Theory and Practice of Science and Technology
2026, VOL. 7, NO. 3, 20-25
DOI: 10.47297/taposatWSP2633-456904.20260703

Research on Quality Control of Gigabit Network Transformers Based on Improved Transformer Neural Networks

Heng Xiao[1], Shengwei Liang[1], Zijie Zhou[1], Yu Wang[1], Hua Sun[1], Dongxu Yang[2]

[1]School of Mechanical Engineering, Xihua University, Chengdu, Sichuan, 610039, China

[2]Deyang Zhida Precise Electronic Co.,Ltd., Deyang, Sichuan, 618500, China

ABSTRACT

The structure of Gigabit network transformers is complex, with significant scale variations in defects such as pin skew and missing magnetic cores. This leads to low efficiency and high false detection rates in traditional manual inspection. To address this challenge and meet the industrial production line's demand for high speed and high precision, this study optimizes the YOLOv5s model with targeted improvements. First, to enhance detection speed, the original SPP module in the network is replaced with the more efficient simSPPF module, accelerating feature fusion and inference. Second, to improve the model's feature extraction capability for complex defects and its understanding of global context, some Conv and C3 modules in the backbone network are replaced with Transformer modules, thereby effectively enhancing the recognition accuracy for small targets and defects with variable morphologies. Experiments on the Gigabit network transformer dataset show that the optimized model significantly outperforms the baseline model in both detection accuracy and inference speed, achieving a balance between precision and efficiency. This provides an effective solution for the intelligent quality inspection of related electronic components.

KEYWORDS

Gigabit network transformer; Defect detection; YOLOv5; Transformer; SimSPPF

1 Introduction

Gigabit network transformers place significantly higher demands on high-speed data transmission capability, electrical isolation, anti-interference performance, miniaturization and integration, temperature adaptability, compatibility, and standardization. However, in the current production process of network transformers, quality inspection still primarily relies on manual visual inspection and electrical performance testing. This approach suffers from low efficiency, strong subjectivity, and limited inspection coverage, making it difficult to meet the manufacturing requirements for high-quality, high-consistency products.

In recent years, scholars both domestically and internationally have conducted extensive research on defect detection using computer vision technology. The research methods can be primarily divided into machine learning-based approaches and deep learning-based approaches. Machine learning-based methods mainly rely on manually designed feature extraction. For instance, Celik et al. proposed a simple detection system combining wavelet transform, Gray-Level Co-occurrence Matrix (GLCM), and a feedforward neural network. However, this method cannot locate the specific position of defects. Qu Erqing et al., addressing the issues of numerous parameters and poor real-time performance in traditional Gabor wavelet filters, proposed a Gabor filter optimization algorithm based on composite differential evolution using a difference algorithm. The optimized model achieved a detection speed of 11 fps, which still falls short of real-time detection requirements. While machine learning-based detection methods are simple and easy to implement, they are heavily influenced by manually extracted features, resulting in poor adaptability and robustness.

Deep learning-based methods primarily include one-stage algorithms like YOLO and SSD, and two-stage algorithms like Faster R-CNN. For example, He et al. proposed a defect detection system based on Faster R-CNN, achieving an mAP of 82.3% for defect detection tasks on the NEU/DET dataset. However, this method has slightly insufficient detection speed, making it difficult to meet the demands of real-time industrial production inspection. Li Weigang et al. utilized a weighted K-means clustering algorithm to propose an improved YOLOv3 model. The improved algorithm achieved an mAP of 80% and maintained a detection speed of 50 fps, but the model has a large number of parameters, making it difficult to deploy. Cheng et al. incorporated a channel attention mechanism into the RetNet network to reduce information loss and used an Adaptive Spatial Feature Fusion (ASFF) module to effectively fuse shallow and deep features extracted by convolutional kernels. They achieved an mAP of 78.25% on the NEU-DET dataset, but the model's accuracy is relatively low. While there is considerable research on algorithms applied to surface defect detection, algorithms that meet real-time detection requirements still face issues of low accuracy and insufficient real-time performance. The aforementioned studies all exhibit certain shortcomings regarding small sample sizes and high-speed detection. Aiming at the problems of insufficient feature extraction capability and low model detection accuracy in current surface defect detection

* **Corresponding Author:** Hua Sun, 0720080008@mail.xhu.edu.cn

algorithms, this paper proposes a new network structure named STYolo based on YOLOv5. Firstly, Transformer modules are integrated with the YOLOv5 model to enhance recognition accuracy. Secondly, a simSPPF module is introduced to replace the SPP module, thereby optimizing the model's training and inference speed.

2 Introduction to the YOLOv5 Algorithm and Improvement Methods

2.1 YOLOv5 Algorithm

YOLOv5 is a deep learning-based object detection algorithm proposed by Alexey Bochkovskiy et al. in 2020. This algorithm employs a single-stage detection approach, dividing the object detection process into three stages: feature extraction, prediction, and post-processing. During the feature extraction stage, it utilizes a lightweight convolutional neural network to rapidly extract feature information from images. Considering the high demands for real-time performance and lightweight design in defect detection, this study comprehensively evaluates the accuracy, efficiency, and size of the recognition model. Consequently, an improved design for the device defect detection network is proposed, based on the YOLOv5s model. The network structure is illustrated in Figure 1.

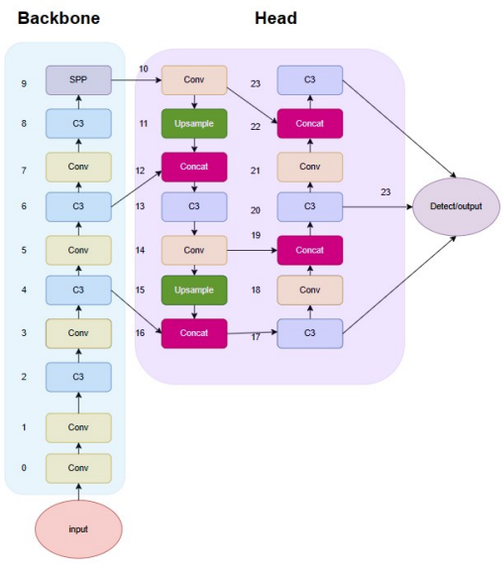


Figure 1 YOLOv5 model structure

2.2 Transformation from SPP Module to simSPPF Module

The SPP (Spatial Pyramid Pooling) module is a classic network structure designed to address the issue of inconsistent object sizes in detection and classification tasks. It was first proposed by Kaiming He et al. in 2014. The core concept is to divide the feature maps generated by convolutional layers into several fixed-size sub-regions and perform max-pooling operations on each. This approach ensures effective detection and classification of objects regardless of their varying positions and scales. The advantages of the SPP module include its adaptability to input images of different sizes and its ability to handle objects of multiple scales. Furthermore, by using fixed sub-regions and max-pooling, it maintains the size of the feature maps, thereby avoiding information loss.

However, the SPP module involves significant computational overhead during pooling due to the need to process sub-regions of varying sizes. Particularly when the number of sub-regions is large, the computational cost becomes substantial, which limits the speed of the SPP module in industrial applications. The structure of SPP is shown in Figure 2.

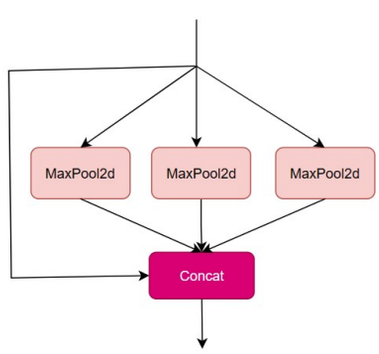


Figure 2 SPP module structure diagram

The SimSPPF module is a data structure commonly used in parsers to merge multiple syntax trees, enabling efficient parsing. It was first proposed in 2002 by Eli J. Fox-Epstein, Daniel P. Friedman, and others, and published at the Conference on Programming Language Design and Implementation. The researchers involved, including Eli J. Fox-Epstein and Daniel P. Friedman, are professors and scholars from Northwestern University in the United States, who have long been engaged in research in programming languages and compiler fields.

The purpose of the SimSPPF module is to address redundancy and efficiency issues in syntax tree merging. Traditional merging methods often generate a large number of duplicate or useless nodes, consuming significant memory and time, which can reduce parsing efficiency. In contrast, the SimSPPF module improves efficiency by streamlining node information and converting multiple tree structures into a more compact form. Specifically, it adds labels to nodes to indicate their type and positional information and employs various optimization techniques to reduce the number of nodes. Additionally, the module can merge different nodes into smaller ones, thereby reducing memory usage and increasing speed. These optimizations enable the SimSPPF module to achieve efficient parsing while also outputting all possible syntax trees, improving both accuracy and coverage.

In summary, the introduction of the SimSPPF module holds significant importance in the fields of parsing and compilers. It not only enhances the efficiency and accuracy of parsing but also helps developers better understand syntax tree structures.In industrial recognition, the SimSPPF module offers the following two advantages:

(1) Faster computation speed: Compared to the SPP module, SimSPPF adopts a new method for computing similarity and feature fusion, significantly reducing computational load and thus improving speed.

(2) Reduced computational load while maintaining accuracy: SimSPPF maintains high accuracy when processing objects of varying sizes. By performing weighted operations on similar subregions, it reduces the computational load of feature weighting. This gives SimSPPF better performance and computational efficiency compared to the SPP module.

Thus, SimSPPF can be seen as an upgrade to SPP, with both being fixed-region pooling methods. Compared to the SPP module, SimSPPF improves speed while maintaining high accuracy. SimSPPF reduces computational load by optimizing similarity calculations and feature fusion, whereas SPP directly performs max pooling on each subregion, resulting in relatively higher computational demands. Therefore, in object detection and classification tasks, different modules can be selected based on specific requirements. Testing shows that a single ConvBNReLU is 18% faster than ConvBNSiLU. The structure of SimSPPF is shown in Figure 3.

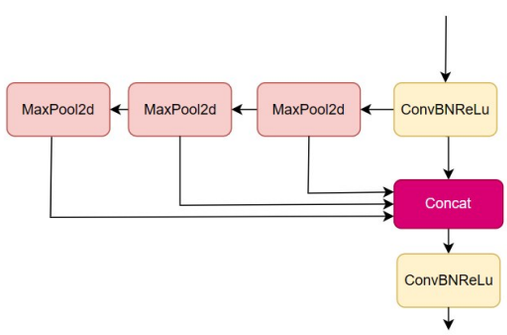


Figure 3 SimSPPF module structure diagram

2.3 Convolutional Layers, C3 Layers, and Transformer Replacement

In YOLOv5, convolutional layers and C3 layers are used for feature extraction. Convolutional layers construct the backbone network to extract image features, while C3 layers achieve downsampling through convolution and channel rearrangement, reducing feature map size while preserving more information. Although these structures perform well and are widely applied in visual tasks, limited data volume and inconspicuous defect features in industrial scenarios constrain recognition accuracy. The Transformer model can better leverage data and improve generalization through pre-training. Originally designed for Natural Language Processing (NLP), the Transformer consists of an encoder and a decoder. ViT first introduced the Transformer encoder into computer vision. This study integrates the Transformer with YOLOv5s by replacing some convolutional blocks and C3 modules with Transformer encoders. The specific process involves splitting the image into patches, adding positional encoding to form a one-dimensional vector, which serves as the encoder input. The attention mechanism can be described as a mapping between a query vector and a series of key-value vector pairs, where the output vector is obtained by weighted summation of the value vectors, with weights determined by the match between the query and keys. This study calculates weights using scaled dot products. When queries, keys, and values belong to the same sequence, it is referred to as self-attention. For efficiency, vectors are typically combined into matrices for computation (as shown in Equation (1) and Figure 4).The multi-head self-attention layer first projects the input matrix (dimension M×d) through different linear projections into h groups of matrices, each

containing a query matrix Q, a key matrix K, and a value matrix V (all of dimension M×d). After computing self-attention separately according to Equation (1), the h output matrices are concatenated and projected into the final output matrix R (dimension M×d), as shown in Equation (2).

$$Attention(Q,K,V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

$$R = Multihead(Q,K,V) = concat(head_1, head_2, head_3.., head_{1=h})W_{hr} \tag{2}$$

The output data of the multi-head self-attention layer is $r_i (i = 1,2,\cdots.,M)$, where $r_i = (r_{i,1}, r_{i,2}, r_{i,3}\cdots.., r_{i,d})$。

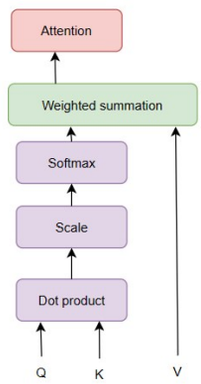


Figure 4 Scaled dot-product attention

In the backbone network of the STYOLO model, the Transformer encoder module is applied multiple times, which facilitates the model's global extraction of deep and shallow features in the image, thereby improving detection accuracy. The Transformer encoder module is shown in Figure 5.

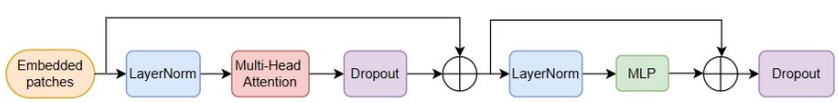


Figure 5 The architecture of transformer encoder

Finally, we designed the overall network model as follows: Figure 6 shows the structure diagram of the improved STYolo model.

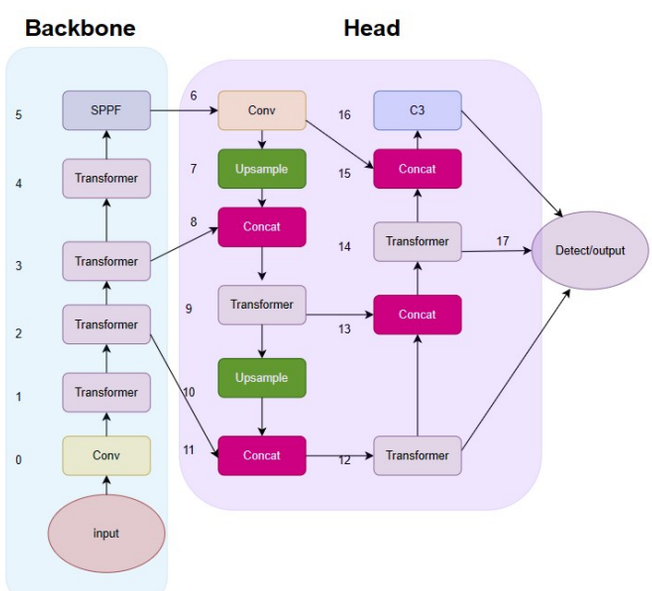


Figure 6 STYolo model structure

3 Experiments and Results Analysis

After constructing the neural network model for workpiece surface defect detection, the model training phase commenced. Based on the defect training dataset, the neural network model proposed in this study and the YOLOv5s object detection model used for comparative experiments were trained separately. Subsequently, the trained models were simultaneously evaluated using the test set. The experiments were conducted under the Ubuntu 16.04 operating system, with Python 3.7 as the programming language and PyTorch 1.8 as the deep learning framework. The hardware configuration consisted of an Intel(R) Core(TM) i9-10980XE CPU and a GeForce NVIDIA RTX 3070 GPU. The schematic diagram of high-speed camera sampling is shown in Figure 7.

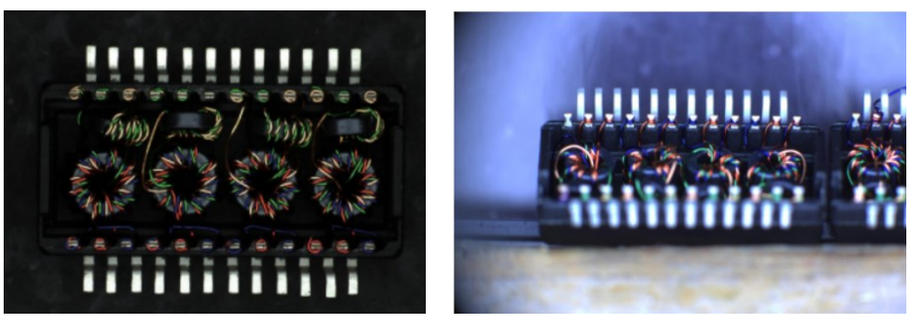


Figure 7 Live sampling chart

In this study, the trained surface defect detection model was first experimentally compared with the trained YOLOv5s network model to verify the effectiveness of the designed surface defect detection model. The statistical analysis results of the evaluation metrics, with the mAP values, are shown in Figure 8.

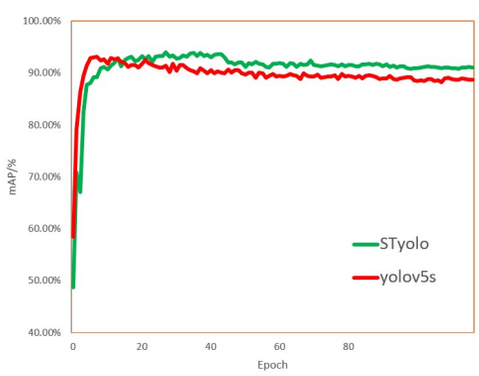
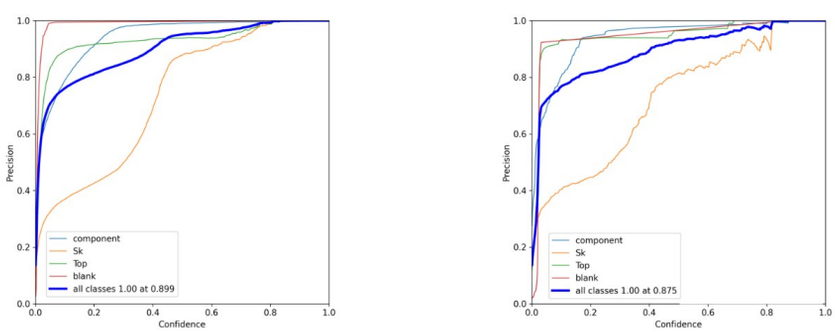


Figure 8 Network model index mAP comparison

From the above figure, it can be observed that the neural network model designed for annular workpiece defect detection in this study shows certain improvements over the YOLOv5s neural network model, with an approximate 5.3% increase in mAP value. In this defect detection neural network, detection is primarily performed for four types of image features in the collected images, namely the workpiece itself (component), pin skew (SK), sharp point (Top), and missing iron core (blank). The precision rates for these four image features are shown in Figure 9 below.



(a) STYolo (b) Yolov5s

Figure 9 Precision rate results display

Figure 9 illustrates the variation in precision with confidence for both the designed network model and the YOLOv5 network model during training. Compared to YOLOv5s, STYolo exhibits a smoother upward trend in precision as confidence increases, with its values slightly higher than those of YOLOv5s. As shown in the figure, among the four detection features in this study, the image feature of pin skew is particularly critical. In the precision comparison for defect detection, when the confidence threshold exceeds 0.5, STYolo achieves a precision 2 to 3 percentage points higher than that of YOLOv5s. From an overall model perspective, when the confidence threshold is above 0.5, the precision of the STYolo model remains consistently above 0.88, showing an improvement compared to YOLOv5s's precision of 0.875.

4 Conclusion

In defect detection algorithms, YOLO has garnered significant attention due to its high accuracy and fast processing speed. However, when applied in industrial scenarios, the algorithm still faces challenges such as low defect recognition rates and insufficient computational efficiency. To address these issues, this paper proposes two improvement measures to optimize the network structure: firstly, replacing the SPP module with the SimSPPF module to reduce computational complexity and enhance processing speed; secondly, introducing Transformer layers to replace the original convolutional layers and C3 layers, leveraging their high recognition capability to improve overall detection performance. Experimental results demonstrate that the proposed improvements significantly enhance both the detection accuracy and operational efficiency of the YOLOv5 algorithm, making it more suitable for practical industrial applications. In future experiments, further optimization could be explored by investigating the impact of Transformer layers with different depths on network recognition accuracy and processing speed.

Funding

This work was supported by the Sichuan Science and Technology Program (NO: 2025ZYD0034 and NO: 2024ZYD0003)

About the Author

Heng Xiao, male, Han ethnicity, from Chengdu, Sic;huan Province, holds a doctorate, is a lecturer at the School of Mechanical Engineering, Xihua University, and his research interests lie in bionic science and engineering.

Hua Sun, male, Han ethnicity, from Qujing, Yunnan Province, holds a doctorate, is an associate professor at the School of Mechanical Engineering, Xihua University, and his research interests are in artificial intelligence and big data.

References

[1] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, TsungYi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In CVPR, 2021.

[2] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In CVPR, 2022.

[3] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. arXiv preprint arXiv:2209.15639, 2022.

[4] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and ShihFu Chang. Open-vocabulary object detection using captions. In CVPR, 2021.

[5] Ce Zhang and Azim Eskandarian. A quality index metric and method for online self-assessment of autonomous vehicles sensory perception arXiv preprint arXiv:2203.02588, 2022.

[6] Zhiheng Wu, Yue Lu, Xingyu Chen, Zhengxing Wu, Liwen Kang, and Junzhi Yu. Uc-owod: Unknownclassified open world object detection. In Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part X, pages 193–210. Springer, 2022.

[7] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. IEEE transactions on neural networks and learning systems, 30(11):3212–3232, 2019.

[8] Ezra Fielding, Clement N Nyirenda, and Mattia Vaccari. 2022. The Classification of Optical Galaxy Morphology Using Unsupervised Learning Techniques. In Prceedings of the International Conference on Electrical, Computer and Energy Technologies (ICECET).

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (2017), 84–90.

[10] R Shirley, K Duncan, MC Campos Varillas, PD Hurley, K Małek, Y Roehlly, MWL Smith, H Aussel, T Bakx, V Buat, et al. 2021. HELP: the herschel extragalactic legacy project. Monthly Notices of the Royal Astronomical Society 507, 1 (2021), 129–155.

[11] K. Cha, J. Seo, and T. Lee. A billion-scale foundation model for remote sensing images. arXiv preprint arXiv:2304.05215, 2023.

[12] P. Akiva, M. Purri, and M. Leotta. Self-supervised material and texture representation learning for remote sensing tasks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8203–8215, 2022.

[13] J. Ding, N. Xue, G.-S. Xia, and D. Dai. Decoupling zero-shot semantic segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 11583–11592, 2022.

[14] K. Heidler, L. Mou, D. Hu, P. Jin, G. Li, C. Gan, J.-R. Wen, and X. X. Zhu. Self-supervised audiovisual representation learning for remote sensing data. International Journal of Applied Earth Observation and Geoinformation, 116:103130, 2023.

[15] X. Lu, X. Sun, W. Diao, Y. Mao, J. Li, Y. Zhang, P. Wang, and K. Fu. Few-shot object detection in aerial imagery guided by textmodal knowledge. IEEE Transactions on Geoscience and Remote Sensing, 2023.