

# High-precision Defect Detection of Glass for Thin-film Transistor Liquid Crystal Display Using YOLO Algorithms

Rihui Tan,<sup>1</sup> Chih-Cheng Chen,<sup>2\*</sup> Kai-An Kuo,<sup>2</sup> Ben-Yi Liao,<sup>2</sup>  
Ching-Kun Chen,<sup>2</sup> Shu-Han Liao,<sup>3</sup> Shi-Kai Guo,<sup>2</sup>  
Kai-Yi Tang,<sup>2</sup> Wei-Hong Lee,<sup>2</sup> and Suting Li<sup>1</sup>

<sup>1</sup>School of Computer Science, Guilin Tourism University, Guilin, China  
Guilin Tourism University, No. 26, Liangfeng Road, Yanshan District, Guilin, Guangxi 541006, China

<sup>2</sup>Department of Automatic Control Engineering, Feng Chia University, Taichung, Taiwan  
No. 100, Wenhua Rd., Xitun Dist., Taichung City 40732, Taiwan (R.O.C.)

<sup>3</sup>Department of Electrical and Computer Engineering, Tamkang University  
No. 151, Yingzhuan Rd., Tamsui Dist., New Taipei City 251301, Taiwan (R.O.C.)

(Received January 27, 2025; accepted May 16, 2025)

**Keywords:** TFT-LCD, glass defects, YOLOv4, object detection, quality control

We applied a deep learning technique to detect defects on the glass used in a thin-film transistor liquid crystal display (TFT-LCD) utilizing a You Only Look Once v4 (YOLOv4) object detection model. TFT-LCD glass defect detection is a critical quality control step in electronics manufacturing. Defects on the glass indicate serious problems in production. Manual inspections are often inefficient and inconsistent, highlighting the need for automated methods. To enhance efficiency and accuracy in the automated detection of defects on the TFT-LCD glass, convolutional neural networks (CNNs) were used. By optimizing and training the YOLOv4 model with a large labeled dataset, a highly efficient object detection method for multiple defects was developed. CNNs based on YOLOv4 showed superior performance in real-time detection and reduced defect detection time. Additionally, smart sensor CCD technology was employed to capture high-resolution images of glass surfaces for precise defect detection. The model leverages deep learning concepts such as feature extraction, data augmentation, and loss function optimization to improve performance. The developed YOLOv4 object detection model can be used for the quality control of automated TFT-LCD production and can help increase production efficiency and reduce defects of the final products.

## 1. Introduction

The thin-film transistor liquid crystal display (TFT-LCD) is indispensable for numerous electronic products including smartphones, televisions, monitors, and laptop computers as it provides clear and high-resolution visual experiences. Despite the rapid development of TFT-LCD production technology, defects on TFT-LCD glass in the manufacturing process are

---

\*Corresponding author: e-mail: [chenccheng@fcu.edu.tw](mailto:chenccheng@fcu.edu.tw)  
<https://doi.org/10.18494/SAM5574>

inevitable. Such defects reduce product quality and affect the lifespan and reliability of the final products. Therefore, defect detection in the glass of TFT-LCD is critical to ensuring product quality, leading to the demand for automated, intelligent solutions. Traditional defect detection methods required labor and much time and was prone to inconsistency. This problem can be addressed by applying advanced machine learning methods, particularly deep-learning-based object detection algorithms.<sup>(1–5)</sup>

You Only Look Once (YOLO) is a neural network known for exceptional real-time performance and accuracy. It uses a feature pyramid network (FPN) framework to enhance iterative convergence. YOLO has been applied in face recognition, autonomous vehicles, automated manufacturing, and surface defect detection.<sup>(6–9)</sup> YOLO can be tailored to various specific needs owing to its versatility and effectiveness.<sup>(10–14)</sup>

Pan *et al.* combined YOLOx-Plus and field-programmable gate arrays (FPGAs) to detect defects on printed circuit boards (PCBs) in real time.<sup>(15)</sup> The YOLOx algorithm was enhanced with PAN+FPN, SimAM, and SIoU modules to boost detection accuracy. Additionally, parameter quantization and an FPGA accelerator were incorporated to achieve faster optimization. This method allowed for the effective detection of PCB defects and addressed the shortcomings of previous methods.<sup>(15)</sup> Du *et al.* developed the refined scale-enhanced (RSE)-YOLO model for the detection, localization, and classification of defects in multilayer ceramic capacitor (MLCC) images.<sup>(16)</sup> The model was created using a residual coordinate weighted convolutional network (RCWCNet) to improve feature extraction and detection accuracy. To enhance feature fusion, the spatial attention pyramid pooling module (SAPPM) was introduced as SAPPM combines local and global information. The model employed a path-aggregated feature pyramid network (PAN) and efficient channel attention (ECA) for enhancing multiscale defect detection and generalization performance.<sup>(16)</sup> Zhang and Yin proposed an enhanced YOLOv5-based method for the defect detection of solar cells. They integrated deformable convolution into a constraint satisfaction problem (CSP) module to enable adaptive scale learning and perceptual domain sizes. The ECA-Net attention mechanism was also used to enhance the model's feature extraction. The network structure was refined by adding defect prediction heads to enhance the detection accuracy in multiple scales.<sup>(17)</sup> Ancha *et al.* used the 'mixed PCB defect detection dataset' (MDD\_PCB) and a YOLO model to detect multiple defects of PCB. They determined real-time inference on the Jetson Nano for more effective PCB defect detection.<sup>(18)</sup>

On the basis of such research results, we developed a YOLOv4 deep learning model to detect defects on the glass of TFT-LCD. Owing to its exceptional accuracy and real-time performance, YOLOv4 was applied to enhance detection accuracy, which contributes to increasing the efficiency of the automated production process. Our approach incorporates smart sensor CCD and advanced deep learning techniques such as multiscale detection, customized loss functions for class imbalance, and data augmentation to improve model robustness. The method reduces human errors, streamlines operations, and improves the accuracy and speed of detecting defects on the TFT-LCD glass. We developed datasets and a training process to optimize the model, which was evaluated through experiments and comparative analyses. The model can be used to enhance the quality and efficiency of TFT-LCD products.

## 2. Methods

### 2.1 Automated optical inspection (AOI)

Fine and minute scratches or cracks on glass cannot be detected accurately with ordinary lenses. To successfully detect scratches on the glass of TFT-LCD, a specialized optical inspection system is required. As the glass has high reflectivity and a rough surface, an internal coaxial light source is used to enhance the detection of topographical differences on the surface. Additionally, a line light source is used to accentuate the edges of test samples. In this study, the AOI method was used to inspect and examine the final products of TFT-LCD. AOI is widely used in electronic product manufacturing and semiconductor production, which demands high precision and efficiency in inspection. Cameras and lighting devices are used to detect defects in solder joints, components on PCBs, and the surfaces of semiconductor wafers. The AOI method is used to inspect the integration, placement, and quality of solder joints, the correct positions, orientations, and defects of the components of PCBs, and surface defects, cracks, scratches, or other imperfections of wafers. It also is used to assess the quality of texts, logos, and labels and measure the size of components on PCBs. The device used for the AOI method is shown in Fig. 1.

In the TFT-LCD manufacturing process, the mother glass or plain glass is used. It is a large transparent component, similar to silicon wafers. The glass surface must be meticulously clean and perfectly flat to be suitable for TFT-LCD applications. Two alkali-free glass substrates are dry-etched to produce red, blue, green, and black colors precisely on their surfaces as a color filter. We designed an AOI method for detecting the defects on the glass of TFT-LCD (Fig. 2). We analyzed the properties of the glass of TFT-LCD and determined conditions under which defects were produced. Then, an appropriate optical mechanism and an imaging method were decided on for defect detection. Cracks and imperfections were detected as defects as shown in Fig. 2. A line scan camera was used to obtain the cross-sectional profile and information in a positioning process.

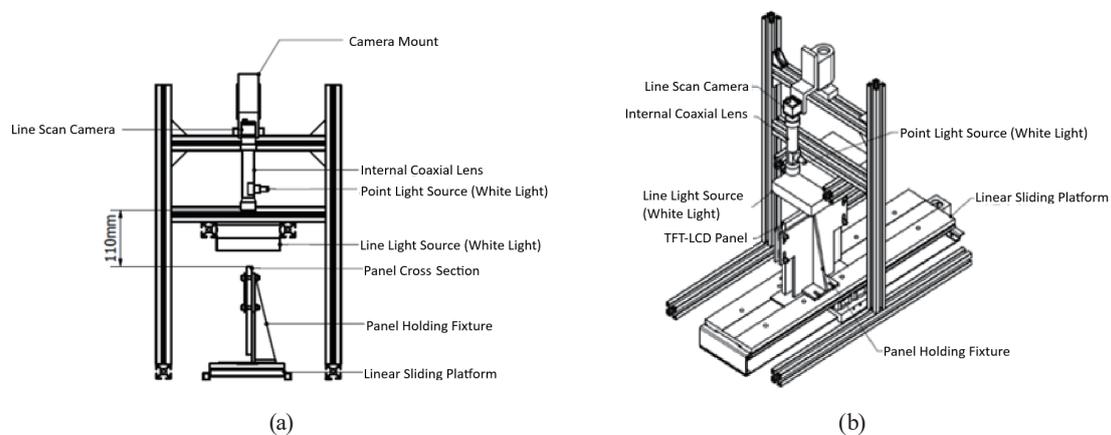


Fig. 1. Device for AOI method: (a) front view and (b) top view.

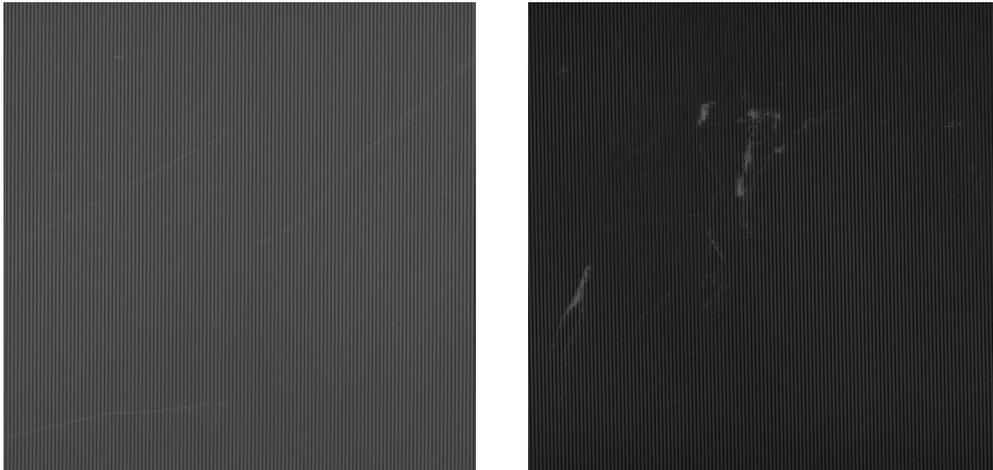


Fig. 2. Defects on glass of TFT-LCD.

## 2.2 YOLOv4 models

YOLO is a technique that enables computers to quickly identify objects within an image and determine their locations. The acronym YOLO stands for “You Only Look Once,” which signifies that the computer only needs to glance at an image once to complete both object detection and localization tasks. Unlike other object detection methods that may use multiple convolutional neural networks (CNNs), YOLO employs a single CNN, making it highly efficient in object detection.<sup>(18)</sup>

The YOLO method divides an image into a grid of smaller cells and analyzes each cell to determine if an object is present and where it is located. After comparing the features of the objects, YOLO can accurately identify the type and position of the objects within the image. The advantage of YOLO lies in its capability to perform object detection and localization in a very short time, making it suitable for real-time applications, such as autonomous driving. To use the YOLO model, one simply needs to prepare a labeled dataset with the objects to be recognized, and then train the model. Once training is completed, the model can be used to detect whether the desired objects are present in an image (Fig. 3).<sup>(2)</sup>

### YOLOv4 model<sup>(2)</sup>

#### a. Architecture (Fig. 3)

- (a) Input: Images, image pyramids, etc.
- (b) Head: This component predicts image features, generating bounding boxes (Bboxes) and predicted classes. It is divided into one-stage and two-stage architectures. As illustrated in the diagram, examples of a one-stage architecture include YOLO and SSD, whereas examples of a two-stage architecture include Faster R-CNN and R-FCN.
- (c) Backbone: The backbone network is typically pretrained. Commonly used backbone architectures include VGG16 and Darknet53. YOLOv4 utilizes CSPDarknet53 as its backbone. CSPDarknet53 is a variant of Darknet53, where CSP stands for “Cross-Stage Partial.” This network structure is designed to enhance model performance, particularly in object detection tasks.

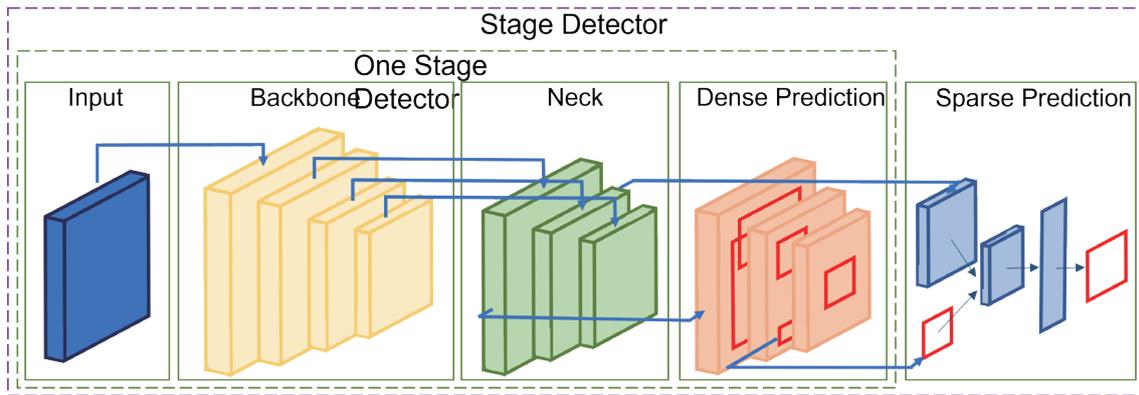


Fig. 3. (Color online) General framework for object detection.<sup>(2)</sup>

(d) Neck: Additional layers or blocks are used for feature extraction, such as the Feature Pyramid Network (FPN) and Path Aggregation Network (PANet). The choice of architecture is primarily based on the following three points:

- i. The input resolution of the network should be sufficiently high to better detect small objects.
- ii. The network should be deeper to cover a larger receptive field.
- iii. The network should have sufficient parameters, as more parameters improve the detection of objects of various sizes in the same image.

(e) Head: Same as YOLOv3 Head

The head of YOLOv3 is the core of the model, responsible for extracting features from the input image and generating object detection results, including object locations and class information.

(f) Structure:

- i. Convolutional Layers: The head of YOLOv3 includes multiple convolutional layers to extract features from the input image. These feature extraction layers convert the image into a representation suitable for object detection.
- ii. Detection Layer: This structure represents the most critical component of YOLOv3's head, as it is responsible for producing the final object detection outputs. The detection layer produces bounding boxes, object classes, and confidence scores. YOLOv3 typically contains multiple detection layers, each handling objects of different sizes.
- iii. Nonmaximum Suppression (NMS): To remove redundant detection results, YOLOv3's head includes a nonmaximum suppression layer. This process ensures that each object is reported only once and eliminates overlapping bounding boxes.

(g) Backbone: CSPDarknet53

CSPDarknet53 is a deep learning model based on the YOLOv3 backbone network Darknet53, incorporating features from CSPNet to produce the backbone architecture. It combines the efficiency of the Darknet architecture with the feature integration capabilities of the CSP mechanism, handling multiscale and complex image data, and enhancing the model's performance and versatility. It is mainly used for computer vision tasks, particularly for optimizing performance in object detection and image classification.

#### b. Features and Structure<sup>(2)</sup>

- (a) **Darknet Architecture:** CSPDarknet53 is based on the Darknet architecture, which is a lightweight and efficient deep learning framework. Darknet is an open-source neural network framework, and Darknet53 is a deep neural network consisting of 53 convolutional layers, used as the backbone network for many object detection models. CSPDarknet53 improves upon Darknet53 by introducing the CSP module to enhance performance. The CSP module divides feature maps into two parts, performs convolution operations on one part, and then merges the result with the other part. This design helps improve the information flow and gradient propagation within the model, thus enhancing training and inference efficiency.
- (b) **CSP Mechanism:** The core mechanism of CSPDarknet53 is the CSP mechanism. This mechanism introduces cross connections of feature maps to enhance feature information transfer and integration. It helps in better handling the information flow between different feature layers, thereby improving the model's performance.
- (c) **Multiscale Features:** CSPDarknet53 can handle features at different scales, which is particularly important for tasks such as object detection. Multiscale features enable the model to address targets of varying sizes and complexities effectively.
- (d) **Object Detection and Image Classification:** The model is primarily applied to object detection and image classification tasks. It can identify objects within images or classify images, and it demonstrates excellent performance in these tasks.

### 2.3 YOLO model evolution and architecture selection

In this study, we initially focused on the YOLOv4 architecture owing to its established balance between detection speed and accuracy, making it suitable for industrial applications. YOLOv4 integrates several key improvements over previous YOLO versions, including the CSPDarknet53 backbone, spatial pyramid pooling (SPP), path aggregation network (PANet) for feature fusion, and advanced training techniques such as mosaic data augmentation and CIOU loss.

As our research progressed, we also incorporated YOLOv7, a more recent iteration in the YOLO family released after the commencement of our study. YOLOv7 introduces several architectural enhancements over YOLOv4, including the following:

1. **Extended Efficient Layer Aggregation Network (E-ELAN):** An improved backbone structure that enhances gradient propagation and computational efficiency.
2. **Model scaling for concatenation-based models:** A novel scaling strategy that maintains an optimal balance between computational complexity and accuracy.
3. **Auxiliary head for training stabilization:** Additional network components that improve training stability and model convergence.
4. **Re-parameterized convolution for effective feature aggregation:** A technique that improves parameter efficiency without increasing inference time.

The inclusion of YOLOv7 in our analysis provides valuable insights into how architectural advancements in object detection models affect performance in the specific context of TFT-LCD glass defect detection. By comparing these two generations of YOLO models, we can better

understand the trade-offs between computational efficiency and detection accuracy in industrial inspection applications.

For our primary implementation and baseline experiments, we utilized YOLOv4 owing to its proven reliability and widespread adoption in industrial applications. YOLOv7 was subsequently introduced for comparative experiments to evaluate potential performance improvements and to assess whether the latest architectural innovations translate to significant benefits in our specific application domain.

## 2.4 Model training and implementation

The initial phase of our experimental methodology focused on implementing and optimizing the YOLOv4 architecture for TFT-LCD glass defect detection. As described in Sect. 2.2, we later extended our investigation to include YOLOv7 to evaluate potential performance improvements offered by its architectural advancements. Both models were subjected to the same training protocol and evaluation metrics to ensure a fair comparison.

## 3. Results and Discussion

### 3.1 Phase 1 training

The YOLOv7 model was trained on 116 images for 200 epochs. As shown in Fig. 4, the highest accuracy attained in defect detection was 77.22%. As the accuracy was not sufficiently high for the model, the results of phase 1 training were used as a reference for further training.

In Fig. 4, 'Box' indicates the average error in detection. The smaller the average error, the more accurately the defects are detected. 'Objectness' indicates the error related to object

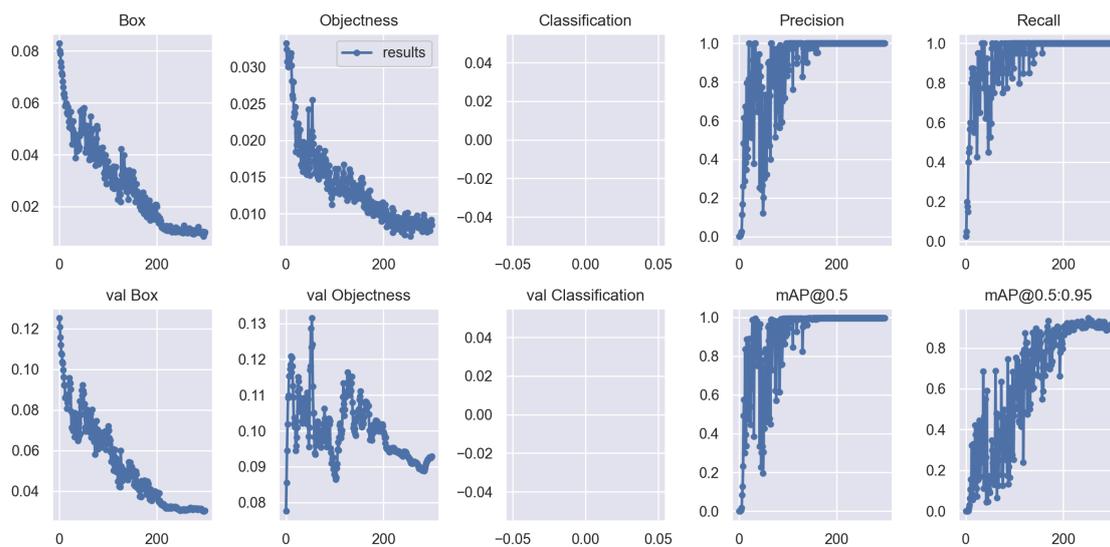


Fig. 4. (Color online) Phase 1 training results.

detection, where a smaller error indicates a higher detection accuracy. ‘Classification’ indicates the classification error. A smaller error indicates a more accurate classification. ‘val Box’ indicates a bounding box error on the validation dataset. ‘val Objectness’ shows the average object detection error on the validation set. ‘val Classification’ is the average classification error on the validation set. ‘Precision’ indicates the correctness of positive predictions, which is shown as the ratio of true positives to true and false positives. ‘Recall’ indicates the completeness of positive predictions, which is calculated as the ratio of true positives to true positives and false negatives. ‘mAP@0.5’ is the average mean average (mAP) greater than 0.5. ‘mAP@0.5:0.95’ is the mAP ranging from 0.5 to 0.95. Except for ‘val Classification’ and ‘Classification’, the x-axis indicates the number of epochs. There were no errors in the classification of defects using the test and validation datasets.

### 3.2 Phase 2 training

The number of training images was increased to 1000, and the YOLOv4 model was used. The maximum number of batches was set to 45000, corresponding to the 300 epochs. The result is shown in Fig. 5. The mAP was recalculated, and the accuracy was improved to 92.9%.

### 3.3 Phase 3 training

The YOLOv7 model was used with the same dataset as that used in the phase 2 training. The number of epochs remained at 300. The number of pixels of all images was changed to  $640 \times 640$ , and the multi-labeling method was adopted to label images. All images for training

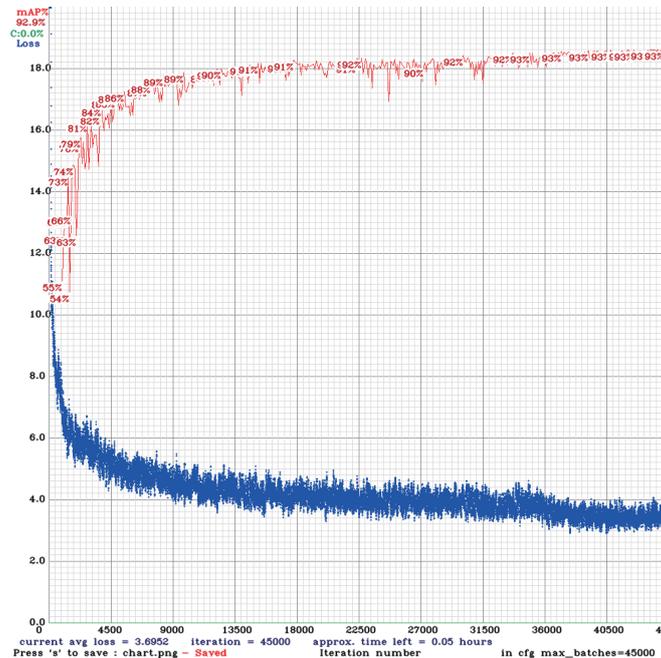


Fig. 5. (Color online) mAP of Phase 2 training results.

underwent grayscale processing. The confusion matrix and the training results are shown in Figs. 6 and 7. The accuracy reached 96.00%.

### 3.4 Phase 4 training

The YOLOv7 model was employed with mosaic data augmentation (Fig. 8), which is a deep learning training method to reorganize images with corresponding Bboxes. Through random

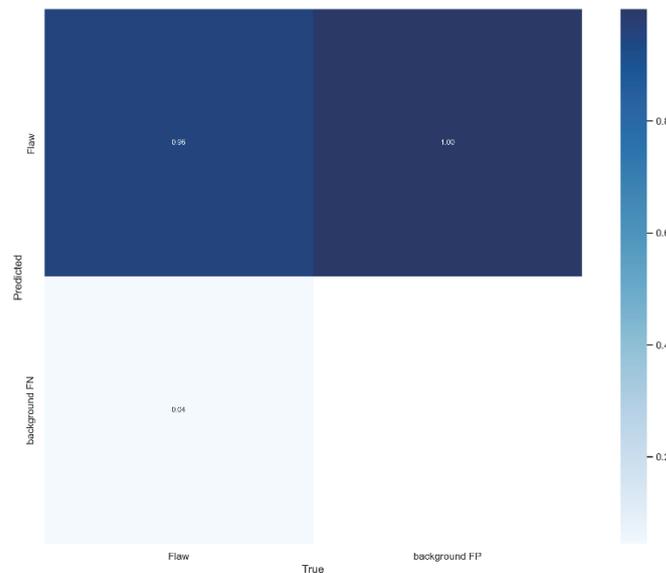


Fig. 6. (Color online) Confusion matrix of phase 3 training.

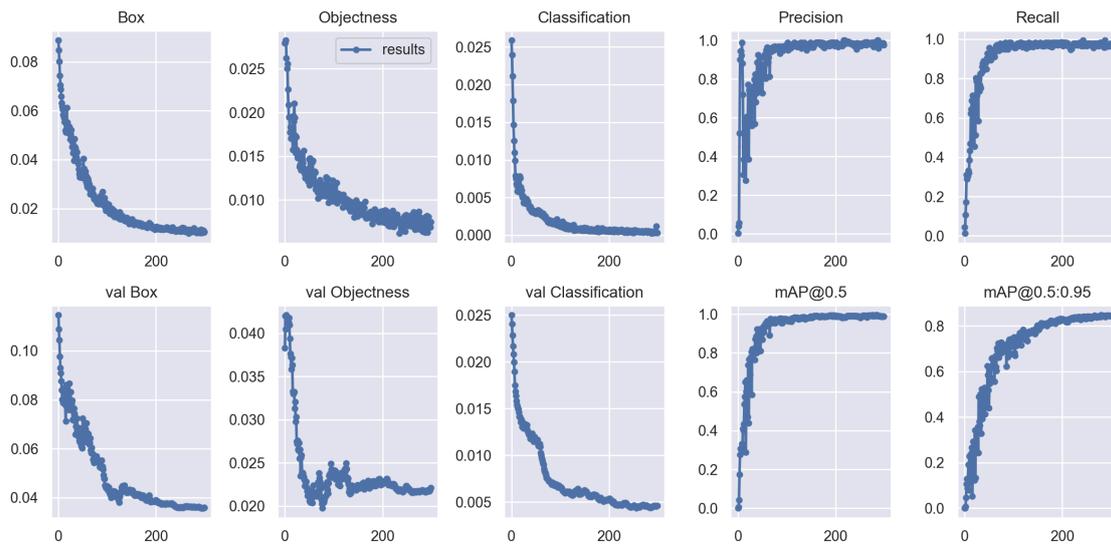


Fig. 7. (Color online) Results of phase 3 training.

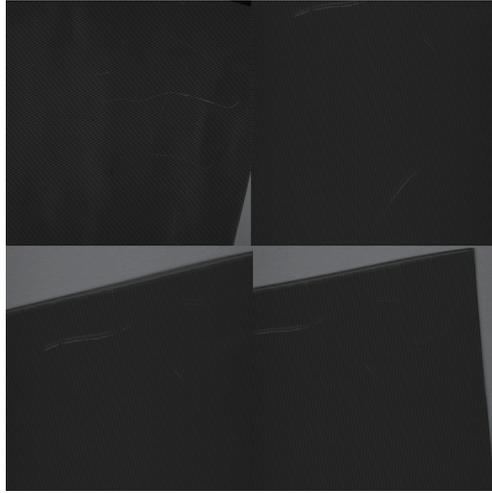


Fig. 8. Mosaic data augmentation.

Table 1  
Results of training.

Training rounds	Hours	Precision
Round 1	14	0.62
Round 2	13	0.68
Round 3	18	0.65

scaling, cropping, and arrangement, images are combined to emphasize the background. By reducing the batch size, we can reduce the burden on the graphical processing unit. The augmentation method enhances the model's training performance. The multilabeling method was also used for overlapped annotations to improve training effectiveness.

The accuracy calculated on the basis of the confusion matrix was only 72%, and similar results were obtained in multiple tests (Table 1). It was found that mosaic data augmentation was only effective in images with complex backgrounds and multiple objects. For the grayscale-processed images, mosaic data augmentation was not effective in training the YOLOv7 model but in training the YOLOv4 model.

### 3.5 Discussion

Although YOLO-based models are customized for the material properties and defect characteristics of specific applications, accurately detecting low-contrast defects remains a common challenge in various industrial settings. Table 2 summarizes and compares our proposed method with several recently published approaches.

Table 2  
Comparison of proposed method with three existing methods.

Aspect	PCB defect detection <sup>(15,18)</sup>	Solar cell defect detection <sup>(17)</sup>	Steel surface defect detection <sup>(4)</sup>	Our proposed method
Material properties	Opaque, fixed color, high contrast	Semitransparent, consistent optical properties	Opaque metal, uniform background	Small, transparent components
Defect characteristics	Pattern matching, lighting-insensitive	Contrast-based identification	Texture-based, consistent lighting	Small-area inspection, controlled lighting
Detection challenges	Standard imaging, fixed patterns	Back-illumination	Direct illumination	Specialized optical setups
Imaging technique	Template matching, pattern recognition	Contrast feature detection	Texture/pattern anomaly detection	Small-scale aberration detection
Model adaptations	PCB-specific YOLO adjustments	Solar-cell-specific YOLOv5 tuning steel	Steel-surface-specific YOLOv4 tuning	Optical-specialized WGSO-YOLO
Scale of inspection	Small components	Medium-sized cells	Large surfaces	Small, precise components
Production environment Constraints	Controlled lighting and positioning	Consistent conditions	Industrial settings	Laboratory-like setups
Model architecture	Low-contrast transparent defect detection	Low-contrast transparent defect detection	Low-contrast transparent defect detection	Low-contrast transparent defect detection
Key innovation	Low-contrast transparent defect detection	Low-contrast transparent defect detection	Low-contrast transparent defect detection	(FPN) / (PANet) used to enhance feature extraction

#### 4. Conclusions

Detecting defects on TFT-LCD glass has challenges as defects such as scratches and imperfections are tiny and are not easily contrasted against the surface. This necessitates deep learning models to identify defects in large background noise. On a highly reflective or glare-prone surface, detecting defects becomes challenging because the defects blend into the background. Additionally, defects on the glass can vary significantly in shape, size, and texture. Therefore, the detection model must be capable of distinguishing these diverse features accurately. To address these challenges, we developed an AOI method to replace conventional grayscale methods. By employing YOLO models, we enhanced training results and significantly improved the accuracy and effectiveness of defect detection.

#### Acknowledgments

This study was supported in part by the National Science and Technology Council (Grant nos. NSTC 113-2221-E-035 -047 -). This work was supported by the Guangxi Key Laboratory of Culture and Tourism Smart Technology at Guilin Tourism University, Guilin, China, under the project entitled “Research on the Innovation-Driven Mechanisms and Pathways of Guangxi’s Island Tourism under the Perspective of ‘Striving for Strength by Leveraging the Sea’” (Guangxi Philosophy and Social Science Planning Research Project: 24JYF019).

## References

- 1 Q. Ling and N. A. M. Isa: IEEE Access **11** (2023) 15921. <https://doi.org/10.1109/ACCESS.2023.3245093>
- 2 B. Alexey, C. Y. Wang, and H. Y. M. Liao: arXiv 10934 (2024) <https://doi.org/10.48550/arXiv.2004.10934>
- 3 Y. Li and J. Li: IEEE Trans. Instrum. Meas. **70** (2021) 1. <https://doi.org/10.1109/TIM.2021.3053987>
- 4 C. Li, A. Xu, Q. Zhang, and Y. Cai: IEEE Access **12** (2024) 37643. <https://doi.org/10.1109/ACCESS.2024.3374869>
- 5 M. Li, H. Wang, and Z. Wan: Comput. Electr. Eng. **102** (2022) 108208. <https://doi.org/10.1016/j.compeleceng.2022.108208>
- 6 W. L. Mao, C. C. Wang, P. H. Chou, and Y. T. Liu: IEEE Sens. J. **24** (2024) 26877. <https://doi.org/10.1109/JSEN.2024.3418618>
- 7 M. Yuan, Y. Zhou, X. Ren, H. Zhi, J. Zhang, and H. Chen: IEEE Trans. Instrum. Meas. **73** (2024) 1. <https://doi.org/10.1109/TIM.2024.3351241>
- 8 P. P. Shinde, P. P. Pai, and S. P. Adiga: IEEE Access **10** (2022) 39969. <https://doi.org/10.1109/ACCESS.2022.3166512>
- 9 L. Yao, B. Zhao, X. Wang, S. Mei, and Y. Chi: IEEE Access **12** (2024) 170227. <https://doi.org/10.1109/ACCESS.2024.3498004>
- 10 W. Yin, S. Lingxin, L. Maohuan, S. Qianlai, and L. Xiaosong: IEEE Access **11** (2023) 10966. <https://doi.org/10.1109/ACCESS.2023.3240894>
- 11 D. P. Carrasco, H. A. Rashwan, M. Á. García, and D. Puig: IEEE Access **11** (2023) 22430. <https://doi.org/10.1109/ACCESS.2021.3137638>
- 12 L. Liu, X. Feng, F. Li, Q. Xian, Z. Chen, and Z. Jia: IEEE Sens. J. **24** (2024) 23940. <https://doi.org/10.1109/JSEN.2024.3407874>
- 13 Q. Lin, K. Takamasu, and M. Chen: IEEE Trans. Instrum. Meas. **73** (2024) 1. <https://doi.org/10.1109/TIM.2024.3384267>
- 14 W. Zhou, C. Li, Z. Ye, Q. He, Z. Ming, J. Chen, F. Wan, and Z. Xiao: IEEE Trans. Instrum. Meas. **73** (2024) 1. <https://doi.org/10.1109/TIM.2024.3390198>
- 15 Y. Pan, L. Zhang, and Y. Zhang: IEEE Access **12** (2024) 61343. <https://doi.org/10.1109/ACCESS.2024.3387947>
- 16 M. Du, M. Chen, X. Cao, and K. Takamasu: IEEE Trans. Instrum. Meas. **73** (2024) 1. <https://doi.org/10.1109/TIM.2024.3375957>
- 17 M. Zhang and L. Yin: IEEE Access **10** (2022) 80804. <https://doi.org/10.1109/ACCESS.2022.3195901>
- 18 V. K. Ancha, F. N. Sibai, V. Gonuguntla, and R. Vaddi: IEEE Access **12** (2024) 100983. <https://doi.org/10.1109/ACCESS.2024.3430329>