Academic Journal of Science and Technology
ISSN: 2771-3032 | Vol. 7, No. 3, 2023

Investigating Lightweight Transformer Models for Defect Detection

Hanyun Wang

School of Electrical Engineering and Information, Southwest Petroleum University, Sichuan 610500, China

Abstract: In industrial production, product defect detection is vital for quality control. Traditional manual inspection is inefficient and error-prone. Deep learning, particularly in image processing, has enabled computer-based automated defect detection. This paper proposes a Visual Transformer-based model to overcome limitations in industrial anomaly detection. Leveraging pretrained Vision Transformer and Point Transformer models, it extracts features from RGB images and point cloud data. Multimodal feature fusion enhances anomaly perception, with residual connections mitigating feature loss. On the MVTec AD dataset, it achieves 96.3% AU PRO for anomaly detection and 99.3% Pixel ROCAUC for anomaly segmentation. To enable deployment on devices like Raspberry Pi, the paper introduces a lightweight model via post-training quantization and pruning. This results in a 28.52% inference speedup with only a 1.08% average detection accuracy drop, facilitating practical industrial applications on compact devices.

Keywords: Image processing; Anomaly detection; Vision Transformer; Quantization; Pruning.

1. Introduction

Industrial anomaly detection combines anomaly detection techniques with manufacturing processes to promptly identify irregularities, such as surface defects, in industrial production, enhancing product quality and efficiency. Traditional quality control methods, relying on random sampling, often miss defects, leading to substantial losses. For example, between January 2021 and May 2022, automakers like Huachen BMW and Mercedes-Benz (China) had to recall 266,203 vehicles due to various defects. Implementing industrial anomaly detection, by scanning product images, offers a cost-effective way to achieve comprehensive defect detection and improve quality standards.[1]

In current image-based industrial anomaly detection, two factors hinder scalable applications: first, the accuracy of detection is often insufficient, requiring manual verification. For example, in recent surface defect detection for steel strips,[2] the mAP_0.5 is only 0.79. Second, detection models tend to be large, demanding high-performance hardware, incurring high costs, and operating slowly, making them unsuitable for real-time applications. For instance, the widely used Vision Transformer model, VIT-L[3] , has 307MB parameters and 190.7G FLOPs, requiring 6.5GB of memory for inference. Consider the popular micro-terminal Raspberry Pi 3b+, which uses a BCM2837B0 chip with a CPU@1.4GHz and supports a maximum of 2GB of memory; it cannot support such large models. Thus, in practical industrial anomaly detection, apart from improving model accuracy, lightweighting models to reduce reliance on hardware computational performance is crucial.

Supervised learning-based industrial anomaly detection falls into two main categories: Object Detection and Pixel Segmentation. Object Detection treats anomalies as targets to segment and recognize using modified classic detection algorithms. Pixel Segmentation aims to refine defect boundaries at the pixel level for higher accuracy. While supervised methods perform well with ample training data, many anomaly detection scenarios lack such data. Data augmentation techniques like GANs are used, but they introduce distribution differences and offer limited performance gains. Additionally, supervised methods are constrained by known anomaly types, making them less adaptable to diverse and evolving industrial anomaly detection challenges where anomaly types can vary significantly.[4]

Unsupervised learning provides a notable advantage over supervised learning by circumventing the need for extensive manual labeling, thereby reducing development costs. It also excels in handling the diverse and ever-changing anomaly types encountered in industrial defect detection. Unlike supervised methods, which often miss anomalies not covered by predefined labels, unsupervised techniques adapt by identifying anomalies based on discrepancies between defect and normal data, without relying on predefined criteria. Unsupervised methods in industrial anomaly detection can be broadly categorized into traditional machine learning and deep learning approaches. They have garnered attention due to their exceptional feature capturing capabilities, eliminating the need for labor-intensive feature engineering. [5]

Ruff and colleagues [6] introduced an anomaly detection method that combines deep learning with Support Vector Data Description (SVDD). This approach maps normal samples into clusters in a feature space and determines whether a point is an anomaly based on its distance from the cluster center. However, this method requires manual specification of cluster centers during clustering, introducing uncertainty into its effectiveness. These methods excel in detecting anomalies by assessing the differences between input samples and those generated by the model, making the model's performance contingent on the quality of the generated images.

2. Organization of the Text

2.1. A Visual Transformer-Based Defect Detection Model

2.1.1. Schematic Diagram

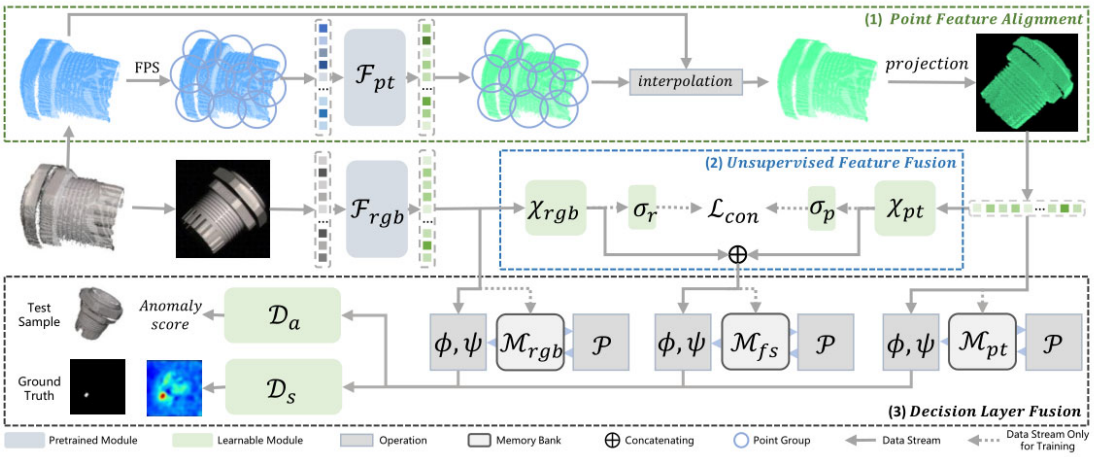

Figure 1. Schematic Diagram of the Visual Transformer-Based Defect Detection Model

The Industrial Anomaly Detection Model based on Visual Transformer takes both 3D point clouds and RGB images as inputs for anomaly detection and segmentation. To address the issue of misalignment between 3D point cloud features and 2D RGB features, this paper employs a point feature alignment method to align the 3D features onto a 2D plane, facilitating feature fusion. However, since 3D point cloud features and RGB features have different distributions, directly concatenating these two types of features using conventional methods would introduce interference between different-dimensional features, severely degrading model performance. Therefore, an unsupervised feature fusion method is used, which employs contrastive loss at the block level to train the feature fusion module.[7]

To tackle feature loss during the fusion process, this paper proposes the use of multiple memory banks to store original color, position, and fusion feature information during training. These memory banks are utilized in the final decision-making process along with the fused multimodal features to calculate anomaly scores, enhancing the model's robustness and generalization.

To improve the model's robustness and generalization, pretrained Vision Transformer [8] and Point Transformer [9] models are adopted as feature extractors for extracting 2D RGB and 3D point cloud features, respectively. The schematic diagram of the Industrial Anomaly Detection Model based on Visual Transformer is shown in Figure.1. Fpt represents the Point Transformer module for extracting 3D point cloud features, while Frgb is the Vision Transformer module for extracting 2D RGB features. The Point Feature Alignment module aligns 3D point cloud features, and its output, along with the output of Frgb, undergoes feature fusion through the Unsupervised Feature Fusion module. During fusion, multiple memory banks are simultaneously generated.

2.1.2. Feature Extractors

As shown in the schematic diagram in Figure.2, it illustrates the structure of the Point Transformer model, which serves as the feature extractor for 3D point cloud data. On the left side, you can see the process of point cloud feature masking and embedding. It involves partitioning the input point cloud data into blocks, applying random masks, and embedding them. This effectively creates partially incomplete point cloud data, which is used for self-supervised training. On the right side, there's the pretraining process with an autoencoder. In this process, mask labels are added to the input sequence of the decoder to reconstruct the masked blocks, facilitating pretraining.[10]

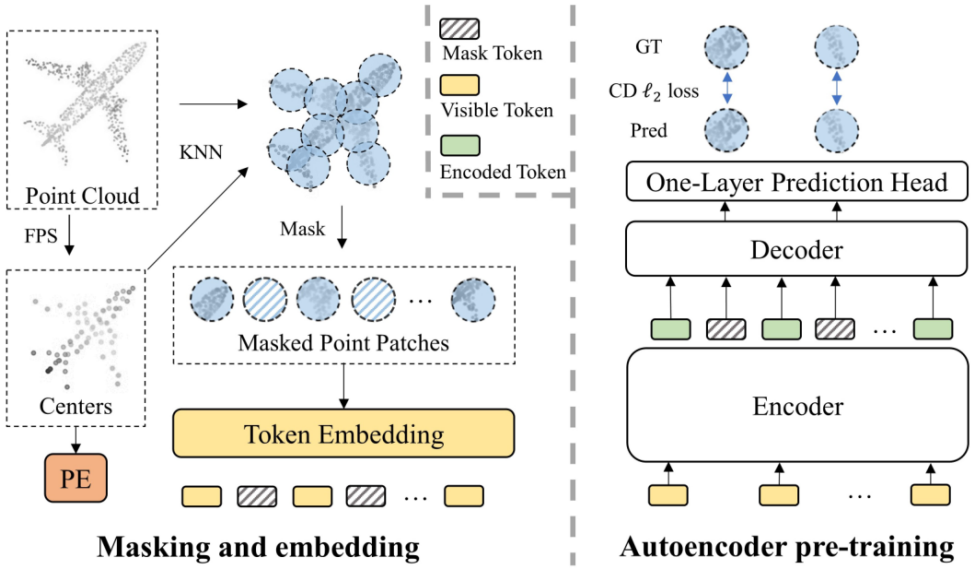

Figure 2. Schematic Diagram of the Point Transformer Model Structure

In recent years, Transformers have emerged as an alternative to convolutional neural networks (CNNs) in the field of image processing. Many Transformer models adopt training strategies similar to those used in natural language processing, involving pretraining on large datasets and fine-tuning on target data for domain-specific applications. However, research has shown that this approach does not necessarily yield significant improvements over CNNs. Instead, it often requires more computational power and data due to the extensive use of Transformer structures.The Vision Transformer model employed in this paper takes a different approach. It is trained using self-supervised methods on a large amount of data. Experimental results indicate that it outperforms traditional supervised pretraining models.[11]

The structure of the Vision Transformer model consists of a teacher network and a student network, forming a knowledge distillation-based self-supervised network. Here, 'x' represents the input image, and 'x1' and 'x2' are random transformations of the input image 'x'. Both the teacher and student networks share the same architecture but have different parameters. The teacher network's output is centered around the batch-wise mean. The output features of each network are normalized using the softmax function, and their similarity is measured using the cross-entropy loss function. Knowledge distillation is achieved by applying the stop-gradient (sg) operator to the teacher network, allowing gradient propagation and updates only through the student network. The teacher network's parameters are then updated using the moving average of the student network's parameters.

2.1.3. Feature Alignment

The English translation for the provided text is as follows:The input point cloud feature, denoted as "p," consists of a sequence of N points. After undergoing farthest point sampling [12], the feature points are divided into M groups, with each group containing S points. Subsequently, a point transformer encodes each point within each group into a feature vector. The output "g" of the point transformer consists of M features, organized into point feature groups, with each group having a single-point feature. Finally, these feature vectors are fed into the Point Transformer for feature extraction.Due to the non-uniform spatial distribution of point features after farthest point sampling, there is an issue of uneven feature density. To address this problem, the original features are interpolated with the features extracted by the Point Transformer. This interpolation is performed for each point in the input point cloud, utilizing inverse distance weighting based on the M point features and M group center points.

2.2. Metrics

2.2.1. Assessment Criteria

The concept of a "confusion matrix," which includes four values: True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN), is an indispensable aspect of evaluation metrics. For industrial anomaly detection, evaluation metrics can be categorized into classification performance indicators and segmentation performance indicators. In practical applications, there is also a focus on the model's execution performance, which typically reflects the model's practicality. Therefore, certain performance metrics are needed to measure the model.

Detection classification metrics. Detection classification performance metrics include accuracy, recall, and precision. Accuracy measures the proportion of correctly classified samples to all samples. Recall represents the proportion of all detected defect samples to the total number of samples, while precision measures the proportion of correctly predicted defect samples.In practical applications, we are more concerned with the proportion of undetected anomalies among all anomaly samples and the proportion of wrongly detected samples among those classified as anomalies. This directly relates to the practical usability of our model. Typically, the false negatives rate (FNR) is used to represent the former, while the false positive rate (FPR) represents the latter. When using evaluation metrics, relying on a single metric can sometimes yield misleading results. For example, using FNR as the sole criterion would encourage a model to classify all samples as anomalies to achieve an FNR of 0. Therefore, evaluating a model often requires a comprehensive approach.The use of the AUC-ROC curve helps balance the trade-off between TPR and FPR. ROC refers to the curve between TPR and FPR, and AUC represents the area under the ROC curve. When the area under the curve is maximized, it indicates the model's best trade-off between detecting anomalies and avoiding false alarms.

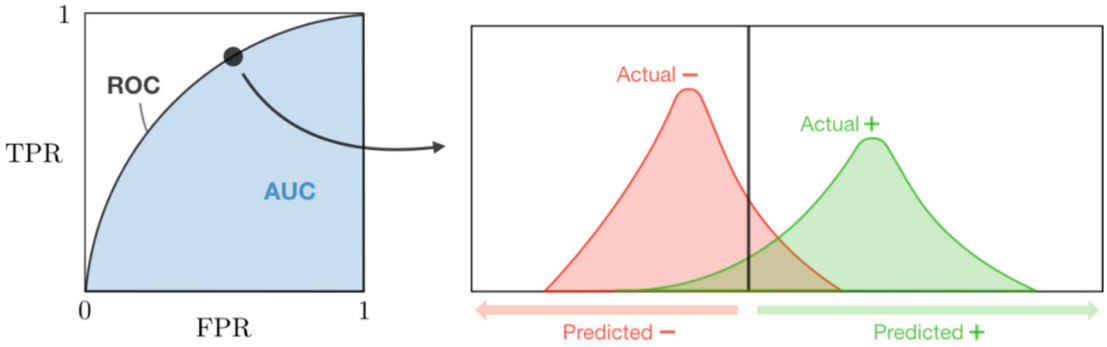

Figure 3. Balanced ROC-AUC Plot

As shown in the figure below (Figure.3), changes in the position of points on the ROC curve represent the model's balance between false negatives and false positives.

Anomaly Segmentation Metrics.For anomaly detection tasks, in addition to determining whether a given image contains anomalies, another crucial task is to locate and segment the anomalous regions. Since the TPR values in the

ROC curve are weighted by the defect area, a larger correctly segmented region can significantly compensate for the quantitative metrics of multiple small defective areas. Therefore, this metric has limitations in industrial defect detection. Bergmann et al. [13] proposed a region-level metric called PRO (Per Region Overlap). It divides the ground truth regions and predicted regions into N regions based on connected components. Then, it calculates the intersection between each actual defect region Gn and the defect region P, as well as the ratio of this intersection to the GT region Gn. Finally, the PRO metric is obtained by weighted averaging according to N connected components.

Model Performance Metrics.In practical industrial anomaly detection applications, due to cost constraints, high-performance computers are typically not available for model inference. Moreover, while maintaining a certain level of detection performance, the practicality of a model improves as its hardware resource requirements decrease. Therefore, it is necessary to incorporate model performance metrics.For image processing models, the most straightforward performance metric is the model's Inference Time (IT). It represents the time it takes for the model to process an input sample and produce an output prediction, with smaller IT values indicating faster inference speed and higher real-time capability. Given that image processing models often have high memory requirements, another important metric is Memory Occupied (MO), which denotes the amount of memory the model consumes during inference, with smaller MO values indicating a more lightweight model.[14]

2.3. Model lightweighting methods

2.3.1. Model Lightweighting

In the Transformer model, the parts that need to be quantized include the linear layer, self-attention layer, LayerNorm layer, and softmax layer. Different quantization methods are applied to different layers to achieve full quantization of the entire structure.In the Transformer, its input is a one-dimensional token embedding sequence. Therefore, it needs to reshape the image into a two-dimensional sequence, where H is the height of the image, W is the width of the image, C is the number of channels in the image, and p represents the resolution of the image, which is the effective sequence length of the Transformer. Since the Transformer uses a constant width across all layers, a trainable linear projection maps each vectorized block to the model dimension d.[15]

For the MLP layer and MSA (Multi-Head Self-Attention) layer, their computational cost comes from the large matrix multiplications, so the goal of quantization here is to convert float matrices into integer ones.LayerNorm has dynamic computation characteristics, which means it cannot be folded into the previous layer, so it needs to be quantized separately. Due to the significant inter-channel variations in LayerNorm, different quantization factors need to be applied to different channels.[16]

As model performance depends on smaller image block sizes and higher resolutions, when the resolution increases and the image block size decreases, the storage and computation of attention maps become a bottleneck limiting the model's performance, directly affecting the throughput and latency of inference. Therefore, it's necessary to create smaller attention maps to improve the model's inference performance.[17]

To compress attention maps into a smaller range for faster inference, attention maps need to be quantized to lower bit-widths. Since the output range of Softmax is fixed in the (0, 1) interval, quantizing Softmax using the log2 function does not require calibration, and log2 quantization can preserve the order consistency between full precision and quantized attention maps.[18]

2.3.2. Pruning-based model lightweighting method

Pruning is an effective method to reduce the enormous inference cost of Transformer models. However, most works on Transformer pruning require retraining the model, which can increase training costs and complexity. Therefore, this paper is based on a fast post-training pruning framework proposed by Kwon et al. [19], which uses a design space exploration auto-searcher for mask fine-tuning to achieve automatic mask finding. This pruning method does not require any further training. Given resource constraints and a sample dataset, this method automatically prunes the Transformer model using structured sparsity techniques. The method mainly consists of three steps:

(1) Lightweight mask search algorithm based on Fisher information to find heads and filters to be pruned.

(2) Mask reshuffling as a complement to the search algorithm.

(3) Mask fine-tuning through a design space exploration auto-searcher.

During the process of Fisher-based mask search and mask reshuffling, the range of mask values is initially restricted to the (0, 1) interval to simplify computations. Obviously, this simplification significantly reduces the pruning accuracy while simplifying inference calculations. Therefore, in this section, this limitation is removed, and the mask value range is expanded to the entire real number domain to fine-tune the mask for improved pruning accuracy.

Existing research, such as that by He et al. [20], suggests minimizing the mask error layer by layer through linear least squares. For each layer, this method reconstructs the output loss of the original model using the remaining heads and filters in the pruned model. This method can reduce cumulative errors, but when the matrix size is large, numerical instability can occur when solving the linear least squares problem. Lin Shuyuan et al. [21] propose using a non-negative matrix under-approximation method to estimate model parameters. This approach optimizes the preference matrix through spatial constraints and sparse constraints. However, when the preference matrix does not have non-negativity, this method cannot be used for optimization.

To address these issues, this paper suggests matching optimization methods for mask fine-tuning using an auto-searcher to find the best fine-tuning method. For the selected optimization method, the searcher first applies it to mask blocks for fine-tuning and then evaluates the model's loss after fine-tuning. Through multiple iterations, the final mask fine-tuning method can be determined. Finally, this method is applied to optimize the mask fine-tuning for the pruned model.

3. Lightweight Experiment

3.1. Pruning Method Experiment and Analysis

The following table, Table.1, displays the experimental results of pruning conducted on the RGB feature extractor Vision Transformer. Pruning is applied exclusively to the weight matrices of the RGB feature extractor, without any pruning or quantization applied to the other half of the model. Pruning experiments are conducted with Flops as the

constraint, where a Flops constraint of 1.0 represents the baseline model without any pruning. In this context, "inference time" refers to the time taken from the input to the pruning module to its output, not the overall model's inference runtime.

Table 1. The experiment results for pruning under different Flops constraints

Flops constraint	0.5	0.6	0.7	0.8	0.9	1.0
Inference time(s)	7.21	7.35	8.23	8.45	9.01	9.48
AU PRO	0.868	0.908	0.917	0.944	0.967	0.967
Image ROCAUC	0.798	0.804	0.831	0.836	0.867	0.884
Pixel ROCAUC	0.915	0.929	0.956	0.977	0.983	0.991

The experimental results from Table.1 show a positive correlation between Flops constraints and the pruning effect. As the Flops constraint increases, the pruned model's inference performance improves, but correspondingly, the inference time also increases. Essentially, this is because the growth in Flops constraints continuously weakens the pruning intensity, resulting in fewer pruned weights and thus improving inference performance. However, it is well-known that pruning operations inevitably lead to a decrease in model detection performance. Therefore, the improvement in model detection performance here is not due to the positive effects of pruning but rather a reduction in the negative effects caused by pruning.

Selecting an appropriate Flops constraint for pruning algorithms involves striking a balance between the model's inference performance and inference speed. To better understand this relationship, the model's inference speed and inference performance trends are plotted against Flops constraints, as shown in Figure.4. Since there is a difference of one order of magnitude between the model's inference time and inference performance, the inference speed is scaled down by a factor of 10 in the trend graph to better observe the changes in trends.

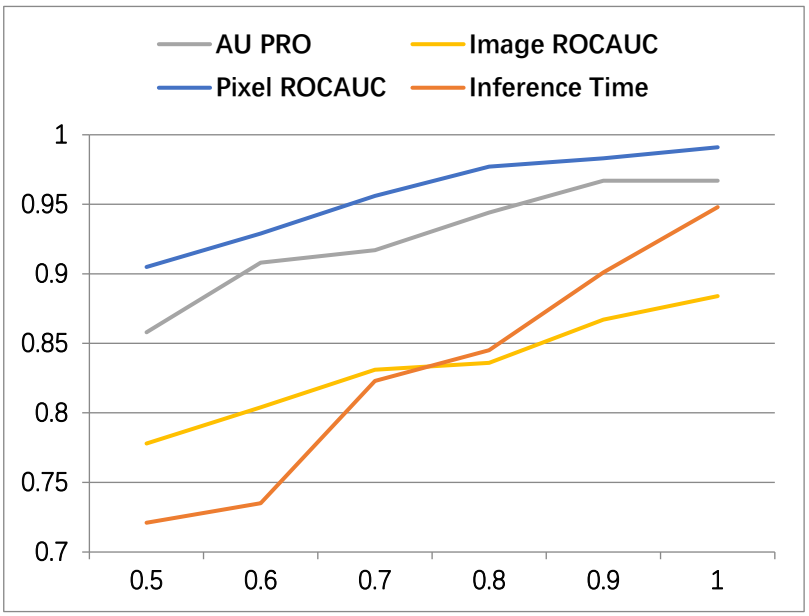

Figure 4. The comparison of pruning performance and its effects on performance trends

3.2. The experiments and analysis of the quantization method

Table 2. The experiment results for pruning under different Flops constraints

Methods	AU PRO	Image ROCAUC	Pixel ROCAUC	Inference Time(s)
Full Precision	0.967	0.884	0.991	5.84
MinMax	0.754	0.611	0.928	4.68
EMA	0.76	0.612	0.931	4.96
Percentile	0.76	0.618	0.932	4.96
OMSE	0.76	0.622	0.936	4.96
Ours	0.771	0.629	0.941	3.01

From the experimental results, it can be observed that the quantization method proposed in this study outperforms other methods in terms of both detection performance and inference speed. the quantization method introduced in this paper exhibits an improvement in detection performance compared to the comparative methods. Additionally, it significantly outpaces these methods in terms of model inference speed, highlighting the effectiveness of the proposed quantization

method.

Compared to the Full Precision baseline model without quantization, the inference speed increased by 48.4%, but the detection performance, measured by AU PRO, decreased by 20.3%. This performance improvement comes at the cost of a substantial drop in detection performance. This is mainly due to the fact that in the quantization experiments, only the Point Transformer was used as the feature extractor, and the RGB feature extractor was not utilized. Consequently, the model's detection performance was significantly reduced in the absence of RGB features. Furthermore, 3D point cloud features are characterized by sparsity and unordered data, making them more suitable for representing point distribution and spatial relationships. However, they are less adept at capturing image texture features. This explains the substantial drop in detection performance when only 3D point cloud features were used. Notably, while the AU PRO and Image ROCAUC metrics experienced a significant decline, the Pixel ROCAUC metric decreased to a lesser extent, indicating that pixel-level segmentation performance was less affected.

3.3. The experiments and analysis of the combination of pruning and quantization

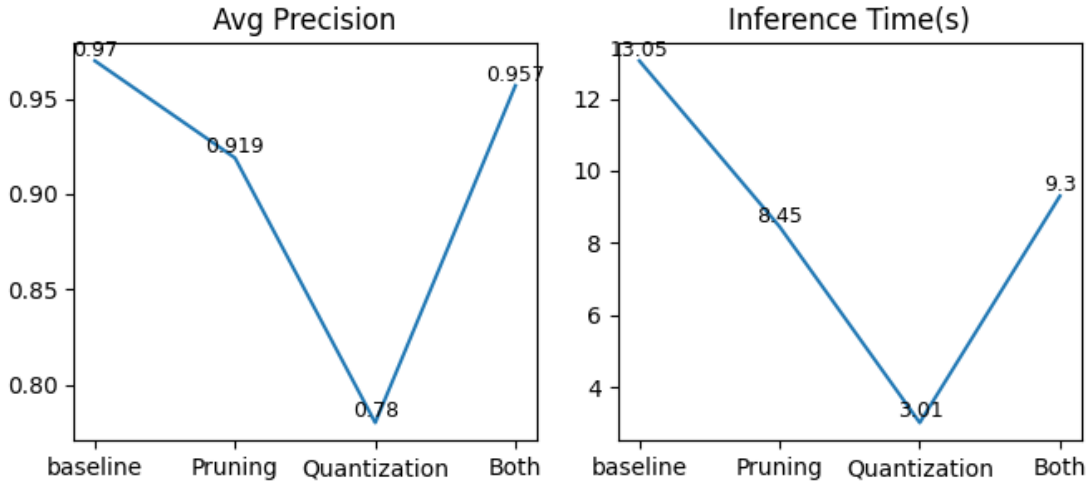

Figure 5. The trend comparison chart between model detection average precision and detection time

From the experimental trend comparison chart in Figure.5, we can observe that the baseline model has the best average detection performance, but at the same time, it also has the longest inference time. Looking at pruning and quantization separately, pruning has a minimal impact on the model's accuracy and a relatively minor reduction in inference time. In contrast, quantization leads to significant performance improvements, reducing detection time to 23% of the baseline model. Clearly, the model's average detection accuracy also drops significantly. This is mainly due to the following two reasons:

The best Flops constraint selected for the pruned model is 0.8, indicating a relatively minor pruning of the model, roughly approximating that only 20% of the model's parameters were pruned. Consequently, there is no significant change in the model's detection accuracy and inference time.

Quantization reduces the computational load of the model by quantizing all parameters to 4 bits, substantially increasing the model's inference speed. However, this performance improvement comes at the cost of parameter precision loss, resulting in a significant drop in average detection accuracy.

When pruning and quantization methods are combined, the features discarded by pruning can be complemented with quantization parameters, while the parameter precision lost due to quantization can be compensated for by pruned features. As a result, the average detection accuracy of the model improves by approximately 4% compared to pruning alone. At this point, the model's average detection accuracy is close to that of the baseline model, while the inference time is significantly lower than that of the baseline model, indicating that this method is a relatively effective model lightweighting approach.

4. Summary

This paper introduces an industrial anomaly detection algorithm based on a Data Transformer. It enhances the model's feature capturing ability by incorporating pre-trained Vision Transformer and Point Transformer models as feature extractors. Feature diversity is improved through multi-modal data fusion. To address issues like feature loss and gradient vanishing in deep networks, residual connections are introduced to fuse input features with generated feature maps. Information loss during the feature fusion process is mitigated by storing original features in multiple memory banks, which are then incorporated into the decision-making process in the final decision layer, enhancing the model's detection capability. The proposed model achieves detection AUC of 95% and segmentation AUC of 95% on the MVTec AD dataset.

The paper also presents methods for model lightweighting based on post-training quantization and pruning. These methods are combined to achieve model compression. While the proposed model surpasses baseline research in terms of detection performance, the use of pre-trained Transformer models and multi-modal data has caused a decrease in detection performance. Therefore, model compression is necessary to improve speed. Two model lightweighting methods are proposed. One is post-training quantization applied to the 3D point cloud feature extractor, converting floating-point calculations into low-bit fixed-point calculations. The paper uses a hierarchical dynamic quantization approach to quantize both feature maps and weights into 4 bits. The other method is model pruning applied to the RGB feature extractor, removing redundant

neuron parameters that are close to zero in the model. Structural sparse methods are employed for model pruning under specific constraints, and an automatic search mechanism is constructed to achieve mask auto-tuning. Experimental results indicate that after lightweighting, the model's detection performance improves by approximately 28.52%, with only a 1.08% decrease in detection performance. This makes it suitable for deployment in industrial applications on small devices like Raspberry Pi and similar platforms.

References

[1] The Overall Analysis Report on the Reasons for Multiple Recalls by Major Automotive Brands in Recent Times by the Product Quality and Safety Research Center of this journal. [J]. China Brands and Anti-Counterfeiting, 2022, 06): 76.

[2] Huang Renbin, Zhan Daohua, Yang Xiuding, et al. Defect Detection Algorithm for Strip Steel Surface Based on Weighted Multiscale Feature Fusion [J]. Computer Integrated Manufacturing Systems, 2023, 1-17.

[3] KIRILLOV A, MINTUN E, RAVI N, et al. Segment Anything [J/OL] 2023, arXiv:2304.02643[https://ui.adsabs.harvard.edu/abs/2023arXiv230402643K.

[4] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017, 39(6): 1137-1149.

[5] YANG H, CHEN Y, SONG K, et al. Multiscale Feature-Clustering-Based Fully Convolutional Autoencoder for Fast Accurate Visual Inspection of Texture Surface Defects [J]. IEEE Transactions on Automation Science and Engineering, 2019, 16(3): 1450-1467.

[6] RUFF L, VANDERMEULEN R A, G RNITZ N, et al. Deep One-Class Classification [C]. In: International Conference on Machine Learning. 2018.

[7] WU P, LIU J, SHEN F. A Deep One-Class Neural Network for Anomalous Event Detection in Complex Scenes [J]. IEEE Transactions on Neural Networks and Learning Systems, 2020, 31(7): 2609-2622.

[8] CARON M, TOUVRON H, MISRA I, et al. Emerging Properties in Self-Supervised Vision Transformers [J/OL] 2021, arXiv:2104.14294[https://ui.adsabs.harvard.edu/abs/2021arXiv210414294C.

[9] PANG Y, WANG W, TAY F E H, et al. Masked Autoencoders for Point Cloud Self-supervised Learning [J/OL] 2022, arXiv:2203.06604[https://ui.adsabs.harvard.edu/abs/2022arXiv220306604P.

[10] Zhao Kailin, Jin Xiaolong, Wang Yuanzhuo. "A Review of Few-Shot Learning Research" [J]. Journal of Software, 2021, 32(02): 349-369.

[11] QI C R, SU H, MO K, et al. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [J]. IEEE, 2017.

[12] QI C R, YI L, SU H, et al. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space [J]. 2017.

[13] BERGMANN P, FAUSER M, SATTLEGGER D, et al. Uninformed Students: Student-Teacher Anomaly Detection with Discriminative Latent Embeddings [J]. IEEE, 2020

[14] DENG J, DONG W, SOCHER R, et al. ImageNet : A Large-Scale Hierarchical Image Database [J]. Proc CVPR, 2009, 2009.

[15] BERGMANN P, JIN X, SATTLEGGER D, et al. The MVTec 3D-AD Dataset for Unsupervised 3D Anomaly Detection and Localization [J]. arXiv e-prints, 2021.

[16] DALAL N. Histograms of oriented gradients for human detection [J]. Proc of Cvpr, 2005

[17] LOWE D G. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Key-points. Int. J. Comput. Vision 60(2), 91-110 [J]. International Journal of Computer Vision, 2004, 60(2):

[18] STEFAN SCHAAL C G A. In: IEEE International Conference on Robotics and Automation, 3, pp.913-918, Georgia, Atlanta. Open Loop Stable Control Strategies for Robot Juggling [J]. 2009.

[19] KWON W, KIM S, MAHONEY M W, et al. A Fast Post-Training Pruning Framework for Transformers [J]. 2022.

[20] HE Y, ZHANG X, SUN J. Channel Pruning for Accelerating Very Deep Neural Networks [J]. 2017.

[21] Lin Shuyuan, Lai Taotao, Yan Yan, et al. "Fitting Multi-Structure Geometric Models Based on Non-Negative Matrix Under-Approximation and Pruning Techniques" [J]. Chinese Journal of Computers, 2021, 44(07): 1414-1429.