


Please cite the Published Version

Nguyen Thi Phuong, Thanh, Cho, Gyu Sung and Chatterjee, Indranath  (2025) Automating container damage detection with the YOLO-NAS deep learning model. Science Progress, 108 (1). ISSN 2047-7163

DOI: <https://doi.org/10.1177/00368504251314084>

Publisher: SAGE Publications

Version: Published Version

Downloaded from: <https://e-space.mmu.ac.uk/638253/>

Usage rights:  [Creative Commons: Attribution-Noncommercial 4.0](https://creativecommons.org/licenses/by-nc/4.0/)

Additional Information: This is an open access article which first appeared in Science Progress, published by SAGE Publications

Enquiries:

If you have questions about this document, contact openresearch@mmu.ac.uk. Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)

Automating container damage detection with the YOLO-NAS deep learning model

Science Progress

2025, Vol. 108(1) 1–21

© The Author(s) 2025

Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/00368504251314084

journals.sagepub.com/home/sci

Thanh Nguyen Thi Phuong¹,
Gyu Sung Cho¹
and Indranath Chatterjee^{2,3,4} 

¹Department of Port Logistics System, Tongmyong University, Busan, Republic of Korea

²Department of Computing and Mathematics, Manchester Metropolitan University, Manchester, UK

³School of Technology, Woxsen University, Hyderabad, India

⁴Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Punjab, India

Abstract

Ensuring the integrity of shipping containers is crucial for maintaining product quality, logistics efficiency, and safety in the global supply chain. Damaged containers can lead to significant economic losses, delays, and safety hazards. Traditionally, container inspections have been manual, which are labor-intensive, time-consuming, and error-prone, especially in busy port environments. This study introduces an automated solution using the YOLO-NAS model, a cutting-edge deep learning architecture known for its adaptability, computational efficiency, and high accuracy in object detection tasks. Our research is among the first to apply YOLO-NAS to container damage detection, addressing the complex conditions of seaports and optimizing for high-speed, high-accuracy performance essential for port logistics. Our method showcases YOLO-NAS's superior efficacy in detecting container damage, achieving a mean average precision (mAP) of 91.2%, a precision rate of 92.4%, and a recall of 84.1%. Comparative analyses indicate that YOLO-NAS consistently outperforms other leading models like YOLOv8 and Roboflow 3.0, which showed lower mAP, precision, and recall values under similar conditions. Additionally, while models such as Fmask-RCNN and MobileNetV2 exhibit high training accuracy, they lack the real-time assessment capabilities critical for port applications, making YOLO-NAS a more suitable choice. The successful integration of YOLO-NAS for automated container damage detection has significant implications for the logistics industry, enhancing port operations with reliable, real-time inspection solutions that can seamlessly integrate into predictive maintenance and monitoring systems. This approach reduces operational costs, improves safety, and lessens the reliance on manual inspections,

Corresponding author:

Gyu Sung Cho, Department of Port Logistics System, Tongmyong University, Busan 48520, Republic of Korea.
Email: gscho@tu.ac.kr



Creative Commons Non Commercial CC BY-NC: This article is distributed under the terms of the Creative Commons Attribution-NonCommercial 4.0 License (<https://creativecommons.org/licenses/by-nc/4.0/>)

which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access page (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

contributing to the development of “smart ports” with higher efficiency and sustainability in container management.

Keywords

Container damage, object detection, computer vision, You Only Look Once—Neural Architecture Search (YOLO-NAS), port efficiency, logistics system, deep learning, risk analysis

Introduction

The global logistics sector increasingly relies on containerized transportation to sustain the flow of goods. Yet, with a significant proportion of containers exceeding two decades in use, structural damage has become a critical issue, threatening not only cargo integrity but also supply chain efficiency.¹ Such damage stems from repeated handling, harsh environmental exposure, and mechanical impacts, which remain inadequately addressed by traditional inspection methods. Manual inspections, still dominant in many ports, are labor-intensive, inconsistent, and prone to human error, especially under the pressure of high-volume operations. Moreover, automated solutions like OCR, laser scanning, and 3D imaging, while useful, fall short in detecting the diverse and often subtle damage patterns present in real-world scenarios.

This research addresses these challenges by focusing on the development of an automated damage detection framework that combines accuracy, scalability, and real-time applicability. The significance of detecting container damage extends beyond operational efficiency—it mitigates risks to cargo safety, reduces downtime, and enhances compliance with international shipping standards. However, the lack of high-quality annotated datasets and adaptable machine learning models has long hindered innovation in this field, creating a compelling motivation for exploring cutting-edge approaches.

Among available deep learning frameworks, YOLO-NAS stands out as an optimal choice due to its superior performance in balancing speed and accuracy.² Unlike older models such as Fast R-CNN and MobileNetV2, which tend to compromise one metric for the other, YOLO-NAS demonstrates exceptional adaptability in dynamic environments. Experimental results in this study show YOLO-NAS surpassing both YOLOv8 and Roboflow 3.0 in precision and recall when applied to container damage detection.³ These findings underline its capability to identify diverse damage types, from minor deformations to critical structural faults, under real-world operational constraints.

The decision to prioritize YOLO-NAS was further influenced by its potential to transform port operations. By reducing reliance on manual inspections, it offers a scalable solution that not only improves accuracy but also aligns with the industry’s push toward automation and digital transformation.

This work contributes to filling the existing gaps in automated container damage detection while advancing the application of deep learning in logistics, an area critical for ensuring global supply chain resilience.

The rest of this paper is structured as follows: the second section presents related concepts; the third section details the methodology, including the dataset, preprocessing, and YOLO-NAS architecture; the fourth section analyzes experimental results, and the fifth section concludes with the study’s implications and future research directions.

Fundamental concepts

Concepts of container terminal

Container terminals serve as pivotal nodes in the global supply chain, facilitating the transfer of containers between ships, trucks, and trains. Strategically located at seaports, these high-traffic facilities are designed to process substantial volumes of containerized cargo, ensuring seamless import and export operations. Equipped with advanced machinery, such as gantry cranes and straddle carriers, terminals optimize cargo handling to minimize delays and operational costs, meeting the diverse logistical demands of international trade.

An understanding of terminal workflows is crucial to contextualize the complexities associated with automating container damage detection. Terminal operations are typically divided into three core areas: vessel operations (loading and unloading containers from ships), truck operations (container transfer to and from trucks), and storage operations

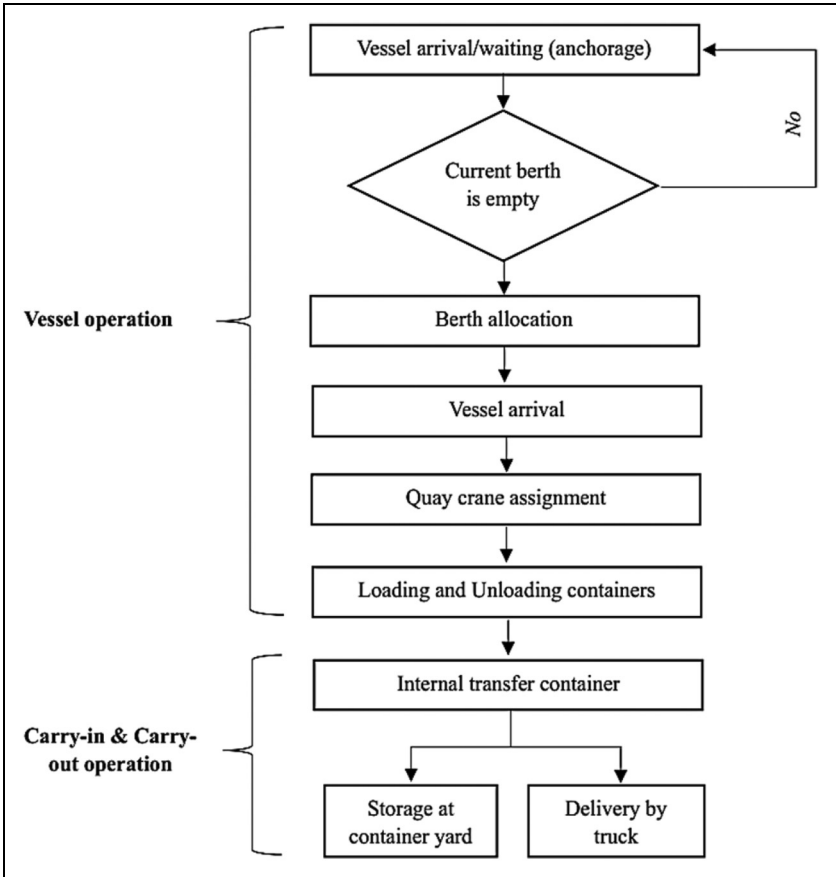


Figure 1. Container terminal operation (source: authors).

(temporary housing of containers). These interdependent processes, illustrated in Figure 1, underscore the rigorous operational environment where damage detection systems must function with precision and efficiency.⁴ Failure to meet these requirements risks disrupting cargo flows, heightening the need for a robust and scalable inspection framework.

To address these challenges, container terminals increasingly integrate cutting-edge technologies, such as AI-driven image analysis and automated scanning systems. These innovations enhance the accuracy and timeliness of damage detection, enabling swift interventions to prevent cargo losses and operational inefficiencies. By ensuring effective damage identification and prevention, these systems contribute to the safety and efficiency of terminal operations, reinforcing their critical role within the global logistics network.

Concepts and impacts of container damage

Container damage is a critical issue in shipping logistics, with profound implications for operational efficiency, safety, and economic viability. During transit, handling, and storage, containers are subjected to extensive mechanical stress and environmental exposure, resulting in structural issues such as dents, corrosion, and cracks (Figure 2). These damages compromise load-bearing capacity and functionality, while also posing significant safety risks. Financially, the depreciation of damaged containers, estimated at 2–5%, represents a substantial cost to the industry.⁵ Moreover, the potential collapse of structurally weakened containers in stacked configurations exacerbates these risks, endangering personnel and causing severe operational disruptions.

Traditional manual inspections remain the predominant method for container damage assessment; however, they are inherently labor-intensive, time-consuming, and prone to inconsistencies arising from human error. With the global container trade steadily increasing, the limitations of manual inspections have become more pronounced, leading to bottlenecks and inefficiencies in high-throughput environments.⁶ Automated detection technologies, by contrast, offer a more scalable and reliable alternative, providing rapid, consistent, and accurate assessments of container integrity. In the demanding operational context of port terminals, where timely damage identification is critical to

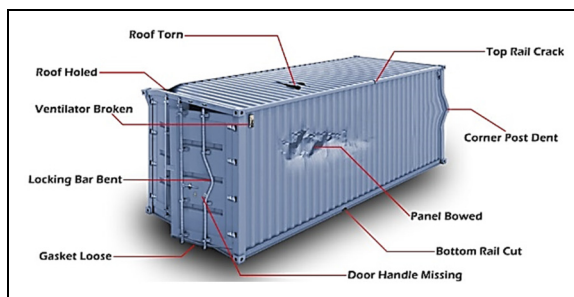


Figure 2. Types of damaged container (source: Oh JH⁶).

maintaining continuity and mitigating risks, such technologies are not just advantageous but indispensable.

This study highlights the potential of leveraging advanced object detection models, specifically YOLO-NAS, to address these challenges. Compared to traditional models, YOLO-NAS demonstrates superior accuracy and speed, making it particularly well-suited to the complex and dynamic conditions of port environments. By enabling early detection of damage, YOLO-NAS not only prevents cascading logistical failures but also enhances overall supply chain safety and efficiency. These attributes underscore its value as a robust and scalable solution for modern port operations.

In this research, “damage” is precisely defined as any visible physical defect or abnormality on a container’s surface or structure that could compromise its integrity, functionality, or safety. This definition encompasses a broad range of defects, including but not limited to dents, fractures, holes, and cracks, all of which have the potential to disrupt transportation and handling processes.

Overview of computer vision for damage detection

Computer vision, a branch of artificial intelligence, empowers machines to process and interpret visual data, enabling automated decision-making based on images or videos. In container damage detection, this technology is indispensable, facilitating rapid inspections that outperform traditional manual methods in both efficiency and accuracy. A key component of computer vision is object detection, which involves locating and identifying objects within an image. In the context of container damage, these techniques are crucial for accurately pinpointing the type and location of defects.⁷

Object detection methods are broadly categorized into region-based and single-shot approaches. Region-based detectors, such as the R-CNN family (e.g. Fast R-CNN and Faster R-CNN), are highly accurate but rely on multi-stage processes, making them unsuitable for real-time applications. Conversely, single-shot detectors like SSD (Single Shot Detector) and YOLO (You Only Look Once) streamline object detection into a single computational pass, prioritizing speed while maintaining competitive accuracy.^{8,9} This tradeoff makes single-shot models particularly effective in dynamic environments, such as high-traffic container terminals, where real-time performance is critical. As noted by Juan et al.,² the choice of detection model depends on specific application demands, including real-time constraints, accuracy requirements, and resource availability.

This technical foundation underscores the advantages of YOLO-NAS, a single-shot detection model optimized for both speed and precision. Its design uniquely addresses the dual requirements of rapid assessments and high accuracy, making it well-suited for container damage detection in fast-paced port environments. By integrating YOLO-NAS, operations can achieve reliable and efficient damage inspections, enhancing overall safety and workflow continuity in logistics systems.

Concepts of deep learning for damage detection

Deep learning, a pivotal subfield of artificial intelligence, has revolutionized container damage detection by significantly improving automation and accuracy. Among its

various methodologies, convolutional neural networks (CNNs) have demonstrated exceptional capabilities in processing large-scale visual data, making them indispensable for identifying complex patterns in container images and videos.^{10,11} By leveraging CNNs, deep learning models can detect a wide range of damage types—such as dents, cracks, and rust—including subtle defects often overlooked by human inspectors. This advancement not only enhances detection precision but also bolsters the reliability of inspections, thereby improving cargo safety and streamlining logistical operations across transportation networks.¹²

Building upon the strengths of CNNs, YOLO-NAS integrates Neural Architecture Search (NAS) to optimize its structure for specific tasks automatically.¹³ This synthesis enables YOLO-NAS to deliver both high accuracy and computational efficiency, crucial for real-time applications in high-demand settings like container terminals. The model's adaptability to diverse operational environments and its balance of speed and precision make it particularly well-suited for container damage detection. By addressing the dual imperatives of rapid assessment and dependable accuracy, YOLO-NAS emerges as a transformative tool for enhancing the safety, efficiency, and resilience of port operations.

Methodology

Overview of the proposed YOLO-NAS model and the YOLO-NAS model architecture

The YOLO (You Only Look Once) series has become a key player in object detection for applications such as robotics, autonomous vehicles, and video surveillance, thanks to its balance of speed and accuracy. Over time, each version has improved upon its predecessor to address challenges and boost performance. This paper presents YOLO-NAS, introduced by Deci in May 2023, which sets a new standard in real-time object detection by surpassing previous YOLO models and leading competitors in both speed and accuracy. YOLO-NAS excels in small object detection, enhances localization accuracy, and delivers superior performance relative to computational cost, making it highly suitable for real-time use on edge devices. Its open-source architecture also supports research purposes.¹⁴

The architecture of YOLO-NAS was developed through a NAS system called AutoNAC, designed to optimize the tradeoff between latency and throughput. This process resulted in three model variants: YOLO-NASS (small), YOLO-NASM (medium), and YOLO-NASL (large). These models differ based on the depth and arrangement of specialized Quantization-aware Skip Propagation (QSP) and Quantization-aware Convolutional Inference (QCI) blocks. These blocks are optimized for 8-bit quantization, reducing accuracy loss after post-training quantization (Sai et al.¹⁵). Unlike traditional architectures, YOLO-NAS can adapt its structure to specific tasks and datasets, making it especially useful for real-time applications and resource-limited environments. Its flexibility in exploring various architectural configurations opens the door to innovative designs that are difficult to achieve manually (Figure 3).

At its core, YOLO-NAS begins with convolutional layers that process input images into low-level feature maps, such as edges and textures, instead of merely increasing

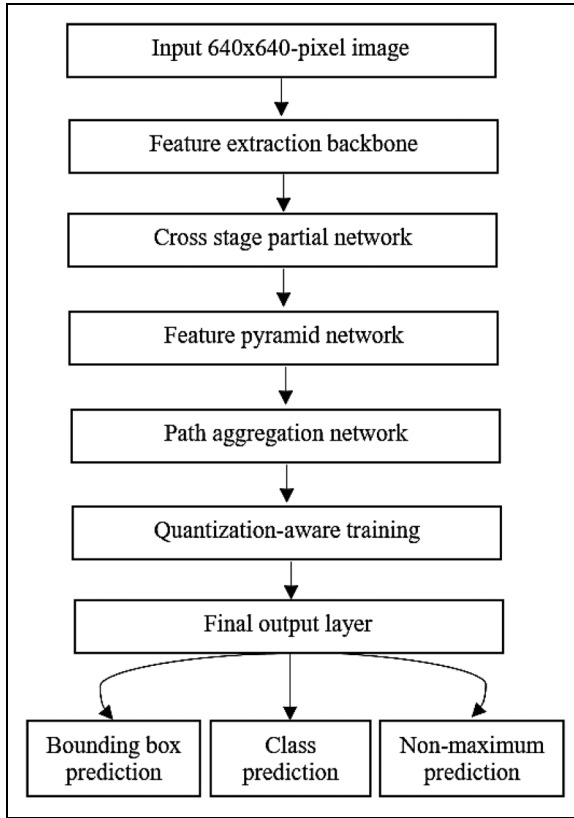


Figure 3. The YOLO-NAS architecture (source: authors).

their dimensions before sending them to deeper layers. Pooling layers downsample these feature maps while preserving high-level information. Additionally, YOLO-NAS employs CSPNet, which splits feature maps into two parts, processing one half through convolutional blocks before merging it back with the other half. This design improves the flow of information and computational efficiency.

Moreover, YOLO-NAS integrates the Path Aggregation Network (PAN) with the Feature Pyramid Network (FPN) to enhance object detection across various sizes. PAN fuses features from top-down and bottom-up pathways, while FPN generates feature maps at multiple levels, enabling the detection of both large and small objects. This combination allows YOLO-NAS to more accurately identify objects in complex environments.

Finally, YOLO-NAS predicts bounding boxes and class probabilities for detected objects in its output layer, using non-maximum suppression (NMS) to remove duplicate detections and keep the most reliable ones. This efficient object detection system is well-suited for a range of applications, including autonomous driving, surveillance,

and medical image analysis, where it consistently demonstrates state-of-the-art performance across diverse tasks.¹⁶

The novelties of YOLO-NAS in container damage detection

This research introduces several novel aspects by applying YOLO-NAS to container damage detection, offering advancements over traditional methods. First, YOLO-NAS utilizes NAS to optimize its architecture, enhancing its ability to detect complex and varied container damage patterns, such as dents, scratches, and structural defects, which may be missed by conventional models. Unlike existing approaches that often focus on specific types of damage or specialized containers, this research emphasizes multi-type damage detection, providing a more comprehensive solution to the diverse and often overlapping damage scenarios encountered in real-world environments.

Another key novelty is YOLO-NAS's real-time processing capability, which is critical in port settings where timely decisions are essential. The scalability of YOLO-NAS also makes it suitable for large-scale deployment, enabling integration into existing port infrastructures for automated and continuous monitoring.

Importantly, this study marks one of the first applications of YOLO-NAS in container damage detection, setting a precedent for the use of this advanced deep learning model in logistics and transportation. These innovations collectively aim to enhance safety, reduce operational costs, and improve the efficiency of container management, addressing current limitations in manual and semi-automated inspection methods. Through these contributions, the research opens new possibilities for the adoption of advanced AI in the logistics industry.

Dataset description and dataset preprocessing

The dataset for this study comprises images of shipping containers collected under diverse real-world conditions at seaports. These images capture a variety of container types and colors, encompassing a wide spectrum of physical defects such as dents, scratches, rust, and cracks—common anomalies in operational port environments. This variety ensures that the dataset is representative of real-world scenarios, enabling the model to generalize effectively across different damage cases encountered in the field.

A robust and well-curated dataset is pivotal for the model's performance, particularly in learning intricate patterns and producing accurate predictions. For this study, a custom hybrid dataset of 4587 carefully selected images was developed. Although the initial collection contained a higher number of images, redundant and low-quality samples were removed during preprocessing to maintain clarity and enhance dataset cohesion. This meticulous refinement ensured that the dataset was efficient and of high quality.

To meet the input requirements of the YOLO-NAS model, all images were resized to 640×640 pixels while preserving their original aspect ratios and stored in PNG format. The dataset, consisting of 4736 images, was split into training, validation, and testing sets in a 70:15:15 ratio. This allocation ensures a balanced evaluation of the model's performance at different stages of training.

The dataset is systematically organized to reflect real-world diversity in container damage scenarios and environmental conditions. This structured dataset ensures that the model encounters a wide range of damage types and operational conditions, enhancing its generalization capability. Table 1 provides a detailed breakdown of the dataset’s composition.

A total of 27,807 annotations were manually labeled across the dataset. These annotations focus on a single class called “damage” category encompassing all visible defects, simplifying the detection process while retaining robustness as can be seen in Figure 4. By consolidating all types of damage into a single label, the model focuses on detecting the presence of any damage, irrespective of its specific type. This simplification prioritizes detection accuracy and speed, aligning with the study’s goal of automating the initial identification of damaged containers.

This single-label approach is particularly advantageous for practical applications in seaport operations, where rapid identification of damaged containers is critical. By streamlining the detection process, this approach provides port operators with actionable insights to quickly identify containers requiring further inspection, without the additional complexity of categorizing damage types. This balance between simplicity and effectiveness ensures that the model remains efficient while meeting the operational demands of real-world deployment.

Implementation

The YOLO-NAS model is a machine-learning algorithm that uses a dataset of images of damaged shipping containers to detect damage. The model is trained using the preprocessed dataset, learning to detect damage by optimizing a predefined objective function. The hyperparameters are then fine-tuned for optimal performance. After training, the model’s performance is evaluated using key metrics like precision, recall, and mean average precision (mAP) to assess its accuracy in detecting damaged objects within shipping containers (Figure 5).

The following algorithm describes the inference process for detecting container damage using a pretrained YOLO-NAS model on Roboflow, following a systematic approach for image preprocessing, prediction, and post-processing. Here is the example code and hyperparameters used to train the proposed model:

Algorithm: Robust container damage detection using YOLO-NAS on Roboflow

Table 1. The breakdown of the dataset’s composition.

Category	Details
Types of damage	Dents (40%), scratches (30%), rust (20%), cracks (10%)
Environmental conditions	Daylight (60%), night (20%), rainy (10%), foggy (10%)
Container types	Standard (70%), refrigerated (20%), open-top (10%), tank (10%)
Container colors	Red (30%), blue (25%), green (20%), yellow (15%), others (10%)



Figure 4. The sample of annotated dataset (source: authors).

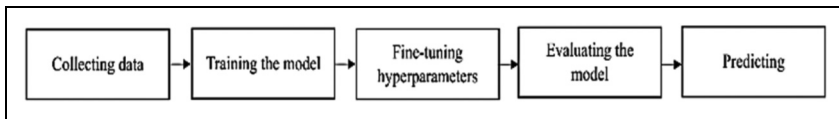


Figure 5. The process of the proposed model (source: authors).

Training setup and hyperparameters

The YOLO-NAS model was trained on a custom dataset with the following hyperparameters:

- Learning rate: 0.001
- Batch size: 16
- Epochs: 300
- Image size: 640×640
- Confidence threshold (τ_c): 40
- Overlap threshold for NMS (τ_o): 30
- Hardware: NVIDIA RTX 3090 GPU

The dataset was split into training (70%), validation (15%), and testing (15%) sets, and included domain-specific augmentations like lighting adjustments to improve model robustness.

Inference method

Input:

I : image file (e.g. container image).

$\tau_c = 40$: confidence threshold for object detection (the confidence score threshold is used to discard low-confidence predictions).

$\tau_0 = 30$: overlap threshold for NMS (the overlap threshold is used in NMS to remove redundant bounding boxes).

T : set of domain-specific transformations (e.g. augmentation).

M : pretrained YOLO-NAS model.

Φ : function for image normalization and preprocessing (image preprocessing function that includes normalization and domain-specific transformations like lighting adjustments, augmentations, etc.).

Ψ : function for post-processing and filtering false positives (post-processing function that filters false positives and refines predictions based on domain-specific knowledge).

Output:

P : prediction results (bounding boxes and class labels).

V : visualized prediction image.

Procedure:

1. Import the Roboflow library:

$$\text{Import } R$$

where R is the Roboflow API.

2. Initialize Roboflow with API key:

$$R.initialize(API_KEY).$$

3. Access workspace and Select project:

$$W = R.get_workspace(workspace_id)$$

$$M = M.get_model(model_endpoint)$$

where W is the workspace and M is the selected model.

4. Preprocess image:

$$I' = \Phi(I)$$

where Φ includes image normalization and domain-specific transformations from T .

5. Predict object detection:

$$P = M.predict(I', \tau_c, \tau_0)$$

where P is the set of predicted bounding boxes and associated class labels, calculated as:

$P = \{ (b_i, c_i, s_i) \mid b_i = \text{bounding box}, c_i = \text{class label}, s_i = \text{confidence score}, \forall_i \in [1, N] \}$ with N being the number of detections.

6. Post-process results:

$$P' = \Psi(P, \tau_c, \tau_0)$$

Apply post-processing steps Ψ including filtering based on domain-specific criteria (e.g. removing false positives related to background noise).

7. Visualize predictions

$$V = \text{visualize}(I', P')$$

8. Return prediction results and Visualization

$$\text{Return } P', V$$

Explanation of key steps. The above algorithm effectively combines the original object detection capabilities of Roboflow with domain-specific enhancements for container damage detection in seaports. Preprocessing involves normalizing image data and applying transformations to replicate real-world conditions. The YOLO-NAS model is then employed to predict bounding boxes and class labels, delivering high-precision object detection with confidence scores. Finally, post-processing steps refine these predictions, improving accuracy by filtering false positives and accounting for the complexities of the seaport environment. This comprehensive approach uses the high performance of YOLO-NAS, optimizing it for the specific challenges of detecting container damage, and ensuring that the system remains robust and reliable in diverse operational settings.

Modification of the algorithm. To address the unique challenges of container damage detection at seaports, several domain-specific modifications were made to the object detection algorithm. First, a custom preprocessing step was introduced to normalize images under varying lighting conditions, such as intense sunlight or low-light scenarios, ensuring consistent image quality for more accurate predictions. Additionally, custom data augmentation techniques were developed, simulating real-world conditions like rust, dirt, and aging containers to enhance the model's generalization capabilities. YOLO-NAS was integrated into the pipeline as the primary detection model, replacing the default Roboflow model, which required specific adjustments for effective utilization.

In post-processing, the algorithm was enhanced to filter out false positives caused by background noise, such as stacked containers and moving cranes, using domain-specific knowledge. Dynamic adjustments to confidence (τ_c) and overlap (τ_o) thresholds were also implemented, allowing the model to adapt in real time to varying visibility conditions. Moreover, the algorithm was integrated into a real-time monitoring system, enabling instant feedback and automated alerts when significant damage is detected addition that was not part of the original Roboflow implementation. These modifications significantly improve the accuracy and reliability of container damage detection in seaports, differentiating this approach from standard object detection applications and contributing to the novelty of the research.

Key performance metrics

Although object detection may seem like a straightforward task in computer vision, it requires a careful and nuanced approach to achieve accurate results. Object detection, as the term implies, involves identifying and locating objects within an image or video frame. Essentially, we train AI to recognize and differentiate between various target objects present in the visual data.

Yet, how can we ensure that the detected objects are accurate and that the algorithm is performing optimally? This is where object detection metrics come into play. Indranath C and Gyusung Cho¹² suggests these metrics to help evaluate the model’s effectiveness. In this paper, metrics such as precision, recall, and mAP are employed to assess the accuracy of the proposed model. The key performance metrics used to gauge the model’s success are outlined below.

Precision is calculated as $\frac{\text{True positives}}{\text{True positives} + \text{false positives}}$.

A high precision value is indicative of the model’s ability to make fewer false positive predictions. In other words, the model is more selective and accurate when predicting positive instances.

Recall is computed as $\frac{\text{True positives}}{\text{True positives} + \text{false negatives}}$.

A high recall value suggests that the model is effective at capturing most of the positive instances, thereby minimizing the number of false negatives.

mAP is determined as $\frac{1}{N} \sum_{i=1}^N AP_i$. It is calculated by finding the average precision for each class and then averaging over several classes. This metric incorporates the tradeoff between precision and recall, providing a comprehensive measure of the model’s performance.

Results

Results and analysis

The results section of this study, as depicted in Figures 6 and 7, provides an exhaustive evaluation of the YOLO-NAS model’s performance in container damage detection,

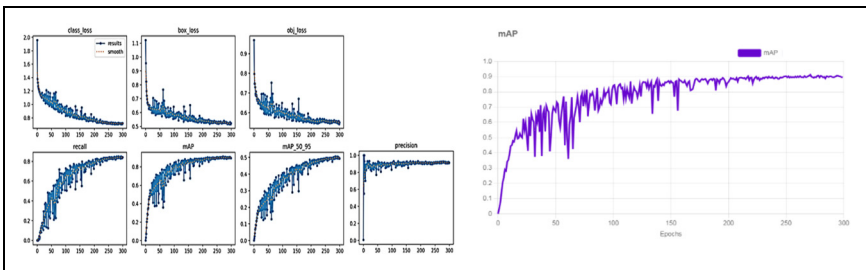


Figure 6. The process of the proposed model (source: authors).



Figure 7. The prediction results (source: authors).

Table 2. The key performance metrics.

mAP	Precision	Recall
91.2%	92.4%	84.1%

utilizing key performance metrics such as precision, recall, and mAP. The model achieved an impressive mAP of 91.2%, reflecting its robust capability to precisely identify relevant instances while effectively minimizing false positives. The precision score of 92.4% underscores the model’s high predictive accuracy, indicating its proficiency in correctly identifying true positives with minimal false positive errors. Additionally, the recall rate of 84.1% demonstrates the model’s comprehensive detection ability, highlighting its effectiveness in capturing a significant proportion of true positive instances. This demonstrates the model’s proficiency in capturing a significant portion of relevant instances (Table 2).

Moreover, the model exhibits a marked and consistent reduction in class_loss, box_loss, and obj_loss values, signifying enhanced classification accuracy, refined object localization, and improved detection robustness. This is corroborated by an upward trajectory in performance metrics, including precision, recall, mAP, and mAP50-95, which collectively indicate balanced and substantial advancements in detection efficacy and model stability.

To ensure methodological rigor and dataset transparency, the dataset was systematically partitioned, with 88% allocated to the training set (9645 images), 6% to the validation set (685 images), and 6% to the test set (687 images), comprising a total of 11,017 images. The dataset encompasses a diverse array of container damage types to enhance the model’s generalizability and ensure representativeness. Furthermore, the dataset distribution is consistent across all stages of model development, contributing to reliable and reproducible outcomes. For transparency and reproducibility, the

dataset is publicly accessible at the Roboflow Container Damage Dataset (<https://universe.roboflow.com/thanh-fscay/container-damage-hmv17/dataset/1>).

This rigorously structured approach, coupled with comprehensive dataset documentation and accessible data sources, reinforces the methodological robustness and reproducibility of the experimental findings, thereby validating the model’s applicability in real-world scenarios.

Discussion of the novelties of this study

YOLO-NAS demonstrates exceptional capabilities in detecting machine anomalies, excelling in tasks requiring high precision and rapid detection. Achieving notable metrics, including a mAP of 91.2%, precision of 92.4%, and recall of 84.1% on a dataset, it has proven reliable for general anomaly detection. However, its application to detecting structural damage in shipping containers presents novel challenges, such as environmental noise at port terminals and diverse damage types.

To address concerns regarding comparison consistency, Table 3 has been updated to evaluate YOLO-NAS, Fmask-RCNN, and MobileNetV2 on container damage datasets. It is important to note that although all models are trained on container damage datasets, which are not identical in terms of image composition and coverage. Fmask-RCNN achieves a low miss rate of 4.599% but suffers from an error rate of 18.887%, limiting its real-time applicability.¹⁷ MobileNetV2 shows promising accuracy for multi-type damage detection but lacks key metrics like mAP and recall, essential for evaluating performance in complex operational contexts. In contrast, YOLO-NAS consistently outperforms these models, demonstrating its robustness and reliability across diverse metrics. These comparisons underscore the adaptability and effectiveness of YOLO-NAS for container damage detection, even within the broader context of varying dataset compositions.

The comparative evaluation in Table 4 and Figure 8 underscores YOLO-NAS’s superiority over other advanced models trained on the same dataset, including YOLOv8 and

Table 3. The comparison between models studied in the literature.

YOLO-NAS	Fmask-RCNN	MobileNetV2
mAP: 91.2% Precision: 92.4% Recall: 84.1%	Miss rate: 4.599% Error rate: 18.887%	Training accuracy: 95.32%

Table 4. The comparison between other deep learning models.

YOLO-NAS	YOLOv8	Roboflow 3.0 Object Detection (Fast)
mAP: 91.2% Precision: 92.4% Recall: 84.1%	mAP: 63.6% Precision: 85.1% Recall: 72%	mAP: 57.9% Precision: 85.4% Recall: 56.5%

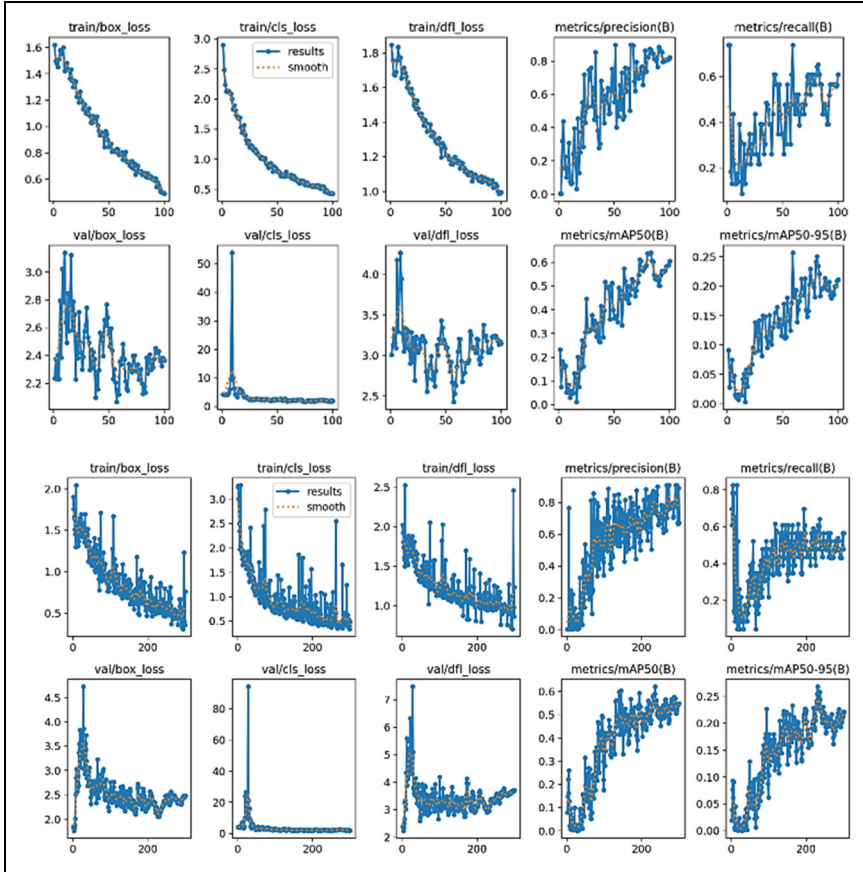


Figure 8. The process of the YOLOv8 model and the process of the Roboflow 3.0 Object Detection (Fast) (source: authors).

Roboflow 3.0 Object Detection (Fast). YOLOv8 achieves a mAP of 63.6%, with a precision of 85.1% and recall of 72%, while Roboflow 3.0 performs at a mAP of 57.9%, precision of 85.4%, and recall of 56.5%. In contrast, YOLO-NAS outperforms these models across all metrics, attaining a mAP of 91.2%, precision of 92.4%, and recall of 84.1%. Training these models under identical conditions and on the same dataset emphasizes YOLO-NAS's effectiveness, demonstrating its capability to address container damage detection challenges with superior accuracy and real-time efficiency. These consistent results solidify YOLO-NAS as an optimal solution for the specific demands of container terminal operations.

To summarize, our analysis reveals that YOLO-NAS exhibits significant advantages over Fmask-RCNN, MobileNetV2, YOLOv8, and Roboflow 3.0 Object Detection (Fast) across various dimensions. Fmask-RCNN, despite its notable accuracy in object

detection, is constrained by the slower inference speeds inherent to its two-stage detection process, rendering it less effective for real-time applications. In contrast, YOLO-NAS utilizes a one-stage detection framework optimized through NAS, achieving a superior balance between speed and accuracy.

Similarly, MobileNetV2, designed for mobile and edge applications, provides efficient inference but compromises accuracy, particularly in detecting small or occluded objects within complex environments. YOLO-NAS overcomes these limitations by leveraging an enhanced FPN, which significantly improves multi-scale detection and ensures consistent performance across diverse object sizes. Compared to YOLOv8, another model within the YOLO family, YOLO-NAS incorporates architectural enhancements such as optimized feature extraction and dynamic activation functions, resulting in improved mAP and faster convergence during training.

Lastly, while Roboflow 3.0 Object Detection offers a user-friendly and rapid solution, it lacks the architectural sophistication and precision of YOLO-NAS, making it less effective for high-stakes applications in challenging environments. Overall, YOLO-NAS surpasses these models not only in accuracy and inference speed but also as a flexible and efficient solution tailored to the unique demands of shipping container damage detection.

By introducing YOLO-NAS to the field of shipping container damage detection, this research pioneers a more efficient, accurate, and scalable approach for port operations. This application represents a clear novelty, as YOLO-NAS has not been previously used in this domain, filling a critical gap in container damage assessment while offering better overall detection capabilities.

Applications

The application of the YOLO-NAS model in port container terminals is crucial for enhancing operational efficiency and safety. Given its impressive performance metrics, YOLO-NAS is well-suited for the demanding conditions of port environments, where real-time accuracy and reliability are paramount.

The YOLO-NAS model can be effectively implemented in a port environment to automate container damage detection, streamline inspection processes, and ensure continuous real-time monitoring. To accommodate the dynamic nature of container movement, the system captures images in real time as containers move through designated detection portals or while being handled by ship-to-shore cranes. High-resolution cameras equipped with the YOLO-NAS model are strategically positioned at critical locations, such as entry and exit points, loading and unloading areas, and transport routes. This setup enables the model to capture multiple angles of each container, ensuring thorough coverage without interrupting container flow. As trucks pass through the detection portals, YOLO-NAS automatically inspects containers for structural deformities, including dents, bulges, cracks, and corrosion, without requiring vehicles to stop. This eliminates the need for manual inspections, significantly reducing gate transaction times and improving overall turnaround efficiency.

In real-time port operations, ensuring the quality of input data for model training is crucial for maintaining high detection accuracy. To meet these technical requirements,

training images are carefully preprocessed to standardize parameters such as resolution, lighting, and contrast, ensuring the model is exposed to realistic conditions similar to those in the operational environment. Furthermore, augmentation techniques, including brightness adjustments and varied angles, are applied to make the model adaptable to the diverse lighting and weather conditions encountered in ports. This approach allows YOLO-NAS to maintain its accuracy and robustness when inspecting containers under different lighting conditions or in adverse weather.

Handling stacked and overlapping containers, which is common in port environments, poses additional challenges. To address this, cameras are positioned at multiple angles to capture unobstructed views of containers even when they are stacked. In addition, YOLO-NAS incorporates advanced image processing techniques such as NMS to minimize overlapping detections, enabling it to isolate individual containers and identify specific damage types despite visual obstructions. In scenarios where stacking creates blind spots, alternative angles are used to capture additional shots, ensuring consistent inspection coverage even in crowded container yards.

By implementing YOLO-NAS within a real-time, automated inspection system, the model can recognize and report container damage instantly. The model triggers an automated alert if damage is detected, enabling rapid response and prioritizing damaged containers for further inspection or repair. This automated process not only enhances safety by preventing damaged containers from entering the supply chain but also helps port terminals comply with international shipping regulations that mandate thorough inspections and documentation of container conditions. Additionally, the data collected through continuous inspections provides valuable insights into damage trends, enabling port operators to optimize maintenance strategies and reduce future incidents.

Overall, the application of YOLO-NAS in port environments addresses the operational demands of real-time inspection, minimizes delays associated with manual checks, and supports data-driven decision-making. By automating container damage detection, YOLO-NAS enhances operational efficiency, safety, and compliance in modern port management.

Scalable deployment strategies for YOLO-NAS in future

Nevertheless, as container damage detection systems transition from research to real-world applications, the scalability and flexibility of deployment frameworks become pivotal. Two primary approaches, cloud-based infrastructure and edge AI implementation, offer complementary solutions to address the unique demands of port operations which can be integrated into future implementations.

A cloud-based deployment centralizes data processing and storage, leveraging powerful computing resources to manage high volumes of container image data. In this setup, images captured at the port are transmitted to a cloud server where the YOLO-NAS model processes them to detect damage. The results are then shared with relevant stakeholders in real time. Consequently, centralized data storage enables the application of advanced analytics and machine learning to detect trends, predict maintenance needs, and improve operational efficiency over time. Besides, cloud platforms can dynamically

scale computing resources to handle fluctuating workloads, particularly during peak traffic.

Meanwhile, Edge AI brings the processing capability directly to the point of data collection, using edge devices such as embedded systems or AI-powered cameras. This approach enables real-time damage detection without relying on constant connectivity to a central server. As a result, by processing data locally, not only does edge AI eliminate the need for large-scale data transmission, making it ideal for time-critical operations, but it also ensures uninterrupted performance even in areas with poor or inconsistent network coverage because of operating independently.

Furthermore, a hybrid approach combining cloud-based and edge AI solutions could leverage the strengths of both strategies. Take edge devices as an example, they can handle initial real-time detection. At port entry points, edge devices embedded in cameras can quickly assess container conditions and flag anomalies. These flagged cases are then uploaded to the cloud for further review and documentation, enabling comprehensive damage tracking without overloading edge systems. This dual-layer architecture ensures responsiveness while supporting large-scale analytics and coordination.

In the future, integration with Internet of Things networks, where edge devices communicate seamlessly with other port systems, such as gate automation and crane monitoring, could further streamline operations. Also, port authorities could allow models deployed at multiple ports to collaboratively improve without sharing raw data, addressing privacy concerns while enhancing performance across locations.

Conclusion and future scope

The YOLO-NAS model demonstrates exceptional effectiveness in detecting container damage, achieving a precision of 92.4%, recall of 84.1%, and mAP of 91.2%. Its ability to identify subtle or complex damages, combined with real-time processing capabilities, makes it a robust solution for the demanding environment of container terminals. By automating inspections, YOLO-NAS minimizes human error, ensures compliance with international shipping regulations, and enhances operational efficiency in port management.

Despite its advantages, the model faces limitations, including its reliance on high-quality, annotated datasets and potential challenges in detecting damage in cluttered or occluded environments. Small defects, such as rust stains, may also go undetected, and its high computational demands can pose difficulties for resource-constrained settings.

Future research on YOLO-NAS could focus on enhancing its adaptability and efficiency for real-world port environments. A key improvement lies in integrating multi-class detection, enabling the system to identify not just container damage but also other anomalies like container mislabeling or structural issues. This would make the model more versatile and applicable to a broader range of use cases. Additionally, exploring hardware-efficient model variants, such as lightweight architectures or quantized versions of YOLO-NAS, would enable its deployment on resource-constrained devices, broadening its adoption in ports with limited infrastructure.

Moreover, extending the testing framework to simulate variable environmental conditions, such as fluctuating lighting, weather changes, or heavy port traffic, would strengthen the model's robustness. Testing under such diverse conditions would ensure better generalization, making the system reliable across different operational settings. Advanced data augmentation techniques could replicate these scenarios during training, preparing the model for the complexities of real-world port environments.

What is more, integrating YOLO-NAS with existing automation systems could create a comprehensive solution, combining damage detection with tasks like container ID recognition, weight monitoring, and gate automation, thereby streamlining port operations. Leveraging YOLO-NAS-generated data for predictive maintenance represents another promising direction. By analyzing damage patterns over time, ports could anticipate maintenance needs, reducing disruptions and enhancing supply chain resilience.

Finally, optimizing the model's computational demands through hardware-specific improvements could make it globally accessible, especially in regions with limited resources. These advancements have the potential to transform port logistics, improving safety, efficiency, and operational effectiveness in global supply chains.

Authors' contributions

TNTP conducted the programming tasks, performed primary data analysis, and drafted the initial manuscript. GSC collected the dataset, identified the research problem, contributed to hypothesis development, and supervised the project; provided significant critical revisions to the manuscript. IC collaborated in problem identification and hypothesis formulation, supervised the overall study, and critically reviewed and refined the manuscript for submission. All authors contributed to the research and approved the final manuscript.


Declaration of conflicting interests

The authors declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The authors disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research was supported by the "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2023RIS-007).

ORCID iD

Indranath Chatterjee  <https://orcid.org/0000-0001-9242-8888>

References

1. Zixin W, Jing G, Qingcheng Z, et al. Multitype damage detection of containers using CNN based on transfer learning. *Math Probl Eng* 2021; 2021: 1–12.
2. Yogesh K and Pankaj K. Comparative study of YOLOv8 and YOLO-NAS for agriculture application. *Int Conf Signal Process Integr Net* 2024; 2024: 72–77.

3. Deepak M, Prashant K, Sunil S, et al. A metaphor analysis on vehicle license plate detection using Yolo-NAS and Yolov8. *J Electr Sys* 2024; 20: 152–164.
4. Indranath C and Gyusung C. Port container terminal quay crane allocation based on simulation and machine learning method. *Sens Mater* 2022; 34: 1–11.
5. Magdalena K, Dorota L, Karolina B, et al. Review of the container ship loading model—cause analysis of cargo damage and/or loss. *Polish Maritime Res* 2022; 29: 26–35.
6. Oh JH, Hong SW, Choi GJ, et al. Development of the container damage inspection system. *J Korean Soc Precis Eng* 2005; 22: 82–88.
7. Li L, Wanli O, Paul F, et al. Deep learning for generic object detection: a survey. *Int J Comp Vis* 2020; 128: 261–318.
8. Chengji L, Yufan T, Jiawei L, et al. Object detection based on YOLO network. In 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), Chongqing, China, 14–16 December 2018, pp. 799–803.
9. Guillem D, Andoni C and Estibaliz L. Pipeline for visual container inspection application using deep learning. In: *Proceedings of the 14th International Joint Conference on Computational Intelligence (IJCCI 2022) – Vol. 1: NCTA*; ISBN 978-989-758-611-8, SciTePress, pp. 404–411. DOI: 10.5220/0011590900003332.
10. Yann L, Yoshua B and Geoffrey H. Deep learning. *Nature* 2015; 521: 436–444.
11. Yanming G, Yu L, Ard O, et al. Deep learning for visual understanding: a review. *Neurocomputing* 2016; 187: 27–48.
12. Olga R, Jia D, Hao S, et al. Imagenet large scale visual recognition challenge. *Int J Comp Vis* 2015; 115: 211–252.
13. Juan D. Understanding of object detection based on CNN family and YOLO. *J Phys* 2018; 1004: 1–8.
14. Juan T, Diana-Margarita C and Julio-Alejandro R. A comprehensive review of YOLO architectures in computer vision: from YOLOV1 to YOLOV8 and YOLO-NAS. *Mach Learn Knowl Extr* 2023; 5: 1680–1716.
15. Sai CM, Chennupati VS, Sagar DP, et al. Detection of hand bone fractures in X-ray images using hybrid YOLO NAS. *Inst Electr Electr Eng* 2024; 12: 57661–57673.
16. Edmundo C, Leo R, Eduardo B, et al. Assessing the effectiveness of YOLO architectures for smoke and wildfire detection. *Inst Electr Electr Eng* 2023; 11: 96554–96583.
17. Xueqi L, Qing L, Jinbo W, et al. Container damage identification based on Fmask-RCNN. *Commun Comp Inf Sci* 2020; 1265: 12–22.