TaylorMade Visual Burr Detection for High-mix Low-volume Production of Non-convex Cylindrical Metal Objects

Tashiro Kyosuke, Takeda Koji, Aoki Shogo, Ye Haoming, Hiroki Tomoe and Tanaka Kanji[a]

University of Fukui, 3-9-1, Bunkyo, Fukui, Fukui, Japan

Keywords: Network Architecture Search, Visual Burr Detection, High-mix Low-volume Production, Non-convex Cylindrical Metal Objects.

Abstract: Visual defect detection (VDD) for high-mix low-volume production of non-convex metal objects, such as high-pressure cylindrical piping joint parts (VDD-HPPPs), is challenging because subtle difference in domain (e.g., metal objects, imaging device, viewpoints, lighting) significantly affects the specular reflection characteristics of individual metal object types. In this paper, we address this issue by introducing a tailor-made VDD framework that can be automatically adapted to a new domain. Specifically, we formulate this adaptation task as the problem of network architecture search (NAS) on a deep object-detection network, in which the network architecture is searched via reinforcement learning. We demonstrate the effectiveness of the proposed framework using the VDD-HPPPs task as a factory case study. Experimental results show that the proposed method achieved higher burr detection accuracy compared with the baseline method for data with different training/test domains for the non-convex HPPPs, which are particularly affected by domain shifts.

1 INTRODUCTION

In this study, we address the problem of visual defect detection (VDD) for high-mix low-volume production of non-convex metal objects (Fig. 1), such as high-pressure cylindrical piping joint parts (VDD-HPPPs). At automatic metal processing site, when drilling holes in metal using a robot hand, defects called burrs occur. The presence of these burrs often causes scratches and cuts on the hands, which deteriorates safety and affects the accuracy of the product. As HPPPs are produced in small lots, the burr inspection process is not fully automated and demands manual effort. In some factories, the visual inspection is carried out by more than six workers, for 18 hours a day, which is laborious and costly. To address this, automatic eye-in-hand VDD presents a promising solution to this problem.

Automatic VDD on metal objects has been a long standing issue in machine vision literature (Chin and Harlow, 1982) and has been energetically studied in various places (Czimmermann et al., 2020; Kumar, 2008; Xie, 2008; Huang and Pan, 2015; Newman and Jain, 1995; Neogi et al., 2014). In recent years,

there have been attempts to use deep learning for this problem. In (Natarajan et al., 2017), a flexible multi-layer deep feature extraction method based on CNN via transfer learning was developed to detect anomalies. In (Cha et al., 2018), structural damage detection method based on Faster R-CNN was developed to address the issues of object size variation, overfitting, and specular reflection. In (Zhao et al., 2020), remarkable progress was made in detecting corrosion of metal parts such as bolts.

Majority of these existing VDD approaches targeted convex metal objects such as flat steel surface (Luo et al., 2020). Therefore, it is often assumed that the risk of multiple reflections is low and simple foreground/background models were used. In contrast, the HPPPs targeted in this study are non-convex objects. Therefore, subtle difference between training and test domains has a large impact on foreground appearance (i.e., burrs) and background prior.

In this study, we address the above issue by introducing a tailor-made VDD framework (Fig. 2) that can be automatically adapted to a new domain (e.g., metal objects, device, viewpoints, lighting). Specifically, we formulate this adaptation task as the problem of network architecture search (NAS) (Elsken et al., 2019) on a deep object-detection network, in

[a] https://orcid.org/0000-0002-1143-5478

domain A
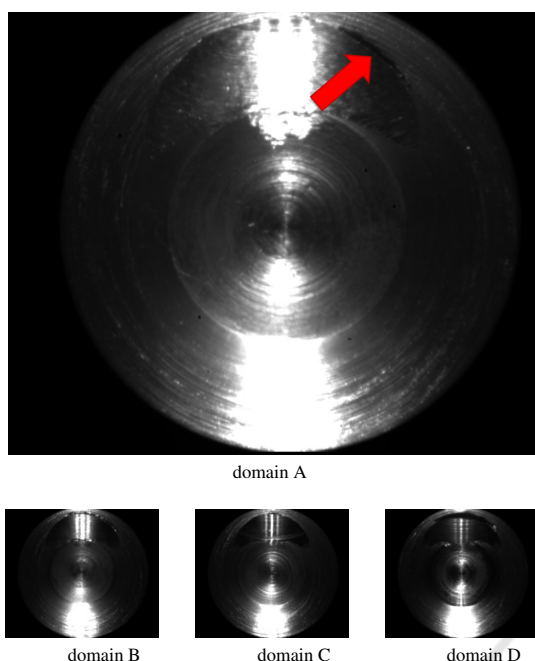

domain B domain C domain D

Figure 1: Visual burr detection in HPPPs.

which the network architecture is searched via rein-forcement learning. We demonstrate the effectiveness of the proposed framework by using the VDD-HPPPs task as a factory case study. Experimental results show that the proposed method achieved higher burr detection accuracy than the baseline method for data with different domains for the non-convex HPPPs, which are particularly affected by domain-shift.

2 VISUAL DEFECT DETECTION (VDD) PROBLEM

Figure 1 shows an example of HPPPs and images taken by the eye-in-hand in four different domains. The arrow indicates the position of the burr.

The HPPPs have a wide variety of registered products, about 20,000 items. The production quantity is mainly small lots, and the monthly production starts from one. This is very different from the conventional applications such as hydraulic parts and engine parts flowing on a dedicated line (Tumer and Bajwa, 1999). Existing image recognition softwares often require as many as seven days to adjust the program to adapt to a new domain, and it is necessary to change the specifications of parts and items. Therefore, they are not suitable for small lot products such as HPPPs.

In this study, we aim to achieve a good trade-off between the online VDD performance and the offline adaptation speed. To this end, it is necessary to ad-

dress the following issues. (1) The appearance looks similar between the burr area and the background area in the image (Fig. 1). (2) The shapes and sizes of burrs are diverse and not easy to generalize. (3) The burrs occur in non-convex cylindrical holes inside the joint parts (Fig. 2), and are thus affected by the dif-fused light reflection that is difficult to model. (4) Even when a calibrated eye-in-hand is used, the view-point can shift randomly up to about 3 pixels in terms of the image coordinate. To solve the above issues, a highly versatile and accurate machine vision method is needed.

3 PROPOSED APPROACH

The proposed approach consists of two distinctive stages: the offline-adaptation and online-detection stages. The adaptation stage is responsible for adapt-ing the deep neural network to a new domain, and it is the pipeline consisting of semi-automatic annotation, model-based coordinate-transformation, tailor-made network-architecture search, and network-parameter fine-tuning. The detection stage is responsible for de-tecting burrs inside a given image, and it consists of the coordinate-transformation and visual burr detec-tion. Hereinafter, each process will be described in detail.
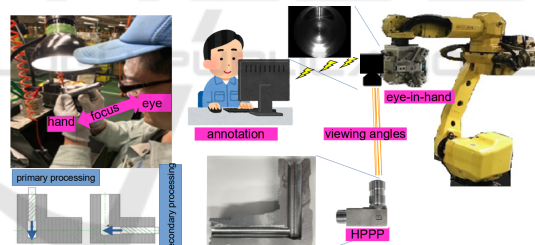

Figure 2: From manual to automatic VDD.

Annotation cost is a major part of the total cost required to adapt a VDD software to small-lot metal projects, such as HPPPs. In our case study's factory site, the annotation is provided by skilled workers in the form of bounding polygons, by using the LabelMe tool (Russell et al., 2008) (Fig. 2). As can be seen from Fig. 1, it would be difficult for a non-skilled per-son even to visually distinguish burrs from the back-ground textures. Surprisingly, skilled human work-ers often become able to distinguish burrs with 100% accuracy after sufficient time spent training (Fig. 3). This indicates that the VDD task may not be infeasi-ble, which has motivated us to develop an automatic VDD system.

We have been developing a user-interface for semi-automatic annotation in our factory site (Fig. 2).
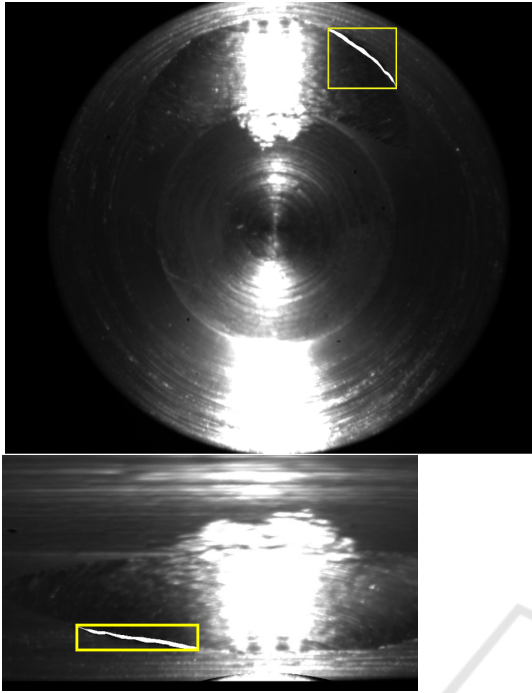
Figure 3: Ground-truth burrs and annotations.

Using it, given a 3D CAD HPPP model and a calibrated camera, a forward/backward projection model of the camera can be obtained. These models enable transferring an annotated bounding polygon in the image coordinate of one viewpoint to that of another viewpoint. This transfer technique eliminates the need for additional annotation per different observation condition on the same HPPP type, which leads to significant reduction in the total annotation cost.

Majority of state-of-the-art object detection networks assume bounding box-shaped annotation (Jiao et al., 2019; Liu et al., 2020; Zou et al., 2019). In contrast, the ground-truth (GT) of burr regions in an input image often have a crescent-like shape and do not fit well into bounding boxes. To solve this, we propose to transform the image coordinate system appropriately. In our specific case, such crescent-like burrs usually occur inside cylindrical HPPPs. Therefore, polar coordinate transformation with the cylinder center coordinates in this image as a hyper-parameter is performed (Fig. 3). Through preliminary experiments, we found that such a cylinder center in a given image can be stably and accurately predicted by using a RANSAC-based circle fitting to the cylinder border circle. Figure 3 shows the result of image transformation. Comparing before and after image transformation, it can be confirmed that the filling rate of the burr region with respect to the bounding box is higher after transformation. In the experiment, polar coordi-
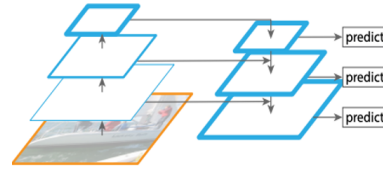

Figure 4: NAS-searched object-detector.

nate images with size 800×1333 pixels were used.

The proposed NAS framework is inspired by (Ghiasi et al., 2019), which we further developed by introducing the following two steps, which are iterated until the time budget is expired: (1) The Controller-RNN creates a candidate architecture and trains a child network with that architecture, and (2) The Controller-RNN is updated by policy gradient with rewards obtained from reinforcement learning. The objective of the proposed NAS is to find an optimal child network, so as to maximize the VDD performance in terms of domain adaptation and generalization. In this framework, a child network is described by a parameter variable that consists of a set of mutually connected network building blocks and their connection relationship (Fig. 4). A building block takes two feature maps from the backbone network (ResNet-50) as input, and integrates them using SUM or global pooling operation, into a new feature map. Thus, each block can be described by a triplet ID: a pair of input feature map IDs and an operation ID ($\in \{$SUM, POOLING$\}$). By definition, the input IDs chosen are not duplicated. In addition, the new feature map created by a block is regarded as an additional candidate input for future building blocks. Thus, the space of the triplet ID can increase as iteration proceeds. At each iteration, the newest feature map is considered as the output of a child network. In the experiments, a set of 4 feature maps P2, P3, P4 and P5, with resolutions (200, 334), (100, 167), (50, 84), and (25, 42), respectively, are used.

We observe that combining the low and high layer feature maps (i.e., combining primitive and semantic features) with NAS is often effective. The reason might be that in our application of VDD-HPPPs, the size of burrs has a large bias and the shape is not constant and thus, the high level semantic information plays an important role. The NAS efficiently searches over the exponentially large number of such combinations, and successfully finds an optimal one, as we will demonstrate in the experimental section.

The process described in Step-2 is detailed in the following. This process aims to maximize the expected reward $J(\theta_c)$ by updating the hyper-parameters θ_c of the Controller-RNN by policy gradient. $J(\theta_c)$ is defined by: $J(\theta_c) = E_{P(a_{1:T};\theta_c)}[R]$, where

$a_{1:T}$ is an action of Controller-RNN. Thus, it is updated by:

$$\nabla_{\theta_c} J(\theta_c) = \sum_{t=1}^{T} E_{P(a_{1:T};\theta_c)} \left[\nabla_{\theta_c} \log P(a_t | a_{(t-1):1}; \theta_c) R \right],$$

where $a_{1:T}$ is an action of Controller-RNN, $m = 1$ is the number of architectures that the Controller-RNN verifies in one mini-batch. T is the number of hyper-parameters to be estimated. R_k is the average-precision (AP) in the k-th architecture. b is the exponential moving average of the AP of the neural network architecture up to that point. The smoothing constant is 0.8. The learning rate is 0.1. The number of training iterations is 3000 per child network. The hyper-parameters in (Ghiasi et al., 2019) are set as follows: cfgs.LR = 0.001, cfgs.WARM-step = 750, and #trials = 500. The NAS with above setting consumes 2 weeks using a graphics processing unit (GPU) machine (NVIDIA RTX 2080). Figure 5 shows progress of NAS search.

The searched architecture is then used to train a burr detector. For the training, the Faster R-CNN algorithm is used. At this time, the parameters for training are set as followings. cfgs.LR=0.001. cfgs.WARM-step=2,500. Subsequent learning rates are reduced 1/10 times when the number of training sessions is 60,000, and 1/100 times when the number of training sessions is 80,000. The number of training iterations is 150,000. The time required is about 2 days using the above same GPU machine.

The detection process takes a query image and predicts bounding boxes of burr regions with a non-maxima suppression. The confidence score is evaluated as the highest probability values among all the classes in the Faster R-CNN. The computation speed per image was around 3 fps.

4 EXPERIMENT

The proposed tailor-made VDD framework has been evaluated using real HPPP dataset in four different domains that is collected on the target factory site. This section describes the dataset, the baseline method, experimental results, and provides discussions.

Figure 1 shows examples of image datasets. We collected four independent collections of images, A, B, C, and D at different domains, and manually annotated every image. The dataset size are 402, 396, 50, and 76 for A, B, C, and D, respectively. Only the sets A and B are large enough and are thus used for NAS, training and testing, while C and D are used only for testing.

The set A or B is split into 1:2:1 subsets namely, NAS, training, and evaluation subsets. The NAS subset is used for evaluating each child network during the NAS task. The union of the NAS and training subsets is used for training. The evaluation subset is used for performance evaluation on a trained NAS-searched VDD model. As aforementioned, every training/test image is transformed to polar coordinate before being input to the training or testing procedure. For NAS and training, a left-right flipping data augmentation is applied.

A deep object detector using a feature pyramid network (FPN) (Lin et al., 2017) is used as a baseline method. The FPN consists of three features, bottom-up direction, top-down direction, and potential connection, to the feature layers of different scales output by the convolutional neural network. This provides both low-resolution semantically strong features and high-resolution semantically weak features. A Faster R-CNN is used for detection task. Despite the efficiency, FPN is based on a manually designed architecture, and thus, it is not optimized for a given specific application.

Figure 5 shows NAS progress when subset A is used as the NAS subset. It was confirmed that the curve rises slightly as the trial proceeds, and that the Controller-RNN learned the generation of a better feature map over time through trial and error, and the search was performed adequately.

As shown in the Fig. 5, the NAS score converged at 110,000-th iteration. We use the architecture at this point to evaluate the VDD performance. Specifically, the convergence is judged if change in AP values between two consecutive training sessions is equal or lower than 0.01. For performance evaluation, average precision (AP) is evaluated for different threshold values on IoU, from 0.5 to 0.95 with 0.05 increment step, and then average of these AP values is used as the performance index.

Table 1 shows performance results.

First, let us discuss the results for trained NAS searched VDD model using the dataset A as the training set. From the evaluation result, it is confirmed that the proposed method provides better performance compared with FPN in all the test sets. Figure 6 shows example detection results for GT/FPN/Proposed (columns) for test domains A/B/C/D (rows) for training domains A/B (left/right), with bounding boxes whose confidence scores are higher than 0.5. Overall, the number of bounding boxes generated by the proposed method tended to be the same as that of the ground-truth. Exceptionally, for testing domain D, there were much false positives for both the proposed and FPN methods. This is
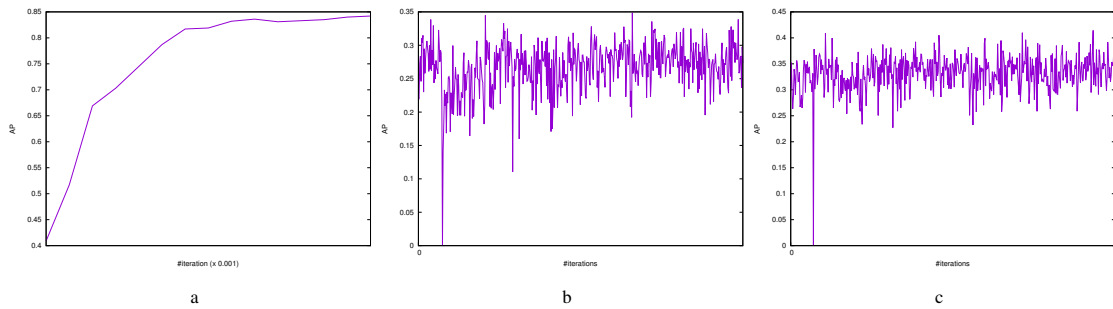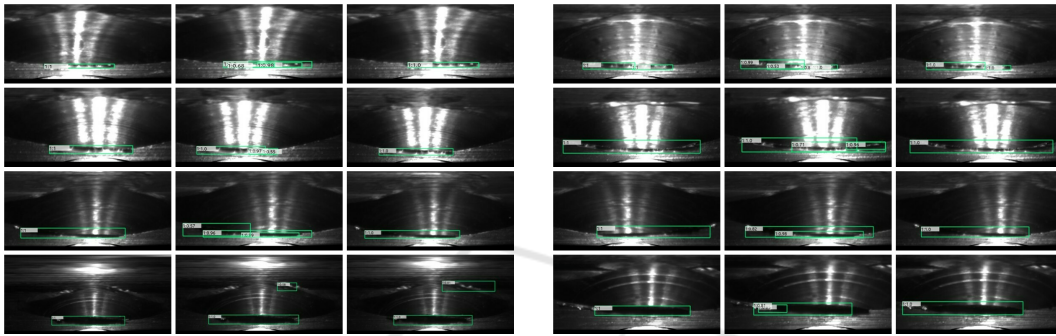
a   b   c

Figure 5: NAS progress.

Figure 6: VDD results.

Table 1: Average-precision performance [%].

training set	A				B			
test set	A	B	C	D	A	B	C	D
FPN	31.4	13.2	36.0	47.6	17.9	39.7	14.3	8.8
Proposed	31.7	30.6	48.9	58.8	33.4	52.1	41.1	44.5

mainly due to the dark lighting conditions. It should be noted that for FPN, the bounding boxes often do not appear, as the confidence score is often lower than 0.5.

Next, let us discuss the results for trained NAS-searched VDD model using dataset B as the training set. The results are reported in Table 1. For FPN, the domain B is challenging and thus its training result was difficult to generalize to other domains. Despite this, the proposed tailor-made VDD framework was able to predict with high accuracy for other domains as well.

From the above results, it could be concluded that VDD performance was significantly improved by the proposed taylor-made VDD despite the fact that the adaptation process is highly automated and efficient.

5 CONCLUSION

In this study, we presented a tailor-made visual defect detection framework that can be adapted to various domains. To the best of our knowledge, we are the first to formulate the VDD-HPPPs as an important and challenging new machine-vision application that is characterized by a combination of non-convex metal parts with complex specular reflections and high-mix low-volume production. In this study, we adapted the NAS technique to search for the optimal architecture of the network. Through a factory case study, we demonstrated that our approach is able to search for a versatile network architecture and enables us to detect burrs with higher accuracy compared with the baseline method under domain shifts.

ACKNOWLEDGEMENTS

REFERENCES

Cha, Y.-J., Choi, W., Suh, G., Mahmoudkhani, S., and Büyüköztürk, O. (2018). Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Computer-Aided Civil and Infrastructure Engineering*, 33(9):731–747.

Chin, R. T. and Harlow, C. A. (1982). Automated visual inspection: A survey. *IEEE transactions on pattern analysis and machine intelligence*, (6):557–573.

Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., and Dario, P. (2020). Visual-based defect detection and classification approaches for industrial applications-a survey. *Sensors*, 20(5):1459.

Elsken, T., Metzen, J. H., Hutter, F., et al. (2019). Neural architecture search: A survey. *J. Mach. Learn. Res.*, 20(55):1–21.

Ghiasi, G., Lin, T.-Y., and Le, Q. V. (2019). Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7036–7045.

Huang, S.-H. and Pan, Y.-C. (2015). Automated visual inspection in the semiconductor industry: A survey. *Computers in industry*, 66:1–10.

Jiao, L., Zhang, F., Liu, F., Yang, S., Li, L., Feng, Z., and Qu, R. (2019). A survey of deep learning-based object detection. *IEEE Access*, 7:128837–128868.

Kumar, A. (2008). Computer-vision-based fabric defect detection: A survey. *IEEE transactions on industrial electronics*, 55(1):348–363.

Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. (2020). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128(2):261–318.

Luo, Q., Fang, X., Liu, L., Yang, C., and Sun, Y. (2020). Automated visual defect detection for flat steel surface: A survey. *IEEE Transactions on Instrumentation and Measurement*, 69(3):626–644.

Natarajan, V., Hung, T.-Y., Vaikundam, S., and Chia, L.-T. (2017). Convolutional networks for voting-based anomaly classification in metal surface inspection. In *2017 IEEE International Conference on Industrial Technology (ICIT)*, pages 986–991. IEEE.

Neogi, N., Mohanta, D. K., and Dutta, P. K. (2014). Review of vision-based steel surface inspection systems. *EURASIP Journal on Image and Video Processing*, 2014(1):1–19.

Newman, T. S. and Jain, A. K. (1995). A survey of automated visual inspection. *Computer vision and image understanding*, 61(2):231–262.

Russell, B. C., Torralba, A., Murphy, K. P., and Freeman, W. T. (2008). Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173.

Tumer, I. and Bajwa, A. (1999). A survey of aircraft engine health monitoring systems. In *35th joint propulsion conference and exhibit*, page 2528.

Xie, X. (2008). A review of recent advances in surface defect detection using texture analysis techniques. *ELCVIA: electronic letters on computer vision and image analysis*, pages 1–22.

Zhao, Z., Qi, H., Qi, Y., Zhang, K., Zhai, Y., and Zhao, W. (2020). Detection method based on automatic visual shape clustering for pin-missing defect in transmission lines. *IEEE Transactions on Instrumentation and Measurement*, 69(9):6080–6091.

Zou, Z., Shi, Z., Guo, Y., and Ye, J. (2019). Object detection in 20 years: A survey. *CoRR*, abs/1905.05055.