EFFNet: Element-wise feature fusion network for defect detection of display panels

Feng He, Jiubin Tan, Weibo Wang, Shutian Liu, Yuemin Zhu, Zhengjun Liu

Highlights

EFFNet: Element-wise feature fusion network for defect detection of display panels

Feng He,Jiubin Tan,Weibo Wang,Shutian Liu,Yuemin Zhu,Zhengjun Liu

- A deep-learning-based method for real-time defect detection of display panels.
- An element-wise feature fusion module (EFFM) is proposed for the feature decoder.
- A comprehensive study of the proposed network and transfer learning strategy.
- The proposed model is highly efficient, effective, and robust to challenging objects.

EFFNet: Element-wise feature fusion network for defect detection of display panels

Feng He[a,b,d], Jiubin Tan[a,b], Weibo Wang[a,b,*], Shutian Liu[c], Yuemin Zhu[d,*] and Zhengjun Liu[c,*]

[a]Institute of Ultra-precision Optoelectronic Instrument Engineering, Harbin Institute of Technology, Harbin, 150080, China
[b]Key Lab of Ultra-precision Intelligent Instrumentation, Harbin Institute of Technology, Harbin, 150080, China
[c]School of Physics, Harbin Institute of Technology, Harbin, 150001, China
[d]CREATIS, INSA Lyon, CNRS UMR 5220, INSERM U1294, Universit de Lyon, Villeurbanne, 69621, France

ARTICLE INFO

Keywords:
Convolutional neural network
Deep learning
Semantic segmentation
Defect detection

ABSTRACT

Online (i.e., real-time) defect detection of display panels after array process is of paramount importance for quality control and yield rate improvement of products in display industry. However, owing to the limitation in feature representation, the performances of traditional defect detection methods are not satisfactory. This paper develops a novel element-wise feature fusion network (EFFNet) to solve the issue and achieve high-accuracy real-time defect detection of display panels. The method adopts a transfer learning and fine-tuning strategy for feature extraction layers and a decoder with relatively less computational complexity. Particularly, a feature fusion module based on element-wise addition of pyramid features is proposed in skip connection to improve detection efficiency and accuracy. Our method is compared with many state-of-the-art CNN-based models. Additionally, the effects of training dataset size, motion blur, and different backgrounds on the performance of the proposed method are investigated. Extensive experiments, including the ablation study, demonstrate that the developed network can accurately detect defects with complex textures, ambiguous boundaries and low contrast. It also has good robustness against motion blur. It outperforms state-of-the-art methods in terms of mIoU, mPA, and F1-Measure. Moreover, it is able to detect defects at speeds of up to 159 fps with input images of size 256×256.

1. Introduction

Display panels such as the thin-film-transistor liquid-crystal display (TFT-LCD) and organic light-emitting diode (OLED) are the main components of display products. However, the manufacturing process of these components is complicated and prone to suffer from different kinds of defects. The array process is the first stage of the entire manufacture. Its commonly occurring defects have received extensive attention due to their adverse impact on the yield, life span, and function of display panels. Therefore, the detection of defects that arose from the array process plays a vital role in the quality control and yield rate improvement of display panels in the actual production process. Generally, the defect inspection approach built on human vision is highly subjective, time-consuming, and labor-intensive. It hardly satisfies the increasing demand, i.e., high accuracy and speed (efficiency), for online defect detection (i.e., real-time analysis and detection of defects in an online or continuous manner during manufacturing). The inception of automatic computer-vision-based defect detection techniques brings a solution to the above issue.

In the last few decades, a vast number of image processing methods have been applied to defect detection, such as Otsu [1–3], Canny edge detector [4, 5], Fourier transform [6, 7], Gaussian and Gabor filters [8–10], optical flow [11], and support vector machine (SVM) [12]. For instance, defects in a glass substrate were detected by first establishing a straight-line interception histogram from the two-dimensional information of an image and then using Otsu criteria to find the best interception threshold [1]. Fourier transform was used to detect and localize small defects in nonperiodical pattern images [7]. The recognition of five types of steel surface defects was achieved by four types of statistical features and enhanced twin SVM [12]. However, the performances of these methods in real-time detection of defects with intricate textures still leave much to be desired. Intrinsically, the above methods mainly rely on handcrafted features and rules that are shallow in feature representation, such as gradient amplitude or local feature similarity. These properties make it difficult for them to effectively and integrally characterize target images, and they usually only perform well in combination with other techniques. Extrinsically, the majority of the features and the surrounding backgrounds of defects, such as the defects in display panels after the array process, are rather complicated.

Recently, deep-learning-based methods have made a huge impact on the field of computer vision [13–16] due to their automatic feature learning and superior feature representation abilities. Convolutional neural networks (CNNs), such as VGG [17] and ResNet [18], have been the most ground-breaking addition. For image processing [19–22], CNNs have overwhelming performance in object recognition compared to traditional non-deep-learning methods. Therefore, CNNs have been consecutively applied to defect detection in industrial production. For example, a U-shaped network [23] based on ResNet (U-ResNet) was proposed to

*Corresponding author
wwbhit@hit.edu.cn (W. Wang); yue-min.zhu@creatis.insa-lyon.fr (Y. Zhu); zjliu@hit.edu.cn (Z. Liu)
ORCID(s):

accurately detect conductive particles after anisotropic conductive film (ACF) bonding in the TFT-LCD manufacturing process. Faster-RCNN [24] was improved by introducing ResNet50 with feature pyramid networks as the backbone and was used for the detection of printed circuit boards. An end-to-end trainable deep convolutional neural network, DeepCrack [25] built on SegNet [26], was proposed for automatic crack detection. Encoder-Decoder Residual Network (EDRNet) [27] with the combination of deep supervision mechanism and fusion loss was developed to detect the surface defects of strip steel.

Figure 1: Typical hard-to-recognize defects in display panels. (a) The weak defect in the light-colored background. (b) The strong defect in the dark-colored background. (c) The malformation defect with an ambiguous boundary. The red marks indicate the regions of defects.

Despite CNN models having made considerable progress in defect detection, the problem of online real-time defect detection was still not comprehensively addressed, especially for the inspection of display panel production. First of all, deep learning is a data-driven technique and the performance of a CNN model is highly subjected to training data. Yet, it is both time-consuming and labor-intensive to obtain a training dataset of display panels with sufficient numbers and variety. On the other hand, the detection of defects in display panels faces many unsolved challenges: 1) low contrast between defect and surrounding background has two forms: weak defects in light-colored background (Fig. 1(a)) and strong defects in dark-colored background (Fig. 1(b)); 2) intricate and diversiform background noise in display panels; 3) malformation defects with ambiguous boundaries: these defects in display panels have an identical color with the surrounding defect-free area, which cannot be distinguished by intensity difference (Fig. 1(c)). These factors undoubtedly bring harsh requirements for the design of a fast and accurate CNN model.

Inspired by Attention U-Net [28] that shows an impressive performance in segmentation tasks even with a scarce amount of labeled training data, we proposed a new CNN model named element-wise feature fusion network (EFFNet) to overcome the aforementioned problems and achieve real-time defect detection of display panels with high accuracy. Considering the difficulty of gathering a dataset of display panel defects that is large enough, transfer learning strategy and data augmentation were used in our method.

The main contributions of this paper are listed as follows:

- A novel EFFNet model based on VGG16 and encoder-decoder architecture was developed to address the problem of real-time defect detection of display panels with multi-class backgrounds, where a transfer learning and fine-tuning strategy was proposed to speed up network training and increase detection performance.
- To the best of our knowledge, this is the first attempt to adopt a CNN model embedded with the additive attention mechanism to solve the problem of online defect detection of display panels after the array process.
- We developed a feature fusion module based on the element-wise addition of size-matched pyramid features. It highlights regions of interest (ROIs) on the same channel of feature maps between shallow and deep features, which exceedingly shortens detection time and improves the identification of defect pixels.

The rest of this article is organized as follows. In Section 2, related work is summarized. In Section 3, the proposed EFFNet is described in detail. Section 4 presents the experiments conducted to evaluate the performance of our method, including ablation study, comparison with state-of-the-art defect detection methods and other segmentation techniques, and discussion on the effects of training dataset size, motion blur, and backgrounds. Finally, Section 5 gives the conclusion.

2. Related work

As early outstanding works, non-deep-learning methods have been applied to achieve automatic defect detection of display panels. A mura defect detection method [29] was proposed for thin-film transistor liquid crystal display (TFT-LCD) panels. This method used the discrete cosine transform and the dual-γ piece-wise exponential transform for image prepossessing, followed by Otsu's method to segment the mura defects. Detection of mura defects on liquid crystal display (LCD) under uneven brightness [30] was later studied through the Gabor filter, the background reconstruction algorithm based on the mura uniform light principle, and the gamma correction.

More recently, deep-learning methods were widely studied. Lee et al. [31] adopted VGG16 to study the classification of defects in TFT-LCD panels. They used several methods including integrated gradients, SmoothGrad, Decovent, guided backpropagation, and deep Talyor to achieve the post hoc analysis of classification. The same year, an end-to-end multi-task learning network architecture [32] that contains an encoder, a feature fusion module, a segmentation head, and a classification head was proposed for the defect detection of mobile phone light guide plate (LGP). Furthermore, a multi-category classification model [33] based on the convolutional neural network working with automatic optical inspection (AOI) was proposed for identifying defective pixels on the TFT-LCD panel. The method used a special training strategy that worked with data augmentation to deal with the class-imbalanced problem caused by outnumbered non-defective pixels. Yao et al. [34] proposed an AYOLOv3-Tiny network in combination with an overlapping pooling

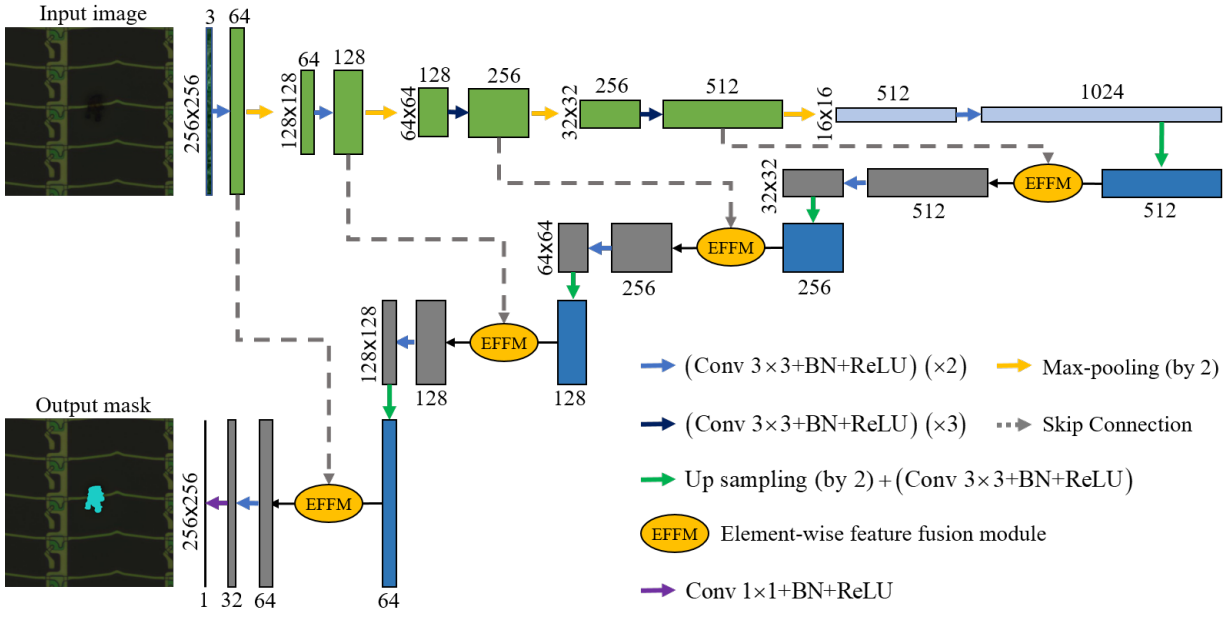

Figure 2: An illustration of the proposed EFFNet model.

spatial attention module (OSM) and a dilated convolution module (DCM) for the defect detection of LGPs. An improved RetinaNet that used ResNeXt50 as the backbone [35] was proposed for LGP defects detection, which adopted a Ghost module to replace the 1×1 convolution in the lower half of the ResNeXt block to reduce the resource parameters and consumption.

However, the above-mentioned works have mainly focused on the defects that occurred at the end of display panel manufacturing and ignored the detection of defects that occurred during the array process, which is also critical for yield improvement.

3. Proposed method

The overall architecture of our EFFNet model, as depicted in Fig. 2, is composed of an encoder and a decoder. The encoder, i.e., defect extraction module, is to extract defect features from complex background layer by layer and form a spatial feature pyramid structure that contains the semantic information of defects at different levels. The decoder embedded with element-wise feature fusion modules, on the other hand, gradually integrates the defect information from the encoded features. After the above processes, the segmentation results that indicate the localization of defects in display panels are obtained.

3.1. Defect extraction module

To mitigate small dataset over-fitting and shorten training time, we adopted the convolutional blocks (ConvBlocks) of VGG16[17] model pre-trained by ImageNet dataset as our defect extraction module. The module extracts a set of hierarchical features with different scales that encode multilevel contextual information from the input image. The detailed structure of the module that contains five ConvBlocks is given in Table 1. Each block consists of several 3×3 convolutional layers and one 2×2 max-pooling layer with a stride of 2 at the end (except the last block). Considering that the wider the network is, the more interactive cross-channel information the network will have, we replaced the last ConvBlock of VGG16 with two convolutional layers of 512 dimensions and 1024 dimensions to increase the network width and enhance the detection performance. Here, we use $\mathrm{E}_l(\cdot)$ to denotes the lth ConvBlock operation of the extraction module for simplicity, where $l \in \{1, 2, 3, 4, 5\}$. Thus, the extracted features f_l of the lth ConvBlock of the module are calculated as

$$f_l = \mathrm{E}_l\left(f_{l-1}\right) \tag{1}$$

where f_{l-1} is the output features of the $(l-1)$th ConvBlock and f_0 (i.e., $l = 1$) is the input image. $\mathrm{E}_l(\cdot)$ changes with l according to Table 1.

In particular, the max-pooling used in our network is to reduce the resolution of input images and expand the receptive field of the network while keeping translation invariance over small spatial shifts. Yet, continuous reduction of the resolution of feature maps would bring the loss of information related to boundary detail to some extent. Therefore, all the max-pooling indices were saved and used in the subsequent corresponding up-sampling layers to keep the integrality of defect representation. Furthermore, batch normalization was added after each convolutional layer to accelerate training and prevent over-fitting, followed by a rectified linear unit (ReLU) activation function to increase the nonlinearity. The feature channels were doubled every time after one block operation while the feature sizes were halved.

Table 1
Defect extraction module.

ConvBlock	Layer
1	[Conv 3×3 + BN + ReLU, C = 64] $\times$ 2 Maxpool 2×2
2	[Conv 3×3 + BN + ReLU, C = 128] $\times$ 2 Maxpool 2×2
3	[Conv 3×3 + BN + ReLU, C = 256] $\times$ 3 Maxpool 2×2
4	[Conv 3×3 + BN + ReLU, C = 512] $\times$ 3 Maxpool 2×2
5	Conv 3×3 + BN + ReLU, C = 512 Conv 3×3 + BN + ReLU, C = 1024

3.2. Feature decoder

As described in previous subsection, f_l ($l \in \{1,2,3,4,5\}$) is the output features of the lth ConvBlock of the encoder. In our decoder, the feature maps obtained from the fifth ConvBlock (i.e., $l = 5$) were first up-sampled to twice their original size by using the stored max-pooling indices from the forth layer of the encoder. Then, a 3×3 convolution operation was added to increase local contextual information and reduce the feature channels by half. The mathematical representation is formulated as

$$x_1 = \sigma_1\left(W_{3\times3} * U\left(f_5\right) + b\right) \tag{2}$$

where x_1 refers to the up-sampled high-level features of the first decoder layer and σ_1 is the ReLU activation function. $W_{3\times3}$ and b represent the 3×3 convolution kernel and bias, respectively. "$*$" denotes convolution operator and $U(\cdot)$ the up-sample operation with saved max-pooling indices and down-sample rate of the corresponding encoding layer, respectively.

Afterwards, an element-wise feature fusion module (EFFM) was developed to guide the skip connection between shallow and deep features, followed by two 3×3 convolutional layers that halve the feature channels. The mathematical representation is given by

$$f'_1 = \Phi\left(\wp\left(f_4, x_1\right)\right) \tag{3}$$

where f'_1 and $\Phi(\cdot)$ refer to the output of the first decoder layer and the convolution operation of two 3×3 convolutional layers, respectively. $\wp(\cdot)$ denotes the EFFM operator.

Similarly, the outputs of the other decoder layers are calculated as

$$f'_m = \Phi\left(\wp\left(\sigma_1\left(W'_{3\times3} * U\left(f'_{m-1}\right) + b\right), f_{5-m}\right)\right) \tag{4}$$

where f'_m ($m \in \{2,3,4\}$) represents the output of the ith decoder layer. $W'_{3\times3}$ designates the 3×3 convolution kernel that remains feature channels.

Finally, the segmentation result Y is computed as

$$Y = \sigma_2\left(W_{1\times1} * f'_4 + b\right) \tag{5}$$

where

$$\sigma_2(z) = \frac{1}{1+e^{-z}} \tag{6}$$

is the sigmoid activation function for normalizing the segmentation score. $W_{1\times1}$ denotes the 1×1 convolution kernel for decoupling the features and mapping them to lower-dimensional space.

3.3. Element-wise feature fusion module

The EFFM (i.e., element-wise feature fusion module) with an additive attention mechanism was developed with the inspiration of Attention U-Net [28] and is presented in Fig. 3. It used high-level features that have relatively more global semantic information as the gating signal to drive the network to focus more on the target pixels. Particularly, the up-sampled feature map $x_{l'}$ ($l' \in \{1,2,3,4\}$) from the decoder layer l' and the feature map f_l ($l < 5$) from the encoder layer l were first halved in channels by $W_{1\times1}$ convolution operation for spatial information extraction. The halved deep-level and shallow-level features were then combined through element-wise addition, followed by the ReLU activation function. The resulting features were passed to a $W_{1\times1}$ convolution layer again to generate a single-channel image-grid attention map. Finally, the Sigmoid function σ_2 was adopted to obtain the attention coefficients ξ_l. Mathematically, the attention coefficients for filtering the output features from the encoder layer l are computed as

$$\xi_l = \sigma_2\left(W_{1\times1} * \left(\sigma_1\left(W_{1\times1} * f_l \oplus W_{1\times1} * x_{l'} + b_1\right)\right) + b_2\right) \tag{7}$$

where $\oplus$ denotes the element-wise addition operator. b_1 and b_2 refer to biases.

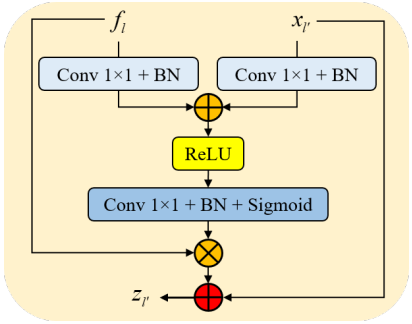

Figure 3: The architecture of element-wise feature fusion module.

The obtained attention coefficients were then used to filter the shallow features from the encoder by multiplication. Thereafter, the results were fused with the deep features that were expected to pass to the next layer of the decoder. Here, we used element-wise addition to achieve the fusion operation instead of channel-wise concatenation that was adopted in Attention U-Net. The key reason is that the former forces the feature fusion module to learn to underline the target regions (i.e., ROIs) on the same channel of feature maps between shallow and deep features during the

training, which avoids introducing extra parameters to the subsequent layers and thus significantly reducing computational complexity. Additionally, reusing coarse information through this improvement in skip connection can facilitate the network to learn more relevant information that improves detection performance. Mathematically, the EFFM output $z_{l'}$ of the l'th decoder layer can be formulated as

$$\begin{aligned} z_{l'} &= \wp\left(f_l, x_{l'}\right) \\ &= \xi_l \otimes f_l \oplus x_{l'} \end{aligned} \tag{8}$$

where $\otimes$ refers to the element-wise product operator.

It is worth mentioning that the attention coefficient map can be learned according to different levels of defect information during the training since the attention works in different layers of the decoder. Furthermore, the attention coefficients can filter the activations during both backward and forward passes, as formulated in Eq. 9

$$\begin{aligned} \frac{\partial z_{l'}^i}{\partial W} &= \frac{\partial\left(\xi_l^i \cdot f_l^i + x_{l'}^i\right)}{\partial W} \\ &= \xi_l^i \frac{\partial f_l^i}{\partial W} + \frac{\partial \xi_l^i}{\partial W} f_l^i + \frac{\partial x_{l'}^i}{\partial W} \end{aligned} \tag{9}$$

where $\xi_l^i \frac{\partial f_l^i}{\partial W}$ is scaled with ξ_l^i, which suppresses gradients derived from regions irrelevant to the target while encouraging the network to gradually learn more relevant features with the increase of the decoder layer.

3.4. Loss function

We updated the parameters of the proposed network by minimizing a pixel-wise loss function (binary cross-entropy)

$$L(g, p) = -\frac{1}{K}\sum_{i=1}^{K}\left[g_i \cdot \log\left(p_i\right) + \left(1 - g_i\right)\log\left(1 - p_i\right)\right] \tag{10}$$

where K is the batch size. g and $g_i \in \{0, 1\}$ are the ground truth mask and the ground truth of the ith image, respectively. p and $p_i \in [0, 1]$ are the network prediction and the predicted probability of the ith image, respectively.

4. Experimental settings

4.1. Dataset

The defect dataset of display panels contains 571 defective images of 1024×768 pixels. It was gathered through a microscope system. The images present six classes of backgrounds. Each of them has different types of defects, such as foreign object, film off, film adhesive, malformation, etc., as shown in Fig. 4.

The publicly available tool LabelMe was used to conduct the pixel-wise annotation to obtain the ground truth of our task. A dataset with a large number and variety is of utmost importance for the outstanding performance of the deep-learning method in defect detection. To meet this requirement and surmount over-fitting, we performed data augmentation on the defective images. The images were enhanced through random cropping (from 288×288 to 256×256), rotation (90^o, 180^o, and 270^o), horizontal and vertical flip, and brightness adjustment ($multiplication factor = [0.5, 1.5]$). The final distribution of the images in the training and testing sets is given in Table 2.

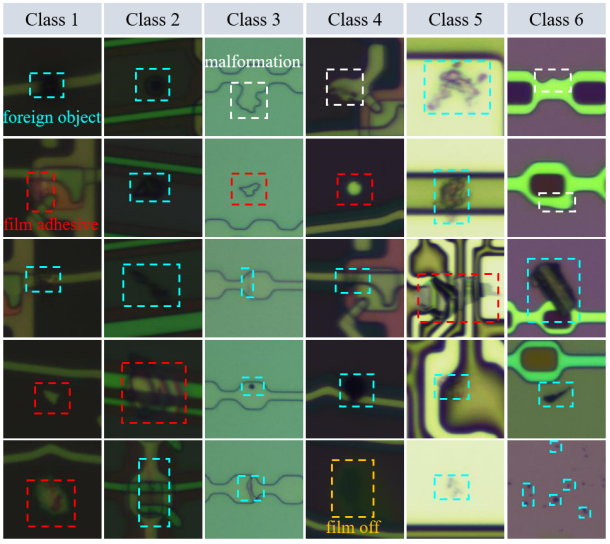

Figure 4: Different defects in six classes of backgrounds of display panels.

Table 2
Distribution of training and testing sets after data augmentation.

Classes	1	2	3	4	5	6	Total
Training	2109	2811	2304	2400	2304	2208	14136
Testing	91	93	144	96	112	108	644

4.2. Implementation details

All the experiments were implemented on the Pytorch framework using a single NVIDIA GTX 2080 Ti GPU (with 11G memory) on windows 10. The initialization parameters of the first four ConvBlocks of our network were loaded from the pre-trained VGG16 network, while the weights of other layers were initialized with the "Kaiming" initializer. For fine-tuning, the learning rates of the pre-trained layers were set to be 10^{-5}, and 10^{-4} was chosen for other layers. We used an Adam optimizer with a batch size of 12 images randomly cropped to 256×256 pixels from input images of 288×288 pixels to train our network.

4.3. Evaluation metrics

To evaluate the performance of our EFFNet model, the mIoU [36], mPA (mean pixel accuracy, Eq. 11), F-measure

(Eq. 12), and the precision-recall (P-R) curve were calculated between the ground truth and the predicted segmentation result.

$$mPA = \frac{1}{N}\sum_{i=0}^{N-1}\frac{p_{ii}}{\sum_{j=0}^{N-1} p_{ij}} \tag{11}$$

$$F-Measure = \frac{\left(1+\beta^2\right) Precision \times Recall}{\beta^2 \times Precision + Recall} \tag{12}$$

where $N = 2$ is the number of pixel classes with a value of 0 or 1. p_{ij} is the number of pixels whose predicted class is i while the ground truth is j. p_{ii} is the number of pixels whose predicted class and the ground truth are the same. β is a weight that weighs the importance between precision and recall. Here, we set $\beta = 1$, i.e., F1-Measure.

We also defined threshold-based detection rate (TDR) expressed as

$$\mathrm{TDR}_{\eta} = \frac{N_{\mathrm{mIoU}\geq\eta}}{N_{\mathrm{img}}} \times 100\% \tag{13}$$

where η designates the threshold of the detection rate, $N_{\mathrm{mIoU}\geq\eta}$ is the number of images whose mIoU is over η, and N_{img} denotes the number of input images. Notably, the acceptable threshold of mIoU is 0.5 in our task. Therefore, we chose 0.5 as the minimum mIoU threshold and took 0.05 as the interval to compute the TDR-mIoU curve to further evaluate the methods.

5. Results and discussion

5.1. Ablation study

This study aimed to verify the effectiveness of EFFM component and transfer learning strategy used in our method.

5.1.1. Effects of EFFM

To evaluate the proposed EFFM used in our method, we implemented a series of comparative experiments: 1) simple encoder-decoder architecture without skip connection: we named it as BaseNet (BN) for simplicity; 2) the BaseNet with skip connection based on simple element-wise addition: named as BN+EA; 3) the BaseNet with feature fusion module based on channel-wise concatenation (brought from Attention U-Net): simplified as BN+CFFM; 4) the BaseNet with the proposed EFFM: BN+EFFM (ours).

The evaluation results are provided in Table 3. Values in bold are the best results and this marking is applied to all the tables in this paper. It is shown that the EFFM favorably boosts the performance of our method in terms of mIoU and F1-Measure compared to simple encoder-decoder architecture BN and BN+EA. Moreover, compared with the channel-wise concatenation from Attention U-Net (i.e., BN+CFFM), the element-wise addition fusion method in EFFM not only significantly increases the network's detection efficiency (by 35 fps) but also improves its performance in mIoU and F1-Measure, which denotes the effectiveness of our refinement. The small decrease in mPA, as we conjecture, is due to the fact that the EFFM prefers to improve the mIoU and F1-Measure in a way that activates more target regions rather than trying to discredit more irrelevant pixels, which to some extent leads to an increase in false positives that is tolerable and negligible in our task.

Table 3
Evaluation results of ablation study in the effects of EFFM.

Method	mIoU	mPA	F1-Measure	Speed (fps)
BN	0.8108	0.8904	0.7311	**184**
BN+EA	0.8282	**0.9055**	0.7613	178
BN+CFFM	0.8364	0.9023	0.7775	124
BN+EFFM (ours)	**0.8377**	0.8941	**0.7867**	159

5.1.2. Transfer learning strategy analysis

Transfer learning is a training strategy that allows the knowledge (network parameters) learned from other tasks to adapt to a new task and thus avoid small dataset over-fitting and accelerate convergence. It can be achieved by either fine-tuning all the pre-trained layers or freezing some layers of them and only allowing the rest layers to be updated. Empirically, shallow layers in CNNs are prone to extract general features of input images, while deep layers prefer to identify the distinctive features of a specific task. Therefore, allowing deep pre-trained layers to be fine-tuned and keeping shallow layers untrainable may lead to better performance of a deep-learning model in a specific task. Inspired by the work of Samala et al. [37], we introduced and analyzed five potential layer-freezing schemes to find the best transfer learning strategy of our model for defect detection of display panels. We first froze all the pre-trained layers, then gradually released each convolutional block in a deep-to-shallow manner to allow it to be fine-tuned until released all the pre-trained layers. They were freezing all the pre-trained layers (Encoder-F4), freezing the first three ConvBlocks (Encoder-F3), freezing the first and second ConvBlocks (Encoder-F2), freezing the first ConvBlock (Encoder-F1), and not freezing any layer (Encoder-F0, namely ours).

Table 4
Evaluation results of our encoder with different transfer learning strategies.

Method	mIoU	mPA	F1-Measure
Encoder-F4	0.8347	0.8884	0.7829
Encoder-F3	0.8299	0.8901	0.7684
Encoder-F2	0.8360	0.8867	0.7810
Encoder-F1	0.8297	0.8852	0.7718
Encoder-F0 (ours)	**0.8377**	**0.8941**	**0.7867**

It can be seen from the comparison in Table 4 that, among all the five transfer learning strategies, the performance of Encoder-F0 is the best. It indicates that, in our

case, freezing the shallow layers (expected to extract general features of inputs) of the pre-trained extraction module does not increase the performance of the proposed model in display panel defect detection. In addition, it in a way prevents the network from learning more information relevant to the target. Hence, Encoder-F0 was selected for our method.

5.2. Comparison with state-of-the-art methods

To better demonstrate the superiority and better applicability of the proposed architecture in online defect detection of display panels, we compared our EFFNet with many state-of-the-art defect detection and object segmentation networks. They were Attention U-Net [28], MultiResUNet [38], U-ResNet [23], EDRNet [27], Huang et al. [39], UTNet [40], ComBiNetS [41], ComBiNetM [41], ComBiNetL [41], and DGNet [42].

Fig. 5 gives the learning curves of our EFFNet and other deep-learning models. It is observed that the proposed network converges faster than the other methods, which, as we infer, is due to the pre-trained layers in its encoder. Our EFFNet also has the lowest overall loss value and smoother loss curve when the loss is getting stable.

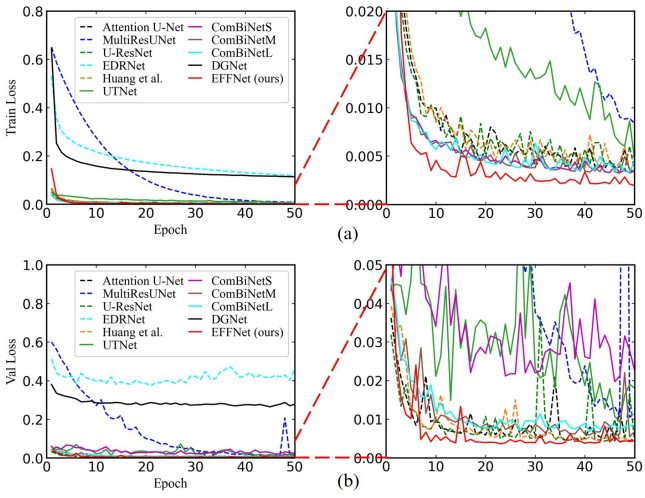

Figure 5: Learning curves of different methods. (a) Training loss. (b) Validation loss. The curves on the right are the enlargements of the corresponding areas of the left learning curves.

The quantitative evaluation results of the deep-learning methods are provided in Table 5. Clearly, our EFFNet outperforms other networks in terms of all the three evaluation metrics. Compared to the second-best results, our method is 1.57% higher in mIoU (vs. EDRNet), 2.36% higher in mPA (vs. Huang et al.), and 2.83% higher in F1-Measure (vs. UTNet). In addition, we studied the efficiency (i.e., detection speed) of our method and other models. The results are given in the last column of Table 5, which shows that our EFFNet achieves the fastest detection speed, 21 fps faster than the second-fastest model MultiResUNet for images with an input size of 256×256. The TDR-mIoU and P-R curves in Fig. 6(a) and Fig. 6(b) also show the superior performance of our method, from which we can see that our EFFNet has the biggest TDR over almost the whole mIoU thresholds and the highest precision over a large range of thresholds.

Table 5
Quantitative comparison of deep-learning methods.

Methods	mIoU	mPA	F1-Measure	Speed (fps)
Attention U-Net	0.8070	0.8646	0.7239	111
MultiResUNet	0.7907	0.8565	0.6729	138
U-ResNet	0.8121	0.8690	0.7290	83
EDRNet	0.8220	0.8697	0.7564	60
Huang et al.	0.8152	0.8705	0.7441	75
UTNet	0.8211	0.8607	0.7584	55
ComBiNetS	0.7362	0.7781	0.5910	30
ComBiNetM	0.7929	0.8237	0.7132	18
ComBiNetL	0.7991	0.8359	0.7198	16
DGNet	0.7960	0.8232	0.7226	98
EFFNet (ours)	**0.8377**	**0.8941**	**0.7867**	**159**

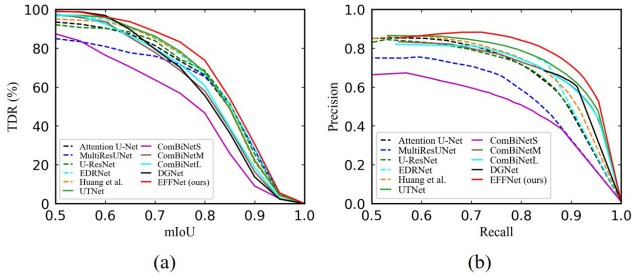

Figure 6: Evaluation curves of our EFFNet and other models. (a) TDR-mIoU curve. (b) P-R curve.

Fig. 7 shows the visual comparison of our EFFNet with other state-of-the-art methods for the segmentation of twelve typical defective images. We can observe that the other state-of-the-art networks fail to detect defects in many challenging cases. For instance, MultiResUNet, U-ResNet, and Huang et al. fail to find the tiny adhesive defect in the second column of class 1 and the malformation defect in the first column of class4. Attention U-Net, EDRNet, ComBiNetS, ComBiNetM, and DGNet fail to detect the low contrast region of the defect in the first column of class 5, which also occurs in U-ResNet and Huang et al.. For the image in the first column of class 3, all the other networks except EDRNet either mistakenly connect two tiny foreign objects as one defect or fail to identify the defect. Furthermore, all the other networks except UTNet and DGNet fail to locate the malformation defect in the first column of class 6. Our EFFNet, on the contrary, can detect the above defects with the results much closer to the corresponding ground truth. Notably, even though other methods can detect defects with high accuracy in many cases, their performance is still worse than that of the proposed network, as seen in the second column of classes 2-6.

In general, three factors may influence the performance of our EFFNet, i.e., the dataset size that decides the network parameters, the motion blur that occurs during the collection

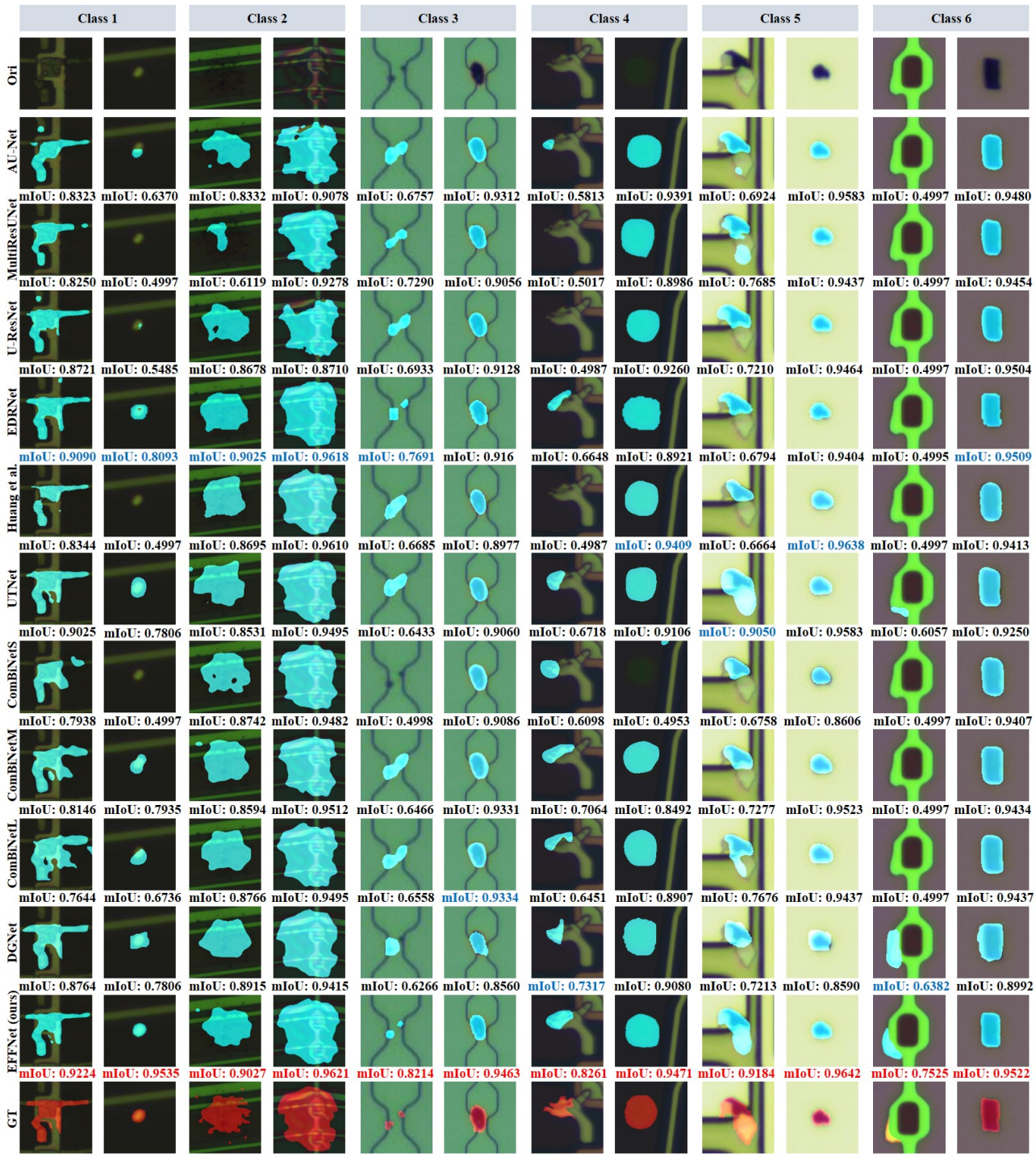

Figure 7: The visual comparison of our EFFNet with other state-of-the-art segmentation methods. Ori means original image, GT represents the ground truth. AU-Net is Attention U-Net. The values underneath the images are the corresponding mIoU of the detection, where the values in red are the best, and the ones in blue are the second best.

of defect images from the production line, and the intricate backgrounds around the defects. To investigate these factors and further validate the performance of our EFFNet, we implemented the following experiments.

5.2.1. Effects of dataset size

We set two groups of sub-training sets randomly extracted from the 100% training dataset at percentages of 20% and 60%, respectively, to train all the deep-learning methods. The evaluation results are given in Table 6. We can see that our network maintains the best performance against all the competitive methods in terms of the three metrics in

Table 6
Evaluation results of deep-learning methods in the study of effects of dataset size.

	60%			20%		
Methods	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure
Attention U-Net	0.7992	0.8569	0.7099	0.7647	0.8233	0.6323
MultiResUNet	0.7807	0.8457	0.6539	0.7387	0.7869	0.5773
U-ResNet	0.8052	0.8599	0.7141	0.7499	0.8139	0.5981
EDRNet	0.8214	0.8670	0.7559	0.8136	0.8638	0.7355
Huang et al.	0.8061	0.8647	0.7269	0.7638	0.8276	0.6406
UTNet	0.8045	0.8634	0.7276	0.6635	0.7457	0.4120
ComBiNetS	0.7262	0.7593	0.5689	0.6715	0.7063	0.4589
ComBiNetM	0.7837	0.8334	0.6939	0.7363	0.7835	0.5900
ComBiNetL	0.7882	0.8181	0.7045	0.7574	0.7966	0.6352
DGNet	0.7926	0.8179	0.7162	0.7834	0.8100	0.6981
EFFNet (ours)	**0.8350**	**0.8951**	**0.7763**	**0.8301**	**0.8868**	**0.7691**

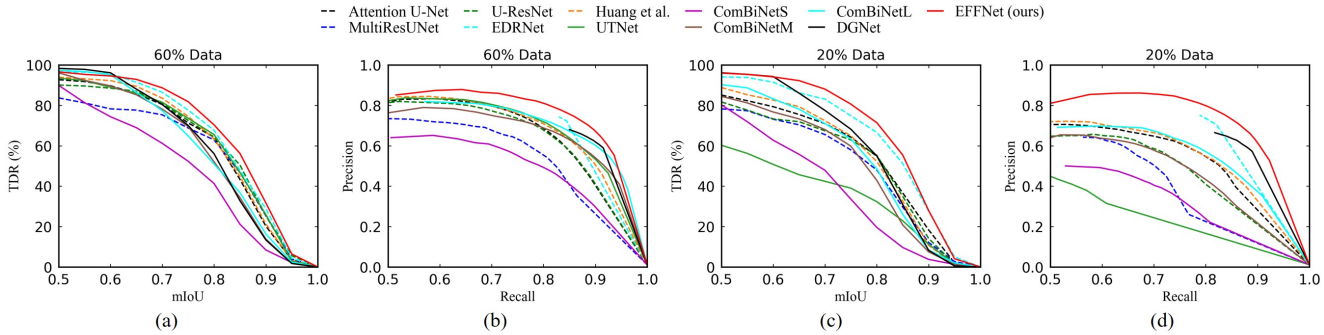

Figure 8: Evaluation curves of CNN-based methods trained with two sizes of datasets. (a) and (c) TDR-mIoU curve. (b) and (d) P-R curve.

both sub-dataset cases, although the performance drops with decreasing dataset size. It is fairly in conformity with the TDR-mIoU and P-R curves depicted in Fig. 8, where our EFFNet is always at the top right corner.

5.2.2. Robustness to motion blur

The defect images of display panels gathered from the production line are more or less subject to motion blur due to the relative movement between products and cameras. In order to study the robustness of our EFFNet to this noise, we added two different degrees (Low and High) of motion blur to the input images. The quantitative experimental results detailed in Table 7 show that our model has the best performance under low motion blur, and the performance is slightly lower than that of EDRNet under high motion blur. Fig. 9 presents the evaluation curves of the deep-learning methods in these two sets of experiments. It is observed that our method outperforms all the other methods

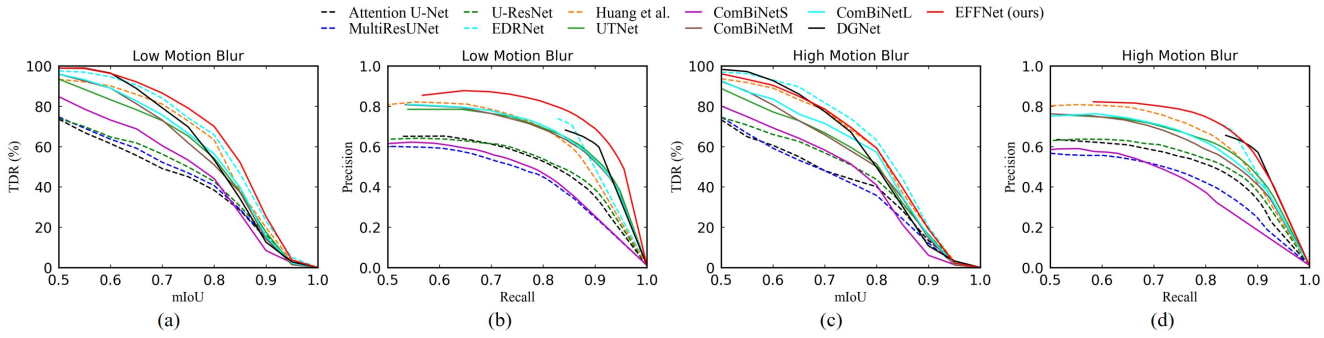

Figure 9: Evaluation curves of deep-learning methods in the robustness experiment. (a) and (c) TDR-mIoU curve. (b) and (d) P-R curve.

Table 7
Evaluation results of deep-learning methods in the study of robustness to motion blur.

	Low motion blur			High motion blur		
Methods	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure
Attention U-Net	0.6694	0.7473	0.5000	0.6646	0.7434	0.4895
MultiResUNet	0.7023	0.7561	0.5136	0.6825	0.7419	0.4806
U-ResNet	0.6756	0.7717	0.5346	0.6745	0.7747	0.5300
EDRNet	0.8161	0.8624	0.7488	**0.8067**	**0.8475**	**0.7335**
Huang et al.	0.7964	0.8504	0.7083	0.7870	0.8390	0.6916
UTNet	0.7719	0.8146	0.6635	0.7489	0.7904	0.6182
ComBiNetS	0.7256	0.7683	0.5672	0.7097	0.7555	0.5361
ComBiNetM	0.7752	0.8052	0.6766	0.7542	0.7861	0.6299
ComBiNetL	0.7848	0.8175	0.6899	0.7698	0.8047	0.6550
DGNet	0.7926	0.8210	0.7172	0.7818	0.8077	0.6956
EFFNet	**0.8276**	**0.8768**	**0.7705**	0.7943	0.8318	0.7079

in terms of the P-R curve under both noise cases, and it also has a higher TDR-mIoU curve under low motion blur. The visual comparison of our EFFNet and other methods in this study is shown in Fig. 10, where we can observe that, while the detection performance drops with the increase of motion blur, our EFFNet has the detection results closest to the ground truth under low motion blur. Furthermore, our EFFNet performs the best under high motion blur, except for some cases (such as the high motion blur case of the third sample image) poorer than EDRNet, compared to other methods.

5.2.3. Performance in different backgrounds

The images used in our experiments have six classes of backgrounds containing different colors and textures, as shown in Fig. 4. We conducted a series of experiments to study the effects of these different complex backgrounds on defect detection performance of the deep-learning methods. Table 8 summarizes the evaluation results of these methods in this study. We can observe that our model achieves the best performance in most of the classes, i.e., classes 1, 2, 4, and 6. It demonstrates that our EFFNet is more tolerant of complex background noise in display panels.

5.3. Analysis of failure cases

As analyzed in previous sections, the proposed network outperforms other state-of-the-art methods. However, there are still some challenging cases for our network and the competitive methods. Fig. 11 shows some examples of the failed segmentation results, which to some extent reveal certain drawbacks of our model.

First, the network misses some regions of interest where the contrast is mighty low, as shown in the first row of Fig. 11. Although these areas can be found by using a lower threshold (0.5 in our task) for the positive pixels, it will bring more false positives and reduce the accuracy. Furthermore, the network mistakenly connects some separate areas as one defect, such as the example in the second row. We infer that this phenomenon is derived from lacking a large training dataset having enough numbers of defective images with a wide variety, which is often necessary for the network to learn more information related to the boundary of defects. Finally, as observed in the last row, the model fail to detect the malformation defect with a tiny size. The conjecture is that, apart from the lack of a large training dataset, essential information related to this defect is lost during the down-sampling process due to its small size. To address these problems, we have to obtain a dataset with enough diversity and amount of images at first, which is a task for us in the future.

6. Conclusion

We proposed a novel EFFNet model for online defect detection that is of great importance for quality control and yield rate improvement of display panels. The modified ImageNet-trained VGG16 ConvBlocks and a fine-tune strategy were introduced for the encoder to extensively extract intricate defect features. Furthermore, an element-wise feature fusion module (i.e., EFFM) based on the additive attention mechanism was developed for our decoder to fuse multi-level features to enhance the detection accuracy while avoiding more computational complexity. Experimental results show that our method performs better than the state-of-the-art defect detection methods and other segmentation models. It also has good robustness against motion blur and small training dataset. Moreover, our model can detect defects at speeds acceptable for real-time defect detection.

Nevertheless, there are still some improvements to be achieved to put the developed network into practice. In future work, it would be interesting to design a more efficient data enhancement strategy to avoid the failure of the proposed method in more challenging cases and optimize the framework to achieve even higher accuracy and faster speed for the detection of different defects in display panels. In addition, validating our method using relevant public datasets (not yet available) is also an interesting complement.

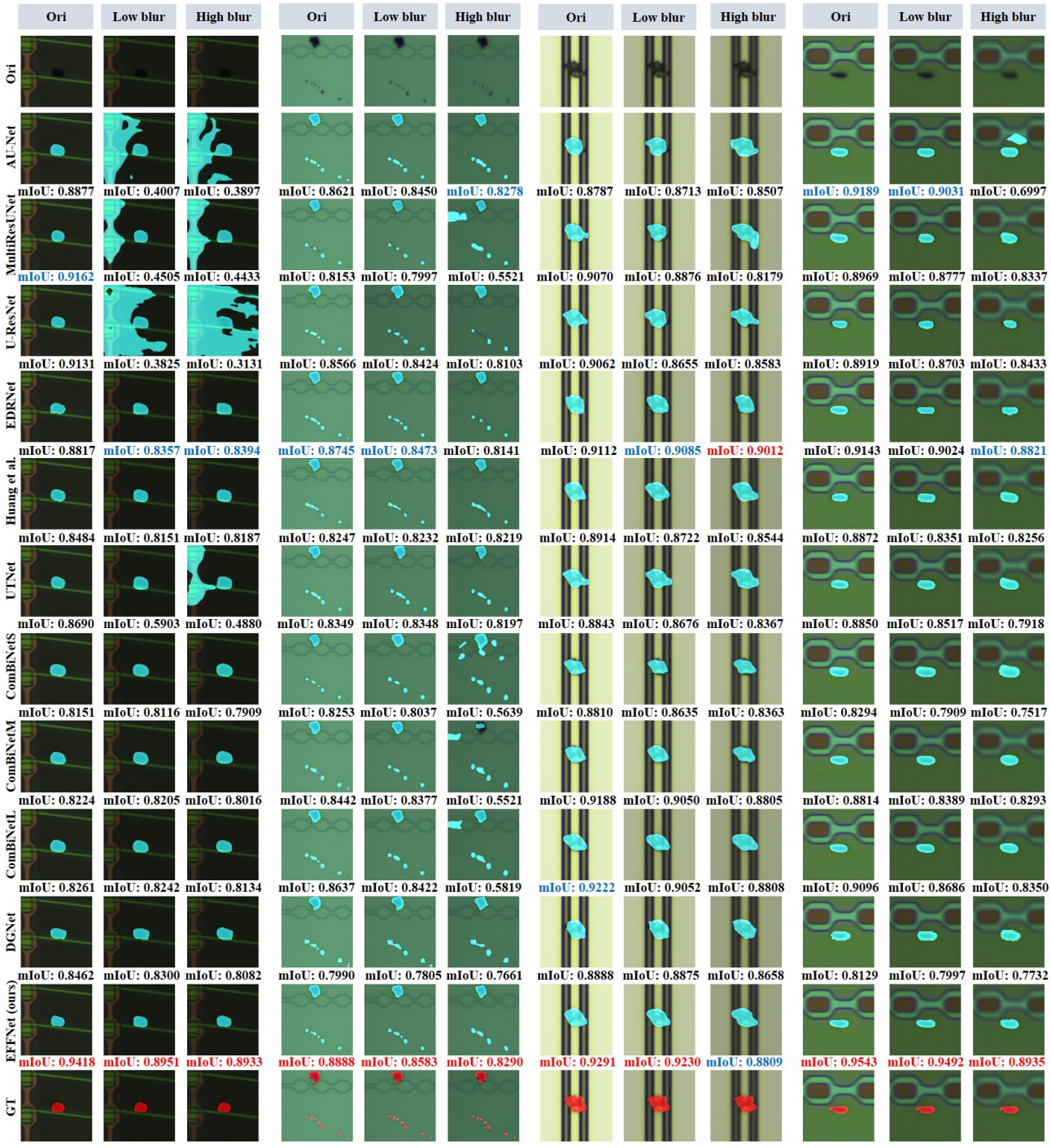

Figure 10: The visual comparison of different methods in the robustness analysis. Ori means original image, GT represents the ground truth. AU-Net is Attention U-Net. The values underneath the images are the corresponding mIoU of the detection, where the values in red are the best, and the ones in blue are the second best.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (under Grants 11874132, 61975044, 51775148, and 12074094).

Table 8
Evaluation results of deep-learning methods in the study of different backgrounds.

	Class 1			Class 2			Class 3		
Methods	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure
Attention U-Net	0.7776	0.8229	0.7043	0.8398	0.8686	0.7872	0.7958	0.9238	0.7071
MultiResUNet	0.7283	0.8310	0.5619	0.8554	0.8917	0.8014	0.7501	0.8598	0.5931
U-ResNet	0.7845	**0.8395**	0.7042	0.8524	0.8849	0.8066	0.7974	0.9196	0.7120
EDRNet	0.7866	0.8031	0.7140	0.8506	0.8664	0.8084	0.8049	0.9180	0.7339
Huang et al.	0.7776	0.8274	0.6984	0.8517	0.8703	0.8025	0.8059	0.9029	**0.7372**
UTNet	0.7741	0.7848	0.6955	0.8261	0.8377	0.7623	**0.8068**	0.8950	0.7354
ComBiNetS	0.6961	0.7412	0.5173	0.8197	0.8294	0.7784	0.7250	0.8055	0.5605
ComBiNetM	0.752	0.7706	0.6567	0.7877	0.8033	0.7333	0.7863	0.8251	0.6975
ComBiNetL	0.7302	0.7444	0.6067	0.8187	0.8245	0.7765	0.7826	0.8778	0.6819
DGNet	0.7500	0.7677	0.6417	0.8357	0.8551	0.7938	0.7843	0.8376	0.7046
EFFNet (ours)	**0.8108**	0.8282	**0.7567**	**0.8709**	**0.9043**	**0.8454**	0.8048	**0.9313**	0.7214
	Class 4			Class 5			Class 6		
Methods	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure	mIoU	mPA	F1-Measure
Attention U-Net	0.8015	0.8286	0.7047	0.8044	0.8657	0.7243	0.8175	0.8389	0.7092
MultiResUNet	0.8224	0.8466	0.7284	0.8087	0.8787	0.7322	0.7906	0.8286	0.6444
U-ResNet	0.8071	0.8144	0.7039	0.8091	0.8773	0.7314	0.8243	0.8478	0.7184
EDRNet	0.8049	0.8362	0.7110	**0.8469**	**0.8964**	**0.8092**	0.8465	0.8749	0.7741
Huang et al.	0.8372	0.8788	0.7832	0.8061	0.8800	0.7441	0.8157	0.8465	0.7062
UTNet	0.8437	0.8649	0.8094	0.8405	0.8913	0.7952	0.8307	0.8582	0.7435
ComBiNetS	0.7533	0.7702	0.6332	0.6945	0.7733	0.5120	0.7189	0.7345	0.5354
ComBiNetM	0.8276	0.8642	0.7750	0.7952	0.8347	0.7233	0.8086	0.8329	0.7025
ComBiNetL	0.8286	0.8483	0.7814	0.8006	0.8413	0.7314	0.8141	0.8414	0.7150
DGNet	0.8215	0.8363	0.7716	0.7739	0.8025	0.6937	0.8122	0.8279	0.7325
EFFNet (ours)	**0.8561**	**0.8794**	**0.8270**	0.8307	0.8934	0.7750	**0.8610**	**0.8986**	**0.8115**

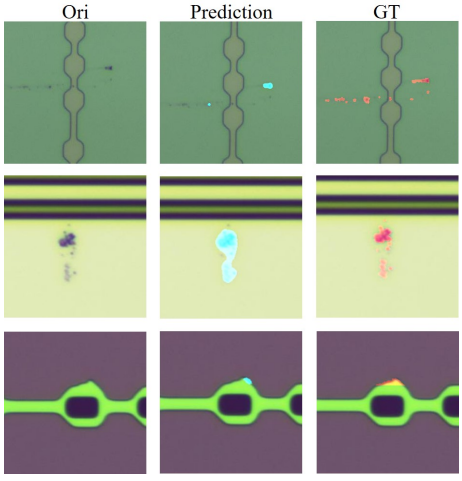

Figure 11: Failures of the proposed method.

References

[1] Z. He, L. Sun, Surface defect detection method for glass substrate using improved otsu segmentation, Applied optics 54 (2015) 9823–9830.

[2] X.-c. Yuan, L.-s. Wu, Q. Peng, An improved otsu method using the weighted object variance for defect detection, Applied surface science 349 (2015) 472–484.

[3] L. Meiju, Z. Rui, G. Xifeng, Z. Junrui, Application of improved otsu threshold segmentation algorithm in mobile phone screen defect detection, in: 2020 Chinese Control And Decision Conference (CCDC), IEEE, 2020, pp. 4919–4924.

[4] P. Wang, Z. Li, Y. Pei, In situ high temperature microwave microscope for nondestructive detection of surface and sub-surface defects, Optics Express 26 (2018) 9595–9606.

[5] G. Wang, W. T. Peter, M. Yuan, Automatic internal crack detection from a sequence of infrared images with a triple-threshold canny edge detector, Measurement Science and Technology 29 (2018) 025403.

[6] B. M. Barnes, M. Y. Sohn, F. Goasmat, H. Zhou, A. E. Vladár, R. M. Silver, A. Arceo, Three-dimensional deep sub-wavelength defect detection using λ= 193 nm optical microscopy, Opt. Express 21 (2013) 26219–26226.

[7] D.-M. Tsai, C.-K. Huang, Defect detection in electronic surfaces using template-based fourier image reconstruction, IEEE Transactions on Components, Packaging and Manufacturing Technology 9 (2018) 163–172.

[8] P. Li, H. Zhang, J. Jing, R. Li, J. Zhao, Fabric defect detection based on multi-scale wavelet transform and gaussian mixture model method, The Journal of The Textile Institute 106 (2015) 587–592.

[9] D.-C. Choi, Y.-J. Jeon, S. J. Lee, J. P. Yun, S. W. Kim, Algorithm for detecting seam cracks in steel plates using a gabor filter combination method, Appl. Opt. 53 (2014) 4865–4872.

[10] L. Tong, W. K. Wong, C. K. Kwong, Differential evolution-based optimal gabor filter model for fabric inspection, Neurocomputing 173 (2016) 1386–1401.

[11] X. Bai, Y. Fang, W. Lin, L. Wang, B.-F. Ju, Saliency-based defect detection in industrial images by using phase spectrum, IEEE Transactions on Industrial Informatics 10 (2014) 2135–2145.

[12] M. Chu, R. Gong, S. Gao, J. Zhao, Steel surface defects recognition based on multi-type statistical features and enhanced twin support vector machine, Chemomet. Intell. Lab. Syst. 171 (2017) 140–150.

[13] W. Ouyang, X. Zeng, X. Wang, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, H. Li, et al., Deepid-net: Object detection with deformable part based convolutional neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2016) 1320–1334.

[14] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 618–626.

[15] P. E. Carbonneau, S. J. Dugdale, T. P. Breckon, J. T. Dietrich, M. A. Fonstad, H. Miyamoto, A. S. Woodget, Adopting deep learning methods for airborne rgb fluvial scene classification, Remote Sens. Environ. 251 (2020) 112107.

[16] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, K. H. Maier-Hein, nnu-net: a self-configuring method for deep learning-based biomedical image segmentation, Nat. Methods 18 (2021) 203–211.

[17] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014).

[18] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[19] J. Lei, X. Gao, Z. Feng, H. Qiu, M. Song, Scale insensitive and focus driven mobile screen defect detection in industry, Neurocomputing 294 (2018) 72–81.

[20] C. Li, X. Wang, W. Liu, L. J. Latecki, B. Wang, J. Huang, Weakly supervised mitosis detection in breast histopathology images using concentric loss, Med. Image Anal. 53 (2019) 165–178.

[21] X. Le, J. Mei, H. Zhang, B. Zhou, J. Xi, A learning-based approach for surface defect detection using small image datasets, Neurocomputing 408 (2020) 112–120.

[22] L. Ruan, B. Gao, S. Wu, W. L. Woo, Deftectnet: Joint loss structured deep adversarial network for thermography defect detecting system, Neurocomputing 417 (2020) 441–457.

[23] E. Liu, K. Chen, Z. Xiang, J. Zhang, Conductive particle detection via deep learning for acf bonding in tft-lcd manufacturing, J. Intell. Manuf. 31 (2020) 1037–1049.

[24] B. Hu, J. Wang, Detection of pcb surface defects with improved faster-rcnn and feature pyramid network, IEEE Access 8 (2020) 108335–108345.

[25] Q. Zou, Z. Zhang, Q. Li, X. Qi, Q. Wang, S. Wang, Deepcrack: Learning hierarchical convolutional features for crack detection, IEEE Trans. Image Process. 28 (2018) 1498–1512.

[26] V. Badrinarayanan, A. Kendall, R. Cipolla, Segnet: A deep convolutional encoder-decoder architecture for image segmentation, IEEE Trans. Pattern Anal. Mach. Intell. 39 (2017) 2481–2495.

[27] G. Song, K. Song, Y. Yan, Edrnet: Encoder–decoder residual network for salient object detection of strip steel surface defects, IEEE Trans. Instrum. Meas. 69 (2020) 9709–9719.

[28] J. Schlemper, O. Oktay, M. Schaap, M. Heinrich, B. Kainz, B. Glocker, D. Rueckert, Attention gated networks: Learning to leverage salient regions in medical images, Med. Image Anal. 53 (2019) 197–207.

[29] S. Jin, C. Ji, C. Yan, J. Xing, Tft-lcd mura defect detection using dct and the dual-γ piecewise exponential transform, Precis. Eng. 54 (2018) 371–378.

[30] Z. Ma, J. Gong, An automatic detection method of mura defects for liquid crystal display, in: 2019 Chinese Control Conference (CCC), IEEE, 2019, pp. 7722–7727.

[31] M. Lee, J. Jeon, H. Lee, Explainable ai for domain experts: a post hoc analysis of deep learning for defect classification of tft–lcd panels, J. Intell. Manuf. 33 (2022) 1747–1759.

[32] Y. Li, J. Li, An end-to-end defect detection method for mobile phone light guide plate via multitask learning, IEEE Trans. Instrum. Meas. 70 (2021) 1–13.

[33] Y.-C. Chang, K.-H. Chang, H.-M. Meng, H.-C. Chiu, A novel multicategory defect detection method based on the convolutional neural network method for tft-lcd panels, Math. Probl. Eng. 2022 (2022).

[34] J. Yao, J. Li, Ayolov3-tiny: An improved convolutional neural network architecture for real-time defect detection of pad light guide plates, Comput. Ind. 136 (2022) 103588.

[35] J. Li, H. Wang, Surface defect detection of vehicle light guide plates based on an improved retinanet, Meas. Sci. Technol. 33 (2022) 045401.

[36] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.

[37] R. K. Samala, H.-P. Chan, L. Hadjiiski, M. A. Helvie, C. D. Richter, K. H. Cha, Breast cancer diagnosis in digital breast tomosynthesis: effects of training sample size on multi-stage transfer learning using deep neural nets, IEEE transactions on medical imaging 38 (2018) 686–696.

[38] N. Ibtehaz, M. S. Rahman, Multiresunet: Rethinking the u-net architecture for multimodal biomedical image segmentation, Neural Netw. 121 (2020) 74–87.

[39] Y. Huang, J. Jing, Z. Wang, Fabric defect segmentation method based on deep learning, IEEE Transactions on Instrumentation and Measurement 70 (2021) 1–15.

[40] Y. Gao, M. Zhou, D. N. Metaxas, Utnet: a hybrid transformer architecture for medical image segmentation, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part III 24, Springer, 2021, pp. 61–71.

[41] M. Ferianc, D. Manocha, H. Fan, M. Rodrigues, Combinet: Compact convolutional bayesian neural network for image segmentation, in: Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part III 30, Springer, 2021, pp. 483–494.

[42] G.-P. Ji, D.-P. Fan, Y.-C. Chou, D. Dai, A. Liniger, L. Van Gool, Deep gradient learning for efficient camouflaged object detection, Machine Intelligence Research 20 (2023) 92–108.