

# Embedding Convolution Neural Network-Based Defect Finder for Deployed Vision Inspector in Manufacturing Company Frontec

**Kyoung Jun Lee**

Big Data Research Center, Kyung Hee Univ. Dept. of Social Network Science  
Benple Inc., Seoul, Korea  
klee@khu.ac.kr

**Jun Woo Kwon**

Kyung Hee University, Seoul, Korea  
jwkwon@khu.ac.kr

**Soohong Min Junho Yoon**

Frontec Co., Ltd.  
Siheung City, Gyeonggi, Korea  
{ft4161, yyunjungh}@hanmail.net

## Abstract

In collaboration with Frontec, which produces parts such as bolts and nuts for the automobile industry, Kyung Hee University and Benple Inc. develop and deploy AI system for automatic quality inspection of weld nuts. Various constraints to consider exist in adopting AI for the factory, such as response time and limited computing resources available. Our convolutional neural network (CNN) system using large-scale images must classify weld nuts within 0.2 seconds with accuracy over 95%. We designed Circular Hough Transform based preprocessing and an adjusted VGG (Visual Geometry Group) model. The system showed accuracy over 99% and response time of about 0.14 sec. We use TCP / IP protocol to communicate the embedded classification system with an existing vision inspector using LabVIEW. We suggest ways to develop and embed a deep learning framework in an existing manufacturing environment without a hardware change.

## Introduction

Manufacturing industry is one of pioneering areas innovatively applying AI technologies in Korea as well as the construction industry (Lee et al. 1995; Lee et al. 1998). We introduce a Convolution Neural Network(CNN)-based system developed to distinguish the defect of weld nuts for a manufacturing company of Korea. Prior to this system, the company Frontec used an existing vision inspector to determine whether a weld nut is defective in size or it has no thread with cameras. However, the customers of the company such as automobile manufacturers started to treat a product as defective if its surface is a bit damaged even though it has no functional defects. If such a defect was found, the customers demanded investigation of all remaining products and Frontec's staffs had to manually identify defects to meet the request. However, it was not possible to achieve consistent quality management because the inspec-

tion criteria were different for each employee and depended on the situation of the employee such as eye fatigue. Therefore, it became necessary to automate the inspection of the surface defect.

The speed of quality inspection should be within 0.2 seconds per weld nut and at least 95% defect classification accuracy must be achieved. The size of the picture is  $1920 \times 1600$ . We develop a CNN-based system that satisfies these requirements and can operate within constraints. Our adjusted VGG (Visual Geometry Group)-based model has an average accuracy of 98% or more on the normal/defect classification problem.

By automating quality inspection, it was possible to place the staff who had carried out the visual inspection, which is a tiring task, into another task, and it was possible to check products consistently through the equipment unlike the subjective person, thereby enhancing the reliability of the product. This increases the efficiency and customer satisfaction of manufacturing companies.

Quality inspection has used image processing techniques and traditional machine learning, where extracting typical domain characteristics requires various kinds of preprocessing using domain knowledge on the structure, statistics, filters, and model-based methodologies (Neogi et al. 2014). However, such manual designs of the traditional machine learning methodologies may not be appropriate for today's manufacturing companies, which have to switch products frequently (Wang et al. 2018).

On the other hand, the performance of the CNN has become similar to or higher than that of humans in many domains (Weimer et al. 2016) and has been being tried in product quality inspection (Masci et al. 2012). Since CNN requires a lot of computations, it seemed difficult to be applied in environments with limited computing infrastructure or when product production is so fast.

Since the speed of quality inspection affects production volume, the inspection should be carried out within the

limited time. The development of the CNN to ensure accuracy and the constraints required by the company is needed for the real factory environment.

### Image-based Quality Inspection Techniques

Companies have used cameras or sensors to digitize the surface or the state of products and attempted to automate quality inspection using statistics, image processing, domain knowledge rule-based system, and machine learning. Statistical methodologies use histogram analysis and auto-correlation analysis to measure the spatial distribution of pixel values. The image processing methodology extracts a feature through a filter or converting an image into another form. However, a new experiment is required to find a filter that is optimized for each domain.

Hough transform (Duda & Hart 1972) 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 the spatial domain but also in the frequency domain. Especially, filters in the frequency domain are used when there is a periodic characteristic, or the patterns of images cannot be detected in the spatial domain. The features obtained in the spatial domain and the frequency domain can be applied to supervised machine learning for defect classification such as k-Nearest Neighbor method. Traditional Support Vector Machines (SVM) have been also 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 for classification of defects by incorporating feature extractors or knowledge-based methodologies implemented by other preprocessing processes such as histogram, edge extraction, and shape extraction (Agarwal et al. 2011).

Especially, before CNN, it was difficult to use images because of hardware performance and the constraint that input should be one dimension. Therefore, Principal Component Analysis is often used as a function to extract features from an image and reduce the dimension to generate input. By Principal Component Analysis (PCA), Luiz et al. (2010) generated feature vectors from the region in which there may be a defect detected by Hough transform and used them as neural network input to classify defects with complex shapes.

Traditional quality management methodologies have required a lot of domain knowledge or a pre-processing filter optimized for domain in order to extract features. Many new methodologies have been developed that do not require much knowledge of the domain, such as CNN, which was originally designed for image analysis. CNN with max

pooling shows better performance than SVM and multi-layer perceptron (Scholz-Reiter et al. 2012).

Janssens et al. (2016) introduced the CNN to reduce the overhead of feature engineering for specific defects of bearings through vibration analysis. The CNN, which receives the Discrete Fourier Transform (DFT) motor signal, classifies the bearing states into four categories. Wang et al. (2016) formed a spectrogram for vibration through a Discrete Wavelet Transform (DWT) and put it into CNN to classify five defect types. Dong et al. (2016) introduce CNN to distinguish six small defects that were difficult to catch by conventional methods due to noise or resonance in the wind turbine vibration data. Ferguson et al. (2017) developed CNN using x-ray images of castings to identify defect locations and types. Ye et al. (2018) introduce a CNN to distinguish the glass surface of touch panel from normal and 10 kinds of defects. We use a variety of pre-processing techniques to reduce overhead of feature engineering and employ CNN for detecting defects relatively small compared to the whole image

### Approach

The AI development project proceeded for seven months from August of 2018 to February of 2019 as in Figure 1.

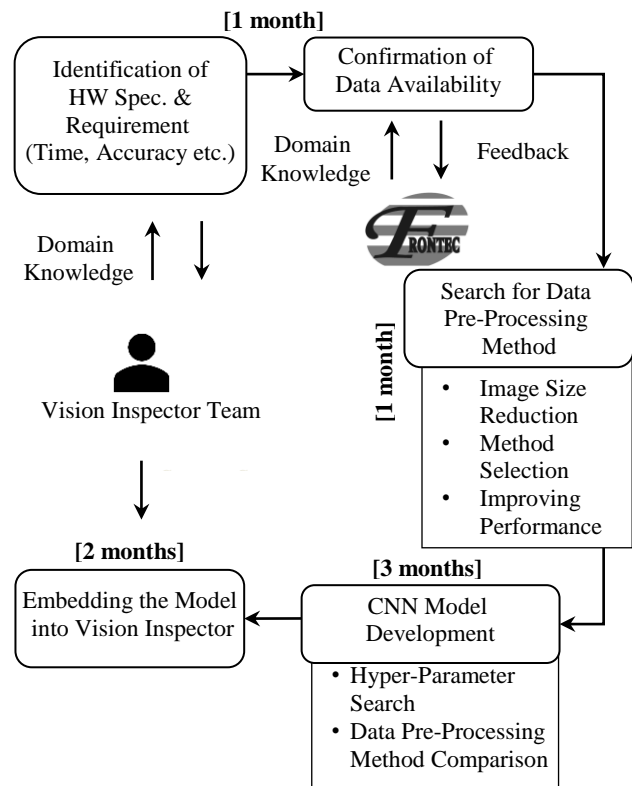


Figure 1. Development Process

From the start, the deployment issue was much emphasized. The AI team consisted of one AI model developer and one AI-factory system integrator to lead and collaborate with vision inspector team. First, we identified the existing vision inspector specification that the AI system will be embedded and agreed with development team on the allowable load levels of vision inspector. We verified that the data provided by the factory were correctly classified and some suspected matters were corrected (the 1st month). Then, we explored the method of extracting only the nut part from the data and searched for the optimal image size to reduce the load (the 2nd month). We confirmed through experiments that it is possible to introduce from the simplest CNN to the deep neural network such as VGG (from the 3rd month to the 5th month). After the decision made to deploy CNN, we explored how to operate the model in the limited environment of the vision inspector (from the 6th month to the 7th month).

### Data

We use two kinds of photos of weld nuts taken by the vision inspector cameras: upper side and 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 2). The upper side surface ‘burst’ is caused by the impurities of the raw material, and the inner chip on the underside surface is a residue generated during cutting process to make thread. Also, defects of the ‘struck’ type are generated by mixing foreign matter and debris during the process. The number of photographs provided for machine learning is 1198 on the upper side: 500 normal works, 431 ‘burst’ and 267 ‘struck’. In the case of the underside, 2031 pictures were received: 500 normal pieces, 658 ‘internal chips’, 410 ‘the struck’, and 463 ‘struck protrusion’.

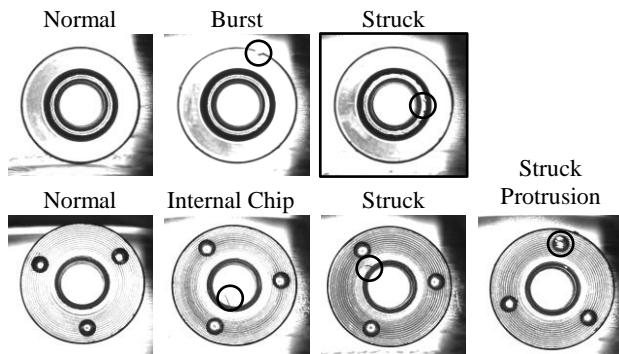


Figure 2. Types of Defects

The images are taken from the two cameras built into the vision inspector as shown in Figure 3. The captured image is grayscale and its size is  $1920 \times 1600$ .

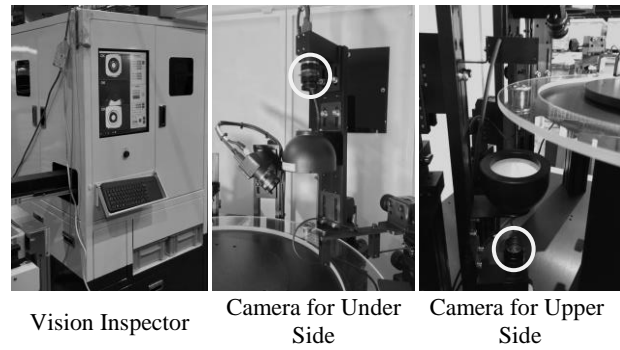


Figure 3. The Vision Inspector and the Cameras

### Pre-Processing of Data

Data preprocessing has two main purposes. The first goal is to reduce the size of the image (Circle Hough Transform and downsampling), and the second one is to improve the performance of the model (DWT and PCA).

#### Circle Hough Transform (CHT)

As seen in Figure 4, the background image is much larger than the weld nut area. If the provided image is used as the input, the initial learning takes much longer. We needed a module to extract only weld nut part from the photo image from vision inspector. CHT algorithm is used to search the circular weld nut region, which is used as the input of the CNN. Figure 4 shows the process of extracting only the weld nut part from the supplied image.

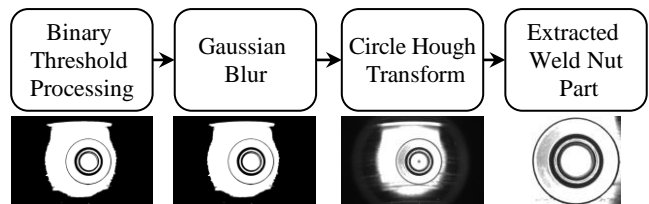


Figure 4. Extracting only the Weld Nut Part in Image

Binary threshold processing is performed on the original image based on the threshold value 90 to make the image a simple monochrome image with no noise. The Gaussian function with a magnitude of  $9 \times 9$  and a standard deviation of 1.5 convolutes the result to perform Gaussian blur, thereby connecting the lines that can be broken. From the Gaussian Blur results, we perform a CHT to find circles with radius of 310 ~ 330 pixels and get the center coordinates and a radius of the circle. We cut out the original image based on coordinates and radius obtained through CHT and extract only the weld nut part. The final size of the extracted image is  $750 \times 750$  to include the background.

## Downsampling Images

The CHT reduced the image size from 1920x1600 to 750x750, but downsamples the image to a size of 224x224 considering the resources the vision inspector consumes in the PCB (Printed Circuit Boards) control. The performance of Inter-area interpolation was the best compared with that of nearest neighbor, linear, cubic, or Lanczos interpolation respectively. The lines are formed the smoothest by the inter-area interpolation.

## Discrete Wavelet Transform (DWT)

We tested Discrete Wavelet Transform to emphasize the weld nut image components extracted by CHT. A filter for performing DWT is a binary orthogonal filter. DWT is applied to decompose the input image into 16 components of five levels. The results of the decomposition are merged as one image to be input to the CNN.

## Principal Component Analysis (PCA)

We tested PCA as a way to generate normal reference images to emphasize the difference between defective and normal product images. The standard image is generated by the process in Figure 5.

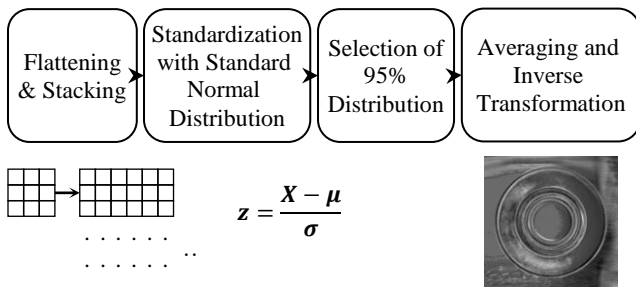


Figure 5. Process of Generating the Normal Reference Image

Smoothing each two-dimensional image into one dimension is accumulated for all the data sets to make one data. The data is normalized to a standard normal distribution and scaled. Assuming that the top 5% of the distribution of the standardization result is noise, only 95% of the components are used. An average is calculated for the result, and the result is inversely transformed to obtain a normal reference image. The difference between the reference image and the input image is shown in Figure 6. (a) is a reference image generated through principal component analysis, (b) is a difference between a reference image and a normal image, and (c) is a difference between a reference image and a defect image. The difference image is input to CNN.

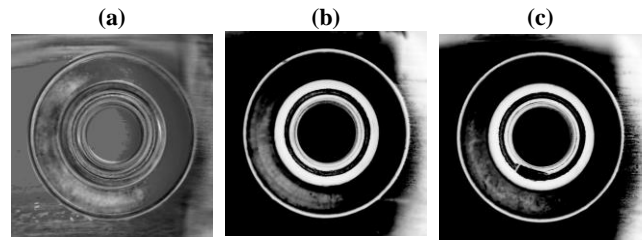


Figure 6. Reference Image and the Differenced Images

## Model

### 2ConvBlock and 4ConvBlock

We define the basic structure of CNN, from convolution layer to maximum pooling layer, as ConvBlock. ConvBlock is stacked to test 2ConvBlock and 4ConvBlock. One ConvBlock is composed of two convolution layers and one pooling layer. In 2ConvBlock, referring to the parameters of VGG-16 (Simonyan & Zisserman 2014), the size of the feature map is set to be twice as large as the depth of the layer. Since the input image is large, the size of the filter and the stride nearer to the input layer are set to be larger. 4ConvBlock doubles the layers in 2ConvBlock. The zero padding is used to equalize the input and output sizes of each convolution layer. The pooling size is set to 2X2 and stride to 2. The number of hidden layers is 1 and the number of hidden nodes is fixed to 100.

### VGG-16

VGG is superimposing sixteen structures consisting of two convolution layers and one pulling layer. A structure has one 3X3 and 2X2 convolution filters with stride one and zero padding convolution layer and a Max-pooling layer of zero padding. VGG has a relatively simple structure than Inception (Szegedy et al. 2017) or ResNet (He et al. 2016). Since the processing speed is important when applied to vision inspector, the size of the original VGG-16 feature map is reduced by half, and the remaining parameters are used in the same manner.

## Experiment for Optimal Deployment

The procedure for optimal structure and parameter search for field deployment is described in Figure 7.

### Data Set

To perform the defect classification experiment with 2ConvBlock, 4ConvBlock, and VGG-16 model, we use the image downsampled to 224x224 by inter-area linear interpolation on the image extracted through CHT.

Images are divided into defect types, with 1198 for the upper side and 2031 for underside, with 20% for test and the 80% for training. The datasets were separated so that the ratios of each classification type appeared identical in the training and test data sets. The number of types to be classified is three on the upper side and four on the underside. In the case of training data sets, data is doubled by random rotation and up/down/right/left flipping.

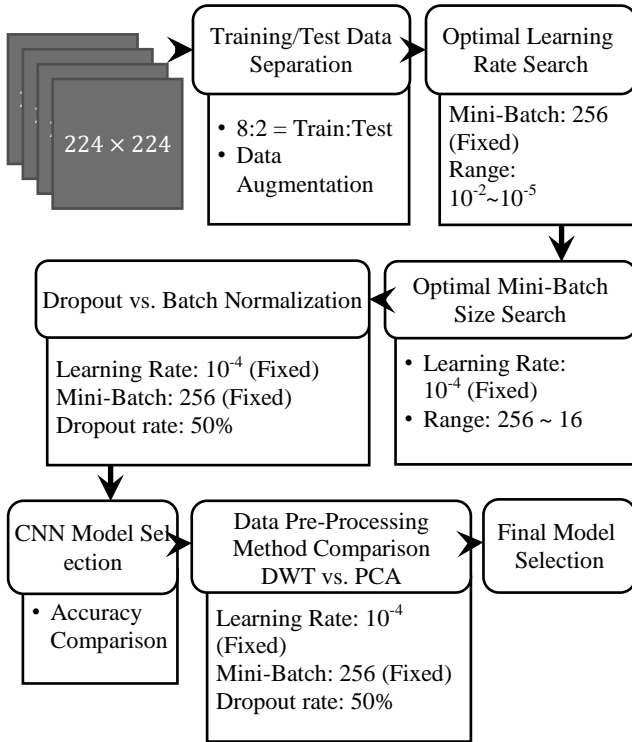


Figure 7. The Experiment Process Summary

## Experiment Design

We tested how the classification accuracy of test data sets changes depending on learning rate, mini-batch size change, dropout usage, and batch normalization in using 2ConvBlock, 4ConvBlock, and adjusted VGG-16 models. To overcome this 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 the situation where defects

are relatively uncommon and the cost of false positive is significantly higher than the cost of 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 the 400 ~ 499 epochs. In defect type classification experiment, the accuracy of each type was measured and the average was recorded. For checking the frequency of cases classifying defects as normal, the Confusion Matrix components for each defect type were combined and used for comparison. The detailed results can be found in Kwon (2019).

## Optimal Learning Rate

We experimented with the decreasing learning rates  $10^{-2}$ ,  $10^{-3}$ ,  $10^{-4}$ ,  $10^{-5}$  and so on. Models achieved high performance when the learning rate was  $10^{-4}$ . The learning rates of  $10^{-5}$  or less are too small to search the global optimum point within 500 epochs. The false-positive component of the Confusion Matrix is the smaller as the learning rate is the smaller. In the case of normal/defect classification problem, the model with the learning rate  $10^{-5}$  is the smallest, and in the defect type classification, one with  $10^{-3}$  has the lowest false positive (FP) rate.

## Optimal Mini-Batch Size

Learning rate  $10^{-4}$  and the mini-batch sizes 16, 32, 64, 128, and 256 are used for the experiments. The mini-batch size 64 or less gave the highest performance. Especially for normal/defect classification, all test sets were correctly classified. The smaller mini-batch size, the more accuracy is achieved in defect type classification. The majority of the Confusion Matrix components have the lowest false-positives at mini-batch size 64.

## Experiment for Dropout and Batch Normalization

Dropout (50%) or batch normalization is placed after fully connected layer. For normal/defect classification, dropout had better performance. Batch normalization had higher accuracy in defect type classification. Since dropout has a lower false-positive, dropout is finally applied.

## Model Selection

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



The VGG-16 model with learning rate of  $10^{-5}$  and mini-batch size 64 showed low FP. PCA and DWT were not always performing well in all cases, but partially accurate, therefore we apply neither to satisfy the operating speed. The operating speed of the model when using GPU is less than 0.078 seconds per inspection. The inspection system deploys the normal/defect classification model first rather than the defect classification model. 2ConvBlock is simple but it shows high accuracy in the problem of normal/defect classification, so it can be used instead of the VGG-16 just in case if the GPU cannot be used in unavoidable accidents.

## Deployment

### Application Environment and Operation Process

The inspection system is constructed as in Figure 8. Weld nuts enter the rail of vision inspector, and the products classified as normal fall from the conveyor belt to basket for packaging operation. The vision inspector takes photos through three cameras for upper side, underside, and thread images. The existing inspector can only examine the size of the welded nuts and the defect in the thread through the existing algorithm. The upper side and underside of the weld nuts are inspected through the model we developed. The inspector has a built-in computer with CPU intel i5 3.1GHz, GPU Nvidia GTX 1080 Ti, RAM 16GB, SSD 256GB, and OS Windows 10 64bit. The vision inspector operates under a program implemented in LabVIEW (Laboratory Virtual Instrument Engineering Workbench), a system design platform used to build automation systems in manufacturing.

For applying the developed model to the existing vision inspector, we designed the model to operate as a server and to 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 8. First, clicking the "Initialization" button confirms the connection with the server. Pressing the "inspection" button puts the weld nut on a rotating transparent circular plate coming in and out of the rail. As the disc rotates, the upper side and underside of the weld nut are sequentially photographed by the camera. The captured result is transmitted to the server. The result of the transfer is in the form of a string, which is transformed into an image form that can be operated on, and then only the weld nut part image is extracted through CHT. If the result of the classifying model is transmitted to the vision inspector, it is finally decided whether or not the product is defect integrating the size defect decision ('size test') and the thread defect decision ('thread test').

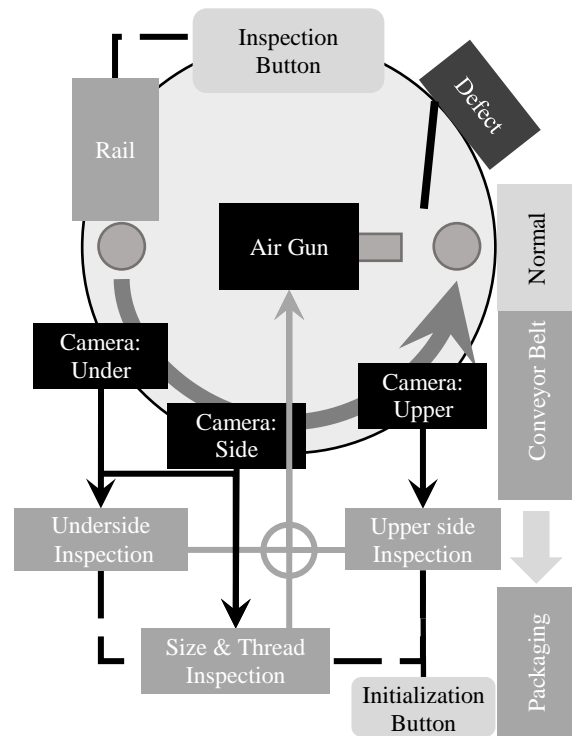


Figure 8. The Quality Inspection System

### Interoperation with Control System

The mechanism inside the vision inspector to apply the CNN model is controlled by LabVIEW (Figure 9).

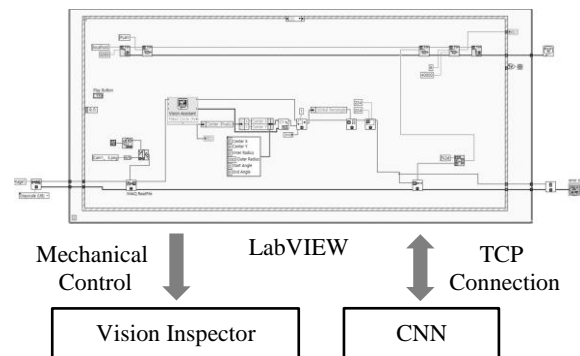


Figure 9. The Architecture of the Vision Inspector

There was a problem in initializing CNN to link the LabVIEW. First, CNN implemented through Python-based TensorFlow operates as Figure 10. The hardware for learning CNN has the CPU Intel Core i9-7900X @ 3.30GHz, 64GB RAM, and GPU Nvidia GTX1080ti 11GB. After performing an action related to machine control with LabVIEW, the process is reinitialized.

Then we should construct the CNN structure through the checkpoint information every time each operation is performed. However, it takes tens of seconds to read because the checkpoint information of the developed VGG-16 has a size 700 MB. The initialization, which takes tens of seconds, is a fatal problem because it takes about 0.2 seconds for a weld nut to reach the front of the air gun from the camera in the vision inspector. To solve this, the initialization and the quality inspection process were separated. The inspection model was designed to operate as a client-server separately from LabVIEW. The CNN runs as a local server on a computer embedded in the inspector.

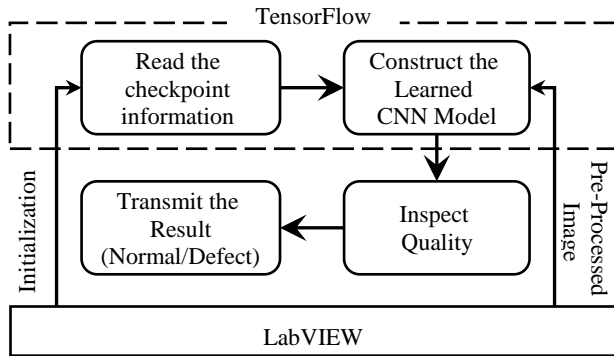


Figure 10. Integration of LabVIEW and TensorFlow

**Initialization, CHT and Downsampling Process**

The processes consist of executing the upper side and underside inspection respectively, and sending and receiving signals using LabVIEW and TCP (Transmission Control Protocol). Pressing the Initialization button in LabVIEW passes the signal to each model for the initialization. By running these quality inspection models on the server, the CNN model exists in the RAM until the inspector system is shut down. So, if you perform the initialization process only once, you do not need to reinitialize the CNN even after the machine control through LabVIEW, and you can eliminate the time lost from the initialization. Figure 11 shows the process of preprocessing the data to operate the model. To minimize the load, AI development team implemented it through a tool that performs CHT and downsampling built into LabVIEW.

**Quality Inspection Process**

The designed LabVIEW process to perform the quality inspection is in Figure 12. When you press the button “Inspection”, LabVIEW sends a signal and the preprocessed images to the server to perform the inspection. The server activates the CNN to give the judgment. The server passes the computed results to LabVIEW, which synthesizes the results, size test result, and thread test result, and drops them into the air gun if the final judgment is normal.

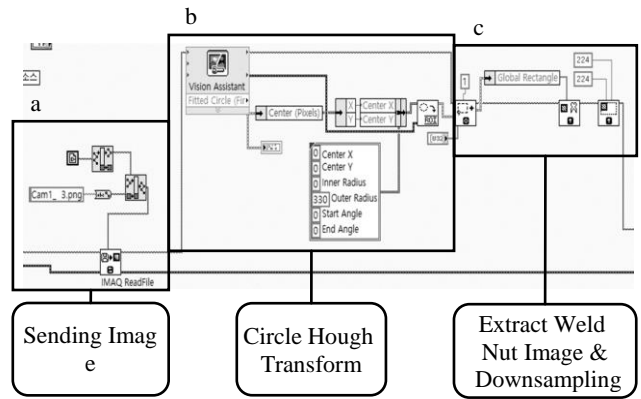


Figure 11. LabVIEW Coding of CHT and Downsampling

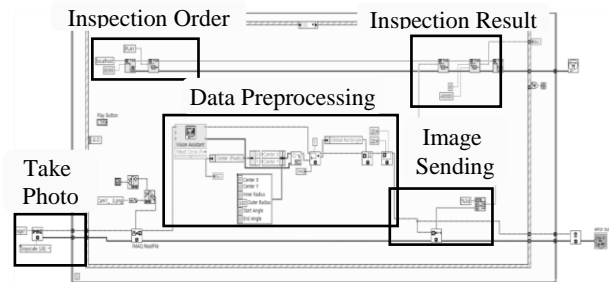


Figure 12. LabVIEW Implementation

**Application Use and Pay-off**

Since the deployment on November 2018, the expected monetary benefit is US\$20,000 per month, a combination of labor costs and failure costs previously incurred to manually perform a quality inspection. The monthly labor cost US\$16,000 includes the overtime work of 4 inspectors and 2 packing workers, and the monthly failure cost US\$4,000 includes the cost for inspection at customer site. Besides weld nut, Frontec has more than ten types of products. If we introduce AI for the rest products, we expect a profit of US\$200,000 per month.

Compared with the large potential benefit of the AI system, the development cost is not so much. Korea Industrial Complex Corporation, supported the Frontec’s seven-month project with US\$90,000 under the grant of the 4th Industrial Revolution Smart Factory Construction Program. The financial support was shared among the AI development team (Benple Inc. and Kyung Hee University), vision inspector team, and the Frontec’s fieldwork team. Since deploying AI in the manufacturing industry is still in the early stage, the AI development team had to contributed much more to even in integrating the developed AI model into the existing system than the vision inspector team. Our experience gives a lesson to AI experts in recognizing their roles in successfully deploying AI in the real business environments.

## Summary and Conclusions

The adjusted VGG-16 with dropout is embedded in the existing vision inspector. For the deployment, CHT and the downsampling process, once developed in OpenCV-Python, are implemented in LabVIEW to satisfy the time and resource requirement. Our case confirms that CNN can perform the quality inspection with consistency and accuracy. Frontec now can feedback quality information in real-time and reduce worker fatigue.

This deployment is a first step to automatic quality management of all products by developing and applying deep learning-based AI to the modern manufacturing industry. It is possible to extend the range of inspectable products by securing additional data. The experience of success will save development costs for further development. By automatically classifying the defect types real time with high accuracy, the company expects the defect rate can be even more lowered by early detection of the cause of the defect. Through this AI success experience, the company now plans to expand the application of AI from the current quality inspection to optimal production scheduling and the preventive maintenance of the factory facilities. Frontec now plays a hub role of diffusing the AI knowhow to related manufacturing companies, especially to forging industry.

## Acknowledgements

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2017S1A5B8059804).

## References

- Agarwal, K.; Shivpuri, R.; Zhu, Y.; Chang, T.; and Huang, H. 2011. Process Knowledge Based Multi-Class Support Vector Classification (PK-MSVM) Approach for Surface Defects in Hot Rolling. *Expert Systems with Applications* 38(6): 7251-7262. doi.org/10.1016/j.eswa.2010.12.026.
- Dong, H.; Yang, L.; and Li, H. 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 Hart, P. E. 1971. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Papers from the 2007 AAAI Spring Symposium. Technical Notes 36*. Menlo Park, CA: Sri International Artificial Intelligence Center.
- Ferguson, M.; Ak, R.; Lee, Y.; and Law, K. 2017. Automatic Localization of Casting Defects with Convolutional Neural Networks. In 2017 IEEE International Conference on Big Data. Boston: Institute of Electrical and Electronics Engineers. dx.doi.org/10.1109/BigData.2017.8258115.
- He, K.; Zhang, X.; Ren, S.; and Sun, J. 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.
- Janssens, O.; Slavkovikj, V.; Vervisch, B.; Stockman, K.; Loccufer, M.; Verstockt, S.; and Van Hoecke, S. 2016. Convolutional Neural Network Based Fault Detection for Rotating Machinery. *Journal of Sound and Vibration* 377: 331-345. doi.org/10.1016/j.jsv.2016.05.027.
- Kwon, J. 2019. Development and Deployment of Deep Learning Based Large-Scale Image Classification System for Automated Quality Inspection in Manufacturing. Master Thesis, Department of Social Network Science, Kyung Hee University.
- Lee, J. K.; Lee, K. J.; Hong, J. S.; Kim, W. J.; Kim, E. Y.; Choi, S. Y.; Kim, H. D.; Yang, O. R.; and Choi, H. R. 1995. DAS: Intelligent Scheduling Systems for Shipbuilding. *AI Magazine* 16(2): 78-94. doi.org/10.1609/aimag.v16i4.1173.
- Lee, K. J.; Kim, H. W.; Lee, J. K.; and Kim, T. H. 1998. FAS-Trak-APT: Case and Constraint-Based Construction Project Planning System. *AI Magazine* 19(1): 13-24.
- Luiz, A.; Flávio, L.; and Paulo, E. 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. doi.org/10.1109/IECON.2010.5675519.
- Masci, J.; Meier, U.; Ciresan, D.; Schmidhuber, J.; and Fricout, G. 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. doi.org/10.1109/IJCNN.2012.6252468.
- Neogi, N.; Mohanta, D.; and Dutta, P. 2014. Review of Vision-Based Steel Surface Inspection Systems. *EURASIP Journal on Image and Video Processing*. 2014(1): 50. doi.org/10.1186/1687-5281-2014-50.
- Scholz-Reiter, B.; Weimer, D.; and Thamer, H. 2012. Automated Surface Inspection of Cold-Formed Micro-Parts. *CIRP Annals-Manufacturing Technology* 61(1): 531-534. doi.org/10.1016/j.cirp.2012.03.131.
- Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv preprint. arXiv:1409.1556. Ithaca, NY: Cornell University Library.
- Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *The Journal of Machine Learning Research* 15(1): 1929-1958.
- Wang, J.; Ma, Y.; Zhang, L.; Gao, R.; and Wu, D. 2018. Deep Learning for Smart Manufacturing: Methods and Applications. *Journal of Manufacturing Systems* 48: 144-156. doi.org/10.1016/j.jmsy.2018.01.003.
- Wang, J.; Zhuang, J.; Duan, L.; and Cheng, W. 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. doi.org/10.1109/ISFA.2016.7790137.
- Weimer, D.; Scholz-Reiter, B.; and Shpitalni, M. 2016. Design of Deep Convolutional Neural Network Architectures for Automated Feature Extraction in Industrial Inspection. *CIRP Annals* 65(1): 417-420. doi.org/10.1016/j.cirp.2016.04.072.
- Ye, R.; Pan, C.; Chang, M.; and Yu, Q. 2018. Intelligent Defect Classification System Based on Deep Learning. *Advances in Mechanical Engineering* 10(3): 1-7. doi.org/10.1177/1687814018766682.