Received: 12 August 2020 | Revised: 2 January 2021 | Accepted: 11 January 2021

DOI: 10.1002/pip.3395

RESEARCH ARTICLE

PROGRESS IN PHOTOVOLTAICS WILEY

Deep learning-based automatic detection of multitype defects in photovoltaic modules and application in real production line

Yang Zhao[1] | Ke Zhan[2] | Zhen Wang[2] | Wenzhong Shen[1,3]

[1]Institute of Solar Energy and Key Laboratory of Artificial Structures and Quantum Control (Ministry of Education), Department of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai, China

[2]AI Department, Shanghai Optech Science and Technology Co., Ltd, Shanghai, China

[3]Collaborative Innovation Center of Advanced Microstructures, Nanjing, China

Correspondence
Wenzhong Shen, Institute of Solar Energy and Key Laboratory of Artificial Structures and Quantum Control (Ministry of Education), Department of Physics and Astronomy, Shanghai Jiao Tong University, Shanghai 200240, China.
Email: wzshen@sjtu.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Numbers: 11974242, 11834011

Abstract

Automatic defect detection in electroluminescence (EL) images of photovoltaic (PV) modules in production line remains as a challenge to replace time-consuming and expensive human inspection and improve capacity. This paper presents a deep learning-based automatic detection of multitype defects to fulfill inspection requirements of production line. At first, a database composed of 5983 labeled EL images of defective PV modules is built, and 19 types of identified defects are introduced. Next, a convolutional neural network is trained on top-14 defects, and the best model is selected and tested, achieving 70.2% mAP_{50} (mean average precision with at least 50% localization accuracy). Then, through analyzing an object detection-based confusion matrix, recognition bias and detection compensation in missed defects that restrain the best model's mAP_{50} are discovered to be harmless to normal/defective module classification in real production line. Finally, after setting specific screen criteria for different types of defects, normal/defective module classification is conducted on additionally collected 4791 EL images of PV modules on 3 days, and the best model achieves balanced scores of 95.1%, 96.0%, and 97.3%, respectively. As a result, this method surely has a highly promising potential to be adopted in real production line.

KEYWORDS
automatic defect detection, convolutional neural network, deep learning, electroluminescence, photovoltaic module

1 | INTRODUCTION

Renewable energies have become an irreversible trend for future power supply. Besides wind and water energy, another one of the most important and promising technologies is solar energy, which supplies around 2% of the world's total energy demand today and is a proven technology to be deployed to a multi-terawatt scale by 2030.[1] A photovoltaic (PV) cell is the basic unit of converting solar energy to electricity, and a number of them are concatenated to form a PV module through some processing stages in production line. During the solar cell production and processing stages, various defects like weak soldering, finger interruption, and crack can be generated due to incorrect manipulations such as deficient soldering, screen printing error, and collision. Among all the types of these defects, part of them hinder the current flow, decrease the module power, and even damage the whole module while others may not infect the module efficiency but the quality grade. In this context, all of them should be inspected carefully in the production process to ensure the efficiency, security, and quality of the PV modules.

One of the most commonly used approaches to detect defects in PV modules is electroluminescence (EL) imaging due to its high resolution to characterize different types of defects.[2,3] But manually inspecting EL images requires well-trained professionals to keep staring at the screen all the time as the line moves. It is not only time-consuming but also expensive and the inspection accuracy may fluctuate due to the boring and mechanical repetition of the

inspection process. Hence, many researches aiming to achieve automatic detection of defects in EL images have been done in the past decade. These studies can be divided into two groups according to their approaches: using conventional signal processing algorithms[4–8] and using artificial intelligence (AI) techniques.[9–16]

Tsai et al.[4] proposed a technique based on independent component analysis to detect the presence/absence of crack, break, and finger interruption in PV modules but cannot distinguish them. Anisotropic diffusion filtering,[5] matched filtering,[6] and vesselness filtering[7] had been adopted to focus on detection of cracks in EL images. Tseng et al.[8] established a method based on binary clustering of features to detect the finger interruption in EL images. Actually, since different types of defects vary considerably in their appearances, using single image processing approach can hardly deal with all of them and this is why many researchers turned into AI techniques that developed rapidly and proved superiority by successful applications[17–21] in various research areas in recent years. Deitsch et al.[9] utilized support vector machine and convolutional neural networks (CNN) to predict the possibility that a solar cell has power loss. This work is one of the pioneering attempts to introduce machine learning and deep learning methods into EL inspection, but it did not concern specific defect types. Mayr et al.[10] used weakly supervised learning based on ResNet50 to segment cracks in EL images and proved effective especially when the data size is limited. Akram et al.[11] introduced five types of defects and did normal/defective classification of cells, but they could not detect and localize specific defects. Tang et al.[12] combined generative adversarial networks with data augmentation to generate more training data and achieved good classification accuracies for images of defect-free (84%), micro-crack (82%), finger-interruption (81%), and break (83%). But there is a specificity that each of their images only carried one type of defects while the practical situation is that different types of defects may exist in the same cell and module in production line.

The purpose of automatic detection is to replace the manual inspection in production line, and it has two requirements: (1) different types of defects should be concerned, and (2) every single defect should be localized and classified, which is essentially an object detection task. But current researches only studied crack, break, and finger interruption[4–16] and cannot handle localization problem well for multitype defects, which is what we aimed to achieve in this paper. We sum up our main contributions as follows: (1) we gathered 5983 EL images of defective modules and labeled all of them, with 19 - categories of defects found and introduced. To our knowledge, this is the first time that so many EL images of defective modules were collected and labeled as well as so many types of defects were introduced in EL inspection. (2) We adopted an existing mask region based CNN (Mask R-CNN) with ResNet-101-FPN backbone model[22] in Detectron2[23] platform to do object detection task on the top-14 types of defects and used the COCO metrics[24] including average precision (AP, averaged over different intersection over union (IoU) thresholds), AP_{50} (AP at IoU threshold = 50%), and AP_{75} (AP at IoU threshold = 75%) to assess the detection results for each type. As far as we know, this is the first time that AP was used in object detection task of EL inspection. (3) Furthermore, to apply the method in real production line, we further calculated and analyzed an object detection-based confusion matrix, and we also investigated detection requirements for each type of defects of production line, set corresponding screening criteria, and conducted normal/defective classification on additional 4791 EL images of PV modules. For all we know, this is the first time that defect screening criteria and automatic detection in production line are involved in literature.

2 | OUR DATASET AND DEFECT TYPES

We discontinuously gathered 5983 EL images of defective nine-busbar (9BB) 6 × 24-half-cell Czochralski (Cz) grown monocrystalline Si modules in a Chinese PV module production plant from January to April, 2020. The company's name is not disclosed here because of nondisclosure agreement. These images were taken by an OPT-M960 EL machine produced by Suzhou Optech New Energy Technology Co., Ltd, with 45 V, 8A, exposure time of 1000 ms and gain factor of 1 set. These EL images were labeled by four well-trained people and an experienced leader using Labelme software, and then the generated annotation files were converted to COCO data format by code to fulfill training demand. Figure 1 illustrates these types of defects we identified, together with the labeled numbers listed in the parentheses. Different types of defects were named according to their various appearances or causes, but the causes for a certain type of defects could be very complicated so we only state major ones here.

These 19 types of defect samples were cropped from different EL images of modules, and we named them as follows: (1) a weak soldering is typically a dark rectangle, extending symmetrically from a busbar. It is usually caused by deficient soldering of the belt on the busbar so it blocks collection of current flow and lowers module efficiency. (2) A black area is an irregular dark region that usually implies silicon material problems like temperature inhomogeneity during firing process, being polluted or lack of minor carriers. In this paper, they were all named as black area because of similarity in their appearances. (3) A scratch is a snatchy dark line with uneven thickness on the surface of a cell. It does no harm to the cell efficiency but influence module appearance and quality grade, so it should also be detected. (4) A finger interruption is a single vertical dark line typically between two neighbor busbars and it is usually caused by screen printing errors. Due to the resolution limitation of our EL camera, the fingers cannot be displayed. (5) A crack is a sharp dark line with even thickness compared to a scratch but sometimes they can look quite like each other and even a professional can hardly distinguish them. A crack is usually caused by thermo-mechanical stresses like collision. (6) A low cell has lower efficiency than other normal cells and appears darker. Cells with sever efficiency difference should not be matched in the same module for power and security considerations. (7) A finger block is a bunch of neighbor finger interruptions. It was differentiated from finger interruption because of its bigger effect on module efficiency so that production line has different inspection requirements for them. (8) The EL images of modules were segmented into

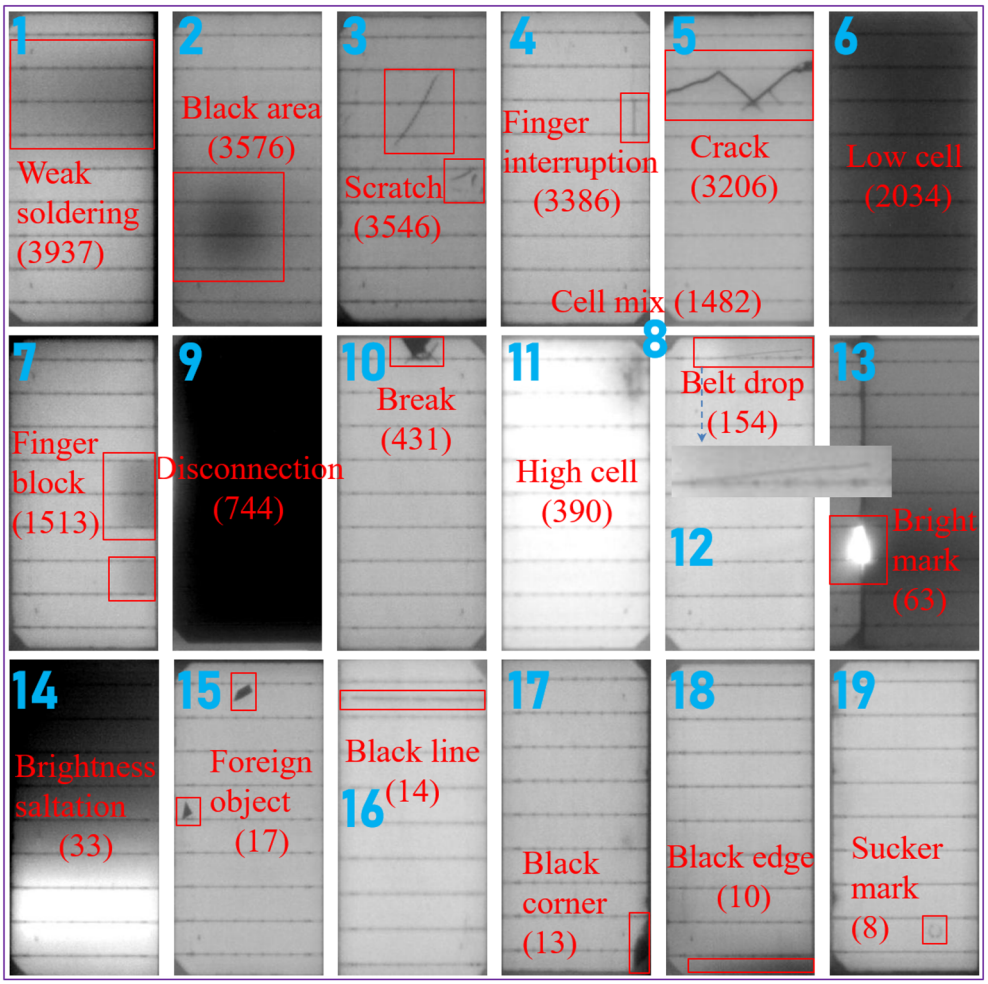

FIGURE 1 Nineteen types of defects (appearance numbers) we labeled in 5983 defective modules: (1) weak soldering, (2) black area, (3) scratch, (4) finger interruption, (5) crack, (6) low cell, (7) finger block, (8) cell mix, (9) disconnection, (10) break, (11) high cell, (12) belt drop, (13) bright mark, (14) brightness saltation, (15) foreign object, (16) black line, (17) black corner, (18) black edge, and (19) sucker mark

subimages to highlight the defects in them and enlarge the data size. A cell mix is a subimage that includes high/low cells and normal cells. (9) A disconnection appears as an absolute dark cell and it is mainly due to cell connection faults. (10) A break is a dark area where part of a cell breaks and falls and it is usually accompanied by cracks. The main cause of break is also thermo-mechanical stresses like collision. (11) A high cell has higher efficiency than other normal cells and appears brighter. It is not a fault but shall be detected and picked out from other cells in the same module, and used in another module with higher efficiency to avoid power waste, required by the producers. (12) A belt drop is a visible separation of the belt from the busbar and blocks current flow. (13) A bright mark is an obvious bright area where current is over intense. The probable reasons can be some abnormalities on the edge that cause increase of leakage current, such as leaked silver paste, problems of soldering, corrosion and firing, and so on. It needs the inspector to check the visual image or even the real module to figure out the exact reason. A bright mark can generate hot spot and even burn the whole module if not repaired in time. (14) A brightness saltation shows that a cell has inhomogeneous series resistance due to cell faults. (15) A foreign object is an anomalous object placed on the cell that can also generate hot spot as time goes by. (16) A black line is a horizontal or vertical long straight dark line that is neither a normal scratch nor a crack. (17) A black corner indicates a corner that is dark but not broken, caused by over corrosion when producing the cells. (18) A black edge is a dark cell edge that is caused by over corrosion. (19) A sucker mark is generated in transportation by machines. These defects were collected from one factory, and other companies may have a few different types due to distinction of manufacturing technology. In addition, part of our defects can also be found in some literatures,[2,25–27] although they may have different names. To our knowledge, except for crack, finger interruption and break that other researchers used to study, the rest of our defects were barely investigated in automatic detection in literature.

According to the influence degree of these 19 types of defects, we can further divide them into four groups: (1) defects that seriously affect module efficiency and durability: weak soldering, crack, disconnection, break, belt drop, bright mark, brightness saltation, and foreign object; (2) defects that partly decrease module efficiency: black area, finger block, black line, black corner, and black edge; (3) defects that barely influence module efficiency but appearance: scratch, finger interruption, and sucker mark; (4) cell efficiency mismatch: high cell, low cell, and cell mix. Note that high cell and low cell are essentially not faults, but they still should be detected and picked out from other normal cells in the same module to avoid waste (high cell) as well as

module efficiency decrease and firing risk (low cell). Besides them, cell mix is an auxiliary labeling trick to show that high/low cells are mismatched with normal cells in a subimage. Based on the difference of influence degree, the four groups of defects are required to be detected and repaired with different tolerance in practical production environment. The first and fourth groups must be all detected and repaired while thresholds of geometric size will be set for the second and third groups to block serious ones and release others, in which way module quality and capacity can be balanced.

It is revealed form Figure 1 that there is a severe number imbalance among these defects and bigger number means higher possibility for a defect to show up in production lines. Considering the amount limitation of the last five types of defects, in this paper, we concentrate on the top 14 types of them.

3 | METHODOLOGY

Figure 2 describes the workflow of our method. The whole process is composed of three stages: (a) data preprocessing, (b) model experiment, and (c) application analysis. Data preprocessing includes building three datasets, oversampling, segmenting defective subimages, and augmenting data. Model experiment contains setting experimental parameters, training, selecting, and testing model. Application analysis involves feasibility analysis, defect screen criteria setting, normal/defective classification, and results report.

3.1 | Data preprocessing

3.1.1 | Building three datasets

In general, the full database should be divided into training, validation and test sets to do machine learning or deep learning. In this paper, training set is used to train the model. Validation set is used to compare the models under different data preprocessing strategies and select the best model. Test set is used to evaluate the selected best model. The evaluation result on test set should be at least well matched with the result on validation set to ensure the reliability of the selected model to be adopted in practice. We randomly divided the 5983 defective module EL images into training, validation and test sets at the ratio of 6:2:2. Since the defect numbers on different modules are not the same, after random module division, the defect amounts of validation set and test set are also slightly different, as shown in columns 3–5 of Table 1.

3.1.2 | Oversampling

There is a severe number imbalance among different types of defects as shown in Figure 1. Inspired by the research[28] in which oversampling was used to increase defective data size and achieved good result, we also utilized oversampling to increase the proportions of some minor defects. In our exploratory experiments, we found that among the defects fewer than 1000, the model performed bad on break, belt drop and bright mark but well on disconnection, high cell, and brightness saltation. This suggested that for those obvious and visually distinctive defects like disconnection, high cell, and brightness saltation, a finite number of them were enough to achieve good results so they were not oversampled. For other three types of

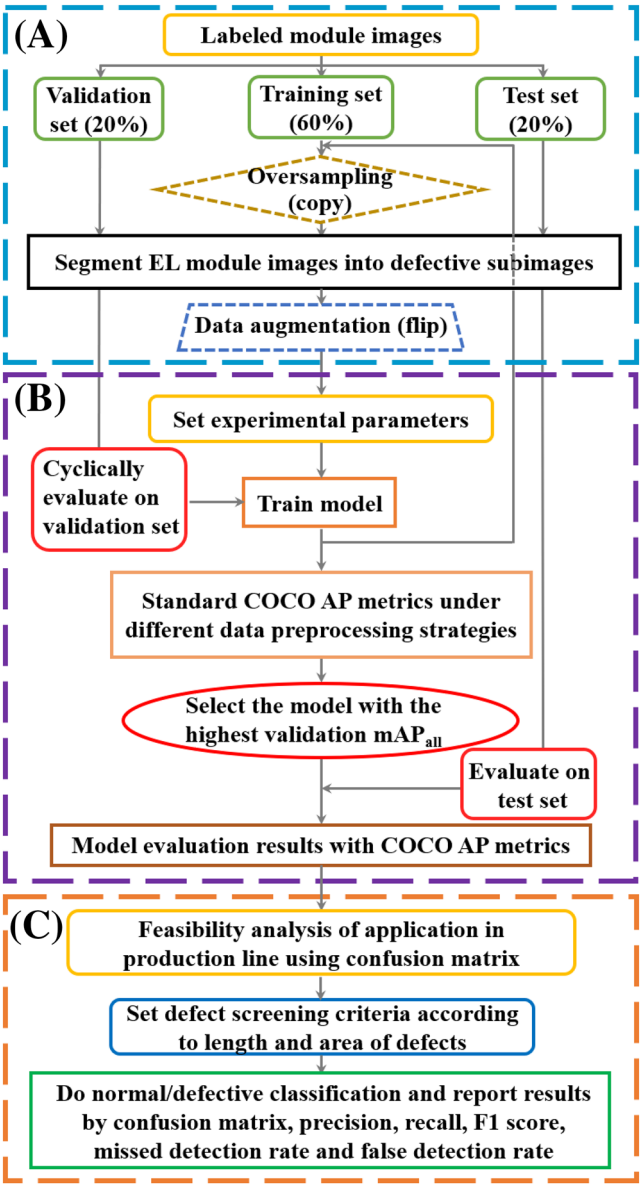

FIGURE 2 Workflow of our method is composed of three stages: (A) data preprocessing including building training, validation and test sets, oversampling, segmenting subimages and augmenting data, (B) model experiment including setting experimental parameters, training model under different data preprocessing strategies as well as selecting and evaluating the best model using mAP_{all}, and (C) application analysis including feasibility analysis using confusion matrix, defect screening criteria setting according to length and area of defects, normal/defective classification and results report by confusion matrix, precision, recall, F1 score, missed detection rate, and false detection rate

TABLE 1 Information of all types of defects including number distribution among training (Train), validation (Val), and test sets, among the four data preprocessing strategies (None, Copy, Flip, and Both) of training set and evaluation results of the best model on validation set (AP_{all}) and on test set (AP_{all}, AP_{50}, and AP_{75})

					Data strategies (train set)				Val set	Test set (%)		
Id	Defect type	Train set	Val set	Test set	None	Copy	Flip	Both	AP_{all}	AP_{all}	AP_{50}	AP_{75}
1	Weak soldering	2378	722	837	2378	3853	9512	15412	48.4	44.1	80.1	43.7
2	Black area	2157	674	745	2157	2602	8628	10408	25.8	26.4	70.8	11.7
3	Scratch	2138	709	699	2138	2978	8552	11912	18.3	19.6	52.2	10.7
4	Finger interruption	2000	705	681	2000	2680	8000	10720	11.3	10.8	41.7	2.0
5	Crack	1884	653	669	1884	3779	7536	15116	46.7	44.8	85.9	42.3
6	Low cell	1291	416	327	1291	1966	5164	7864	66.7	58.9	64.0	59.8
7	Finger block	997	203	313	997	1377	3988	5508	43.2	42.9	85.2	35.3
8	Cell mix	928	288	266	928	1243	3712	4972	86.1	84.6	84.7	84.7
9	Disconnection	443	139	162	443	663	1772	2652	97.5	98.5	99.9	99.9
10	Break	253	66	112	253	1543	1012	6172	35.9	30.6	66.4	22.7
11	High cell	201	99	90	201	336	804	1344	84.8	78.3	83.5	80.3
12	Belt drop	93	40	21	93	1023	372	4092	22.0	31.4	52.4	30.8
13	Bright mark	35	20	8	35	385	140	1540	17.8	37.8	64.9	22.6
14	Brightness saltation	21	7	5	21	181	84	724	85.8	37.9	50.5	39.2
								mAP	49.3	46.2	70.2	41.8

Note: Each mAP value is the mean of corresponding AP values of all the 14 types of defects.

defects, we oversampled (copied) three times for the modules carrying break, 10 times for the modules carrying belt drop and bright mark to improve their proportions. As a side effect of copying on modules, other types of defects in these modules were also duplicated to different extent. Moreover, for the defects more than 1000, the model did not perform well on scratch and finger interruption but since there were already large numbers of them, copying more would make little sense so they were not specifically oversampled like break, belt drop, and bright mark.

3.1.3 | Segmenting subimages

We segmented the module images into defective subimages based on two considerations: (1) a module is dominated by its background which is unrelated information for the detection of defect, so it was segmented into defective subimages to highlight the defects. (2) The size of a module EL image is 6500 × 3200 pixels and should be scaled to 800 × 800 pixels to fulfill the input size limitation of the CNN model we used.

3.1.4 | Augmenting data

This operation is to generate additional training data from a limited training set. Common approaches to do augmentation include flip, cropping, translation, rotation, brightness, and contrast changes. In this paper, we only adopted horizontal, vertical, and diagonal flip and other approaches have not been used yet. Extra experiments still remain to be done to determine a better augmentation strategy.

Figure 2a presents the workflow of data preprocessing: at first, the 5983 labeled EL images of defective PV modules were randomly divided into training set, validation set, and test set at the ratio of 6:2:2. Then, oversampling was done on training set to copy more modules that carry break, belt drop, and bright mark. Next, all module images of training set, validation set, and test set were segmented into subimages. At the end, subimages of training set were flipped along vertical, horizontal, and diagonal lines to produce more training data. Furthermore, to investigate the particular effects of oversampling (copy) and data augmentation (flip), we set contrast experiments under four data preprocessing strategies: (1) None: neither copy nor flip was applied; (2) Copy: only copy was applied; (3) Flip: only flip was applied; and (4) Both: both copy and flip were applied. The number distribution under these four strategies are shown in columns 6–9 of Table 1.

3.2 | Model experiment

3.2.1 | Setting experimental parameters

The deep learning model we adopted in this work is Mask R-CNN with ResNet-101-FPN backbone[22] in Detecton2[23] platform. Before training, some settings should be done at first. In this work, we set 300 k as maximum iteration, 0.01 as basic learning rate, and 8 as batch size for 4 RTX 2080Ti GPU (2 per GPU). Additionally, the basic

learning rate was set to decrease by 10 twice at the 150kth and 200kth iterations. Other values of these parameters were also tried in our exploratory experiments and they are not shown for the triviality of their results. Besides, we set 12.5 k as the iteration period to cyclically evaluate the model on validation set and save intermediate models during training process. All other parameters were maintained as defaults in Detectron2.

3.2.2 | Training, selecting, and testing model

With datasets prepared and settings done, we just need to run a python file in Detectron2 to start training the model on training set and it goes automatically until 300 k iterations. During the training process, validation set was cyclically used to evaluate the model performance, and results of standard COCO AP metrics[24] were stored in tfevents file, which could be loaded and visualized by tensorboard. Among the various AP metrics, mean average precision over all IoU thresholds (mAP_{all}, the same as AP in official web page and we call it mAP_{all} in this paper to avoid confusion) is the single most important and common used metric in object detection task so among the models that were stored during training process and under the four data preprocessing strategies, the one that had the highest mAP_{all} on validation set was selected as the best model. Then, the selected best model was evaluated on test set with the same COCO AP metrics as an ultimate measurement. By the way, only AP values on bounding boxes will be reported since masks are for instance segmentation and not necessary for object detection task in this work.[22]

Figure 2b illustrates the process of our model experiment: at first, we set experimental parameters, then started to train the model on training set. During training process, validation set was cyclically used to evaluate the model performance, and each evaluation result of standard COCO AP metrics was saved. After training under the four data preprocessing strategies finished, the model that had the highest mAP_{all} on validation set was selected as the best model, and then it was ultimately evaluated on test set with the gained COCO AP metrics as the final results.

3.3 | Application analysis

3.3.1 | Feasibility analysis

AP metrics are excellent evaluation indicators for characterizing model performance on each type of defects in object detection task, but they are independent single values and cannot display the details of the model predictions among different types. Thus, we calculated a confusion matrix based on our defect detection problem as a supplement to display the wrong predictions and missed defects of the selected best model on test set. Analyzing these inaccurate predictions helps us more clearly understand the model performance and see if it is feasible to be used in production line.

3.3.2 | Defect screening criteria setting

Actually, EL inspection for a PV module in production line is a normal/defective classification task, and a module to be defective or normal is based on whether the detected defects surpass their screening criteria or not. Typical criteria include specific thresholds of length and area for certain types of defects. Moreover, multiple thresholds of length and area can be set to divide the module into different quality classes. The criteria are not unchangeable and up to requirements from producers. In this paper, we set basic screening criteria, which involves the length and area limitation but ignored the quality class division to simplify the inspection process.

3.3.3 | Normal/defective classification and results report

To demonstrate the effectiveness of our method, we collected additional 4791 EL images of PV modules taken on 3 days in a real production line and did normal/defective classification using the selected best model. These images included normal and defective ones, but the numbers were not clear at the beginning. They were inspected by the best model and divided into normal and defective classes and then we manually checked the classified EL images one by one. For this normal/defective classification task, we report the classical confusion matrix which is composed of true positive (TP), false positive (FP), true negative (TN), and false negative (FN). In EL inspection, since we care more about defective modules, TP represents how many ground truth defective modules were correctly predicted as defective ones, FP represents how many ground truth normal modules were wrongly predicted as defective ones, TN represents how many ground truth normal modules were correctly predicted as normal ones, and FN represents the ground truth defective modules that were wrongly predicted as normal ones. With them determined, precision and recall can be calculated from Equations 1 and 2 while precision describes among the modules the best model predicted as defective, how many of them are truly defective and recall describes among the modules that are truly defective, how many of them are predicted as defective by the best model. At the end, F1 score (Equation 3) is the ultimate indicator, which is harmonic average of precision and recall. Practically, the production line tends to use another two wilder measurements: missed detection rate (MDR) (Equation 4) and false detection rate (FDR) (Equation 5) that, respectively, show the proportions of FN and FP among the whole capacity of 1 day as the capacity is easier to count (although it may not be very scientific since if all the modules are perfect, the two rates will be very low but it should owe to the good manufacturing instead of the good performance of our model). After a thorough consideration of production efficiency and quality, the producers and managers determined to set 0.2% and 2% as thresholds for MDR and FDR, respectively, to assess whether the performance of the best model reaches the standard and can be really adopted in production line.

$$\text{precision} = \frac{TP}{TP + FP} \tag{1}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{2}$$

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{3}$$

$$\text{MDR} = \frac{FN}{TP + FP + TN + FN} \tag{4}$$

$$\text{FDR} = \frac{FP}{TP + FP + TN + FN} \tag{5}$$

Figure 2c shows the process of this section: at first, we did feasibility analysis of applying the best model in production line using a confusion matrix that is calculated based on our object detection task. Next, we set specific screening criteria for different types of defects according to their length and area. Then we use the best model to conduct normal/defective classification with the screening criteria, and finally, the inspection results are reported by confusion matrix, precision, recall, F1 score, MDR, and FDR.

4 | EXPERIMENTAL RESULTS AND DISCUSSION

4.1 | Experiments to select the best model

Figure 3a presents the mean training processes over all 14 types of defects under the four data preprocessing strategies with mAP_{all} of validation set and total loss[22] against iteration and learning rate (Lr). As we can see, each mAP_{all} curve of validation set starts at a high point, which is due to the use of a pretrained model that avoided training from scratch and saved a lot of time. As iteration goes, mAP_{all} rises on the whole and the first decrease of learning rate at the 150kth iteration gives a sudden jump and drop for each mAP_{all} curve and its corresponding total loss curve. But the second decrease of learning rate at the 200kth iteration has no obvious effect which means the model performance has already touched its extreme and cannot be further improved by this way. Furthermore, there is a slight trend for the four mAP_{all} curves to fall off after passing their peak values, and it suggests that this part of each training process is overfitting the training set which has no more effect to improve the model performance on validation set. Another notable observation is that, compared with total loss curves that drop off relatively more smoothly, mAP_{all} curves have higher fluctuation degrees that reflect the real evolution processes of model performance while the functionality of total loss curves is just to ensure the convergence of training processes.

As mentioned in Section 3.2, the model that has the highest mAP_{all} value on validation set under the four data preprocessing strategies should be selected as the best model and we have red-circled the best one in Figure 3a. To our surprise, the best model does not take place on curve 4 but on curve 3 which implies a negative effect of copy on mean performance of the model on all types of defects. Actually this conclusion can also be drawn by comparing curves 1 and 2 in Figure 3a. Although the highest mAP_{all} value of curve 2 is bigger than that of curve 1, curve 2 is lower than curve 1 for all the remaining iterations after passing its peak which also suggests a probable negative influence of copy. To investigate this deeper, we give training process illustrations of the three specifically copied minor defects: break, belt drop, and bright mark (Figure 3b–d) and another incidentally copied defect: high cell (Figure 3e) for comparison. As can be calculated from columns 6 and 7 of Table 1, copy increased proportions of break, belt drop, and bright mark from 1.7%, 0.6%, and 0.2% to 6.3%, 6.9%, and 1.6% while the proportion of high cell stays around 1.4%. Comparing curves 1 and 2 in each of Figure 3b–d, we can find that copy surely improves the model performance for all of the three minor defects but the improvement degree declines from break to bright mark which also indicates that the contributions of duplicated data are not stable and up to the nature of different types of defects. To be clear, for break, more features can still be extracted by loading its duplicated data, but for belt drop and bright mark, the remaining features are much fewer and/or more difficult to be extracted than for break. Moreover, increasing the proportions of the three minor defects through copying has negative influence for some other defects. Figure 3e presents high cell as a representation whose curve 2 is almost the lowest all the way. Besides, focusing on curve 3 in each of the Figure 3b–e, we can find that flip also has various effects on different defects, but averaging on all 14 types, the ultimate result is that copy brings down the mean performance of the model while flip enhances it and this is why the best model is on curve 3 but not on curve 4 in Figure 3a.

4.2 | Evaluation result of the best model

After the best model was selected, we evaluated its performance on test set, and AP_{all} values on both validation set and test set for each type of defects are listed in columns 10 and 11 of Table 1. As we can see, the results on these two datasets generally match with each other with only one exception: brightness saltation. It has 85.8% AP_{all} on validation set but 37.9% AP_{all} on test set and a possible cause is the bias between the limited data, 7 and 5 defects in the two datasets. The mAP_{all} of test set is 46.2%, just slightly lower than 49.3% of validation set that demonstrates that the experiments and results are generally reliable and the best model can be adopted to predict other new data directly. Besides AP_{all} and mAP_{all}, we also list AP_{50}, AP_{75} for each type of defects to describe their results under 50% and 75% IoU thresholds in the last two columns (12 and 13) of Table 1 and the best model achieved 70.2% mAP_{50} and 41.8% mAP_{75}. Actually, pursuing high mAP value under strict localization accuracy demand like 75% IoU threshold is still a harsh challenge in computer vision and mAP_{50} is much easier to achieve a higher value than mAP_{75}. Besides, in our EL inspection task, it is still quite clear and recognizable if a defect is predicted by the model with a bounding box whose localization

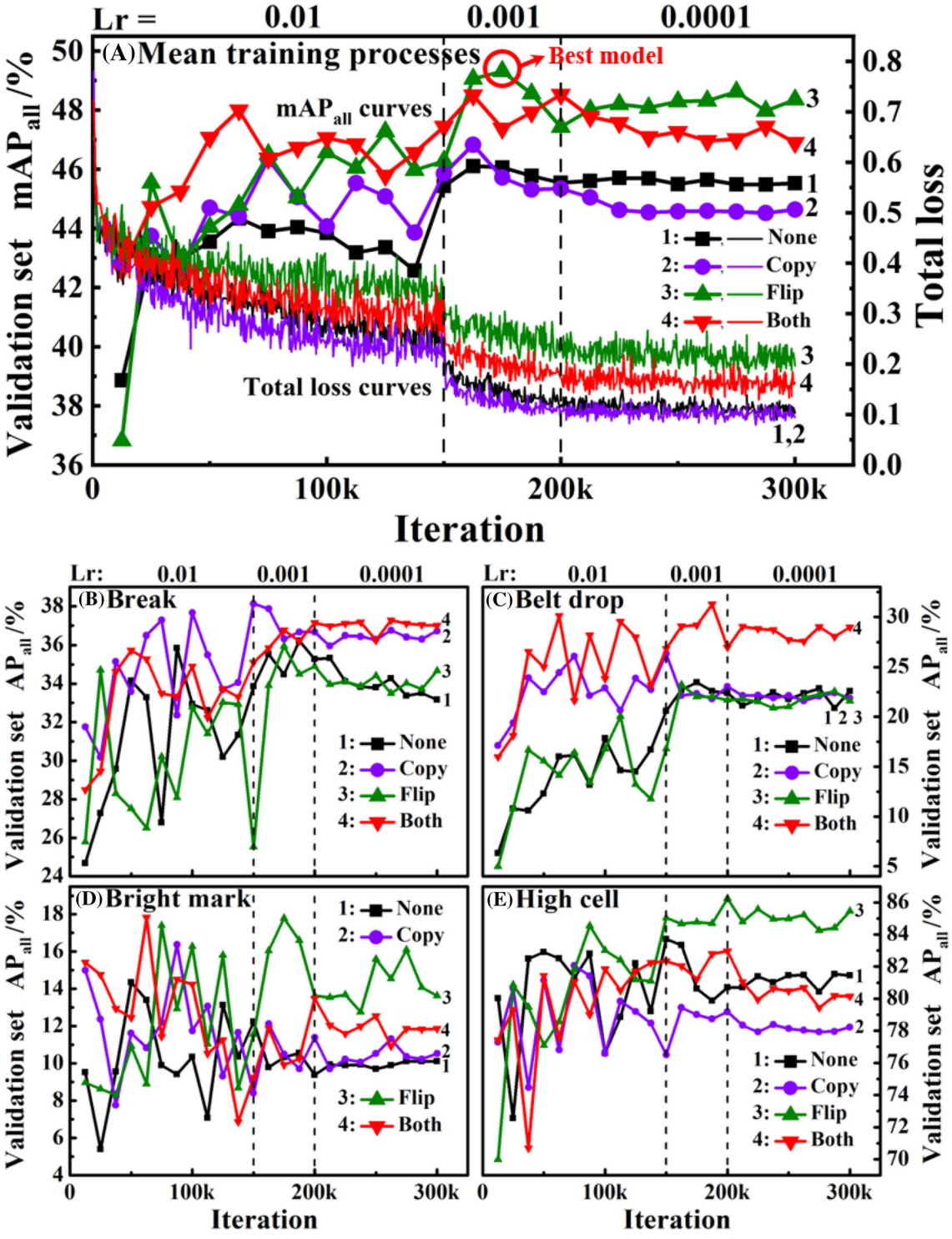

FIGURE 3 (A) Mean training processes of the model over all types of defects under the four data preprocessing strategies with mAP_{all} of validation set and total loss against iteration and learning rate (Lr). The best model with the highest mAP_{all} is marked by a red circle. Training processes of (B) break, (C) belt drop, (D) bright mark, and (E) high cell under the four data preprocessing strategies with AP_{all} of validation set against iteration and Lr

accuracy is 50% so we mainly use 70.2% mAP_{50} value to describe the mean performance of the best model.

We can analyze per-defect performance directly by their AP_{50} values of test set as shown in column 12 of Table 1, and some conclusions can be drawn. At first, the difficulty of a certain defect type for the CNN model to learn varies with types and this can be found by comparing their sample numbers of training set and AP_{50} values of test set. As we can see, belt drop, scratch, brightness

saltation and finger interruption are the four defects whose AP_{50} values are lower than 60%, but their original samples are at different orders of magnitude. Belt drop and brightness saltation only have fewer than 100 samples in training set while scratch and finger interruption have more than 2000, so we think belt drop and brightness saltation are mainly restrained by the limited samples but scratch and finger interruption are really difficult for the CNN to learn. For scratches, the probable reason maybe the CNN model was confused by the high variation in their length, severity and even radian, but for finger interruptions, we can only infer that their features are not strong enough for the CNN to train well. In this case, human perform obviously better because we are more robust to the variations and slight features. To precisely analyze the reasons, more specific researches should be done in future work. Second, from the perspective of defect groups introduced in Section 2, scratch and finger interruption both belong to the third group of defects that barely influence module efficiency but appearance and the best model generally performed well on other groups, except for some defects

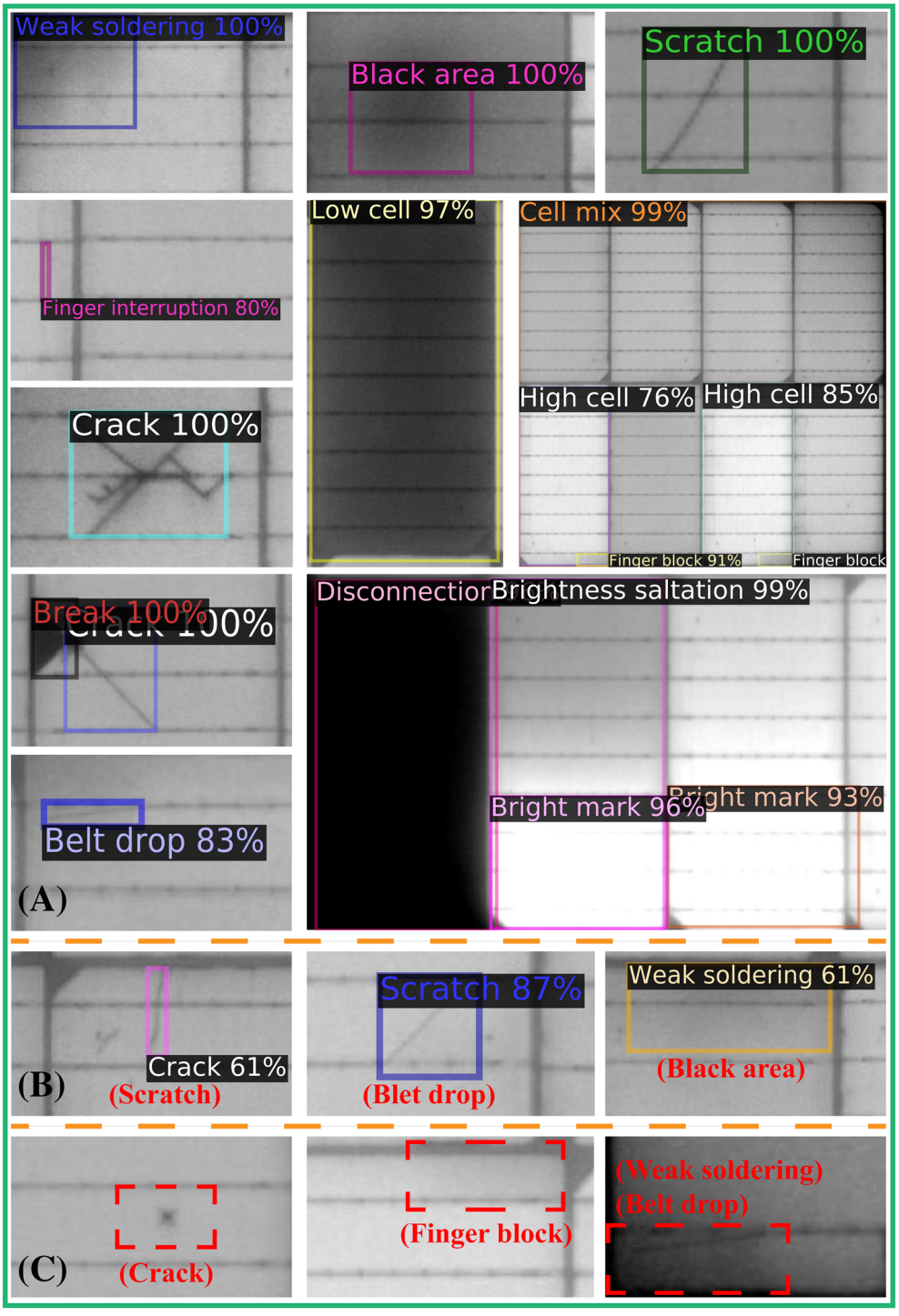

FIGURE 4 Defect prediction examples by the best model on test set. (A) Correct predictions for all 14 types of defects. (B) Wrong predictions with ground truth defect types given in parentheses. (C) Missed defects marked by dashed boxes with ground truth defect types given in parentheses

TABLE 2 Confusion matrix computed on test set between its ground truth defects (G) and predictions (P) from the best model

	P														
G	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Missed
1	679	2													156
	81.1%	0.2%													18.6%
2	15	575	8		1		3								143
	2.0%	77.2%	1.1%		0.1%		0.4%								19.2%
3		12	420	8	16							1			242
		1.7%	60.1	1.1%	2.3%							0.1%			34.6%
4			12	380			3								286
			1.8%	55.8%			0.4%								42.0%
5		5	35		566										63
		0.7%	5.2%		84.6%										9.4%
6						236		5							86
						72.2%		1.5%							26.3%
7		2					271								40
		0.6%					86.6%								
8						10		229			1				26
						3.8%		86.1%			0.4%				9.8%
9									162						0
									100%						0%
10		2			1		1			78					30
		1.8%			0.9%		0.9%			69.6%					26.8%
11								2			77				11
								2.2%			85.6%				12.2%
12			2									13			6
			9.5%									61.9%			28.6%
13													5		3
													62.5%		37.5%
14														4	1
														80.0%	20.0%
Unlabeled	156	177	271	277	88	301	55	142	1	41	37	13	1	4	
	10.0%	11.3%	17.3%	17.7%	5.6%	19.2%	3.5%	9.1%	0.1%	2.6%	2.4%	0.8%	0.1%	0.3%	

Note: 50% is set as thresholds for both IoU and score. 1–14 represent the 14 types of defects. The upper number in each grid indicates how many of the corresponding row type of defects (ground truth) are detected as the corresponding column type (prediction) of defects, and the lower number in each gird is its proportion calculated within the row. The numbers on the diagonal line indicate the correctly predicted defects while the numbers out of the diagonal line in rows and columns 1–14 are wrong predictions among different types of defects. The last column and row represent the missed defects and unlabeled predictions, respectively. Cells are colored according to their percentage values.

with limited samples like belt drop and bright mark. To sum up, 70.2% mAP_{50} value is generally a good result but still has room to be improved.

Figure 4 illustrates the defect prediction results by the best model on test set with (a) correct predictions, (b) wrong predictions, and (c) missed defects. Each prediction in Figure 4a,b is composed of three elements: a bounding box for defect localization, a class label for defect classification, and a score to show the model's confidence in the classification. Figure 4a exhibits some correct predictions for all 14 types of defects with high localization accuracies and scores. Except for them, we also present three wrong predicted defects in Figure 4b whose ground truth types are given in parentheses. In addition, Figure 4c shows three missed defects marked by dashed boxes that the best model was not able to recognize and their ground truth classes are also given in parentheses.

5 | APPLICATION IN PRODUCTION LINE

5.1 | Feasibility analysis using confusion matrix

Considering that 70.2% mAP_{50} is not a dramatic result and the various AP_{50} values cannot directly reflect the ratios of wrong predictions and missed defects for each type of them, these results were not able to give us a clear picture of how well the best model would perform if used in real production line. So we calculated an object detection based confusion matrix as given in Table 2 to display the prediction details and do more analysis. This confusion matrix is a visualization of AP_{50} values in a simplified case that predictions with scores lower than 50% were discarded. It is calculated by matching every ground truth (G) defects with every model predictions (P) if their IoU values are greater than 50% and each of ground truth defects and predictions can only be matched once. More computing details can be acquired from part 1 of the supporting information.

In Table 2, numbers 1–14 represent the 14 types of defects with first column and row indicating G and P, respectively. The upper number in each grid describes how many of the corresponding row type (ground truth) of defects are detected as the corresponding column type (prediction) of defects. The lower number in each gird is its proportion calculated within the row and the sum of all proportions in each row may be 0.1% greater or smaller than 100% due to rounding. The numbers on the diagonal line represent the correctly predicted defects while the numbers out of the diagonal line are wrong predictions among different types. Moreover, the numbers and proportions in the last column 'Missed' belong to the defects that are not predicted by the best model. Besides these information, there is another case that some defects or parts of cell background that were not labeled are also predicted by the model and we list them as 'Unlabeled' in the last row of Table 2.

As we can observe from Table 2, the proportions of correct predictions on the diagonal line have similar properties with AP_{50} values in column 12 of Table 1 that defects with higher AP_{50} values tend to have higher rates to be correctly predicted. Except for them, the wrong predictions only take tiny shares but the missed defects have really unignorable proportions which cannot be accepted if used in real production line. By manually checking these missed defects, we found that they can be divided into three cases and only the third case harms the inspection in production line.

The first case is named as recognition biased misses and it has three subcomponents: (1) close defects that have been separately labeled are predicted as one, (2) a single defect is recognized into separate parts, and (3) slight shift of predicted bounding boxes. The missed defects belonging to these three subcomponents of the first case were counted into misses in calculations of AP_{50} and confusion matrix as their IoU values were lower than 50%, but from the perspective of practical EL inspection, they are all distinguished from cell background and thought to be correct predictions since there is no conception of IoU when doing inspection and the recognition bias is acceptable in production line.

The second and third cases of missed defects are all true misses, which means they are not predicted with any bounding boxes by the best model, but they are a bit different. Before introducing them, we shall reveal a significant distinction between object detection task and classification task. Object detection task pursues completeness of detections and is accurate to each defect, but normal/defective classification task of a module is an existence problem, which means as long as a defect that surpasses the screen criteria is correctly predicted, the module should be classified as defective, even if other defects in the same module are incorrectly predicted or missed. This phenomenon is a compensation of detection as the wrong predictions and missed defects are compensated by the correct predictions and do not affect the inspection result of the module. So these compensated misses correspond to the second case, and the uncompensated misses are the third case in which the module would be wrongly classified as normal for the incapacity of their predictions.

As for the unlabeled predictions, there are also three cases: (1) recognition biased predictions that detected labeled defects but IoU values were lower than 50% due to recognition biases, (2) correct predictions for unlabeled defects, and (3) wrong predictions for unlabeled defects or for parts of cell background that were incorrectly recognized as defects.

We looked into the missed defects and unlabeled predictions one by one and segregated them into their respective cases through checking predicted classes and existence of recognition bias and compensation (subimage-scale compensation). The statistics result of proportion distributions of the six cases is given in Figure 5, and for every type of defects, each amount of three cases for missed defects and unlabeled predictions corresponds to its number in the last column and row of Table 2. Besides, we also give specific illustrations of six cases in supporting information (part 2). As Figure 5 shows, the uncompensated misses only take small shares while compensated misses take major parts for almost every type. Moreover, since the numbers of uncompensated misses were counted on subimages, the missed defects still have chance to be further compensated by correct predictions in other subimages in the same module. So at the end, their influence would fall off to a much lower level. As for the

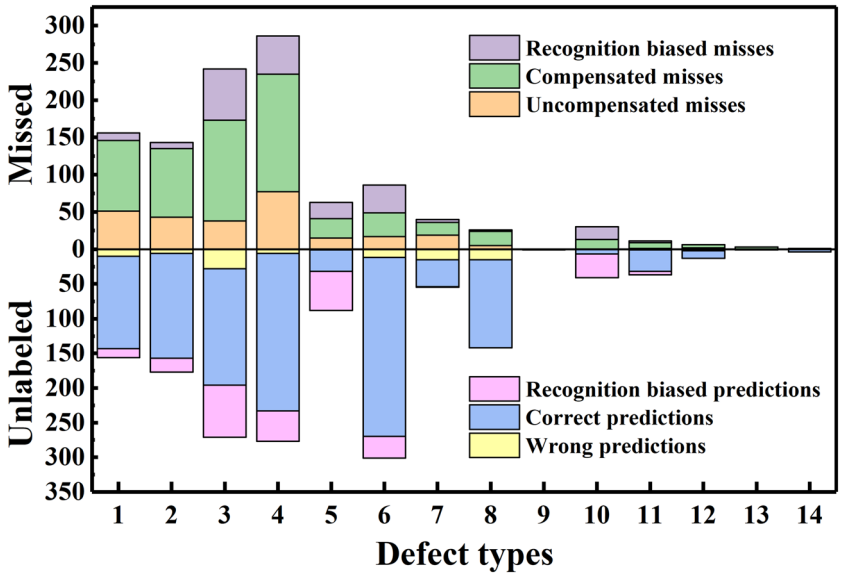

FIGURE 5 Proportion distributions of the special cases for the missed defects and unlabeled predictions in Table 2

unlabeled predictions, the wrong predictions only occupy tiny proportions for each type and are not big issues to concern. According to these results and analysis, the model surely has the potential to perform well in the normal/defective classification of modules in production line.

5.2 | Screening criteria and classification results

To conduct normal/defective classification in production line, specific screening criteria should be set to selectively block and release different types of defects. According to the discussion in Section 2, different permission degrees were set for the four groups of defects: (1) fully permitted: finger interruption, (2) conditionally permitted: scratch with length smaller than half the width of a cell, black area with area smaller than 20% of the size of a cell and finger block with area smaller than 10% of the size of a cell, (3) fully forbidden: all other types of defects. The harm of a single finger interruption to the cell efficiency and appearance is negligible, and its frequency of occurrence tends to be high so it is always ignored and released to ensure capacity of production. Scratch, black area, and finger block also occur very frequently in production line, and their sizes can vary a lot so thresholds of length and area are set to block serious ones of them. We adopted approximations in the calculations that the length of the diagonal line of a predicted bounding box was used to replace the length of a scratch, and the area of a bounding box was used to replace the area of black area and finger block. As for other types of defects, since they have more serious impact on module efficiency and safety, they are all set to be forbidden once correctly predicted.

To test the best model's performance on classification task in production line, we additionally gathered 4791 EL images of PV modules taken on 3 days during production. These images included normal and defective ones, and their numbers were not clear at the beginning. At first, these modules were inspected and classified into normal and defective groups by the selected best model on the basis of screening criteria. Then the experienced leader manually checked them one by one to determine the values of TP, FP, TN, and FN that we introduced in Section 3.3, and the statistics result is given in Table 3. Take the data on day 1 as an example, there were 1562 normal modules, and the model predicted 1536 (TN) of them as normal ones while the left 26 (FP) modules were wrongly predicted as defective ones. There were 276 defective modules, and the model retrieved 274 (TP) of them with only 2 (FN) missed. According to the Equations (1) and (2), we achieved a good result of precision (91.3%), which means among the modules that the model predicted as defective, 91.3% of them were truly defective, and a marvelous value of recall (99.3%), which shows 99.3% of the ground truth defective modules were successfully retrieved. As we know, recall is more important for inspection in production line since once a defective module is mistakenly released, it becomes a final product and the defects in it cannot be checked anymore. Hence, the recall value of 99.3% is extremely reliable to block as many defective modules as possible. The model also yielded from Equation 3 a great F1 score (95.1%) as a balance of precision and recall. Except for the precision, recall and F1 score results commonly used in literature, we also achieved very low MDR (0.1%) and FDR (1.4%) that production line cares about according to the Equations (4) and (5). They are both, respectively, below the two thresholds (0.2% and 2%) so the performance of the best model reaches the standard of production line. However, the distances between the two rates and their thresholds are not obvious, which means the results may not be always reliable if practically applied, just as the other two MDRs on day 2 (0.2%) and day 3 (0.3%) shown. In this case, some unconventional and empirical tricks were used to additionally lower the MDR and FDR, such as manually adjusting confidence thresholds and setting brightness thresholds for different types of defects according to the missed and wrong detections, so that some of them can be corrected and

TABLE 3 Normal/defective classification results on additional 4791 EL images of PV modules taken on 3 days

	Prediction					
	Day 1		Day 2		Day 3	
Ground truth	Normal modules	Defective modules	Normal modules	Defective modules	Normal modules	Defective modules
Normal modules	1536 (TN)	26 (FP)	1503 (TN)	19 (FP)	786 (TN)	16 (FP)
Defective modules	2 (FN)	274 (TP)	4 (FN)	277 (TP)	3 (FN)	345 (TP)
Precision	91.3%		93.6%		95.6%	
Recall	99.3%		98.6%		99.1%	
F1 score	95.1%		96.0%		97.3%	
MDR (<0.2%)	0.1%		0.2%		0.3%	
FDR (<2%)	1.4%		1.1%		1.4%	
Compensation rate	8.8% (24/274)		14.1% (39/277)		24.6% (85/345)	

the results are continuously eligible for 2 weeks' check before finally deploying the model in line. Further description of the adjustments will be very trivial and not expanded in this paper. Furthermore, considering the existence of detection compensation, we counted the numbers of compensated modules in which some of the defects are missed but compensated by other correct predictions in the same module, and it was 24 with corresponding compensation rates of 8.8% for the data on day 1. Generally, this compensation rate was not too high which means the defects on most of the TP modules were completely predicted. Finally, we can conclude that our deep learning based automatic detection of multi-type defects presented in this paper surely has a highly promising potential to be directly adopted in production line according to the model results in Table 3. In fact, we have been deploying this technique in plants of Chinese companies with capacity up to 15GWp per year to automatically inspect the PV modules.

However, we cannot ignore the fact that there are still some limitations of our method. First, although the corresponding compensation rates of 8.8%, 14.1%, and 24.6% for 3 days are not too high, they still indicate that defects in small parts of the TP modules have not been completely predicted. The present best model still has room to be improved to lower these compensation rates and pursue higher AP values at the same time. Possible solutions include using better augmentation strategy, adopting more complicated deep learning structures, gathering and labeling more EL images of defective modules with minor defects. Second, just as Greulich et al.,[29] recently studied, the reproducibility problem of human labeling does exist that our four well trained people surly have divergence of some ambiguous defects and some of them were wrongly labeled or forgot to label. Although the four people would discuss about the ambiguities and an appointed experienced leader would make the decision, mistakes were not evitable. But human labeling seems not replaceable in the short term, especially for large projects with many different categories of objects. Luckily, some assistant auto-labeling skills like smart labeling proposed by Kunze et al.[30] may improve the labeling efficiency and accuracy to some extent in future work.

6 | CONCLUSION

In this paper, we firstly introduced 19 types of defects that we labeled in 5983 EL images of defective PV modules, divided them into four groups according to their influence degree, and chose the top 14 as research objects. Next, we trained Mask R-CNN with ResNet-101-FPN backbone and selected a best model with highest mAP_{all} value on validation set. Meanwhile, we also found that improving proportions of minor defects through copy has unstable effects for themselves and negative influence for some other types, and averagely, the best model with highest mAP_{all} value turned out to take place when only flip was applied. Then we measured the best model's performance on test set and achieved 70.2% mAP_{50} value. We found four types of defects have lower than 60% AP_{50} values, among which scratch and finger interruption that barely influence module efficiency are just the two most difficult defects for the CNN to learn. We inferred variations in length, severity, and radian of scratch and slight features of finger interruption are the reasons why the model cannot learn them well, but more specific researches should be down in future to give precise explanation. In addition, the best model generally performs well on other defects of different influence degrees, except for certain types with limited samples such as belt drop and brightness saltation. To further display the prediction details and analyze the feasibility of applying the best model in real production line, we calculated an object detection based confusion matrix and discovered three special cases for both missed defects and unlabeled predictions. On the one hand, among the missed defects, only uncompensated misses harm the normal/defective classification in production line and they just take small shares for each type of defects. On the other hand, wrong predictions also occupy tiny parts in unlabeled predictions, so the model's performance in production line would be much better than 70.2% mAP_{50} value showed. Finally, we set screening criteria for different types of defects to selectively block and release them and conducted the normal/defective classification task on the additionally collected 4791 EL images of PV modules on 3 days in production line. We successfully achieved superior F1 scores of 95.1%, 96.0%, and 97.3%, together with marvelous recall

values of 99.3%, 98.6%, and 99.1% (i.e., the model retrieved almost all of the defective modules) and quite good precision results of 91.3%, 93.6%, and 95.6%. Meanwhile, we also realized very low MDR (0.1%) and FDR (1.4%) that production line cares about and used supplementary tricks to keep them stably eligible before deploying the model in line. At the end, we can conclude that our deep learning based automatic detection of multi-type defects surly has a highly promising potential to be directly adopted in real production line.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (nos. 11834011 and 11974242).

ORCID

Yang Zhao https://orcid.org/0000-0002-5633-9973

REFERENCES

1. Needleman DB, Poindexter JR, Kurchin RC, Peters IM, Wilson G, Buonassisi T. Economically sustainable scaling of photovoltaics to meet climate targets. *Energ Environ Sci*. 2016;9(6):2122-2129.
2. Köntges M, Kurtz S, Packard CE, et al. Review of failures of photovoltaic modules. *IEA-PVPS*. 2014;T13-T01.
3. Breitenstein O, Bauer J, Bothe K, et al. Can luminescence imaging replace lock-in thermography on solar cells? *IEEE J Photovolt*. 2011;1(2):159-167.
4. Tsai DM, Wu SC, Chiu WY. Defect detection in solar modules using ICA basis images. *IEEE Trans Ind Inform*. 2013;9(1):122-131.
5. Anwar SA, Abdullah MZ. Micro-crack detection of multicrystalline solar cells featuring an improved anisotropic diffusion filter and image segmentation technique. *EURASIP J Image Video Process*. 2014;15.
6. Spataru S, Hacke P, Sera D, IEEE. Automatic detection and evaluation of solar cell micro-cracks in electroluminescence images using matched filters. in 2016 IEEE 43rd Photovoltaic Specialists Conference, IEEE, New York; 2016:1602.
7. Stromer D, Vetter A, Oezkan HC, Probst C, Maier A. Enhanced crack segmentation (ECS): a reference algorithm for segmenting cracks in multicrystalline silicon solar cells. *IEEE J Photovolt*. 2019;9(3):752-758.
8. Tseng DC, Liu YS, Chou CM. Automatic finger interruption detection in electroluminescence images of multicrystalline solar cells. *Math Probl Eng*. 2015;2015:879675.
9. Deitsch S, Christlein V, Berger S, et al. Automatic classification of defective photovoltaic module cells in electroluminescence images. *Sol Energy*. 2019;185:455-468.
10. Mayr M, Hoffmann M, Maier A, Christlein V, IEEE. Weakly supervised segmentation of cracks on solar cells using normalization L-P norm. in 2019 IEEE International Conference on Image Processing, IEEE, New York; 2019:1885.
11. Akram MW, Li GQ, Jin Y, et al. CNN based automatic detection of photovoltaic cell defects in electroluminescence images. *Energy*. 2019;189:116319.
12. Tang WQ, Yang Q, Xiong KX, Yan WJ. Deep learning based automatic defect identification of photovoltaic module using electroluminescence images. *Sol. Energy*. 2020;201:453-460.
13. Chen H, Wang S, Xing J. Detection of cracks in electroluminescence images by fusing deep learning and structural decoupling. 2019 Chinese Automation Congress (CAC). Proceedings 2019:2565.
14. Gundawar S, Kumar N, Meetei NR, Krishna Priya G, Puthanveetil SE, Sankaran M. Deep learning-based automatic micro-crack inspection in space-grade solar cells. *Advances in Small Satellite Technologies. Proceedings of 1st International Conference on Small Satellites. Lecture Notes in Mechanical Engineering (LNME)*. 2020:293.
15. Karimi AM, Fada JS, Hossaine MA, et al. Automated pipeline for photovoltaic module electroluminescence image processing and degradation feature classification. *IEEE J Photovolt*. 2019;9(5):1324-1335.
16. Rahman MRU, Chen HY. Defects inspection in polycrystalline solar cells electroluminescence images using deep learning. *IEEE Access*. 2020;8:40547-40558.
17. Dunderdale C, Brettenny W, Clohessy C, van Dyk EE. Photovoltaic defect classification through thermal infrared imaging using a machine learning approach. *Prog Photovoltaics*. 2020;28(3):177-188.
18. Fan SX, Li JB, Zhang YH, et al. On line detection of defective apples using computer vision system combined with deep learning methods. *J Food Eng*. 2020;286:110102.
19. Gao WY, Su C. Analysis on block chain financial transaction under artificial neural network of deep learning. *J Comput Appl Math*. 2020;380:112991.
20. Ghosh K, Stuke A, Todorovic M, et al. Deep learning spectroscopy: neural networks for molecular excitation spectra. *Adv Sci*. 2019;6:1801367.
21. Liu ZW, Yan S, Liu HG, Chen XF. Superhigh-resolution recognition of optical vortex modes assisted by a deep-learning method. *Phys Rev Lett*. 2019;123(18):183902.
22. He KM, Gkioxari G, Dollar P, Girshick R, IEEE. Mask R-CNN. in 2017 IEEE International Conference on Computer Vision, IEEE, New York; 2017:2980.
23. Detectron2, https://github.com/facebookresearch/detectron2
24. MS COCO AP metrics and contest, https://cocodataset.org/#home
25. Breitenstein O, Bauer J, Trupke T, Bardos RA. On the detection of shunts in silicon solar cells by photo- and electroluminescence imaging. *Prog Photovolt*. 2008;16(4):325-330.
26. Demant M, Welschehold T, Kluska S, Rein S. Microcracks in silicon wafers II: implications on solar cell characteristics, statistics and physical origin. *IEEE J Photovolt*. 2016;6(1):136-144.
27. Zafirovska I, Juhl MK, Weber JW, Wong J, Trupke T. Detection of finger interruptions in silicon solar cells using line scan photoluminescence imaging. *IEEE J Photovolt*. 2017;7(6):1496-1502.
28. Bartler A, Mauch L, Yang B, Reuter M, Stoicescu L, IEEE. Automated detection of solar cell defects with deep learning. In 2018 26th European Signal Processing Conference, IEEE Computer Soc, Los Alamitos; 2018:2035.
29. Greulich JM, Demant M, Kunze P, et al. Comparison of inline crack detection systems for multicrystalline silicon solar cells. *IEEE J Photovolt*. 2020;10(5):1389-1395.
30. Kunze P, Greulich J, Rein S, Ramspeck K, Hemsendorf M, Vetter A, Demant M. Efficient deployment of deep neural networks for quality inspection of solar cells using smart labeling, presented at *the 37th European PV Solar Energy Conference and Exhibition*, 2020:11.

SUPPORTING INFORMATION

Additional supporting information may be found online in the Supporting Information section at the end of this article.

How to cite this article: Zhao Y, Zhan K, Wang Z, Shen W. Deep learning-based automatic detection of multitype defects in photovoltaic modules and application in real production line. *Prog Photovolt Res Appl*. 2021;1–14. https://doi.org/10.1002/pip.3395