Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.085757

ARTICLE

Research on Hybrid Unsupervised–Supervised Learning Fusion Method for Defect Detection of Stamped Parts

Zhikai Chi[1], Jiaxu Ning[1,*], Delong Zhang[1] and Changsheng Zhang[2,3]

[1]School of Information Science and Engineering, Shenyang Ligong University, Shenyang, China
[2]School of Computer Science and Engineering, Ningxia Institute of Science and Technology, Shizuishan, China
[3]School of Software, Northeastern University, Shenyang, China

*Corresponding Author: Jiaxu Ning. Email: ningjiaxu@sylu.edu.cn

Received: 17 May 2026; Accepted: 24 August 2026; Published: 15 September 2026

ABSTRACT: Aiming at the problem that factory stamping parts have various types of defects, random locations, different sizes, and both known and unknown defects, it is difficult for traditional single inspection methods to achieve both accurate classification and generalized identification capabilities. To this end, the Hybrid Unsupervised Learning-Supervised Learning Fusion Defect Detection (HUSLFDD) model is proposed. The model adopts a dual-branch shared backbone network architecture, in which the supervised learning branch focuses on the accurate classification of known defects, and the unsupervised learning branch realizes feature capture and identification of unknown defects. The weighted fusion of the dual-branch results is completed through the image scorer, and fixed validation-calibrated parameters is achieved, ultimately taking into account detection accuracy and generalization capabilities. A large number of experimental results show that the proposed model has an F1 score of 91.1%–94.2%, a recall rate of 94.4%–100%, and an image-level AUROC of 80.6%–95.6% on four comprehensive test sets, which is significantly better than detection methods of single supervised learning, single unsupervised learning, and a simple combination of the two. At the same time, the model detection speed reaches 9.14 FPS, meeting the needs of industrial real-time detection. This research provides efficient and reliable technical support for quality control of stamping parts in the automotive manufacturing industry.

KEYWORDS: Stamped parts; defect detection; supervised learning; unsupervised learning

1 Introduction

1.1 Research Motivation

As core components in the automotive manufacturing sector, stamped parts are widely used in critical areas such as the vehicle body frame, chassis systems, and engine compartments. Their surface quality and structural integrity directly determine vehicle driving safety, structural durability, and overall assembly performance [1]. In industrial production processes, stamped parts are susceptible to various factors that can cause common surface defects such as cracks, deformation, holes, burrs, and indentations [2]. Additionally, due to dynamic changes in production conditions, new and unknown types of defects may emerge, posing significant challenges for quality inspection [3].

In traditional industrial production, surface defect detection initially relied on manual visual inspection. Workers used their naked eyes to determine whether parts had any abnormalities; however, because some defects are extremely subtle, and due to visual fatigue from prolonged work and subjective differences among individuals [4], the reliability and consistency of this method are difficult to guarantee. Furthermore, training a professional defect inspector requires a significant investment of time and financial resources [5]. Consequently, as modern industrial production lines demand ever-higher standards of product quality and efficiency, relying solely on manual visual inspection has gradually become insufficient to meet practical needs [6]. In recent years, to address these challenges and improve inspection accuracy and efficiency, automated real-time defect detection methods based on deep learning have been extensively researched and applied [7].

1.2 Literature Review

Deep learning offers significant advantages in industrial image defect detection. Supervised learning methods, centered on convolutional neural networks, rely on large-scale labeled defect samples to perform feature learning and defect classification. Ansari et al. [8] constructed a CNN classification model, combining CAD and XCT-assisted annotation, Hyperband hyperparameter optimization, and data augmentation strategies to achieve efficient detection of *in-situ* porosity defects in the laser powder bed fusion process, thereby alleviating issues such as complex annotation, dataset imbalance, and insufficient monitoring automation. Latham and Giannetti [9] integrated expert knowledge with machine learning for width-related defect detection and root cause analysis in hot strip mills, thereby reducing reliance on manual labor and diagnostic delays. Ni et al. [10] proposed the FMR-YOLO algorithm, an improved version of YOLOv8, to address issues such as inadequate feature extraction of steel surface defects and limitations in detection speed and accuracy. Liu et al. [11] introduced global attention and a cascaded fusion network to address the challenges of unstructured, multi-scale, and sparse data associated with steel surface defects. While these methods achieve high detection accuracy under conditions of known defects and sufficient samples, in real-world industrial scenarios—where defect samples are limited, features are weak, and defect shapes, sizes, and locations are highly variable—they struggle to cover all defect types, with particularly limited capabilities for identifying rare defects [12].

Unsupervised learning can autonomously uncover data distributions and correlations in the absence of labeled data to perform defect classification, with reconstruction and feature embedding methods being the most widely applied. Luo et al. [13] proposed AMI-Net, which targets multi-scale semantic features for reconstruction. By combining a masking strategy with an adaptive mask generator to suppress reconstruction in defect regions, it addresses issues in traditional reconstruction methods such as incorrect reconstruction of defects, unstable mask detection, and low inference efficiency. Deng et al. [14] designed a "noise-to-norm" reconstruction paradigm based on M-Net and residual attention modules. By introducing Gaussian noise, they prevent excessive reconstruction of anomalous regions and enhance the model's robustness to variations in target locations. Yang and Guo [15] constructed an unsupervised detection model based on a visual Transformer, combining a memory module with a coordinate attention mechanism to enhance global feature extraction capabilities, thereby mitigating issues such as overgeneralization of autoencoders, failure of defect reconstruction, and high sample annotation costs. Peng et al. [16] proposed an improved GANomaly-based method for industrial surface defect detection and localization, which improves the stability of irregular texture defect detection through skip connections, self-attention, and an improved loss function. The PatchCore method proposed by Roth et al. [17] employs a pre-trained ResNet for feature extraction, significantly improving detection inference speed. Liang et al. [18] designed a two-stage progressive learning framework that achieves feature transfer from synthetic anomalies to real anomalies through feature space alignment, thereby enhancing the model's ability to detect unknown anomaly types. Wang et al. [19] proposed a large-scale visual model anomaly detection method based on style enhancement. This method simulates domain differences under different manufacturing conditions through style perturbation and extracts general features by combining a pre-trained visual model, thereby improving the cross-domain generalization ability in powder bed additive manufacturing scenarios. At the same time, this method integrates multi-domain knowledge representation to enhance feature alignment and information complementarity, alleviating the problems of traditional methods' strong dependence on annotation and insufficient cross-domain adaptability. Wang [20] designed the LSCA contrastive learning model for semantic segmentation. Incorporating a defect memory bank alongside class-aware semantic contrast and attention fusion modules, the model mines global defect semantic features from limited annotated data, thereby addressing issues—such as sample imbalance and inadequate defect feature extraction—faced by traditional supervised segmentation models in the detection of layer-wise defects in additive manufacturing.

1.3 Contributions

Most existing industrial defect detection studies are focused on single scenarios or specific defect types, which are difficult to meet the engineering requirements of simultaneous real-time identification of known and unknown defects in automobile stamping production. Among them, supervised learning methods have a significantly reduced ability to detect new defects outside the training set, while unsupervised learning methods have the problem of low accuracy in detecting known defects. Furthermore, generative models such as GAN also have the drawbacks of poor training stability and high design difficulty. In addition, considering the complex characteristics of automobile stamping cracks, such as random location, different size, and diverse shape, this paper proposes a surface defect detection model for automobile stamping. (HUSLFDD) is used to improve the detection of known and unknown crack defects in the production process. It is important to clarify that this model is not a simple combination of existing modules such as YOLOv8, C2f, SPPF, and PAN-FPN, but rather integrates these classic components as basic building blocks into a newly designed unified framework. Its core innovation lies in the system-level integration of supervised defect detection and unsupervised anomaly learning based on a shared backbone network. At the same time, it constructs an adaptive fusion mechanism that can adapt to industrial scenarios and perform joint reasoning for known and unknown defects. The main contributions of this paper are as follows:

1. A unified dual-branch defect detection framework (HUSLFDD) is proposed to jointly model known and unknown defects in stamped parts. Unlike conventional approaches that separately apply supervised detection (e.g., YOLO-based models) or unsupervised anomaly detection, the proposed method integrates both within a shared backbone architecture, enabling consistent feature representation and complementary learning for industrial inspection scenarios where known and unknown defects coexist.
2. A lightweight, multi-module fusion backbone network is designed to provide efficient feature extraction capabilities for HUSLFDD, addressing the issues of information redundancy and scale feature loss during the extraction of stamping part defect features. This network consists of CBS, C2f, and SPPF modules, achieving lightweight design while efficiently capturing multi-scale features of stamping part images, balancing detection speed and comprehensive feature extraction.
3. A supervised branch feature detection network based on PAN-FPN is designed to achieve high-precision localization and classification of cracks in known stamped parts, improving the uneven accuracy problem in multi-scale defect detection. This network integrates shallow, medium, and deep multi-scale features from the backbone network through bidirectional feature fusion from top to bottom and bottom to top. It also employs separate detection heads to independently complete bounding box regression and class confidence prediction, improving the detection efficiency and accuracy of cracks at different scales and locations.

4. A validation-calibrated weighted fusion strategy is designed to unify supervised detection confidence and unsupervised anomaly scores into a single decision metric. This mechanism enables dynamic fusion of heterogeneous outputs from both branches, improving robustness under mixed known–unknown defect distributions and enhancing decision reliability in industrial environments.

2 Method

2.1 Hybrid Unsupervised Learning-Supervised Learning Fusion Defect Detection Model (HUSLFDD)

This section presents HUSLFDD, a defect detection model that combines unsupervised and supervised learning. This model aims to score anomalies in images of manufactured stamped parts to determine the presence of defects. The model consists of three main parts: a supervised learning branch, an unsupervised learning branch, and a scorer. The framework of the HUSLFDD model is shown in Fig. 1.

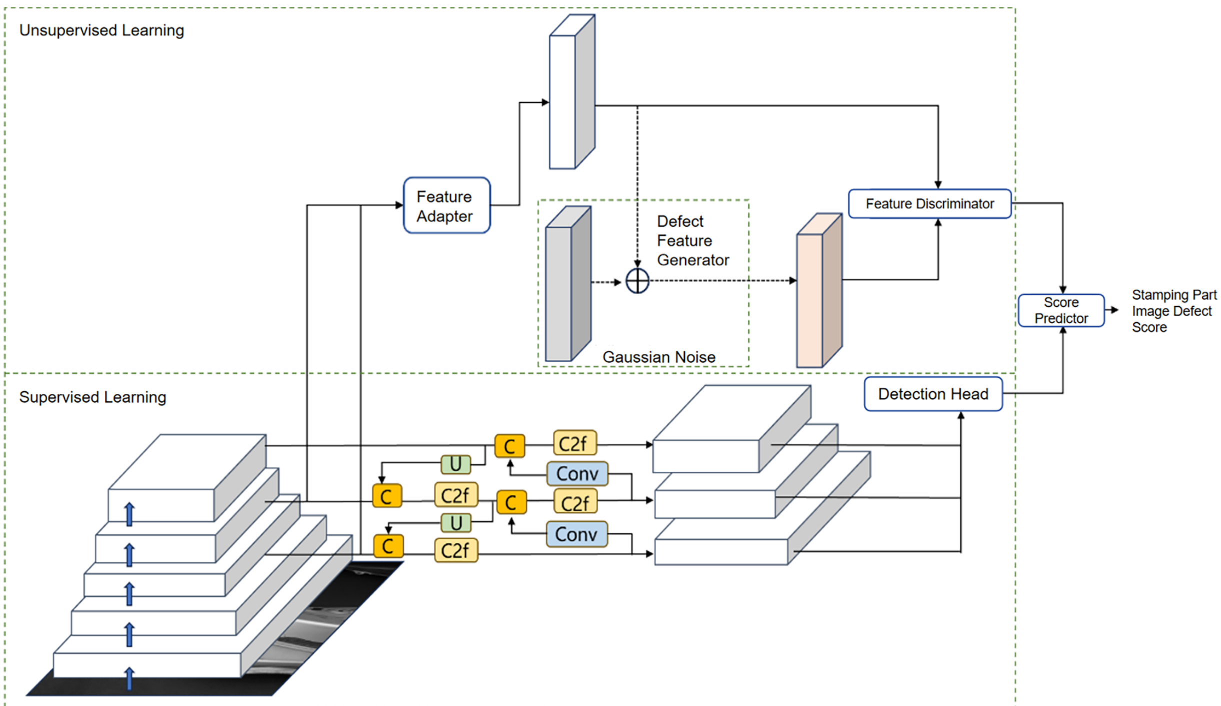


Figure 1: A defect detection model framework combining unsupervised and supervised learning.

The HUSLFDD framework adopts a dual-branch architecture with a shared backbone. For an input stamping-part image x, the lightweight backbone $B(\cdot)$, composed of CBS, C2f, and SPPF modules, is executed only once to generate multi-scale feature maps:

$$\left\{F_{low}, F_{mid}, F_{high}\right\} = B(x) \tag{1}$$

These shared feature maps are then delivered to two task-oriented branches. The supervised branch $G_s(\cdot)$ is built on the YOLOv8 detection paradigm and receives $\left\{F_{low}, F_{mid}, F_{high}\right\}$ as input. Through PAN-FPN bidirectional aggregation and decoupled detection heads, it outputs bounding boxes, defect categories, and confidence scores for known defect types:

$$D_s = G_s\left(F_{low}, F_{mid}, F_{high}\right) = \{(b_i, c_i, p_i)\}_{i=1}^{N}, \tag{2}$$

where b_i, c_i and p_i denote the predicted bounding box, defect category, and confidence score of the i-th candidate defect region, respectively. The unsupervised branch $G_u(\cdot)$ uses the same backbone features instead of extracting another independent feature representation. It contains a feature adapter, a defect feature generator, and a feature discriminator. The feature adapter aligns the shared backbone feature distribution with the feature space required by anomaly discrimination:

$$Z = A(F), \tag{3}$$

where F denotes the selected backbone feature representation and $A(\cdot)$ is the adapter. During training, Gaussian noise is added to normal features to generate synthetic abnormal features, and the discriminator is optimized to separate normal features from generated abnormal features. The generated abnormal features are obtained by perturbing normal feature representations with Gaussian noise, thereby simulating potential unknown defect distributions without requiring real anomaly annotations. During inference, the defect feature generator is disabled, and the discriminator directly assigns an anomaly score to each adapted feature location.

$$S_u = Q(Z) = \{q_j\}_{j=1}^{M} \tag{4}$$

The interaction between the two branches is therefore realized at two levels. First, they share the same backbone feature extractor, which reduces repeated computation and keeps the feature basis consistent for both known-defect detection and unknown-defect anomaly scoring. Second, their outputs are fused by the image scorer, where the supervised branch contributes precise localization and confidence information for known defects, while the unsupervised branch contributes sensitivity to unseen or weakly represented defect patterns. This design avoids the information inconsistency caused by independently trained feature extractors and improves both inference efficiency and generalization.

2.2 Lightweight Multi-Module Integrated Backbone Network

The HUSLFDD model backbone network consists of three modules: Conv-BatchNorm-SiLU (CBS), C2f, and Spatial Pyramid Pooling Fast (SPPF). The CBS module mainly consists of convolutional layers, batch normalization layers, and activation functions, as shown in Fig. 2.

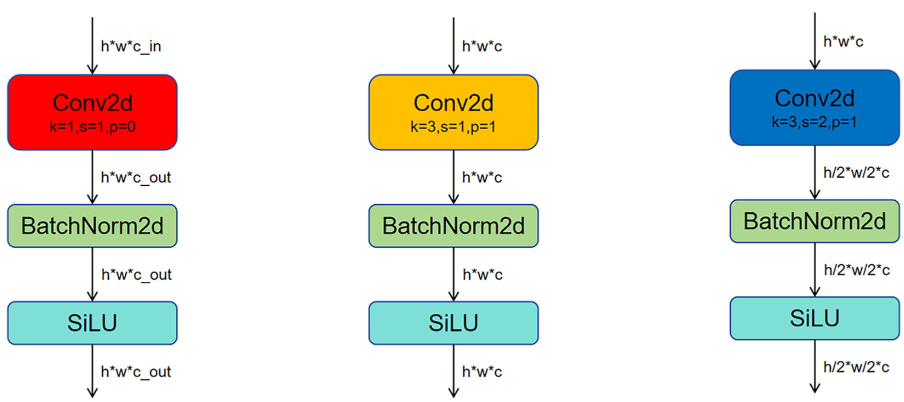


Figure 2: CSB module structure.

Convolutional layers perform convolution operations in the width and height dimensions to extract features. Batch normalization layers adjust the distribution of convolutional outputs through normalization and scaling, stabilizing the data scale, accelerating model convergence, and improving generalization ability.

Activation functions (SiLu) introduce non-linear factors, enabling the model to learn more complex mapping relationships; their definitions are as follows:

$$SiLu(x) = x \cdot \sigma(x), \tag{5}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{6}$$

The C2f module first generates intermediate feature maps via the CBS module, then splits them into two paths. One path is directly fed into the neck stitching block, while the other path enters the bottleneck block, which consists of two layers of convolution and skip connections, to reduce computation and enhance nonlinear expressive power. The bottleneck block output features are stitched together and then output by the final CBS module. Its forward propagation process is shown in Fig. 3. The SPPF module consists of an initial CBS, three levels of max pooling, a stitching layer, and a terminal CBS. It can efficiently extract multi-scale features, and its structure is shown in Fig. 4. Max pooling is used to compress features and reduce computational complexity. The stitching layer fuses multi-scale features in the channel dimension and then inputs them into the final CBS module. This backbone network can effectively obtain rich multi-scale information of stamping parts images under the premise of lightweight design.

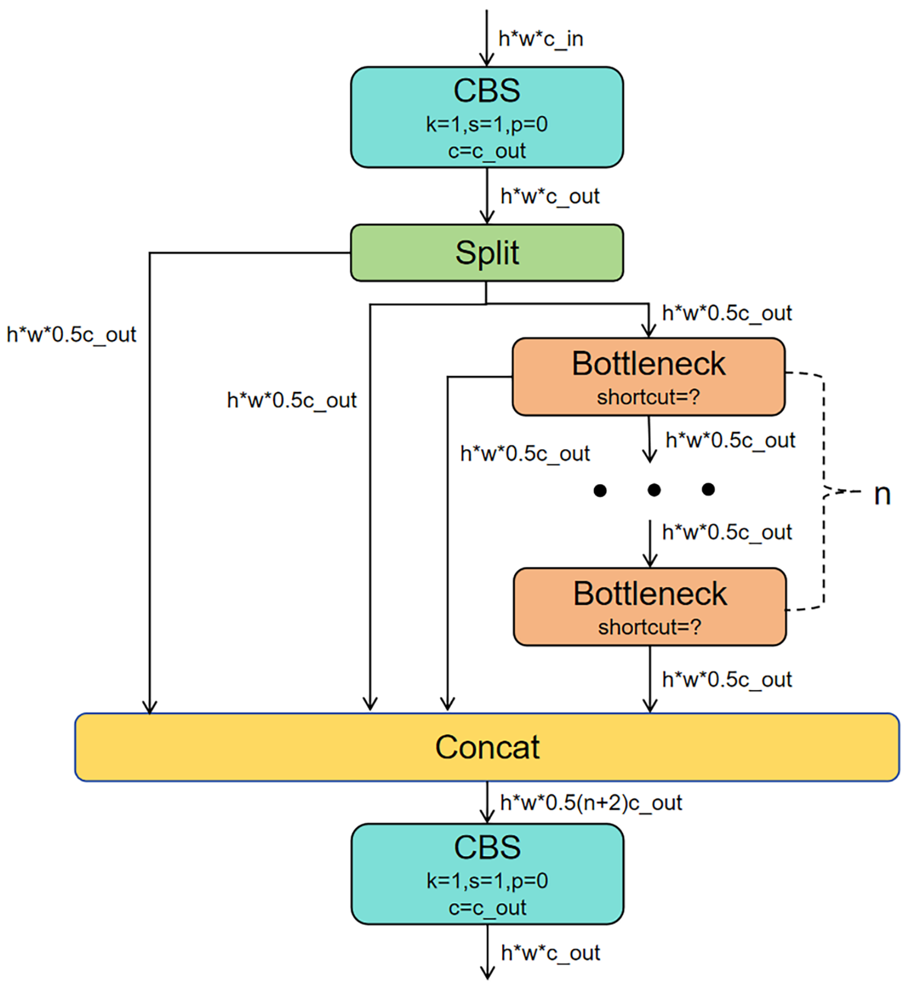


Figure 3: C2f module forward propagation process.

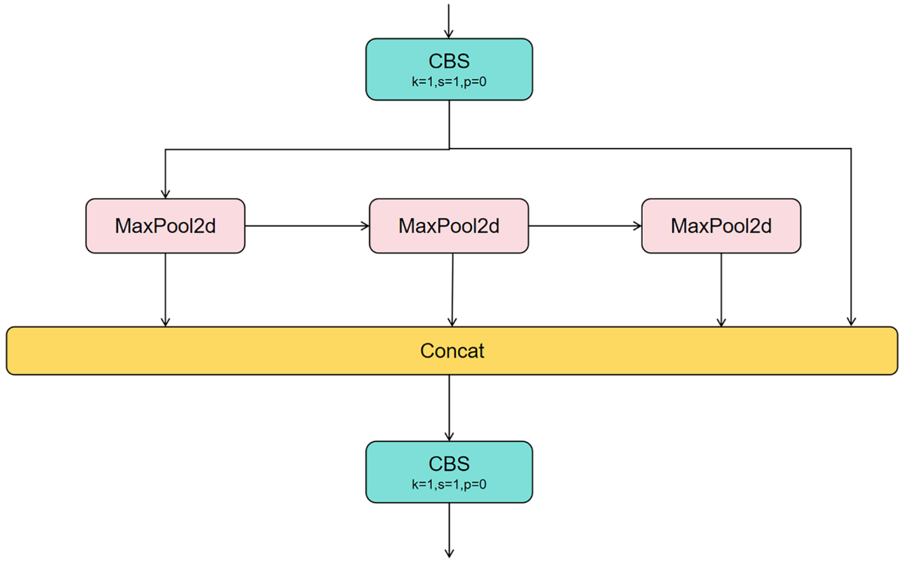


Figure 4: SPPF module structure.

2.3 Supervised Learning Branch Feature Detection Network

The supervised learning branch of the feature detection network includes a feature enhancement network and a detection head, and the computation process is shown in Algorithm 1. The feature enhancement network is based on the PAN-FPN architecture, which can efficiently fuse features of stamped parts at multiple scales and adapt to the detection of defects of different sizes. Among them, FPN fuses high and low-level features from top to bottom, while PAN strengthens the transmission of shallow information to higher levels from bottom to top; the combination of the two realizes bidirectional feature flow, which improves the fusion efficiency while reducing the amount of computation and meets the requirements of real-time detection.

Algorithm 1: Supervised learning branch feature detection network computation flow

Input: Shallow feature map F_low, mid-level feature map F_mid, and deep feature map F_high output from the backbone network.
Output: A list of detection results is "detections".
1: T1 ←Upsampling F_high matches the size of F_mid
2: M1 ← Concatenate feature map F_mid with T1 and then perform convolution
3: T2 ← Upsampling M1 matches the F_low size.
4: Feat_small ← Concatenate feature maps F_low and T2, then perform convolution
5: D1 ← Downsampling S1 matches size M1
6: Feat_mid ← Concatenate feature maps M1 and D1 and then convolve them
7: D2 ← Downsampling M2 matches the F_high size
8: Feat_large ← Concatenate feature maps F_high and D2, then perform convolution
9: detections ← Initialize an empty list
10: **for** F in [Feat_small, Feat_mid, Feat_large] do
11: x_bb ← F sequentially passes through the two CBS modules of the bounding box prediction branch
12: bbox_pred ← x_bb is input into a 1 × 1 convolutional layer used for bounding box prediction

(Continued)

Algorithm 1 (continued)

x_cls ← F sequentially passes through the two CBS modules of the category confidence prediction branch
cls_pred ←Input x_cls into a 1 × 1 convolutional layer for class prediction
for i in range(number of predicted bounding boxes) do
det.bbox ← bbox_pred[i]
det.confidences ← cls_scores[i]
Add det to the detections list
end for
end for

Specifically, layers four, six, and eight of the backbone network correspond to shallow, medium, and deep feature maps, respectively, and serve as inputs to the feature enhancement network. The Feature Pyramid Network (FPN) upsamples the deep features, adjusting their size to the resolution of the medium feature map, and fuses them. The fused result is then upsampled again and fused with the shallow feature map. The Path Aggregation Network (PAN) downsamples the shallow feature map and fuses it with the medium feature map. The fused result is then downsampled again and fused with the deep feature map. Finally, through bidirectional aggregation by the FPN and PAN, three feature maps of stamped parts at different scales are obtained, corresponding to small, medium, and large defect regions, respectively. These feature maps are then input into the detection head.

The detection head is responsible for detecting defect regions and consists of two tracks. The first track predicts the bounding box of the defect region, and the second track predicts the category. Both tracks are composed of two CBS modules followed by a convolutional layer, as shown in Fig. 5. By separating separation and detection, the detection head module makes network training and inference more efficient.

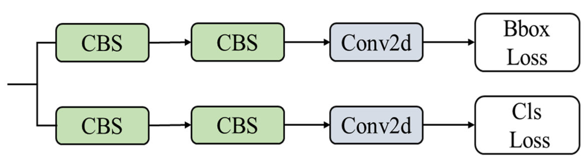


Figure 5: Detection head structure.

2.4 Image Scorer

During the model inference stage, the stamping part image scorer integrates the output of the unsupervised branch feature discriminator with the output of the supervised branch detection head to obtain the stamping part defect score; the higher the score, the greater the probability that the image contains defects.

Specifically, for each stamping part test image $x_i \in X_{test}$, the output of the feature discriminator in the unsupervised learning branch is $D(q^i_{h,w})$, where $q^i_{h,w}$ represents the feature at image position (h, w). Then, the defect score for each feature is:

$$s^i_{h,w} = -D(q^i_{h,w}) \tag{7}$$

Each defect region of different sizes has a corresponding sensitive response point. This study selects the maximum value among all feature anomaly scores as the image defect score, which can effectively quantify the probability of the existence of defects at different scales.

$$S_s\left(x_i\right) = \max_{(h,w)\in W_0\times H_0} s^i_{h,w} \tag{8}$$

The supervisory branch inspection head outputs test images of stamped parts, showing the defect locations and corresponding confidence levels. If one or more defects are detected, the highest confidence level is taken as the anomaly score; if no defects are detected, the anomaly score is 0. This is formally expressed as:

$$S_y\left(x_i\right) = \begin{cases} \max\left(confidence\_score\right), object > 0 \\ 0, object = 0 \end{cases} \tag{9}$$

After scoring the unsupervised and supervised learning branches, HUSLFDD calculates a final composite score by weighting the scores from both branches. Images with a final score greater than a threshold are classified as defective, while those with a score less than or equal to the threshold are classified as normal.

$$S\left(x_i\right) = w \cdot S_s\left(x_i\right) + (1 - w) \cdot S_y(x_i) \tag{10}$$

$$\begin{cases} S\left(x_i\right) > \tau \Rightarrow Defect\ images \\ S\left(x_i\right) \le \tau \Rightarrow Normal\ image \end{cases} \tag{11}$$

here, w and τ represent the fusion weight and the decision threshold, respectively. Both parameters are determined using the validation set during the model development phase and remain fixed across all test datasets during the inference phase; the overall process for calculating the defect score is illustrated in Algorithm 2.

Algorithm 2: Image scorer calculation process

Input: unsupervised branch test image feature scores (anomaly_scores), supervised branch detection results (detections), unsupervised branch weights (w1), supervised branch weights (w2), defect score threshold
Output: Defect score of the test image, whether it is a defective image

1: max_unsupervised ←Find the maximum value in anomaly_scores
2: **if** detections is not empty:
3: max_supervised ←Iterate through the detections and take the highest confidence score.
4: **else:**
5: max_supervised ← 0
6: **end If**
7: final_score ← w1 × max_unsupervised + w2 × max_supervised
8: **if** final_score > Defect score threshold:
9: The output `final_score` is the defect image.
10: **else:**
11: The output `final_score` is not a defect image.
12: **end If**

2.5 Model Training Process

The HUSLFDD model combines supervised and unsupervised learning to achieve more efficient and accurate defect detection. Therefore, two branches need to be trained separately during the training phase. The HUSLFDD model first trains the supervised learning branch. The stamping parts dataset used in the training phase consists of two parts: defect images and label files for the defect images. The loss function mainly consists of classification loss, localization loss, and object identification loss, defined as follows:

$$Total\ Loss = \lambda_{cls} \cdot L_{cls} + \lambda_{loc} \cdot L_{loc} + \lambda_{obj} \cdot L_{obj} \tag{12}$$

here, λ_{cls}, λ_{loc}, and λ_{obj} are hyperparameters used to balance the weights of the various loss terms.

After training the supervised learning branch, the unsupervised learning branch is trained. The training dataset for the unsupervised learning part consists of images of defect-free stamped parts, which differ somewhat from the training images used in the supervised learning branch. To utilize the pre-trained feature extraction network in the supervised learning branch, HUSLFDD uses the backbone network of the supervised learning branch as a pre-trained model for feature extraction from the images of defect-free stamped parts. To enhance the generalization ability of the HUSLFDD model, the stamped part dataset used for training in the supervised learning branch differs somewhat from that used in the unsupervised learning branch. Therefore, the extracted features are fed into a feature adapter to reduce domain differences. Then, the features output by the feature adapter, along with the synthesized features with added noise, are fed into a feature discriminator. This allows the discriminator to distinguish between normal features and synthesized abnormal features in the stamped part images. The training objective is as follows:

$$L = min \sum_{x^i \in X_s} \sum_{h,w} \frac{\max\left(0, th^+ - D\left(q^i_{h,w}\right)\right) + \max\left(0, -th^- + D\left(q^{i-}_{h,w}\right)\right)}{H_0 * W_0} \tag{13}$$

where $D\left(q^i_{h,w}\right)$ is the score of the feature discriminator for the feature at position (h, w) of sample x^i in the stamping part training set, and $D\left(q^{i-}_{h,w}\right)$ is the score of the feature discriminator for the abnormal feature synthesized by adding noise to the feature at position (h, w) of sample x^i in the stamping part training set. th^+ and th^- are defaulted to 0.5 and −0.5 to prevent overfitting, and $max(\cdot)$ is the maximum value.

3 Experimental Setup and Dataset

3.1 Experimental Setup

The hardware development environment for the HUSLFDD model is a 64-bit Windows 11 Home operating system, an AMD Ryzen 7 8745H processor with Radeon 780M Graphics (3.80 GHz), 16 GB of memory, and an NVIDIA GeForce RTX 4050 Laptop GPU. The software development environment used in this paper is PyCharm Community Edition, primarily using Python. Python and its rich open-source libraries provide powerful development tools and support, making model building, training, and evaluation more convenient and efficient.

To ensure a fair comparison, all baseline methods and the proposed HUSLFDD model were trained and evaluated using the same training/testing dataset splits, image preprocessing pipeline, input image resolution, hardware and software environment, and evaluation metrics. For the compared methods, hyperparameter settings followed the official implementations or the configurations recommended in the corresponding original publications. No additional test-set-specific tuning was performed.

3.2 Dataset Introduction

This paper uses two sets of stamped part image datasets: the Stamped Part Crack Anomaly Detection Dataset (SPCAD) [21] and the Stamped Part Crack Annotation Dataset (PCAD). The SPCAD dataset was jointly collected by Northeastern University and BMW Brilliance Automotive Ltd. from real automotive stamping production lines. The original SPCAD dataset contains 1586 stamped part images with a uniform resolution of 2048 × 2048, including three categories of samples: normal workpieces, minor cracks, and severe cracks. We split the complete original dataset into two independent subsets to adapt the dual-branch model: the SPCAD experimental subset: 892 images were selected for unsupervised branch training and cross-product generalization testing, further divided into four product categories as shown in Table 1. All product subsets were strictly divided into training and testing sets according to an 8:2 ratio; the PCAD annotation dataset: the remaining 694 images from the original SPCAD were manually annotated using the YOLO standard format, specifically for supervised branch training. PCAD was divided into 556 training images and 138 test images according to the 8:2 rule. The same training/testing split was consistently applied to all baseline methods and the proposed HUSLFDD model throughout the experiments, ensuring a fair and reproducible comparison. Each PCAD image is accompanied by a corresponding .txt annotation file, recording the defect category index, normalized center coordinates of the bounding box, and normalized width and height. During the annotation phase, all visible crack areas are first manually selected, then the boundary fit and label consistency are checked. Ambiguous samples are re-annotated to reduce annotation noise. This paper defines two types of industrial defects: known defects are crack samples contained in the PCAD training set and accurately learned by the supervised branch; unknown defects come from four product subsets in SPCAD, whose product appearance, texture, and crack morphology differ from the PCAD annotation training set, reflecting the real-world situation of coexisting new and old defects on the production line. During model training, the supervised branch uses all annotated PCAD training images; the unsupervised branch uses only normal samples from the SPCAD training and validation set, without inputting any defect labels. During the testing phase, SPCAD and PCAD test samples are merged to construct four comprehensive test sets, each containing normal samples, known defects, and unknown defects.

Table 1: Information on each subset of the stamping part defect detection dataset.

Dataset	Training and Validation	Test-Normal	Test-Crack	Resolution
Product 1	216	30	24	2048 × 2048
Product 2	216	31	23	2048 × 2048
Product 3	144	25	11	2048 × 2048
Product 4	138	24	10	2048 × 2048

To simulate the complex situation of known and unknown defects coexisting in real-world industrial scenarios, the SPCAD and PCAD test sets were combined to construct four comprehensive test sets. The sample size of each test set is shown in Table 2. During the model training phase, the SPCAD dataset was used for training the unsupervised learning branch, and the PCAD dataset was used for training the supervised learning branch, allowing each branch to focus on its own task characteristics. Fig. 6 shows image examples from the SPCAD and PCAD datasets, respectively.

Table 2: Number of images in the four comprehensive test sets.

Test Set	Normal Samples	Known Defect Images (PCAD)	Unknown Defect Images (SPCAD)	Total Defective Images	Purpose
Integrated_1	50	10	24	34	Higher proportion of unknown defects
Integrated_2	50	12	23	35	Higher proportion of unknown defects

(Continued)

Table 2 (continued)

Test Set	Normal Samples	Known Defect Images (PCAD)	Unknown Defect Images (SPCAD)	Total Defective Images	Purpose
Integrated_3	44	20	11	31	Higher proportion of known defects
Integrated_4	41	21	10	31	Higher proportion of known defects

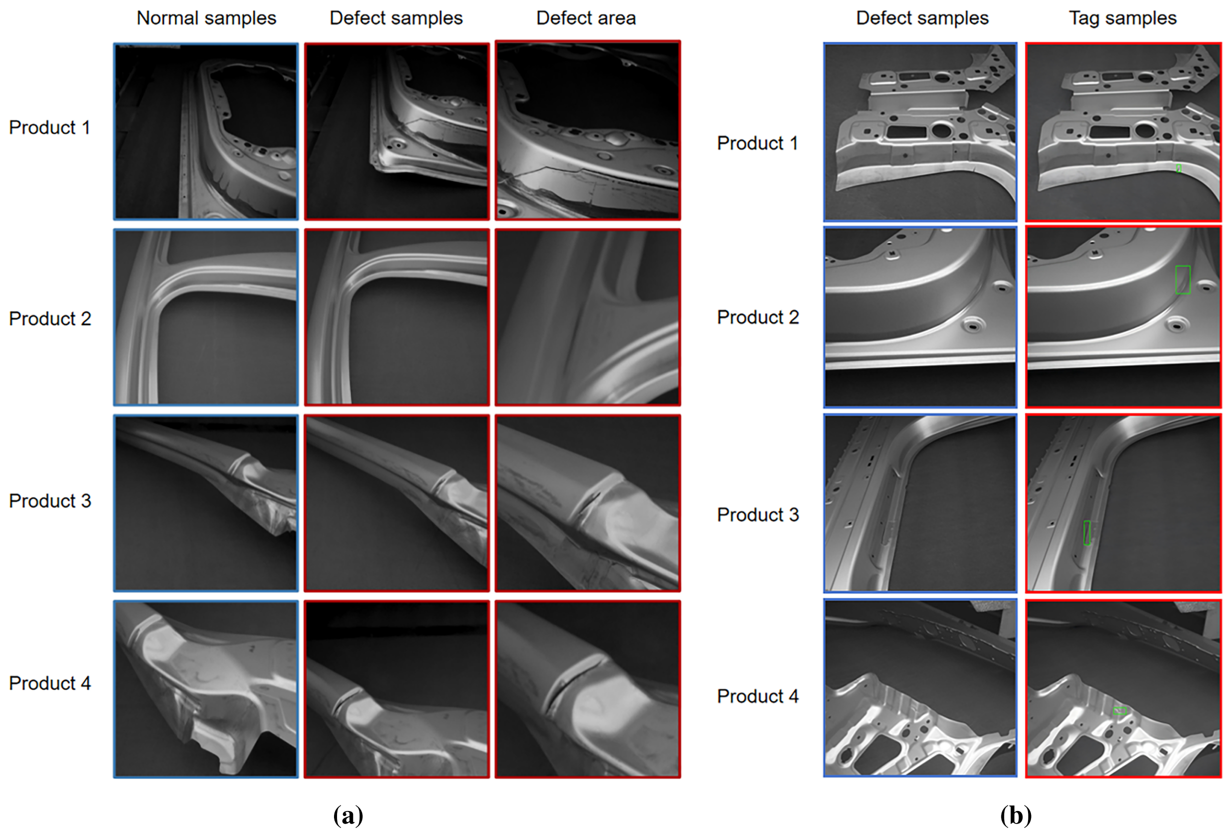


Figure 6: (**a**) SPCAD dataset image examples; (**b**) PCAD dataset image examples.

3.3 Parameter Selection

The supervised learning branch of the HUSLFDD model employs the YOLOv8 architecture for object recognition tasks. Specifically, the backbone network consists of a series of convolutional layers, dual-feature fusion convolutional layers, and fast spatial pyramid pooling layers, designed for efficient image feature extraction. The detection head integrates additional convolutional layers and upsampling layers to achieve accurate localization of targets at different scales. For the unsupervised learning branch, HUSLFDD innovatively reuses the backbone network from the supervised learning branch as a powerful feature extraction network, subsequently connecting a feature adaptation network containing convolutional operations and a fully connected layer as a feature discrimination network. Given that the performance of deep learning models is influenced by numerous training parameters, such as the input image size, batch size, learning rate, number of training epochs, and the choice of optimization algorithm, the parameter selection for the

HUSLFDD supervised learning branch during the training phase is shown in Table 3, and the parameters for each module of the model are shown in Table 4.

Table 3: Training parameters of the supervised learning branch.

Supervised Learning Branch Training Parameters	Parameter Value
Input dimensions	640 × 640
Batch size	16
Number of training rounds	200
Optimizer	Auto
Learning rate	0.01

Table 4: Parameters of the supervised learning module.

Supervised Learning Module	Input Dimensions	Output Size
Backbone network	640 × 640 × 3	80 × 80 × 256 40 × 40 × 512 20 × 20 × 512
Detection head	80 × 80 × 256 40 × 40 × 512 20 × 20 × 512	80 × 80 × 256 40 × 40 × 512 20 × 20 × 512

The training parameters of the unsupervised learning branch of the HUSLFDD model proposed in this paper are shown in Table 5. Since the unsupervised learning branch involves adding noise to the features extracted from normal samples, the noise settings are also listed in the table. After obtaining sample features from the backbone network, the unsupervised learning branch of the HUSLFDD model needs to adapt the sample input features to the feature discrimination network and the feature input feature adaptation network. Table 6 shows the input and output channel sizes of the above two modules, and the detailed parameters of each module structure are shown in Tables 7 and 8.

Table 5: Training parameters of the unsupervised learning branch.

Unsupervised Learning Branch Training Parameters	Parameter Value
Input dimensions	640 × 640
Batch size	16
Number of training rounds	150
Optimizer	Adaptive Moment Estimation (Adam)
Learning rate	0.0001 (Feature Adaptation Network), 0.0002 (Feature discrimination network)
Noise specifications	Gaussian noise N~(0, 0.015)

Table 6: Input and output channel sizes of the module.

Module	Input Channel	Output Channel
Feature Adaptation Network	384	384
Feature Discrimination Network	384	1

Table 7: Structure and parameters of the adapter.

Feature Adaptation Network	Input Channel	Output Channel
Adaptive average pooling layer	(Feature length, 1, −1)	384
Fully connected layer	384	384

Table 8: Structure and parameters of the discriminator.

Feature Discrimination Network	Input Channel	Output Channel
Fully connected layer 1	384	384
Fully connected layer 2	384	1

The HUSLFDD model proposed in this paper uses a scorer that comprehensively considers the judgment results of supervised and unsupervised learning branches for stamping test samples, and finally classifies samples with scores exceeding the defect threshold as defective samples. Given that the detection capabilities of the two branches may differ across the four different test sets, the scorer specifically adjusts the weights of the results based on the unsupervised and supervised learning branches for the four comprehensive test sets. The weights of the unsupervised learning branch results and the defect score thresholds for each stamping test set are shown in Table 9.

Table 9: Parameters of the scorer.

Scorer Parameters	Parameter Value
Integrated_1 Unsupervised Learning Branch Weights	0.8
Integrated_2 Unsupervised Learning Branch Weights	0.8
Integrated_3 Unsupervised Learning Branch Weights	0.3
Integrated_4 Unsupervised Learning Branch Weights	0.3
Defect score threshold	0.1

3.4 Evaluation Index

This paper selects image-level AUROC, recall, and F1 score as core evaluation metrics, and accuracy as a secondary supplementary metric to achieve a comprehensive evaluation of model performance. Image-level AUROC is obtained by plotting the ROC curve and calculating the area under the curve, which can intuitively reflect the overall detection performance and generalization ability of the model. Recall represents the proportion of samples correctly identified by the model among all real defect samples, directly reflecting the coverage of defect samples. The calculation formula is:

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

where TP represents the number of samples that actually contain defects and are correctly predicted as defects, and FN represents the number of samples that actually contain defects but are misclassified as normal. The F1 score is used to balance precision and recall, and is calculated using the following formula:

$$F1\,score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{15}$$

Precision represents the proportion of true positive samples among those predicted as positive. In industrial stamping parts production, missed defects can lead to significant time and cost losses in subsequent processes, making recall a crucial indicator. Misjudging normal workpieces also results in waste; the F1 score can account for both types of errors, providing a balanced evaluation. For class-imbalanced data with a very low proportion of defective samples, image-level AUROC can provide a stable and reliable global performance assessment; accuracy calculation is intuitive and reflects overall prediction correctness, serving as a rapid auxiliary evaluation basis.

4 Experiments and Analysis

4.1 Model Comparison Experiment

The HUSLFDD model proposed in this paper is compared with classic anomaly detection methods such as LGSS [21], PaDiM [22], CFA [23], FastFlow [24], CS-Flow [25], MMR [26], RD4AD [27], DRA [28], DevNet [29], BGAD [30], FMR-YOLO [10], PatchCore [31], GANomaly [32], ViT-AE [33], ToCoAD [18]. To verify the effectiveness of the proposed HUSLFDD model in the task of detecting defects in stamped parts, it was compared with 11 unsupervised anomaly detection methods, including PaDiM, CFA, FastFlow, CS Flow, MMR, RD4AD, LGSS, PatchCore, GANomaly, ViT-AE, and ToCoAD, and 4 supervised methods, including DRA, DevNet, BGAD, and FMR-YOLO, on four sets of stamped part test sets. Image-level AUROC was used as the quantitative evaluation index. The comparison results of image-level AUROC of each model on the four stamped part test sets are shown in Tables 10 and 11.

Table 10: Comparison of unsupervised learning-based defect detection methods: image-level AUROC (%).

Dataset	Integrated_1	Integrated_2	Integrated_3	Integrated_4
PaDiM	77.9	80.2	71.6	77.2
CFA	71.7	75.0	70.5	77.8
FastFlow	64.7	79.3	75.7	85.1
CS-Flow	77.9	81.2	76.3	86.7
MMR	75.0	82.7	79.2	85.2
RD4AD	76.5	80.2	74.0	83.2
PatchCore	79.1	80.1	72.3	83.5
GANomaly	83.7	79.8	80.1	78.7
ViT-AE	81.2	81.9	78.7	79.3
ToCoAD	84.5	79.8	79.4	85.7
LGSS	85.7	82.1	83.0	88.8
HUSLFDD	85.4	82.7	80.6	92.4

Table 11: Comparison of supervised learning-based defect detection methods: image-level AUROC (%).

Dataset	Integrated_1	Integrated_2	Integrated_3	Integrated_4
DRA	86.8	80.2	71.6	77.2
DevNet	71.7	75.0	70.5	77.8
BGAD	64.7	79.3	75.7	85.1
FMR-YOLO	84.9	80.7	79.8	87.3
HUSLFDD	85.4	82.7	80.6	92.4

Unsupervised comparative experiments show that HUSLFDD performs excellently under all integration settings. The Integrated_4 test set demonstrates the best performance, with HUSLFDD achieving an AUROC of 92.4%, 3.6% higher than the second-best method, LGSS, proving its superior performance in label-free defect detection. On the other test sets, HUSLFDD is highly competitive: in Integrated_1, its AUROC is 85.4%, essentially on par with LGSS (85.7%); in Integrated_2, its image-level AUROC is consistent with MMR (82.7%); and in Integrated_3, its AUROC of 80.6% is second only to LGSS (83.0%), significantly outperforming other methods. Furthermore, the AUROC improves by 7% from Integrated_1 to Integrated_4, indicating that the integration strategy effectively achieves continuous performance optimization. In summary, HUSLFDD performs best under optimal configurations, exhibits stable performance across different settings, and demonstrates good robustness and performance ceiling.

Supervised comparative experiments show that HUSLFDD outperforms all ensemble test sets. In the Integrated_4 test set, its AUROC reaches 92.4%, significantly surpassing supervised methods such as DRA, DevNet, and BGAD, making it the optimal model for this configuration. On other test sets, HUSLFDD performs excellently: in Integrated_1, its image-level AUROC of 85.4% is slightly lower than DRA (86.8%), but significantly better than DevNet, BGAD and FMR-YOLO; in Integrated_2, its 82.7% AUROC is higher than all compared methods, making it the best on this dataset; in Integrated_3, its 80.6% AUROC also leads all compared supervised methods. From Integrated_1 to Integrated_4, its AUROC steadily improves, validating the effectiveness and performance gains of the ensemble strategy in supervised scenarios. In summary, HUSLFDD outperforms mainstream supervised methods, exhibiting stable performance, a significant upper limit, and excellent defect detection capabilities in labeled scenarios.

4.2 Ablation Experiment and Result Analysis

The proposed HUSLFDD model employs a detection architecture with independent training of supervised and unsupervised learning branches, followed by weighted fusion during the testing phase. To verify the effectiveness of the dual-branch fusion mechanism, while maintaining other parameters consistent, two control groups were set up: one with only the unsupervised learning branch (ULO) and the other with only the supervised learning branch (SLO). These were compared with the complete HUSLFDD model using ablation techniques. The experimental results are shown in Tables 12–15.

Experimental results show that a single branch has significant limitations: unsupervised branches struggle to accurately detect specific defects, while supervised branches exhibit poor generalization ability when annotations are insufficient. HUSLFDD outperforms the single-branch approach across all four test sets, achieving higher AUROC in Integrated_1, improving F1 scores by 3.7% and 9.0% in Integrated_2, significantly enhancing AUROC and F1 in Integrated_3, and substantially improving accuracy and F1 in Integrated_4. This model, through complementary dual-branch architecture, balances high recall with high accuracy, effectively balancing false negatives and false positives, validating the effectiveness and necessity of the dual-branch fusion mechanism, and demonstrating practical value in industrial defect detection.

Table 12: Performance comparison between single-branch method and HUSLFDD on the integrated_1 test set (%).

Model	F1 Score	Image-Level AUROC	Recall Rate	Accuracy
ULO	94.2	83.6	100.0	90.2
SLO	87.7	89.0	78.1	82.5
HUSLFDD	94.2	95.2	100.0	90.2

Table 13: Performance comparison between single-branch method and HUSLFDD on the integrated_2 test set (%).

Model	F1 Score	Image-Level AUROC	Recall Rate	Accuracy
ULO	87.4	82.7	87.0	81.0
SLO	82.1	82.6	72.2	76.1
HUSLFDD	91.1	89.3	94.4	85.9

Table 14: Performance comparison between single-branch method and HUSLFDD on the integrated_3 test set (%).

Model	F1 Score	Image-Level AUROC	Recall Rate	Accuracy
ULO	91.0	87.3	92.9	85.8
SLO	86.4	88.0	77.6	81.1
HUSLFDD	93.0	95.6	94.9	89.0

Table 15: Performance comparison between single-branch method and HUSLFDD on the integrated_4 test set (%).

Model	F1 Score	Image-Level AUROC	Recall Rate	Accuracy
ULO	89.6	90.4	98.8	83.3
SLO	85.0	81.9	78.3	79.8
HUSLFDD	94.1	92.4	96.4	91.2

4.3 Ablation Study on the Shared Backbone and Fusion Strategy

To verify the advantages of weighted fusion with a shared backbone network, this paper sets up a simple fusion method (Simple Combination of Two Approaches, SCTA) as a control: this method inputs the test images into independently trained unsupervised and supervised models respectively, directly concatenates the two detection results, without sharing the backbone network or performing feature-level fusion. In contrast, HUSLFDD performs one shared feature extraction and then sends the shared feature maps to both branches. Therefore, the comparison with SCTA not only evaluates result-level fusion, but also verifies whether shared representation learning can reduce redundant computation and improve the consistency of dual-branch decisions. The results of comparing SCTA and HUSLFDD on four stamping part test sets are shown in Table 16.

Table 16: Performance comparison between SCTA and HUSLFDD (%).

Test Set	Model	F1 Score	Image-Level AUROC	Recall Rate	Accuracy
Integrated_1	**SCTA**	94.0	93.9	100.0	90.0
Integrated_1	**HUSLFDD**	94.2	95.2	100.0	90.2
Integrated_2	**SCTA**	89.7	82.7	96.3	83.1
Integrated_2	**HUSLFDD**	91.1	89.3	94.4	85.9
Integrated_3	**SCTA**	92.6	94.0	95.9	88.2
Integrated_3	**HUSLFDD**	93.0	95.6	94.9	89.0
Integrated_4	**SCTA**	90.6	87.9	98.8	85.1
Integrated_4	**HUSLFDD**	94.1	92.4	96.4	91.2

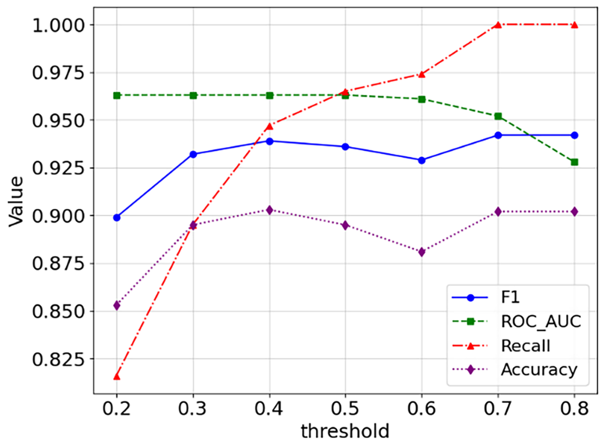
Experiments show that HUSLFDD outperforms SCTA in F1 score, image-level AUROC, and accuracy. Specifically, Integrated_4 shows a 3.5% improvement in F1 score and a 6.1% improvement in recall; Integrated_2 shows a 6.6% improvement in image-level AUROC. While HUSLFDD's recall decreases slightly, the improved precision leads to a significant improvement in F1 score, resulting in better overall classification. In terms of speed, SCTA achieves 8.65 FPS with a single image processing time of approximately 115.6 ms, while HUSLFDD reaches 9.14 FPS with a single image processing time of approximately 109.4 ms, demonstrating higher inference efficiency.

4.4 Branch Weight Selection Experiment

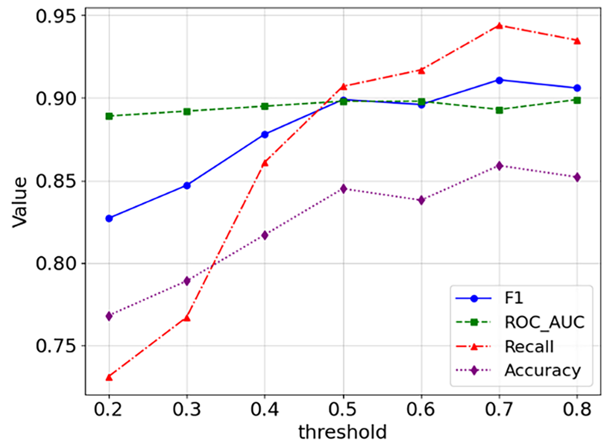
The image scorer of the HUSLFDD model fuses the outputs of unsupervised and supervised branches with weights, and the branch weights directly affect the detection results. To determine the optimal weight ratio, other parameters were fixed, and the weights of the unsupervised branch were successively adjusted under the constraint that the sum of the weights of the two branches is 1. Experiments were carried out on four stamping part test sets, and the results are shown in Fig. 7.

On the Integrated_1 test set, recall continuously increases with increasing weight, reaching 1.0 with a weight of 0.7 and maximizing image-level AUROC; therefore, this weight was selected. On the Integrated_2 test set, all metrics show an overall upward trend, with 0.7 remaining the optimal weight. On the Integrated_3 test set, all metrics perform evenly with a weight of 0.3; further increases would lead to a decrease in AUROC, so 0.3 was chosen. On the Integrated_4 test set, the F1 score and precision peak with a weight of 0.3, and both AUROC and recall exceed 0.9, meeting the detection requirements.

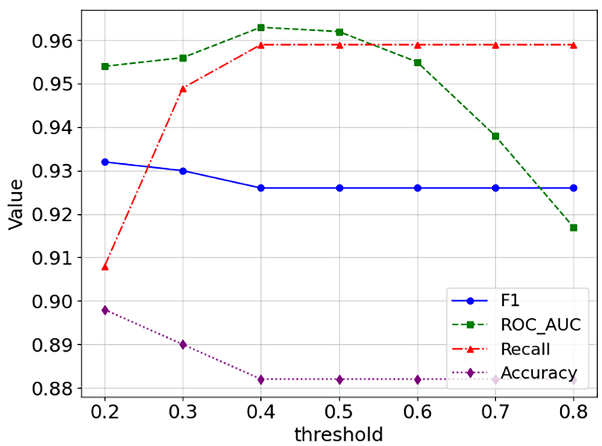
The difference in weight allocation stems from the dataset distribution: Integrated_1 and Integrated_2 have a higher proportion of unknown defects, and the unsupervised branch is more capable of detecting new defects, thus receiving higher weights; Integrated_3 and Integrated_4 have a predominance of known defects, and the supervised branch's discrimination is more reliable, therefore receiving lower weights. In practical industrial applications, the weights can be flexibly adjusted based on indicator preferences and the probability of unknown defects occurring.



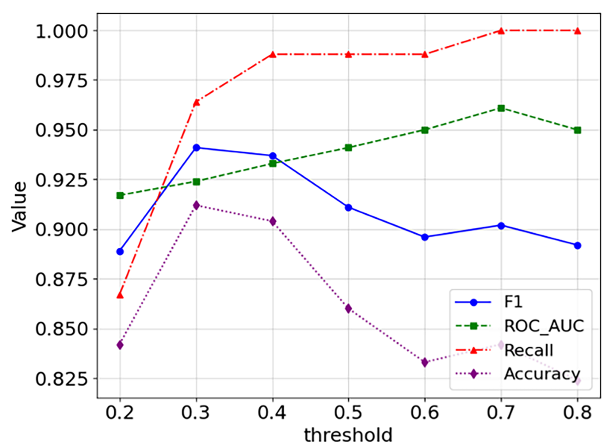
(a) Integrated_1 test set



(b) Integrated_2 test set



(c) Integrated_3 test set



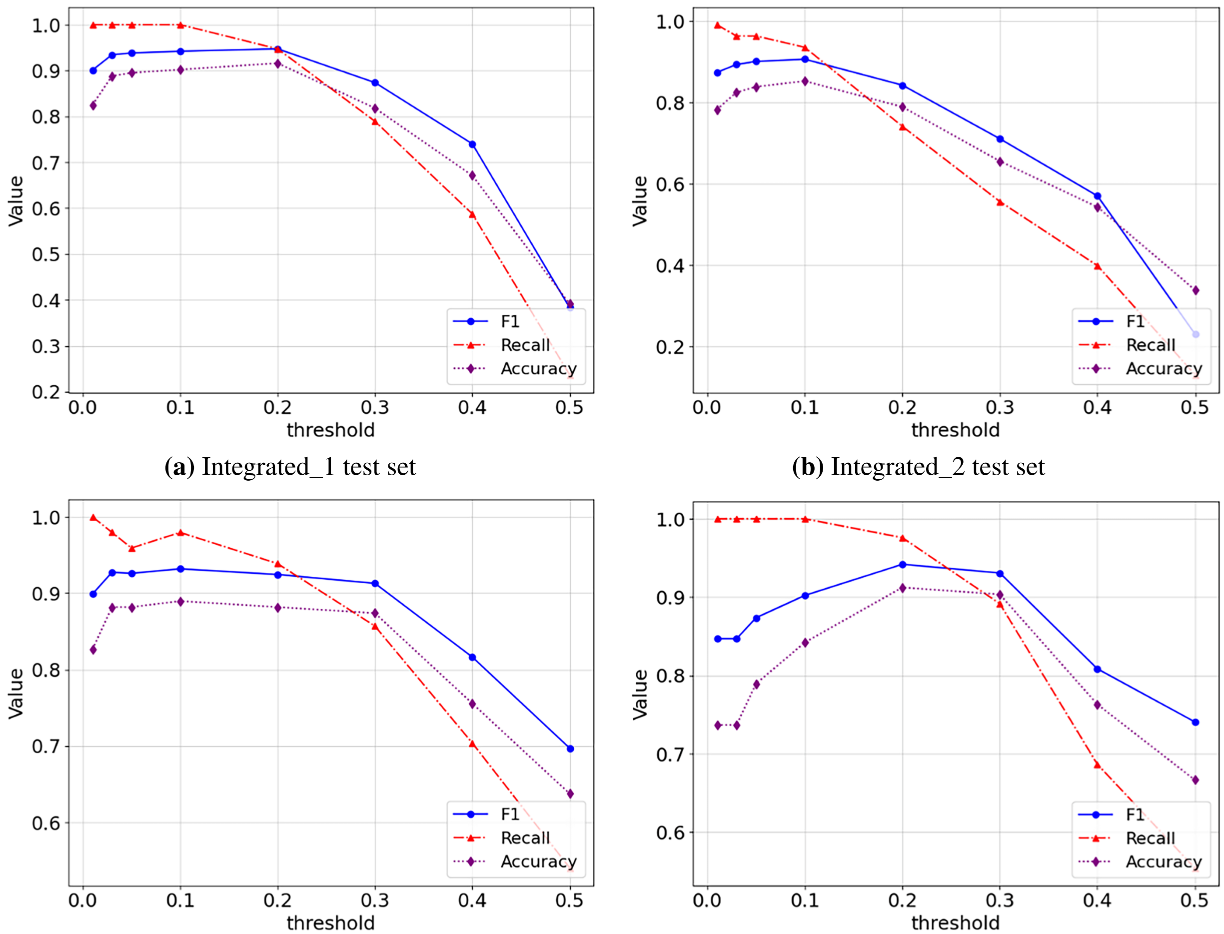
(d) Integrated_4 test set

Figure 7: Impact of branch weights in unsupervised learning.

4.5 Defect Threshold Effect Experiment

To analyze the impact of the defect score threshold on the model's detection performance, while keeping other conditions consistent, the classification threshold was gradually increased from 0.1 to 0.9 in increments of 0.1. Experiments were conducted on four stamping part test sets, and the results are shown in Fig. 8.

Experiments show that when the threshold is below 0.1, the model has a high recall rate, but low precision and F1 score, making it prone to misclassifying normal workpieces as defects. At a threshold of 0.1, all indicators are generally balanced. As the threshold increases, the judgment criteria become stricter, the risk of missed detections increases, and the overall recall rate decreases. Specifically, the F1 score and precision of the Integrated_2 test set continuously decrease with increasing threshold; the F1 score and precision of the Integrated_1, Integrated_3, and Integrated_4 test sets all initially increase slightly and then gradually decrease. Based on these experimental results, a lower defect score threshold is more suitable for industrial inspection scenarios with high recall requirements and a tolerable false positive rate.



(a) Integrated_1 test set **(b)** Integrated_2 test set

(c) Integrated_3 test set **(d)** Integrated_4 test set

Figure 8: Impact of defect threshold.

4.6 Limitations and Industrial Deployment

Although HUSLFDD achieves satisfactory performance on both known and unknown defect detection, several limitations remain. First, although the dataset was collected from real industrial production lines, extremely complex lighting variations, such as severe reflections, shadows, and illumination fluctuations, may still affect feature extraction and anomaly discrimination. Second, very subtle defects with extremely weak visual characteristics remain challenging for both supervised detection and unsupervised anomaly learning. Third, although the proposed dual-branch framework alleviates the influence of limited defect samples, severe class imbalance may still affect the learning of rare defect categories. Finally, significant changes in production environments, such as new products, different cameras, or imaging conditions, may introduce domain shifts that reduce model performance.

From the perspective of industrial deployment, the proposed HUSLFDD framework can be deployed on mobile terminals equipped with modern NVIDIA GPUs. In this study, all experiments were conducted on an RTX 4050 Laptop GPU, and the proposed method achieved an inference speed of 9.14 FPS, which satisfies the real-time inspection requirements of medium-speed automotive stamping production lines. Since the supervised and unsupervised branches share a common backbone network, redundant feature extraction is avoided, reducing computational overhead and memory consumption compared with independently deployed dual-model solutions.

5 Conclusion

This study focuses on the core challenges in automotive stamping part defect detection, including the complexity of defect types, large scale spans, and the interweaving of known and unknown defects. Furthermore, a single learning paradigm struggles to balance detection accuracy and generalization ability. To address these challenges, an unsupervised-supervised fusion model for stamping part defect detection, HUSLFDD, is proposed, aiming to provide reliable technical support for intelligent quality control in industrial settings. To resolve the issues of redundant feature extraction information and easy loss of multi-scale features in the backbone network, the model employs a lightweight multi-module fusion backbone composed of CBS, C2f, and SPPF. This ensures lightweight design while efficiently capturing multi-scale features of stamping part images, effectively solving the problem of balancing feature extraction efficiency and completeness. To address the imbalance in detection accuracy across different scales and insufficient feature fusion, a bidirectional feature aggregation network is constructed based on PAN FPN. This network achieves efficient bidirectional aggregation of shallow, medium, and deep multi-scale features, alleviating the significant difference in detection accuracy between large-scale and small defects. To address the pain points of inaccurate localization of known defects and inability to identify unknown defects, a dual-branch collaborative mechanism is adopted. The supervised branch focuses on accurate classification of known defects, while the unsupervised branch captures potential patterns of unknown defects. An adaptive weighted image scorer dynamically fuses the results of both branches to unify the determination of known and unknown defects. Experiments and ablation analysis validate that the proposed model effectively overcomes the limitations of single learning methods, achieving systematic improvements in feature preservation, feature fusion, and defect discrimination, providing a reliable technical solution for intelligent inspection of stamped parts. Future research will further address more complex industrial scenarios, focusing on robust defect detection under low light, multi-view, and cross-condition conditions. It will also explore new paradigms for defect detection, including unsupervised cross-domain generalization, lightweight mobile-terminal deployment, and industrial large-scale model empowerment, promoting more stable and universal application of the model in actual production lines.

Acknowledgement: Not applicable.

Funding Statement: This research was supported by the Basic Research Program of Liaoning Provincial Department of Education (LJ212510144014) and the Natural Science Foundation of Ningxia (2024AAC03349).

Author Contributions: The authors confirmed their contributions to this paper as follows: Zhikai Chi: model architecture design and drafting of the initial manuscript; Jiaxu Ning: dataset retrieval, provision of research resources and analysis results; Delong Zhang and Changsheng Zhang: rigorous review of the technical content and approval of the final version. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: Data will be available on request.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Berkmans F, Bartkowiak T, Grochalski K, Wieczorowski M, Bigerelle M. Curvature-based multiscale feature extraction for surface quality inspection in manufacturing. Int J Adv Manuf Technol. 2026;142(9):4567–92. doi:10.1007/s00170-025-17145-8.
2. Ke H. Using convolutional neural networks for material surface quality inspection and classification. Int J Comput Intell Syst. 2025;18(1):224. doi:10.1007/s44196-025-00951-z.
3. El Wahabi A, Mesquiny A, El Ahmadi O, Baraka IH, Hamdoune S, Boudhir AA. Enhancing quality inspection in automotive manufacturing through deep learning and transfer learning. Neural Comput Appl. 2025;37(18):11711–36. doi:10.1007/s00521-024-10534-2.
4. Duarte J, Claro MF, Vitoriano PMA, Amaral TG, Pires VF. Detection and classification of defects on metal surfaces based on a lightweight YOLOX-tiny COCO network. Eng. 2025;6(11):302. doi:10.3390/eng6110302.
5. Wang Y, Zhang L, Zhao X, Tang B, Yang W. Review of research on ceramic surface defect detection based on deep learning. Electronics. 2025;14(12):2365. doi:10.3390/electronics14122365.
6. Tuyboyov O, Muxiddinov Z, Shamiliddin S, Mahliyo A. Enhanced surface defect detection in industrial manufacturing using convolutional neural networks and advanced imaging techniques. Diffus Found Mater Appl. 2025;38(4):39–50. doi:10.4028/p-ksfub8.
7. Zhang JY, Huang L, Guan YJ. Real-time defect detection in concrete structures using attention-based deep learning and GPR imaging. Sci Rep. 2025;15(1):35507. doi:10.1038/s41598-025-19596-1.
8. Ansari MA, Crampton A, Garrard R, Cai B, Attallah M. A convolutional neural network (CNN) classification to identify the presence of pores in powder bed fusion images. Int J Adv Manuf Technol. 2022;120(7):5133–50. doi:10.1007/s00170-022-08995-7.
9. Latham S, Giannetti C. A tool to combine expert knowledge and machine learning for defect detection and root cause analysis in a hot strip mill. SN Comput Sci. 2023;4(5):628. doi:10.1007/s42979-023-02104-5.
10. Ni Y, Wu Q, Zhang X. FMR-YOLO: an improved YOLOv8 algorithm for steel surface defect detection. IET Image Process. 2025;19(1):e70009. doi:10.1049/ipr2.70009.
11. Liu G, Chu M, Gong R, Zheng Z. Global attention module and cascade fusion network for steel surface defect detection. Pattern Recognit. 2025;158(1):110979. doi:10.1016/j.patcog.2024.110979.
12. Shu X, Guo Y, Yang W, He Z, Sheng L, Lu X. Review of wafer defect detection based on supervised learning algorithms. Eng Comput. 2026;43(4):1566–93. doi:10.1108/ec-02-2025-0184.
13. Luo W, Yao H, Yu W, Li Z. AMI-net: adaptive mask inpainting network for industrial anomaly detection and localization. arXiv:2412.11802. 2024.
14. Deng S, Sun Z, Zhuang R, Gong J. Noise-to-norm reconstruction for industrial anomaly detection and localization. Appl Sci. 2023;13(22):12436. doi:10.3390/app132212436.
15. Yang Q, Guo R. An unsupervised method for industrial image anomaly detection with vision transformer-based autoencoder. Sensors. 2024;24(8):2440. doi:10.3390/s24082440.
16. Peng J, Shao H, Xiao Y, Cai B, Liu B. Industrial surface defect detection and localization using multi-scale information focusing and enhancement GANomaly. Expert Syst Appl. 2024;238(3):122361. doi:10.1016/j.eswa.2023.122361.
17. Roth K, Pemula L, Zepeda J, Schölkopf B, Brox T, Gehler P. Towards total recall in industrial anomaly detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 14298–308. doi:10.1109/CVPR52688.2022.01392.
18. Liang Y, Hu Z, Huang J, Di D, Su A, Fan L. ToCoAD: two-stage contrastive learning for industrial anomaly detection. IEEE Trans Instrum Meas. 2025;74:1–9. doi:10.1109/tim.2025.3545987.
19. Wang K, Liang X, Xu J, Zhang S, Tan J. Style-augmented large-scale vision model with domain-generalized knowledge fusion for anomaly detection in powder bed additive manufacturing. Inf Fusion. 2026;130(1):104108. doi:10.1016/j.inffus.2025.104108.
20. Wang K. Contrastive learning-based semantic segmentation for *in-situ* stratified defect detection in additive manufacturing. J Manuf Syst. 2023;68(7):465–76. doi:10.1016/j.jmsy.2023.05.001.

21. Dong X, Zhang C, Wang D, Guo Q, Deng X, Li C. Inspection of cracking in stamping parts surfaces using anomaly detection. Eng Appl Artif Intell. 2025;143(1):110006. doi:10.1016/j.engappai.2025.110006.
22. Defard T, Setkov A, Loesch A, Audigier R. PaDiM: a patch distribution modeling framework for anomaly detection and localization. In: Pattern recognition. ICPR international workshops and challenges. Cham, Switzerland: Springer International Publishing; 2021. p. 475–89. doi:10.1007/978-3-030-68799-1_35.
23. Lee S, Lee S, Song BC. CFA: coupled-hypersphere-based feature adaptation for target-oriented anomaly localization. IEEE Access. 2022;10:78446–54. doi:10.1109/ACCESS.2022.3193699.
24. Yu J, Zheng Y, Wang X, Li W, Wu Y, Zhao R, et al. FastFlow: unsupervised anomaly detection and localization via 2D normalizing flows. arXiv:2111.07677. 2021.
25. Rudolph M, Wehrbein T, Rosenhahn B, Wandt B. Fully convolutional cross-scale-flows for image-based defect detection. In: Proceedings of the 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2022 Jan 3–8; Waikoloa, HI, USA. Piscataway, NJ, USA: IEEE; 2022. p. 1829–38. doi:10.1109/WACV51458.2022.00189.
26. Zhang Z, Zhao Z, Zhang X, Sun C, Chen X. Industrial anomaly detection with domain shift: a real-world dataset and masked multi-scale reconstruction. Comput Ind. 2023;151(6):103990. doi:10.1016/j.compind.2023.103990.
27. Deng H, Li X. Anomaly detection via reverse distillation from one-class embedding. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 9727–36. doi:10.1109/cvpr52688.2022.00951.
28. Ding C, Pang G, Shen C. Catching both gray and black swans: open-set supervised anomaly detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. Piscataway, NJ, USA: IEEE; 2022. p. 7378–88. doi:10.1109/cvpr52688.2022.00724.
29. Pang G, Shen C, Hengel AVD. Deep anomaly detection with deviation networks. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2019 Aug 4–8; Anchorage, AK, USA. New York, NY, USA: ACM; 2019. p. 353–62. doi:10.1145/3292500.3330871.
30. Yao X, Li R, Zhang J, Sun J, Zhang C. Explicit boundary guided semi-push-pull contrastive learning for supervised anomaly detection. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. Piscataway, NJ, USA: IEEE; 2023. p. 24490–9. doi:10.1109/CVPR52729.2023.02346.
31. Wang L, Ma FL, Yu QS, Wei MY. Research on anomaly detection algorithms for scientific satellite images based on PatchCore. Front Data Comput. 2025;7(4):143–54. (In Chinese). doi:10.11871/jfdc.issn.2096-742X.2025.04.012.
32. Liu Y, Tan M, Gao Y, Dai J. Research on GANomaly anomaly detection method based on double discriminant. J Univ Electron Sci Technol China. 2024;53(4):568–75. (In Chinese). doi:10.12178/1001-0548.2023157.
33. Aslam MM, Mehmood MZ, Khan SM, Akram MU, Zeb J. ViT-AE: a self-supervised vision transformer autoencoder for baggage threat localization. In: Proceedings of the 2025 4th International Conference on Communication, Computing and Digital Systems (C-CODE); 2025 Oct 1–2; Islamabad, Pakistan. Piscataway, NJ, USA: IEEE; 2025. p. 1–6. doi:10.1109/C-CODE67372.2025.11204130.