localization. Department of Automation, University of Science and Technology of China, Meka Technology Co.,Ltd. arXiv:2303.15140v2 [cs.CV], 28 Mar 2023.

[8] S.Zagoruyko and N.Komodakis, "Wide residual networks," in BMVC,2016.

[9] Fenghua Zhu, Guoxi Li, Zhenjiang Li, Cheng Chen, Ding Wen, "A Case Study of Evaluating Traffic Signal Control Systems Using Computational Experiments", IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 4, pp.1220-1226, 2011.

[10] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. In Proceedings of the IEEE/CVFConference on Computer Vision and Pattern Recognition,pages 9664–9674, 2021. 2, 4, 5

[11] Gang Xiong, Xiaoyu Chen, Nan Shuo, Yisheng Lv, Fenghua Zhu, Tianci Qu, Peijun Ye. Collaborative Optimization of Cyber Physical Social Systems for Urban Transportation Based on Knowledge Automation. IFAC PAPERSONLINEnull. 2020, 53(5): 572-577, http://dx.doi.org/10.1016/j.ifacol.2021.04.144.

[12] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in ICLR, 2014.

[13] Gang Xiong, Xiaoyu Chen, Nan Shuo, Yisheng Lv, Fenghua Zhu, Tianci Qu, Peijun Ye. Cyber-Physical-Social Systems for Smart City: An Implementation Based on Intelligent Loop. IFAC PAPERSONLINEnull. 2020, 53(5): 501-506, http://dx.doi.org/10.1016/j.ifacol.2021.04.136.