PAPER

FA-YOLO: A High-Precision and Efficient Method for Fabric Defect Detection in Textile Industry

Kai YU[†], Wentao LYU[†a)], Xuyi YU[†], Qing GUO[††], Weiqiang XU[†], *Nonmembers, and* Lu ZHANG[†], *Member*

SUMMARY The automatic defect detection for fabric images is an essential mission in textile industry. However, there are some inherent difficulties in the detection of fabric images, such as complexity of the background and the highly uneven scales of defects. Moreover, the trade-off between accuracy and speed should be considered in real applications. To address these problems, we propose a novel model based on YOLOv4 to detect defects in fabric images, called Feature Augmentation YOLO (FA-YOLO). In terms of network structure, FA-YOLO adds an additional detection head to improve the detection ability of small defects and builds a powerful Neck structure to enhance feature fusion. First, to reduce information loss during feature fusion, we perform the residual feature augmentation (RFA) on the features after dimensionality reduction by using 1×1 convolution. Afterward, the attention module (SimAM) is embedded into the locations with rich features to improve the adaptation ability to complex backgrounds. Adaptive spatial feature fusion (ASFF) is also applied to output of the Neck to filter inconsistencies across layers. Finally, the cross-stage partial (CSP) structure is introduced for optimization. Experimental results based on three real industrial datasets, including Tianchi fabric dataset (72.5% mAP), ZJU-Leaper fabric dataset (0.714 of average F1-score) and NEU-DET steel dataset (77.2% mAP), demonstrate the proposed FA-YOLO achieves competitive results compared to other state-of-the-art (SoTA) methods.

key words: *fabric defect detection, feature augmentation, attention mechanism, cross-stage partial, YOLOv4*

1. Introduction

Fabric defect detection is very important in the quality control of the fabric industry, as it can help manufacturers detect production problems early and improve product quality. By using efficient detection methods, defects in products can be quickly and accurately detected, avoiding subjective manual inspection. This saves a lot of time and resources, and improves the production efficiency. In the past, the defect detection in the fabric images was performed manually. Because of the diverse types of fabric defects and the complexity of the background and texture, this work heavily depends on the experience of inspectors, which is hard to meet the real-time requirements. Traditional methods [1], [2] based on computer vision may improve the detection effectiveness, but they have inherent difficulties, such as low accuracy, slow speed, cumbersome process, and poor generalization performance. In contrast, the detection methods based on deep learning has great potential to tackle with these issues.

The computer vision method based on deep learning adopts an end-to-end (E2E) solution, and automatically extracts the most descriptive features of the target category through a neural network, with higher accuracy and less manual intervention. Many works have been widely applied in the field of surface defect detection [3], [4], remote sensing image detection [5], medical image processing [6], and autonomous driving [7], just name a few. These methods are usually divided into one-stage object detection method and two-stage object detection method. The one-stage object detection method represented RetinaNet [8], YOLO series [9], which are characterized by direct regression to obtain detection results. Compared to the two-stage methods, one-stage methods have less parameters and higher inference speed. Among them, YOLOv4 [10] is widely used in various fields due to its excellent performance. For example, in the field of medical imaging, F. Abdurahman et al. [6] added a shallow large-scale detection layer to detect small objects in blood microscopic images. For the detection of aircraft objects in remote sensing images, Y. Yang et al. [5] used the depth-separable convolution to replace the normal convolution in the backbone network, at the cost of a certain accuracy loss in exchange for a significant reduction in the amount of parameters. In [11], D. Wang et al. also added a large-scale detection layer for the pest detection. By adding more residual units in the low-level layers, more feature information for the precise localization of small pests were extracted accurately. These YOLOv4-based detection methods can adapt well to their respective application scenarios.

However, for the fabric defect detection scene, the existence of complex fabric backgrounds and diverse defect categories can lead to degradation of the detector. Peng et al. [12] proposed a priori anchor convolutional neural network (PRAN-Net) algorithm to detect all tiny and some specific fabric defects. Z. Zhao et al. [13] proposed a cascaded Faster R-CNN for defect detection in fabric images. The different types of defective images were first pre-classified, and then Faster R-CNN is used for the specific localization of defects. For the task of fabric defect detection, J. Wu et al. [14] proposed a structure with dilated convolutions similar to the Inception structure, and used it to replace the traditional convolutions in Faster R-CNN to achieve the diversity of feature extraction. For the defect detection of fabric images, M. An et al. [15] replaced the backbone network VGG-16 with ResNet-101 in Faster R-CNN, and added the feature pyramid network (FPN) structure to perform multi-

Manuscript received March 14, 2023.
Manuscript revised July 17, 2023.
Manuscript publicized September 4, 2023.

[†] The authors are with Key Laboratory of Intelligent Textile and Flexible Interconnection of Zhejiang Province, Zhejiang Sci-Tech University, Hangzhou 310018, China.

[††] The author is with the Zhejiang Technical Innovation Service Center, Hangzhou 310007, China.

a) E-mail: alvinlwt@zstu.edu.cn (Corresponding author)

DOI: 10.1587/transfun.2023EAP1030

scale prediction, which greatly improved the detection effect. Zheng et al. [16] added SE attention mechanism in YOLO to improve the accuracy of fabric defect detection. Luo et al. [17] used deep separable convolution in YOLO to further improve the detection rate of the network.

The existing YOLOv4-based detection methods can detect the selected objects very well. However, most of these methods are oriented to simple objects, and perform poorly in complex fabric images. After analysis, we believe that an excellent fabric defect detection method should at least consider the following issues.

- Small defects occupy a small proportion on large-scale feature maps and are easily ignored. Thus, too much information should not be lost during network propagation.
- Due to the variety of backgrounds in fabric images and unclear distinctions between categories, the model needs to be more sensitive to global context information and be able to focus on important defect information instead of background.
- The semantic misalignment between feature maps at different levels may lead to information loss or redundancy, thereby affecting the accuracy and robustness of object detection.

Based on this, we present a novel defect detection method for fabric images in this paper, called Feature Augmentation YOLO (FA-YOLO). FA-YOLO is sensitive to global context information and has strong detection capabilities at all scales. Note that our model focuses on combining some effective techniques and components with YOLOv4, aiming to achieve better fabric defect detection performance with less impact on efficiency. Some existing structures are proven to be independently effective under a certain architecture (such as MemFRCN [18] and MobileNet [19]). However, they may perform poorly or even have negative effects when the basic architectures are replaced or other structures are co-embedded. Thus, after conducting independent and joint experiments on these structures based on YOLOv4, we select the ones that can achieve consistent improvement, that is, the improvements brought by these structures are orthogonal. Many of these structures cannot be directly applied to YOLOv4, so some modifications are required. Compared with YOLOv4, it achieves higher detection accuracy and greatly reduces the amount of model parameters. The main contributions of this paper are as follows:

- We introduce Residual Feature Augmentation (RFA) in YOLOv4 to perform multi-scale information enhancement on the low-level features after channel reduction, so as to compensate for information loss of small defects.
- By introducing a parameter-free attention mechanism at an appropriate position, the network can filter out irrelevant background information and pay more attention to defect features.
- Adaptive Spatial Feature Fusion (ASFF) alleviates the variability between features at each scale through dynamic weighted fusion between layers. We redesign the basic modules of the feature fusion network with CSP structure to reduce the complexity of the network.
- Multiple experiments based on three public industrial datasets (Tianchi, ZJU-Leaper and NEU-DET) are conducted. Results show that the proposed method effectively improves the performance in fabric defect detection. It outperforms other state-of-the-art (SoTA) detection methods on multiple metrics while maintaining real-time detection performance.

The rest of the paper is organized as follows. In Sect. 2, the proposed method is described in detail. Section 3 provides multiple sets of experimental results and performance analysis. Section 4 concludes this paper.

2. Framework of Feature Augumention YOLOv4 (FA-YOLO)

Our proposed FA-YOLO includes Backbone, Neck and Head. The overall architecture of FA-YOLO is shown in Fig. 1. The parts different from YOLOv4 are marked in red.

2.1 Backbone

The feature extraction network of YOLOv4 is CSPDarknet53. Compared with ResNet50, CSPDarknet53 has slightly lower classification accuracy, but higher accuracy in object detection. Note that the architecture of CSPDarknet53 conforms to all optimal architectural features obtained by the network architecture search (NAS) technique [10], and direct embedding of other modules will result in performance degradation. As the network level becomes deeper, the receptive field of each unit on the feature map becomes larger and the semantic information becomes richer. However, due to reduction of the resolution, the position information will also be lost, which is adverse for detecting small objects. On the contrary, the low-level features are rich in geometric details due to their smaller receptive field and larger image resolution [20]. Thus, in order to specifically detect a large number of small-scale fabric defects in the dataset, we add an additional large-scale feature map as output. This means that we select output results of the last four downsamplings layers C2, C3, C4, and C5 in the backbone as the input for the subsequent Neck fusion.

2.2 Neck

Considering the characteristics of fabric defects, we introduce the following structures and methods to improve the detection effect. Among them, the SimAM module, RFA module, and ASFF module are more helpful for fabric defect detection, while the other structures will slightly improve the

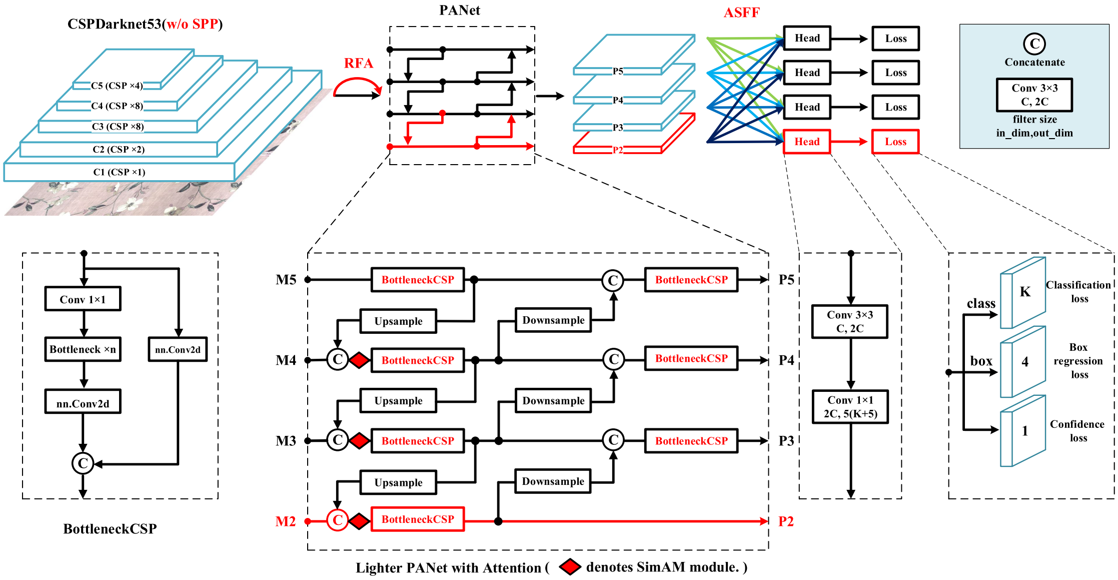


Fig. 1 Illustration of FA-YOLO network architecture.

effectiveness of fabric detection while improving the general detection effect.

2.2.1 SimAM Module

Interference from complex backgrounds can cause textural features of fabric targets to be overwhelmed. The attention mechanism can dynamically assign weights to features, and thereby emphasizing important features and suppressing noisy features. In order to effectively improve expressive ability of the network with a small computational cost, we introduce the parameter-free attention module SimAM to filter the interference from complex backgrounds.

The SimAM module finds the importance of each neuron through the energy function, which can infer three-dimensional weights from the current neurons, so that the network can learn more discriminative neurons to highlight important features. In addition, since the 3D attention mechanism can highlight neurons with spatial inhibitory effects, the problem of feature misalignment caused by direct superposition of features of different scales is alleviated to a certain extent [21]. The following equation is the energy function defined in the SimAM module to measure the linear differentiability between neurons,

$$e_t(w_t, b_t, y, x_i) = \frac{1}{M-1}\sum_{i=1}^{M-1}(-1-(w_t x_i + b_t))^2 + (1-(w_t t + b_t))^2 + \lambda w_t^2, \qquad (1)$$

where t and x_i represent the target neuron and other neurons of a single channel in the input feature X, and i is the index in the spatial dimension. M is the number of neurons on the channel, and w_t and b_t are the weights and biases of the linear transformation, respectively.

Moreover, this equation has a closed-form solution by assuming that all pixels on a single channel follow the same distribution, resulting in following expression for the minimum energy.

$$e_t^* = \frac{4(\hat{\sigma}^2 + \lambda)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\lambda}, \qquad (2)$$

where $\hat{u} = \frac{1}{M}\sum_{i=1}^{M} x_i$ and $\hat{\sigma}^2 = \frac{1}{M}\sum_{i=1} M(x_i - \hat{\mu})^2$, the importance of each neuron can be obtained by $\frac{1}{e_t^*}$.

The "diamonds" symbols in Fig. 1 represent where we embed the attention module SimAM to the Neck.

2.2.2 Residual Feature Augmentation

The features extracted by the backbone network are all dimensionality-reduced before feature fusion, which will inevitably result in information loss. In fabric defects, there are many small and oddly shaped defects, which will inevitably cause the loss of necessary boundary features. In order to alleviate the loss of boundary features of defects, we perform RFA on the dimensionality-reduced features. First, multi-branch adaptive pooling is used to obtain feature maps of different scales. The adaptive spatial fusion (ASF) structure is then used for weighted fusion. The weights are generated in the form of $Convolution + Sigmoid$, similar to the attention mechanism. Finally, the result M6 containing multi-scale information is fused into the dimension-reduced feature M5. The RFA and ASF structure are shown in Fig. 2. Different from the operation of enhancing high-level semantic information in AugFPN, we introduce RFA to reduce the information loss of low-level features for small fabric defects.

2.2.3 Large-Scale Detection Layer

In the original YOLOv4, the sizes of the feature map used for prediction are $m/8 \times m/8$, $m/16 \times m/16$, and $m/32 \times m/32$

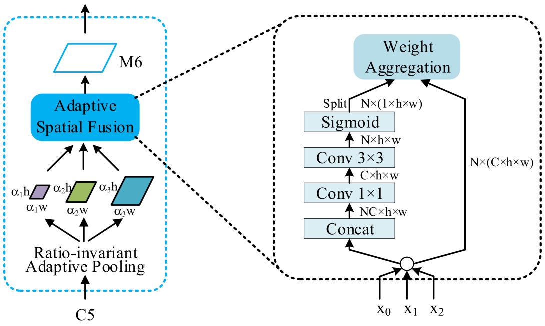


Fig. 2 Illustration of the RFA process. The right subpart shows the detail of the ASF structure.

when the input scale is set as $m \times n$. These feature maps are not conducive to the detection of small objects due to their low resolution. Therefore, we additionally output a feature map of $m/4 \times m/4$ in Backbone network, and fuse the feature maps of these four scales in Neck. Finally, the fused feature maps of $m/4 \times m/4$, $m/8 \times m/8$, $m/16 \times m/16$, and $m/32 \times m/32$ scales are used for multi-scale prediction.

2.2.4 Adaptive Spatial Feature Fusion

Due to the inconsistency of semantic information between layers, it may cause conflicts by directly adding or concatenating fusion of feature maps, so that the features of each scale cannot be fully utilized. Therefore, ASFF is performed on the four output feature maps P2, P3, P4, and P5 before prediction. ASFF enables the network to directly learn how to spatially filter features at other levels, and thereby retaining only useful information for combination. For features at a certain level, features at other levels are first resized to the same resolution and simply integrated. They are then trained to find the best blend. At each spatial location, features from different levels are adaptively fused together. It dynamically learns the weights for feature fusion at different scales, and introduces almost negligible inference time [22]. Furthermore, cross-layer information exchange is better facilitated through direct feature fusion between layers. In other words, the high-level layers can convey sufficient semantic information to the low-level layers more easily, and the low-level layers can also provide positioning information that high-level layers neglect. The calculation formula for executing ASFF is as follows:

$$\boldsymbol{Y}_{ij}^{m} = \alpha_{ij}^{m}\boldsymbol{P}_{ij}^{2\to m} + \beta_{ij}^{m}\boldsymbol{P}_{ij}^{3\to m} + \gamma_{ij}^{m}\boldsymbol{P}_{ij}^{4\to m} + \lambda_{ij}^{m}\boldsymbol{P}_{ij}^{5\to m}, \tag{3}$$

where $\boldsymbol{Y}_{ij}^{m}$ represents the feature vector at position $(i,\ j)$ in the output feature map $\boldsymbol{Y}^{m}$. $\boldsymbol{P}_{ij}^{n\to m}$ represents the feature vector at position (i, j) in the feature map adjusted from level $k(k = 2,\ 3,\ 4,\ 5)$ to level m. α_{ij}^{m}, β_{ij}^{m}, γ_{ij}^{m}, λ_{ij}^{m} respectively represent the weighted parameters for the fusion of feature maps from four different levels, which are obtained by 1×1 convolution of input feature maps from different levels and then obtained through the softmax function. It follows the following constraints:

$$\alpha_{ij}^{m} + \beta_{ij}^{m} + \gamma_{ij}^{m} \ + \lambda_{ij}^{m} = 1. \tag{4}$$

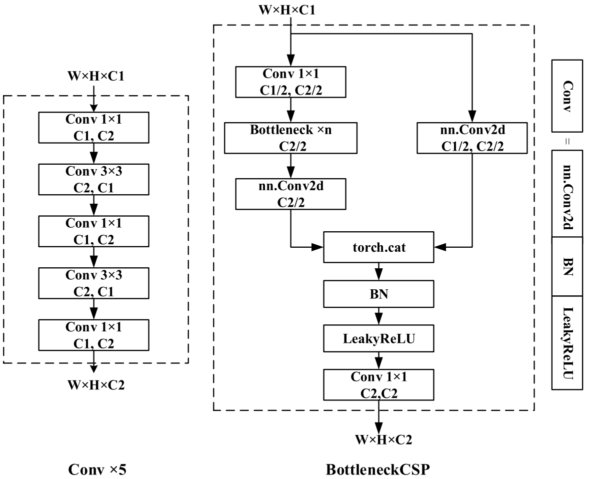


Fig. 3 Detailed structure comparison of Conv×5 convolutional block and BottleneckCSP module.

2.2.5 Activation Function

Compared with other activation functions, Mish performs better in object detection tasks. Considering its relatively high complexity, we only upgrade the activation function in the BottleneckCSP module from the LeakyReLU to Mish.

2.2.6 Structural Optimization

Since CSPDarknet53 is relatively lightweight, the computationally intensive part of the YOLOv4 network is mainly in Neck. Thus, in order to reduce complexity of the network and the smallest impact on the accuracy of the network, we first use the CSP structure to optimize convolutional blocks in Neck. Through the clever design of the CSP structure, the learning ability of the convolutional block can be enhanced, while reducing the amount of network parameters and the burden of inference. We use the number of channels on each branch of the Bottleneck module, thereby greatly reducing the network parameters. Moreover, we find that the accuracy of the model decrease after replacing the convolutional block before SPP with a BottleneckCSP module, and thus we retain it at first. Consider the number of channels in the deep layer of the network is huge, we remove the SPP module and the previous Conv 3×3 convolutional block together, which greatly reduces the amount of network parameters and accelerates the inference speed. By performing this, it can be found that the accuracy of the network is even slightly improved. Figure 3 shows a comparison of the two structures.

2.3 Head

Due to the diversity of fabric defects in dataset, the original

Table 1 The results of the integration of components based on the YOLOv4 baseline. ("✓" means that the structure is contained, and "-" means that the structure is not contained). "Partial CSPized" means that part of the convolutional block in the PAN is replaced with a CSP structure).

Group	Partial CSPized	Detection Head	ASFF	RFA	SimAM	w/o SPP	NMS-finetuned	mAP/%	Param size/MB
G1	-	-	-	-	-	-	-	65.2	244.2
G2	✓	-	-	-	-	-	-	65.4	171.6
G3	✓	✓	-	-	-	-	-	67.0	172.5
G4	✓	✓	✓	-	-	-	-	69.3	195.7
G5	✓	✓	✓	✓	-	-	-	69.9	198.6
G6	✓	✓	✓	✓	✓	-	-	71.0	198.6
G7	✓	✓	✓	✓	✓	✓	-	71.2	174.6
G8	✓	✓	✓	✓	✓	✓	✓	72.5	174.6

three shapes of prior anchor boxes for the Pascal VOC dataset do not cover most cases well. Therefore, we increase the number of prior anchor boxes on each unit to 5 and use the k-means++ algorithm to generate them. Corresponding to the design of the previous Neck, we add a large-scale detection head at the end for better detection of small objects.

3. Results

3.1 Experimental Setup

We use PyTorch to implement the networks and experiments with an NVIDIA GeForce RTX 2070 GPU. FA-YOLO starts to train with an initial learning rate of 1e-2, and the strategy is cosine learning rate decay. SGD is selected as the optimizer, and the values of momentum and weight decay are set to 0.928 and 0.0005, respectively. Three metrics, including average precision (mAP), frames per second (FPS), and model storage size (Model size, MB), are used to evaluate the performance of each detector. In the experiments, the ratios of training, validation, and test sets are, respectively, selected as 0.75, 0.1, and 0.15.

3.2 Introduction of Real Industrial Datasets

The effectiveness of FA-YOLO is evaluated based on three real industrial defect datasets, including the Aliyun Tianchi fabric dataset [23], ZJU-Leaper defect dataset [24], and surface defect dataset of steel strip (NEU-DET) [25].

1) Tianchi fabric dataset: this dataset contains 4371 images with defects and 4371 normal images, including 15 types of defects. Due to the high similarity between certain defect categories in the Tianchi dataset, in our experiment, we selected 9 representative defect categories with a larger sample size as our benchmark dataset, including 'sewing', 'sewing_print', 'scrimp', 'bug', 'flaw', 'color_shade', 'miss_ print', 'hole', and 'fold'. The original size of each image is 4096×1696 in pixel. Each image is divided into two sub-images of 2048×1696 in pixel, and the images without defects are removed. Finally, 4972 images are collected as the new dataset. Among them, 3592 pieces are randomly selected for training, 634 pieces are used for validation, and 746 pieces are used for test.

2) ZJU-Leaper dataset: this dataset is a fabric defect dataset released by Zhejiang University, which contains 98777 high-quality images, including 27650 defect images. It is divided into five main texture groups. Each group contains 3∼5 similar patterns. In ZJU-Leaper dataset, simple patterns (such as solid and striped patterns) are integrated into Group 1. Group 2 consists of patterns with small repeating patterns (or primitives), such as grids and dots, which display a clear visual arrangement. Group 3 and Group 4 are respectively composed of checkered and floral fabrics, with complex arrangements or patterns. Finally, four types of grey fabrics (neither bleached nor dyed) are collected from the factories that formed Group 5.

3) NEU-DET dataset: this dataset contains six kinds of steel surface defects, including rolled-in scale, patches, crazing, pitted surface, inclusion and scratches. There are 1800 grayscale images in total, with 300 samples for each category of defects.

3.3 Ablation Experiment

In order to further analyze the performance of the improved YOLOv4 model, we decompose model into 8 groups (G1, G2, $\cdots$, G8), each of which is improved on the basis of the previous group. The experimental results are shown in Table 1.

As shown in Table 1, the consistent improvements can be obtained by integrating each component into the baseline. G1 is the original YOLOv4, used as the baseline. The results of G2 show that by introducing the CSP structure in Neck, the model size can be reduced to 171.6 MB, a reduction of 72.6 MB, and the accuracy can also be improved. This shows that the partial replacement of the CSP structure is very beneficial for detection. Note that replacing all convolutional blocks in Neck with CSP structure will result in a slight drop in accuracy, which is inconsistent with our goal. For the generation of the aspect ratios of the anchor boxes, we compare the effects of two clustering algorithms, k-means and k-means++. The number of cluster centers k is used as a variable, and the average intersection over union (IoU) is selected as an indicator. The comparison results are shown in Fig. 4. It can be seen that the performance of k-means++ is always better than k-means. Thus, the k-

Table 2 Class-wise performance comparison (mAP) between FA-YOLO and other SoTA methods based on Tianchi fabric dataset.

Methods	sewing	sewing_print	bug	hole	scrimp	flaw	miss_print	color_shade	fold	mAP/%
Faster R-CNN [26]	73.3	77.0	86.8	50.8	41.7	58.3	56.0	66.1	87.9	66.4
Cascade R-CNN [27]	84.3	78.9	83.3	48.1	35.5	60.6	56.9	74.9	87.3	67.8
Sparse R-CNN [28]	84.8	**86.3**	69.8	**57.9**	32.9	53.3	50.6	69.4	85.5	65.6
YOLOv3 [29]	89.8	71.2	72.0	43.2	45.1	54.5	14.7	33.4	82.1	56.2
YOLOv4 [10]	89.7	81.6	75.9	48.4	**54.0**	65.7	31.7	52.2	87.8	65.2
YOLOX [30]	**90.5**	79.5	85.8	41.3	47.7	55.0	32.1	64.2	79.1	63.9
YOLOv4-CSP [31]	89.5	77.7	75.5	45.2	50.9	64.1	26.9	43.0	86.3	62.1
QueryDet [32]	86.6	81.9	87.4	47.8	45.4	57.0	49.2	67.9	**90.0**	68.1
Proposed	80.8	74.6	**92.1**	54.7	42.6	**69.7**	**69.8**	**83.9**	83.7	**72.5**

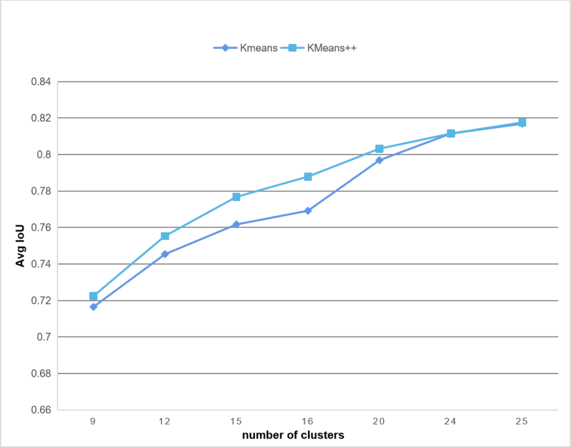


Fig. 4 Comparison between k-means++ and k-means algorithms. The vertical axis is the average IoU, and the horizontal axis is the number of clustering centers.

Table 3 Performance comparison between our proposed FA-YOLO and other SoTA methods based on Tianchi fabric dataset.

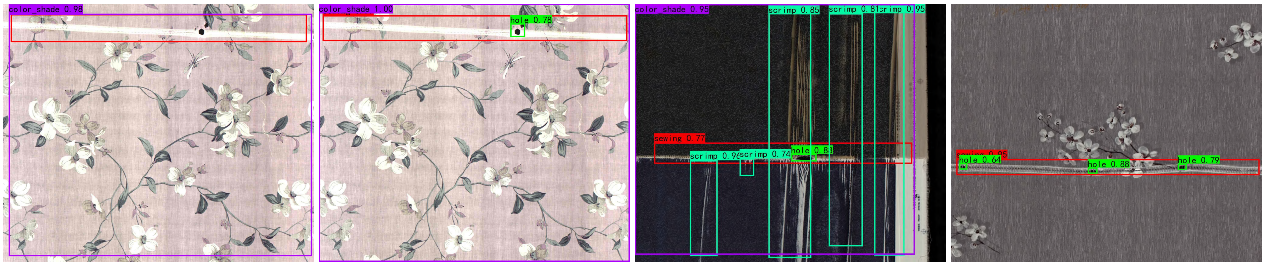
Methods	mAP/%	Model size(MB)	FPS
Faster R-CNN [26] (ResNet-50+FPN)	66.4	315.4	15
Cascade R-CNN [27] (ResNet-50+FPN)	67.8	552.9	10
Sparse R-CNN [28] (ResNet-50+FPN)	65.6	-	12
QueryDet [32]	68.1	315.5	16
YOLOv3 [29]	56.2	235.2	36
YOLOv4 [10]	65.2	244.2	48
YOLOv4-CSP [31]	62.1	200.4	51
YOLOX [30]	63.9	**71.9**	**69**
Proposed	**72.5**	174.6	40

means++ algorithm is selected to generate the anchor boxes in our method. From the experimental results of G3, it can be seen that using the k-means++ algorithm and adding detection head can increase mAP by 1.6%. Note that adding the detection head in the low-level layer will not have a great impact on the parameter due to the small number of channels. In G4, we perform adaptive spatial feature fusion on the output of PAN to enhance direct information exchange between various layers. In this way, a mAP improvement of 2.3% is achieved at the cost of an additional 23.2 MB of parameters. After a trade-off, we also introduce it into the network. In G5, it has achieved a 0.6% mAP improvement over G4, introducing only 2.9 MB of additional parameter cost. The reason is that we have compensated for information loss caused by dimensionality reduction of 1×1 convolutional. In the follow-up G6, by introducing the attention mechanism at the locations with rich information, an improvement of 1.1% on mAP is achieved without introducing additional parameters. It shows that embedding the attention mechanism in the right position does bring some benefits, as it can filter out interference from complex backgrounds to a certain extent. In G7, we use SPP optimization to improve our model by 0.2% on mAP, and reduce the number of parameters by 24.0 MB. Finally, by adjusting the threshold of non-maximum suppression, we achieve the best accuracy of 72.5% on mAP. Through the above steps, we have steadily improved the performance of YOLOv4, without any other tricks. The obtained FA-YOLO improves the mAP value by 7.3% compared to the baseline, and maintains original real-time performance under the premise of reducing the amount of model parameters by 28.5%.

3.4 Performance Comparison with Other SoTA Methods

Several SoTA models, including Faster R-CNN, Cascade R-CNN, Sparse R-CNN, QueryDet, YOLOX and Scaled-YOLOv4, are also performed for comparison. Three parameters, including mean average precision (mAP), model size, and frame per second (FPS), are used to evaluate the performance of various methods.

The first set of experiments is from Tianchi dataset. The results are shown in Table 2 and Table 3. It can be seen from Table 2 that our FA-YOLO has achieved the best mAP values, and the best performance in ‘bug’, ‘flaw’, ‘miss_print’ and ‘color_shade’ defect categories. Compared with three one-stage methods, YOLOv3, YOLOv4, and YOLOv4-CSP, our method improves significantly in accuracy, surpassing their mAP values of 16.3%, 7.3%, and 10.4%, respectively. YOLOX performs best in the ‘sewing’ defect category, but its mAP is 8.6% lower than our proposed model. QueryDet performs best in the ‘fold’ defect category, but its mAP is 4.4% lower than our proposed model. Furthermore, for some



(a) results of YOLOv4 model (b) results of FA-YOLO model (c) dense defects of FA-YOLO model (d) overlapping defects of FA-YOLO model

Fig. 5 Detection results of our FA-YOLO for complex fabric images.

two-stage methods, such as Faster R-CNN, Cascade R-CNN, and Sparse R-CNN, our method not only respectively outperforms them by 6.1%, 4.7%, and 6.9% on mAP, but also it has great advantages in model size and inference speed.

Figure 5 shows the detection results of our FA-YOLO model for complex fabric images. The results in Fig. 5(a) and (b) respectively from YOLOv4 and FA-YOLO models. It can be seen that our method can detect the 'hole' defects, which are ignored by YOLOv4, with a high degree of confidence. Furthermore, our FA-YOLO can well detect dense defects in fabric images with complex backgrounds. This is confirmed by Fig. 5(c) and (d). As we see, some dense and overlapping defects can be successfully located by our FA-YOLO model.

Considering the two-stage methods have no advantages over the one-stage methods in terms of model parameters and inference speed, we only compare our method with the one-stage methods. For our proposed FA-YOLO, the model size is 174.6 MB, which is 69.6 MB less than YOLOv4 and 25.8 MB less than YOLOv4-CSP. The main reason is that we optimize the convolutional block by introducing a lighter CSP structure, and removing the SPP of the network and its previous convolutional block. In addition, FA-YOLO reaches 40 FPS in the inference speed of the model, which is only 8 FPS lower than YOLOv4, surpassing YOLOv3 and QueryDet, which may meet requirements of real-time detection in industrial scenarios. Note that in real industrial application scenarios, the FPS of the general algorithm is above 30, which can meet the needs of real-time detection. In particular, the performance of YOLOX is very impressive, with an inference speed of 69 FPS and a model size of 71.9 MB. Although our proposed FA-YOLO is slightly lower in inference speed than YOLOv4 and YOLOX, it is generally believed that the accuracy is more important when real-time performance of detection is required.

The second set of experiments come from ZJU-Leaper dataset [24]. According to the evaluation indicators in [24], we use F1-score to evaluate the performance of different models. The comparison is shown in Table 4. As we see, FA-YOLO has achieved the best overall score based on 5 sets of data, and the average F1-score from 5 sets of data reaches 0.714. Our best results are respective achieved in Group2, Group3 and Group5. Note that Group2 and Group3 are relatively complex backgrounds of fabric patterns. It can be seen that FA-YOLO also shows a strong advantage in defect detection for complex backgrounds.

Table 4 Comparison between test results (F1-score) of various methods and proposed FA-YOLO based on ZJU-Leaper dataset.

Methods	Group1	Group2	Group3	Group4	Group5	Overall
Faster R-CNN [26]	0.642	0.701	0.612	0.624	0.542	0.624
Cascade R-CNN [27]	0.697	0.712	0.702	0.674	0.521	0.661
Sparse R-CNN [28]	0.708	0.715	0.682	**0.704**	0.738	0.701
QueryDet [32]	**0.713**	0.702	0.696	0.681	0.732	0.705
YOLOv3 [29]	0.564	0.608	0.653	0.602	0.506	0.587
YOLOX [30]	0.651	0.680	0.648	0.617	0.731	0.665
Proposed	0.694	**0.728**	**0.729**	0.663	**0.756**	**0.714**

Table 5 Comparison of test results of various methods and proposed FA-YOLO based on NEU-DET dataset.

Methods	Backbone	mAP/%
Faster R-CNN [26]	ResNet-50	76.6
Cascade R-CNN [27]	ResNet-50	75.8
Sparse R-CNN [28]	ResNet-50	74.5
YOLOv3 [29]	DarkNet	72.3
YOLOv4 [10]	CSPDarkNet	71.1
YOLOX [30]	CSPDarkNet	75.3
Proposed	CSPDarkNet	**77.2**

In order to further verify the effectiveness of FA-YOLO, we also conduct the third set of experiments based on public NEU-DET dataset (steel defect). The comparison results are shown in Table 5. It can be seen that the mAP of our proposed method has reached the highest, scoring 77.2%. Although we do not optimize the model for this dataset, our model still achieves good results. This also prove that our method may have good generalization ability to other industrial defect datasets. To sum up, compared with other SoTA methods, our FA-YOLO model has achieved significant advantage in accuracy under the condition of satisfying real-time defect detection in industrial scenarios. FA-YOLO also presents better performance to locate targets in fabric images with complex backgrounds.

4. Conclusion

In this paper, we propose a novel defect detection method (called FA-YOLO) for fabric images based on YOLOv4. The k-means++ algorithm is first used to perform dimensionality clustering to generate prior anchor boxes. Then, we unify the large-scale detection head, residual feature augmentation, CSP structure, attention mechanisms, and ASFF in YOLOv4. Through these strategies, we have greatly strengthened the efficiency of feature fusion and improved the learning ability of the network. Experimental results based on real industrial datasets demonstrate the effectiveness of our method. By comparison with other models, FA-YOLO shows better detection performance. Note that the proposed method is implemented in pure Pytorch and no other acceleration methods are used, there is still a lot of room for improvement in the inference speed of the model. Future works include the use of structural re-parameterization to achieve lossless model pruning and the use of C language to compile some modules for acceleration.

Acknowledgments

We appreciate the support from the National Natural Science Foundation of China (No.61601410, U22A2004), the Key Research and Development Program Foundation of Zhejiang (No.2022C01079), and the 2021 Industrial Technology Foundation Public Service Platform Project (No.2021-0174-1-1).

References

[1] Y. Sun and J. Yu, "Fault detection of rolling bearing using sparse representation-based adjacent signal difference," IEEE Trans. Instrum. Meas., vol.70, pp.1–16, 2021.

[2] Y. Wang, X. Yu, and C. Wu, "Optic disc detection based on saliency detection and attention convolutional neural networks," IEICE Trans. Fundamentals, vol.E104-A, no.9, pp.1370–1374, Sept. 2021.

[3] Y. Huang, J. Jing, and Z. Wang, "Fabric defect segmentation method based on deep learning," IEEE Trans. Instrum. Meas., vol.70, pp.1–15, 2021.

[4] H. Xie, Y. Li, X. Li, and L. He, "A method for surface defect detection of printed circuit board based on improved YOLOv4," 2021 IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE), pp.851–857, IEEE, 2021.

[5] Y. Yang, G. Xie, and Y. Qu, "Real-time detection of aircraft objects in remote sensing images based on improved YOLOv4," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), vol.5, pp.1156–1164, IEEE, 2021.

[6] F. Abdurahman, K.A. Fante, and M. Aliy, "Malaria parasite detection in thick blood smear microscopic images using modified YOLOV3 and YOLOV4 models," BMC Bioinformatics, vol.22, no.1, pp.1–17, 2021.

[7] W. Lyu, Q. Lin, L. Guo, C. Wang, Z. Yang, and W. Xu, "Vehicle detection based on an imporved faster r-cnn method," IEICE Trans. Fundamentals, vol.E104-A, no.2, pp.587–590, Feb. 2021.

[8] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," Proc. IEEE International Conference on Computer Vision, pp.2980–2988, 2017.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.779–788, 2016.

[10] A. Bochkovskiy, C.-Y. Wang, and H.-Y.M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," arXiv preprint, arXiv:2004.10934, 2020.

[11] D. Wang, Y. Wang, M. Li, X. Yang, J. Wu, and W. Li, "Using an improved YOLOv4 deep learning network for accurate detection of whitefly and thrips on sticky trap images," Trans. ASABE, vol.64, no.3, pp.919–927, 2021.

[12] Y. Kahraman and A. Durmuşoğlu, "Deep learning-based fabric defect detection: A review," Text. Res. J., vol.93, no.5-6, pp.1485–1503, 2023.

[13] Z. Zhao, K. Gui, and P. Wang, "Fabric defect detection based on cascade faster R-cnn," Proc. 4th International Conference on Computer Science and Application Engineering, pp.1–6, 2020.

[14] J. Wu, J. Le, Z. Xiao, F. Zhang, L. Geng, Y. Liu, and W. Wang, "Automatic fabric defect detection using a wide-and-light network," Appl. Intell., vol.51, no.7, pp.4945–4961, 2021.

[15] M. An, S. Wang, L. Zheng, and X. Liu, "Fabric defect detection using deep learning: An improved faster R-approach," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), pp.319–324, IEEE, 2020.

[16] L. Zheng, X. Wang, Q. Wang, S. Wang, and X. Liu, "A fabric defect detection method based on improved YOLOv5," 2021 7th International Conference on Computer and Communications (ICCC), pp.620–624, IEEE, 2021.

[17] X. Luo, Q. Ni, R. Tao, and Y. Shi, "A lightweight detector based on attention mechanism for fabric defect detection," IEEE Access, vol.11, pp.33554–33569, 2023.

[18] T. Lu, S. Jia, and H. Zhang, "MemFRCN: Few shot object detection with memorable faster-RCNN," IEICE Trans. Fundamentals, vol.E105-A, no.12, pp.1626–1630, Dec. 2022.

[19] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," arXiv preprint, arXiv:1704.04861, 2017.

[20] J. Jing, D. Zhuo, H. Zhang, Y. Liang, and M. Zheng, "Fabric defect detection using the improved YOLOv3 model," J. Eng. Fiber. Fabr., vol.15, p.1558925020908268, 2020.

[21] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "SimAM: A simple, parameter-free attention module for convolutional neural networks," International Conference on Machine Learning, pp.11863–11874, PMLR, 2021.

[22] Y.-C. Lee, H.-W. Hsu, J.-J. Ding, W. Hou, L.-S. Chou, and R.Y. Chang, "Backbone alignment and cascade tiny object detecting techniques for dolphin detection and classification," IEICE Trans. Fundamentals, vol.E104-A, no.4, pp.734–743, April 2021.

[23] A. Tianchi, "Smart diagnosis of cloth flaw dataset," https://tianchi.aliyun.com//dataset/dataDetail?dataId=79336

[24] C. Zhang, S. Feng, X. Wang, and Y. Wang, "ZJU-leaper: A benchmark dataset for fabric defect detection and a comparative study," IEEE Trans. Artif. Intell., vol.1, no.3, pp.219–232, 2020.

[25] Y.Y. Kechen Song, "Neu surface defect database," http://faculty.neu.edu.cn/songkechen/zh_CN/zdylm/263270/list/index.htm

[26] R. Girshick, "Fast R-CNN," Proc. IEEE International Conference on Computer Vision, pp.1440–1448, 2015.

[27] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving into high quality object detection," Proc. IEEE Conference on Computer Vision And Pattern Recognition, pp.6154–6162, 2018.

[28] P. Sun, R. Zhang, Y. Jiang, T. Kong, C. Xu, W. Zhan, M. Tomizuka, L. Li, Z. Yuan, C. Wang, and P. Luo, "Sparse R-CNN: End-to-end object detection with learnable proposals," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.14454–14463, 2021.

[29] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," arXiv preprint, arXiv:1804.02767, 2018.

[30] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "YOLOx: Exceeding yolo series in 2021," arXiv preprint, arXiv:2107.08430, 2021.

[31] C.-Y. Wang, A. Bochkovskiy, and H.-Y.M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.13029–13038, 2021.

[32] C. Yang, Z. Huang, and N. Wang, "QueryDet: Cascaded sparse query for accelerating high-resolution small object detection," 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022.

Kai Yu received the B.E degree in the School of Information Science and Engineering from Zhejiang Sci-Tech University, in 2021. He is currently pursuing the master degree at the School of Information, Zhejiang Sci-Tech University. His research interests include deep learning and computer vision.

Wentao Lyu received the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University in 2015. From 2008 to 2010, he was a Senior Algorithm Engineer in Eutrovision Inc., Shanghai, China. From 2018 to 2019, he was a Visiting Scholar with University of Waterloo, Waterloo, ON, Canada. He has served as a co-drafter for the Construction guidelines of future factory of Zhejiang Province, China. He is currently an Associate Professor at the Zhejiang Sci-Tech University, Hangzhou, China. His research interests include deep learning, computer vision, industrial intelligence and intelligence manufacturing.

Xuyi Yu received the B.E degree in the School of Information Science and Engineering from Hangzhou Normal University, in 2020. He is currently pursuing the master degree at the School of Information, Zhejiang Sci-Tech University. His research interests include deep learning and computer vision.

Qing Guo received the M.Sc. degree in material science and engineering from Southeast University in 2015, Nanjing, China. He is currently the deputy director of department of smart manufacturing, Zhejiang Technology Innovation Service Center, Hangzhou 310007, China. He has served as a co-drafter for the Construction guidelines of future factory of Zhejiang Province, China. His research interests include industrial intelligence and intelligence manufacturing.

Weiqiang Xu received his M.Sc. degree in Communications and Information System from Southwest Jiao-Tong University, China, and his Ph.D. degree in Control Science and Engineering from Zhejiang University, China, in 2003 and 2006, respectively. He was a Postdoctoral Researcher with Zhejiang University and a Visiting Scholar with Columbia University, New York, NY, USA. He is currently a Professor with the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou. His research interests mainly include Internet of things, industrial Internet, industrial intelligence and intelligence manufacturing.

Lu Zhang received the B.E degree in the School of Information Science and Technology from Beijing Information Science and Technology University in 2020, Beijing, China. She is currently pursuing the master degree at the School of Information Science and Technology, Zhejiang Sci-Tech University, Hangzhou, China. Her research interests include machine learning and computer vision.