ISSN: 2710-0588

LwDetFormer: A Hybrid Mobile and Transformer-Based Deep Learning Model for Lockwire Object Detection

Jinning Zhang*, Xiangguo Sun

Mechanical Engineering College, Sichuan University of Science and Engineering, Yibin, Sichuan, China

*Corresponding author: Xiangguo Sun

Abstract

Lockwire plays a crucial role in various industries, including aviation, automotive, power, oil, and gas, by ensuring the secure fastening of critical components. However, detecting Lockwire objects poses significant challenges due to their reflective metallic materials and surface textures that closely resemble the surrounding environment. Traditional object detection methods struggle with these characteristics, especially when working with small datasets and variable lighting conditions. To address these issues, we propose LwDetFormer, a hybrid CNN-Transformer model designed for high-precision detection of Lockwire objects in complex backgrounds. LwDetFormer integrates MobileNet's local feature extraction capabilities with Transformer's global feature modeling abilities, enhancing accuracy and robustness. The model includes innovative modules such as Spatial Pyramid Pooling Focus, Feature Enhancement Module, Feature Fusion Module, and Spatial Context Awareness Module. Experimental results on a test set containing 2,000 images show that LwDetFormer outperforms advanced models like YOLOv8, MobileNetv3, and MobileFormer in terms of precision and recall, achieving a precision rate of 95.9% and a recall rate of 92.0%. These findings highlight LwDetFormer's potential for improving safety and efficiency in the inspection of aircraft engine fuse parts.

Keywords

Lockwire Object Detection, Small Target Detection, Complex Backgrounds.

1. Introduction

Lockwire has a wide range of applications, including but not limited to industries such as aviation, automotive, power, oil, and gas. In the aviation sector, the aerospace manufacturing industry extensively uses Lockwire to ensure the secure fastening of critical components, preventing safety incidents caused by loose parts. Object detection is an important task in the field of computer vision, aimed at identifying and locating objects of interest within images. In the manufacturing and maintenance of electronic devices, accurate identification of Lockwire components for fuses is crucial for ensuring the normal operation of equipment.

Currently, deep learning models applied to Lockwire object detection generally face a challenge: achieving high accuracy detection and recognition on smaller datasets is difficult. This is mainly attributed to the characteristics of Lockwire itself, which predominantly uses reflective metallic materials with surface features that closely resemble the surrounding environment's texture. Consequently, in real production environments, the shape, texture, and background of fuse parts have minimal differences, making it challenging for traditional object detection methods to achieve satisfactory results. Additionally, dataset annotation requires specialized knowledge and is labor-intensive, resulting in limited available annotated dataset sizes and further exacerbating this challenge. Moreover, images collected in real-world environments often

ISSN: 2710-0588

cannot meet the uniform requirements of shooting conditions, and variations in lighting conditions can impact image quality to varying degrees.

To address these issues, this paper proposes a hybrid CNN-Transformer model named LwDetFormer. This model aims to achieve high precision and robustness against interference under high recall conditions in the task of detecting fuse targets in aviation engines. The goal is to effectively solve the challenges faced in Lockwire object detection.

1.1. Related Work

Mask R-CNN (Region-based Convolutional Neural Network) is a deep learning model for instance segmentation proposed by the team of SUN Junhua[1]. It adds a branch for predicting object masks on top of Faster R-CNN, enabling multi-task learning for object detection, classification, and segmentation. Mask R-CNN has demonstrated excellent performance on the COCO dataset, particularly in handling complex scenes and overlapping objects. However, traditional Mask R-CNN models may encounter accuracy degradation when dealing with small targets like Lockwire that have reflective characteristics and surface textures similar to their backgrounds. This is mainly due to the convolutional neural network part struggling to fully capture the subtle features of such targets.

MobileNet[2], proposed by Howard et al., is a lightweight convolutional neural network designed to run on mobile and embedded devices. It significantly reduces computational load and model parameters through depthwise separable convolutions and inverted residual structures while maintaining high accuracy. Models in the MobileNet series, such as MobileNetV2 and MobileNetV3, have achieved notable results in tasks like image classification, object detection, and semantic segmentation. However, despite its excellence in handling local features, MobileNet's global feature extraction capability is relatively weak, which may limit the detection accuracy of small targets like Lockwire in complex backgrounds.

Vision Transformer (ViT)[3], proposed by Dosovitskiy et al., is a visual model based on the Transformer architecture. ViT segments images into a series of fixed-size patches (tokens) and models global relationships among these patches via a Transformer encoder to achieve image classification. Pre-trained on large datasets such as ImageNet-21k and JFT-300M, ViT exhibits outstanding performance. However, when trained from scratch on medium-sized datasets like ImageNet, ViT's performance typically falls short compared to convolutional neural networks (CNNs). This is primarily because its simple tokenization process fails to effectively model local structural information in images, such as edges and lines, leading to lower training sample efficiency.

MobileFormer[4], proposed by Chen et al., is a hybrid model that combines the advantages of MobileNet and Transformer through a parallel structure and bidirectional bridging mechanism, achieving effective integration of local processing and global interaction. MobileFormer has achieved significantly better performance than single MobileNet or Transformer models in tasks like image classification and object detection. Its key innovation lies in using very few tokens (e.g., 6 or fewer) as input to the Transformer, thereby greatly reducing computational costs. Additionally, MobileFormer enables bidirectional feature fusion between MobileNet and Transformer through a lightweight cross-attention mechanism. However, the performance of MobileFormer in handling small targets with complex backgrounds and reflective characteristics, such as Lockwire, still needs further validation.

Given the strengths and weaknesses of the aforementioned models, this paper proposes a hybrid CNN-Transformer model named LwDetFormer, specifically for Lockwire object detection. This model combines the local feature extraction capabilities of MobileNet and the global feature modeling capabilities of Transformer, achieving high-precision detection of Lockwire in complex backgrounds through parallel structure and bidirectional feature fusion mechanisms. The LwDetFormer model aims to address the precision and recall challenges

ISSN: 2710-0588

faced by traditional deep learning models in Lockwire object detection, providing strong support for overall inspection of aviation engines.

1.2. Research Gaps and Innovations

Despite significant achievements in the field of object detection by previous researchers, there remain several research gaps and challenges.[5] Issues such as lighting variations and complex backgrounds in real production environments further increase the difficulty of object detection.[6]The following are several problems associated with Lockwire in real production environments, leading to lower accuracy and correct detection rates for Lockwire:

1. Lockwire is predominantly made from reflective metallic materials, whose surface features closely resemble the texture of the surrounding environment, making it easy to cause confusion during identification. This results in high overall false detection and misclassification rates for Lockwire in actual environments.

2. The annotation of datasets requires specialized knowledge and is labor-intensive, resulting in limited available annotated dataset sizes. This poses a challenge for deep learning-based detection methods.

3. Images collected from real-world environments often do not meet the requirements of uniform shooting conditions, and actual collected images will be affected by varying degrees of lighting.

To address these challenges, this paper proposes the LwDetFormer model approach, which is a target detection model based on a hybrid CNN-Transformer architecture.[7] It aims to solve the problems of detection confusion, training with small datasets, and the impact of lighting on detection accuracy in Lockwire object detection.

2. LwDetFormer

2.1. Framework

The LwDetFormer model consists of MobileFormer blocks, Spatial Pyramid Pooling Focus (SPPF), Feature Enhancement Module (FEM), Feature Fusion Module (FFM), Spatial Context Awareness Module (SCAM), and multiple detection heads. The overall structural diagram is shown in the figure below:

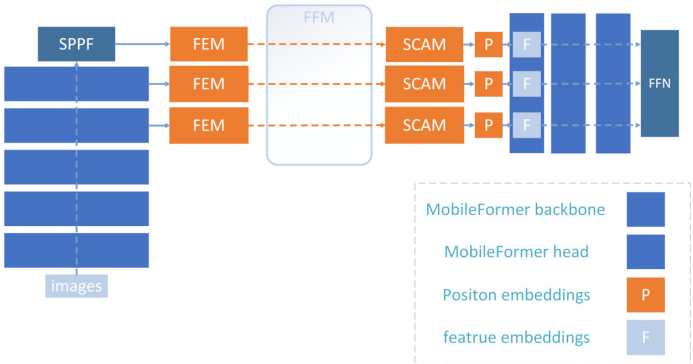

Figure 1. Main Architecture of LwDetFormer

When an input image of size 640×640 is fed into the model, it first passes through a backbone network composed of five MobileFormer blocks and one Spatial Pyramid Pooling Focus (SPPF). The first MobileFormer block accepts the 640×640 image along with global tokens. Up until the third MobileFormer block's output, each MobileFormer block sequentially outputs feature maps and tokens at resolutions of 1/2, 1/4, and 1/8 of the original image size. Subsequently, the next two MobileFormer blocks and the SPPF output feature maps at resolutions of 1/16,

ISSN: 2710-0588

1/32, and 1/64 of the original image size, which are then fed into three Feature Enhancement Modules (FEM) in the neck. These three FEM modules produce three feature maps, X_1, X_2 and X_3 .After being processed by the Feature Fusion Module (FFM), these feature maps become X_{11}, X_{22}, and X_{33}. They are further processed by the Spatial Context Awareness Module (SCAM) to produce the neck's output feature maps, X_{111}, X_{222}, and X_{333}.

The three feature maps, X_{111}, X_{222}, and X_{333}, output from the neck are then fed into the head. The head consists of three MobileFormer blocks and a Feed-Forward Network (FFN). In the head, the 1/32 MobileFormer block receives feature embeddings from the neck, global tokens from the backbone network, and initially generated 100 object queries. Within the former sub-blocks of this MobileFormer block, position embeddings corresponding to the feature embeddings are calculated. The updated feature embeddings, position embeddings, and refined object queries are then passed on to subsequent MobileFormer blocks.

Finally, the FFN used for prediction generates detection box coordinates and detection classes based on the feature embeddings, position embeddings, and object queries output by the MobileFormer blocks.

2.2. MobileFormer Block

The MobileFormer block consists of four key components: the Mobile sub-block, the Former sub-block, and two cross-attention modules in both directions (Mobile→Former and Mobile←Former). The data input to the MobileFormer block includes a feature map X and a set of global tokens Z.[4]

The feature map X first enters the Mobile sub-block, where it undergoes depthwise separable convolutions and dynamic ReLU activation functions to extract local features. Meanwhile, the global tokens Z enter the Former sub-block, where they are processed through multi-head attention mechanisms and a feed-forward neural network (FFN) to encode global features.

The output feature map X from the Mobile sub-block is then fed into the Mobile→Former cross-attention mechanism, which fuses the local features into the global tokens Z, generating updated global tokens Z' . These updated tokens Z' are subsequently passed through the Former→Mobile cross-attention mechanism, which fuses the global features back into the feature map X, producing an updated feature map X' .

The final outputs of the MobileFormer block include the updated feature map X' and the updated global tokens Z' . These outputs serve as inputs for the next MobileFormer block, continuing to participate in subsequent feature extraction and fusion processes.

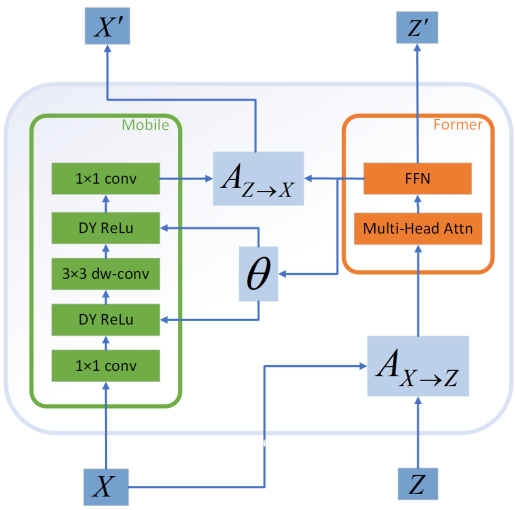

Figure 2. Computation Logic of the MobileFormer Block

ISSN: 2710-0588

The Mobile sub-block employs depthwise separable convolution, which consists of two depthwise convolutions and one pointwise convolution, and uses Dynamic ReLU as the activation function. The Dynamic ReLU obtains parameters processed from all global tokens output by the Former sub-block, which are processed through two layers of Multi-Layer Perceptron (MLP) with two intermediate ReLU layers. Specifically, this can be represented as:

$$Hidden = relu(Z_1 W_1 + b_1) \tag{0.1}$$

$$\theta = relu(Hidden \cdot W_2 + b_2) \tag{0.2}$$

Among these, $W_1 \in R^{\mathrm{d\times h}}$ represents the expansion layer weight matrix, and $W_2 \in R^{\mathrm{h\times 2C}}$ represents the parameter generation layer weight matrix. Global tokens, after being processed by $A_{X\to Z}$ module, are fed into the Former sub-block. After going through the stacked multi-head attention and Feed-Forward Network (FFN) for image encoding in the Former sub-block, they are output as global tokens for the next Mobile-Former block. Additionally, during the output process, the Former sub-block provides a branch. This branch, along with the feature map obtained from the Mobile sub-block after being processed by $A_{X\to Z}$ module, is output as the feature map for the next Mobile-Former block.

The structure of bidirectional bridge that achieves the bidirectional fusion of local and global features through a lightweight cross-attention mechanism can be described as follows:

$$A_{X\to Z} = [Attn(\tilde{z}_i W_i^Q, \tilde{x}_i, \tilde{x}_i)]_{i=1:h} W^O \tag{0.3}$$

$$A_{Z\to X} = [Attn(\tilde{x}_i, \tilde{z}_i W_i^K, \tilde{z}_i W_i^V)]_{i=1:h} \tag{0.4}$$

$$Attn(\tilde{z}_i W_i^Q, \tilde{x}_i, \tilde{x}_i) = soft\max(\frac{(\tilde{z}_i W_i^Q)(\tilde{x}_i)^T}{\sqrt{d_k}}) \cdot \tilde{x}_i \tag{0.5}$$

$$Attn(\tilde{x}_i, \tilde{z}_i W_i^K, \tilde{z}_i W_i^V) = soft\max(\frac{\tilde{x}_i(\tilde{z}_i W_i^K)^T}{\sqrt{d_k}}) \cdot (\tilde{z}_i W_i^V) \tag{0.6}$$

The segment describes a process in which $A_{X\to Z}$ and $A_{Z\to X}$ utilize an attention mechanism to integrate token z_i with feature $\widetilde{x}_i$. The former computes using the query Q from the attention mechanism, treating feature $\widetilde{x}_i$ as both the key and value. The latter obtains and uses keys K and values V through projection matrices W_i^K and W_i^V. This bi-directional bridging ensures effective communication and collaboration between the Mobile component and the Former component, ultimately outputting updated local feature maps X' and global tokens Z'.

2.3. Feature Enhancement Module (FEM)

The FEM[8] is a lightweight and efficient network structure originally designed to enhance the feature representation capability of small objects in remote sensing images. In this model, the FEM is adopted as a module in the neck, receiving outputs from the last two MobileFormer blocks and the SPPF in the backbone network. The specific structure of the FEM module is shown in the figure below:

ISSN: 2710-0588

$$W_1 = f_{3\times3}^{conv}(f_{1\times1}^{conv}(F)) \tag{0.7}$$

$$W_2 = f_{3\times3}^{diconv}(f_{3\times1}^{conv}(f_{1\times3}^{conv}(f_{1\times1}^{conv}(F)))) \tag{0.8}$$

$$W_3 = f_{3\times3}^{diconv}(f_{1\times3}^{conv}(f_{3\times1}^{conv}(f_{1\times1}^{conv}(F)))) \tag{0.9}$$

$$X = Cat(W_1, W_2, W_3) \oplus f_{1\times1}^{conv}(F) \tag{0.10}$$

In this process, F represents the input feature map, and X represents the output feature map. After inputting the feature map F into the FEM, it splits into four branches. The first branch outputs the feature map W1 after undergoing standard convolutions of 1×1 and 3×3. The second branch, after experiencing standard convolutions of 1×1, 1×3, and 1×1, undergoes a 3×3 dilated convolution. The third and fourth branches follow a similar pattern. Finally, the feature maps W_1、W_2 and W_3 output from the first three branches are concatenated together, and the feature map from the fourth branch is added element-wise to this combined output, ultimately producing the output feature map X.

2.4. Feature Fusion Module (FFM)

The Feature Fusion Module[8] (FFM) improves upon the Bidirectional Feature Pyramid Network (BiFPN) by employing a CRC strategy to enhance information interaction between feature maps of different scales. The FFM module takes the feature maps X_2, X_3, X_4 output by three Feature Extraction Modules (FEM) as inputs. Using an upward CRC strategy, it fuses the feature map X_4', which has been processed through the FEM module, CBS, and upsampling after SPPF output, with X_3 to obtain X_3'. Similarly, this operation fuses X_3' with X_2 to get X_2', thus yielding the first half of the FFM output results: X_2', X_3', and X_4'. The principle of the first half is as follows in the following formulas:

$$X_4' = f_{up}^{2\uparrow}(CBS(X_4)) \tag{0.11}$$

$$X_3' = CRC(X_3, X_4') \tag{0.12}$$

$$X_2' = CRC(X_2, f_{up}^{2\uparrow}(CBS(CSP(X_3')))) \tag{0.13}$$

The latter half of the FFM utilizes downsampling operations on X_2, X_3, and X_4 from top to bottom. X_2' undergoes a convolution with a stride of 2 and is then processed with the CRC strategy, where it is fused with feature map X_3 and the processed X_3' to obtain an updated X_3''. Similarly, through a comparable operation, X_4'' is also obtained. Ultimately, the latter half of the FFM outputs the resulting X_2'', X_3'', and X_4''. The principle of the latter half of the FFM is shown as follows:

$$X_2'' = CSP(X_2') \tag{0.14}$$

ISSN: 2710-0588

$$X_3'' = CSP(CRC(X_3, CBS(CSP(X_3')), CBS(X_2'')))) \tag{0.15}$$

$$X_4'' = CSP(CRC(CBS(X_4), CBS(X_3'')) \tag{0.16}$$

The FFM first performs upsampling and downsampling operations on high-level and low-level feature maps to align them to the same spatial resolution. Then, it utilizes the CRC strategy to re-weight the feature maps based on channel information, thereby achieving effective fusion of feature maps at different scales. This design not only improves the quality of multi-scale feature fusion but also avoids a significant increase in computational complexity. It helps enhance the network's semantic representation capability for small targets, improving the accuracy and robustness of small target detection.

2.5. Spatial Context Awareness Module (SCAM)

The Spatial Context Awareness Module[8] (SCAM) is an efficient method for global context feature representation aimed at enhancing the detection capabilities of small targets in remote sensing images. SCAM captures global context information by integrating global average pooling and global maximum pooling, and utilizes this information to guide pixel learning of the relationships between space and channels.

$$Q_i^j = P_i^j + a_i^j \sum_{j=1}^{N_i} \left[\frac{\exp(\omega_{qk} P_i^j)}{\sum_{n=1}^{N_i} \exp(\omega_{qk} P_i^n)} \cdot \omega_v P_i^j \right] \tag{0.17}$$

$$a_i^j = \frac{\exp([avg(P_i); \max(P_i)] P_i^j)}{\sum_{n=1}^{N_i} \exp([avg(P_i); \max(P_i)] P_i^n])} \cdot \omega_v \tag{0.18}$$

The SCAM incorporates three streams to ensure its attention mechanism. In the first stream, the input feature map undergoes a softmax normalized fusion of Global Average Pooling $avg(P_i)$ and Global Maximum Pooling $\max(P_i)$, resulting in a_i^j. In the latter two streams, the feature maps are transformed using simplified 1×1 convolutions, referred to as ω_{qk} and ω_v, into QK and V respectively. Here, QK retains the HWC dimensions unchanged, while the number of channels C_V in V is adjusted to align with subsequent computations. After processing QK with softmax, it is matrix-multiplied with V. The inclusion of a_i^j, combined with a residual connection that adds the input feature P_i^j, results in the output Q_i^j.

The first branch of SCAM is used to refine contextual details, the second branch is for computations involving feature maps ω_v, and the third branch is dedicated to generating attention maps. By performing matrix multiplication and broadcast Hadamard product operations on the outputs of these branches, SCAM facilitates cross-channel and spatial contextual feature interactions. This mechanism suppresses irrelevant backgrounds and enhances the distinction between the target and background. Such a design not only improves the network's global correlation ability but also helps mitigate background confusion in small target detection, thereby enhancing the accuracy and robustness of detecting small targets.

ISSN: 2710-0588

3. Data Set Creation

3.1. Mixup Data Augmentation Method

The Lockwire object detection task demands high quality and a large quantity of training data. Lockwire has reflective characteristics, with surface features very similar to the texture features of its surrounding environment, which can easily lead to false detections. Given that annotating datasets requires professional knowledge in aviation mechanics and is labor-intensive, the currently available annotated datasets are relatively small in size.[9] Furthermore, due to safety and confidentiality considerations, these datasets are typically restricted within the collecting entity and cannot be shared. In the field of deep learning, small-scale datasets often fail to adequately train complex models, thereby affecting the model's generalization ability. Additionally, Lockwire images are affected by various lighting conditions, and this variation in illumination further increases the difficulty of object detection.

To address these issues, we introduce the mixup data augmentation method. As shown in the figure, images (a) and (b) are the original Lockwire images. After performing the mixup operation on these two images, image (c) is obtained. Mixup constructs new training samples through linear interpolation. This approach can expand the distribution of training data, which helps improve the model's generalization ability. Especially when dealing with small-scale datasets and under complex lighting conditions, mixup can effectively alleviate overfitting problems and enhance the model's adaptability to unseen samples.

Figure 3. Application of Mixup Data Augmentation on the Fuse Dataset

The core idea of mixup is to construct new training samples based on the linear combination of two samples and their labels. Suppose we have two training samples (x_i, y_i) and (x_j, y_j), where x represents the input data, and y represents the corresponding labels. The parameter λ is drawn from a Beta distribution, i.e., $\lambda \sim Beta(\alpha, \alpha)$, where α is a hyperparameter that controls the shape of the distribution. The λ values generated by the Beta distribution fall within the interval $(0,1)$.

According to the mixup method, the newly constructed sample $(\tilde{x}, \tilde{y})$ can be calculated using the following formulas:

$$\tilde{x} = \lambda x_i + (1-\lambda) x_j \tag{0.19}$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j \tag{0.20}$$

ISSN: 2710-0588

In this process, λ determines the weight of the original samples x_i and x_j and their corresponding labels y_i and y_j in the linear combination. Therefore, through this method, mixup not only enhances the diversity of the training data but also encourages the model to produce smoother decision boundaries between different classes. This helps reduce the overfitting phenomenon and may improve the model's performance on the test set.

3.2. Dataset Classification

The dataset classification strategy in this article is based on a Bayesian classification decision analysis of maintenance tasks distributed over the time dimension within a single day. Ignoring the impact of extreme weather on the distribution of daylight over the time dimension within a single day, this article roughly sets daylight and time as uniformly distributed, and discusses the influence of the distribution of maintenance tasks over the time dimension within a single day on classification decisions.[9]

This passage outlines an analytical approach for classifying datasets by considering how maintenance tasks are distributed throughout a single day, assuming a uniform distribution of daylight and time, to study the effects on classification decisions, barring the influence of extreme weather conditions.

First, the time periods are categorized into three states: morning, noon, and evening, with the distribution of tasks among these categories represented as: $y = \{y_1, y_2, y_3\}$. Here, it is assumed that after the intelligent upgrade of the entire machine maintenance system, the total time for testing a single machine is 1 hour. The morning period is from 8:00 to 12:00, the noon period is from 12:00 to 14:00, and the afternoon period is from 14:00 to 18:00. In other words:

$$P(y_1) = 0.4 \tag{0.21}$$

$$P(y_2) = 0.2 \tag{0.22}$$

$$P(y_3) = 0.4 \tag{0.23}$$

$$N_{train}(y_i) = N_{total} \cdot P(y_i) \tag{0.24}$$

$N_{train}(y_i)$ represents the number of samples in the training dataset corresponding to y_i, N_{total} represents the total number of samples in the training dataset, and $P(y_i)$ represents the probability values of the morning, noon, and evening categories. Such a prior probability distribution directly guides the data collection phase, with the data volume ratio for morning, noon, and evening being $2:1:2$. Taking the total number of samples in the training dataset N_{total} as 20,000, the numbers of samples in the training dataset for the three categories of morning, noon, and evening $N_{train}(y_i)$ are respectively 8,000, 4,000, and 8,000. The sample numbers are shown in the table below:

ISSN: 2710-0588

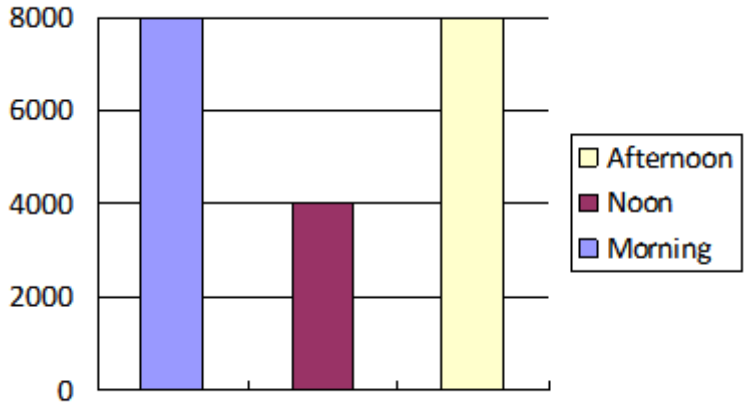

Figure 4. Table of Sample Counts

3.3. Data Preprocessing

Despite setting up review tasks and classification strategies in this article, considering the applicability and robustness of subsequent deep learning model object detection tasks under most circumstances, this article applies grayscale processing and brightness mean processing to the dataset images with different lighting and colors. [10]As shown in the figure below, from top to bottom, the first row represents the original images collected in the morning, noon, and evening from left to right, respectively. The second row shows the grayscale processing results for the morning, noon, and evening, while the third row presents the results after both grayscale and brightness mean processing for the morning, noon, and evening:

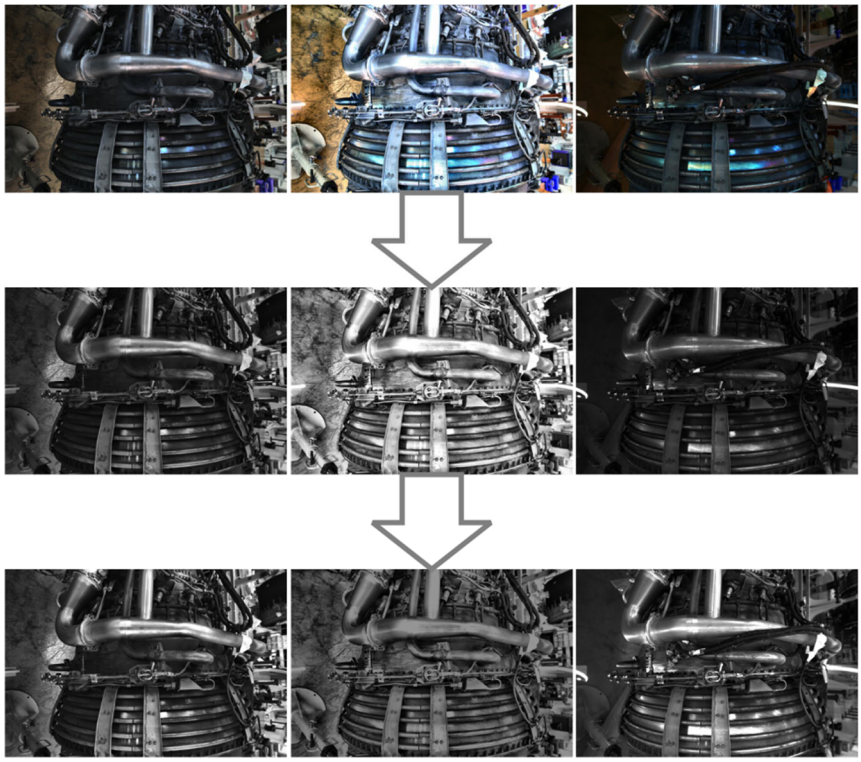

Figure 5. Effects After Grayscale Processing (Second Row) and Mean Brightness Normalization (Third Row)

ISSN: 2710-0588

4. Experiments

4.1. Experimental Metrics

To comprehensively and objectively measure the performance of the model, this article uses a series of evaluation metrics widely recognized in the field of deep learning object detection:

Precision P is the proportion of samples that are actually positive among those predicted as positive by the model. The formula for calculating precision is:

$$P = \frac{TP}{TP + FP} \tag{0.25}$$

Where TP represents the number of samples correctly predicted as positive by the model, and FP represents the number of samples incorrectly predicted as positive by the model. The higher the precision, the greater the proportion of actual positive samples among those predicted as positive by the model, which means the lower the misclassification rate of the model.

Compared to precision P, recall R focuses more on the proportion of all actual positive samples that are correctly predicted by the model. The formula for calculating recall is:

$$R = \frac{TP}{TP + FN} \tag{0.26}$$

Where FN represents the number of actual positive samples that were incorrectly predicted as negative by the model. The higher the recall, the more actual positive samples the model can identify, meaning fewer positive samples are missed. This metric is crucial for application scenarios where comprehensive coverage of positive samples is a priority, such as in information retrieval, criminal investigation, etc.[11]

The confusion matrix, also known as an error matrix, is a standard format for indicating accuracy evaluation. It clearly presents the results of model classification through a matrix format, aiding in understanding the model's performance across different categories. For the experiments in this article, since there are two detection targets, and to make the detection results clearer, a new "background" and "clamp" category should be added to represent the background and contrast parts like hose clamps.

For a single detection target in this article, the confusion matrix typically includes four elements to describe the effectiveness of the target detection: TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative), which respectively denote the quantities of predictions that are actually positive when predicted positive, actually negative when predicted positive, actually negative when predicted negative, and actually positive when predicted negative. For precision P and recall R, the values of the elements in the confusion matrix can be interconverted.

Each column of the confusion matrix represents the predicted class, and the total number of each column indicates the number of data instances predicted to belong to that class. Each row represents the true class to which the data actually belongs, and the total number of instances in each row indicates the number of data instances in that particular class. Each cell in the matrix shows the number of samples where the actual class is the row class and the predicted class is the column class.

4.2. Experimental Results

After testing four deep learning models—YOLOv8, MobileNetv3, MobileFormer, and LwDetFormer—on the test set, we obtained the model test confusion matrices based on the test

ISSN: 2710-0588

set. The test set used in the experiment contains 2,000 images of aircraft engine parts, with annotation instances for hose clamps and fuses being 200 and 1,800, respectively. Below are the confusion matrices for the four models:

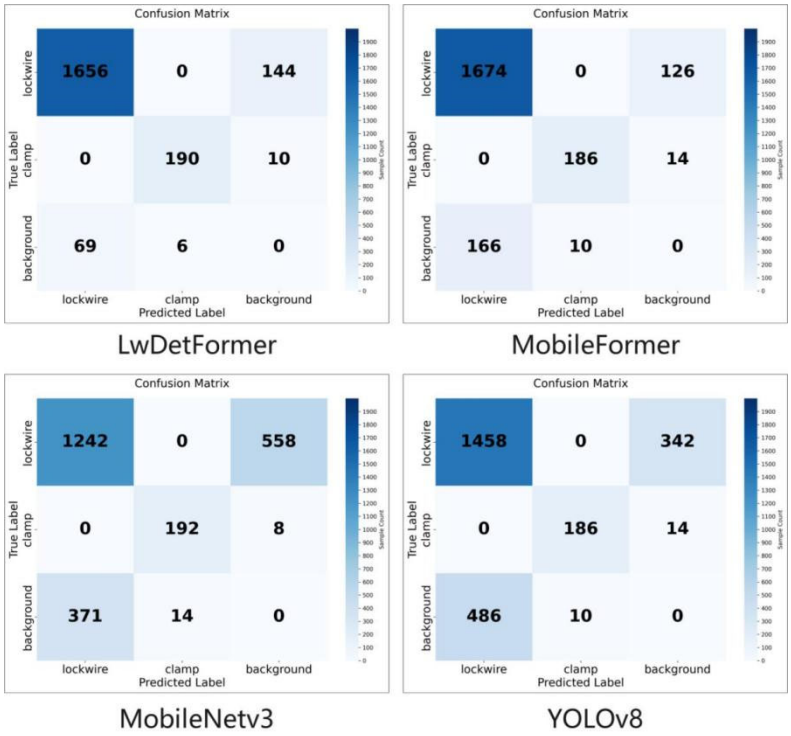

Figure 6. Confusion Matrix of Various Models on the Test Set

From the confusion matrices obtained by testing on the aforementioned test set, we can derive the TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative) for each of the four models. Based on these values, we can calculate the precision P and recall R for each of the four models. For detecting lockwire (fuse), the detection performance of LwDetFormer has improved:

Table 1. Metric Calculation Results of Various Models on the Fuse Dataset

	YOLOv8	MobileNetv3	MobileFormer	LwDetFormer
TP	1458	1242	1674	1656
FP	486	371	166	69
TN	210	214	210	206
FN	342	558	126	144
P	75.0%	77.0%	90.98%	95.9%
R	81.0%	69.0%	93.0%	92.0%

By comparing the performance of the YOLOv8, MobileNetv3, MobileFormer, and LwDetFormer models on the test set, it is evident that LwDetFormer shows significant advantages in detecting lockwire (fuse). LwDetFormer has a precision rate of 95.9%, which is notably higher than the other three models (YOLOv8 at 75.0%, MobileNetv3 at 77.0%, and MobileFormer at 90.98%). This indicates that among samples predicted as positive by LwDetFormer, the proportion of actual positives is the highest, meaning it has the lowest misclassification rate. With a recall rate of 92.0%, LwDetFormer is only slightly lower than MobileFormer's 93.0%, but significantly

ISSN: 2710-0588

higher than YOLOv8's 81.0% and MobileNetv3's 69.0%. This suggests that LwDetFormer can identify more actual positive samples with fewer missed detections. Looking at the values of TP (True Positive), FP (False Positive), TN (True Negative), and FN (False Negative) from the confusion matrix, LwDetFormer not only excels in reducing false positives (FP is only 69) but also performs well in increasing the recognition of true positives (TP is 1656). Additionally, LwDetFormer has the least number of false negatives (FN) among the four models, with only 144, indicating fewer instances of missed detections.

LwDetFormer effectively enhances the detection accuracy and recall for small targets like lockwire, which have complex backgrounds and reflective characteristics, by combining MobileNet's local feature extraction capabilities with Transformer's global feature modeling abilities. Especially when dealing with challenges such as smaller dataset sizes and variations in lighting conditions, LwDetFormer demonstrates superior performance, providing strong support for the target detection of aircraft engine fuse parts lockwire. These results validate the effectiveness and advancement of LwDetFormer as a novel hybrid CNN-Transformer architecture. Future work could further explore how to optimize this model to adapt to more diversified application scenarios and attempt training on larger datasets to test its generalization ability.

5. Summary and Outlook

In response to the challenges of Lockwire target detection, this article proposes a hybrid CNN-Transformer deep learning model named LwDetFormer. Aimed at achieving high-precision detection of Lockwire under conditions of complex backgrounds, reflective characteristics, and small-scale datasets, the model combines the local feature extraction capabilities of MobileNet with the global feature modeling capabilities of Vision Transformer, thereby effectively improving the recognition accuracy for Lockwire objects that have similar surface features and backgrounds. Experimental results show that LwDetFormer outperforms other advanced models such as YOLOv8, MobileNetv3, and MobileFormer in terms of precision and recall, making significant progress particularly in reducing false positives and false negatives.

Despite its strong performance in Lockwire target detection, there are several directions worth further exploration for LwDetFormer:

Stabilizing Model Parameters: Future work could involve continuing to optimize the LwDetFormer model structure, such as adjusting or improving parameter settings within the MobileFormer blocks to enhance model efficiency and performance. Additionally, considering applying this model to other similar small target detection tasks to verify its versatility and adaptability.

Dataset Augmentation: Although mixup data augmentation methods help mitigate the challenges posed by small datasets, a richer annotated dataset remains key to enhancing model performance.

Application Expansion: Beyond the detection of aircraft engine fuse parts, LwDetFormer may also be applicable to the detection of small and complex objects in other fields.

Real-time Processing Capability: With the growth of practical application demands, enhancing the real-time processing speed and efficiency of LwDetFormer becomes an important topic. This includes but is not limited to algorithm-level optimizations and the application of hardware acceleration technologies to better meet the industry's need for rapid response.

Through continuous efforts and improvements in these areas, it is believed that LwDetFormer and its subsequent versions will play an important role in more practical applications, bringing higher safety and efficiency to related industries.

ISSN: 2710-0588

References

[1] Zhang Fengfei, Sun Junhua. An Instance Segmentation Method of Aviation Engine Fuses Based on Improved Mask R-CNN[J/OL]. Measurement & Control Technology, 1-9 [2025-03-14]. http://kns.cnki.net/kcms/detail/11.5347.TB.20250122.1826.002.html.

[2] Howard A, Sandler M, Chu G, et al. Searching for MobilenetV3[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2019: 1314-1324.

[3] Yuan L, Chen Y, Wang T, et al. Tokens-to-token ViT: Training Vision Transformers from Scratch on ImageNet [C]//Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021: 558-567.

[4] Chen Y, Dai X, Chen D, et al. Mobile-former: Bridging mobilenet and transformer[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2022: 5270-5279.

[5] Zhao X, Wang L, Zhang Y, et al. A review of convolutional neural networks in computer vision[J]. Artificial Intelligence Review, 2024, 57(4): 99.

[6] Mehta S, Rastegari M. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer[J]. arXiv preprint arXiv:2110.02178, 2021.

[7] Guo D, Zhang C, Yang G, et al. Siamese-RCNet: Defect Detection Model for Complex Textured Surfaces with Few Annotations[J]. Electronics, 2024, 13(24): 4873.

[8] Zhang Y, Ye M, Zhu G, et al. FFCA-YOLO for small object detection in remote sensing images[J]. IEEE Transactions on Geoscience and Remote Sensing, 2024, 62: 1-15.

[9] Liu Y, Li H, Hu C, et al. Learning to aggregate multi-scale context for instance segmentation in remote sensing images[J]. IEEE Transactions on Neural Networks and Learning Systems, 2024, 36(1): 595-609.

[10] Liu Jinqiu. Fault Analysis and Repair of a Small Component in an Aero Engine [J]. Value Engineering, 2018, 37(2): 100-102.

[11] Feng Maoxing. An Object Detection Algorithm for Aero-Engine Fasteners Based on Improved YOLOv5s [J]. Science and Technology & Innovation, 2024, (24): 20-22+27. DOI: 10.15913/j.cnki.kjycx.2024.24.006.