High-Precision Quality Inspection for Screws Using Artificial Intelligence Technology

Qian Huang
Assistant Professor, School of Architecture, Southern Illinois University Carbondale, IL, USA
email: qhuang@siu.edu

Abstract

Bolts, nuts, and screws are widely used in the construction industry, so their quality is critical to the success of a construction project. Inferior blots, nuts, and screws are prone to slippage, cracks or breaks in roofs or walls, and ultimately post life-threatening risks for construction workers and building users. To avoid these dangerous consequences, it is required to carefully inspect and control the quality of each bolt, nut, and screw. Yet, unfortunately, manual inspection conducted by humans is a time-consuming and labor-intensive task, hence, it results in poor detection accuracy and low detection throughput. To address this challenge, in this work, we investigate the use of convolutional neural networks (CNNs) based artificial intelligence to realize high-precision and high-throughput automatic quality inspection. First, we take pictures for 8,200 screw surfaces from a screw manufacturer. After a careful quality examination, each screw picture is marked as "defect-free" or "defective" in a dataset. Then, we explore and propose a low-complexity and low-cost CNN-based neural network architecture. These labeled screw images in the dataset are used to train parameters of the proposed neural network architecture and to verify the resultant detection accuracy. Our experimental results show that the quality detection accuracy reaches 95.13% at steady state, and the detection throughput is 2 or 3 orders of magnitude higher than that of humans.

Keywords: Screw Heads, Defective Inspection, Low Complexity, Convolutional Neural Networks

1. Introduction

Bolts, nuts, and screws are essential elements of construction materials that hold multiple mechanical parts together. As a result, their head defects (*e.g.*, cracks, misalignments, damages) affect the operation and safety of construction projects. Inferior blots, nuts, and screws are prone to slippage, cracks or breaks in roofs or walls, and ultimately post life-threatening risks for construction workers and building users. Hence, it is mandatory to carefully inspect all manufactured bolts, nuts and screws before using them in construction projects (Zawada et al, 2018). Moreover, since a large number of screws can be produced by manufacturing equipment in a short time, we envision that a fast and highly accurate quality inspection method is required. In this way, once defective bolts, nuts, or screws are identified, manufacturers can remove them immediately before packing and shipping them to customers.

With the rapid advancement of computer vision technologies, particularly the emerging artificial intelligence (AI) algorithms (LeCun et al, 2015), machine vision has great potential in inspection, sorting, and quality control of construction materials at the manufacturing stages. Traditionally, manufacturing equipment needs to be frequently stopped and idle for a certain period of time for human-conducted onsite quality inspection. This intermittent manufacturing and inspection manner is inefficient for achieving high-throughput automated production (Martinez et al, 2019). In addition, onsite quality inspection personnel typically need years of working experiences. Furthermore, a human inspector may take tens of seconds or even minutes to complete quality assessment of a screw. In contrast, because AI algorithms have the unique ability to automatically extract and learn intrinsic features from input raw data, the combination of computer vision technologies and AI algorithms is expected to result in low cost, high efficiency, and high throughput quality monitoring and real-time inspection (Ahmed et al, 2012; Fathi et al, 2015). We believe that a screw head image captured by cameras contains a lot of embedded information about its quality degree, which can be automatically and implicitly derived using artificial intelligence algorithms. Therefore, this study plans to investigate and develop effective AI algorithms for rapid and accurate screw quality inspection. The targeted AI algorithm should be user-friendly, reliable, robust, and inexpensive to implement.

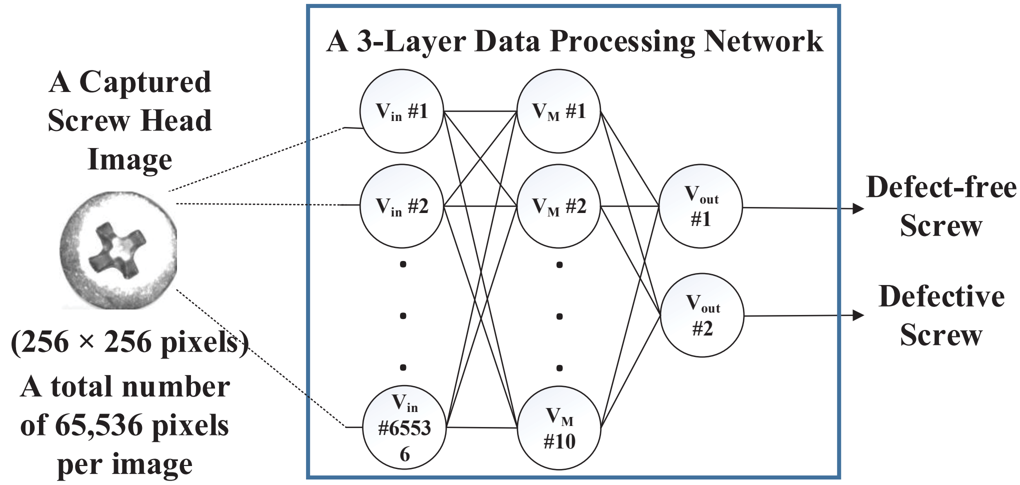

Figure 1: An Example of Using AI Algorithms for Screw Quality Inspection

As illustrated in Figure 1, AI algorithms put a bunch of original data (*i.e*., each pixel value from a captured screw head image) into a complex data processing network (*e.g*., a 3-layer fully-connected network), and then check if the output result of this network meets requirements - if yes, the network will be used as the target model; if not, the parameters in the network will be repeatedly updated until the output result meets the requirements. Such data processing networks typically consist of several data processing layers, and the network processing capability increases as the number of layers increase.

In this study, due to the high degree of variability in screw head defects (*e.g*., cracks, misalignments, damages, broken edges at different locations) as shown in Figure 2, there are no clues about how to effectively extract the intrinsic features of defective screw heads. Hence, it is difficult to manually extrapolate useful features or patterns for quality judgment directly from input screw head images. Note that the image capture position of the screw always changes slightly as the screw moves under the camera. Therefore, as shown in Figure 2, the orientation of screw heads and the appearance of the cross recess are not the same in different images. The detection algorithms to be developed in this study should deal with this orientation variations properly. On the other hand, through literature review, convolutional neural networks (CNN) have demonstrated excellent performance in speech or image recognition/classification, and natural language processing. Although the problems solved in these areas are not the same, these application methods can be summarized as follows: CNNs can automatically learn features from large-scale input raw data and generalize the results to unknown data of the same type. The strong learning ability, fast training speed, and high accuracy of CNNs overcome inherent shortcomings of the traditional neural networks. Therefore, to address this severe challenge of rapid and high-precision inspection of screw quality, we propose to develop appropriate CNN-based AI algorithms. Our proposed CNN-based AI algorithm will create non-linear mappings from a screw head image to a quality decision.

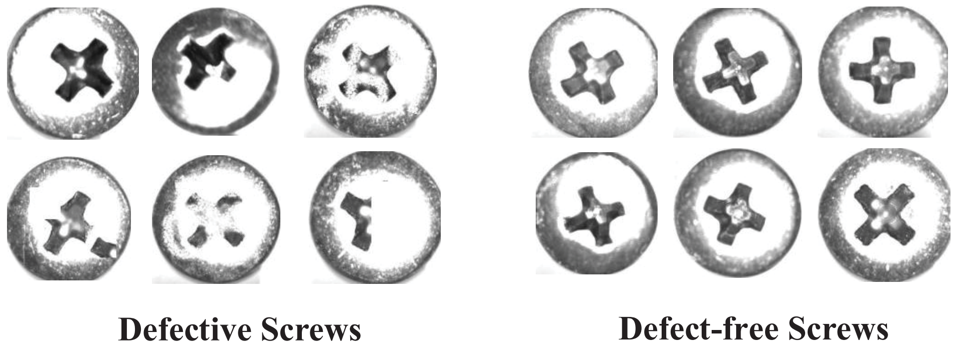

Figure 2: Image Examples of Defective and Defect-free Head Screws

The rest of this paper is organized as follows. Section 2 reviews the literature study related to the use of AI algorithms in object identification or quality inspection. Section 3 describes the proposed low-complexity CNN-based network architecture. Section 4 introduces the experimental test of the proposed CNN architecture using established screw image datasets, and compares this work with existing state-of-the-art designs in the literature. Section 5 concludes this work.

2. Literature Review

In this section, recent research progress and achievements in the area of CNN-based object detection are reviewed and discussed. A variety of CNN architectures have been created for specific applications. For example, the researchers in (Cha et al, 2017) have presented a deep CNN architecture for crack detection in civil infrastructures. Although the reported detection accuracy is very good (about 98%), yet, this CNN architecture is complex, including four convolution layers and two pooling layers. Other CNN architectures for concrete cracks detection is more complex, such as the adoption of 13 convolutional layers in (Silva et al, 2018). The researchers in (Kim et al, 2018) have presented a region-based fully convolutional network for construction object detection. Using the proposed network consisting of 3 convolutional layers, the experimental results have shown a detection accuracy of 96%. The researchers in (Chen et al, 2018) developed deep CNNs to identify defective states of catenary support devices in the electrified railway industry. Even though the experimental measurements show a high detection accuracy and strong robustness in complex outdoor environments, the required CNN architecture is composed of 6 convolutional layers. Later, a deep CNN architecture was developed to detect the defects on metal screw surfaces in (Song et al, 2018). Based on the conventional LeNet-5 (LeCun et al, 1998), this complex CNN architecture utilizes 3 convolutional layers, 3 pooling layers, and 3 fully-connected layers. Furthermore, the input screw image in (Song et al, 2018) is limited to 32×32 pixels. If high-resolution screw images are used as inputs, such as 256×256 pixels or higher, the corresponding CNN architecture will be more complicated.

It can be clearly seen from the above discussion that despite the superior object detection performance, these existing CNN architectures (Cha et al, 2017; Silva et al, 2018; Chen et al, 2018; Song et al, 2018; Kim et al, 2018) rely on complex neural networks, which require a significant amount of computing resources and storage memories to rapidly detect defects. To overcome this resource challenge and still obtain accurate quality assessments, we will explore a low-complexity, resource-efficient CNN architecture, which supports end-to-end computation – from the input screw head images down to an output decision for "defective" or "defect-free". In this sense, our proposed CNN architecture has the potential to accommodate resources of cost-effective hardware platforms, such as a low-cost embedded system consisting of only a microprocessor and limited memory space.

3. Proposed Low-Complexity CNN Architecture

A CNN architecture is composed of various functional layers: convolutional layers, pooling layers, ReLU layers, and fully-connected layers. Convolution is a mathematical operation to simplify data representations by filtering out unwanted noise. As a basic layer of CNNs, a convolutional layer often consists of more than one convolution kernel. Each kernel is convoluted with the input data to form a feature map. Several convolution kernels have been designed to perform image edge detection, sharpening, blurring, etc. Therefore, features (*e.g.*, edges and curves) in images can be extracted through different convolution kernels. The size and number of convolution kernels are pre-defined parameters in CNN architectures. When training a convolutional neural network, the trainable parameters in convolution kernels are automatically adjusted to get better results. This process is called feature learning or feature extraction. In other words, a convolution layer extracts the hidden features from input data and outputs a feature map, which is often with a large dimension. As each convolution kernel can grab the presence of a specific feature, people choose to use multiple kernels to capture different features.

It is well known that a feature map usually has spatial correlation – a pixel is similar to the pixels around it in a large probability. If adjacent pixels are merged, the feature map size will be reduced. Therefore, a pooling layer performs feature selection on the original feature map by taking the maximum or average value to remove redundant features, and reconstructs a new feature map with a small dimension. In average pooling, the average of all values in the pooled area is used as the pooling result. In maximum pooling, the maximum of all values in the pooled area as a pooling result. After pooling operations, the remained information expresses the feature characteristics better. A ReLU (rectified linear unit) layer usually placed after a pooling layer. The goal of the ReLU layer is to introduce nonlinear features into a CNN by forcing its output to zero when the input is negative. To continuously extract deep feature maps, multiple rounds of convolution-pooling-ReLU layers are often created in CNN architectures. As a result, these CNN architectures can compress the height and width of input images, while increasing the number of channels (*i.e.*, depth).

After grabbing enough features of an input image, the final step is how to identify or classify it. The role of a fully-connected layer is to observe the output of previous layer (generally a feature map containing high-level features), and then determine which features are the most relevant to a particular class. A fully connected layer maps the detected features from input images to a separable space, such as a decision of "defective" or "defect-free".

So far, there is no explicit criteria to guide designers to determine CNN architectures, such as the number of total layers, the number of convolution layers, the size and number of convolution kernels, etc. This is because a neural network is largely dependent on the size, type, hidden features, and complexity of processing tasks. In this work, we figure out how to choose a low-complexity and high-precision CNN architecture through the "trial and error" approach. In this way, an appropriate CNN architecture is proposed in Figure 3, which consists of only 1 convolution layer and 1 pooling layer.

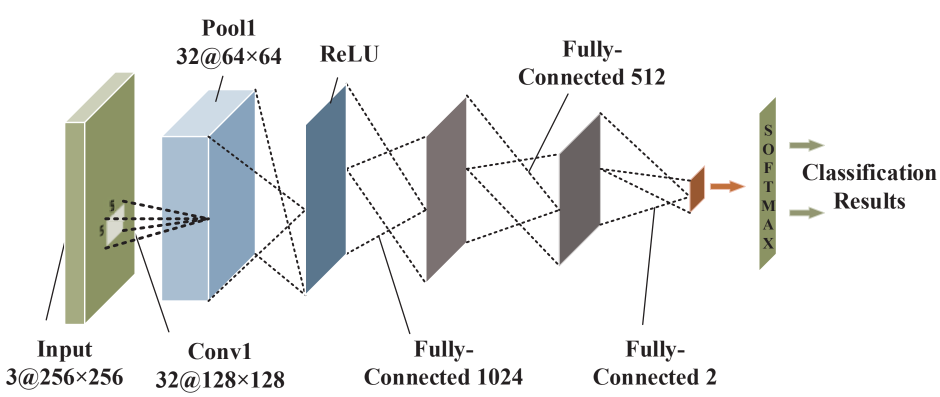

Figure 3: The Proposed Low-Complex CNN Architecture for Screw Quality Classification

As shown in Figure 3, the proposed CNN architecture consists of one input layer, one convolution layer, one pooling layer, one ReLU layer, three fully-connected layers, and a softmax layer. As the resolution of each screw head image is 256×256 pixels with 3 channels (RGB), the input layer has a dimension of 256×256×3. Then, in the convolution layer, 32 convolution kernels with a size of 5×5 are used. The padding option is chosen to be "same", so it makes sure that the output size after convolution is the same as the input size. Hence, the resultant feature map size after convolution is still 256×256×32. After the pooling layer, the size of these feature maps shrinks to 128×128×32. Next, the nonlinearity is added through the ReLU layer, the latest feature maps are passed to the three fully connected layers. Finally, the "defective" or "defect-free" classification decision is provided from the softmax layer. This CNN architecture has been described and implemented in software code using the Python language.

4. Methodology and Design Considerations

Figure 4 illustrates the flowchart of training and validating the proposed CNN architecture. The CNN training is conducted with an established dataset of 8,200 screw images with a dimension of 256×256 pixels. 4,100 screw images are included in a training dataset, while the other 4,100 screw images are included in a test dataset. That means that we use 4,100 training images to train our proposed CNN architecture, and use another 4,100 test images to validate the classification accuracy of the trained CNN architecture. The goal of CNN training is to optimize trainable parameters (weights and biases in each neuron) to maximize the chance of correct defects decision. All trainable parameters are randomly initialized during the training process. Then, the CNN training is based on gradient descent algorithm in the framework of TensorFlow. The gradient descent algorithm consists of two steps. The first step is a feedforward step, which calculates the output value of the CNN for an input image. The second step is a back propagation step, which deviates from the calculated output value of the CNN to modify the network parameters and to slightly improve its performance on an input image. In this manner, all trainable parameters will be updated iteratively.

Learning rate is an important custom parameter that affects the training convergence speed and final classification accuracy of CNNs. For a too high learning rate, the weight update is very large, so CNNs can converge quickly. But it also raises a problem that the weight is not accurate enough to achieve the best solution. Otherwise, for a too small learning rate, CNN training will converge very slowly or even impossible to learn at all. In this study, we carefully tuned the value of learning rate to obtain a good trade-off between the training convergence speed and final classification accuracy. The learning rate for training our proposed CNN architecture was chosen to be 0.001.

Instead of training each input image, multiple input images are entered as a batch (*e.g.*, epoch). The goal is to make the learning process less sharp and the convergence direction more consistent. A large batch size reduces training time and improves stability. However, a large batch size can lead to a decline in model generalization capability. In this study, we chose the epoch size to be 64.

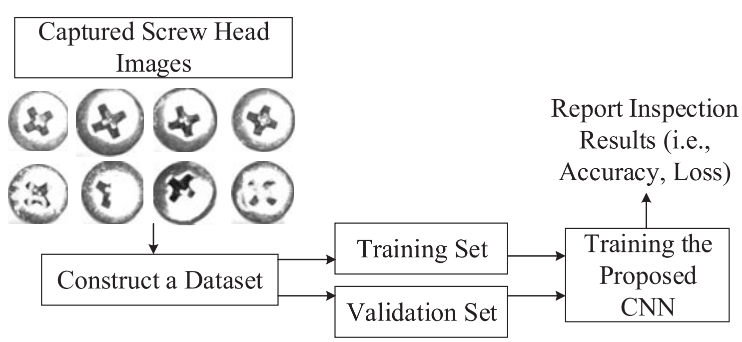

Figure 4: Flowchart of Training and Validating the Proposed CNN Architecture

5. Experimental Results and Discussions

Figure 5 shows how the detection accuracies vary with the number of training epoch. The final training accuracy can be as high as 99.4%, which is much better than conventional methods for screw defect inspection. The final validation accuracy is about 95.13%, which is comparable with the state-of-the-art works in the literature. Figure 6 plots the simulation results for the training and validation losses with respect to the number of training epoch. This figure shows our chosen learning rate is appropriate.

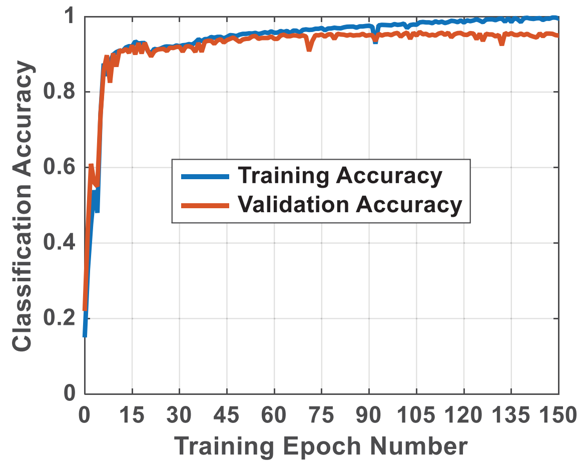

Figure 5: Classification Accuracy vs. Training Epoch Number

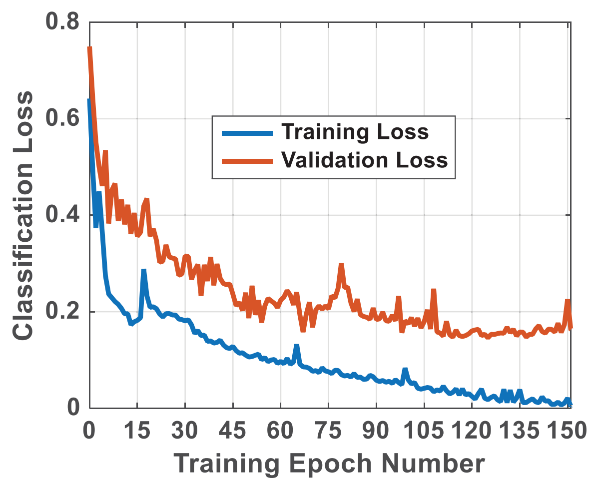

Figure 6: Classification Loss vs. Training Epoch Number

Table 1 summarizes the performance comparison of proposed low-complexity CNN with the state-of-the-art CNN architectures in the literature. Compared with (Cha et al, 2017; Silva et al, 2018; Chen et al, 2018; Song et al, 2018; Kim et al, 2018), the proposed CNN architecture is relatively simple to implement and obtains a comparable classification accuracy. Therefore, our proposed CNN architecture has the potential to be implemented in cost-effective hardware platforms, such as a low-cost embedded system consisting of only a microprocessor and limited memory space. Our proposed AI algorithm spends around 1 second to complete the quality inspection of a screw. Compared with human-conducted quality inspection, our algorithm has 2 or 3 orders of improvement in the detection throughput, hence enabling high-throughput automated production of construction materials.

Table 1: Comparison Summary of this Work with Existing CNN Architectures

	Purpose	CNN Architecture	Classification Accuracy
(Cha et al, 2017)	*crack detection*	*4 convolution layers 2 pooling layers*	*98%*
(Silva et al, 2018)	*concrete crack detection*	*13 convolution layers*	*92.27%*
(Kim et al, 2018)	*construction equipment*	*3 convolution layers*	*96%*
(Song et al, 2018)	*defective metal screw surface*	*3 convolution layers 3 pooling layers*	*98%*
(Chen et al, 2018)	*Defective fastener detection*	*6 convolution layers*	*92.78%*
This work	*defective screw head*	*1 convolution layer 1 pooling layer*	*95.13%*

6. Conclusions

In order to achieve rapid, high-throughput, and high-precision screw head quality inspection, a vision-based artificial intelligence algorithm is investigated in this work. The proposed convolutional neural network (CNN) architecture consists of only one convolution layer and one pooling layer, so it is a low-complexity network architecture with an acceptable high accuracy of 95.13%. This study has the great potential to replace human-conducted onsite inspections to realize high-throughput automated production of construction materials.

7. Limitations and Future Work

Despite the very high detection accuracy, this AI architecture is very complex. Thus, it is a challenge to implement our proposed AI algorithm is resource-limited computational platforms, such as an embedded system which is connected with a camera that captures raw screw head images. A typical embedded system consists of only a quad-core microprocessor and limited memory size. To solve this problem, I will investigate low-complexity, energy-efficient AI algorithm implementation, and adopt it to a typical hardware platform (*e.g.*, Raspberry PI) to achieve fast end-to-end quality inspection.

Acknowledgements

The authors would like to acknowledge the support by the 2019 CASA summer research grant of Southern Illinois University Carbondale.

References

Ahmed, M., Haas, C., & Haas, R. (2012). Using digital photogrammetry for pipe-works progress tracking. *Canadian Journal of Civil Engineering*, 39(9), 1062-1071.

Cha, Y., Choi, W., & Buyukozturk, O. (2017). Deep learning-based crack damage detection using convolutional neural networks. *Journal of Computer-Aided Civil and Infrastructure Engineering*, 32(5), 361-378.

Chen, J., Liu, Z., Wang, H., Nunez, A., & Han, Z. (2018). Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network. *IEEE Transactions on Instrumentation and Measurements*, 67(2), 257-269.

Fathi, H., Dai, F., & Lourakis, M. (2015). Automated as-built 3D reconstruction of civil infrastructure using computer vision: achievements, opportunities, and challenges. *Advanced Engineering Informatics*, 29(2), 149-161.

Kim, H., Kim, H., Hong, Y., & Byun, H. (2018). Detecting construction equipment using a region-based fully convolutional network and transfer learning. *ASCE Journal of Computing in Civil Engineering*, 32(2), 04017082.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278 – 2324.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436-444.

Martinez, P., Ahamd, R., & Hussein, M. (2019). A Vision-based system for pre-inspection of steel frame manufacturing. *Automation in Construction*, 97, 151-163.

Silva, W., & Lucena, D. (2018). Concrete cracks detection based on deep learning image classification. *Proceedings*, 2(8), 489.

Song, L., Li, X., Yang, Y., Zhu, X., Guo, Q., & Yang, H. (2018). Detection of micro-defects on metal screw surfaces based on deep convolutional neural networks. *Sensors*, 18, 3709.

TensorFlow, https://www.tensorflow.org/

Zawada, A., & Wierucka, I. (2018). A case study in technological quality assurance of a metric screw thread. *Measurement*, 114, 208-217.