applied sciences

MDPI

Article

Transfer Learning-Based Object Detection Model for Steel Structure Bolt Fastening Inspection

Jaehyun Choi, Minhui Ha and Jin Gang Lee *

School of Architectural Engineering, Korea University of Technology and Education, Cheonan 31253, Republic of Korea; jay.choi@koreatech.ac.kr (J.C.); alsgml779@koreatech.ac.kr (M.H.)
* Correspondence: jglee@koreatech.ac.kr

Abstract: As improper inspection of construction works can cause an increase in project costs and a decrease in project quality, construction inspection is considered a critical factor for project success. While traditional inspection tasks are still mainly labor-intensive and time-consuming, computer vision has the potential to revolutionize the construction inspection process by providing more efficient and effective ways to monitor the progress and quality of construction projects. However, previous studies have also indicated that the performance of vision-based site monitoring heavily relies on the volume of training data. To address the issues of challenging data collection at construction sites, this study developed models using transfer learning-based object detection models incorporating data augmentation and transfer learning. The performance of three object detection algorithms was compared based on average precision and inference time for detecting T/S bolt fastening of steel structure. Despite the limited training data available, the model's performance was improved through data augmentation and transfer learning. The proposed inspection model can increase the efficiency of quality control works for building construction projects and the safety of inspectors.

Keywords: construction inspection; computer vision; deep learning; object detection

Citation: Choi, J.; Ha, M.; Lee, J.G. Transfer Learning-Based Object Detection Model for Steel Structure Bolt Fastening Inspection. *Appl. Sci.* **2023**, *13*, 9499. https://doi.org/10.3390/app13179499

Academic Editors: Zhen Lei, Sungkon Moon, Joo-Sung Lee and Kyubyung Kang

Received: 10 July 2023
Revised: 7 August 2023
Accepted: 14 August 2023
Published: 22 August 2023

1. Introduction

To ensure the success of construction projects, it is imperative to incorporate a comprehensive quality management plan that aligns with the construction schedule and budget plan [1]. The inspection of the construction process holds immense significance in maintaining quality control; however, the increasing complexity of modern construction projects has made this task considerably more difficult and hazardous [2]. Moreover, the labor-intensive nature of the construction industry presents additional challenges, as the associated risks directly contribute to a decline in both project quality and productivity [3]. Within this context, the convergence of technologies including artificial intelligence (AI), drones, big data, and the Internet of things (IoT) throughout industries is gaining momentum as a result of Industry 4.0 trends aimed at overcoming low productivity, safety, and quality of construction works. Data that had to be collected manually by the construction site manager can be automatically collected with IoT technology, and large-scale big data can be analyzed using AI algorithms. Regarding inspection work for quality management, the construction progress can be recorded with a camera and analyzed with deep learning algorithm-based computer vision technology. Recently, research and applications for the automation of construction inspection using computer vision technology are gaining momentum due to the outstanding performance of deep learning algorithms such as convolutional neural networks (CNN) [4]. Several studies have demonstrated the applicability of computer vision algorithms for quality management at the construction site by detecting construction objects such as structural damage [5], concrete samples [6], and cropped bolts [7].

However, previous studies have indicated that the performance of deep learning-based algorithms is significantly influenced by the amount of training data, particularly

the quantity of images used to train the model [8]. Due to the unique nature and variability of construction projects, collecting training images for construction inspection can be both time-consuming and hazardous [9]. This study aims to address the challenges associated with image collection by implementing data augmentation techniques and developing object detection models based on transfer learning. Data augmentation is a technique used to artificially increase dataset size by adjusting the parameters of the image dataset [10]. And transfer learning is a method of reusing a pre-trained model on a new test to improve algorithm performance with small datasets [11].

Specifically, this study evaluates the performance of three deep learning-based object detection models including Faster RCNN, RetinaNet, and YOLOv3. With collected images of steel frames with bolts, three object detection models are trained and tested for their accuracy and speed performance. As a result of the comparison, the Faster R-CNN model was better in terms of accuracy, and the YOLOv3 model was better in terms of speed. In addition, the applicability of transfer learning to the YOLOv3 model was tested to improve accuracy. The process and results of this case study demonstrate the feasibility of utilizing transfer learning algorithms for bolt fastening inspection tasks, which traditionally rely on visual inspection by human inspectors. This research contributes to overcoming the limitations of labor-intensive inspection activities and has the potential to enhance productivity in quality management.

2. Literature Review

Construction inspection refers to testing, reviewing, and verifying construction materials, methods, and products to ensure adequate quality control of construction works following the drawings or specifications of facilities. According to the Korean Ministry of Land, Infrastructure and Transport's construction inspection work guidelines, inspection work is performed by visual inspecting, surveying, testing, and supervising. Inspectors normally rely on a visual inspection, which is labor-intensive because the managers need to visit and check the specifications, numbers, and conditions to ensure that the construction is consistent with the design drawings [12]. It is difficult for a single site manager to collect comprehensive construction site information by visiting specific places in large-scale construction sites. This can impose an excessive inspection workload on individual site managers, resulting in decreased work efficiency and increased exposure to safety accidents [13]. As mentioned earlier, the shortage of construction site personnel due to aging and COVID-19 is increasing the need for automation of inspection tasks. In this situation, the inspection work should be performed for the entire target of construction progress, but it is not possible to do so. As a result, this becomes an important cause of quality problems in building projects.

To overcome such problems, construction managers have utilized cameras for observing construction progress. Since it is possible to install CCTV (Closed Circuit Television) with a safety budget for construction projects, site monitoring using cameras is becoming common practice [14]. Observing a construction site and collecting site information through recorded video requires enormous time and effort from the construction manager [15]. In this aspect, the rise in the availability of construction site videos and the rapid advancements in deep learning algorithms have generated an increasing need for leveraging computer vision to analyze site conditions. Computer vision makes it possible for machines to understand images like humans (or better than humans in terms of capability and speed). Thereby, its application area is expanding, such as facial recognition technology, medical image analysis, and self-driving cars. Technically, the researchers developed deep learning-based algorithms to identify and manipulate specific image characteristics by pixel level to understand the information in images [16]. As a result, CNN-based deep learning algorithms are showing excellent performance in the classification, detection, localization, and tracking of objects in images and videos. And these recent developments and applications of computer vision algorithms can facilitate automated site monitoring

by identifying various resources present at the construction site, including their quantity, location, and status [17].

Research using deep learning-based object detection algorithms in construction sites continues to increase and is mainly focusing on site monitoring, construction safety management, and quality management. Computer vision technology has been tested to enable the automated detection and tracking of heavy equipment, such as excavators, commonly employed in civil engineering sites [14,18,19]. Also, computer vision algorithms are being applied to monitor construction workers, enabling the classification of their behavioral patterns [20], or the analysis of construction work productivity [21]. Kim et al. (2018) proposed a combined application of CNN and LSTM (long short-term memory) to detect construction equipment and classify work activities [22]. In other study, the construction equipment detection model was applied to estimate the productivity of earthwork [23].

As part of site monitoring, many researchers have actively been testing computer vision for construction safety management to lower accident and death rates. Shim et al. (2019) utilized YOLOv3, a deep learning-based object detection algorithm, to develop and implement an algorithm that effectively identifies risks by analyzing the situations involving construction machine operators and the surrounding workers during earthwork activities [24]. Similarly, Kim et al. (2019) used a CNN-based construction object detection model to identify collisions at construction sites [25]. Lee et al. (2019) employed YOLOv3 to develop an algorithm that automatically detects whether construction workers are wearing proper safety equipment [26].

In terms of quality control and inspection, which is the field of this research, several studies have focused on utilizing object detection algorithms for quality management, particularly in the detection and classification of damages in constructed structures. For instance, Park et al. (2019) conducted real-time evaluations of structural safety by detecting surface damages such as cracks, peeling, spalling, and rebar exposure [5]. An et al. (2017) employed a hybrid image scanning system mounted on unmanned aerial vehicles (UAVs) to capture image data of concrete samples, using a convolutional neural network (CNN) to detect surface conditions [6]. Lee et al. (2019) utilized a deep learning algorithm based on the regional convolutional neural network (R-CNN) to detect bolt images, enabling the estimation of bolt-loosening angles based on cropped bolt images [7]. Zhou et al. (2021) utilized a CNN-based YOLOv3 algorithm for detecting fractured bolts in steel bridges [27]. In addition to construction sites, Bahrami and Wang (2022) proposed high-resolution and temporal context region-based (HRTC R-CNN) for corrosion defect detection on shipping containers [28]. These studies demonstrate the diverse applications of object detection algorithms for quality management, providing valuable insights into the identification and assessment of structural damages in construction projects.

In the computer vision-based approaches that have been introduced so far, large-scale image data collection is essential to train deep learning algorithms. For example, An et al. (2017) developed a deep convolutional neural network with 200,000 images of concrete cracks [6], and Lee et al. (2019) used 50,000 images in the CIFAR-10 dataset [29] for training pre-training [7]. Bahrami and Wang (2022) also collected 10,000 images and separated the dataset into three parts: the training set had 8500 images, the validation set contained 1500 images, and the test set contained 1500 images. Recording video and collecting images manually requires time-consuming efforts. In addition, extensive amounts of image collection also require significant annotation work. Generally, previous studies collected images and annotated specific objects in images with annotation software, such as Labelimg, LabelMe, and Amazon Mechanical Turk [30,31]. Researchers are struggling with extensive amounts of annotation work. For instance, Soltani et al. (2016) estimated the annotation time as 11 s per one image [32], and Luo et al. (2018) spent more than 200 h to manually label 7790 images [18]. Overall, the difficulties of image data collection and annotation can be an obstacle to applying vision-based quality management at construction sites.

3. Methodology

This research proposed a systemic approach to applying an object detection algorithm for construction inspection. To address the difficulties of data collection, this research tested data augmentation and transfer learning for improving the performance of computer vision algorithms with a small amount of data. This process includes (1) data collection and configuration, (2) data augmentation, (3) object detection algorithm selection, and (4) transfer learning.

3.1. Data Collection and Configuration

Deep learning models rely on the availability of training, validation, and test datasets for effective training and evaluation. The training set is used to train the model, while the validation set serves to assess the performance of the trained model. Typically, different parameters and models are tested on the validation set to identify the optimal model that achieves the best performance for the intended analysis. Once the preferred model is chosen based on the validation set, the test set is employed to evaluate the expected performance of the selected model. It is important to acknowledge that the validation set has already been utilized in multiple models, which introduces the possibility of random selection bias as models may have been chosen based on better-observed performance. To mitigate this potential bias, the performance of the final model is evaluated using a separate test set containing data that was not included in the training process. This approach helps reduce the likelihood of biased model selection and provides a more accurate assessment of the model's true performance.

3.2. Data Augmentation

Object detection networks require three types of information: an image, a corresponding class label, and a bounding box that specifies the object's position within the image. Bounding boxes consist of a combination of numbers indicating the location of four points, and the number of combinations increases according to the number of bounding boxes in the image. To utilize the collected images as an object recognition database, it is necessary to annotate each image by assigning classification information to identify the target object and indicating its precise location within the image using bounding boxes. This process ensures the images are appropriately labeled, facilitating effective object recognition tasks. The most common way to obtain construction site images would be through manual data collection and annotation. In general, previous studies in construction have collected site images directly from site videos and manually annotated them by drawing bounding boxes on the positions of objects in images. However, due to the unique characteristics of construction site conditions, sufficient training data collection is often challenging in terms of time and cost [33,34]. Insufficient training data can cause issues like overfitting and underfitting in deep learning models, consequently limiting the practicality and applicability of vision-based approaches within the construction industry [9].

To address the challenges associated with limited image data collection and the complex task of annotation, this research employs a data augmentation technique. Data augmentation involves generating new artificial data within a virtual environment, using the original dataset as a basis, rather than relying solely on data collected directly from real-world construction sites. For image data, a variety of methods such as cropping, rotating, flipping, and color adjustments are applied to the original dataset, effectively increasing the volume and diversity of available training data [10]. This approach enhances the robustness and generalization capabilities of deep learning models used in vision-based applications for the construction industry.

3.3. Object Detection Algorithm Selection

When selecting object recognition algorithms, the primary consideration should be given to either object detection accuracy and speed, depending on its purpose, and it is needed to train and evaluate multiple algorithms to determine the most suitable one. Object

detection accuracy is determined by comparing the predicted results of the model with the ground truth, which includes the object's bounding box position and class label. Evaluating the accuracy of object detection algorithms typically involves metrics such as the precision–recall (PR) curve and mean average precision (mAP). To assess the proposed model's performance, standard evaluation indices such as precision and recall, based on a confusion matrix, are utilized. In this study, the presence or absence of detection was determined based on the bounding box criterion, including True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN). However, as detection outcomes vary with the reliability threshold, precision and recall calculated from TP, FP, FN, and TN cannot be expressed as fixed values. Generally, previous studies have used the reliability threshold based on the intersection over union (IoU) that measures the overlapping area between the ground truth and predicted bounding boxes [35]. In the computer vision domain, 0.5 or 0.75 is the common threshold value for IoU, and this study selected a lower IoU threshold of 0.5 to consider roughly shaped target objects such as fastened bolts [36]. In addition to the IoU threshold, precision and recall would be highly affected by the confidence threshold of the detector. The object detection model provides confidence scores of all detected objects that reflect the probability that the bounding box includes target objects, and the confidence threshold can be used to filter out false positives [37]. Therefore, precision and recall exhibit an inverse relationship, with high precision tending to correspond to low recall, and vice versa. Generally, detection performance is evaluated using a precision–recall graph, where precision represents the ratio of correctly detected data among all detection results, reflecting how well the predicted results align with the actual objects. The validation of this chapter aims to test the overall performance of the following detection algorithms, and thus rather than measuring precision and recall rates with specific confidence threshold values, the average precision (AP) and mean average precision (mAP) are used as an overall detection performance indicator. The AP is calculated by obtaining the area under the precision–recall curve to provide a quantitative evaluation of object detection models. And mAP serves as the average of AP values across multiple objects.

Regarding the speed of the algorithm, the inference time refers to the time it takes for a trained model to process an input image or video and generate the desired output, such as object detection or image classification results. It is a critical factor in real-time or time-sensitive applications where timely processing is required. The inference time is influenced by various factors, including the complexity of the model architecture, the size and resolution of the input data, the available computational resources (e.g., CPU or GPU), and any optimizations or hardware accelerations implemented.

Table 1 compares the performance of the three algorithms [38]. Table 1 indicates that Faster R-CNN achieves high accuracy; however, it exhibits slower object detection speed compared to the other algorithms. Zhou et al. (2019) conducted a study utilizing the COCO (Common Objects in Context) dataset to evaluate the performance of representative object detection algorithms, including Faster R-CNN, RetinaNet, and YOLOv3, based on average precision and inference time [38]. The COCO dataset, developed by Microsoft, is a large-scale image dataset designed for tasks such as object detection, segmentation, key points detection, and caption generation. It comprises 330,000 images, 1.5 million object instances, 80 object categories, and 91 stuff categories [39]. In terms of speed (inference time), YOLOv3, a 1-stage detector, operates at a rate of 45 frames per second, enabling real-time detection. YOLOv3 exhibits the fastest object detection speed, ranging from 24 to 50.3 ms, but shows a slightly lower precision of 10–20% compared to the other two models. RetinaNet, developed to address the lower accuracy of 1-stage detector algorithms, exhibits slower object detection speed compared to YOLOv3 but achieves higher detection accuracy. While RetinaNet is less accurate than 2-stage detectors, it offers faster detection speed. Faster R-CNN, a two-stage detector, and an improved version of the classic R-CNN object detection algorithm feature slower detection speed due to the prerequisite RoI (Region of Interest) step for classification [1]. However, it should be noted that the performance of these algorithms may vary when tested with the new training data, as obtaining large-scale

data like the COCO dataset from construction sites can be challenging. Therefore, in this study, the aim is to enhance detection performance by applying data augmentation and transfer learning while testing these three aforementioned algorithms.

Table 1. Comparison of performance among algorithms trained with COCO dataset.

Algorithms	Minimum Average Precision (%)	Maximum Average Precision (%)	Minimum Inference Time per Image (ms)	Maximum Inference Time per Image (ms)
YOLOv3	28	33	23	50
RetinaNet	34	38	55	115
Faster R-CNN	39	41	70	145

3.4. Transfer Learning

Transfer learning is a technique that enhances performance by utilizing pre-trained parameters from extensive data in related domains [11]. In transfer learning, the process involves removing the classifier from the original model, adding a new classifier tailored to the specific objective, and training the modified model. When working with a small dataset, transfer learning leverages the feature extraction layer's parameters from a model trained on a large-scale image dataset, and fine-tunes the weights using the available dataset, which only has parameters related to the final classification layer. Alternatively, when dealing with a large dataset, it is possible to update parameters across all pre-trained layers through the training process. Figure 1 illustrates three strategies for transfer learning. Strategy 1 entails training the model from scratch, utilizing only the pre-trained model's architecture, which requires a substantial dataset and high-performance computing systems. In Strategy 2, low-level layers extract general features, while high-level layers focus on capturing specific features, thus determining the extent of retraining the neural network parameters. This strategy proves useful when working with a large dataset or a smaller model. Strategy 3 involves freezing the convolutional base and solely training the classifier. This method is suitable for scenarios with limited computing power or small datasets. Therefore, the third strategy was applied in the case study to address the issues of a small training dataset.

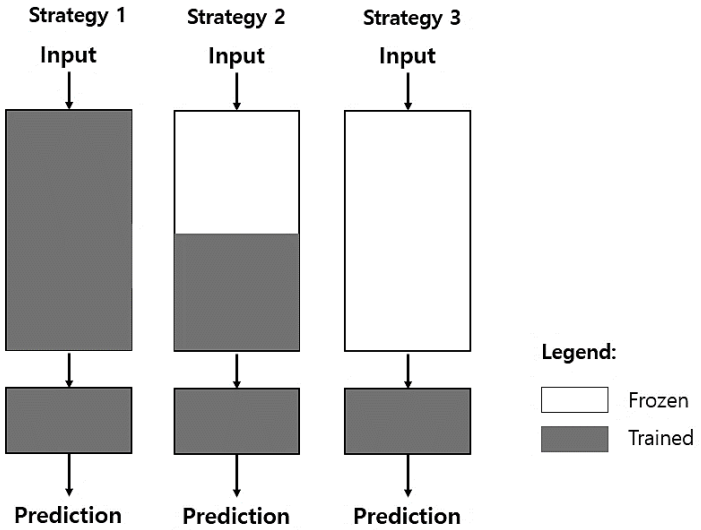

Figure 1. Three strategies of transfer learning.

- Strategy 1: retrain the entire model;
- Strategy 2: freeze some layers of the convolutional base and retrain the remaining layers and classifier;
- Strategy 3: freeze the convolutional base and retrain only the classifier.

4. Case Study

This research tested the proposed approach for real-time inspection of T/S bolts in steel structures. The process of the case study is in the following order: (1) data collection and pre-processing; (2) training and testing candidate object detection model under the same conditions by comparing their average precision and inference time; (3) transfer learning for selected object detection model for improving model performance; and (4) discussions to determine the optimal algorithm.

4.1. Data Collection and Pre-Processing

This research focuses on inspecting the fastening bolts of steel structures, which have become increasingly important due to the structural safety of large-scale construction projects [27]. The conventional method requires inspectors to visually inspect each bolt to ensure proper fastening [40]. To develop an object detection model for bolt fastening inspection, two methods were employed to collect image data. Firstly, image data was collected during visits to university building construction sites where steel structures were being erected. Secondly, images of bolts in steel structures were obtained from internet websites. By using a web-crawling approach, images of bolts on steel frames were collected from a Google image search. This web-crawling method utilized several keywords for searching training images, such as 'T/S bolt', 'bolt on steel frame', and 'T/S bolt fastening'. After searching and collecting images, irrelevant images are manually deleted from the training dataset. This deletion process is decided based on the similarity with real images collected at the construction site. Next, 30 images were collected from construction sites, and a total of 174 relevant images were retrieved from internet searches, resulting in a dataset of 204 images. The dataset was then divided into 154 training images, 20 validation images, and 30 test images. After image collection, the LabelImg tool, implemented in Python, was utilized to annotate bounding boxes around the target objects. This process generated corresponding information in XML format, where the class indicated the type of object, (x, y) denoted the upper-left coordinate of the bounding box, and (w, h) represented the width and height of the bounding box. As the detection model aimed to identify fractures in T/S bolt pintails, the bolts could be categorized into two classes: "no pintail", indicating a fastened bolt where the pintail is not visible, and "pintail", indicating an unfastened bolt where the pintail is visible.

4.2. Data Augmentation

In this study, data augmentation was carried out using the Python library "imgaug". This library enables the augmentation of training images along with their corresponding labeling information, allowing for augmentation even when the data is already labeled. The original training images underwent augmentation processes such as rotation (every 15 degrees), vertical and horizontal flipping, as well as adjustments to brightness, contrast, and color. Since various images can be collected depending on the angle taken by the actual camera, this task is effective in improving the reality of the training dataset [10]. Through 11 iterations of data augmentation, a total of 1694 training images, 220 validation images, and 330 test images were obtained (Figure 2).

4.3. Model Training and Comparison

To determine the most suitable algorithm for bolt fastening inspections, this study conducted a comparative analysis of three deep learning-based object detection algorithms: YOLOv3, RetinaNet, and Faster R-CNN. Various variables were considered during the training of the deep learning models. The epoch, which signifies the number of passes the machine learning algorithm completes on the entire training dataset, was employed (Figure 3). Batch size, representing the number of training examples utilized in a single iteration of neural network training, was also considered. Additionally, the learning rate, a constant determining the step size in each iteration, and the optimal functions such as Sigmoid, RMSProp, and Adam were explored. Adam, combining the strengths

of Momentum and RMSProp, emerged as the most widely used training optimization technique due to its ability to incorporate past changes in gradient and prioritize recent information. In this study, the batch size and number of epochs were fixed at 16 and 150, respectively, while Adam was selected as the optimal function for training each object detection model.

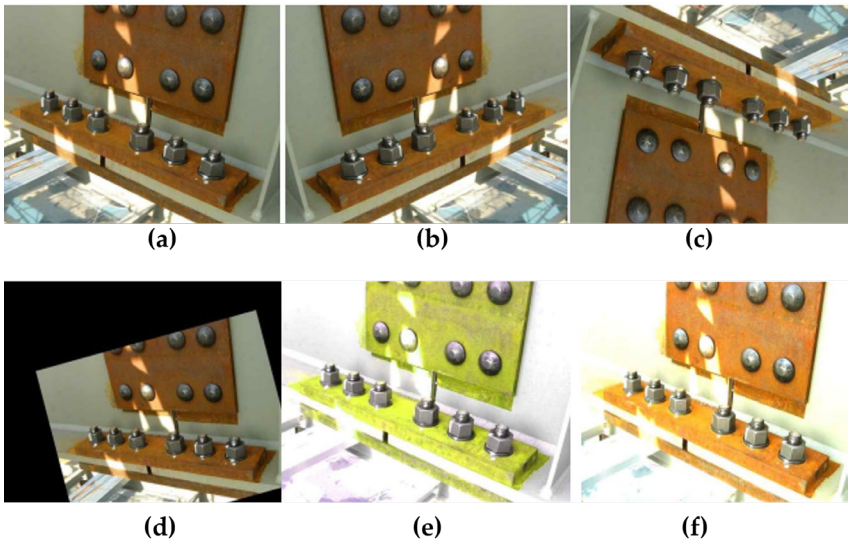

Figure 2. Examples of image augmentation through data augmentation techniques: (**a**) original image, (**b**) horizontal flipping, (**c**) vertical flipping, (**d**) rotation, (**e**) color adjustment, and (**f**) brightness adjustment.

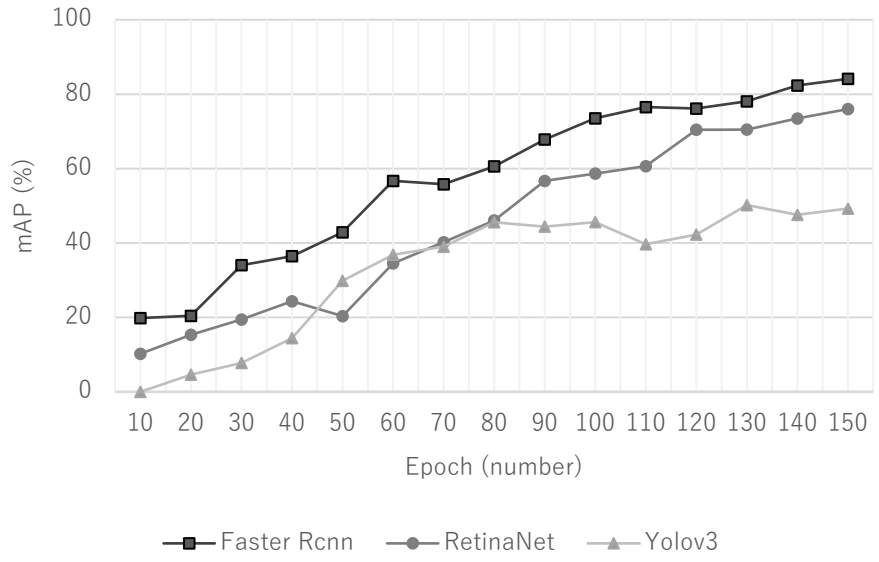

Figure 3. Comparison of mAP among algorithms.

Table 2 presents the performance test results for the three object detection algorithms. The loss value represents the disparity between predicted and correct values during machine learning iterations, with the training process aimed at minimizing this value. Mean average precision (mAP) serves as the average of AP values across multiple objects. Inference time per image denotes the duration required to detect a single frame. The detection speed of the algorithms was measured by determining the inference time per image during the detection of test images. Following a comparison of the accuracy and speed of the object detection algorithms, Faster R-CNN exhibited the highest accuracy (84.12%), followed by

RetinaNet and YOLOv3. YOLOv3 demonstrated the fastest performance, with an inference time of 0.0089 s per frame, followed by RetinaNet and Faster R-CNN.

Table 2. Comparison of performance among algorithms.

Algorithms	Validation of Classification Loss (%)	Validation of Objectness Loss (%)	mAP (%)	Inference Time per Image (s/img)
YOLOv3	4.372	0.6885	49.25	0.0089
RetinaNet	1.115	0.2787	75.99	2.569
Faster R-CNN	1.093	0.2732	84.12	29.29

In Figure 4, the test result images are displayed, depicting 4 fastened T/S bolts and 10 unfastened T/S bolts. The ground truth of the test image consists of 10 bounding boxes classified as the pintail class and 4 bounding boxes as the no pintail class. Figure 4b–d demonstrate the outcomes of test image detection using models trained with the YOLOv3, RetinaNet, and Faster R-CNN algorithms, respectively. For example, in Figure 4b, all bounding boxes are classified as 'pintail'. And blue bounding boxes are the 'pintail' class, and bounding boxes with the sky (light) blue color are the 'no pintail' class in Figure 4d. Lastly, in Figure 4d, blue bounding boxes are the 'pintail' class, and red bounding boxes are the 'no pintail' class.

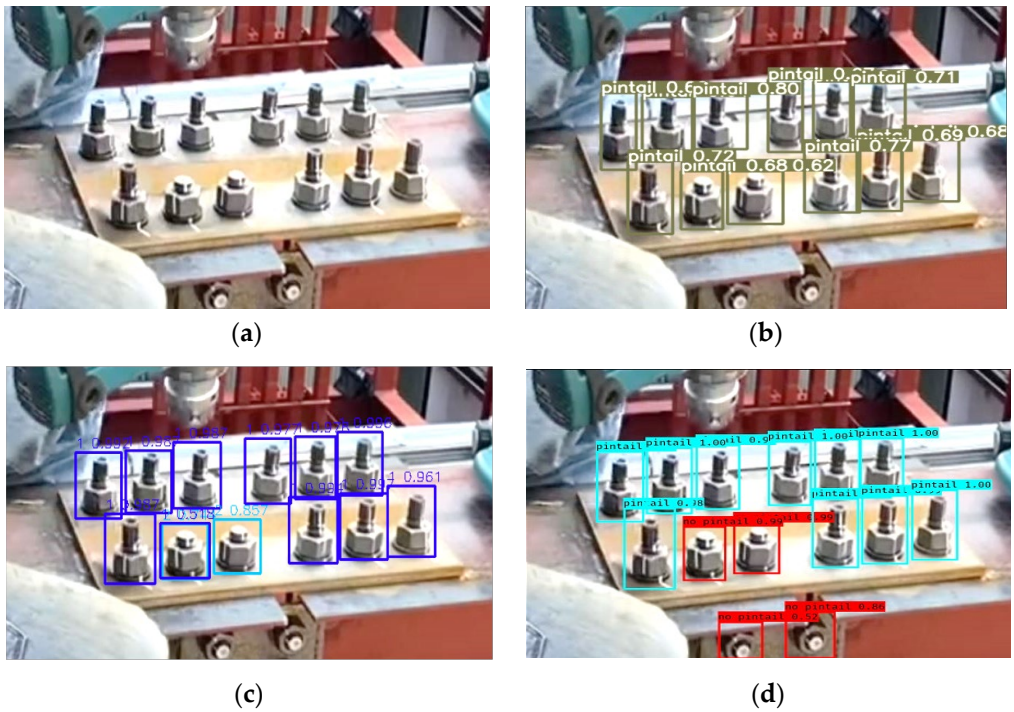

Figure 4. Test results of comparing deep learning algorithms. (**a**) Test image; (**b**) YOLOv3; (**c**) RetinaNet; (**d**) Faster R-CNN.

In terms of the detection result, in Figure 4b, YOLOv3 detected only 12 out of the 14 objects and failed to identify 2 objects in the no pintail class located below the center of the test image. The class probability for each object ranged from 62% to 80%. YOLOv3 exhibited the fastest inference time per image (0.0089 s) among the three algorithms. In Figure 4c, RetinaNet successfully detected 12 objects in the test image but also missed 2 objects in the no pintail class positioned below the center of the test image. The class probability for each object ranged from 51% to 99%, indicating better accuracy than YOLOv3. The

inference time per image for RetinaNet was 2.569 s. Finally, Figure 4d illustrates that Faster R-CNN detected all 14 objects in the test image, with class probabilities ranging from 52% to 100%, showcasing the highest accuracy among the three algorithms. However, Faster R-CNN exhibited the slowest inference time per image (29.29 s) compared to the other two algorithms.

4.4. Transfer Learning

Based on the performance comparison results, Faster R-CNN demonstrated the highest accuracy, while YOLOv3 exhibited the shortest inference time per image among the three algorithms. Considering the requirement for real-time inspection, it becomes challenging to develop object detection models with slower detection speeds such as Faster R-CNN and RetinaNet. Hence, this study aimed to enhance the accuracy of the YOLOv3 model and employed transfer learning to improve its performance.

TensorFlow Keras offers a range of transfer learning models, and ResNet50 was chosen as the network for transfer learning in this research. Since the dataset used in this study is relatively small, the third strategy was employed. The parameters of the pre-trained neural network are kept unchanged, and the existing convolutional base as a feature extractor is utilized, followed by the introduction of a new classifier [41]. Figure 5 presents the mAP values obtained during the model training using Strategy 3 transfer learning, in which mAP values consistently converge after 100 epochs.

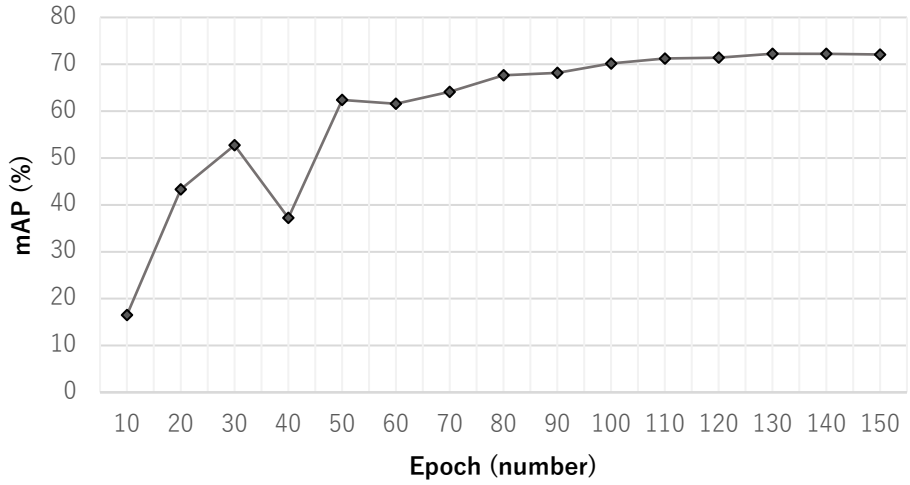

Figure 5. mAP value after transfer learning (YOLOv3).

4.5. Results and Discussions

In the previous chapter, YOLOv3 was considered suitable for real-time inspection due to its significantly shorter inference time per image compared to other algorithms. However, its accuracy was comparatively lower than the other algorithms. Transfer learning was applied to improve the detection performance of the YOLOv3 model. Table 3 presents a comparison of the performance between the YOLOv3 model after transfer learning, YOLOv3 before transfer learning, RetinaNet, and Faster R-CNN models.

When comparing the performance of the transfer learning YOLOv3 model with RetinaNet and Faster R-CNN after transfer learning, the objectness loss validation decreased by 0.5% compared to before transfer learning. Consequently, the differences compared to RetinaNet and Faster R-CNN decreased by 0.0024 and 0.0279, respectively. The mean average precision (mAP) improved by 22.85% compared to YOLOv3 before transfer learning, resulting in reduced differences of 3.9% and 8.13% compared to RetinaNet and Faster R-CNN, respectively. Additionally, the inference time per image decreased to 0.045 s, which is nearly twice as fast as YOLOv3 before transfer learning.

Table 3. Algorithm performance comparison after transfer learning.

Algorithm	Validation of Classification Loss (%)	Validation of Objectness Loss (%)	mAP (%)	Inference Time per Image (sec/img)
YOLOv3 (after transfer learning)	3.382	0.1811	72.09	0.0042
YOLOv3 (before transfer learning)	4.372	0.6885	49.25	0.0089

Lastly, when comparing the transfer learning YOLOv3 model with Faster R-CNN, which exhibited the highest accuracy, the former displayed slightly lower accuracy with a mAP difference of 8.13%. However, the detection speed was 700 times faster. Therefore, the transfer learning YOLOv3 model can be considered an appropriate detection model for real-time bolt fastening at construction sites.

5. Conclusions

In the construction industry, performing comprehensive inspections has become increasingly challenging as building projects grow in size and complexity. To address these difficulties, the industry has been embracing computer vision algorithms, in particular, have shown promise in enhancing the accuracy and efficiency of construction inspections. However, the limited collection of sufficient data poses a significant challenge to the applicability of vision-based approaches. Therefore, to address the limitation of data collection on construction sites, this study aims to develop and evaluate deep learning-based object detection models incorporating data augmentation and transfer learning approaches.

The focus of this study is on T/S high-tension bolts, commonly used to join steel materials in construction projects. To train the object detection models, an image dataset was collected by searching Internet websites and visiting steel construction sites. The dataset underwent preprocessing to prepare it for training the deep learning algorithms, including Faster R-CNN, RetinaNet, and YOLOv3. The performance of these models was then compared to identify the most suitable algorithm for construction inspections. Notably, there were clear differences in terms of accuracy and speed among the models based on each algorithm. Faster R-CNN demonstrated the highest accuracy, but its object detection speed was too slow for real-time construction inspections. On the other hand, YOLOv3 exhibited the fastest object detection speed but the lowest accuracy among the three algorithms. RetinaNet fell between Faster R-CNN and YOLOv3 in terms of detection accuracy and speed. Given the importance of fast detection speed for real-time construction inspection, this study employed transfer learning to improve the accuracy of the YOLOv3-based model. By leveraging transfer learning, the YOLOv3 model's mean average precision (mAP) value increased by 22.84%, from 49.25% to 72.09%. As a result, a YOLOv3-based object detection model suitable for construction inspections was developed through transfer learning, achieving similar accuracy to RetinaNet even with the smallest inference time.

The findings of this study indicate that the proposed object detection model can significantly enhance the efficiency of construction inspections with a small amount of training data. By leveraging the proposed vision-based approaches for checking bolt fastening on steel frames, project managers can conduct inspections with higher accuracy and efficiency compared to traditional visual judgment methods employed by inspectors. Although there are possible benefits of the proposed methodology in this research, there are still limitations to be addressed in further research. This research validated the proposed methodology with bolt fastening inspection on the steel frame. The proposed approach can be utilized for other construction inspection tasks, such as form installation, rebar installation, or finishing works. But it needs additional tasks of data customization, training algorithm, and validation. Hence, it is required to investigate the specific characteristics of inspection work, and to develop a vision-based algorithm for diverse features of image data from construction sites. Moreover, the proposed model can be applied to real construction sites,

allowing inspections to be conducted in areas with limited accessibility or requiring elevated work, using images captured by drones or robots. Therefore, further research should be conducted to integrate deep learning technologies into unmanned aerial vehicles, such as drones, and to develop diverse datasets and models for inspecting different construction phases. After more research achievement and technical development, vision-based inspection can contribute to improving both the efficiency of construction site management and the quality of buildings.

Author Contributions: Conceptualization, M.H., J.G.L. and J.C.; methodology, M.H.; validation, M.H.; investigation, J.G.L.; writing—original draft preparation, M.H. and J.C.; writing—review and editing, J.G.L.; project administration, J.C.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2021R1F1A1062967) and the Education and Research Promotion Program of KOREATECH in 2023.

Data Availability Statement: Some data that support the findings of this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Iyer, K.C.; Jha, K.N. Factors affecting cost performance: Evidence from Indian construction projects. *Int. J. Proj. Manag.* **2005**, *23*, 283–295. [CrossRef]
2. Zhang, X.; Bakis, N.; Lukins, T.C.; Ibrahim, Y.M.; Wu, S.; Kagioglou, M.; Aouad, G.; Kaka, A.P.; Trucco, E. Automating progress measurement of construction projects. *Autom. Constr.* **2009**, *18*, 294–301. [CrossRef]
3. Yates, J.K.; Epstein, A. Avoiding and minimizing construction delay claim disputes in relational contracting. *J. Prof. Issues Eng. Educ. Pract.* **2006**, *132*, 168–179. [CrossRef]
4. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1–9. [CrossRef] [PubMed]
5. Park, J.H.; Kim, T.H.; Choo, S.Y. A development on deep learning-based detecting technology of rebar placement for improving building supervision efficiency. *J. Archit. Inst. Korea* **2020**, *36*, 93–103.
6. An, Y.K.; Jang, K.Y. Deep learning-Based structural crack evaluation technique through UAV-mounted hybrid image scanning. *J. Korean Assoc. Spat. Struct.* **2017**, *17*, 20–26.
7. Lee, S.Y.; Hyeon, T.K.; Park, J.H.; Kim, J.T. Bolt-loosening detection using vision-Based deep learning algorithm and image processing method. *J. Comput. Struct. Eng. Inst. Korea* **2019**, *32*, 265–272. [CrossRef]
8. Yu, J.; Farin, D.; Krüger, C.; Schiele, B. Improving person detection using synthetic training data. In Proceedings of the IEEE International Conference on Image Processing, Hong Kong, China, 26–29 September 2010.
9. Lee, J.G.; Hwang, J.; Chi, S.; Seo, J. Synthetic Image Dataset Development for Vision-Based Construction Equipment Detection. *J. Comput. Civ. Eng.* **2022**, *36*, 04022020. [CrossRef]
10. Khosla, C.; Saini, B.S. Enhancing performance of deep learning models with different data augmentation techniques: A survey. In Proceedings of International Conference on Intelligent Engineering and Management, London, UK, 17–19 June 2020.
11. Lee, M.H.; Yoo, Y.J.; Jo, Y.W.; Lee, J.K. Transfer learning for object detection in infrared image. In Proceedings of the Conference of the Korean Institute of Communications and Information Sciences, Online, 13 November 2020.
12. Jo, S.; Lee, J.G.; Choi, J. A Framework of Automating Inspection Task Generation for Construction Projects. *Korean J. Constr. Eng. Manag.* **2023**, *31*, 40–50.
13. Yu, J.; Son, B. A Study on Benchmarking the Countermeasures Strategy for Tackling the Construction Labor Shortage-Focusing UK's MMC & Singapore's Buildability. *Korean J. Constr. Eng. Manag.* **2022**, *30*, 54–64.
14. Kim, J.; Ham, Y.; Chung, Y.; Chi, S. Systematic camera placement framework for operation-level visual monitoring on construction jobsites. *J. Constr. Eng. Manag.* **2019**, *145*, 04019019. [CrossRef]
15. Hwang, J.; Kim, J.; Chi, S.; Seo, J. Development of training image database using web crawling for vision-based site monitoring. *Autom. Constr.* **2022**, *135*, 104141. [CrossRef]
16. Anitha, A.; Shivakumara, P.; Jain, S.; Agarwal, V. Convolution Neural Network and Auto-encoder Hybrid Scheme for Automatic Colorization of Grayscale Images. In *Smart Computer Vision*; Springer: Cham, Switzerland, 2023; pp. 253–271.
17. Chi, S.; Caldas, C.H. Automated object identification using optical video cameras on construction sites. *Comput.-Aided Civ. Infrastruct. Eng.* **2011**, *26*, 368–380. [CrossRef]
18. Luo, X.; Li, H.; Cao, D.; Dai, F.; Seo, J.; Lee, S. Recognizing diverse construction activities in site images via relevance networks of construction-related objects detected by convolutional neural networks. *J. Comput. Civ. Eng.* **2018**, *32*, 04018012. [CrossRef]

19. Son, H.; Kim, C.; Hwang, N.; Kim, C.; Kang, Y. Classification of major construction materials in construction environments using ensemble classifiers. *Adv. Eng. Inform.* **2014**, *28*, 1–10. [CrossRef]
20. Luo, X.; Li, H.; Yu, Y.; Zhou, C.; Cao, D. Combining deep features and activity context to improve recognition of activities of workers in groups. *Comput.-Aided Civ. Infrastruct. Eng.* **2020**, *35*, 965–978. [CrossRef]
21. Gong, J.; Caldas, C.H.; Gordon, C. Learning and classifying actions of construction workers and equipment using Bag-of-Video-Feature-Words and Bayesian network models. *Adv. Eng. Inform.* **2011**, *25*, 771–782. [CrossRef]
22. Kim, J.; Chi, S. Action recognition of earthmoving excavators based on sequential pattern analysis of visual features and operation cycles. *Autom. Constr.* **2019**, *104*, 255–264. [CrossRef]
23. Kim, H.; Bang, S.; Jeong, H.; Ham, Y.; Kim, H. Analyzing context and productivity of tunnel earthmoving processes using imaging and simulation. *Autom. Constr.* **2018**, *92*, 188–198. [CrossRef]
24. Shim, S.B.; Choi, S.I. Development on identification algorithm of risk situation around construction vehicle using YOLO-v3. *J. Korea Acad.-Ind. Coop. Soc.* **2019**, *20*, 622–629.
25. Kim, D.; Liu, M.; Lee, S.; Kamat, V.R. Remote proximity monitoring between mobile construction resources using camera-mounted UAVs. *Autom. Constr.* **2019**, *99*, 168–182. [CrossRef]
26. Lee, O.S.; Mun, T.U.; Lee, D.K. Safety equipment wearing detection using YOLO based on deep learning. In Proceedings of the IEEK Conference, Gangneung, Republic of Korea, 22–23 November 2019.
27. Zhou, J.; Huo, L. Computer Vision-Based Detection for Delayed Fracture of Bolts in Steel Bridges. *J. Sens.* **2021**, *2021*, 8325398. [CrossRef]
28. Bahrami, Z.; Zhang, R.; Wang, T.; Liu, Z. An end-to-end framework for shipping container corrosion defect inspection. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 5020814. [CrossRef]
29. Abouelnaga, Y.; Ali, O.S.; Rady, H.; Moustafa, M. Cifar-10: Knn-based ensemble of classifiers. In Proceedings of the International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 15–17 December 2016.
30. Russell, B.C.; Torralba, A.; Murphy, K.P.; Freeman, W.T. LabelMe: A database and web-based tool for image annotation. *Int. J. Comput. Vis.* **2008**, *77*, 157–173. [CrossRef]
31. Sorokin, A.; Forsyth, D. Utility data annotation with amazon mechanical turk. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Anchorage, AK, USA, 23–28 June 2008.
32. Soltani, M.M.; Zhu, Z.; Hammad, A. Automated annotation for visual recognition of construction resources using synthetic images. *Autom. Constr.* **2016**, *62*, 14–23. [CrossRef]
33. Handa, A.; Patraucean, V.; Badrinarayanan, V.; Stent, S.; Cipolla, R. Understanding real world indoor scenes with synthetic data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
34. Tremblay, J.; Prakash, A.; Acuna, D.; Brophy, M.; Jampani, V.; Anil, C.; To, T.; Cameracci, E.; Boochoon, S.; Birchfield, S. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
35. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
36. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
37. Fang, W.; Ding, L.; Luo, H.; Love, P.E. Falls from heights: A computer vision-based approach for safety harness detection. *Autom. Constr.* **2018**, *91*, 53–61. [CrossRef]
38. Zhou, X.; Wang, D.; Krähenbühl, P. Objects as points. *arXiv* **2019**, arXiv:1904.07850.
39. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV: 13th European Conference, Zurich, Switzerland, 6–12 September 2014.
40. Rajan, A.J.; Jayakrishna, K.; Vignesh, T.; Chandradass, J.; Kannan, T.T.M. Development of computer vision for inspection of bolt using convolutional neural network. *Mater. Today Proc.* **2021**, *45*, 6931–6935. [CrossRef]
41. Day, O.; Khoshgoftaar, T.M. A survey on heterogeneous transfer learning. *J. Big Data* **2017**, *4*, 29. [CrossRef]