2023 IEEE 12th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS) | 979-8-3503-5805-6/23/$31.00 ©2023 IEEE | DOI: 10.1109/IDAACS58523.2023.10348813

The 12th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
7-9 September, 2023, Dortmund, Germany

Defects Localization in Images Using Deep Learning-based Classification with CAM Output

Rytis Augustauskas [1] , Lukas Zabulis [2], Arūnas Lipnickas [2], Simas Jokubauskas [1]
[1]Agmis, Savanorių 321c, LT-50120 Kaunas, Lithuania, www.agmis.com
rytis.augustauskas@agmis.com, simas.jokubauskas@agmis.com
[2] Department of Automation, Kaunas University of Technology, Studentų 48-109, LT-51367 Kaunas, Lithuania,
www.ktu.edu, lukas.zabulis@ktu.lt, arunas.lipnickas@ktu.lt

Abstract—**Computer vision-empowered quality is an important feature in modern manufacturing. These systems can perform defect inspections faster and more accurately than humans. There has been a rise in adopting deep-learning-based solutions for visual inspection tasks. However, new data is still necessary for each specific problem, and precise dataset annotation can be time-consuming considering precision-demanding approaches such as segmentation or region detection. This investigation proposes a method inspired by class activation mapping-based (CAM) to enhance deep neural network-based image classification algorithms by outputting an attention map where the defective area is indicated. With additional lightweight operations, it helps to roughly localize defects place in the image without using pixel-wise annotations. The proposed method is tested on glass bottles (*Oliena*), printed circuit boards (*PCB defects*), and industrial machine tool component surface defects (*BSData*) datasets.**

Keywords—defect detection; defect classification; quality inspection; explainable AI; deep learning.

I. Introduction

Inspecting manufacturing processes has become an essential part of Industry 4.0. Quality assessment at each step of the production line enables the detection of flaws at early fabrication stages, reducing materials usage and thus cutting manufacturing costs. This leads to better sustainability, a key component of Industry 5.0 [1]. Human-performed visual inspections are monotonous, time-consuming, and susceptible to fatigue-caused errors. In addition, they can create bottlenecks in the manufacturing pace as more personnel is needed to keep up with the massive production. More complex defect cases make relying on human inspection problematic due to increased subjectivity, resulting in inconsistent accuracy rates and error-prone inspections.

Automated computer vision (CV)-based quality inspection is an essential feature in modern manufacturing. Since the visual properties of produced items are imperative, it is necessary to maintain high standards throughout the entire fabrication process. In all scenarios, flaws should be detected, and defective products must be fixed, recycled, reused, or discarded. Manufacturers can use their time more efficiently and ensure high-quality products by conducting constant inspections throughout production, resulting in cost-savings and more sustainable manufacturing practices due to reduced raw material and energy usage.

Recently, there has been a significant rise in adopting deep-learning-based solutions for visual inspection tasks. The conventional image processing methods have failed to solve highly challenging problems, such as those in *ImageNet* [2] image classification competitions, unlike rule-based methods, data-driven algorithms, and deep convolution- or visual transformer-based networks, which transform a spectrum of data into multiple abstraction levels. These powerful solutions attempt to learn from an expert labeled 'knowledge' to predict unseen cases. Thus, deep learning models may provide suitable data-driven solutions for defect detection in manufacturing, even for dynamic data that present various and complex defects.

However, new data is necessary for each specific problem. Annotations still need to be performed by humans, and labeling tiny details in a large dataset is time-consuming. Labeling can even become mind-numbing in segmentation tasks that entail precise pixel-level annotations. Image classification can be engaged in tasks that do not require high specificity since the data sample can be managed by sorting through different directories or marked with a specific tag. While this approach looks tempting due to easier data preparation, it may not be suited for prediction reasoning or troubleshooting the problems with the algorithm (to which details it is reacting).

In this paper, a robust method inspired by class activation mapping-based (CAM) [3] is proposed to enhance the deep neural network-based image classification algorithm to output an additional attention map, which gives a rough estimation of the defect placed in the image in the single prediction iteration while not significantly increasing its calculations and not using pixel-wise annotations. Second, the non-trainable output is attached to neural network architectures, which output a rough attention map indicating the defective area in the image. Three popular architectures with standard and shorted configurations with a higher feature resolution are considered. Experiments are conducted on PCB, glass bottles, and industrial machine tool (drill) surface defects datasets. Code is outsourced in [4].

The manuscript is structured as follows. Section 2 reviews the existing approaches for defect detection. Section 3 presents an overview of the datasets used in this research. Section 4 covers the implementation of the proposed method and experiments methodology. Section 5 presents the results, and Section 6 summarizes the conclusions and future research directions.

II. Related work

Optical quality inspection is one of the most popular and widely utilized techniques in modern manufacturing [5]. Since it is a non-invasive inspection technique, a computer vision-based system might be applied to various observable cases. Specific visible features can even indicate the structural deterioration of manufactured goods. High interest can be seen even in tracing the defects in the wood industry [6]. Authors are utilizing deep learning-based region detection in scanned images. Recently, a huge, labeled dataset has been outsourced by Kodytek et al. [7], which shows the importance of particular quality control in wood material production. Moreover, the visual inspection method-inspired glass bottle production defects detection is proposed by Versini et al. [8], where authors utilize a segmentation model with a novel loss function for cracks in the glass segmentation. In the research, segmentation model performance is enhanced while evaluating average precision for binary segmentation. Another segmentation approach for various open surface defects databases is proposed by Üzen [9], where researchers are engaging modified *UNet* with *EfficientNet* backbone and modified skipped connection. An interesting approach for printed circuit board (PCB) defect detection is proposed by Kim et al. [10], where authors used an autoencoder with skipped connection (similar to *UNet*) for image PCB reconstruction where input and output (reconstructed) images are compared with structural similarity index measurement to indicate ‘faulty’ reconstructed details, in this case, defects. In this research, the autoencoder was trained only on defect-free samples. A lightweight convolutional neural network with pyramid feature extraction prior model for rail surface image defect detection by classification is introduced in [11]. In the approach described by Bergmann et al. [12], a few unsupervised techniques are proposed for anomaly detection and localization. Techniques proposed by the authors utilize only good samples for the training models. Also, some research outsources the dataset for benchmarking anomaly detection in the data. Different methods can be seen, such as classification, region detection, or segmentation, engaged by researchers in the field for various defect data. Precisely labeled data is needed for approaches such as region detection or segmentation, which can become a work. Only a few overviewed manuscripts [10], [12] engaged defect detection problems with less time-consuming data preparation approaches.

III. Data overview and preparation

A. Overview of investigated datasets

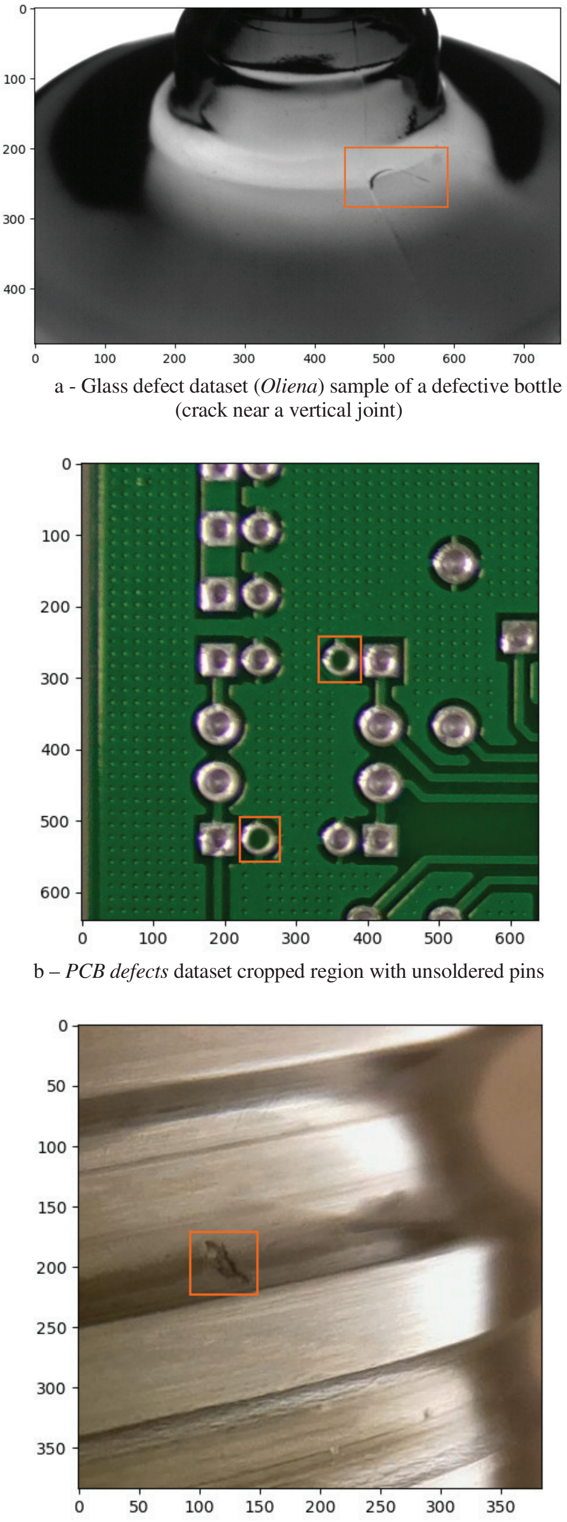
In this investigation, three different open image datasets are taken into consideration. All sets consist of different defects in the visual data. The first set, the glass bottle defects dataset (Oliena) published by Versini et al. [8], consists of 520 images (361 for training and 159 for testing). All images are greyscale with pixel-wise annotations. The second set introduces industrial machine tool component surface (ball screw drive (*BSData*) spindles) images with wear-off (pitting) defects [13] labeled pixel-wise. The dataset consists of 394 high-resolution and different-size RGB images (324 for training and 70 for testing). The third one is the printed circuit board dataset (*PCB defects*) published by Ding et al. [14] consists of 1386 high-resolution different-size printed circuit board images (split in 1109 for training and 277 for testing) with the following defects: missing holes, mouse bite, open circuits, short, spur, and spurious copper. This dataset is semi-artificial because image editor software generated defects on the given PCBs. The dataset consists of region- and class-wise annotation for the defect. The summary is given the Table I.

Table I. Datasets summary

	Oliena	*BSData*	*PCB defects*
Samples	520	394	1386
Width [px]	752	460 - 1016	1586-2530
Height [px]	480	1130 - 2928	2240-3056
Type	Grey	RGB	RGB

B. Data preparation

For this investigation, the binary image data classification approach is engaged. For this reason, each of the datasets was divided into separate classes/directories: defective and good. Pixel-wise and region labels were discarded in the training. Only the glass bottle defects dataset is kept at the original size. Others are divided into smaller regions with overlap using a sliding window approach due to huge image dimensions and different original sample sizes. The PCB dataset was divided into 640x640 pixel regions with 128-pixel overlap, and the BSD dataset was divided into 384x384 pixel regions with 128 pixels overlap. The divided region was assigned to defective or good samples according to the region consisting of defective pixels with an assigned number of pixels threshold: the PCB dataset was 1000px, and in the industrial machine tools wear-off set was 10px. Threshold values were picked considering defect size and partly visible defects' descriptive characteristics. Also, all six defect classes presented in the PCB dataset were considered a single defect. Prepared samples from each dataset are given in Figure 1.



a - Glass defect dataset (*Oliena*) sample of a defective bottle (crack near a vertical joint)

b – *PCB defects* dataset cropped region with unsoldered pins

c – *BSData* cropped region with pitting development

Figure 1. Prepared data samples (a, b, c)

IV. Neural network model implementation and training/inference methodology

A. *Neural network backbone*

Popular lightweight models were selected for this investigation as a backbone for binary classification. All chosen architecture belongs to the convolutional neural network family. First, MobileNetV3 [15], which is the successor from the previous generation, enhances with a squeeze and excitation [16] module. Like other MobileNets, this generation is lightweight and suitable even for mobile devices for computer vision tasks. Another low-latency architecture utilized in this research is EfficientNetV2's [17] lightest configuration *B0*. Authors of the second-generation models highlight higher efficiency through parameter reduction and faster training than the previous-generation solution. The last architecture explored in this manuscript is ConvNextTiny [18]. This model is bigger than the previous two and is designed to keep up (and overcome) visual transformers in image processing tasks. All this architecture has 5 downscale stages; the last stage, bottleneck, generally gives 1/32 of the original input resolution. The output from the feature extraction goes through the global average pooling layer, and the final (binary) prediction comes from a single neuron with sigmoid activation.

B. *Modification and additional CAM output*

The additional output is added to the generic binary image classifier for prediction explainability, which is used as rough segmentation/defect localization. A class activation map (CAM) [3] is utilized for it since it is one of the easiest ways of explaining the model requiring the least amount of modification in the model architecture with few additional calculations. Additionally, since there is only one neuron in the output (binary classification with sigmoid activation), this general explainability and rough defect estimation is constant and always dedicated to this specific prediction and its weights. The formula for forming CAM-based output is given in the following expressions (1):

$$\mathbf{X}_{CAM} = \mathbf{X}_{\text{Latent}} \cdot \mathbf{W}_{\text{B}}\,, \tag{1}$$

where $\mathbf{X}_{CAM}$ – CAM output matrix which is equal to $\mathbf{X}_{\text{Latent}}$ – feature extractor latent space tensor before global average pooling and $\mathbf{W}_{\text{B}}$ – binary prediction output neuron weights dot product.

Besides an explainability output, the model architecture's final feature extraction stage is taken out for a few experimental trials that increase output resolution 2 times while output 1/16 of the original image input. Every tested configuration parameter count is given in TABLE II, where '_s' in the model's name stands for shortened model version with a higher output resolution. Additionally, floating point operations per second are given for each architecture (expressed in megaFLOPS - 10^6). As can be seen, additional CAM output does not severely increase overall operation in each neural network configuration.

TABLE II. CONFIGURATIONS PARAMETER COUNT

Model	Parameters	MFLOPS*
ConvNextTiny	27,820,897	1427.881
ConvNextTiny + CAM	27,820,897	1427.893
ConvNextTiny_s	12,348,385	1182.035
ConvNextTiny_s + CAM	12,348,385	1182.059
EfficientNetB0	5,920,593	236.641
EfficientNetB0 + CAM	5,920,593	236.662
EfficientNetB0_s	1,470,661	178.607
EfficientNetB0_s + CAM	1,470,661	178.650
MobileNetV3	2,997,313	71.645
MobileNetV3 + CAM	2,997,313	71.660
MobileNetV3_s	882,265	55.334
MobileNetV3_s + CAM	882,265	55.377

**Input size 128x128x3*

C. Training and evaluation

Models pre-trained on the ImageNet1K database [1] were picked for all experiments. Models were trained as binary defect classifiers for each defect dataset. Training parameters for each dataset are shown in TABLE III.

TABLE III. TRAINING PARAMETERS

	Oliena	*BSData*	*PCB defects*
Learning rate	10^{-3}, except 10^{-4} with ConvNextTiny		
Warmup epochs	8	2	2
Learning rate scheduler	Cosine decay		
Optimizer	AdamW		
Training epochs	50	15	15
Batch size	4	16	8
Trains samples	361	5518	13480
Input size	752x480	384x384	640x640
Augmentation in the training pipeline	Rotation (±5°), shift (±10%) – p=0.5; random brightness (±10%) and gamma (±10%) – p=0.2; flip – p=0.5	Shift (±5%) – p=0.5; random brightness (±10%) and gamma (±10%) – p=0.2; flip – p=0.5	Shift (±10%) – p=0.5; random brightness (±10%) and gamma (±10%) – p=0.2; flip – p=0.5

A data augmentation pipeline was made for the image data loader with the given probabilities (p) for each augmentation technique. Training datasets are shuffled at the beginning of the epoch. The best weights are picked according to the highest accuracy yield on the test set.

Additional explainability evaluation with annotation presented in the original datasets labeling is made by rescaling it to the original input dimension and normalizing the CAM output to the range 0 – 255. Output thresholded with a lower intensity value of 127. This output label is only considered when the binary prediction is more than 0.5 – if the model predicted the given image bounding area as defective. Otherwise, it is black (defect-free).

Experiments are done with NVidia A100 GPU, 48GB RAM, and 16CPU cores machine in Ubuntu 20.04 environment with Tensorflow 2.12.0.

V. RESULTS

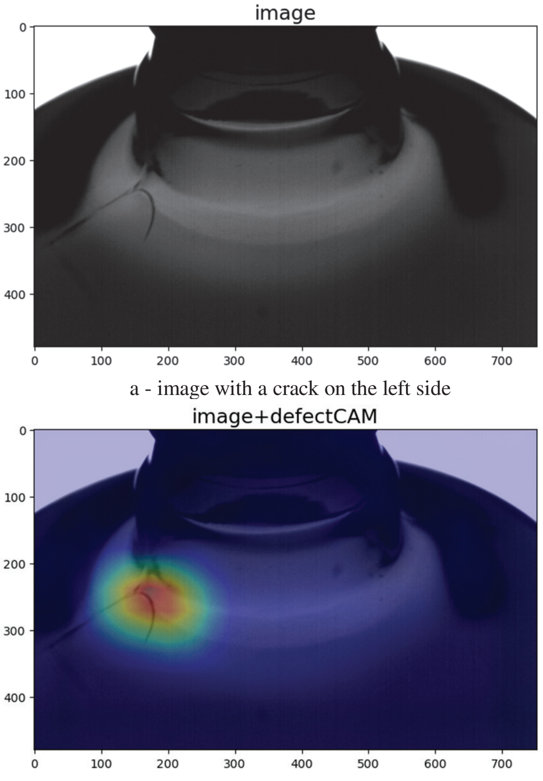
Defect binary classification accuracy for each database is given in TABLE IV. '**_s**' in the model's name stands for shortened model version with a higher output resolution.

TABLE IV. BINARY DEFECT CLASSIFICATION ACCURACY

Model	Oliena	BSData	PCB defects
ConvNextTiny	0.974	**0.979**	0.982
ConvNextTiny_s	**0.981**	0.978	0.982
EfficientNetB0	0.949	0.973	**0.983**
EfficientNetB0_s	0.930	0.973	0.981
MobileNetV3	0.962	0.975	0.982
MobileNetV3_s	0.942	0.964	0.981

The most significant difference between models is in the glass defect classification, where *Convnext_s* surpasses the second-best solution by almost 2% in accuracy. However, in other datasets, architectures' differences become marginal and insignificant. Even lightweight solutions, such as MobileNetV3, can keep up with more parametrized solutions.

Aside from the generic classification, defect localization from explainability through CAM output is considered. The model tends to react only to the defective area while interpreting the sample with the defect. A few samples are the following image where regular EfficientNetV2B0 defect prediction (p=0.9473) with CAM output is shown in Figure 2.

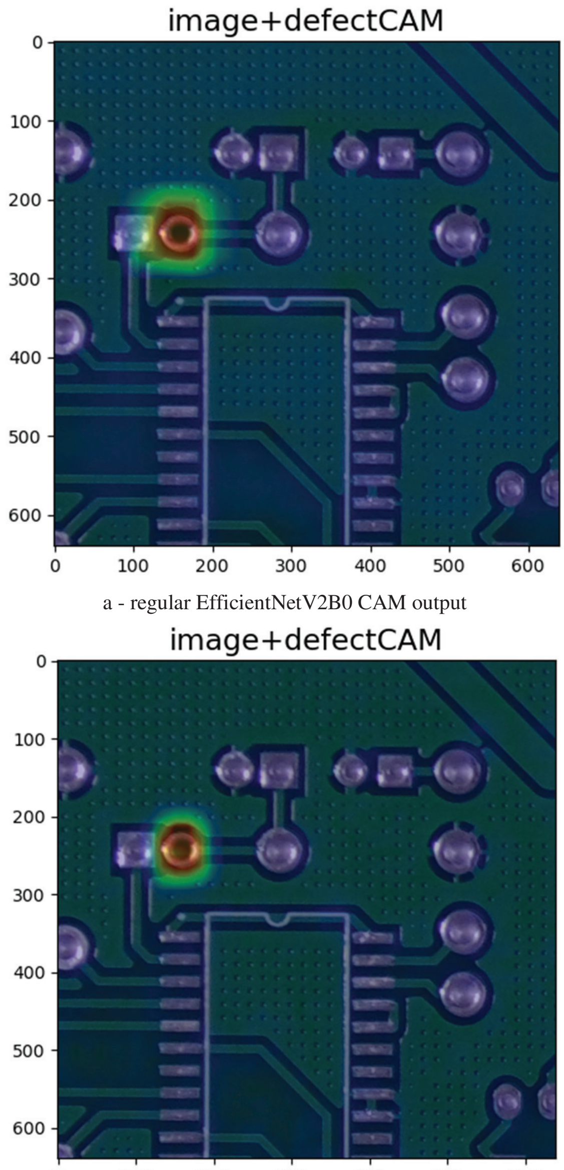


a - image with a crack on the left side

b – image with expanded normalized CAM output

Figure 2. Glass bottle defect (*Oliena*) detection with EffiecientNetV2B0 model enhanced with CAM output (a and b)

Another case on the PCB highlights the different precision of regular and shortened versions of the model since they present different resolutions in the image. The better model's ability to react to the defect details can be seen with a higher resolution in CAM output (Figure 3).

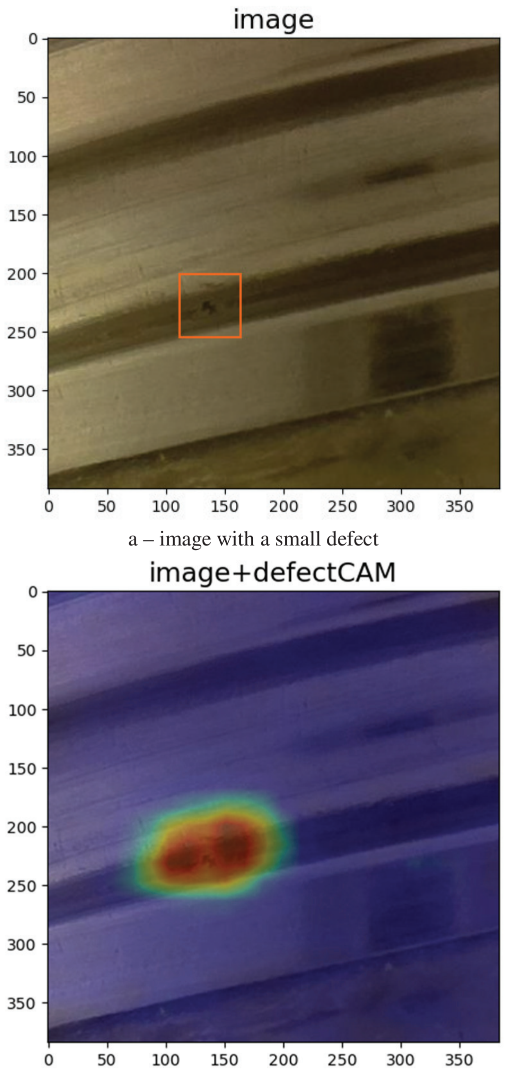


a - regular EfficientNetV2B0 CAM output

b – shortened EffiecientNetV2B0 CAM output

Figure 3. Regular and shortened EfficientNetV2B0 CAM outputs on PCB dataset (a and b). Shortened version displays a slightly better reaction exactly to the defect place

However, not all the time models CAM display properties to react to exact defect spots. In a few experiments, it was encountered that even the model version with higher CAM output resolution still considers way more area as defective than it is (Figure 4).



a – image with a small defect

b - image with expanded normalized CAM output

Figure 4. *BSData* dataset defective image with a small defect and shortened version of ConvNextTiny_s CAM output. The model tends to react to significantly more area around the defect

Prediction and label overlap experiment statistical evaluation is given in TABLE V, TABLE VI, and TABLE VII, where the column labeled '*Cfg.*' refers to the specific model architecture (configuration) tested 1-ConvNextTiny, 2 - ConvNextTiny_s, 3 - EfficientNetB0, 4 - EfficientNetB0_s, 5 - MobileNetV3, 6 - MobileNetV3_s.

TABLE V. *OLIENA* DEFECT LOCALIZATION IN THE IMAGE STATISTICAL EVALUATION

Cfg.	**Acc**	**Pre**	**Rec**	**F1**	**IoU**
1	0.857	0.21	**0.95**	0.24	0.20
2	0.953	0.29	0.93	0.34	0.27
3	0.959	0.26	0.88	0.32	0.24
4	0.980	0.40	0.80	0.42	0.32
5	0.978	0.36	0.85	0.41	0.31
6	**0.981**	**0.41**	0.82	**0.44**	**0.33**

TABLE VI. *PCB DEFECTS* LOCALIZATION IN THE IMAGE STATISTICAL EVALUATION

Cfg.	Acc	Pre	Rec	F1	IoU
1	0.923	0.74	**0.96**	0.72	0.70
2	**0.998**	**0.95**	0.90	**0.90**	**0.87**
3	0.996	0.87	0.94	0.87	0.82
4	0.997	0.90	0.92	0.88	0.84
5	0.997	0.89	0.93	0.88	0.84
6	0.997	0.89	0.93	0.88	0.84

TABLE VII. *BSDATA* DEFECT LOCALIZATION IN THE IMAGE STATISTICAL EVALUATION

Cfg.	Acc	Pre	Rec	F1	IoU
1	0.927	0.73	**0.99**	0.75	0.73
2	0.984	0.80	0.96	0.82	0.78
3	0.992	0.85	0.96	0.87	0.83
4	0.991	0.83	0.97	0.85	0.82
5	**0.995**	**0.88**	0.95	**0.89**	**0.85**
6	0.987	0.82	0.94	0.83	0.79

MobileNetV3_s showed the lowest results IoU (0.33) when tested with the *Oliena* set. These datasets consist of a narrow label that the model failed to recreate (react on to narrow crack, see Figure 2). On the other hand, the ConvNextTiny_s model produced an IoU value of 0.87 when evaluated with a PCB dataset, while the MobileNetV3 model achieved an IoU value of 0.85 when tested with a *BSData* dataset. In the PCB dataset, this modification yields a slightly more precise reaction to the defect since the defect is round or rectangular, making the generated CAM output overlap in a bigger area with the label. Notably, downscaled versions of CNN models with a higher resolution output from the CAM demonstrated better defect segmentation accuracy in two cases.

VI. CONCLUSION

This paper introduced lightweight defect localization through a simple class activation map (CAM) approach, which empowered rough defect localization in the image by modifying the binary classification model. The investigation was made on glass bottles, PCB, and industrial machine tool component surfaces (ball screw drive spindles) defects datasets with popular neural network architectures. Also, these models were modified by taking out one downscale stage, which gave two times higher resolution in the CAM output. The presented approach was not able to highlight a narrow defect; rather, it reacted to more areas around them. Still, it can be used as an approximate defect position estimator without pixel-wise annotation and labels required.

Future work might explore alternative solutions, such as visual transformers or combining methods with a higher resolution receptive field that might yield better segmentation in narrow defects cases.

REFERENCES

[1] X. Xu, Y. Lu, B. Vogel-Heuser, and L. Wang, "Industry 4.0 and Industry 5.0—Inception, conception and perception," *J Manuf Syst*, vol. 61, pp. 530–535, Oct. 2021, doi: 10.1016/j.jmsy.2021.10.006.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2009, pp. 248–255. doi: 10.1109/CVPR.2009.5206848.

[3] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning Deep Features for Discriminative Localization," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.

[4] R. Augustauskas, "Code implementation," Apr. 16, 2023. https://github.com/rytisss/DL-defect-classification-with-CAM-output/ (accessed Apr. 16, 2023).

[5] T. Czimmermann *et al.*, "Visual-based defect detection and classification approaches for industrial applications—A SURVEY," *Sensors (Switzerland)*, vol. 20, no. 5, pp. 1–25, 2020, doi: 10.3390/s20051459.

[6] A. Urbonas, V. Raudonis, R. Maskeliunas, and R. Damaševičius, "Automated identification of wood veneer surface defects using faster region-based convolutional neural network with data augmentation and transfer learning," *Applied Sciences (Switzerland)*, 2019, doi: 10.3390/app9224898.

[7] P. Kodytek, A. Bodzas, and P. Bilik, "A large-scale image dataset of wood surface defects for automated vision-based quality control processes," *F1000Res*, vol. 10, p. 581, Jun. 2022, doi: 10.12688/f1000research.52903.2.

[8] E. Versini, L. Snidaro, and A. Liani, "SCL—Segmentation–Classification combined Loss for surface defect detection," *Expert Syst Appl*, vol. 198, p. 116710, Jul. 2022, doi: 10.1016/j.eswa.2022.116710.

[9] H. Üzen, M. Turkoglu, M. Aslan, and D. Hanbay, "Depth-wise Squeeze and Excitation Block-based Efficient-Unet model for surface defect detection," *Vis Comput*, Mar. 2022, doi: 10.1007/s00371-022-02442-0.

[10] J. Kim, J. Ko, H. Choi, and H. Kim, "Printed Circuit Board Defect Detection Using Deep Learning via A Skip-Connected Convolutional Autoencoder," *Sensors*, vol. 21, no. 15, p. 4968, Jul. 2021, doi: 10.3390/s21154968.

[11] Y. Liu, H. Xiao, J. Xu, and J. Zhao, "A Rail Surface Defect Detection Method Based on Pyramid Feature and Lightweight Convolutional Neural Network," *IEEE Trans Instrum Meas*, vol. 71, pp. 1–10, 2022, doi: 10.1109/TIM.2022.3165287.

[12] P. Bergmann, K. Batzner, M. Fauser, D. Sattlegger, and C. Steger, "The MVTec Anomaly Detection Dataset: A Comprehensive Real-World Dataset for Unsupervised Anomaly Detection," *Int J Comput Vis*, vol. 129, no. 4, pp. 1038–1059, Apr. 2021, doi: 10.1007/s11263-020-01400-4.

[13] T. Schlagenhauf and M. Landwehr, "Industrial machine tool component surface defect dataset," *Data Brief*, vol. 39, p. 107643, Dec. 2021, doi: 10.1016/j.dib.2021.107643.

[14] R. Ding, L. Dai, G. Li, and H. Liu, "TDD-net: a tiny defect detection network for printed circuit boards," *CAAI Trans Intell Technol*, vol. 4, no. 2, pp. 110–116, Jun. 2019, doi: 10.1049/trit.2019.0019.

[15] A. Howard *et al.*, "Searching for MobileNetV3," *CoRR*, vol. abs/1905.02244, 2019, [Online]. Available: http://arxiv.org/abs/1905.02244

[16] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE, Jun. 2018, pp. 7132–7141. doi: 10.1109/CVPR.2018.00745.

[17] M. Tan and Q. V Le, "EfficientNetV2: Smaller Models and Faster Training," in *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, M. Meila and T. Zhang, Eds., in Proceedings of Machine Learning Research, vol. 139. PMLR, 2021, pp. 10096–10106. [Online]. Available: http://proceedings.mlr.press/v139/tan21a.html

[18] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022