CARL-YOLOF: A well-efficient model for digital printing fabric defect detection

Jingwei Wu¹, Pengfei Li¹, Huanhuan Zhang¹, and Zebin Su¹,²

Abstract
Deep-learning models have been successfully applied to fabric defect detection fields. However, due to the lack of available dataset for the digital printing fabric defect detection field, the study is still under-explored. Besides, there are two tough challenges that currently existing deep-learning models cannot resolve well. On the one hand, the actual production requires the models to improve its detection accuracy as much as possible while ensuring real-time detection. On the other hand, the objectives of classification and localization are correlated, so the link between classification and localization should be established for an excellent object detection model. To solve these problems, such Classification-Aware Regression Loss (CARL) is embedded into the YOLOF to correlate the classification and localization tasks so that the model is named CARL-YOLOF. The performance of the proposed model has been trained and evaluated by the self-built digital printing fabric defect detection dataset (DPFD-DET). Experimental results demonstrate that CARL-YOLOF achieves 0.54 AP on COCO metrics, which achieves a gain of 0.04 compared with YOLOF. Besides, CARL-YOLOF maintains the speed advantage of YOLOF, which reaches 42 FPS. Therefore, the proposed model is suitable for the real-time digital printing fabric defect detection.

Keywords
Digital printing, fabric defect detection, YOLOF, CARL, deep learning

Introduction
The printing and dyeing industry is an important part of textile industry, and the textile printing is a key operation to enhance the additional value of textiles.¹ Digital printing technology, a new type of printing technology, provides textiles with more realistic, brighter and natural printing effects, and further enhances the additional value of textiles compared with the customary printing technology (e.g. roller printing, screen printing, transfer printing, etc.).² At the same time, a series of complex factors in printing process may be harmful for textile quality³ such as flawed sizing and dyeing for while gray fabrics,⁴ instable ink-jet printing system,⁵ drips of water from steamer etc. Which will lead to printing defects such as PASS track, PASS tracks, ink leakage, blank, ghost, fluff, water leakage, pulp, etc. Figure 1 visualizes samples of the digital printing fabric defects. Therefore, digital printing fabric defect detection has become an effective yet indispensable link of printing and dyeing quality monitoring.

In recent years, it is still a hot topic that the theoretical knowledge of deep learning is applied effectively to fabric defect detection field. Peng et al.⁶ proposed a Priori Anchor Convolutional Neural Network (PRAN-Net) for fabric defect detection to improve the detection and location accuracy of fabric defects. Zhu et al.⁷
proposed a modified DenseNet model to better suit a resource-constrained edge computing scenario. Li and Li proposed three tricks to further improve the precision for fabric defect detection based on Cascade R-CNN. However, due to the limitations of the available dataset, the previous research objects of fabric defects mainly focused on white gray fabrics, pure color fabrics, and single-texture color fabrics, while limited research was conducted on digital printing fabrics.

The improvement of digital printing fabric defect detection performance is closely related to the development of detectors. The definition of positive and negative anchors and loss function are two core components of detector, their continuous enrichment in theoretical knowledge has great significance to the development of detector.

In terms of the definition of positive and negative anchors, we simply review the following representative methods such as Max-IoU, ATSS, Uniform-match. First, Max-IoU adopts IoU (Intersection over Union) between anchors and ground-truth boxes as a measure of positive and negative anchor, it has been flexibly applied to SSD, BAGS, and PISA. Then, ATSS uses dynamic IoU threshold to more efficiently match a stable number of positive anchors for each object. Unlike the above-mention methods, Uniform-match is based on the center point distance between the ground-truth boxes and the anchor boxes, the prediction boxes corresponding to the anchor boxes to match a fixed number of positive anchors for each ground-truth boxes, which improves the model’s ability to detect each object in a sparse space.

As far as the loss function of object detection is concerned, it is specifically divided into classification loss function and bounding box regression loss function, which cooperate to complete the detection task. For the classification loss function, it often be asked to improve the model’s ability to detect positive samples through hard sampling or soft sampling. For the bounding box regression loss function, it often considers how to make the bounding box better regression, which is reflected in the development trajectory of the regression loss function. However, the above two tasks do not generate a direct connection, so the generation of classification perception regression ideas has aroused widespread attention of researchers.

In this paper, CARL-YOLOF is proposed to address the problems as aforementioned, which is inspired by YOLOF and CARL. The main contributions of our study can be concluded into three aspects:

a. The proposed CARL-YOLOF achieves higher detection accuracy compared with YOLOF.
b. We set up a digital printing defect dataset DPFD-DT, which provides research sample for the field of digital printing.
c. Our model is more suitable for the real-time digital printing fabric defect detection compared with other advanced detectors.

![Figure 1. Some example images in the digital printing defect database: (a) PASS track, (b) PASS tracks, (c) ink leakage, (d) blank, (e) ghost, (f) fluff, (g) water leakage, and (h) pulp.](image-url)
The YOLOF baseline

As a representative of the current advanced detectors, YOLOF achieves an effective balance between speed and accuracy compared with other detectors, and its success stems from the objective analysis for feature pyramids networks (FPN)\textsuperscript{23} based on a large number of ablation experiments. Therefore, a conclusion can be got that FPN can efficiently improve the accuracy of the detector, which mainly depends on divide-and-conquer strategy instead of multiple level features fusion. In addition, the introduction of FPN will cause the detection speed of the detector to be seriously hindered, because its complex path connection method aggravates the memory access frequency in the forward inference process, not to mention other more complex path connection methods.\textsuperscript{24–26} To solve these problems, YOLOF proposed Dilated Encoder with a simple network structure, bridging the accuracy gap between Dilated Encoder and FPN, the structure can be visualized as shown in Figure 2. First, $1 \times 1$ convolution was applied to channel reduction with rate of 4. Then, $3 \times 3$ convolution with dilation was used to enlarge the receptive field. In addition, $1 \times 1$ convolution was adopted to recover the number of channels. At last, shortcut between the input and output was used to explore the model’s perception for large-scale objects, while preserving the original input perception for small-sized objects.

CARL-YOLOF

The function of the detector can be summarized as the realization of two specific tasks, classification and regression. However, excellent completion of either one does not directly reflect the accuracy of the detector, the truth can be visualized as shown in Figure 3. As shown in Figure 3, the False Positive is reserved by non-maximum suppression (NMS), it significantly affects the performance of the detector. It can be seen that the False Positive’s confidence meets the requirements of positive score threshold, but their boxes obviously does not meet the requirements of positive IoU threshold, so we try to build a loss function that correlated bounding box regression and classification to further improve the accuracy of the detector.

Inspired by YOLOF and CARL, we proposed the CARL-YOLOF, the training pipeline can be visualized as follows: As shown in Figure 4, the training pipeline of CARL-YOLOF is designed based on YOLOF and CARL Module. For YOLOF, it is composed of three Components: (1) the Backbone, which is selected as ResNet50 for a quantitative study in the paper; (2) the Neck, which consists of four consequent Dilated Encoder; (3) the YOLOF Head, which is formulated according to the number of classes, standard of anchor generator and the number of input channels from Neck. CARL Module establishes the link between classification and regression. Specially, the regression branch gets supervised by the classification loss, the positive anchors with higher predicted probabilities receive large gradients for the regression, which enforces the model to pay more attention to them during the training process. Inspired by YOLOF and CARL, the proposed CARL-YOLOF maintains the streamlined network structure of YOLOF, and the embedded CARL Module strengthened the model’s attention to important samples. Therefore, compared with YOLOF, CARL-YOLOF achieves higher detection accuracy under the premise of not affecting the inference speed.

CARL module

CARL Module can be visualized as Figure 4, it contains two processes. First, we define the positive and negative anchors according to relation between anchor boxes and ground-truth boxes. Then, we establish loss function to train the model. For the former, Uniform-Match strategy is used to define positive and negative anchors according to relation between anchor boxes and ground-truth boxes. Then, we establish loss function to train the model.

For the former, Uniform-Match strategy is used to define positive and negative anchors. Specially, it matches a fixed number of positive anchors for each ground truth box based on the center point distance between the ground truth boxes and the anchor boxes, predicted boxes corresponding to the anchor boxes, which ensure that there are a large number of positive examples in the sparse anchor space.
For the latter, Focal Loss is applied to the classification task to improve the model’s ability for detecting difficult samples. In terms of box regression loss, we introduce CIoU Loss and CARL to construct the connection between classification and regression to improve the performance of the detector. The proposed regression loss function can be donated as follows:

\[ L_{carl} = \sum_{i=1}^{n} c_i L_{CIoU} \left( d_i, \hat{d}_i \right) \]  

\[ c_i = \frac{u_i}{\sum_{i=1}^{n} u_i} \]  

\[ u_i = \left( \left(1 - b\right) p_i + b \right)^k \]  

Specially, where degree factor \( k \) denotes the how much regression emphasis on important positive anchors, \( p_i \) means the predicted probability of the positive anchors, \( b \)
Figure 5. Examples of collected digital printing fabric defect images. Specially, digital printing fabric defect images (a)−(d) are captured from four different surface regions 1, 2, 3, and 4 respectively in the raw image.

is scale factor that $p_i$ impact on the regression task and $0 \leq b \leq 1$, then we use an exponential function to transform the $p_i$ to $u_i$, and then rescale it according to the average value of all samples to get $c_i$. $d_i$ denotes the output regression offset, $\hat{d}_i$ denotes the expected regression offset. $L_{CIoU}$ is the commonly used CIoU loss.

Obviously, there is a positive correlation between $p_i$ and $c_i$. In terms of positive anchors, a higher $p_i$ means additional increase of regression loss, which enforces the model to pay more attention to them during the training stage. Likewise, the positive anchors with higher $p_i$ are more important, because they are the first ones that reserved during the NMS in the inference stage. CARL Module enforces the proposed model to pay more attention to important samples instead of unimportant samples, which further improves the detection accuracy compared with YOLOF.

Experiments and results

Digital printing dataset

In this paper, we set up a comprehensive dataset DPFD-DET for the detection of digital printing defects. To create our dataset, we first collect digital printing defects images containing eight types of defects (PASS track, PASS tracks, ink leakage, blank, ghost, fluff, water leakage, pulp). Next, since high-resolution digital printing fabric defect images aggravate model’s training difficulty, we randomly crop the defect regions into lower resolution including $256 \times 256$, $384 \times 384$, $416 \times 416$, $512 \times 512$, the cropping operation is showed in Figure 5. Then, we label the collected dataset into eight classes by LabelImg software. Finally, the dataset was divided into training set, validation set and test set in the ratio 8:1:1. The detailed statistics of per class are presented in Table 1.

Implementation details and metrics

Implementation details

The initialization weights of all backbones are transferred from the trained models on the COCO2017 dataset. All other weights are initialized through the “Xavier” paradigm. For the fine-tuning process, we fix the max-epoch as 30, and use SGD as optimizer. For the SGD optimizer, the basic learning rate is set as 0.001, the weight decay is set as 0.0001, and the momenta constant is set as 0.9, batch size is set as 4. All the model trainings are implemented on a deep learning workstation with Intel E5-2680v3 processor(2.5 GHz), 128 GB memory and 1 NVIDIA GeForce 2080 Ti.

Metrics. For our experiments, FPS and COCO metrics ($AP$, $AP_{50}$, $AP_{75}$, $AP_S$, $AP_M$, $AP_L$) are used for evaluating model’s performance. (1) FPS: the detection speed; (2) COCO metrics: the detection accuracy; the evaluate metrics FPS is respectively defined as follow:

$$FPS = \frac{N_{\text{frames}}}{T_{\text{total}}}$$

Where $N_{\text{frames}}$ represents the total image frames, $T_{\text{total}}$ indicates the total spend time. To calculate FPS (include
inference stage and the post-processing stage), batch size is set as 1 and the shape of the input image is fixed to $416 \times 416$.

### Overall experiments

We used ResNet50 as the backbone network, and adopted the same configuration to validate our experimental results. The performance of CARL-YOLOF compared with other advanced models is visualized in Table 2. Compared with YOLOF, CARL-YOLOF finishes different degrees of improvement in $AP$, $AP_{75}$, $AP_{50}$, $AP_{S}$, $AP_{M}$, $AP_{L}$ metrics based on the same inference speed, respectively improved by 0.04, 0.03, 0.03, 0.04. Compared with other models, our model shows the best performance in the comprehensive $AP$ metrics. Besides, our model locates in leading position in $AP_{75}$ and $AP_{L}$ metrics, showing that it can meet more stringent bounding box regression criteria and has excellent ability to detect large-scale defects. However, our model shows the lower performance in the $AP_{50}$, $AP_{S}$, $AP_{M}$ metrics compared with Faster-RCNN, which originates from more streamlined network structure reduces its expressive ability.

Table 1. The detailed statistics of per class of the DPFD-DET.

| Dataset | Class          | Training set | Validation set | Test set | Total  |
|---------|----------------|--------------|----------------|---------|--------|
| DPFD-DET| PASS track     | 1140         | 285            | 285     | 1710   |
|         | PASS tracks    | 450          | 113            | 113     | 676    |
|         | Ink leakage    | 1500         | 274            | 274     | 2048   |
|         | Blank          | 990          | 247            | 247     | 1484   |
|         | Ghost          | 700          | 176            | 176     | 1052   |
|         | Fluff          | 1564         | 392            | 392     | 2348   |
|         | Water leakage  | 590          | 146            | 146     | 882    |
|         | Pulp           | 80           | 20             | 20      | 120    |

Table 2. The performance of CARL-YOLOF compared with the others on DPFD-DET test set.

| Framework | Backbone | FPS | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_{S}$ | $AP_{M}$ | $AP_{L}$ |
|-----------|----------|-----|------|-----------|-----------|----------|----------|----------|
| ATSS      | ResNet50 | 25  | 0.53 | 0.73      | 0.59      | 0.28     | 0.36     | 0.60     |
| Faster RCNN | ResNet50 | 14  | 0.51 | 0.76      | 0.58      | 0.43     | 0.52     | 0.55     |
| FCOS      | ResNet50 | 29  | 0.46 | 0.73      | 0.48      | 0.24     | 0.31     | 0.51     |
| RetinaNet | ResNet50 | 26  | 0.39 | 0.55      | 0.47      | 0.18     | 0.24     | 0.48     |
| RepPoints | ResNet50 | 25  | 0.40 | 0.64      | 0.42      | 0.26     | 0.31     | 0.43     |
| YOLOF     | ResNet50 | 42  | 0.50 | 0.73      | 0.56      | 0.19     | 0.34     | 0.59     |
| our       | ResNet50 | 42  | 0.54 | 0.73      | 0.59      | 0.22     | 0.34     | 0.63     |

Table 3. The hyperparameters ($k, b$) of the CARL module’s effect on detection accuracy.

| $k$ | $b$ | $AP$ | $AP_{50}$ | $AP_{75}$ | $AP_{S}$ | $AP_{M}$ | $AP_{L}$ |
|-----|-----|------|-----------|-----------|----------|----------|----------|
| 0.5 | 0   | 0.52 | 0.72      | 0.59      | 0.21     | 0.35     | 0.61     |
| 0.5 | 0.2 | 0.54 | 0.73      | 0.59      | 0.22     | 0.34     | 0.63     |
| 0.5 | 0.4 | 0.51 | 0.73      | 0.58      | 0.21     | 0.40     | 0.61     |
| 0.5 | 0.6 | 0.50 | 0.74      | 0.54      | 0.22     | 0.34     | 0.60     |
| 0.5 | 0.8 | 0.52 | 0.74      | 0.58      | 0.28     | 0.33     | 0.61     |
| 0.5 | 1   | 0.50 | 0.72      | 0.55      | 0.19     | 0.32     | 0.59     |
| 1   | 0   | 0.47 | 0.65      | 0.53      | 0.17     | 0.23     | 0.58     |
| 1   | 0.2 | 0.47 | 0.65      | 0.56      | 0.18     | 0.30     | 0.58     |
| 1   | 0.4 | 0.48 | 0.66      | 0.52      | 0.19     | 0.29     | 0.50     |
| 1   | 0.6 | 0.49 | 0.69      | 0.55      | 0.19     | 0.32     | 0.58     |
| 1   | 0.8 | 0.47 | 0.70      | 0.50      | 0.19     | 0.31     | 0.55     |
| 1   | 1   | 0.50 | 0.73      | 0.56      | 0.19     | 0.34     | 0.59     |
| 2   | 0   | 0.49 | 0.66      | 0.57      | 0.18     | 0.25     | 0.59     |
| 2   | 0.2 | 0.48 | 0.69      | 0.55      | 0.19     | 0.31     | 0.59     |
| 2   | 0.4 | 0.49 | 0.68      | 0.54      | 0.21     | 0.33     | 0.58     |
| 2   | 0.6 | 0.51 | 0.71      | 0.58      | 0.22     | 0.34     | 0.60     |
| 2   | 0.8 | 0.54 | 0.74      | 0.61      | 0.22     | 0.33     | 0.63     |
| 2   | 1   | 0.52 | 0.74      | 0.59      | 0.21     | 0.37     | 0.62     |

**Ablation experiments of hyper-parameters.** In order to verify the impact of the hyperparameters ($k, b$) of the CARL module on the detection accuracy, the degree factor $k$ adopts three alternative constants values (0.5, 1, 2), $b$ is sets as a constant (0–1 with an interval of 0.2).

Through the Table 3, a conclusion can be summarized that object detection task is the unification of the classification task and the bounding box regression task. Specifically, when $b$ is constant, hyperparameter $k$ often dominates the influence of the classification probability on the bounding box regression; when $k$ is constant, the scale hyperparameter $b$ can reflect the proportion of the impact...
of classification probability on the bounding box regression, so the detector accuracy improves with respect to a good selection of the constant $b$. A large number of ablation experiments have been verified, hyperparameter setting ($k=2, b=0.8$) in CARL make our model achieve the best detection accuracy. Compared with YOLOF ($k=1, b=1$), the $AP$ metrics improved by 0.04. Figure 6 shows detection examples on DPFD-DET test set with CARL-YOLOF model.

**Conclusion**

In this paper, we propose CARL-YOLOF model for digital printing fabric defect detection. The proposed model efficiently balances the detection accuracy and speed to a certain extent, which has theoretical reference value for the solution that digital printing fabric defects are difficult to detect.

Classification-aware regression strategy can effectively strengthen the link between classification task and regression task in YOLOF model. Experimental results show that CARL-YOLOF with ResNet50 backbone network achieves different levels of improvement in $AP$, $AP_{75}$, $AP_{5}$, $AP_{L}$ metrics in our own DPFD-DET. Besides, CARL-YOLOF maintains the speed advantage of YOLOF, which reaches 42 FPS. Therefore, the proposed model is suitable for the real-time digital printing fabric defect detection.

**Declaration of Conflicting Interests**

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

**Funding**

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the National Natural
Science Foundation of China (61902302), The key research and development program funded by Shaanxi province (2019ZDLGY01-08), The key research and development program funded by Shaanxi province (2021GY-311), the Shaanxi Provincial College of Science and Technology Youth Talent Support Project (20200115), Xi’an Science and Technology Plan Project (21XJZZ0011), Innovation Capability Support Project of Shaanxi (2021TD-29).

ORCID iDs
Huanhuan Zhang https://orcid.org/0000-0001-5745-1775
Zebin Su https://orcid.org/0000-0002-8822-7589

Reference
1. Yang XH, Sui JH and Zhang XW. Auto-generating uniform stochastic web images for ink-jet printing textiles. Text Res J 2010; 80(18): 1942–1948.
2. Wang C and Wang C. Surface pretreatment of polyester fabric for ink jet printing with radio frequency O2 plasma. Fibers Polym 2010; 11(2): 223–228.
3. Cui P and Xue Y. An intelligent weavable dyeing method for the environmentally-friendly dyeing of fabric. Alex Eng J 2021; 60(5): 4639–4649.
4. Ma P. Sizing and dyeing property improvement of cotton fabric treated with corona discharge. Asian J Chem 2012; 24(9): 3833–3836.
5. Dehghani A, Jahanshah F, Borman D, et al. Design and engineering challenges for digital ink-jet printing on textiles. Int J Clothing Sci Technol 2004; 16(1/2): 262–273.
6. Peng P, Wang Y, Hao C, et al. Automatic fabric defect detection method using PRAN-Net. Appl Sci 2020; 10(23): 8434.
7. Zhu Z, Han G, Jia G, et al. Modified densenet for automatic fabric defect detection with edge computing for minimizing latency. IEEE Internet Things J 2020; 7(10): 9623–9636.
8. Li F and Li F. Bag of tricks for fabric defect detection based on Cascade R-CNN. Text Res J 2021; 91(5-6): 599–612.
9. Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2017; 39(6): 1137–1149.
10. Zhang S, Chi C, Yao Y, et al. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp.9759–9768. New York: IEEE.
11. Chen Q, Wang Y, Yang T, et al. You only look one-level feature. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp.13039–13048. New York: IEEE.
12. Liu W, Anguelov D, Erhan D, et al. SSD: Single shot multi-box detector. In: European conference on computer vision, 2016, pp.21–37. Cham: Springer.
13. Li Y, Wang T, Kang B, et al. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp.10991–11000. New York: IEEE.
14. Cao Y, Chen K, Loy CC, et al. Prime sample attention in object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp.11583–11591. New York: IEEE.
15. Wang Y, Wang C, Zhang H, et al. Automatic ship detection based on RetinaNet using multi-resolution Gaofen-3 imagery. Remote Sens 2019; 11(5): 531.
16. Joseph REDMON and Farhadi ALI. YOLOv3: an incremental improvement. arXiv preprint arXiv:1804.02767, 2018, 2, 4, 7, 11.
17. Bochkovskiy A, Wang CY and Liao HYM. Yolo4: optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.7.
18. Purkait P, Zhao C and Zach C. SPP-Net: deep absolute pose regression with synthetic views. arXiv preprint arXiv:1712.03452, 2017.
19. Girshick R. Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, 2015, pp.1440–1448. New York: IEEE.
20. Yu J, Jiang Y, Wang Z, et al. Unitbox: an advanced object detection network. In: Proceedings of the 24th ACM international conference on Multimedia, 2016, pp.516–520. Amsterdam: ACM.
21. Rezagholif Shu, Tsoi N, Gwak JY, et al. Generalized intersection over union: a metric and a loss for bounding box regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp.658–666. New York: IEEE.
22. Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: faster and better learning for bounding box regression. In: Proceedings of the AAAI conference on artificial intelligence, 2020, 34(07), pp.12993–13000. New York: AAAI.
23. Lin TY, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp.2117–2125. New York: IEEE.
24. Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2020, pp.10991–11000. New York: IEEE.
25. Li Y, Wang T, Kang B, et al. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp.10991–11000. New York: IEEE.
26. Qiao S, Chen LC and Yuille A. Detectors: detecting objects with recursive feature pyramid and switchable atrous convolution. arXiv 2020. arXiv preprint arXiv:2006.02334.