Visual inspection and development of an artificial intelligence-based automated assessment of water channel piling sheets according to Dutch standards

Maskam, Richie; Amiri-Simkooei, Alireza; Nederveen, Sander Van; Visser, Maarten; Fotouhi, Mohammad

DOI
10.1108/SASBE-08-2024-0314

Publication date
2025

Document Version
Final published version

Published in
Smart and Sustainable Built Environment

Citation (APA)
Maskam, R., Amiri-Simkooei, A., Nederveen, S. V., Visser, M., & Fotouhi, M. (2025). Visual inspection and development of an artificial intelligence-based automated assessment of water channel piling sheets according to Dutch standards. *Smart and Sustainable Built Environment*, 1-20. https://doi.org/10.1108/SASBE-08-2024-0314

Smart and Sustainable Built Environment

Visual inspection and development of an artificial intelligence-based automated assessment of water channel piling sheets according to Dutch standards

Received 11 September 2024
Revised 15 February 2025
31 July 2025
19 September 2025
Accepted 19 September 2025

Richie Maskam
Department of Materials, Mechanics, Management and Design, Materials and Environment, Delft University of Technology, Delft, The Netherlands
Alireza Amiri-Simkooei
Department of Control and Operations, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands
Sander Van Nederveen
Department of Materials, Mechanics, Management and Design, Integral Design and Management, Delft University of Technology, Delft, The Netherlands
Maarten Visser
Department of Life Cycle Asset Management, Witteveen + Bos NV, Deventer, The Netherlands, and
Mohammad Fotouhi
Delft University of Technology, Delft, The Netherlands

Abstract

Purpose – This study aims to automate the visual inspection of piling sheets in water channel construction using artificial intelligence (AI). By employing image classification and object detection techniques, the research focuses on extracting and analysing geometric features to enhance the accuracy and efficiency of the inspection process. It also addresses key challenges associated with the unique characteristics of construction materials and the limited variability of available inspection datasets.

Design/methodology/approach – Convolutional neural networks (CNNs) with varying complexities are employed for image classification, across four and six classes, and for object detection of piling sheets in water channel environments. A dataset provided by Witteveen + Bos is preprocessed to generate training sets, and the CNN architectures are optimized for enhanced performance. The accuracy and efficiency of the proposed models are evaluated and compared against traditional manual inspection methods.

Findings – The AI-driven approach significantly reduces processing time, evaluating 40,000 images in just 11.9 h, compared to approximately one month using manual assessment. The 4-class classification model achieves an accuracy of 96%, while the 6-class model attains 72%. The object detection model produces a mean average precision (mAP) of 79%. These results meet the performance standards set by the Dutch company Witteveen + Bos, which demonstrate the effectiveness of AI in automating the inspection of piling sheets.

Originality/value – This study introduces a novel AI-based approach for assessing piling sheets, demonstrating substantial improvements over traditional inspection methods. It introduces a systematic evaluation of various

Smart and Sustainable Built Environment
Emerald Publishing Limited
e-ISSN: 2046-6102
p-ISSN: 2046-6099
DOI 10.1108/SASBE-08-2024-0314

CNN architectures and hyperparameters to optimize the models specifically for piling sheet inspection rather than relying on off-the-shelf solutions. The use of CNNs for both image classification and object detection adheres to relevant Dutch engineering standards. Notably, the reduction in processing time, from one month to around 12 h, represents a major advancement in the efficiency of civil engineering inspections.

Keywords Deep learning, Structural health monitoring, Object detection, Piling sheet

Paper type Research article

1. Introduction

Visual inspection is widely recognized as one of the most cost-effective non-destructive evaluation (NDE) methods utilized across various industries, including aerospace, civil engineering and manufacturing (Kumar and Mahto, 2013; Roberge, 2007a, b; Tabatabaeian *et al.*, 2024). This technique is favoured for its affordability and straightforward implementation, making it a preferred approach for assessing structural integrity and identifying potential defects. Nevertheless, despite these benefits, visual inspection can become a labour-intensive and monotonous task, especially in extensive construction projects where precision and consistency are crucial (Megaw, 1979; See *et al.*, 2017; John and Herman, 2017).

A representative case study is the visual inspection of water channel pilling sheets, which exemplifies such challenges. These piling sheets are crucial components in water channel construction, extending for kilometres and subjected to environmental degradation over time. Regular inspections are therefore crucial to ensure their continued integrity. At Witteveen + Bos, a Dutch engineering firm, the current practice involves manual visual inspection carried out by skilled assessors. The inspection system comprises a boat with multiple cameras and a global positioning system (GPS) unit, which travels along the channel capturing georeferenced images. Although this setup streamlines data collection, the subsequent manual review of tens of thousands of images, for example, 40,000 images within a month, remains a time-consuming and labour-intensive task.

According to the Witteveen + Bos assessment team, manual image reviews present several challenges: they are costly due to the extensive time required, not reporting all problems due to unawareness or hesitance of the inspector, fatigue due to repetitiveness and long periods, and subject to inconsistency between individual inspectors. To address these limitations, this study proposes the use of convolutional neural networks (CNNs) to automatically classify acquired images of damaged and undamaged pilling sheets. The objective is to identify which components of the CNN architecture should be optimized to obtain the most accurate and reliable inspection outcomes, while also considering factors such as training dataset requirements and computational efficiency. This approach aims to develop a robust model and dataset pipeline from raw image data, ultimately to automate visual inspection for planning and predictive maintenance.

1.1 Literature review

The automation of visual inspections in construction has gained considerable attention, particularly in the assessment of heritage and infrastructure contexts, through the application of artificial intelligence (AI) and advanced imaging techniques. A key area within AI is deep learning, which leverages deep artificial neural networks to model complex patterns in data (Amiri-Simkooei *et al.*, 2024). Its recent success can be attributed to several factors, including the increased computational power of graphical processing units (GPUs), the abundance of available data, and advances in optimization algorithms. Deep learning has many applications in the domain of computer vision and image processing, where three fundamental techniques play a significant role (Chollet, 2021):

(1) Image classification: adding a single or multiple labels to an image.

(2) Image segmentation: grouping the pixels of an image into specific classes.

(3) Object detection: identifying and localizing objects within an image using bounding boxes.

In deep learning, different architectures can be employed for classification tasks, with the CNNs being the most widely used due to their effectiveness in image-based applications (Khallaf and Khallaf, 2021). For example, an AI-based automatic visual inspection system for built heritage was developed by Mansuri and Patel (2022). They utilized object detection techniques like Faster R-CNN to enhance inspection accuracy and efficiency. Similarly, a digitally enhanced framework for the visual inspection of masonry bridges was proposed by Talebi *et al*. (2022). They integrated non-destructive testing (NDT) technologies to improve safety and reduce human error. Furthermore, Mohy *et al*. (2024) demonstrated the application of deep learning and computer vision for safety management on construction sites, emphasizing the role of AI in identifying hazards and ensuring safety compliance.

Object detection using CNNs offers substantial benefits for inspectors performing visual assessments across various domains, including wind turbines, buildings, and aircraft. This technology focuses on accurately localizing objects within images. Several CNN architectures have been developed for image classification, including AlexNet (Krizhevsky *et al*., 2012), FuseNet (Rahimian *et al*., 2020), ZF Net (Zeiler and Fergus, 2014) and ResNet (He *et al*., 2016). For example, Fotouhi *et al*. (2021) utilized AlexNet and ResNet to detect damage in composite materials using visual imagery. This approach achieved over 87% precision in identifying embedded defects without the need for human intervention.

In the construction industry, CNNs have shown promising advancements in applications such as safety, progress monitoring, productivity tracking, and quality inspection (Paneru and Jeelani, 2021). For example, Yin *et al*. (2020) demonstrated the effectiveness of CNNs for automated defect detection in sewer pipes, leading to enhanced quality control processes. Despite these advances, the adoption of CNN models in construction remains limited compared to other industries (Abioye *et al*., 2021). Several challenges hinder broader implementation, including the lack of robust guidelines and the limited availability of high-quality datasets, as highlighted by Paneru and Jeelani (2021) and other researchers (Bilal and Oyedele, 2020).

In the Netherlands, NEN-2767 is a national standard for life-cycle asset management, designed to assess and document the condition of assets in an objective and systematic manner (NEN.nl, 2019). This assessment is based on a scale of intensity, which assigns condition scores to various asset components. Witteveen + Bos utilizes this standard through Digigids, a digital tool that incorporates NEN-2767 to monitor the condition of piling sheets. The system decomposes assets into smaller components and categorizes their conditions into four classes: good, acceptable, moderate and bad (Digigids.hetwaterschapshuis.nl, 2022). These classifications, established in consultation with Witteveen + Bos experts, serve as the basis for image classification tasks in this research.

A major challenge in developing autonomous inspection systems for piling sheets is the lack of publicly available and domain-specific image datasets. Although Witteveen + Bos has standardized visual inspection protocols, there remains a significant gap in datasets tailored for object detection models. Moreover, the existing literature reflects limited exploration of automated inspection methods specific to piling sheets. Most related studies focus on similar applications such as corrosion detection in industrial pipelines. For example, the inspection of localized corrosion on industrial pipes using deep learning analysis stands as a comparable case (Bastian *et al*., 2019). The suggested deep learning method efficiently replaces manual inspections and other non-vision-based non-destructive evaluation methods for pipeline corrosion. In a related study, Simonyan and Zisserman (2014) trained a CNN to classify four distinct categories of corrosion, each representing different levels of severity. Building upon this foundation, Wang *et al*. (2022) pursued a similar CNN approach, though with a revised corrosion grading system that addresses uniform corrosion patterns. While these studies employed custom CNN architectures, alternative models such as ResNet (He *et al*., 2015) and Inception (Szegedy *et al*., 2014) offer promising avenues for exploration due to their superior performance in general computer vision tasks. In another study, Forkan *et al*. (2022) employed the Masked R-CNN architecture to automate corrosion detection in telecommunication

towers. While methodologies for distance estimation from images, exemplified by "You Only Look Once" (YOLO) (Qiao and Zulkernine, 2020), have been explored, yet there is currently no systematic investigation into extracting condition-related data from images of piling sheets.

1.2 Research gap and contribution of this work

The current literature reveals significant gaps, including a lack of real-life datasets, clearly defined classification guidelines and the use of CNNs with varying complexities. This study represents the first effort to automate the inspection of piling sheets using CNN models with varying degrees of complexity and computational time to find optimal classification solutions. The application of CNN to the piling sheets inspection in this work is twofold:

First, to ensure the effectiveness of our CNN model, we conduct a systematic evaluation of multiple architectures and tuning strategies. Rather than relying on predefined structures, various CNN configurations are tested, which include adjusting the number of layers, kernel sizes, activation functions, and regularization techniques. This iterative process allows us to identify optimal architecture tailored to our specific engineering application. In addition, we perform a hyperparameter tuning study, varying key parameters such as learning rate, batch size, number of filters and optimization algorithms. Each configuration is assessed using performance metrics such as accuracy, loss convergence, and generalization capability on unseen data. The results from this parametric study not only facilitate the selection of an effective CNN model but also provide insights into the influence of different hyperparameters on model performance. This structured methodology aligns with best practices in engineering applications of machine learning, where models are increasingly used to analyse complex patterns and extract meaningful insights.

Second, the study leverages real-world data collected during field inspections, with image classification labels aligned with the official inspection criteria of Witteveen + Bos. To automate the inspection process, both the surface and underwater structures of the port were scanned with a multi-sensor system (MSS). The resulting point clouds were autonomously classified into damaged and undamaged areas. Following the approach proposed by Duan *et al.* (2022), the dataset is categorized into general and domain-specific image datasets. The CNN-based image classification and object detection techniques are subsequently applied to extract critical geometric information from the piling sheets. By tuning model parameters to optimize the image classification model and using object detection to retrieve dimensional information, the CNN models are effectively customized for the automated inspection of piling sheets in water channel environments.

2. Background and methodology

2.1 Methodology

Without delving into details, this project draws inspiration from three influential CNN architectures:

(1) AlexNet, pioneering the use of stacked convolutional layers combined with max-pooling operations, significantly improves performance in image classification tasks (Krizhevsky *et al.*, 2012).

(2) ResNet, introducing residual connections to mitigate the vanishing gradient problem, enables the training of much deeper networks (He *et al.*, 2016).

(3) GoogLeNet, which proposed the inception module, is a network-in-network structure that processes information through parallel convolutional paths of varying kernel sizes, which enhance both efficiency and accuracy (Szegedy *et al.*, 2014).

There are several metrics available to investigate the performance of the model, including accuracy, precision, loss, recall and F1 score. The accuracy of a model is determined by

comparing the number of correct predictions (both positives and negatives) with the total number of predictions made. It is calculated using the following formula:

$$Accuracy = \frac{TP + FN}{TP + TN + FP + FN} \tag{1}$$

where *TP* is the true positive, *FP* is the false positive, *TN* is the true negative, and *FN* is the false negative. The loss value is used to measure the disparity between the target and the model's prediction (Chollet, 2021). When dealing with data where the output represents probabilities of multiple classes, the widely used loss function is Shannon entropy. In scenarios with more than two classes (i.e. one-hot encoded true label, which holds for this study), categorical Shannon entropy is recommended. The cross-entropy loss is defined as:

$$Loss = -\sum_{i=1}^{c} y_i \log \widehat{y}_i \tag{2}$$

where y_i is the target value for the class *i*, $\widehat{y}_i$ is the predicted probability for class *i*, and c is the number of classes.

Object detection involves identifying and localizing objects within an image using bounding boxes. Two primary types of object detectors are commonly: two-stage detectors and single-shot detectors. Two-stage detectors first divide the image into smaller regions and apply a classification algorithm to label each proposed region. An example of this approach is the Fast Region-based CNN (FR-CNN), which refines both object localization and classification in a sequential process (Girshick, 2015). In contrast, single-shot detectors aim to directly estimate the bounding boxes of objects in a single forward pass through the network, which makes them significantly faster and more suitable for real-time applications. A prominent example of this category is the You Only Look Once (YOLO) algorithm, which directly predicts bounding boxes and class probabilities from the entire image in one evaluation.

Several evaluation metrics have been considered in object detection tasks. In this study, precision and recall metrics are employed to assess model performance. Precision measures the proportion of correctly identified positive instances among all predicted positives, which reflects the accuracy of the model's positive predictions. On the other hand, recall quantifies the proportion of actual positive instances that are correctly identified by the model, which indicates its effectiveness in capturing relevant objects. Both metrics are necessary to assess the model's performance in correctly identifying objects and minimizing false positives and false negatives. These metrics are respectively expressed as

$$Precision = \frac{TP}{TP + FP}, Recall = \frac{TN}{TP + FN} \tag{3}$$

The F1score is a single metric that captures the trade-off between precision and recall, and it offers a comprehensive evaluation of the model's performance. By combining both precision and recall into a harmonic mean, it provides a balanced measure of the model's effectiveness in object detection tasks. F1score is defined as

$$F_{1score} = 2\frac{Precision \times Recall}{Precision + Recall} \tag{4}$$

The mean Average Precision (mAP) is another key metric that provides an overall assessment of the model's performance across all detected classes. It is computed by first calculating the average precision (AP) for each individual class (APi) and then taking the mean of these

values. This metric captures both precision and recall over multiple thresholds and is formally expressed as

$$mAP = \frac{1}{n}\sum_{i=1}^{N} AP_i \tag{5}$$

which offers insights into the general performance of the model. The intersection over union (IoU) is another important measure in object detection, which estimates the distance from the bounding boxes. It is computed as the ratio of the area of overlap (AoO) and the area of union (AoU), expressed as follows:

$$IoU = \frac{AoO}{AoU} \tag{6}$$

The above metrics will be presented to assess the performance of the object detection results.

2.2 Life-cycle asset management

Life-cycle asset management refers to the practice of maximizing the value and performance of an asset through information about its conditions and remaining life span (Roberge, 2007a, b). This approach is built upon two key components: the total cost of the asset over its service life-cycle, and the assessment of its physical condition. By integrating these two components, life-cycle asset management supports informed decision-making regarding maintenance, modernization and replacement strategies. Information about an asset's condition can be gathered using various methods, typically categorized as either destructive or non-destructive methods. Destructive methods require involve removing the asset from service to perform tests, whereas non-destructive methods allow for *in situ* assessment without impairing the asset's functionality (Roberge, 2007a, b). Among non-destructive evaluation techniques, visual inspection is one of the most widely used due to its simplicity and low cost. However, this method heavily relies on the expertise and judgment of the inspector, who must be able to detect critical defects and identify signs of impending failure. A key limitation of visual inspection is that it can be labour-intensive and repetitive, particularly in large-scale infrastructure projects.

Witteveen + Bos has defined specific requirements for applying image classification and object detection techniques in the inspection of pilling sheets. These include utilizing Digigids-based classes, namely, good, acceptable, moderate and bad, as shown in Figure 1, and

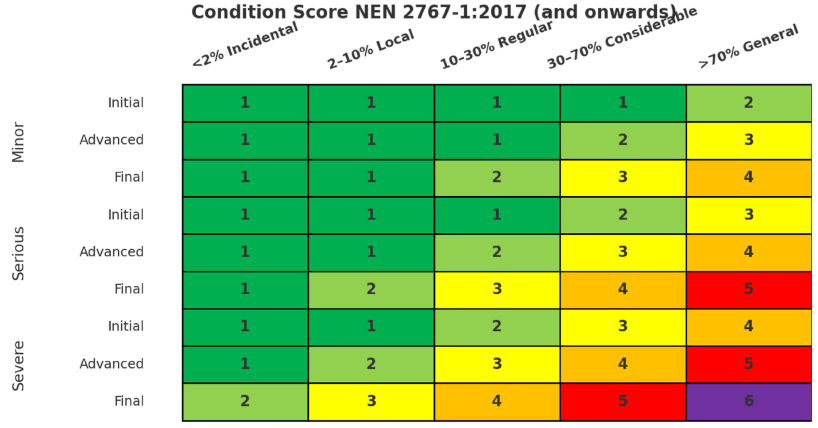

Figure 1. The scoring of a component within the asset (Nen.nl, 2019). Source: Authors' own creation/work

achieving performance metrics consistent with benchmarks from related studies, where classification accuracy of around 90% is commonly reported (Borges Oliveira *et al.*, 2021). Additionally, Witteveen + Bos aimed to extract geometric information from piling sheet images through object detection and post-processing techniques. The key objectives include estimating the height above the waterline and identifying the type of piling sheet. To support the above-mentioned tasks, object detection algorithms were implemented and further refined through post-processing methods. The system was configured to ensure reliable estimation of height and accurate classification of piling sheet types. The importance of mAP and IoU as standard evaluation metrics for object detection was also recognized. Although reported mAP and IoU values vary significantly across the literature, the target was to achieve a minimum performance threshold of 70%, which is in line with accepted standards in comparable applications.

Several general requirements were also determined, covering aspects such as the development, environment, programming language, and data storage. The programming language used was Python with the libraries such as Keras (Chollet, 2015), OpenCV (Bradski, 2000), Sklearn (Pedregosa *et al.*, 2011) and Pandas (Pandas Development Team, 2020. These libraries were chosen for their extensive documentation, robust functionality, and active user communities, which facilitated rapid development and troubleshooting. A cloud-based computing environment was used for model training, along with cloud storage to manage the training datasets. Although this setup differed from the internal systems currently used at Witteveen + Bos, it offered a cost-effective and scalable solution for training deep learning models.

2.3 Data sets

The dataset for image classification was developed in close collaboration with the experts from Witteveen + Bos. The company is responsible for monitoring the "Hoofdvaarweg Lemmer-Delfzijl" (HLD) waterway, as shown in Figure 2. As part of their comprehensive inspection program covering different structural elements such as bridges, piling sheets and aqueducts, a boat equipped with high-resolution imaging sensors was deployed along the waterway to capture images and create a point cloud. These images were manually reviewed according to

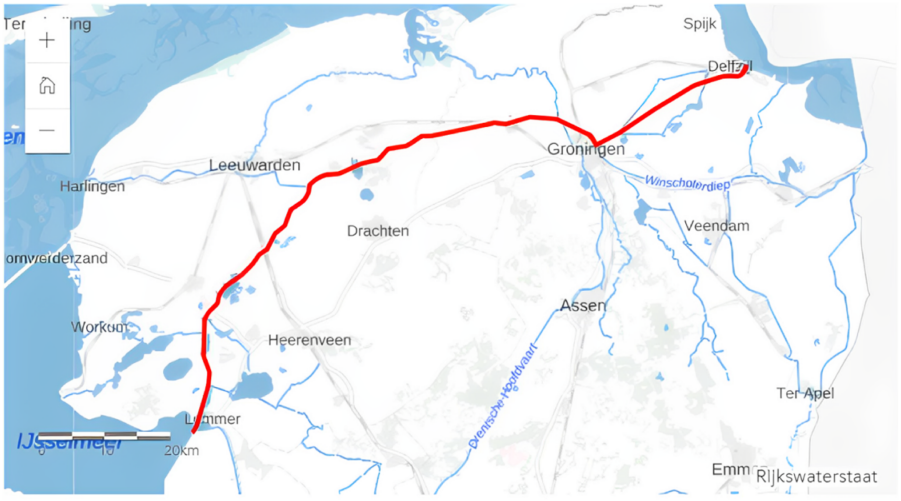

Figure 2. The Hoofdvaarweg Lemmer–Delfzijl (HLD) route, a major inland shipping route in the Netherlands. Source: Rijkswaterstaat.nl. Authors' own creation/work

Digigids guidelines and georeferenced using a geographic information system (GIS). To optimize storage space, the images were resized before being uploaded to a cloud-based storage platform.

The initial review of the dataset resulted in the identification of three broad classes: Grass, Rock and Metal. Although the waterway also contains piling sheets made of wood and concrete, the focus of this study was on metal piling sheets, which are particularly susceptible to corrosion. The metal images were initially classified into four classes including "metal good", "metal acceptable", "metal moderate", and "metal bad". Figures 3 and 4 show sample images of the classified groups for the metals. These classifications were based on expert assessment and aligned with Digigids evaluation criteria. It was however noted that the dataset was imbalanced, with a limited number of samples in the "metal bad" class. To maintain class balance and minimize potential negative impacts on model performance, the classes were consolidated, inspired by the guidance provided by Buda *et al*. (2018), who systematically analysed the effects of class imbalance in CNNs. Their study recommends mitigating imbalance through class consolidation or resampling to ensure more effective learning and to reduce performance bias toward overrepresented classes. This indicates that "metal bad" and "metal moderate" were merged into a single class called "metal bad" whereas the classes "metal good" and "metal acceptable" were combined into a class named "metal good". This adjustment, as shown in Figure 5, resulted in a more balanced distribution of images across classes and hence improved the robustness and generalizability of the image classification model.

Table 1 provides the total number of images used for the classification task. To ensure model training and validation, a standard 70 and 30% split was employed, which is commonly used in image classification tasks. Using this setup, 70% of the dataset is allocated to train the model and the remaining 30% is reserved to validate the model's performance. The dataset was shuffled and split three times to achieve the desired cross-validation setup. This method provides deeper insights into the dataset's characteristics. Notably, substantial variations in performance across different folds may indicate either a highly heterogeneous dataset or an unstable training algorithm.

The object detection dataset was created using LabelImg (Tzutalin, 2022), a widely used annotation tool that enables the creation of bounding boxes for object detection tasks. Each image in the dataset requires a corresponding annotation file that defines the coordinates and labels of all detected objects. LabelImg provides a flexible tool for accomplishing this task, supporting various annotation formats. For this research, the object detection task focused on identifying two specific types of objects, selected based on their relevance to piling sheet inspection:

Figure 3. Two sample images of classes good (left) and acceptable (right) for the six-class dataset. Source: Authors' own creation/work

Figure 4. Two sample images of classes moderate (left) and bad (right) for six-class dataset. Source: Authors' own creation/work

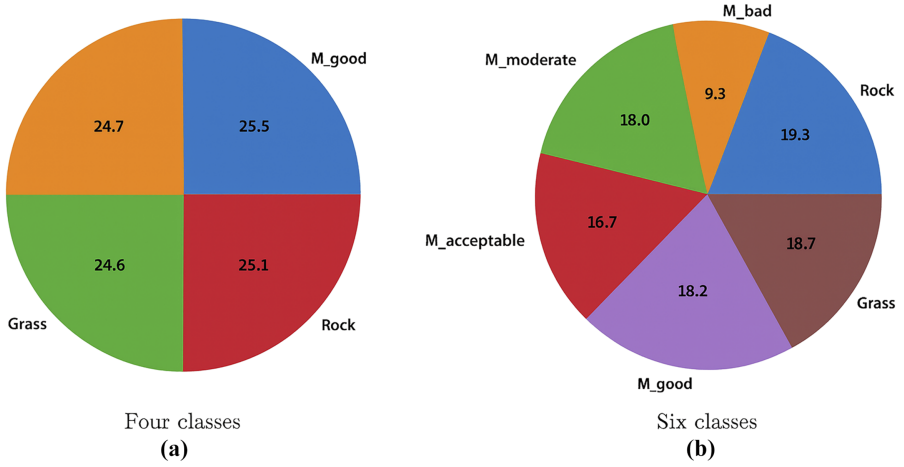

Figure 5. Pie chart to check the balance (%) of datasets classification for the (a) four classes and (b) six classes. Source: Authors' own creation/work

Table 1. Training and validation set for four and six classes

Classes	Total images	Selection	Number of images	Percentage (%)
4	2,534	Training	1,774	70
4	2,534	Validation	760	30
6	1,801	Training	1,301	70
6	1,801	Validation	540	30

Source(s): Authors' own creation/work

(1) Bumps: The distance between bumps can be used to decide on the dimension and scale of the object.

(2) Anchors: They serve as a reference object to convert the pixel distance to the actual distance.

Around 250 images were manually annotated for training purposes, with an additional 100 images reserved for testing the performance of the object detection models. The annotation process involved manually drawing bounding boxes around the target objects in each image, which was a labour-intensive task that required precision and consistency. The process was further complicated by vegetation obscuring some objects, making accurate annotation particularly challenging. The models were tested on a separate set of 100 images, distinct from those used in the training of both the classification and object detection models. Unlike the common practice of splitting data into training, validation, and testing sets using percentage ratios, folder-based approach was adopted. This strategy was chosen to better reflect real-world application by simulating the processing of folders from the Witteveen + Bos servers. The specific filenames of the images were necessary to couple them with additional data, such as geographic coordinates. The models were designed to automatically analyse all files and store the results in a structured data frame. The complete datasets used in this study have been made publicly available on 4TU.ResearchData (Maskam *et al*., 2022) and Kaggle, to support further research and replication.

2.4 Model architecture

CNN-based classification models with three block variations were tested, and YOLOv4 was used as the backbone for object detection. The image classification model was developed through iterative experimentation with architectural structure, filter sizes, and the number of layers. The architecture was initially designed with a block containing two convolution layers followed by a maxpooling layer. Subsequently, a second block was added, incorporating a residual connection to enhance learning stability. Finally, a third block was created by adding a convolution layer within the residual connection, which further enriches the model's capacity for feature extraction. The different blocks and structures are illustrated in Figure 6. Throughout each run, slight modifications were made, including changes to the type and number of blocks, the filter sizes in the convolution layers and the use of an augmentation layer. An overview of the different architectural variations tested is shown in Figure 7. It should be noted that the execution of this task in the cloud environment was affected by certain limitations. Constraints on memory resources and GPU usage time imposed by the platform influenced the scope and complexity of model development. As a result, careful resource optimization and management were essential during the model training and evaluation phases.

For the object detection task, an off-the-shelf algorithm called YOLOv4 was employed. This choice was made to quickly obtain a proof of concept on the applicability of the method

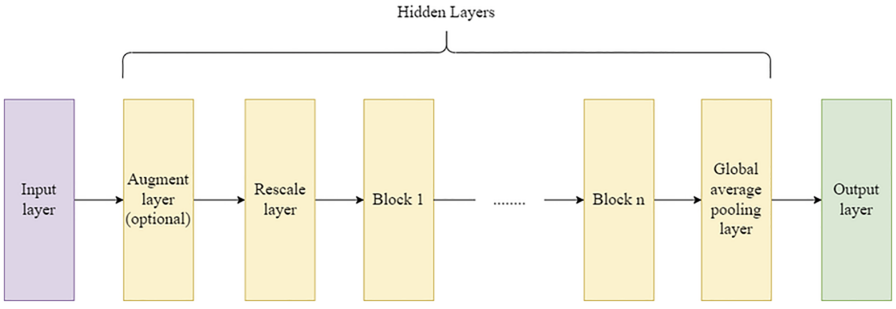

Figure 6. Overview of the CNN architecture, showing the sequential arrangement of convolution, pooling, and residual blocks used in this study. Source: Authors' own creation/work

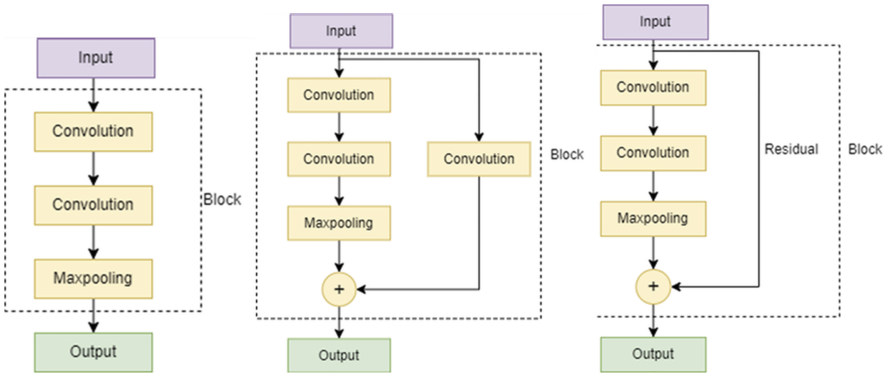

Figure 7. Variations of the CNN block architecture tested in this study, showing different configurations of convolutional layers, residual connections, and filter sizes building on the general framework in Figure 6. Source: Authors' own creation/work

(Tulbure *et al*., 2022). To configure the algorithm, the steps provided by Alexey Bochkovskiy were used to make the necessary adjustments and settings (Bochkovskiy, 2017). Figure 8 illustrates the flowchart of the YOLO algorithm. The YOLO algorithm outputs a vector containing the class label and the coordinates of the bounding boxes. The coordinates were used to extract important information such as the heights and distances. The heights were calculated by taking the average height of the boxes, while the distances between the bumps were obtained by measuring the distance between the coordinates of the first two boxes. To ensure accurate measurements, predefined thresholds were introduced and if the calculated distance was too small/large, it would not be used. To establish the relationship between the real-world distance and the pixel distance, a reference object was used. By calculating the ratio between the actual distance and the pixel distance of the reference object, the algorithm could estimate the height and distance between the bumps. This enabled precise estimates by accounting for the scaling factor between physical dimensions and the corresponding pixel distances captured by the YOLO algorithm. Figure 9 presents an example of the results obtained from the YOLO algorithm for object detection. In this case, all relevant objects were correctly identified, with an estimated height of 56.1 cm and a distance of 85.5 cm between the detected bumps. According to experts at Witteveen + Bos, these values are considered plausible, indicating that object detection, when combined with a known reference, can provide reliable geometrical data for asset evaluation.

3. Implementation and results

The datasets were compiled with the support of Witteveen + Bos experts and annotation software. The optimal model architecture was achieved through iterative experimentation with different structures, depths, and filter sizes. One of the best-performing models was identified; it underwent further evaluation to assess its effectiveness. For the image classification model,

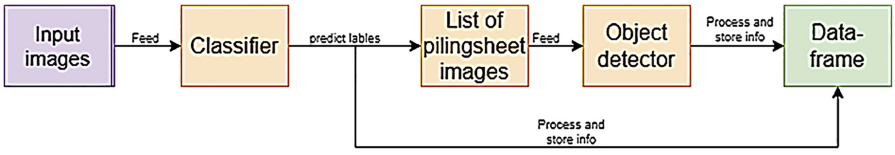

Figure 8. Flowchart of the YOLOv4 algorithm. Source: Authors' own creation/work

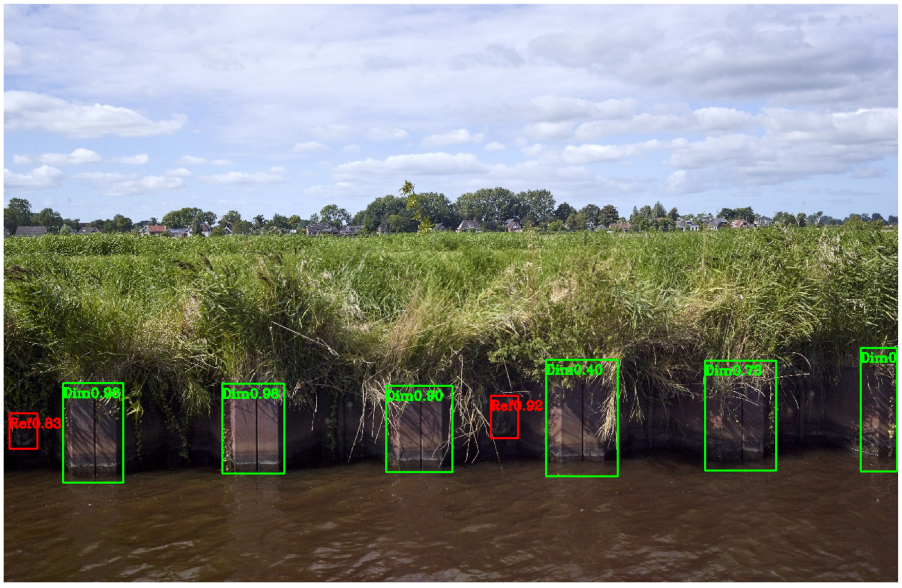

Figure 9. Example output of the YOLOv4 object detection model, showing detected bumps (Dim, used for dimensional estimation) and anchors (Ref, used as reference objects for scaling). Source: Authors' own creation/work

cross-validation and confusion matrices were employed. Cross-validation involved training the model on different shuffled subsets of the original data to optimize its weights, while confusion matrices offered insights into the model's classification accuracy and served as a tool for improvement. For object detection, performance was evaluated using metrics such as precision, recall and the Aps, calculated via the Darknet framework. These metrics were used to assess the model's accuracy in both detecting and localizing objects.

In addition to the standard performance evaluation metrics, a preliminary time-cost analysis was conducted to investigate the computational efficiency of the models. Expert reviews were also collected to obtain insights and recommendations specifically aligned with the practical needs of Witteveen + Bos. To evaluate the performance of the four-class models, various configurations were tested and compared. Table 2 summarizes these configurations. Among them, the best-performing model featured a residual connection (ResCon type) architecture with five blocks, no augmentation and a consistent filter size of 32 across all layers. Table 3 shows the different configurations used for the six-class model. The best-performing model used a ResCon structure with six blocks, an augmentation layer, and varying filter sizes per block. In addition, the Inception model also provided satisfactory results, while certain variants of the VGG16 architecture demonstrated reasonable performance.

Table 4 presents the results for the four-class model when trained and validated on different shuffled data. In this case, performance remained consistent across different folds, with no significant variation observed. However, a slight difference between training-validation and model test (MT) accuracy was observed. Table 5 presents the performance of the six-class model under the same conditions, highlighting its effectiveness across multiple datasets, albeit with some discrepancies in MT. Compared to the four-class model, this discrepancy became more pronounced in the six-class model, where significant differences are clear between the accuracy scores of the training and validation phases compared to the MT accuracy. While the train and validation accuracy averaged around 95%, the MT accuracy ranged from 50% to

Table 2. Models with different structures, blocks, augmentation and filters for the four classes

Type	Blocks	Trainable parameters	Augment layer	Fillers	Train accuracy (%)	Validation accuracy (%)	MT accuracy (%)
VGG16	5	0.65	No	32-64-128-256-256	25	25	22
VGG16	5	0.02	No	32-32-32-32-32	93	92	84
VGG16	5	0.05	Yes	32-32-32-32-32	25	25	13
VGG16	4	0.32	Yes	32-64-128-256	25	25	13
ResCon	4	0.33	No	32-64-128-256	98	98	87
ResCon	4	0.33	Yes	32-64-128-256	97	97	83
ResCon	5	0.02	No	32-32-32-32-32	98.3	94.4	96
ResCon	5	0.66	No	32-64-128-256-256	98.6	97.7	85
ResCon	5	0.66	Yes	32-64-128-256-256	96.8	95.5	67
ResCon	5	0.02	Yes	32-32-32-32-32	94.4	92.7	9
Inception	5	0.03	No	32-32-32-32-32	96.8	95.7	91
Inception	4	0.43	Yes	32-64-128-256	92.9	87	80
Inception	5	0.86	No	32-64-128-256-256	27.17	25.26	22
Inception	5	0.86	Yes	32-64-128-256-256	88.7	82.5	80

Source(s): Authors' own creation/work

Table 3. Models with different structures, blocks, augmentation and filters for the six classes

Type	Blocks	Trainable parameters	Augment layer	Fillers	Train accuracy (%)	Validation accuracy (%)	MT accuracy (%)
VGG16	5	0.02	No	32-32-32-32-32	90	90	57
VGG16	6	0.03	No	32-32-32-32-32-32	98.8	87.5	52
ResCon	6	1	Yes	32-64-128-3x256	97.5	95.3	68
ResCon	6	1	No	32-64-128-3x256	98	95	65
ResCon	6	0.03	No	32-32-32-32-32-32	99.99	95	65
Inception	5	0.03	No	32-32-32-32-32	95	93	65
Inception	6	0.04	No	32-32-32-32-32-32	99	94	66
Inception	6	0.04	Yes	32-32-32-32-32-32	92.4	91.8	66

Source(s): Authors' own creation/work

Table 4. Best-performing four-class model with different shuffled training and validation data

Seed	Train accuracy (%)	Validation accuracy (%)	MT accuracy (%)
0	98.3	94.4	96
1	97.3	95	92
2	96.3	95.6	85

Source(s): Authors' own creation/work

70%. This clear contrast could stem from several factors, for example, insufficient training data or the presence of overfitting, a common problem in machine learning models. Furthermore, a significant difference of 20% was observed in the accuracy of different folds in the model test, indicating that certain portions of the dataset may possess better quality or suitability for training the model.

Table 5. Best performing six-class model with different shuffled training and validation data

Seed	Train accuracy (%)	Validation accuracy (%)	MT accuracy (%)
0	97.5	95.3	68
1	97.5	95.3	50
2	97.5	95.3	72

Source(s): Authors' own creation/work

4. Discussion of results

This section discusses the findings from the models. It includes analyses of confusion matrices, object detection metrics, and the time-cost implications. Expert reviews were also conducted. These reviews provided practical insights and helped contextualize the results.

4.1 Confusion matrix

The confusion matrix is used to compare true labels with predicted labels. Figures 10 and 11 display the confusion matrices for material type and rust grade classification, respectively. The confusion matrices in Figures 10 and 11 are based on an independent subset of 100 test images, separated using a folder-based approach, to better reflect the real-world application of processing image batches from Witteveen + Bos servers. The four-class model achieved higher overall accuracy, although some misclassifications occurred within the metal classes. In contrast, the six-class model showed weaker performance, with a higher incidence of false positives, particularly in the rust grade classes. This difficulty in accurately predicting rust grades is likely due to the limited dataset and the subtle differences between the Digigids-defined classes.

4.2 Object detection metrics

The object detection performance was evaluated using metrics provided by Darknet, the framework used to train the YOLOv4 model. These metrics include the mAP and IoU, which are standard indicators to assess object detection quality.

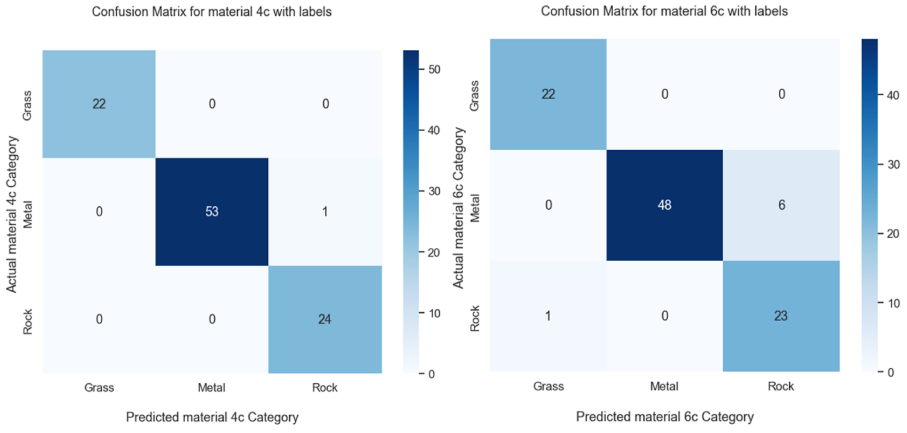

Figure 10. Confusion matrix of the material type for 4 and 6 classes. Source: Authors' own creation/work

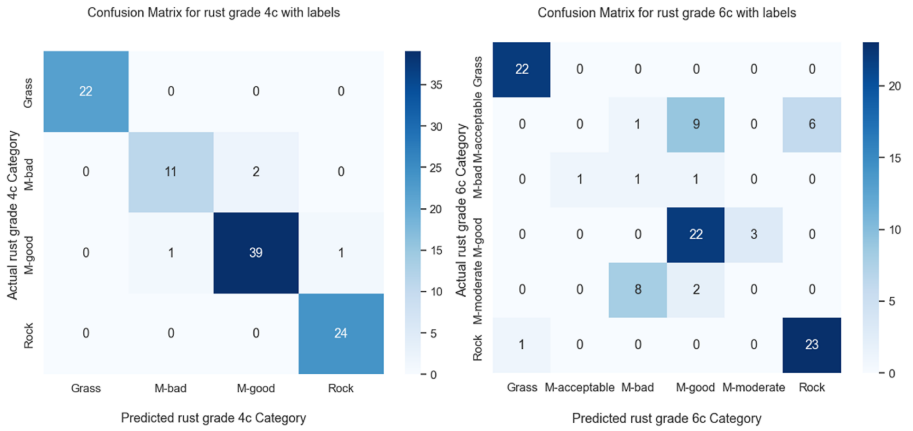

Figure 11. Confusion matrix of rust grade for the 4 and 6 classes. Source: Authors' own creation/work

As shown in Table 6, the model achieved an average mAP of approximately 79.2% at a 50% confidence threshold, which indicates that the majority of predicted bounding boxes were correctly aligned with ground truth annotations. The corresponding IoU was around 66.4%, reflecting a reasonable overlap between predicted and actual object locations. These results suggest that the YOLOv4 model performed effectively to detect the target objects within the given dataset.

This project aligns with broader deep learning initiatives focused on image-based analysis, particularly those involving large image datasets. Despite working with a smaller dataset of around 300 images, the model achieved an mAP exceeding 70%, meeting the performance requirements by Witteveen + Bos. A distinguishing feature of this study is that the dataset was collected from real-world field conditions, unlike many previous studies that rely on synthetic datasets. For comparison, Wang *et al*. (2022) developed a four-class corrosion detection model for industrial pipes using over 100,000 images, achieving an mAP of 90.96%. Guo *et al*. (2021) achieved a 90% mAP in detecting rail track components with 1,000 images, while Zhang *et al*. (2022) reached an mAP of 90.7% in aircraft tube fault detection using 3,000 images. In another study, Qiao and Zulkernine (2020) achieved an mAP of 99.16% in vehicle detection using the Microsoft Common Objects in Context (MS COCO) dataset. These comparisons highlight a clear trend: larger and more diverse training datasets tend to yield higher detection accuracy. Nevertheless, the good performance achieved in this study, despite

Table 6. Model with confidence threshold 25%

Model confidence threshold: 25	
Precision	0.84
Recall	0.91
F1-score	0.88
TP	768
FP	144
FN	72
avg IoU	66.40%
MAP@0.50	79.23%

Source(s): Authors' own creation/work

limited training data, demonstrates the model's robustness and the value of a high-quality dataset.

The best-performing configurations from Tables 5 and 6 were used in a cross-validation setup to evaluate the effects of training on different subsets of the data. The MT accuracy results from both Table 4 (four-class model) and Table 5 (six-class model) reveal significant differences. In particular, the six-class model illustrates a discrepancy of 22% across different folds, which may suggest a potential imbalance in the dataset or greater difficulty in distinguishing between the defined classes. In contrast, the four-class model exhibits a smaller variation of 11%, which implies that models trained on fewer and more distinct classes tend to generalize more consistently. These findings highlight the challenges in classification tasks when class boundaries are less clear and when sample sizes are limited.

4.3 Time-cost analysis

A preliminary time-cost analysis was conducted to compare the conventional (current) method with the proposed approach. Currently, assessing 40,000 images requires two experts working for one month. In contrast, the proposed classification and object detection model processes 100 images in approximately 77–107 s (see Table 7). Extrapolating this to 40,000 images, the total computation time would be around 8–12 h. This represents a significant improvement in processing efficiency.

A cost comparison was also conducted by evaluating the time required to train the models against the time spent by employees on manual assessment. Although specific salary information was not obtained for ethical reasons, an estimate was made using the average monthly salary in the Netherlands for professionals in this field, based on a one-month assessment period. Training costs vary depending on the method used. In total, moder training took approximately nine hours. The first training option, i.e. using our cloud environment, is free of charge but comes with resource limitations. For the second option, that is, cloud computing services used in Witteveen + Bos, the costs range from €0.65 to €5.70 per hour. Finally, a physical GPU such as the NVIDIA TESLA T4, which powers our cloud environment, can be purchased for around USD 1670 for on-premise model training.

To evaluate the robustness of the proposed CNN model under varying environmental conditions, a sensitivity analysis was conducted. We focus on the impact of daylight variation on piling sheet inspection and classification results. Since lighting conditions can significantly influence image-based inspection systems, assessing this effect is crucial for real-world applicability. For this analysis, the lighting conditions of 100 images were artificially modified to simulate a transition from daytime to nighttime illumination. The modified images were then tested using the trained CNN model to quantify any performance degradation (so only on the testing data). The results showed a 5% reduction in classification accuracy under altered lighting conditions. This suggests that while the model maintains a reasonable level of robustness, variations in illumination can still affect its performance. This highlights the need for future research involving the collection of real-world experimental data under varying lighting conditions and environmental factors.

Table 7. Expected inference times

	4-class algorithm	6-class algorithm
Classification	21 s	32.3 s
Object-detection	56 s	75 s
Total on 100 images	77 s	107.3 s
Expected on 40 k	8.6 h	11.9 h

Source(s): Authors' own creation/work

4.4 Research potential and future directions

The effectiveness of this research and the applicability of the proposed model were investigated by academic and industrial experts. Their assessment focused on two key aspects: *accessibility* and *feasibility.*

For accessibility, industrial experts found that the algorithm largely met expectations. However, they observed that some measurements estimated using the Euclidean distance between bounding boxes did not fully align with the material specifications provided by the piling sheet fabricator. It was also noted that the classification algorithm could help reduce the subjectivity inherent in the NEN2767 standard. This standard relies on condition scores to estimate an asset's remaining life. This classification could assist less experienced inspectors in making assessments more consistent with those of their peers.

Regarding feasibility, the classification component was considered effective. However, experts suggested that six-class system could be improved and advised caution when interpreting distance estimations. Potential applications beyond the initial scope were also highlighted. This includes asset management tasks such as assessing the condition of roofs, floors, and columns, as well as detecting people and safety equipment during roof renovations. Additionally, the method used to structure the CNN was found beneficial. The ability to observe how changes to layers and filters affected performance provided valuable insights for fine-tuning models in future applications.

These insights highlight the need for further research on improved data sets, such as the images with the potential for depth profile measurement of the piling sheets, and data augmentation techniques or adaptive preprocessing methods to enhance model generalization across different lighting conditions. Since variations in illumination and other environmental conditions can affect classification performance, future research could also focus on generating multiple samples by altering environmental conditions to define the best input data set and its limitations, and to improve the model's resilience and applicability across different real-world settings.

5. Concluding remarks

This paper investigated the application of deep learning-based computer vision techniques for life-cycle asset management of water channel piling sheets, which utilize both image classification and object detection. A four-class and a six-class classification model were developed, with the four-class model demonstrating superior performance and meeting the requirements set by Witteveen + Bos. Additionally, a dataset was created for YOLOv4 to detect dimensional features and calculate distances between bumps. Validation results indicated that an accuracy of 96% was achieved by the four-class classification model, while the object detector reached a mean average precision of 79.23%. Although distance estimations were preliminary, effective detection of features and counting of objects were accomplished, significantly reducing inspection time to over 25 times faster than manual inspection. This approach was shown to be versatile, applicable to various assets, including building elements like roofs, floors, windows and doors in real estate, as well as pavements, bridges, and dikes in civil engineering. The research highlights the importance of a well-developed dataset and appropriate model configurations in developing effective deep learning solutions.

For future research, it is recommended that the dataset be expanded to include images from various angles, locations and times of year to improve model robustness. A comprehensive feasibility analysis should be conducted to evaluate the investment value of the automated solution for Witteveen + Bos's future projects. Additionally, the interoperability of the developed model with other relevant management software should be explored to further integrate and optimize its application. Future research should explore enhanced datasets, adaptive preprocessing and data augmentation to improve model generalization and resilience across varying environmental conditions.

References

Abioye, S.O., Oyedele, L.O., Akanbi, L., Ajayi, J.M., Davila Delgado, M., Bilal, M., Akinade, O.O. and Ahmed, A. (2021), "Artificial intelligence in the construction industry: a review of present status, opportunities and future challenges", *Journal of Building Engineering*, Vol. 44, 103299, doi: 10.1016/j.jobe.2021.103299.

Amiri-Simkooei, A., Tiberius, C. and Lindenbergh, R. (2024), "Deep learning in standard least-squares theory of linear models: perspective, development and vision", *Engineering Applications of Artificial Intelligence*, Vol. 138, 109376, doi: 10.1016/j.engappai.2024.109376.

Bastian, B.T.N., Ranjith, S.K. and Jiji, C. (2019), "Visual inspection and characterization of external corrosion in pipelines using deep neural network", *NDT and E International*, Vol. 107, 102134, doi: 10.1016/j.ndteint.2019.102134.

Bilal, M. and Oyedele, L.O. (2020), "Guidelines for applied machine learning in construction industry—a case of profit margins estimation", *Advanced Engineering Informatics*, Vol. 43, 101013, doi: 10.1016/j.aei.2019.101013.

Bochkovskiy, A. (2017), "Keras", available at: https://github.com/AlexeyAB/darknet (accessed February 2025).

Borges Oliveira, D.A., Ribeiro Pereira, L.G., Bresolin, T., Pontes Ferreira, R.E. and Reboucas Dorea, J.R. (2021), "A review of deep learning algorithms for computer vision systems in livestock", *Livestock Science*, Vol. 253, 104700, doi: 10.1016/j.livsci.2021.104700.

Bradski, G. (2000), "The OpenCV library", Dr. Dobb's Journal of Software Tools, available at: https://bibbase.org/network/publication/bradski-theopencvlibrary-2000 (accessed February 2025).

Buda, M., Maki, A. and Mazurowski, M.A. (2018), "A systematic study of the class imbalance problem in convolutional neural networks", *Neural Networks*, Vol. 106, pp. 249-259, doi: 10.1016/j.neunet.2018.07.011.

Chollet, F. (2015), "Keras", available at: https://github.com/fchollet/keras (accessed February 2025).

Chollet, F. (2021), *Deep Learning with Python*, 2nd ed., Manning, available at: https://www.manning.com/books/deep-learning-with-python-second-edition (accessed February 2025).

Digigids.hetwatershapshuis.nl (2022), "Digigids 2019", available at: https://digigids.hetwaterschapshuis.nl/ (accessed February 2025).

Duan, R., Deng, H., Tian, M., Deng, Y. and Lin, J. (2022), "Soda: a large-scale open site object detection dataset for deep learning in construction", *Automation in Construction*, Vol. 142, 104499, doi: 10.1016/j.autcon.2022.104499.

Forkan, A.R.M., Kang, Y.-B., Jayaraman, P.P., Liao, K., Kaul, R., Morgan, G., Ranjan, R. and Sinha, S. (2022), "Corrdetector: a framework for structural corrosion detection from drone images using ensemble deep learning", *Expert Systems with Applications*, Vol. 193, 116461, doi: 10.1016/j.eswa.2021.116461.

Fotouhi, S., Pashmforoush, F., Bodaghi, M. and Fotouhi, M. (2021), "Autonomous damage recognition in visual inspection of laminated composite structures using deep learning", *Composite Structures*, Vol. 268, 113960, doi: 10.1016/j.compstruct.2021.113960.

Girshick, R. (2015), "Fast R-CNN", doi: 10.48550/ARXIV.1504.08083.

Guo, F., Qian, Y. and Shi, Y. (2021), "Real-time railroad track components inspection based on the improved YOLOv4 framework", *Automation in Construction*, Vol. 125, 103596, doi: 10.1016/j.autcon.2021.103596.

He, K., Zhang, X., Ren, S. and Sun, J. (2015), "Deep residual learning for image recognition", doi: 10.48550/ARXIV.1512.03385.

He, K., Zhang, X., Ren, S. and Sun, J. (2016), "Deep residual learning for image recognition", *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770-778, doi: 10.1109/cvpr.2016.90, available at: https://ieeexplore.ieee.org/document/7780459 (accessed February 2025).

John, M. and Herman, S. (2017), *Project Management for Engineering, Business and Technology*, 5th ed., Routledge, available at: https://www.routledge.com/Project-Management-for-Engineering-

Business-and-Technology/Nicholas-Steyn/p/book/9780367277345?srsltid=AfmBOops3yeUOu39aMyNm3pmEiMi1mzb-_X2YEY2oZfAc5vQmGqEL5Nu (accessed February 2025).

Khallaf, R. and Khallaf, M. (2021), "Classification and analysis of deep learning applications in construction: a systematic literature review", *Automation in Construction*, Vol. 129, 103760, doi: 10.1016/j.autcon.2021.103760.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012), "Imagenet classification with deep convolutional neural networks", *Advances in Neural Information Processing Systems*, Vol. 25, doi: 10.5555/2999134.2999257.

Kumar, S. and Mahto, D.G. (2013), "Recent trends in industrial and other engineering applications of non-destructive testing: a review", *International Journal of Scientific Engineering and Research*, Vol. 4 No. 9, available at: https://ssrn.com/abstract=2770922 (accessed February 2025).

Mansuri, L.E. and Patel, D.A. (2022), "Artificial intelligence-based automatic visual inspection system for built heritage", *Smart and Sustainable Built Environment*, Vol. 11 No. 4, pp. 623-639, doi: 10.1108/sasbe-09-2020-0139.

Maskam, R., Amiri-Simkooei, A.A., van Nederveen, S., Fotouhi, M. and Visser, M. (2022), "Data underlying the student thesis: automated monitoring of corrosion on piling sheets", doi: 10.4121/21387930.v1.

Megaw, E.D. (1979), "Factors affecting visual inspection accuracy", *Applied Ergonomics*, Vol. 10 No. 1, pp. 27-32, doi: 10.1016/0003-6870(79)90006-1.

Mohy, A.A., El-Diraby, T.E. and Ahmed, K. (2024), "Innovations in safety management for construction sites: the role of deep learning and computer vision techniques", *Construction Innovation*, Vol. 24 No. 1, pp. 75-91, doi: 10.1108/ci-04-2023-0062.

Nen.nl (2019), "Nen 2767-1+c1:2019 nl", available at: https://www.nen.nl/en/nen-2767-1-c1-2019-nl-256366 (accessed February 2025).

Pandas Development Team (2020), "Pandas-dev/pandas: pandas", doi: 10.5281/zenodo.3509134.

Paneru, S. and Jeelani, I. (2021), "Computer vision applications in construction: current state, opportunities and challenges", *Automation in Construction*, Vol. 132, 103940, doi: 10.1016/j.autcon.2021.103940.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, É. (2011), "Scikit-learn: machine learning in Python", *Journal of Machine Learning Research*, Vol. 12, pp. 2825-2830, doi: 10.5555/1953048.2078195.

Qiao, D. and Zulkernine, F. (2020), "Vision-based vehicle detection and distance estimation", *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 2836-2842, doi: 10.1109/SSCI47803.2020.9308364.

Rahimian, F.P., Seyedzadeh, S., Oliver, S., Rodriguez, S. and Dawood, N. (2020), "On-demand monitoring of construction projects through a game-like hybrid application of BIM and machine learning", *Automation in Construction*, Vol. 110, 103012, doi: 10.1016/j.autcon.2019.103012.

Roberge, P.R. (2007a), "Life cycle asset management", in *Corrosion Inspection and Monitoring*, John Wiley & Sons, pp. 86-90, available at: https://app.knovel.com/hotlink/khtml/id:kt007WP8B1/corrosion-inspection/life-cycle-asset-management (accessed February 2025).

Roberge, P.R. (2007b), "Visual and enhanced visual inspection", in *Corrosion Inspection and Monitoring*, John Wiley & Sons, pp. 336-337, available at: https://app.knovel.com/hotlink/khtml/id:kt007WPEA2/corrosion-inspection/visual-enhanced-visual (accessed February 2025).

See, J.E., Drury, C.G., Speed, A., Williams, A. and Khalandi, N. (2017), "The role of visual inspection in the 21st century", *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Sage CA, Los Angeles, CA, Vol. 61 No. 1, pp. 262-266, doi: 10.1177/1541931213601548.

Simonyan, K. and Zisserman, A. (2014), "Very deep convolutional networks for large-scale image recognition", doi: 10.48550/ARXIV.1409.1556.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2014), "Going deeper with convolutions", doi: 10.48550/ARXIV.1409.4842.

Tabatabaeian, A., Fotouhi, S. and Fotouhi, M. (2024), "Visual inspection of impact damage in composite materials", in *Non-destructive Testing of Impact Damage in Fiber-Reinforced Polymer Composites*, Woodhead Publishing, pp. 43-67, doi: 10.1016/B978-0-443-14120-1.00002-9.

Talebi, S., Smith, R. and Nguyen, H. (2022), "The development of a digitally enhanced visual inspection framework for masonry bridges in the UK", *Construction Innovation*, Vol. 22 No. 2, pp. 233-248, doi: 10.1108/CI-10-2021-0201.

Tulbure, A.-A., Tulbure, A.-A. and Dulf, E.-H. (2022), "A review on modern defect detection models using DCNNs – deep convolutional neural networks", *Journal of Advanced Research*, Vol. 35, pp. 33-48, doi: 10.1016/j.jare.2021.03.015.

Tzutalin (2022), "LabelImg", available at: https://github.com/tzutalin/labelImg (accessed February 2025).

Wang, Y., Shen, X., Wu, K. and Huang, M. (2022), "Corrosion grade recognition for weathering steel plate based on a convolutional neural network", *Measurement Science and Technology*, Vol. 33 No. 9, 095014, doi: 10.1088/1361-6501/ac7034.

Yin, X., Chen, Y., Bouferguene, A., Zaman, H., Al-Hussein, M. and Kurach, L. (2020), "A deep learning-based framework for an automated defect detection system for sewer pipes", *Automation in Construction*, Vol. 109, 102967, doi: 10.1016/j.autcon.2019.102967.

Zeiler, M.D. and Fergus, R. (2014), "Visualizing and understanding convolutional networks", *Computer Vision–ECCV 2014: 13th European Conference*, Springer International Publishing, pp. 818-833, available at: https://arxiv.org/abs/1311.2901 (accessed February 2025).

Zhang, J., Wei, S., Qi, M. and Wang, P. (2022), "Improved aircraft flared tube defect detection algorithm of YOLOv4 network structure", *Journal of Physics: Conference Series*, Vol. 2252 No. 1, 012050, doi: 10.1088/1742-6596/2252/1/012050.

Corresponding author
Mohammad Fotouhi can be contacted at: m.fotouhi-1@tudelft.nl