EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Relative Position Encoding-Enhanced Vision Transformer for Image-Level Defect Classification in Manufacturing Textured-Surface Inspection

Jiamin Liu[1,*], Yuhui Sun[1]

[1]NONCOMMISSIONED OFFICER ACADEMY OF PAP, Hangzhou, Zhejiang, 310000, China

Abstract

INTRODUCTION: In textile and textured-surface manufacturing, visual defects in woven fabrics, leather-like materials, tiles, and grid products directly influence product grading, downstream cutting, assembly reliability, after-sales repair, and supply-chain delivery stability. Weak defects embedded in repetitive textures are still difficult for manual inspection and conventional handcrafted vision methods.
OBJECTIVES: This study constructs an image-level defective/non-defective classification model for manufacturing quality inspection by enhancing Vision Transformer with relative spatial modeling, so that subtle texture disruption can be identified before products enter subsequent processing or distribution stages.
METHODS: A ViT-Small backbone was combined with learnable relative position encoding. Input images were resized, normalized, divided into 16 × 16 patches, and transformed into token embeddings. Relative spatial bias was inserted into multi-head self-attention to encode patch-to-patch displacement while retaining absolute positional information. The model was evaluated with precision, accuracy, recall, F1-score, and AUROC on a supervised MVTec-derived texture subset.
RESULTS: ViT-RPE outperformed ResNet-18, an activation-embedded CNN, EfficientFormer-L1, Swin-Tiny, and the baseline ViT under the same supervised image-level classification protocol. It achieved an accuracy of 0.954 ± 0.004 and an AUROC of 0.977 ± 0.003 on DAGM, and an accuracy of 0.939 ± 0.005 and an AUROC of 0.969 ± 0.004 on the MVTec-derived subset. Ablation results showed that combining absolute and relative positional information produced the best performance, with limited additional computational cost.
CONCLUSION: The method provides a compact image-level pre-screening solution for automated product inspection in manufacturing lines, especially where repetitive texture, material appearance consistency, and rapid quality sorting are required.

Keywords: Vision transformer, relative position encoding, surface defect classification, industrial visual inspection, manufacturing quality inspection

Received on 16 June 2026, accepted on 23 July 2026, published on 16 September 2026

doi: 10.4108/eetsis.13596

*Corresponding author. Email: ljm1232026@163.com

1. Introduction

Textured surface defect inspection is a key quality-control task in manufacturing systems such as textile production, leather-like material processing, ceramic tile fabrication, and patterned component inspection. Surface defects, including holes, broken yarns, stains, missing ends, scratches, and local structural distortion, can affect product grading, appearance consistency, mechanical reliability, downstream cutting or assembly, and subsequent maintenance or distribution decisions [1,2]. In many production lines, visual inspection is still partly completed by operators. However, manual inspection is easily affected by experience, fatigue, lighting variation, and line speed, which may lead to unstable judgments and delayed quality feedback.

Machine-vision-based inspection has therefore been widely introduced into manufacturing quality inspection to improve detection efficiency, repeatability, and process traceability [3,4]. Traditional methods usually rely on handcrafted features, including histogram statistics, co-occurrence features, Gabor filters, wavelet transform, Fourier analysis, and local binary patterns [2,3]. These methods can be applied to relatively regular textures under controlled imaging conditions. However, their performance often decreases when defects are weak, irregular, or embedded in complex repetitive backgrounds. In addition, handcrafted features are usually difficult to transfer across different material batches, fabric types, processing conditions, or production environments.

Deep learning has brought clear progress to industrial surface inspection in recent years [4-6]. CNN-based models can automatically learn hierarchical visual features and have been used in manufacturing defect recognition, detection, and segmentation tasks. Typical models include Faster R-CNN, YOLOv3, and U-Net [7-9]. Although CNNs are effective in extracting local features, their receptive fields are still built layer by layer. For repetitive industrial textures, subtle defects may appear as small local changes while also breaking the broader texture arrangement. It is therefore difficult for conventional CNN structures to capture local abnormal details and global texture consistency at the same time.

Vision Transformer (ViT) offers another way to process image information. It divides an image into patch sequences and uses self-attention to model relationships among patches [10]. Compared to CNNs, ViT is better suited to capturing long-range dependencies and global context information. This property is very useful in making textured-surface inspection, since many defects are not only isolated visual changes, but also disturbance in the surrounding material pattern or process induced texture law. However, the standard ViT mainly uses absolute position encoding, which records the location of each patch but does not directly describe the relative spatial relationship between patches [11].

For textured product images, relative spatial information is important. Many defects are not determined by their absolute position in the image, but by their deviation from the normal arrangement of neighboring texture regions. For example, a broken yarn, local stain, scratch, or abnormal surface mark may become recognizable only when compared with adjacent material patterns. Thus, the introduction of relative location information into the self-attention process can be helpful for the identification of weak and irregular defects. Similar ideas have also been discussed in previous studies on relative position representation in self-attention models [12-13]. Unlike the window-restricted relative bias used in Swin Transformer or position-neighborhood approaches that construct task-specific local anomaly representations, the proposed design retains global self-attention and absolute positional embeddings while introducing a lightweight learnable two-dimensional relative bias into every ViT-Small encoder block, which is particularly suitable for capturing both full-image texture regularity and patch-level structural disruption in image-level textured-surface defect classification with limited computational overhead.

Based on this idea, a relative position encoding enhanced Vision Transformer is used for image-level textured surface defect classification in manufacturing inspection. The task is defined as defective/non-defective image-level classification rather than pixel-level localization, which is suitable for rapid pre-screening before cutting, assembly, packaging, warehouse sorting, or downstream quality review. A learnable relative position bias is added to the attention calculation, so that token interaction is influenced by both visual feature similarity and relative spatial displacement. This design helps the model capture local structural irregularities while still preserving global contextual information [14].

The proposed framework combines Vision Transformer with learnable relative position encoding for textured product surface defect classification. Relative spatial relationships are incorporated into the self-attention process to improve the representation of subtle structural inconsistencies in repetitive manufacturing textures. Comparative experiments and ablation studies on public datasets are further conducted to evaluate whether the framework can support image-level product inspection with limited additional computational cost.

2. Materials and Methods

2.1 Overall framework

The proposed method adopts an image-level binary classification framework that includes image preprocessing, patch embedding, Transformer-based feature extraction, and defective/non-defective prediction. Input images are first resized to a unified resolution and normalized before training. The processed images are then divided into non-overlapping patches, and each patch is converted into a token embedding through linear projection. These token sequences are subsequently fed into a Vision Transformer backbone, where relative position encoding is incorporated into the self-attention process. The final

encoded representation is delivered to a lightweight classification layer for defective/non-defective prediction [15].

Figure 1 shows the ViT-RPE framework's overall workflow. Compared with the baseline Vision Transformer, the main difference lies in the introduction of learnable relative spatial bias into the attention calculation. By incorporating relative displacement information between neighboring patches, the model can better capture subtle structural inconsistencies in textured regions while still preserving global contextual information.

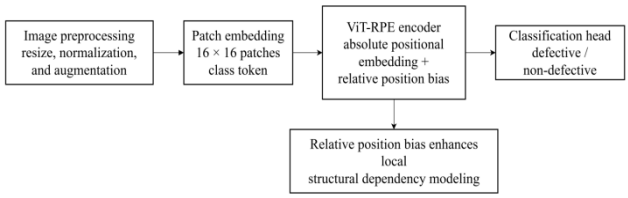

Figure 1. The overall workflow of a ViT-RPE framework for image-level texture product surface defects classification in Manufacturing Inspection

The overall procedure mainly contains four stages: image normalization and data augmentation, patch partition and token embedding, Transformer encoding with relative position-aware self-attention, and image-level defective/non-defective classification. The framework keeps the overall architecture relatively simple while improving the modeling of local spatial relationships in textured surface images.

2.2 Datasets

Two public datasets were used in this study: DAGM 2007 and a supervised image-level classification subset reorganized from MVTec AD. The two datasets were trained and evaluated separately, and no image was transferred between them. For both datasets, the target label was converted into a binary image-level label, where defective images were assigned label 1 and non-defective images were assigned label 0. Pixel-level masks and defect annotations were used only to determine whether an image contained a defect and were not provided as model inputs.

Before data augmentation or normalization-statistic calculation, the image paths were sorted and divided into training, validation, and testing subsets using a fixed random seed of 42. Stratification was performed according to the joint distribution of the original texture category and the converted binary label. The resulting partitions were fixed for all comparison models and ablation experiments. Channel-wise normalization statistics and class weights were estimated exclusively from the training subset. The normalization statistics were calculated from resized but unaugmented training images after channel conversion. Stochastic augmentation was applied online only to training samples. The validation subset was used for early stopping and model selection, while the testing subset remained inaccessible until final evaluation.

For the MVTec-derived experiment, images from the official training and testing directories were reorganized to construct a closed-set supervised classification task. Consequently, this protocol differs from the original one-class anomaly-detection and localization setting of MVTec AD. The reported results are used only for comparisons among models retrained under the same supervised split and should not be directly compared with published results obtained under unsupervised, one-class, few-shot, or pixel-level localization protocols. The category-level sample distribution is reported in Table 1, and the exact image lists and split-generation script are provided in Supplementary Files S1 and S2.

Table 1. Summary of dataset composition and data split used in this study

Dataset	Category	Non-defective	Defective	Training N/D	Validation N/D	Testing N/D	Dataset
DAGM 2007	All ten classes	14000	2100	11200/1680	1400/210	1400/210	DAGM 2007
MVTec-derived	carpet	308	89	246/71	31/9	31/9	MVTec-derived
MVTec-derived	grid	285	57	229/45	28/6	28/6	MVTec-derived
MVTec-derived	leather	277	92	221/74	28/9	28/9	MVTec-derived
MVTec-derived	tile	263	84	210/67	26/9	27/8	MVTec-derived

MVTec-derived	Total	1133	322	906/257	113/33	114/32	MVTec-derived

DAGM 2007 dataset

DAGM 2007 contains ten classes of grayscale textured images with artificially generated surface defects [16]. Each image was assigned a binary label according to its original annotation: images associated with a defect annotation were labeled as defective, and images without a defect annotation were labeled as non-defective. The original texture-class identifier was retained as a stratification variable, although the final prediction target contained only two classes. The dataset contained 14,000 non-defective images and 2,100 defective images. Samples were stratified within each combination of original texture class and binary label and then divided into training, validation, and testing subsets at a ratio of 80:10:10 using random seed 42. After the class-level partitions were generated, the subsets from the ten texture classes were merged, resulting in 12,880 training images, 1,610 validation images, and 1,610 testing images. The training subset contained 11,200 non-defective and 1,680 defective images; both the validation and testing subsets contained 1,400 non-defective and 210 defective images. The split was completed before any image augmentation. Each original file path appeared in only one subset, and augmented variants were generated online only from training images. The defect masks were used for label conversion and were excluded from network input. This protocol evaluates closed-set image-level defect classification across the ten DAGM texture classes and does not involve pixel-level defect localization.

MVTec-derived textured subset

Four texture categories were selected from MVTec AD: carpet, grid, leather, and tile [17]. For each category, all images in the official train/good directory and the normal images in the official test/good directory were assigned the non-defective label. Images located in the defect-type subdirectories of the official test directory were assigned the defective label. Defect masks were used only to verify defect presence and were not used during model training or evaluation. After label conversion, the original MVTec directory boundary was not retained because the objective was to construct a supervised image-level classification task. Within each texture category, samples were stratified by the binary label and divided into training, validation, and testing subsets using random seed 42. The four category-level training subsets were then merged for model training, while category identifiers were retained in the split manifest. This procedure produced 1,163 training images, 146 validation images, and 146 testing images. The complete distribution of non-defective and defective samples is presented in Table 1.

All splits were generated before augmentation. No original image, file path, or augmented derivative was shared across subsets. Normalization statistics and class weights were estimated only from the training subset. Images belonging to the same defect subtype could occur in different subsets because the experiment followed a closed-set supervised classification protocol; therefore, the reported results measure recognition of known texture categories and defect patterns rather than generalization to unseen defect types. The exact filename-level partition is provided in Supplementary File S1.

2.3 Image preprocessing and data augmentation

All images were resized to 224×224224\times224224×224 pixels using bilinear interpolation. For DAGM, each grayscale image was replicated along the channel dimension to form three identical channels before tensor conversion, solely to match the three-channel input of ViT-Small. ImageNet normalization statistics were not used. Instead, the channel-wise mean and standard deviation were calculated separately for each dataset from resized, unaugmented training images after channel conversion. The same fixed statistics were then applied to the training, validation, and testing subsets. Because the three DAGM channels contained identical intensity values, their calculated means and standard deviations were also identical.

Online augmentation was restricted to the training subset. Random horizontal flipping and random vertical flipping were independently applied with probabilities of 0.5. Random rotation was activated with a probability of 0.5, and its angle was uniformly sampled from $[-15\circ, 15\circ]$ using bilinear interpolation. Brightness and contrast adjustment was applied with a probability of 0.5, with the corresponding factors independently sampled from [0.8,1.2]. The operations were executed in the order of resizing, channel conversion, geometric augmentation, photometric augmentation, tensor conversion, and normalization. Validation and testing images underwent only resizing, channel conversion, tensor conversion, and normalization. The same preprocessing and augmentation implementation was used for all comparison models.

2.4 Vision Transformer backbone

The Vision Transformer represents each textured-surface image as a series of fixed patch tokens [18]. Let the input image be denoted as $X \in R^{H\times W\times C}$, where H, W, and C denote the image height, width, and number of channels, respectively. Both H and W are required to be divisible by the patch size P. The numbers of patches along the vertical and horizontal directions are $G_H = H / P$ and

$G_W = W / P$, respectively, and the total number of patches is $N = G_H G_W$. After flattening, the i-th patch is represented as $x_i \in R^{P^2 C}$ and projected into a D-dimensional token:

$$z_i = x_i W_E + b_E + e_i^{abs}, \quad i = 1,2,...,N \qquad (1)$$

where $W_E \in R^{P^2 C \times D}$ and $b_E \in R^D$ denote the learnable patch-projection matrix and bias vector, respectively, and $e_i^{abs} \in R^D$ is the absolute positional embedding of the i-th patch. A learnable class token $z_{cls} \in R^D$, together with its absolute positional embedding e_{cls}^{abs}, is concatenated with all patch tokens to form the input sequence:

$$Z^{(0)} = [z_{cls} + e_{cls}^{abs}; z_1; z_2; ...; z_N] \in R^{(N+1) \times D} \qquad (2)$$

For the l-th Transformer encoder block and the h-th attention head, the query, key, and value matrices are calculated as

$$Q^{(l,h)} = Z^{(l-1)} W_Q^{(l,h)}, \quad K^{(l,h)} = Z^{(l-1)} W_K^{(l,h)}, \quad V^{(l,h)} = Z^{(l-1)} W_V^{(l,h)} \qquad (3)$$

where $W_Q^{(l,h)}$, $W_K^{(l,h)}$, $W_V^{(l,h)} \in R^{D \times d_h}$, $d_h = D / M$, and M is the number of attention heads. Without relative position bias, the output of the h-th attention head is

$$O^{(l,h)} = Soft\max\left(\frac{Q^{(l,h)} (K^{(l,h)})^T}{\sqrt{d_h}} \right) V^{(l,h)} \qquad (4)$$

The outputs of all attention heads are concatenated and projected before being passed to the residual connection, layer normalization, and feed-forward network. The output of multi-head is concatenated with a feed-forward network with residual links and normalization. In this study, a ViT-Small backbone was used because it provides sufficient representation capacity for 224 × 224 industrial texture images while keeping the model size suitable for image-level pre-screening. The patch size was set to P=16, producing a 14 × 14 patch grid. The embedding dimension was set to D=384, the number of encoder blocks was L=12, and the number of attention heads was M=6. These settings keep dh=64, which is a stable attention dimension commonly used in Transformer implementation and avoids excessive computational cost for manufacturing inspection images.

2.5 Relative position encoding module

Motivation

Absolute positional embedding gives each token a fixed location identity, but it does not directly describe the relative displacement between different tokens. In textured surface defect classification, many anomalies are related to changes in local texture arrangement rather than to their absolute coordinates in the image. For example, a weak stain, broken yarn, or local structural distortion is often recognized because it breaks the continuity of neighboring texture regions. Therefore, modeling relative spatial relationships among patches can help the network capture subtle and irregular defects more effectively.

Relative position formulation

To enhance spatial relationship modeling, a learnable relative position bias was introduced into the attention score matrix [19]. For an input image resized to 224×224 P=16, the patch grid size is G×G=14×14. For two patch tokens i and j, their grid coordinates are denoted as (u_i, v_i) and (u_j, v_j). The relative displacement between them is defined as:

$$\Delta u_{ij} = u_i - u_j, \quad \Delta v_{ij} = v_i - v_j \qquad (5)$$

where Δu_{ij} and Δv_{ij} describe the vertical and horizontal offset between two patches. Since G=14, both offsets fall within [−(G−1),G−1], namely [−13,13]. To retrieve the corresponding bias from a trainable table, the two-dimensional displacement is converted into a one-dimensional index:

$$r_{ij} = (\Delta u_{ij} + G - 1)(2G - 1) + (\Delta v_{ij} + G - 1) \qquad (6)$$

The learnable relative bias table is denoted as $B \in R^{(2G-1)^2 \times M}$, where M is the number of attention heads. For the h-th head, the relative bias between token i and token j is obtained as $B_{r_{ij},h}$. The attention calculation with relative position encoding is then formulated as:

$$A_h^{RPE} = Softmax\left(\frac{Q_h K_h^T}{\sqrt{d_h}} + B_h \right) V_h \qquad (7)$$

where $B_h \in R^{N \times N}$ is the relative bias matrix for the h-th attention head. Since the bias is added prior to the softmax, the attention weight distribution can be adjusted based on the displacement of the patch. In repetitive manufacturing textures, neighboring regions usually follow stable spatial arrangements [20]. When a stain, broken yarn, scratch, or local structural distortion appears, the relative relationship between adjacent patches changes. The proposed relative bias therefore provides an explicit spatial prior for identifying such local texture disruptions while preserving the global modeling capability of ViT.

Integration strategy

Relative position bias was inserted into every Transformer encoder block. Each of the L=12 encoder blocks maintained an independent bias table $T^{(l)}$, and the bias tables were not shared across layers. Within each table, separate bias values were learned for the M=6 attention heads, allowing different heads to model distinct spatial dependencies. Relative bias was calculated only for patch-to-patch interactions. The entries associated with class-token-to-patch, patch-to-class-token, and class-token-to-class-token attention were fixed at zero because the class token does not have a physical location on the two-dimensional patch grid [21].

Absolute positional embeddings were retained at the input stage to encode the global location identity of each patch, whereas the relative bias in each encoder block represented pairwise spatial displacement. All absolute and relative positional parameters were optimized jointly with the Transformer backbone. In the reported experiments, all images were resized to 224×224, resulting in a fixed 14×14 patch grid, and no interpolation was required. When an input resolution different from 224×224 is used, the patch-related absolute positional embeddings are reshaped into a two-dimensional grid and resized by bicubic interpolation. For each attention head, the learned relative-bias map is also reshaped from $(2G_H-1)\times(2G_W-1)$ and interpolated to the target size $(2G'_H-1)\times(2G'_W-1)$. The class-token embedding and all zero-valued class-token bias entries remain unchanged.

2.6 Classification head

A lightweight classification head was used to generate the final image-level prediction. The class token of the last Transformer encoder block is passed through the fully connected layer, and then the softmax activation is used to predict the defect or non-defect of the input image.

The classification head contained only one linear projection layer with an output dimension of 2. This simple structure was used to keep the comparison focused on the feature representation ability of the Transformer backbone and the effect of relative position encoding, rather than on additional classifier complexity.

2.7 Loss function

Since the task in this study was formulated as binary image-level classification, the training objective was defined using weighted cross-entropy loss:

$$L_{WCE}=-\frac{1}{B}\sum_{n=1}^{B}\sum_{c=0}^{1}w_c y_{n,c}\log\left(\hat{p}_{n,c}+\varepsilon\right) \quad (8)$$

where B is the mini-batch size, $y_{n,c}\in\{0,1\}$ is the one-hot ground-truth label of sample n for class c, and $\hat{p}_{n,c}$ is the probability predicted by the softmax layer. Class c=1 represents defective images, while c=0 represents non-defective images. The class weight was calculated only from the training subset as

$$w_c=\frac{N_{train}}{2N_c} \quad (9)$$

where N_{train} is the total number of training samples and N_c is the number of samples belonging to class c. For DAGM, $N_{train}=12880$, $N_0=11200$,and $N_1=1680$ yielding $[w_0,w_1]=[0.5750,3.8333]$. For the MVTec-derived subset, $N_{train}=1163$, $N_0=906$, and $N_1=257$, yielding $[w_0,w_1]=[0.6418,2.2626]$. The weights were computed before training and remained fixed across all epochs, comparison models, and five repeated runs.

The larger weight assigned to defective samples increases their contribution to the optimization objective and imposes a stronger penalty when a defective image receives a low positive-class probability. This setting reduces majority-class dominance and generally supports defect recall by discouraging false-negative predictions. A more defect-sensitive decision boundary may also increase false-positive predictions and thereby constrain precision. The weights were determined solely by training-set frequencies and were not tuned using validation or testing metrics. The constant ε=10−8 was used to maintain numerical stability in the logarithmic operation.

2.8 Experimental settings

All experiments were implemented using Python 3.10 and PyTorch 2.1. The training and testing was done on an NVIDIA RTX 4090 GPU with 24GB memory. AdamW was chosen as the optimizer due to its ability to decouple weight decay from gradient-based parameter updating, which is suitable for Transformer training. The initial learning rate was 1.0 × 10-4, and the weight decay was 1.0 × 10-4. This setting was used to keep parameter updates stable for medium-scale texture datasets while reducing overfitting in the classification head and attention layers.

Based on the GPU consumption at 224 x 224 and the ViT-Small backbone, the batch size was 16. The maximum number of training epochs has been set to 100. A cosine annealing learning-rate schedule was adopted to provide a relatively large update step in the early stage and a smoother convergence process in the later stage. Early stopping was applied when the validation AUROC did not improve for

15 consecutive epochs, preventing unnecessary training after convergence.

For the Vision Transformer backbone, the patch size was 16, the embedding dimension was 384, the number of encoder layers was 12, and the number of attention heads was 6. The relative-bias parameters were initialized with a truncated normal distribution with a standard deviation of 0.02. Each Transformer encoder block used its own relative-bias table, and each attention head learned an independent bias vector. The class-token-related entries in the complete attention-bias matrix were fixed at zero and were excluded from relative-displacement indexing. All experiments used a fixed input resolution of 224×224. For inference with another resolution, the input dimensions were adjusted to multiples of the patch size, and both patch absolute-position embeddings and two-dimensional relative-bias maps were resized by bicubic interpolation before attention computation. The patch size of 16 produces 196 image patches under a 224 × 224 input resolution, which preserves local texture details while avoiding an excessively long token sequence. The embedding dimension of 384 and 12 encoder layers follow the ViT-Small configuration and provide sufficient feature representation for image-level textured product inspection. Six attention heads make the dimension of each head equal to 64, which supports stable multi-head attention calculation with moderate computational cost.

The data sets were classified into three subsets: training, validation, and testing. For DAGM, an 80:10:10 image-level split was used. For the MVTec-derived textured subset, images were repartitioned within each category by stratified sampling to preserve the defective/non-defective ratio. All reported results were obtained from fixed data partitions. Preprocessing, data augmentation, and class-weight calculation were carried out using only the training data to avoid information leakage. The results of each experiment are reported as mean ± standard deviation.

The filename-level partitions were generated once using random seed 42 and remained unchanged across all baseline comparisons and ablation studies. Each model was trained five times on the same partition using initialization seeds 0, 1, 2, 3, and 4. Validation AUROC was used for early stopping and checkpoint selection, and the testing subset was evaluated once for each selected checkpoint. For visual error analysis, the run with validation AUROC closest to the median of the five runs was used. Its misclassified test images were separated into false-negative and false-positive groups, from which five representative samples covering both datasets, both error directions, and different texture appearances were selected. This post-hoc analysis was conducted after final testing and was not used for model selection or parameter adjustment.

Image resizing, channel conversion, and deterministic normalization were applied to all subsets, while random horizontal and vertical flipping, rotation, brightness adjustment, and contrast adjustment were restricted to the training subset. Normalization statistics were calculated separately for each dataset from resized, unaugmented training images after channel conversion; ImageNet statistics were not used. The resulting fixed statistics were applied to all three subsets. The same split manifests, preprocessing sequence, augmentation parameters, optimizer settings, and evaluation code were used for all compared models.

Algorithm 1. Training procedure of the proposed ViT-RPE framework

Input: Training set D_{train}, validation set D_{val}, image size 224×224, patch size P=16, encoder depth L=12, attention heads M=6, maximum epoch E=100.
Output: Optimized ViT-RPE parameter count Θ*.

1. Initialize patch projection parameters, absolute positional embeddings, class token, Transformer encoder parameters, relative bias table B, and classification head.

2. For each epoch e=1,2,…,E:

3. For each mini-batch $\{X_n, y_n\}_{n=1}^{N_b}$ from D_{train}:

4. Resize X_n, replicate the grayscale channel when required, apply online geometric and photometric augmentation, convert the image into a tensor, and normalize it using the fixed training-set statistics.

5. Divide X_n into 16×16 non-overlapping patches.

6. Project patches into token embeddings and add absolute positional embeddings.

7. For each Transformer encoder block:

8. For each encoder block l and attention head h, calculate $Q^{(l,h)}$, $K^{(l,h)}$, and $V^{(l,h)}$ from $Z^{(l,h)}$.

9. Calculate $(\Delta u_{ij}, \Delta v_{ij})$ and index r_{ij} for all patch-token pairs.

10. Retrieve $b_{ij}^{(l,h)}$ from the layer-specific and head-specific table $T^{(l)}$, and set all class-token-related bias entries to zero.

11. Construct $\widetilde{B}^{(l,h)}$ and calculate $O_{RPE}^{(l,h)}$ according to Equation (7).

12. Feed the final class token into the classification head.

13. Calculate weighted cross-entropy loss L_{CE}.

14. Update parameter count with AdamW.

15. Evaluate AUROC and F1-score on D_{val}.

16. Apply cosine annealing learning-rate update.

17. If validation AUROC does not improve for 15 consecutive epochs, stop training.

18. Return the parameter count Θ* with the best validation AUROC.

2.9 Evaluation metrics

Accuracy, precision, recall, F1-score, and the area under the receiver operating characteristic curve (AUROC) were used to evaluate image-level defect classification. Defective images were treated as the positive class, and non-defective images were treated as the negative class. Accuracy is defined as

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

Precision is defined as

$$\text{Precision} = \frac{TP}{TP+FP} \tag{11}$$

Recall is defined as

$$\text{Recall} = \frac{TP}{TP+FN} \tag{12}$$

F1-score is defined as

$$\text{F1} = \frac{2\times\text{Precision}\times\text{Recall}}{\text{Precision}+\text{Recall}} \tag{13}$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives, respectively. AUROC is calculated as the area under the receiver operating characteristic curve:

$$AUROC = \int_0^1 TPR(x)dx \tag{14}$$

where x denotes the false-positive rate and $TPR(x)$ denotes the corresponding true-positive rate at different classification thresholds.

3. Results and Discussion

3.1 Classification results on the DAGM dataset

Table 2 summarizes the image-level classification results on the DAGM dataset. ViT-RPE achieved the best overall performance, with an accuracy of 0.954 ± 0.004, a recall of 0.956 ± 0.004, an F1-score of 0.953 ± 0.004, and an AUROC of 0.977 ± 0.003. Compared with the baseline ViT, its accuracy and AUROC increased by 0.009 and 0.008, respectively.

Among the newly added Transformer baselines, Swin-T achieved an accuracy of 0.950 ± 0.005 and an AUROC of 0.974 ± 0.003, while MobileViT-XS obtained 0.943 ± 0.006 and 0.967 ± 0.004. ViT-RPE exceeded Swin-T by 0.004 in accuracy and 0.003 in AUROC, indicating that global patch interaction combined with explicit relative displacement modeling remains effective for repetitive textures. MobileViT-XS provided lower computational complexity, although its reduced representation capacity led to weaker classification performance. The performance differences were moderate but remained consistent across the five repeated runs.

Table 2. Image-level classification performance on the DAGM dataset

Method	Accuracy	Precision	Recall	F1-score	AUROC
ResNet-18	0.919 ± 0.008	0.914 ± 0.009	0.918 ± 0.007	0.916 ± 0.008	0.946 ± 0.005
Activation-embedded CNN	0.931 ± 0.006	0.935 ± 0.007	0.924 ± 0.006	0.929 ± 0.006	0.956 ± 0.004
MobileViT-XS	0.943 ± 0.006	0.941 ± 0.006	0.944 ± 0.006	0.942 ± 0.006	0.967 ± 0.004
Baseline ViT	0.945 ± 0.005	0.942 ± 0.005	0.940 ± 0.006	0.941 ± 0.005	0.969 ± 0.003
Swin-T	0.950 ± 0.005	0.948 ± 0.005	0.951 ± 0.005	0.949 ± 0.005	0.974 ± 0.003
Proposed ViT-RPE	0.954 ± 0.004	0.951 ± 0.005	0.956 ± 0.004	0.953 ± 0.004	0.977 ± 0.003

3.2 Classification results on the MVTec-derived textured subset

The classification results on the MVTec-derived texture subset are shown in Table 3. ViT-RPE achieved an accuracy of 0.939 ± 0.005, a precision of 0.936 ± 0.006, a recall of 0.941 ± 0.005, an F1-score of 0.938 ± 0.005, and an AUROC of 0.969 ± 0.004. Compared with the baseline ViT, the proposed model improved accuracy by 0.013 and AUROC by 0.012.

Swin-T produced the strongest result among the added baselines, reaching an accuracy of 0.934 ± 0.005 and an AUROC of 0.965 ± 0.004. ViT-RPE retained improvements of 0.005 and 0.004 over Swin-T on the two metrics.

MobileViT-XS achieved an accuracy of 0.924 ± 0.006 and an AUROC of 0.956 ± 0.004, reflecting the trade-off between compact model size and recognition performance. The lower results relative to DAGM are associated with the larger appearance variation, less regular texture structures, and more diverse defect patterns in the selected MVTec categories. The comparison indicates that the proposed relative-position design remains competitive against both hierarchical and lightweight Transformer architectures under the same supervised classification protocol.

The MVTec-derived experiment represents a closed-set supervised image-level classification task constructed by reorganizing both normal and defective images from the original MVTec AD directories. All comparison models in Table 3 were retrained using the same filename-level partitions, preprocessing pipeline, and binary labels. Accordingly, the results support only internal comparisons under the present supervised protocol. They should not be directly compared with published MVTec AD results obtained under one-class training, unsupervised anomaly detection, few-shot adaptation, or pixel-level localization settings.

Table 3. Image-level classification performance on the MVTec-derived textured subset

Method	Accuracy	Precision	Recall	F1-score	AUROC
ResNet-18	0.899 ± 0.008	0.893 ± 0.009	0.901 ± 0.008	0.897 ± 0.008	0.934 ± 0.005
Activation-embedded CNN	0.911 ± 0.007	0.916 ± 0.007	0.905 ± 0.008	0.910 ± 0.007	0.944 ± 0.005
MobileViT-XS	0.924 ± 0.006	0.920 ± 0.007	0.928 ± 0.006	0.924 ± 0.006	0.956 ± 0.004
Baseline ViT	0.926 ± 0.006	0.921 ± 0.006	0.930 ± 0.007	0.925 ± 0.006	0.957 ± 0.004
Swin-T	0.934 ± 0.005	0.931 ± 0.006	0.936 ± 0.005	0.933 ± 0.005	0.965 ± 0.004
Proposed ViT-RPE	0.939 ± 0.005	0.936 ± 0.006	0.941 ± 0.005	0.938 ± 0.005	0.969 ± 0.004

3.3 Cross-dataset ablation study on position encoding strategy

To evaluate the contribution and cross-dataset consistency of positional information, four configurations were tested on both DAGM and the MVTec-derived textured subset: no positional encoding, absolute positional encoding only, relative positional encoding only, and combined absolute and relative positional encoding. All configurations used the same backbone, data partitions, training parameters, and five initialization seeds. The results are reported in Table 4.

On DAGM, removing all positional information produced the lowest accuracy and AUROC of 0.929±0.007 and 0.951±0.005, respectively. Absolute positional encoding increased the two metrics to 0.938±0.006 and 0.960±0.004, while relative positional encoding achieved 0.949±0.005 and 0.973±0.003. Combining both forms of positional information yielded the highest accuracy of 0.954±0.004 and AUROC of 0.977±0.003.

The MVTec-derived subset showed the same ranking. The configuration without positional encoding achieved an accuracy of 0.903±0.008 and an AUROC of 0.938±0.006. Absolute positional encoding increased these values to 0.926±0.006 and 0.957±0.004, and relative positional encoding further improved them to 0.934±0.005 and 0.965±0.004. The combined configuration obtained the best accuracy of 0.939±0.005 and AUROC of 0.969±0.004, exceeding the absolute-only setting by 0.013 and 0.012 and the relative-only setting by 0.005 and 0.004, respectively.

The identical ordering across the two datasets indicates that absolute and relative positional information provide complementary functions. Absolute embeddings preserve the global location identity of individual patches, whereas relative bias represents pairwise spatial displacement and local texture arrangement. Their combination consistently improved accuracy, recall, F1-score, and AUROC under both regular synthetic textures and more appearance-diverse real texture categories. Although the numerical gains over relative encoding alone were moderate, the cross-dataset results support the stability of the combined positional design.

Table 4. Ablation study on position encoding strategy (DAGM)

Setting	Accuracy	Precision	Recall	F1-score	AUROC
No position encoding	0.929 ± 0.007	0.924 ± 0.008	0.931 ± 0.007	0.927 ± 0.007	0.951 ± 0.005
Absolute position	0.938 ± 0.006	0.936 ± 0.006	0.934 ± 0.007	0.935 ±	0.960 ± 0.004

encoding			0.006	
Relative position encoding	0.949 ± 0.005	0.946 ± 0.005	0.950 ± 0.005	0.948 ± 0.005	0.973 ± 0.003
Absolute + relative position encoding	0.954 ± 0.004	0.951 ± 0.005	0.956 ± 0.004	0.953 ± 0.004	0.977 ± 0.003

3.4 Computational cost analysis

For industrial inspection, computational efficiency is also important. Table 5 compares the complexity of the Transformer-based models under the same 224 × 224 input resolution, batch size of 1, and hardware environment. MobileViT-XS had the lowest computational demand, with 2.3 M parameters and 0.70 G FLOPs, while Swin-T required 28.3 M parameters and 4.49 G FLOPs. The baseline ViT contained 22.1 M parameters and required 4.62 G FLOPs.

Introducing relative position bias increased the parameter count from 22.1 M to 22.4 M and the FLOPs from 4.62 G to 4.68 G. The inference time increased from 13.8 ms to 14.5 ms per image. ViT-RPE was 2.2 ms faster than Swin-T and obtained higher classification performance on both datasets. Although MobileViT-XS provided the highest inference efficiency, its accuracy and AUROC were lower. These results indicate that ViT-RPE achieves a balanced trade-off between representation accuracy and computational cost for image-level manufacturing defect pre-screening.

Table 5. Compare the complexity of the baseline ViT with the proposed approach

Method	Parameters (M)	FLOPs (G)	Inference time per image (ms)
MobileViT-XS	2.3	0.70	8.4
Baseline ViT	22.1	4.62	13.8
Swin-T	28.3	4.49	16.7
Proposed ViT-RPE	22.4	4.68	14.5

3.5 Visual error analysis and limitations

The results support the assumption that textured defects can be better classified when both global context and relative spatial relationships are considered. The moderate performance gain is consistent with the lightweight modification introduced into the ViT backbone. Since the baseline Transformer already captures global contextual information, the relative position bias mainly provides additional local spatial cues for identifying subtle texture disruptions. The inverse-frequency class weights further prevented the majority non-defective class from dominating the training objective. Their higher values for defective samples placed greater emphasis on false-negative tendencies and supported recall, while increased sensitivity to positive predictions could affect precision through additional false positives. Because no class-weight ablation was conducted, this effect is interpreted from the optimization mechanism rather than treated as an independently measured performance gain.

Several limitations remain. First, this work only evaluates image-level classification and does not examine defect localization. Second, the MVTec-derived experiment was used as a supplementary supervised classification test, rather than the standard MVTec anomaly detection protocol. Third, the validation was based on public datasets, and a larger real manufacturing datasets from textile, leather-like material, ceramic tile, and patterned component production has not yet been included. In addition, the reported differences were obtained under the current data split and random-seed settings. More repeated experiments, category-level analysis, statistical testing, and visualization results would help further verify the robustness of the conclusions.

Future work will focus on combining image-level screening with localization-oriented prediction, testing the method on larger real textile datasets, and comparing it with recent lightweight Transformer-based inspection models.

4. Conclusion

A Vision Transformer-based framework was developed for image-level textured surface defect classification in manufacturing inspection. Moreover, a learnable relative position bias was added to the self-attention mechanism to help the model capture the patch-to-patch spatial relationships in the repetitive product textures. This design improves the representation of local structural changes while retaining the global contextual modeling ability of the Transformer backbone.

Experiments with DAGM 2007 dataset and supervised MVTec-derived texture subsets have shown that the proposed approach is superior to the comparison models. On DAGM, it achieved an accuracy of 0.954 ± 0.004 and an AUROC of 0.977 ± 0.003. On the MVTec-derived subset, the accuracy and AUROC reached 0.939 ± 0.005 and 0.969 ± 0.004, respectively. The ablation results also showed that combining absolute and relative positional information was more effective than using either strategy alone. The additional computational cost was limited,

which indicates that the method is suitable for image-level pre-screening in automated product inspection, textile manufacturing, textured material processing, and production-line quality sorting.

There are still several limitations to note. The current experiments focus on image-level classification and do not evaluate defect localization. The MVTec-derived experiment was also conducted under a supervised auxiliary protocol rather than the original anomaly-detection setting. In future work, the method will be further tested on larger real manufacturing datasets, combined with localization-oriented prediction, and compared with more lightweight Transformer-based inspection models for process monitoring, repair decision support, and downstream quality traceability.

References

[1] Kahraman Y, Durmuşoğlu A. Deep learning-based fabric defect detection: A review[J]. Textile Research Journal, 2023, 93(5-6): 1485-1503.

[2] Smith A D, Du S, Kurien A. Vision transformers for anomaly detection and localisation in leather surface defect classification based on low-resolution images and a small dataset[J]. Applied sciences, 2023, 13(15): 8716.

[3] Hassan S A, Beliatis M J, Radziwon A, et al. Textile fabric defect detection using enhanced deep convolutional neural network with safe human–robot collaborative interaction[J]. Electronics, 2024, 13(21): 4314.

[4] Carrilho R, Yaghoubi E, Lindo J, et al. Toward automated fabric defect detection: A survey of recent computer vision approaches[J]. Electronics, 2024, 13(18): 3728.

[5] Tabernik D, Šela S, Skvarč J, et al. Segmentation-based deep-learning approach for surface-defect detection. Journal of Intelligent Manufacturing. 2020;31:759–776. doi:10.1007/s10845-019-01476-x.

[6] Mewada H, Pires I M, Engineer P, et al. Fabric surface defect classification and systematic analysis using a cuckoo search optimized deep residual network[J]. Engineering Science and Technology, an International Journal, 2024, 53: 101681.

[7] Aksakalli I K, Demir K, Sokmen O. A hybrid PatchNet-Attention based deep learning architecture for multi-type fabric defect classification in textile manufacturing and quality control[J]. Engineering Science and Technology, an International Journal, 2025, 72: 102231.

[8] Machado R, Barros L A M, Vieira V, et al. Textile defect detection using artificial intelligence and computer vision—a preliminary deep learning approach[J]. Electronics, 2025, 14(18): 3692.

[9] Erdogan M, Dogan M. Enhanced curvature-based fabric defect detection: a experimental study with gabor transform and deep learning[J]. Algorithms, 2024, 17(11): 506.

[10] Geze R A, Akbaş A. Detection and classification of fabric defects using deep learning algorithms[J]. Politeknik Dergisi, 2023, 27(1): 371-378.

[11] Liu Z, Lin Y, Cao Y, et al. Swin Transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021:10012–10022. doi:10.1109/ICCV48922.2021.00986.

[12] Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Vedaldi A, Bischof H, Brox T, Frahm J M, eds. Computer Vision – ECCV 2020. Cham: Springer; 2020:213–229. doi:10.1007/978-3-030-58452-8_13.

[13] Batzner K, Heckler L, König R. Efficientad: Accurate visual anomaly detection at millisecond-level latencies[C]//Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2024: 128-138.

[14] Hyun J, Kim S, Jeon G, et al. Reconpatch: Contrastive patch representation learning for industrial anomaly detection[C]//Proceedings of the IEEE/CVF winter conference on applications of computer vision. 2024: 2052-2061.

[15] Bae J, Lee J H, Kim S. PNI: Industrial anomaly detection using position and neighborhood information[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2023: 6373-6383.

[16] Ruan J, He J, Tong Y, et al. Knowledge Embedding Relation Network for Small Data Defect Detection[J]. Applied Sciences, 2024, 14(17): 7922.

[17] Ozek A, Seckin M, Demircioglu P, et al. Artificial intelligence driving innovation in textile defect detection[J]. Textiles, 2025, 5(2): 12.

[18] Cauteruccio F, Marchetti M, Traini D, et al. Adaptive patch selection to improve Vision Transformers through Reinforcement Learning: F. Cauteruccio et al[J]. Applied Intelligence, 2025, 55(7): 607.

[19] Wu K, Peng H, Chen M, et al. Rethinking and improving relative position encoding for vision transformer[C]//Proceedings of the IEEE/CVF international conference on computer vision. 2021: 10033-10041.

[20] Ma X, Zhang Z, Yu R, et al. SAVE: Encoding spatial interactions for vision transformers[J]. Image and Vision Computing, 2024, 152: 105312.

[21] Heo B, Park S, Han D, et al. Rotary position embedding for vision transformer[C]//European Conference on Computer Vision. Cham: Springer Nature Switzerland, 2024: 289-305.

[22] Jeong J, Zou Y, Kim T, et al. Winclip: Zero-/few-shot anomaly classification and segmentation[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2023: 19606-196