DOI: 10.1609/aaai.12001

SPECIAL TOPIC ARTICLE

Deploying an Artificial Intelligence-based defect finder for manufacturing quality management

Kyoung Jun Lee[1] | **Jun Woo Kwon**[1] | **Soohong Min**[2] | **Jungho Yoon**[2]

[1] School of Business, Dept. of Big Data Analytics, Kyung Hee University, Hoegi 1, Dongdaemun, Seoul 02447, Korea

[2] Frontec Co, 64 Huimanggongwon, Siheung, Gyeonggi. 15098, Korea

Correspondence
Kyoung Jun Lee, Kyung Hee University.
Email: klee@khu.ac.kr

Funding information
Ministry of Education of the Republic of Korea and the National Research Foundation of Korea, Grant/Award Number: NRF-2020S1A5B8103855

Abstract

This paper describes how the Big Data Research Center of Kyung Hee University and Benple Inc. developed and deployed an artificial intelligence system to automate the quality management process for Frontec, an SME company that manufactures automobile parts. Various constraints, such as response time requirements and the limited computing resources available, needed to be considered in this project. Defect finders using large-scale images are expected to classify weld nuts within 0.2 s with an accuracy rate of over 95%. Our system uses Circular Hough Transform for preprocessing as well as an adjusted VGG (Visual Geometry Group) model. Our convolutional neural network (CNN) system shows an accuracy of over 99% and a response time of about 0.14 s. To embed the CNN model into the factory, we reimplemented the preprocessing modules using LabVIEW and had the classification model server communicate with an existing vision inspector. We share our lessons from this experience by explaining the procedure and real-world issues developing and embedding a deep learning framework in an existing manufacturing environment without implementing any hardware changes.

INTRODUCTION

The manufacturing and construction industries have always been at the forefront of innovative applications of AI technologies (Lee et al. 1995, 1998). While AI has clearly redefined how these industries approach planning and scheduling, it has also enhanced less obvious elements of the creation process. Manufacturers are required to inspect quality in order to ensure that the quality of their products meet customer demands. When this process is done manually by employees, consistent quality inspections become impossible as workers become tired, allowing for the possibility of inspection error. To address this problem, manufacturers have been using cameras and laser sensors to document the surface or state of their products in an effort to automate quality inspection using statistical methodologies, image processing, and machine learning (Neogi et al. 2014; Xie 2008).

We introduce a Convolution Neural Network (CNN)-based system designed to identify defects in weld nuts for a manufacturing company located in Korea. Prior to this system, Frontec used a vision inspector to automatically determine whether a weld nut was defective in size or whether or not it had a thread. A problem arose, however, when some of the company's customers, automobile manufacturers such as Hyundai Motors, considered a product to be defective if its surface was slightly damaged even if it had no functional defects. If a flaw on the surface was found, the customers would demand that the remaining products be investigated, requiring Frontec's staffs to manually

FIGURE 1 Manual quality inspection

FIGURE 2 The vision inspector at the factory

identify aesthetic defects, the process of which is shown in Figure 1.

Frontec thus faces a new kind of quality management problem, the problem of "aesthetic quality" or "appearance quality." Before Frontec decided to develop an AI-based system to address this problem, workers had to inspect the quality of the weld nuts manually. In addition to this manual quality inspection being unable to keep up with the product production speed (about 100,000 per day), the workers struggled with this tedious task, producing inconsistent quality management inspections as a result of eye fatigue and differing inspection criteria among employees. It therefore became necessary to automate this process.

Soohong Min, Frontec's CEO, thus asked the company that manufactured the vision inspector (Figure 2) if they could automate the new inspection task as well as the existing inspection of size and thread, but the manufacturer was unable to meet that request.

Researchers from Kyung Hee University and Benple Inc. visited Frontec's factory to analyze the details of the problem. The operating process was as follows: When the weld nuts entered the vision inspector from the production line and arrived at the rotating circular glass plate (Figure 3), the light sensor perceived the existence of the product and sent a signal to the PLC (Programmable Logic Controller). The cameras took pictures (1920 × 1600 in size) of each product and sent them to the PC. The PC sent the normal/defective signal to the PLC, and an air gun then sent each product to its appropriate group. The process had to take place within 0.2 s per weld nut with a defect classification accuracy of at least 95%. We developed a CNN-based system that satisfies these requirements and works within these constraints. Our adjusted VGG (Visual Geometry Group)-based model achieves an average defect classification accuracy of at least 99%.

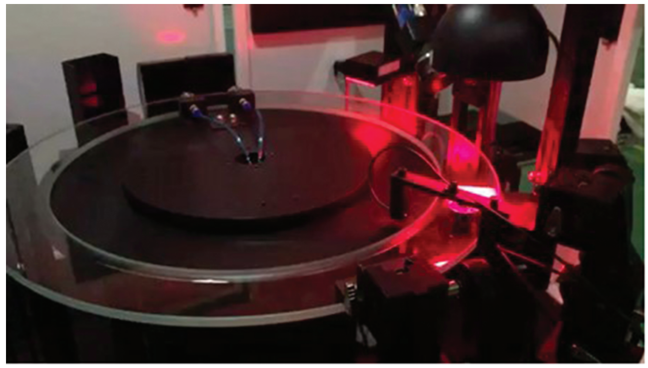

FIGURE 3 The glass plate in vision inspector

By automating quality inspection, not only is it possible to reassign staff once responsible for the tiring task of visual inspection to other tasks, but it is also possible to check products more consistently, thereby enhancing the reliability of the product. These changes, of course, have the potential to increase the efficiency and customer satisfaction of manufacturing companies.

Quality inspection has usually used image processing techniques and traditional machine learning in which extracting typical domain characteristics requires various kinds of preprocessing based on domain knowledge of the structure, statistics, filters, and model-based methodologies (Neogi et al. 2014). Such manual designs of traditional machine learning methodologies, however, may not be appropriate in the context of today's manufacturing companies, as they usually manufacture a variety of products and are required to switch between them frequently (Wang et al. 2018). The performance of CNNs, on the other hand, has become similar to or even higher than that of humans in many domains (Weimer et al. 2016) and has been applied in a few instances of product quality inspection (Masci et al. 2012). Since a CNN requires significant computations, however, applying one in environments with limited computing infrastructure or with demanding production requirements presents a challenge. As the speed of a quality inspection affects production volume, inspections need to be conducted as quickly as possible. The development of a CNN that can ensure accuracy while meeting the constraints and targets of a company is thus needed if it is to be applied in a real-world factory context.

IMAGE-BASED QUALITY INSPECTION TECHNIQUES

A research environment and a real-world environment of application are obviously different. A convolutional neural network in a real-world environment does not use original images with much noise. Therefore, the product images obtained in such an environment must be preprocessed so that they can be properly used in a high-performance convolutional neural network.

Companies have attempted to automate quality inspection using statistics, image processing, domain knowledge rule-based systems, and machine learning, to name a few methods. Statistical methodologies use a histogram analysis and an autocorrelation analysis to measure the spatial distribution of pixel values. The methodology of image processing extracts a feature through a filter or by converting an image into another form. However, a new experiment is always required to find a filter that is optimized for each domain.

Hough transform (Duda and Hart 1971) is used to detect well-defined forms, such as lines and circles. When processing an image, a filtering operation can be performed not only in a spatial domain but also in a frequency domain. Filters in a frequency domain are often used when there is a periodic characteristic or when the patterns of images cannot be detected in a spatial domain. The features obtained in these two domains can be applied to supervised machine learning to classify defects using the k-Nearest Neighbor method. Traditional Support Vector Machines (SVM) have also been used extensively for binary classification and are often used to find defects on the surface of metal products. SVM-based binary classifiers are assembled to implement a multi-class classifier and used to classify defects by incorporating feature extractors or knowledge-based methodologies implemented by other preprocessing processes such as histograms, edge extraction, and shape extraction (Agarwal et al. 2011).

Traditional quality management methodologies require significant domain knowledge or a preprocessing filter optimized for the domain in order to extract features. However, recent methodologies such as CNNs, which were originally designed for image analysis, do not require much knowledge of the domain. CNNs with max-pooling showed better performance than SVMs and multilayer perceptrons (Scholz-Reiter et al. 2012) and can correctly classify defects with or without texture (Masci et al. 2012).

Janssens et al. (2016) developed a CNN to reduce the overhead involved in feature engineering to identify defects of bearings through vibration analysis. Their CNN, which received a Discrete Fourier Transform (DFT) motor signal, classified the bearing states into four categories. Wang et al. (2016) formed a spectrogram for vibration through a Discrete Wavelet Transform (DWT) and embedded it within a CNN to classify five defect types. Dong et al. (2016) introduced the use of a CNN in distinguishing six small defects that were difficult to identify using conventional methods due to noise or resonance in the wind turbine vibration data. Ferguson et al. (2017) developed a CNN using x-ray images of castings to identify defect locations and types. Weimer et al. (2016) aimed to minimize human intervention and attempted to use a CNN for a classification in a dataset containing six good products and six artificially generated defective ones for a hypothetical micro manufacturing situation. Ye et al. (2018) introduced a CNN to distinguish a normal glass surface of a touch panel from 10 defective variations. We use a variety of preprocessing techniques to reduce overhead of feature engineering and employ a CNN to detect defects that are relatively small compared to the whole image.

APPROACH

Our project was conducted over 7 months from August of 2018 to February of 2019, as illustrated in Figure 4.

Deployment was our priority from the very beginning. Our AI team consisted of one AI model developer and one AI-factory system integrator who would lead and collaborate with the vision inspector team. We first identified the specifications of the existing vision inspector in which the AI system would be embedded and determined the allowable load levels of vision inspector with the development team. In the first month, we verified that the data provided by the factory were correctly classified and then addressed some potential issues. In the second month, we explored methods to extract the part showing the nut from the image data and searched for an optimal image size to reduce the load. From the third to fifth months, we confirmed through experiments that it would be possible to use both a simple CNN and a deep neural

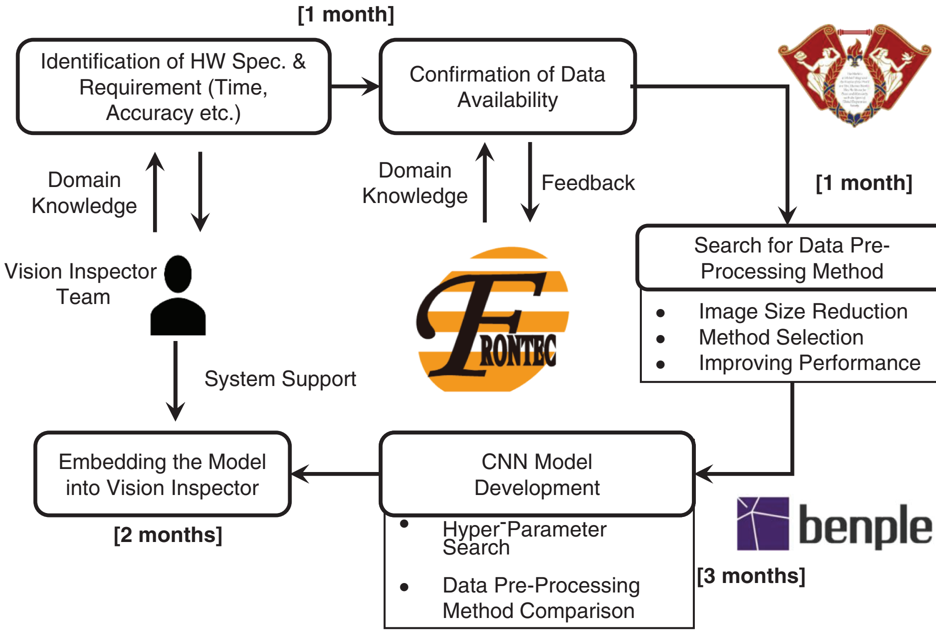

FIGURE 4 Development process

network such as VGG. After deciding to make and deploy a CNN, from the sixth to seventh month, we explored how to operate the model with the limitations of the vision inspector.

Our AI team has always emphasized the necessity of on-site R&D, that is, research and development based on field applications rather than solely lab experiments. Through an on-site visit of the factory at an early stage in the project, the AI team was able to quickly grasp what needed to be done, and the CEO's support also helped shorten the time needed to address the problem at hand. In spite of the complications usually involved in manufacturing, the factory employees also responded quickly whenever they could be of help. Creating an AI model was not the ultimate end; we had to look at how the AI model would be integrated with the factory environment.

In the first month of developing our CNN, the task of finding and developing an AI model posed somewhat of a challenge. The AI team could not immediately find a proper CNN model. Of course, a deep learning project does not always guarantee success; usually, at the start of a project, there is no certainty that a deep learning trial will be successful. AI researchers, however, have no other choice but to conduct one. When developing a deep learning system, it is especially difficult to know in advance how much data should be collected. We do not know the reason for a failure in a deep learning model because we do not yet have enough data or the neural network cannot yet solve this kind of problem. It is perhaps for these reasons that compares deep learning to alchemy (Hutson 2018) and describes it as engineering before we had the physics to understand what is going to work and why (Somers 2017).

DATA

We use two kinds of photos of weld nuts taken by the vision inspector cameras – one of the upper side surface and one of the underside. There are two types of upper side defects—"burst" and "struck"—and three types of underside defects—"internal chip," "struck," and "protrusion struck" (Figure 5). A "burst" on the upper side surface is caused by impurities in the raw material, and an "internal chip" on the underside surface results from a residue generated during the thread-cutting process. "Struck" defects are generated by the mixture of foreign matter and debris during the process. Based on ImageNet cases and other projects, the AI team asked the company to prepare about one thousand images of data of each defect category and of normal cases. As Frontec already had a vision inspector, the company could quickly generate image data of both defective and normal products by inserting the already-classified products into the vision inspector and taking pictures of them with the embedded PC. It is said that data issues are some of the most common sticking-points in any AI project and that data-wrangling of various sorts takes up about 80% of the time consumed in a typical AI project (The Economist 2020). In our case, the data preparation consumed only 5% of the project duration, which we consider to be one of the

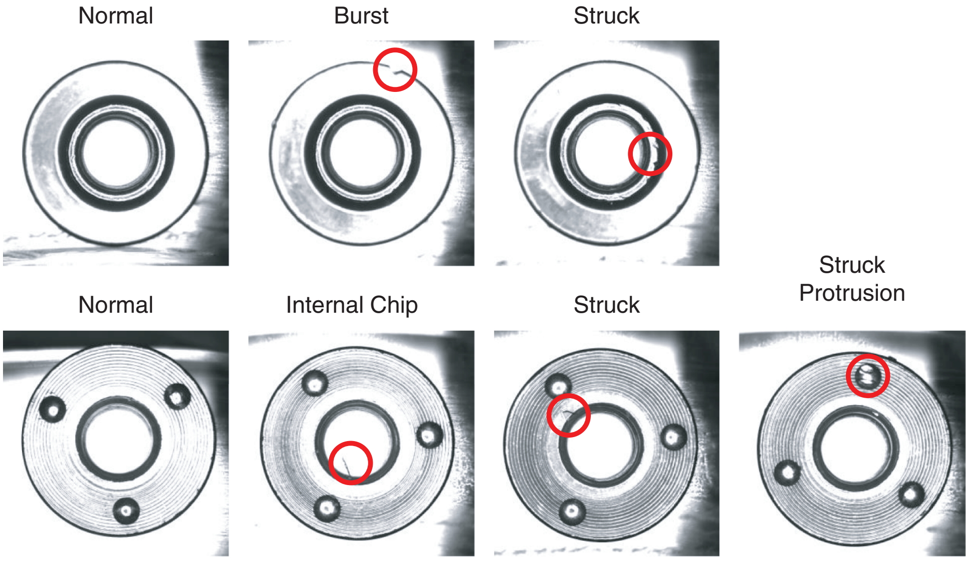

FIGURE 5 Types of defects

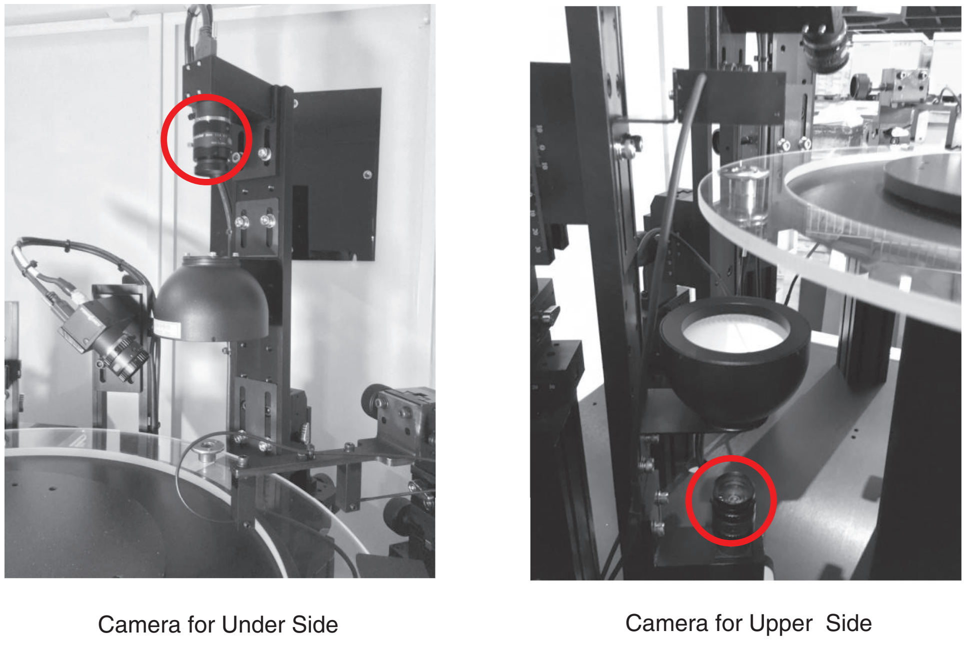

FIGURE 6 The cameras in vision inspector

reasons our project proceeded so smoothly. We received 1,198 pictures of upper side surfaces—500 of normal surfaces, 431 of "burst" defects, and 267 of "struck" defects—and 2,031 pictures of underside surfaces—500 of normal surfaces, 658 of "internal chip" defects, 410 of "struck" defects, and 463 of "struck protrusion" defects. Although the numbers of pictures received were less than what we requested, the AI team could augment the data by rotation and flipping.

The images are taken from the two cameras built into the vision inspector as shown in Figure 6. The captured image is in grayscale with a size of 1920 × 1600.

PREPROCESSING OF DATA

Although the development of CNN technology significantly lessens the preprocessing burden in image-based

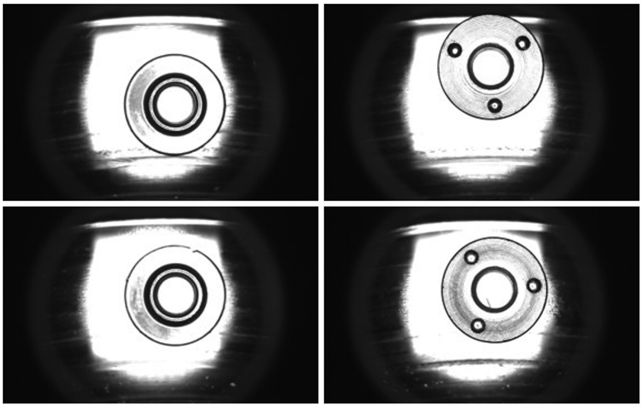

FIGURE 7 The original image data samples

quality management, it does not eliminate the work involved in preprocessing. The actual application of automated quality control at a smart factory in a real-world context is based on understanding, implementing, and integrating these preprocessing methodologies. Data preprocessing has two main purposes. The first is to reduce the size of the image (Circle Hough Transform and downsampling), and the second is to improve the performance of the model (DWT and PCA).

Circle Hough transform (CHT)

As seen in Figure 7, the background image is much larger than the weld nut area.

If the original image provided is used as the input for a learning neural network, the learning will take much longer. We needed a module to extract the part with the weld nut from the photo image supplied by the vision inspector. We used a CHT algorithm to search the circular weld nut region, which would be input for the CNN. Figure 8 shows the process of extracting only the part of the weld nut from the supplied image.

Binary threshold processing was performed on the original image based on a threshold value of 90 to make the image a simple monochrome image with no noise. A Gaussian function with a magnitude of 9 × 9 and a standard deviation of 1.5 convoluted the result to add Gaussian blur, thereby connecting the lines that can be broken. From the Gaussian Blur results, we performed a CHT to find circles with a radius of 310 to 330 pixels and determine the center coordinates and radiuses of the circles. We cut out the original image based on the coordinates and radiuses obtained through the CHT and extracted only the part with the weld nut. The final extracted image included the background and was 750 × 750 in size.

Downsampling images

The CHT reduced the image size from 1920 × 1600 to 750 × 750 and then downsampled it to 224 × 224 to account for the resources the vision inspector consumes in the PCB (Printed Circuit Boards) control. The performance of inter-area interpolation was the most effective compared to that of nearest neighbor, linear, cubic, or Lanczos interpolation, respectively (Figure 9). The smoothest lines were formed by inter-area interpolation.

Discrete wavelet transform (DWT)

Wavelet transform is widely used for inspections of surface defects in fiber, welding, and soldered PCB boards, etc. (Kim et al. 1999; Liu and MacGregor 2006; Mar et al. 2011). We tested Discrete Wavelet Transform to emphasize the weld nut image components extracted by CHT. DWT, which requires a binary orthogonal filter, is applied to decompose the input image into 16 components of five levels (Figure 10). The results of the decomposition are merged into one image to be input into the CNN.

Principal component analysis (PCA)

Features are extracted using principal component analysis (PCA), which is often used for dimensionality reduction

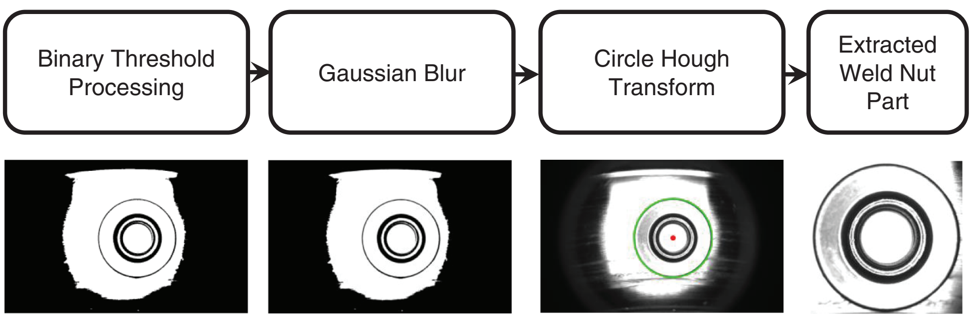

FIGURE 8 Extracting only the weld nut part in image

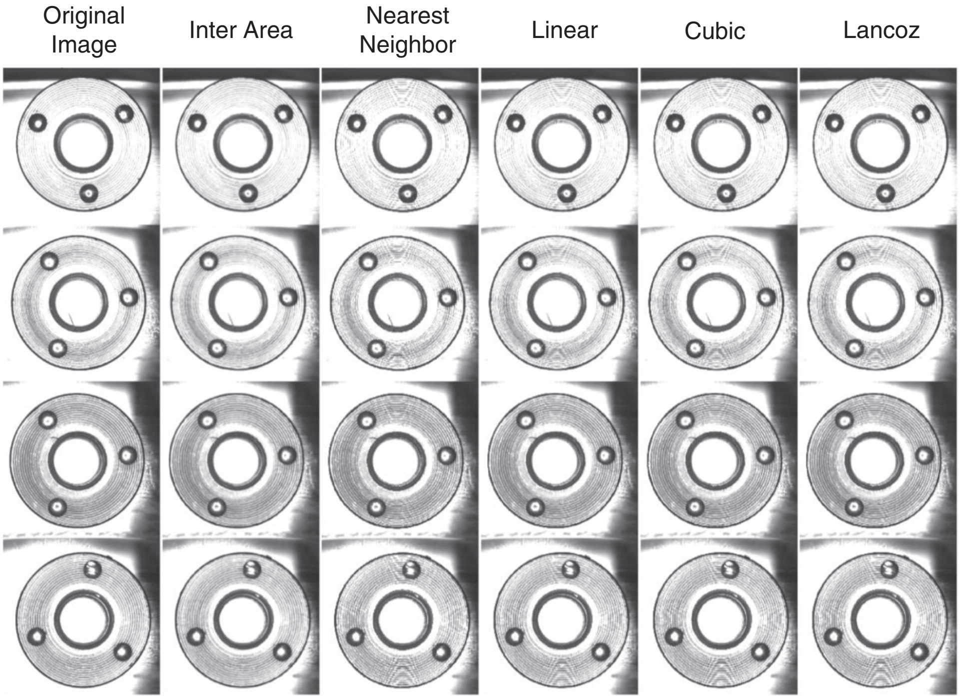

FIGURE 9 Comparison of interpolation

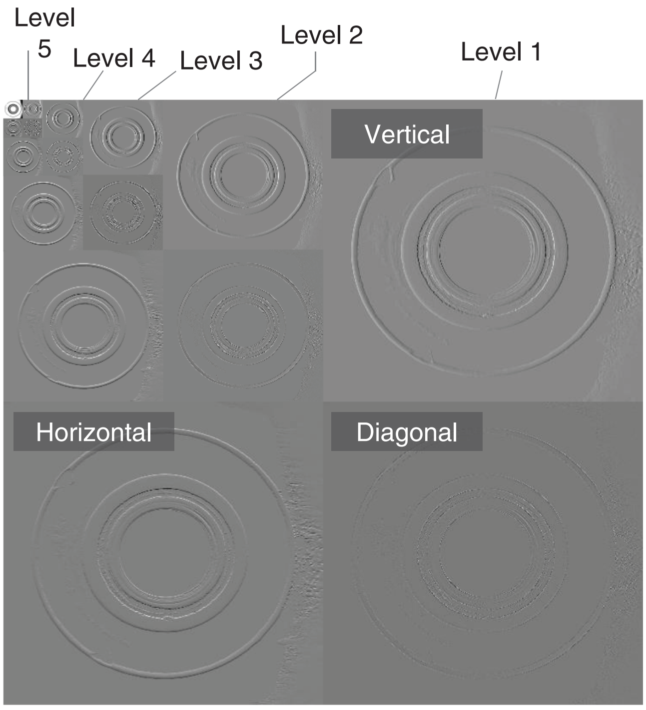

FIGURE 10 Discrete wavelet transform

and feature extraction and is also used as input for machine learning models (Ke and Sukthankar 2004; Rodrigues et al. 2010). Principal component analysis is a method of creating a new vector that can be expressed in a low dimension while maintaining the variance of the original data through linear combination, and accordingly, the variance of the new vector is set at a maximum (Abdi and Williams 2010).

Before CNNs were introduced, it was difficult to use images due to limitations in hardware performance and the requirement that input be one dimension. Therefore, Principal Component Analysis (PCA) is often used as a function to extract features from an image and reduce the dimension to generate input. Using PCA, Luiz et al. (2010) generated feature vectors from a region where a defect could be detected by Hough transform and used them as neural network input to classify defects with complex shapes. In Kumar (2003), the feature vector extracted to find defects in fibers using a multi-layer perceptron was reduced by using the principal component analysis, and the dimension of the feature space was reduced and used as input for the multi-layer perceptron. We tested PCA as a way to generate normal reference images to accentuate the difference between defective and normal product images. The standard image is generated by the process shown in Figure 11.

Each two-dimensional image was flattened into one dimension and accumulated to form one dataset. The data was normalized to a standard normal distribution and scaled. Assuming that the top 5% of the distribution of the standardization result was noise, only 95% of the components were used. An average was calculated for the results, which were inversely transformed to obtain a normal reference image. The difference between the

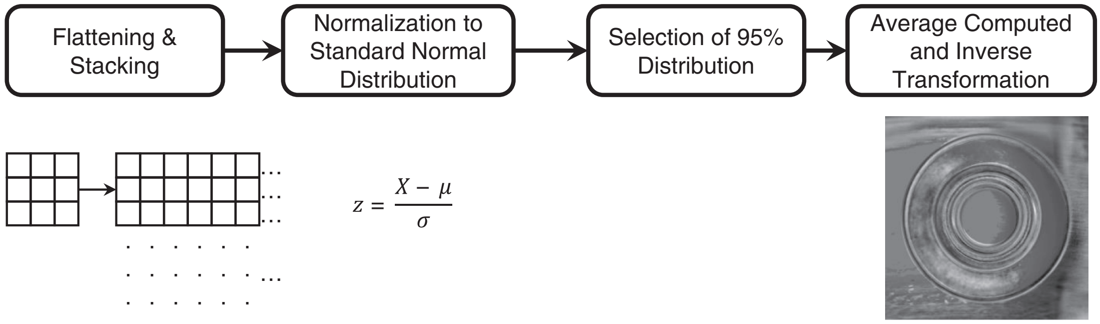

FIGURE 11 Process of generating the normal reference image

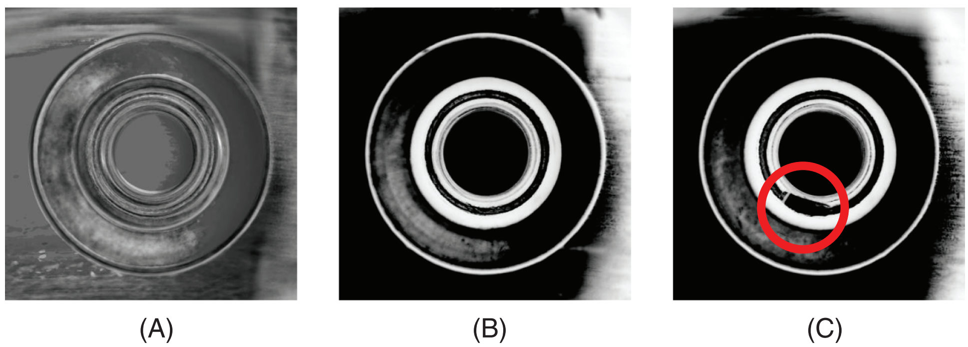

FIGURE 12 Reference image and the differenced images

reference image and the input image is shown in Figure 12. (a) is a reference image generated through principal component analysis, (b) shows the difference between the reference image and the normal image, and (c) shows the difference between the reference image and the image of a defect. The difference image was input into the CNN.

MODEL

2ConvBlock and 4ConvBlock

We defined a basic structure for the CNN from the convolution layer to the maximum pooling layer as ConvBlock. The ConvBlock was stacked to test 2ConvBlock and 4ConvBlock. One ConvBlock is composed of two convolution layers and one pooling layer (Figure 13). For 2ConvBlock, using the parameters of VGG-16 (Simonyan and Zisserman 2014), the size of the feature map was set to be twice as large as the depth of the layer. Since the input image was large, the size of the filter and the stride nearer to the input layer were set to be larger. 4ConvBlock doubled the number of layers in 2ConvBlock. The zero padding layer was used to equalize the input and output sizes of each convolution layer. The pooling size was set to 2×2 and the stride to 2. The number of hidden layers was 1, and the number of hidden nodes was fixed at 100.

VGG-16

VGG was used to superimpose sixteen structures consisting of two convolution layers and one pulling layer. A structure consisted of one 3×3 and one 2×2 convolution filter with stride one, a zero padding convolution layer, and a Max-pooling layer of zero padding. VGG has a relatively simple structure compared to that of Inception (Szegedy et al. 2017) or ResNet (He et al. 2016). Since processing

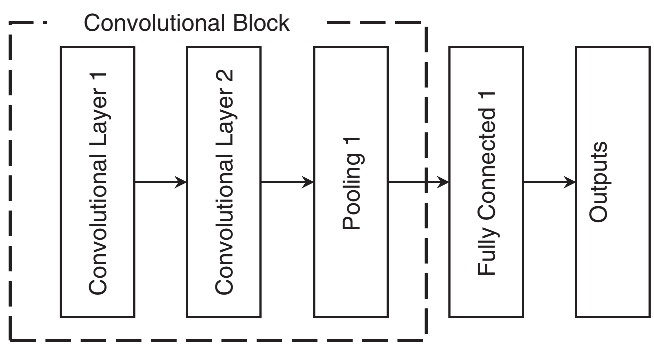

FIGURE 13 Conv block

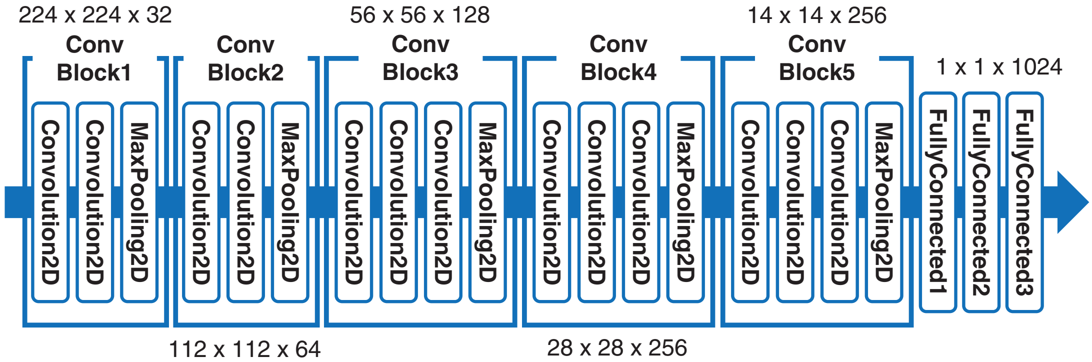

FIGURE 14 Adjusted VGG-16

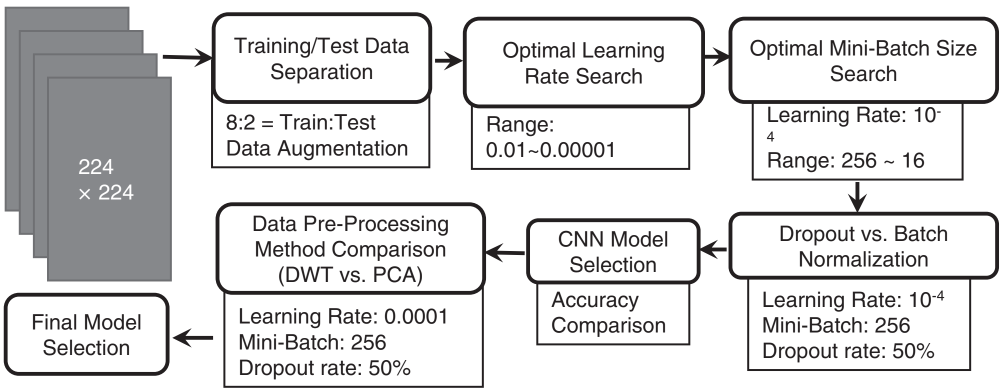

FIGURE 15 The experiment process summary

speed was important when applied to the vision inspector, the size of the original VGG-16 feature map was reduced by half, and the remaining parameters were used in the same manner (Figure 14).

EXPERIMENT FOR OPTIMAL DEPLOYMENT

The procedure of searching for the optimal structure and parameters for field deployment is shown in Figure 15.

Data set

To perform the defect classification experiment with the 2ConvBlock, 4ConvBlock, and VGG-16 models, we used the image downsampled to 224 × 224 by inter-area linear interpolation on the image extracted through CHT.

Images were divided into defect types, with 1,198 for the upper side surface and 2,031 for the underside surface, with 20% for tests and 80% for training. The datasets were separated so that the ratios of each classification type appeared identical in the training and test data sets. Three types were classified for the upper side and four for the underside. In the case of training data sets, the data was doubled by random rotation and up/down/right/left flipping.

During the first month, we were not able to find a set of parameters that satisfied the required accuracy level. In the second month, we were finally able to find the optimal parameters for the deep learning model, and the quality control system reached an accuracy rate of 99%, higher than our 95% target rate.

Experiment design

We tested how the classification accuracy of test data sets would change depending on learning rate, mini-batch size change, dropout usage, and batch normalization using the 2ConvBlock, 4ConvBlock, and adjusted VGG-16 models. To overcome an overfitting problem of deep learning, we used mini-batch, batch normalization, and dropout. Mini-batch means reducing the size of the batch, which is a unit of data to be learned. Batch normalization means scaling and shifting the distribution of inputs in mini-batch units. Dropout (Srivastava 2014) has the effect of learning generalized models by arbitrarily disconnecting connections between nodes at a certain rate.

The evaluation metrics used include accuracy, recall, precision, and F1 score. Confusion Matrix components (e.g., TP: True Positive, FN: False Negative, FP: False Positive, TN: True Negative) were measured for each experiment. Our case corresponds to situations in which defects are relatively uncommon and the cost of a false positive is significantly higher than the cost of a false negative.

Each experiment was classified into normal/defect classification and defect type classification. The experiment performed 500 epochs and measured the average accuracy and Confusion Matrix components of 400–499 epochs. In our defect type classification experiment, the accuracy of each type was measured and the average was recorded. To check the frequency of cases classifying defects as normal, the Confusion Matrix components for each defect type were combined and used for comparison.

Optimal learning rate

We experimented with the decreasing learning rates 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}, etc. The models achieved high performance when the learning rate was 10^{-4}, and the learning rates of 10^{-5} or less were too small to search the global optimum point within 500 epochs. The false-positive component of the Confusion Matrix grew smaller with smaller learning rates. For the normal/defect classification problem, the model with the learning rate 10^{-5} was the smallest, and for defect type classification, the model with 10^{-3} had the lowest false positive (FP) rate.

Optimal mini-batch size

A learning rate of 10^{-4} and mini-batch sizes of 16, 32, 64, 128, and 256 were used for the experiments. A mini-batch size of 64 or less gave the highest performance. For normal/defect classification, all test sets were correctly classified. We found that the smaller mini-batch size, the more accurate was the defect type classification. The majority of the Confusion Matrix components had the lowest false-positives at a mini-batch size of 64.

Experiment for dropout and batch normalization

Dropout (50%) or batch normalization was applied to the fully connected layers. For normal/defect classification, dropout had better performance. Batch normalization was more accurate in defect type classification. Dropout was applied because it had a lower false-positive.

Model selection

We selected VGG-16, which was more accurate for both classifications. The 2ConvBlock model had a high accuracy rate (F1) of 0.99 or higher in the case of the normal/defect classification, but the VGG-16 model was better suited for the vision inspector in cases where the FP had to be lower and the defect type had to be classified.

The VGG-16 model with a learning rate of 10^{-5} and a mini-batch size of 64 showed a low FP. As PCA and DWT were only partially accurate, we applied neither to satisfy the required operating speed. The operating speed of the model when using GPU was less than 0.078 s per inspection. The inspection system deployed the normal/defect classification model first rather than the defect classification model. Although 2ConvBlock model was simple, it showed high accuracy for normal/defect classification, showing that it can be used instead of VGG-16 if the GPU cannot be used in unavoidable accidents. VGG-16 had the lowest FP in the defect type classification. Dropout was applied because its FP value was lower than that of batch normalization. As neither PCA nor DWT performed consistently, they were not applied in order to satisfy the required operating speed.

DEPLOYMENT

Application environment and operation process

The inspection system was constructed as described in Figure 16. The weld nuts entered the rail of vision inspector, and the products classified as normal fell from the conveyor belt and into basket for packaging operation. The vision inspector used three cameras to take pictures of the upper side surface, underside surface, and thread. The inspector was able to examine only the size of the welded nuts and defects in the thread through an existing algorithm. The upper side and underside surfaces of the weld nuts were inspected through the model we developed. The inspector had a built-in computer with a CPU with the following specs: Intel i5 3.1 GHz, GPU Nvidia GTX 1080 Ti, RAM 16GB, SSD 256GB, and OS Windows 10 64bit. The vision inspector was run by a program implemented in LabVIEW (Laboratory Virtual Instrument Engineering Work-bench), a system design platform used to build automation systems in manufacturing.

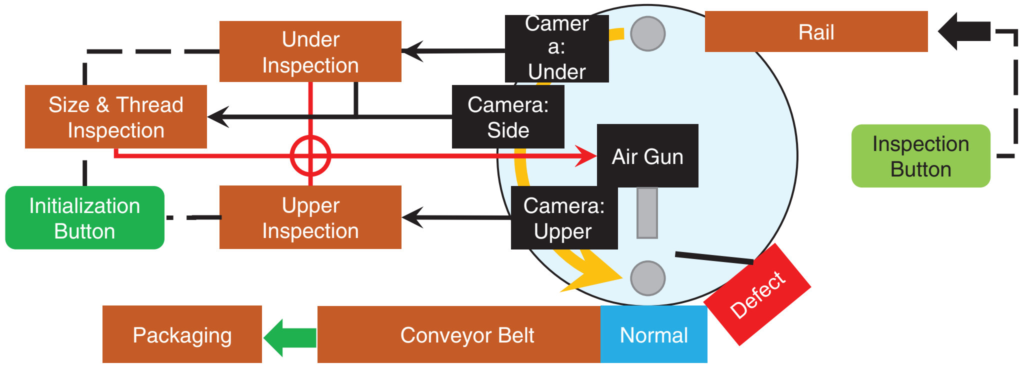

FIGURE 16 The quality inspection system

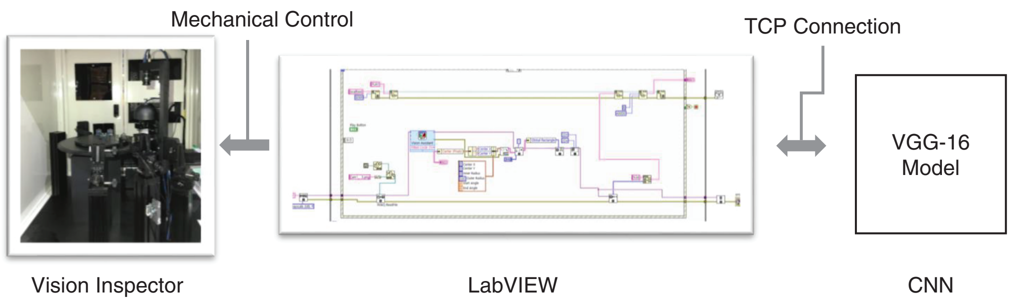

FIGURE 17 The architecture of the vision inspector

To link the developed model with the existing vision inspector, we designed the model to operate as a server and communicate with the inspector. The process of bringing the weld nut into the inspector to make a normal or defect decision is shown in Figure 16. Clicking the "Initialization" button confirmed the connection with the server, and pressing the "Inspection" button put the weld nut on a rotating transparent circular plate coming in and out of the rail. As the disc rotated, the upper side and underside surfaces of the weld nut were sequentially photographed. The captured results were then transmitted to the server. The results of the transfer were in the form of a string, which was transformed into an image that could be utilized. The image of the weld nut was then extracted through CHT. When the result of the classifying model was transmitted to the vision inspector, the inspector used the size defect decision ("size test") and the thread defect decision ("thread test") to determine whether or not the product was defective.

Interoperation with control system

The mechanism inside the vision inspector that applied the CNN model was controlled by LabVIEW (Figure 17).

Linking the CNN to LabVIEW was initially a problem. The CNN implemented through Python-based TensorFlow operated as shown in Figure 18. The hardware for learning the CNN was a CPU with Intel Core i9-7900X @ 3.30 GHz, 64GB RAM, and GPU Nvidia GTX1080ti 11GB. After performing an action related to machine control with LabVIEW, the process was reinitialized.

We then had to construct the CNN structure with checkpoint information every time an operation was performed. However, it took too much time to read because the information of the developed VGG-16 model had a size 700 MB. The initialization, which took tens of seconds, presented a serious problem because it took about 0.2 s for a weld nut to travel from the camera in the vision inspector to the front of the air gun. To address this problem, the initialization and the quality inspection processes were separated. The inspection model was designed to operate as a client-server separate from LabVIEW. The CNN ran on a local server located in the inspector's computer.

Initialization, CHT, and downsampling process

The processes consisted of executing the upper side and underside inspections respectively and sending and

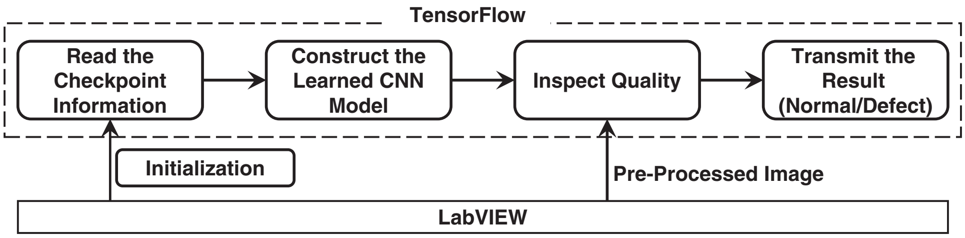

FIGURE 18 Integration of LabVIEW and TensorFlow

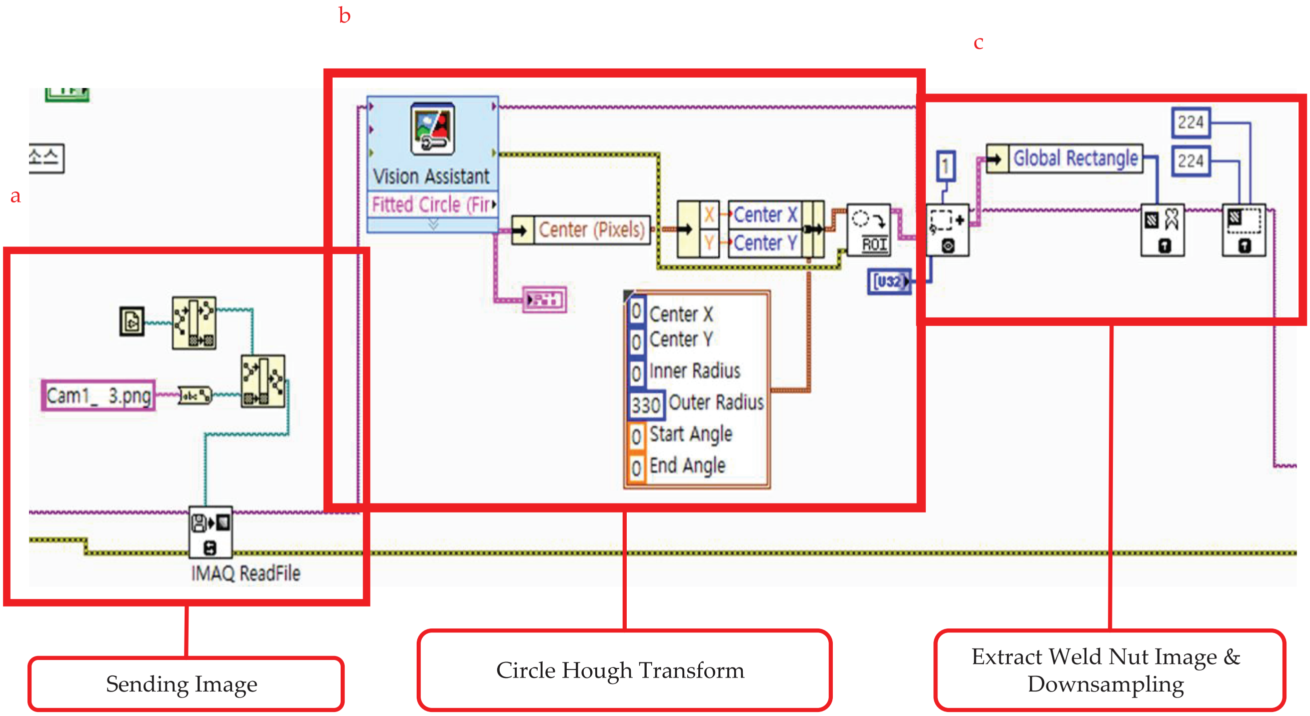

FIGURE 19 LabVIEW coding of CHT and downsampling

receiving signals using LabVIEW and TCP (Transmission Control Protocol). Pressing the "Initialization" button in LabVIEW passed the signal to each model for the initialization. By running these quality inspection models on the server, the CNN model existed in the RAM until the inspector system was shut down. If the initialization process was performed only once, the CNN would thus not need to be reinitialized. Figure 19 shows the process of preprocessing the data to operate the model. To minimize the load, the AI development team ran it through a LabVIEW tool that performs CHT and downsampling.

APPLICATION USE AND PAY-OFF

When first deployed in November 2018, the expected monetary benefit of implementing this change was US$20,000 per month, which consisted of a combination of labor costs and failure costs previously incurred in manual quality inspections. The monthly labor cost was US$16,000, which included the overtime work of four inspectors and two packing workers, and the monthly failure cost was US$4,000, which included the cost of inspections at customer sites. Besides weld nuts, however, Frontec has more than ten types of products. If AI is applied to their other products, monthly savings could reach up to US$200,000.

Compared to this potential benefit, the costs required to develop such a system do not pose much of a burden. This project also received some financial support. The Korea Industrial Complex Corporation supported Frontec's 7-month project with a US$90,000 grant under the 4th Industrial Revolution Smart Factory Construction program.

Since the deployment of AI in the manufacturing industry is still in its early stages, the AI development team had to put in some extra effort to integrate their developed AI model into the existing system. Our experience can serve as a reminder to AI experts of their potential roles in deploying AI in real-world business environments.

SUMMARY AND CONCLUSIONS

The adjusted VGG-16 model with dropout was embedded in the existing vision inspector. For its deployment, CHT and the downsampling process, once developed in OpenCV-Python, were reimplemented in LabVIEW to satisfy time and resource requirements. Our case confirms that a CNN can perform quality inspections with consistent accuracy. The company now receives feedback on product quality in real-time while reducing worker fatigue.

This deployment is a first step in automating product-level quality management by developing and applying deep learning-based AI. It is possible to extend the range of inspectable products by collecting additional data. More AI applications will also reduce development costs. The company expects that the defect rate can be lowered even further by detecting the cause of the defects early on. The company plans to expand their application of AI to enhance production scheduling and the preventive maintenance of the factory facilities. AI thus has the potential to create jobs in a variety of sectors. Frontec now plays a leading role in diffusing AI knowledge and application to related manufacturing companies, especially the forging industry.

We judge that an image-based quality management AI system for manufacturing cannot yet be extended to various factory environments without human intervention or project-type services even though some companies claim to have reached such a solution. Staff are still needed for AI modeling tasks such as choosing a proper convolution neural network and optimizing its parameters to satisfy the management requirements. Capable staff would need to be knowledgeable of statistical inference capabilities, data preprocessing techniques, and machine learning mechanisms. Some specialized solutions will reduce the customer's workload in data labeling, AI modeling, and machine training, but we still need to perform specialized projects to get satisfactory results. While this paper describes only one instance of successful AI deployment in the manufacturing industry, it is evidence of the transformative potential that AI holds for quality management systems overall.

ACKNOWLEDGEMENTS

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2020S1A5B8103855).

REFERENCES

Abdi, H., and L. J. Williams. 2010. "Principal Component Analysis." *Wiley Interdisciplinary Reviews: Computational Statistics* 2(4): 433–59.

Agarwal, K., R. Shivpuri, Y. Zhu, T. Chang, and H. Huang. 2011. "Process Knowledge Based Multi-Class Support Vector Classification (PK-MSVM) Approach for Surface Defects in Hot Rolling." *Expert Systems with Applications* 38(6): 7251–62. https://doi.org/10.1016/j.eswa.2010.12.026.

Dong, H., L. Yang, and H. Li. 2016. "Small Fault Diagnosis of Front-End Speed Controlled Wind Generator Based on Deep Learning." *WSEAS Transactions on Circuits and Systems* 15(9): 64–72.

Duda, R.O., and P. E. Hart. 1971. "Use of the Hough Transformation to Detect Lines and Curves in Pictures." Paper presented at the AAAI 2007 Spring Symposium. Technical Notes 36. Menlo Park, CA: SRI International Artificial Intelligence Center.

Ferguson, M., R. Ak, Y. Lee, and K. Law. 2017. "Automatic Localization of Casting Defects with Convolutional Neural Networks." Presented at 2017 IEEE International Conference on Big Data. Boston: Institute of Electrical and Electronics Engineers. http://doi.org/10.1109/BigData.2017.8258115

He, K., X. Zhang, S. Ren, and J. Sun. 2016. "Deep Residual Learning for Image Recognition." In Proceedings of The IEEE Conference on Computer Vision and Pattern Recognition. Las Vegas: Conference on Computer Vision and Pattern Recognition.

Hutson, M. 2018. "AI researchers allege that machine learning is alchemy." Science, May 3, https://doi.org/10.1126/science.aau0577.

Janssens, O., V. Slavkovikj, B. Vervisch, K. Stockman, M. Loccufier, S. Verstockt, and S. Van Hoecke. 2016. "Convolutional Neural Network Based Fault Detection for Rotating Machinery." *Journal of Sound and Vibration* 377: 331–45. https://doi.org/10.1016/j.jsv.2016.05.027.

Ke, Y., and R. Sukthankar. 2004. "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors." *CVPR* 2(4): 506–13.

Kim, S., M. H. Lee, and K. B. Woo. 1999. "Wavelet Analysis to Fabric Defects Detection in Weaving Processes." In Industrial Electronics, ISIE'99. Proceedings of the IEEE International Symposium on, Vol. 3, pp. 1406–9. IEEE.

Kumar, A. 2003. "Neural Network Based Detection of Local Textile Defects." *Pattern Recognition* 36(7): 1645–59.

Lee, J. K., K. J. Lee, J. S. Hong, W. J. Kim, E. Y. Kim, S. Y. Choi, H. D. Kim, O. R. Yang, and H. R. Choi. 1995. "DAS: Intelligent Scheduling Systems for Shipbuilding." *AI Magazine* 16(2): 78–94. https://doi.org/10.1609/aimag.v16i4.1173.

Lee, K. J., H. W. Kim, J. K. Lee, and T. H. Kim. 1998. "FASTrak-APT: Case and Constraint-Based Construction Project Planning System." *AI Magazine* 19(1): 13–24.

Liu, J. J., and J. F. MacGregor. 2006. "Estimation and Monitoring of Product Aesthetics: Application to Manufacturing of Engineered Stone Countertops." *Machine Vision and Applications* 16(6): 374.

Luiz, A., L. Flávio, and E. Paulo. 2010. "Automatic Detection of Surface Defects on Rolled Steel Using Computer Vision and Artificial Neural Networks." In IECON 2010–36th Annual Conference on

IEEE Industrial Electronics Society. Glendale: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IECON.2010.5675519.

Mar, N. S. S., P. K. D. V. Yarlagadda, and C. Fookes. 2011. "Design and Development of Automatic Visual Inspection System for PCB Manufacturing." *Robotics and Computer-Integrated Manufacturing* 27(5): 949–62.

Masci, J., U. Meier, D. Ciresan, J. Schmidhuber, and G. Fricout. 2012. "Steel Defect Classification with Max-Pooling Convolutional Neural Networks." In the 2012 International Joint Conference on Neural Networks. Brisbane: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IJCNN.2012.6252468.

Neogi, N., D. Mohanta, and P. Dutta. 2014. "Review of Vision-Based Steel Surface Inspection Systems." *EURASIP Journal on Image and Video Processing* 2014(1): 50. https://doi.org/10.1186/1687-5281-2014-50.

Rodrigues, M. T., F. L. Padua, R. M. Gomes, and G. E. Soares. 2010. "Automatic Fish Species Classification Based on Robust Feature Extraction Techniques and Artificial Immune Systems." In 2010 IEEE Fifth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 1518–25. IEEE.

Scholz-Reiter, B., D. Weimer, and H. Thamer. 2012. "Automated Surface Inspection of Cold-Formed Micro-Parts." *CIRP Annals-Manufacturing Technology* 61(1): 531–4. https://doi.org/10.1016/j.cirp.2012.03.131.

Simonyan, K., and A. Zisserman. 2014. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. arXiv preprint. arXiv:1409.1556. Ithaca, NY: Cornell University Library.

Somers, J. 2017. "Is AI Riding a One-Trick Pony?" *MIT Technology Review*. http://technologyreview.com/2017/09/29/67852/is-ai-riding-a-one-trick-pony/.

Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. 2014. "Dropout: A Simple Way to Prevent Neural Networks from Overfitting." *The Journal of Machine Learning Research* 15(1): 1929–58.

Szegedy, C., S. Ioffe, V. Vanhoucke, and A. A. Alemi. 2017. "Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning." In *AAAI*, Vol. 4, p. 12.

The Economist. "For AI, data are harder to come by than you think." *Technology Quarterly, The Economist*. June 11th 2020.

Wang, J., J. Zhuang, L. Duan, and W. Cheng. 2016. "A Multi-Scale Convolution Neural Network for Featureless Fault Diagnosis." In 2016 International Symposium on Flexible Automation. Cleveland: Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ISFA.2016.7790137.

Wang, J., Y. Ma, L. Zhang, R. Gao, and D. Wu. 2018. "Deep Learning for Smart Manufacturing: Methods and Applications." *Journal of Manufacturing Systems* 48: 144–56. https://doi.org/10.1016/j.jmsy.2018.01.003.

Weimer, D., B. Scholz-Reiter, and M. Shpitalni. 2016. "Design of Deep Convolutional Neural Network Architectures for Automated Feature Extraction in Industrial Inspection." *CIRP Annals* 65(1): 417–20. https://doi.org/10.1016/j.cirp.2016.04.072.

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* 7(3): 1–22.

Ye, R., C. Pan, M. Chang, and Q. Yu. 2018. "Intelligent Defect Classification System Based on Deep Learning." *Advances in Mechanical Engineering* 10(3): 1–7. https://doi.org/10.1177/1687814018766682.

AUTHOR BIOGRAPHIES

Kyoung Jun Lee is a professor of the Business School and the department of Big Data Analytics and the director of AIBM (Artificial Intelligence and Business Models) Lab and Big Data Research Center at Kyung Hee University in Seoul, Korea. He won the IAAI Awards in 1995, 1997, and 2020 from AAAI and worked as a visiting scientist/professor at CMU, MIT, and UC Berkeley. He was the 2017 president of the Korean Intelligent Information Systems Society. He founded Benple Inc., an IoT & AI-based company, and Allwin Inc., a patent-based e-commerce company. He received the 2017 Presidential Award for his contribution to Korea's e-government system. He holds BS, MS, and PhD degrees in Management Science from KAIST and completed graduate work in Public Administration at Seoul National University.

Jun Woo Kwon is a researcher for the Big Data Research Center of Kyung Hee University, Korea. He won the Innovative Applications of Artificial Intelligence Award in 2020 from the Association for Advancement of Artificial Intelligence. He developed and maintained the AI Vision Inspection System of Frontec and conducted AI projects for LG Electronics and the Korea Management Association. He holds a BS degree in Electronics and Radio Engineering and an MS degree in Social Network Science, both from Kyung Hee University.

Soohong Min is the CEO of Frontec Co., located in Siheung City, Korea. He received the Iron Tower Order of Industrial Service Merit from the President of Korea in 2016.

Jung Ho Yoon is a managing director of Frontec Co. He served at Kia Motors from 1985 to 1999.

How to cite this article: Lee, K. J., J. W. Kwon, S. Min, and J. Yoon. 2021. "Deploying an Artificial Intelligence-Based Defect Finder for Manufacturing Quality Management." *AI Magazine* 42: 5–18. https://doi.org/10.1609/aaai.12001.