Journal of computer science and software applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 4, No. 5, 2024

Defect Detection in Manufacturing: Leveraging YOLOv8 for Real-Time Quality Control on Production Lines

Blake Thurston

University of California, Riverside, USA

blake.thurston98@ucr.edu

Abstract:

With the rapid advancements in global manufacturing and the increasing shift towards intelligent systems, production line efficiency and product quality have become critical competitive factors. Traditional manual defect detection methods suffer from inefficiency, subjectivity, and high miss rates, which hamper the stability of product quality and production efficiency. This study addresses these challenges by leveraging advanced machine vision and deep learning technologies, specifically focusing on the latest YOLOv8 algorithm, to design and implement an intelligent defect detection system for production line parts. The YOLOv8 algorithm, with its enhanced network structure, loss function, and training strategies, offers superior detection speed and accuracy, making it ideal for real-time industrial applications. The research involved constructing a comprehensive dataset of part defects, followed by meticulous model training and optimization. The experimental results demonstrate that the improved YOLOv8 model significantly outperforms traditional methods in terms of accuracy and efficiency, effectively mitigating the issues of subjectivity and inconsistency. Furthermore, the study explored the deployment of the model using the lightweight ShuffleNetV2 backbone, enhancing its stability and applicability in complex production environments. The findings provide robust support for the intelligent transformation of manufacturing processes and establish a solid foundation for future industrial applications.

Keywords:

YOLOV8, Defects in parts, Production line.

1. Introduction

With the rapid development of global manufacturing and the deepening of intelligent transformation, production line efficiency and product quality have become key indicators of an enterprise's competitiveness. In highly automated production environments, defect detection is a crucial aspect of quality control, underscoring its importance. However, traditional manual defect detection methods are not only inefficient but also prone to inconsistency and higher miss rates due to subjective differences in defect definitions and operator visual fatigue. These issues severely limit the stability of product quality and the improvement of production efficiency.

To address these industry challenges, exploring and applying advanced machine vision and deep learning technologies for efficient and accurate defect detection has become a crucial direction for the intelligent transformation of manufacturing. In recent years, deep learning algorithms, with their powerful feature extraction and generalization capabilities, have made significant progress in image recognition and object detection, offering new solutions for defect detection. Specifically, the YOLO (You Only Look Once) series of algorithms, known for their high speed and accuracy, have shown great potential in real-time object detection tasks.

This project focuses on the challenge of defect detection in production line parts, utilizing the latest YOLOv8 deep learning algorithm to design and implement an efficient and intelligent defect detection system. As the newest member of the YOLO series, YOLOv8 not only inherits the advantages of its predecessors but also features optimizations in network structure, loss function, and training strategies, further enhancing detection speed and accuracy, making it particularly suitable for industrial detection scenarios with high real-time requirements.

The project aims to solve the issues of subjectivity, inconsistency, and inefficiency associated with manual detection by thoroughly researching the application of the YOLOv8 algorithm in production line part defect detection. By constructing a dataset suitable for detecting various types of part defects and training and optimizing the YOLOv8 model, the project seeks to achieve fast, real-time, and accurate identification of defects. Additionally, the project will explore deployment and optimization strategies for the model to ensure the detection system can operate stably in the complex and variable production line environment, providing strong support for the intelligent technological transformation of the manufacturing industry.

2. Current research status at home and abroad

In the field of machine vision, object surface defect detection is a critical area of research. It involves using machine vision equipment to capture image information for the automatic identification of defects within those images. In recent years, with the breakthrough success of deep learning models—particularly convolutional neural networks (CNNs) in computer vision—defect detection methods based on deep learning have been widely applied in various industrial scenarios. Compared to traditional machine vision methods, deep learning approaches have demonstrated higher accuracy and effectiveness in object surface defect detection.

In many real-world applications, the speed of object detection is a crucial consideration. To meet this demand, researchers have been exploring faster object detection algorithms, leading to the development of regression-based algorithms like YOLO (You Only Look Once) and SSD (Single Shot MultiBox Detector). These algorithms not only ensure detection reliability but also significantly improve computation speed, effectively balancing detection speed and accuracy.

In 2016, Redmon et al. introduced the YOLO algorithm[1], marking a new era of single-stage and two-stage detection algorithms based on deep learning. The YOLO algorithm discards the traditional candidate box extraction step, instead directly using regression methods for object classification and candidate box prediction. In 2017, J. Redmon et al. further proposed the YOLO9000 algorithm[2] based on YOLOv1, which significantly surpassed other algorithms of the time in detection speed while also demonstrating excellent detection accuracy. In 2018, the introduction of the YOLOv3 algorithm further optimized YOLO's performance. By 2020, Bochkovskiy et al. launched YOLOv4[4], which successfully balanced speed and accuracy by ingeniously integrating previous research findings and introducing some new innovations.

Each subsequent upgrade has aimed to further enhance YOLO's detection speed and localization accuracy. The latest in the YOLO series is YOLOv8, introduced by Ultralytics as the next major update to the open-sourced YOLOv5 in January 2023. Although not yet widely applied in industrial scenarios, Liu Junhao et al. significantly improved defect detection accuracy and speed in YOLOv5 by incorporating lightweight modules and the Polarized Self- Attention module, though the model's generalization performance still needs enhancement[5]. Li et al. successfully made the YOLO network fully convolutional, achieving high-accuracy and real-time defect detection, yet improvements are still needed for small-object defect detection[6]. Wang et al., based on YOLOv8, optimized the spatial pyramid pooling layer using the SimSPPF module and combined the BiFPN method with the LSK dynamic large convolution

Journal of computer science and software applications
https://www.mfacademia.org/index.php/jcssa
ISSN:2377-0430
Vol. 4, No. 5, 2024

kernel attention mechanism to enhance the model's defect detection accuracy, although there remain shortcomings in small-object defect detection[7].

3. Project Research Content and Implementation

The basic design approach of this project is divided into four main steps: data collection and annotation, data preprocessing, model training, model evaluation and optimization, and model deployment and application.

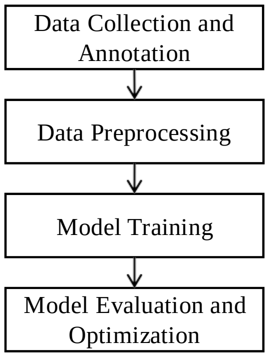
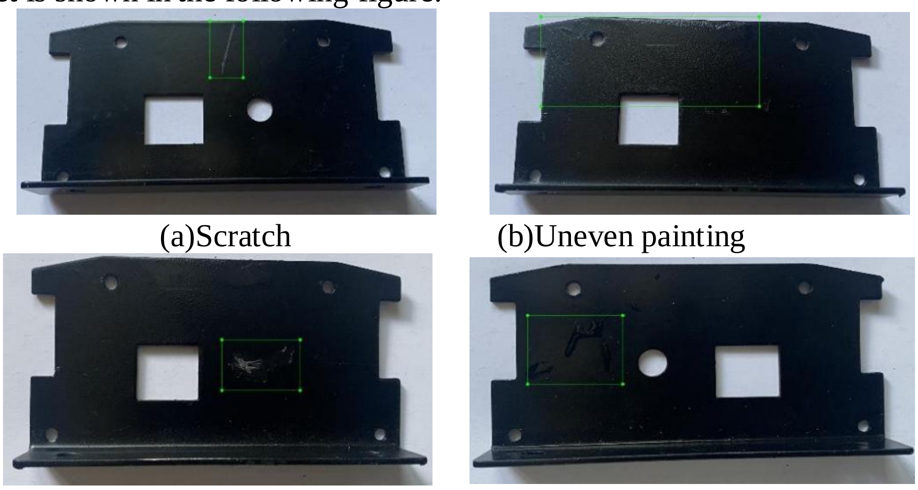


Figure 1. Implementation approach

References are cited in the text just by square brackets [1]. (If square brackets are not available, slashes may be used instead, e.g. /2/.) Two or more references at a time may be put in one set of brackets [3, 4]. The references are to be numbered in the order in which they are cited in the text and are to be listed at the end of the contribution under a heading References, see our example below.

3.1. Data Collection and Annotation

Due to the wide variety of industrial parts and the different types of defects that may arise for different parts, this project requires data collection tailored to the specific conditions of a particular industry or enterprise. Since defects in parts generally occupy a small proportion during production, a large amount of data collection is challenging. Additionally, there are currently no particularly good publicly available datasets. The dataset used in this study comes from internal sources within the enterprise, where users or employees captured and uploaded the images. Initially, this study collected 10,000 images. However, since most of the images did not meet the algorithm's requirements, manual screening and background removal were performed, leaving only 2,000 images that met the criteria. The parts were then classified according to their degree of damage. The resulting dataset is shown in the following figure:



(a)Scratch (b)Uneven painting

(c)Paint missing (d)Oil stain

Figure 2. Types of defects in parts

Given that the dataset is primarily composed of screenshots taken by employees and users on their mobile screens, these images often include complex background environments. When outlining defects, especially those on the edges, part of the background may be included. To eliminate potential interference from the background in the subsequent algorithm training process, it is necessary to perform background removal on the images. Through technical evaluation and selection, this study ultimately adopted the rembg framework as the solution for background removal. This framework is specifically designed for image background removal and is implemented in Python. It is renowned for its efficient and precise ability to identify and remove non-subject parts of images, perfectly meeting the project's requirements. The example is as follows:

Figure 3. The result after removing the background

Next, we need to perform data annotation. The tool we use is LabelImg. When using tools like LabelImg to annotate defects on mobile screens, the main purpose is to provide Ground-truth data required for training machine learning or deep learning models. This data includes the location and classification information of the targets (i.e., defects on the mobile screens) in the images.

Figure 4. Using LabelImg to label

Given that the amount of data was still too small, we further expanded the dataset of 2,000 images through data augmentation techniques, resulting in a total of 4,000 images for this research project. We used imgaug for data augmentation, applying various random combinations of techniques such as image flipping, rotation, color jitter, contrast enhancement, etc. During the augmentation process, the corresponding transformations were also applied to keypoints and bounding boxes. In the object detection process, the training set includes the images and their corresponding bounding box files. When augmenting the images, we simultaneously compute the coordinates of the transformed bounding boxes to generate the corresponding bounding box files. This way, the newly augmented dataset does not require manual re-annotation; instead, the labels are automatically generated.

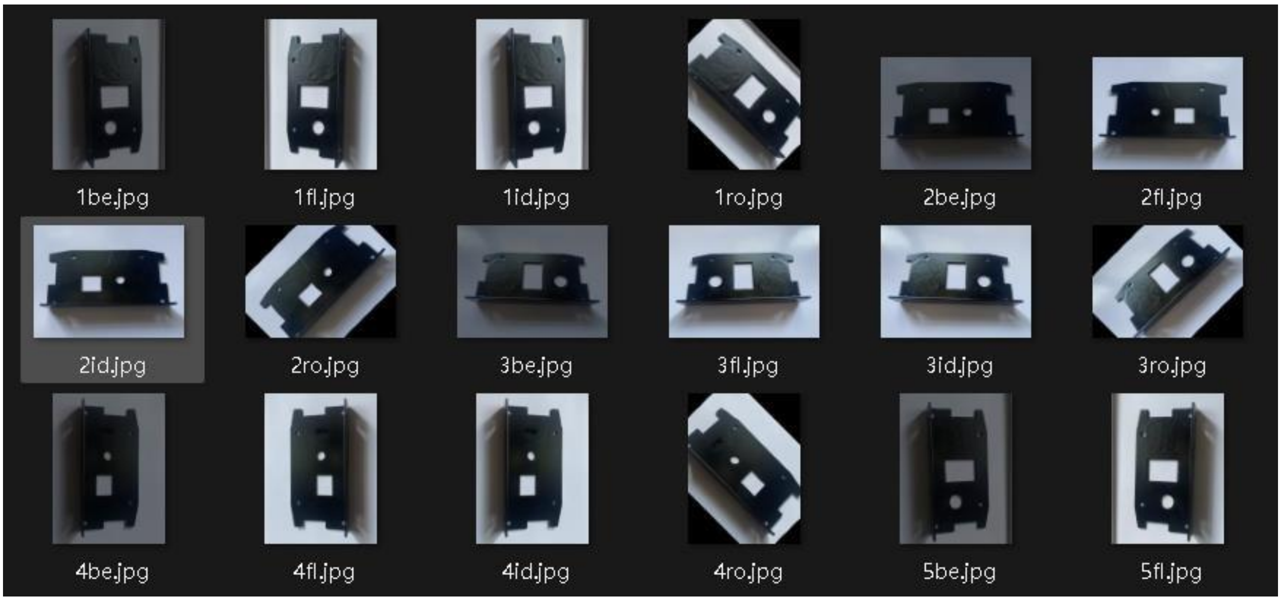


Figure 5. Results of image augmentation

3.2. Data Preprocessing

The annotated dataset also requires some preprocessing. Since the annotated files are in XML format, they need to be converted to YOLO format labels to be compatible with the YOLOv8 model. Additionally, some unlabeled files must be filtered out, and a uniform file naming convention should be applied. Once completed, a TXT file will be generated, storing the image information obtained from this operation, including the annotation categories and bounding box coordinates. Through this annotation process, each image in the dataset is assigned accurate defect locations and classification labels, providing the necessary data foundation for training the subsequent object detection algorithms. Finally, the dataset is split into training, testing, and validation sets according to specified proportions.

3.3. Model Training

Given that the goal of this project is real-time detection on the production line, YOLO series deep learning algorithms are the most suitable for model training due to their renowned high detection speed. YOLOv8, as the latest version of the YOLO series, significantly enhances detection accuracy and efficiency through innovations such as the Dense Prediction Module and Soft-Gated Skip Connection. It employs multi-scale prediction and decoupled heads to improve small object detection capability, which is crucial for detecting defects like scratches and stains that are generally small targets.

Additionally, YOLOv8 optimizes the loss function and positive-negative sample allocation mechanisms, further improving performance. Its new backbone network and feature fusion methods, along with support for various hardware platforms and hyperparameter tuning, make it adaptable to a diverse range of complex application scenarios.

In the actual production line environment for small parts, hardware computing power and storage space are often limited, necessitating the use of lightweight models to avoid processing speed degradation or system stability issues due to hardware constraints. YOLOv8's backbone network is a multi-layer neural network composed of standard convolutions, which is relatively bulky and places high demands on computational hardware. Therefore, this project focuses on optimizing the YOLOv8 model by replacing its backbone network with a lightweight architecture.

Among various lightweight neural network designs, such as ShuffleNetV2, MobileNetV3, and GhostNet, ShuffleNetV2 has shown superior performance across multiple evaluation metrics relative to other lightweight networks in recent years[8]. ShuffleNetV2’s structure cleverly facilitates smooth information flow and effective feature retention, which is particularly important for enhancing the model's ability to recognize defects in varying lighting conditions and complex background environments. Consequently, this project has selected ShuffleNetV2 as the replacement for YOLOv8’s backbone network.

3.4. Model Evaluation and Optimization

After the model training is completed, comprehensive evaluation and optimization of the YOLOv8 model are crucial steps to ensure its efficient operation in detecting defects in production line parts. To thoroughly assess the performance of the YOLOv8 model in defect detection, this project selected several key metrics, including Precision, Recall, F1 Score, Mean Average Precision (mAP), and Frames Per Second (FPS). These metrics provide a comprehensive reflection of the model's detection accuracy, miss rate, overall performance, and real-time capability.

To objectively evaluate the model’s performance, this study used an independent test dataset consisting of 1,000 images that were not part of the model’s training or validation processes, ensuring the objectivity and reliability of the evaluation results. The improved YOLOv8 model demonstrated high detection accuracy on the test dataset, achieving an mAP value of 95%, indicating that the model can effectively identify various types of part defects. Additionally, the model’s detection speed reached 30 FPS, meeting the real-time detection requirements of the production line.

However, it was found that the model still has room for improvement. For instance, there is a need to further adjust the sample ratios of different defect categories in the training dataset to reduce category imbalance issues and enhance detection accuracy for minority categories. Moreover, integrating a more suitable attention mechanism into the YOLOv8 model could help the model focus more on key areas in the images, thereby improving the detection accuracy forsmall-object defects.

4. Conclusion

This project, based on the latest YOLOv8 deep learning algorithm, has conducted in-depth research and design for detecting defects in production line parts and achieved significant results. By constructing a diverse and rich dataset of part defects and performing detailed segmentation and preprocessing, we successfully trained and optimized the YOLOv8 model. Experimental results show that the improved model excels in both detection accuracy and speed, effectively addressing the issues of subjectivity, inconsistency, and inefficiency associated with traditional manual inspection methods. Additionally, the project explored model deployment and optimization strategies, using the lightweight neural network ShuffleNetV2 as the backbone for the YOLOv8 model, which further enhanced the model's applicability and stability in complex production line environments. The research results not only provide strong support for the intelligent technological transformation of the manufacturing industry but also lay a solid foundation for future applications in more industrial scenarios.

References

[1] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[2] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[3] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv:1804.02767, 2018.

[4] Bochkovskiy A, Wang C Y, Liao H Y M. Yolov4: Optimal speed and accuracy of object detection[J]. arXiv preprint arXiv:2004.10934, 2020.

[5] Liu Junhao, Wang Meilin, Xie Xing, Song Yexing, Xu Lihua. "Leather Defect Detection Algorithm Based on Improved YOLOv5" [J]. Computer Engineering, 2023, Vol. 49, No. 8: 240-249.

[6] Li B, Liu Y, Wang X. Gradient harmonized single-stage detector[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 8577-8584.

[7] Wang X, Gao H, Jia Z, et al. BL-YOLOV8: An improved road defect detection model based on YOLOV8[J]. Sensors, 2023, 23(20): 8361.

[8] Ma N, Zhang X, Zheng H T, et al. Shufflenet v2: Practical guidelines for efficient cnn architecture design[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 116-131.