Metal Diaphragm Seal Defect Detection Using Convolutional Neural Networks

1st Syed Muhammad Kazim
Kyung Hee University
Seoul, South Korea

2nd Jawad Tayyub
Endress + Hauser
Maulburg, Germany
jawad.tayyub@gmail.com

3rd Muhammad Sarmad
NTNU
Trondheim, Norway
muhammad.sarmad@ntnu.no

4th Patrick Werner
Endress + Hauser
Maulburg, Germany
patrick.werner@endress.com

Abstract—Vision-based defect detection of manufactured parts is a process that largely requires manual inspection and verification. This is a laborious task which uses precious worker time as well as being repetitive and mundane. We tackle the problem of fine-grained defect detection on shiny and reflective metal diaphragm seals. We propose to use a convolutional neural network (CNN) that can detect defects on images of metal surfaces with concentric rings texture. Our proposed pipeline consists of a localization and a detection stage. The localization module isolates a circular region of interest (ROI) in an image by using a CNN-based circle detector. A detection module then predicts a segmentation map of the ROI by segmenting out each detected defect. Defected parts are then identified by aggregating defected segments and thresholding total defected area to a certain value. This value is chosen by domain experts. The developed system is deployed in a real industrial setting and continuous worker feedback is used for evaluation. We demonstrate the superiority of our approach through extensive experimentation and ablations studies.

Index Terms—Defect Detection, Neural Networks, Industrial Data

I. INTRODUCTION

Manufacturing industry hold high potential for the application of artificial intelligent (AI) technologies. With the introduction of the fourth industrial revolution, Industry 4.0, AI and and machine learning (ML) are being used increasingly for automation of various manual tasks. One such task is inspection of manufactured parts for defects [1]. Defect detection is one of the most common activities during a production process. Defects are defined as any damage to the part during production. There are often multiple functional tests in place within a process to detect defected parts, however, minute cosmetic defects such as scratches or blemishes do not affect the functionality of the part and therefore have to be manually inspected. This is a time consuming and laborious task. Moreover, it is not possible to achieve consistent quality management since inspection criteria is different for each worker and dependent on "human" factor such as mood, tiredness and boredom. Therefore an automated inspection system is desirable.

We propose a convolutional neural network based approach to automatically classify parts as defected or nominal. Our problem is challenging since we tackle shiny metal membranes which are highly reflective and prone to lighting artifacts. We adopt a [2] to the membrane dataset and show the efficacy

of this architecture in relation to multiple others. We further apply explainable AI techniques, namely GradCAM and RISE, to build trust and faith in the model [3]–[5]. We find that our method saves time, introduces consistency and generally enhances the inspection process resulting in more reliable products and higher customer satisfaction.

In our problem, we require no specialized hardware, and any arbitrarily shaped scratch visible to the human eye counts as a defect.

II. RELATED WORK

Over the past decade, deep learning frameworks have been applied successfully to a myriad of problems including, but not limited to, classification, detection and segmentation [6]–[8]. Now deep learning has become a commercial necessity due to its ability to automate mundane processes. Inspired by these successes, we tailor popular networks for the classification of defective membranes, removing human input entirely. This problem persists especially in manufacturing industries, where defect detection is still an expensive process requiring human intervention.

Historically, and in some industries to this day still, defect detection has been carried out by human experts. Due to the laborious nature of this work, some degree of automation is required. In early works, images were processed and analysed using conventional image processing methods. These include edge detection algorithms, thresholding gray-scale images and image segmentation [9], [10]. However, most of these techniques are only effective for specialized cases such as where the shapes of defects must be consistent.

The major drawback of this family of methods is that they are not robust against noise or variation in the input, which exist very commonly in an industrial setting. Classical machine learning techniques are a better alternative because they offer more robustness to noise. Zhang et al. [11] proposed multi-calss SVM for the detection of surface defecst in shiny metals. For an agriculture relate application, Qingzhong et al. [12] applied small neural networks to detect and segment defects in apples. Iker et al. [13] deployed decision tress to recover surface information as a step in high-precision foundry. However, all these methods rely on hefty feature extraction and/or pre-processing and still do not generalize sufficiently well to unseen data. This is best exemplified in
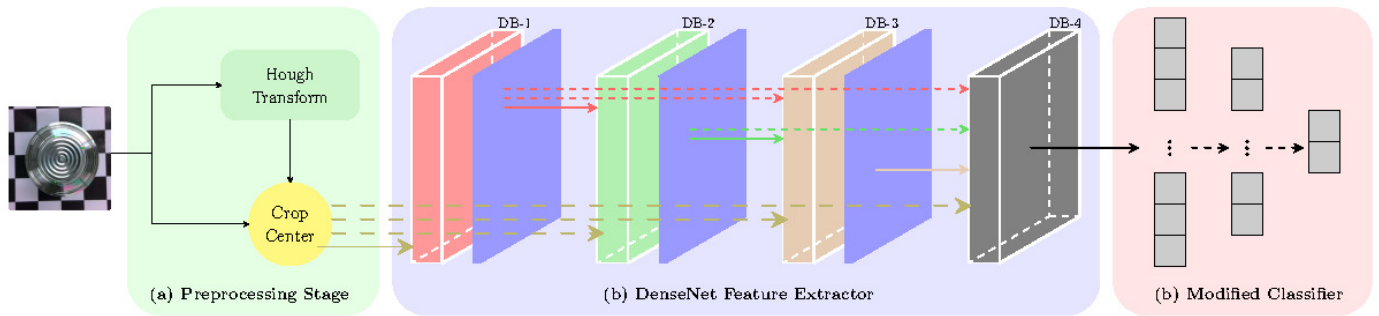
Fig. 1. Topology of the proposed method. (a) A raw input image is centered and cropped to fixed dimensions using the coordinates of the membrane center found by applying Hough transform. (B) The DenseNet feature extractor consists of four dense blocks (DB), transition layers (blue blocks), and direct and skip connections represented with solid and dashed arrows. The black arrow represents a 7×7 global average pool followed by a vectorization of the 2-D array [ref]. (c) The modified classifier is a fully-connected neural network with the last two layers of sizes 256 and 2, respectively. The model outputs the probability of the membrane in the image being defective.

[11], which achieves a training accuracy of 100% but a testing accuracy of only 85%.

In recent times, deep learning has been applied successfully for facial recognition, fault detection, classification, segmentation, and a host of image-related applications. Deep learning has been proven to be robust against lightning, backgrounds, shapes and sizes, in its ability to recognize patterns in images [14]. These qualities are pivotal for defect detection and were left desired by the methods before.

There has also been extensive research on the topic of defect detection using deep learning. However, most of it has been for highly specialized cases including pavement defects [15], fabric defects [16], and other industrial applications [12]. A more relevant and recent work by Zhufeng et al. [17] proposed the use of Mask R-CNN for the detection and segmentation of defects in architectural glass panels. They reported 96.5% mean average precision (mAP) with IOU50. However, they require specialized hardware to take microscopic images of the panels and their mAP values reduced to 65.3% with IOU75.

III. METHODOLOGY

Our pipeline is developed for the automatic classification of good and defective of shiny and reflective diaphragm seals. A fixed camera unit in an industry is set up which takes full-size images of single units of membranes. In the pre-processing stage, we deduce the center of the concentric circles on the surface of the membranes using Hough transform to center the membranes in the images. We crop the output such that the inputs to our network are square images. Our pipeline consists of a preprocessing stage followed by a classification module $\mathcal{C}$ as demonstrated in Fig. 1.

A. Defect Classification $\mathcal{C}$

This module functions as a binary classifier that classifies a membrane as either good or defective. Input to this module is a centered image with dimension $H \times W \times 3$, and output is a label 1 or 0, specifying good and defective, respectively. A membrane is considered defective if there exists even a single scratch on its surface, irrespective of the shape and size, visible to the human eye.

For this module, we use a DenseNet-121 [2], [18], pretrained on ImageNet with a top-1 error of 25.35%. We modify the DensNnet classifier by supplanting with our own: Two fully-connected (FC) layers $(1024, 256) - (256, 2)$. The first FC is followed by a ReLU activation function and a dropout with probability 0.25, while the second is followed by a LogSoftmax function. The output of the module is given as

$$p_i = \mathcal{C}\left(I_i\right), \tag{1}$$

where I_i is the index of the ith input image and p_i is a two-element prediction vector.

Our choice of a pretrained DenseNet allows the training of our network to be relatively inexpensive and quick. Moreover, we retain the benefits of the densely connected convolutional network. In our training process, we only optimize the weights of the new classifier. The negative log-likelihood loss function $\mathcal{L}$ used to train this architecture is defined as [19]:

$$\mathcal{L}\left(y\right) = -\frac{1}{N} \sum_{j=1}^{N} \log\left(p_j\left[y\right]\right), \tag{2}$$

where $y = \arg\max p_j$ and N is the batch size.

IV. EXPERIMENTS

We used PyTorch [18] for our implementation. All networks were trained on a single Nvidia RTX 2060 (?) graphics card.

A. Dataset

For this work, we introduce a membrane defect detection (MDD) dataset $\mathcal{D}$, which consists of 400 nominal and 400 defective membranes of size 1920×1080, hereby referred to as subset A and B, respectively. For subset B. we use the web-based open source Computer Vision Annotation Tool (CVAT) [20] to annotate the region of interest (ROI) using three points as shown in Fig. 2. These annotations are verified by industry experts.
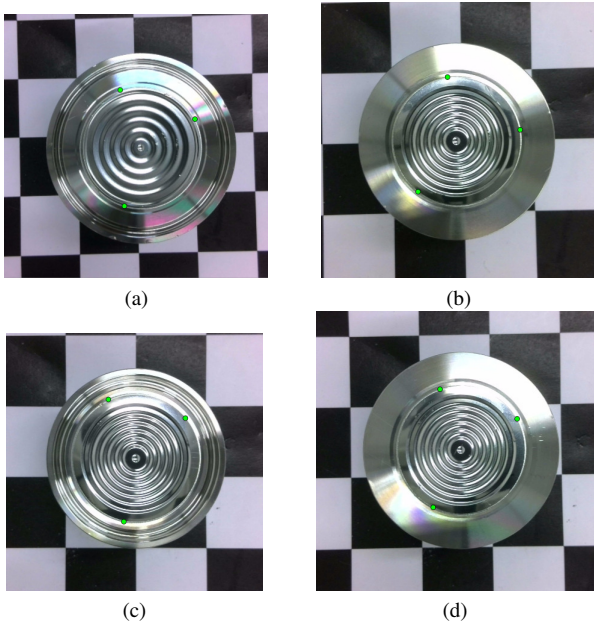
Fig. 2. Dataset $\mathcal{D}$ with green dots' annotations for the region of interest.

As a pre-processing step, we apply the Hough transform on each element of $\mathcal{D}$ to identify the center c_i for $i \in \{1, 2, \cdots, 800\}$ of the concentric rings. Using the annotations of B, we also found the maximum diameter $d_{\max}$ of the ROI of the membrane to be less than 1200 pixels. Using c_i and $d_{\max}$, we center and crop our raw images. As a final step, we pad zeros to each image to ensure each element is sized 1200×1200. We notate this dataset with $\mathcal{D}_1$. We also use the ROI annotations om $\mathcal{D}_1$ to extract the ROI from the background. We notate this dataset with $\mathcal{D}_2$.

B. Evaluation Metric

We report the training loss with respect to epochs, and the cross validation classification accuracy and the confusion matrices of the test sets for both $\mathcal{D}_1$ and $\mathcal{D}_2$. Moreover, we also provide GradCAM and RISE results to help inspect the focus of our networks when trained with $\mathcal{D}_1$ and $\mathcal{D}_2$.

V. Results and Discussion

We used datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ to train module $\mathcal{C}$, separately. We divide each dataset using a $90 - 10$ train-test split. Thus our testset consists of 80 randomly selected images from the dataset. Of the training set, we allocate 10% of the datapoints for validation of the model during training.

Fig. 3 and 4 demonstrate the training and validation losses for $\mathcal{C}$ with datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively. In all experiments considered in this paper, we train our model for 400 epochs, save the parameters with the best validation loss, and evaluate the accuracy of the testset on these parameters. The test accuracy of $\mathcal{C}$ is 96.5% with $\mathcal{D}_1$ and 97.0% with $\mathcal{D}_2$.

We also explore explainable AI techniques, namely, Grad-CAM and RISE to perform a more nuanced analysis of the networks. Both these techniques indicate that our model assigns notably more significance to the ROI as exemplified
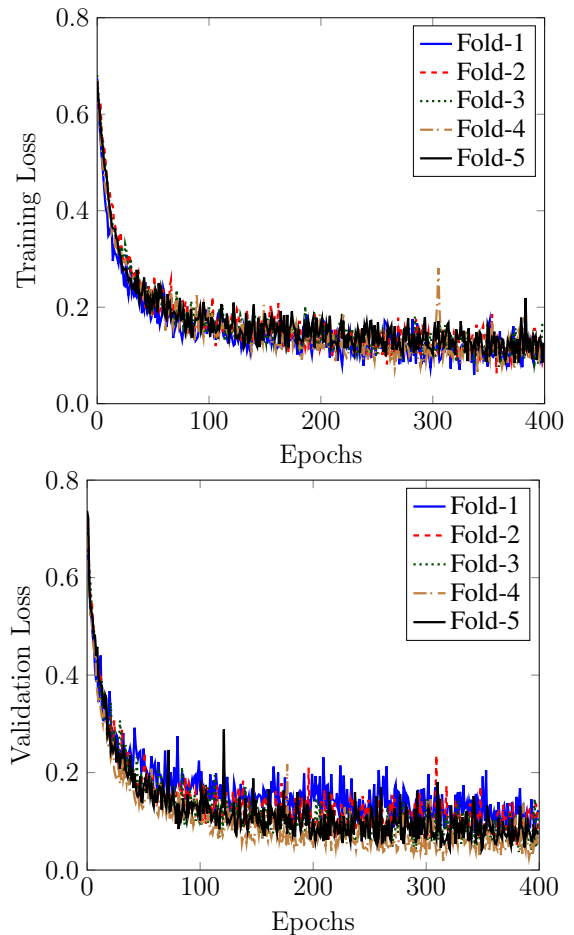

Fig. 3. 5-fold cross-validation losses when $\mathcal{C}$ is trained for 400 epochs using dataeset $\mathcal{D}_1$.

by Fig. 5. Moreover, in the absence of the background as in $\mathcal{D}_2$, $\mathcal{C}$ tends to concentrate more on the defective regions.

Considering the advantages offered by $\mathcal{D}_2$, we train multiple models on this dataset. The fully-connected layers of these models have been replaced with our modified classifier. We note that VGG-19 and DenseNet-201 offer slight improvements over DenseNet-152, while the remaining models perform poorly. The cross-validation accuracy of these models is summarized in Table I.

TABLE I
PERFORMANCE OF MODELS ON $\mathcal{D}_2$

Models	Accuracy (%)
DenseNet-121	97.00
DenseNet-201	98.00
VGG-19	98.50
ResNet-152	79.00
EfficientNet	67.50

VI. Conclusion

In this paper, we address the task of automatic detection of arbitrarily shaped defects using a customized deep learning ar-
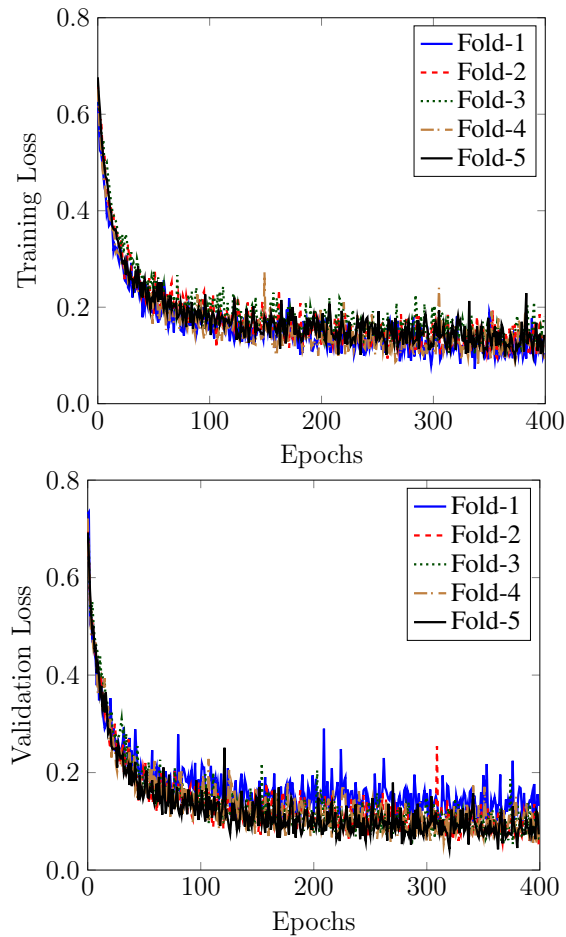
Fig. 4. 5-fold cross-validation losses when $\mathcal{C}$ is trained for 400 epochs using dataeset $\mathcal{D}_2$.

Fig. 5. Explainability of $\mathcal{C}$. Accentuated white annotations are added to demonstrate the shape and location of the defects on the membranes. Regions with most contribution to the outcome are delineated in yellow. $(a) - (b)$ Visualisations of GradCAM and RISE, respectively, on model trained on $\mathcal{D}_1$, and $(c) - (d)$ for model trained on $\mathcal{D}_2$.

chitecture. Substantial portions of these models are pre-trained, and only the parameters of our classification sub-module is optimized, making are method considerably resource efficient and computationally inexpensive, We have also demonstrated that our pipeline supports multiple deep learning architectures. Moreover, we have presented a solution to a specific, real-world laborious problem, which was not dealt with previously. Our topology performs just as well or better than contemporary deep learning solutions for similar tasks. We further provide a layer of interpretability of our framework using explainable AI techniques, and utilize the insights for comparison of the datasets. Lastly and most importantly, the high degree of accuracy without the need for specialized equipment makes our framework readily deployable in industries saving invaluable time and cost.

REFERENCES

[1] J. Yang, S. Li, Z. Wang, and G. Yang, "Real-time tiny part defect detection system in manufacturing using deep learning," *IEEE Access*, vol. 7, pp. 89 278–89 291, Jun. 2019.

[2] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
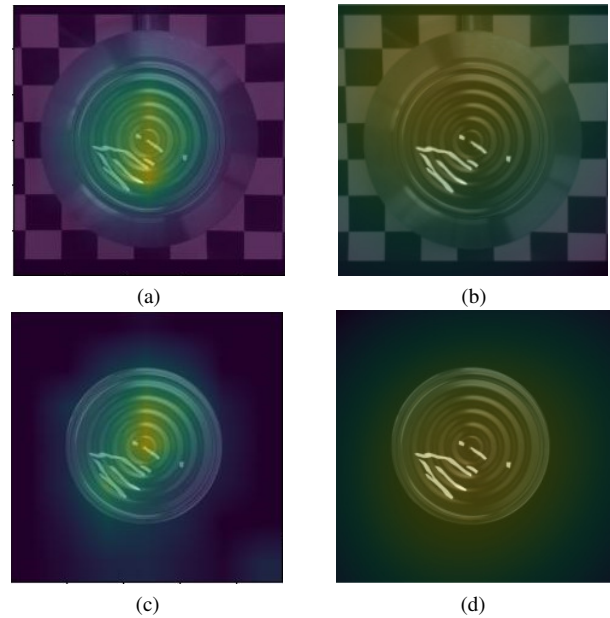
[3] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv:1806.07421*, Sep. 2018.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and 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.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, Aug. 2016, pp. 1135–1144.

[6] R. Huang, J. Pedoeem, and C. Chen, "YOLO-LITE: a real-time object detection algorithm optimized for non-GPU computers," in *2018 IEEE International Conference on Big Data*. IEEE, Dec. 2018, pp. 2503–2510.

[7] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020, pp. 1055–1059.

[8] M. F. Ercan, A. L. Qiankun, S. S. Sakai, and T. Miyazaki, "Circle detection in images: A deep learning approach," in *Global Oceans 2020: Singapore–US Gulf Coast*. IEEE, Apr. 2020, pp. 1–5.

[9] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognit. Lett.*, vol. 27, no. 14, pp. 1644–1649, Oct. 2006.

[10] V. Bruni and D. Vitulano, "A generalized model for scratch detection," *IEEE Trans. Image Process.*, vol. 13, no. 1, pp. 44–50, Jan. 2004.

[11] Z. Xue-Wu, D. Yan-Qiong, L. Yan-Yun, S. Ai-Ye, and L. Rui-Yu, "A vision inspection system for the surface defects of strongly reflected metal based on multi-class SVM," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 5930–5939, Nov. 2011.

[12] Q. Li, M. Wang, and W. Gu, "Computer vision based system for apple surface defect detection," *Computers and electronics in agriculture*, vol. 36, no. 2-3, pp. 215–223, Nov. 2002.

[13] I. Pastor-López, I. Santos, A. Santamaría-Ibirika, M. Salazar, J. D. la Peña-Sordo, and P. G. Bringas, "Machine-learning-based surface defect detection and categorisation in high-precision foundry," in *2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA)*. IEEE, Nov. 2012, pp. 1359–1364.

[14] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard, "The robustness of

deep networks: A geometrical perspective," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 50–62, Nov. 2017.

[15] W. Cao, Q. Liu, and Z. He, "Review of pavement defect detection methods," *IEEE Access*, vol. 8, pp. 14 531–14 544, Jan. 2020.

[16] A. Kumar, "Computer-vision-based fabric defect detection: A survey," *IEEE Trans. Ind. Electron.*, vol. 55, no. 1, pp. 348–363, Jan. 2008.

[17] Z. Pan, J. Yang, X. Wang, J. Liu, and J. Li, "Surface scratch detection of monolithic glass panel using deep learning techniques," in *International Conference on Computing in Civil and Building Engineering*. Springer, Jul. 2020, pp. 133–143.

[18] A. Paszke, S. Gross, F. Massa, A. Lerer *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[19] S. Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. IEEE, Dec 2020, pp. 1–7.

[20] B. Sekachev, N. Manovich, M. Zhiltsov, A. Zhavoronkov *et al.*, "opencv/cvat: v1.1.0," Aug. 2020.