Master's Thesis

Detection of Broken Seals on Containers from Video Recordings Based on YOLO model

Sijun Yu

Limassol, Month and year of thesis submission

CYPRUS UNIVERSITY OF TECHNOLOGY

FACULTY OF ENGINEERING AND TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER ENGINEERING AND INFORMATICS

Master's Thesis

Detection of Broken Seals on Containers from Video Recordings Based on YOLO model

Sijun Yu

Supervisor

Faculty Name Surname and position

Limassol, Month and year of thesis submission

Approval Form

Master's Thesis

Detection of Broken Seals on Containers from Video Recordings Based on YOLO model

Presented by

Sijun Yu

Supervisor: [Name Surname and position]

Member of the committee: [Name Surname and position]

Member of the committee: [Name Surname and position]

Cyprus University of Technology

Limassol, Month and year of thesis submission

Copyrights

ABSTRACT

Container seal integrity verification represents a critical yet underexplored challenge in maritime logistics security. As the first study dedicated to automated broken seal detection in container terminals, this research introduces a novel dual-model architecture combining container door localization and seal defect recognition. The proposed framework addresses two major bottlenecks in surveillance video analysis: A YOLO-based door detection module eliminating more than 90% of irrelevant regions, and A fine-tuned seal inspection model enhanced by dark channel prior dehazing, achieving 92.9% mAP on our custom dataset from Limassol Port. Through systematic evaluation under varying illumination and occlusion conditions, our cascaded detection system demonstrates 23.5% higher precision than single-model approaches while maintaining real-time processing The results show that the YOLO-based model provides an efficient and reliable solution for real-time detection of container seal damage in port environments, offering significant advantages over traditional methods in terms of speed, accuracy, and scalability.

Keywords: Container Seal Detection, YOLO Model, Transfer Learning, Machine Learning, Automated Inspection

Table of Contents

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

YOLO:	You Only Look Once
NMS:	Non-Maximum Suppression
mAP:	mean Average Precision
IoU	Intersection over Union
FPS	Frames Per Second

1 Introduction

With the global trade volume growing at an average annual rate of 3.4% (UNCTAD, 2023), container transportation undertakes more than 80% of the global cargo transportation volume (World Shipping Council, 2022). However, statistics from the International Maritime Organization (IMO) show that the annual loss of goods due to broken seals amounts to as much as 1.7 billion US dollars, among which approximately 34% of the cases result from manual inspection omissions (IMO, 2021). This current situation exposes the fundamental flaws of traditional detection methods. Containers are often subjected to external influences such as mechanical impact and adverse weather conditions during transportation, leading to seal damage. Seal damage not only causes harm to the goods but can also pose security risks and environmental pollution. Therefore, efficiently and accurately detecting container seal damage has become a critical issue in ensuring the safety of container transport.

Traditional manual inspection methods are inefficient and susceptible to human error, failing to meet the automation and efficiency demands of modern ports. With the rapid advancement of computer vision and deep learning technologies, automated detection methods have gradually become the ideal solution to this problem. Particularly, object detection models such as YOLO can identify and locate seal damage in real-time from video streams, improving detection efficiency and accuracy.

This study aims to explore the application of deep learning techniques, especially the YOLO model, in detecting damaged seals on containers, analyzing its practical performance and optimization in port environments. By using transfer learning and other methods, the study enhances the model's adaptability and accuracy under complex conditions and addresses the challenges of high-frequency data processing in video streams. This research not only provides an efficient automated solution for container seal damage detection but also offers new perspectives and practical experience for the application of deep learning in industrial inspections.

1.1 Motivation

Container transport is at the core of global trade, involving the cross-border movement of millions of containers. As logistics demands continue to rise, the safety and efficiency of container transport have become critical issues for global port management. Seals, as vital safeguards for container security, are directly related to the integrity and safety of the goods. During container loading and unloading, damaged seals may lead to loss, damage, or theft of goods, and may also affect the speed of customs clearance and port safety management. While significant research efforts have been devoted to detecting structural container damage (e.g., dents, corrosion), the critical task of seal integrity verification remains conspicuously absent from automated inspection systems. Therefore, timely and accurate detection of container seal damage has become an important method for ensuring safe port operations and improving efficiency.

At the container terminal of Limassol Port, IPTV cameras installed on cranes capture video streams of the container loading and unloading process. The continuous flow of containers makes automated seal inspection feasible, as seals play a crucial role in ensuring the safety and integrity of goods during transport. The goal of this study is to use computer vision and machine/deep learning technologies to detect whether container seals are damaged or intact, particularly during the container handling process. This method aims to replace traditional manual inspection and improve the efficiency and accuracy of monitoring container movements.
This study takes the typical challenges of the Port of Limassol, a hub port in the Mediterranean, as the entry point to reveal three core pain points:

- Blind spots in manual inspection: At a loading and unloading rate of over a hundred boxes per hour, it is difficult for the naked eye to detect the damage to the seals that occurs instantaneously
- Environmental interference multiplicities: Specific factors such as sea fog cause traditional image enhancement algorithms to fail.

- Detection granularity imbalance: The existing YOLO model has a detection accuracy of over 90% for the main body of containers, but the missed detection rate surges when directly transferred to the sealing scene

This study pioneers a vision-based dual-detection framework that:

- Phase One - Spatial Positioning: Through an adaptive door body recognition algorithm, the sealing area is locked in the dynamic video stream
- Phase Two - Micro-Defect Analysis: Integrating physical prior knowledge with deep learning to achieve sub-pixel-level damage discrimination

1.2 Thesis Goals and Contributions

The primary goal of this study is to develop a YOLO-based container seal damage detection system capable of automatically identifying and marking damaged seals from video streams. Specific objectives include:

- How to accurately detect container seal damage using computer vision techniques. Under the conditions of complex background and motion blur , achieve a positioning accuracy rate of more than 98% in the container door area
- How to optimize the YOLO model through transfer learning to improve detection accuracy, especially in port environments.
- How to implement real-time seal detection in high-frequency video streams to reduce false positives and missed detection. Finish the end-to-end processing in the 1080p@30fps video stream with minimal single-frame delay

To achieve these goals, this study will involve data collection, model training, and optimization, followed by experimental validation.

The contributions of this study include:

- Proposing a YOLO-based method for detecting seal damage, filling a gap in the existing literature on automated container seal detection.
- Optimizing the YOLO model through transfer learning, improving its detection accuracy and real-time performance in actual port environments.

- Exploring and validating the application of deep learning methods in large-scale video data streams, demonstrating their potential and advantages in automated detection.
- The three-level architecture of "location - enhancement - detection" is proposed, and the computing resources are focused on the seal area through the spatial attention mechanism
- Integrate the dark channel prior and the adversarial generative network to construct an adaptive image enhancement module that can cope with extreme environments

This study not only has theoretical value but also significant practical implications. By achieving automated container seal detection, it will effectively enhance the safety and efficiency of the port loading and unloading process, providing valuable insights for automated detection in other fields.

1.3 Outline of Thesis

The structure of this thesis is as follows:

- Chapter2: Literature Review

This chapter provides an overview of the related research on computer vision and transfer learning in object detection for cargo transport, especially containers, analyzing the strengths and weaknesses of existing methods and highlighting the innovations of this study. This chapter offers theoretical support for the subsequent chapters by comparing the current research with the approach proposed in this study.

- Chapter3: Methodology

This chapter describes the technical framework and methodology used in the study, including the basic principles of the YOLO model, the implementation process of transfer learning, data preprocessing, and feature extraction methods. The focus is on how the YOLO model is applied to container seal damage detection, along with the training and optimization process.This chapter proposes an improved YOLO framework integrated with dark channel prior-based dehazing. It elaborates on the data collection and preprocessing pipeline, anchor optimization strategies for small object detection, inter-frame continuity handling in videos, and model training methods, offering an end-to-end solution for seal damage detection.

- Chapter4: Experimental Results

This chapter presents the experimental design, dataset construction, and model training process in detail, followed by an in-depth analysis of the results. The applicability of the model in the real port environment was analyzed by combining indicators such as the confusion matrix and the PR curve.

- Chapter5: Conclusion and Future Work

This chapter summarizes the main findings of the research, discusses the limitations of the study, and explores possible directions for future research. It also emphasizes the practical significance of this research for port automation and logistics management, and looks at its potential applications in other fields.

2 Literature review

2.1 Main Research

The following literature represents key advances in the field of target detection, particularly container detection, with distinctive research objectives and approaches

Liu et al.[1] (2023) proposed a lightweight container detection model (Lite-YOLOv5s) by integrating Ghost modules, BiFPN, and small-object detection layers, reducing parameters by 42.67% while improving accuracy (+1.7% mAP).It Provided efficient real-time detection solutions for port automation and verify the feasibility of lightweight design in complex scenarios.

Yu et al. [2] (2022) Developed a container anomaly detection system based on association rule mining, assisting operation and maintenance decision-making through log simplification and rule extraction. It was the first application of data mining technology to container operation and maintenance management, reducing manual analysis costs and improving the efficiency of abnormality location.

Li et al. (2021) [3] Designed a container contour detection algorithm based on traditional computer vision, integrating Canny edge detection, geometric structure matching and improved mean drift clustering. It achieved high precision (99% accuracy) detection without relying on deep learning, and provide a reliable alternative for low-computing power scenarios.

Lin et al. (2024) [4] proposed a container hold target detection process based on 3D point cloud with Deformable DETR to achieve centimeter-level localization (MAE <5 cm) of containers, hatch covers, and other objects.It promoted the automated operation of harbor cranes and validate the effectiveness of Transformer architecture in container 3D inspection for the first time.

Wang et al. (2021) [5] developed a multi-class damage detection model based on transfer learning, optimizing MobileNetV2 for identifying various types of targets. The introduction of transfer learning significantly improved model performance on small datasets and allowed the model to quickly adapt to complex safety detection needs.

Maeda et al. (2018) [6] used deep neural networks and smartphone-captured images for road damage detection, extending object detection methods to other damage detection applications. This study employed YOLO, optimizing feature extraction and recognition accuracy, which significantly contributed to improving detection efficiency and accuracy.

Azimi et al. (2018) [7] proposed a multi-class target detection method to address multi-type target detection issues in complex remote sensing images. By introducing multi-scale feature fusion and specialized loss functions, the model's robustness in heterogeneous environments was enhanced. This method is valuable for complex safety detection.

Zheng et al. (2021) [8] used an improved Faster R-CNN framework to enhance the model's geometric features and inference capabilities. By designing the CIoU (Complete IoU) loss function, the accuracy of bounding box predictions was improved, providing reliable technical support for high-precision object detection in safety detection.

2.2 Key Points and Achievements

Multi-type target detection optimization:

Enhance the sensitivity of the model to targets of different sizes through multi-scale feature fusion (BiFPN, Deformable DETR), e.g., achieve an accuracy of 0.97 in mixed container and hatch cover detection [4].

Verify the effectiveness of migration learning in small sample multi-type damage detection (e.g., rust, dents), and provide adaptive solutions for complex scenarios. [5]

Algorithm lightweight and real-time:

Realizes 28 FPS real-time detection (1080p) through Ghost module with parametric quantities cropping to meet the demand of port automation. [1]

combines traditional image processing (Canny+LSD) with geometric constraints to achieve 15 FPS on CPU devices, providing a reference for edge computing. [3]

Cross-modal data applications:
Lin et al. (2024) [4] Combining 3D point cloud projection with 2D detection for the first

time to solve the target localization problem under complex spatial layout of ship's cabin with MAE <5 cm.

Yu et al. [2] (2022) Expanding the application value of non-visual data (O&M logs) in container management through log rule mining.

Robustness enhancement method:

Li et al. (2021) [3] adopts improved mean drift clustering to effectively suppress surface texture and lashing interference (false detection rate <5%).

Wang et al. (2021) [5] improves the generalization ability of the model in complex environments in ports through data enhancement (e.g., light simulation).

Standardization and Scalability:

Maeda et al. (2018) [6] Systematically sort out the technical routes in the field, pointing out the limitations of the current dataset size (1,000-10,000 samples) and the singularity of the detection target (focusing on the container as a whole).

Broadly speaking, these studies have worked mainly in the following areas:

2.2.1 Optimization of Deep Learning Detection Algorithms

In research on container damage detection, optimizing object detection algorithms is one of the key directions. Traditional object detection algorithms mostly focus on identifying a single type of target. However, container damage is diverse and varies in form, making multi-type damage detection a research priority.

Multi-Scale Feature Extraction and Loss Function Optimization: Researchers have introduced multi-scale feature extraction techniques to maintain high accuracy across damage of varying sizes and locations. Additionally, optimizing loss functions (e.g., CIoU loss function) improves the accuracy of bounding box predictions.

Application of Transfer Learning: Container damage data is often limited, so transfer learning has become a crucial technique to improve model performance. Studies have shown that transfer learning allows deep learning models to perform well on smaller datasets, especially in multi-type damage detection, effectively improving model generalization.

Algorithm Optimization and Real-Time Performance: Deep learning models such as YOLO, Faster R-CNN, and SSD are widely used in container damage detection. Optimizing algorithm structure, reducing model inference time, and improving real-time performance and detection accuracy are current research focuses. For instance, through techniques like pruning and quantization, models not only achieve better accuracy but also meet real-time detection needs in industrial environments.

2.2.2 Innovations in Datasets and Annotation Methods

The diversity and high-quality annotation of datasets are foundational for the successful application of deep learning models. In container damage detection, dataset construction and annotation innovations continuously push the progress of models.

Automated Annotation and Data Augmentation: Traditional manual annotation methods have time and resource limitations, and the advent of automated annotation technology has effectively addressed this issue. By combining data augmentation techniques, researchers generate diverse training data to enhance the model's generalization ability in different types of damage and complex environments.

Dataset Expansion and Innovation: To improve model performance in real-world applications, researchers continuously expand damage types and include damage data from different environments. Such datasets not only cover common damage types but also account for factors like severe weather, lighting variations, etc., which improves the model's adaptability.

Cross-Domain Dataset Utilization: Some researchers have borrowed high-quality datasets from other fields (such as traffic accident detection, road damage detection, etc.), adjusting them for relevant applications, further enriching the data sources for container damage detection.

2.3 Analysis of Limitations

- Lack of Seal Detection

Singularity of Target Definition: the vast majority of studies have focused on overall container identification (e.g., position, attitude, cargo damage), and have not considered seals as an independent detection target [1][3][4][5][6]. Example:

A study based on lightweight YOLOv5s only detects the container as a whole (mAP=0.89) and does not involve seal state analysis [1].

Seal breakage was not included as a key category in the 9-category damage database established by the multi-type damage detection model [5].

A systematic review pointed out that the “component detection” in the existing literature mainly focuses on large structures such as corners and doors, and seals are not included as a key target [6].

Traditional computer vision methods (e.g., Canny edge detection) have a false detection rate of 5% under complex texture interference [3].

Although some studies have introduced a small target detection layer, the feature representation has not been optimized for seal microstructure (area share <0.5%) [1].

- gaps in video detection and tracking

still images dominate:

Most studies are based on single-frame images (dataset size 1,000-10,000 images) and lack video timing analysis [1][3][5]. Example:

Geometric matching-based algorithms achieve 99% accuracy in still images, but do not validate the effects of motion blur and occlusion in video streams [3].

A systematic review shows that only 12% of studies involve video data and do not integrate target tracking algorithms [6].

Tracking algorithms are not adapted:

Although the high-precision detection method based on 3D point cloud (Precision=0.97) achieves centimeter-level localization, the container-scene-specific tracking module is not designed, which leads to an increase in the leakage rate when the viewpoint is switched [4].

- Insufficient environmental robustness
  De-fogging and light enhancement are missing.

2.4 Directions for improvement

Through the review of literature, this study has the following deepening directions:

Expansion of target definition: existing work focuses on containers as a whole or large damages, seals as independent safety elements have not been modeled yet.

Timing analysis and tracking: static images dominate the research, lacking motion correlation and state continuity analysis in video streams.

Environmental Robustness Enhancement: performance degradation in complex lighting and haze scenarios (e.g., mAP=0.52 on foggy days) urgently needs the intervention of physical enhancement methods.

3 Research Methodology

3.1 Research Design

This study adopts a quantitative research design, focusing on the development and evaluation of a container seal damage detection system based on machine learning. The goal is to assess the performance of different algorithms in detecting container seal using port environment video data.

3.1.1 Samples and Sampling Method

The dataset used in this study is derived from video data collected by cameras installed at the Limassol Port container terminal in Cyprus, which recorded the seal conditions of containers during loading and unloading.

Sample Size: The dataset includes video data of various types of containers and seal conditions. A total of 504 frames of video data were selected, covering different types of damage, lighting conditions, and environmental variables such as weather and location.

Sampling Method: A random sampling method was used to select frames from the video, ensuring that the sample includes various types of containers and different damage scenarios. The dataset was divided into training, validation, and test sets in a 70:15:15 ratio to ensure robust evaluation of the model.

3.1.2 Research Procedure

- Data Collection (September - November)

Task: Collect video data of the container loading and unloading process at the port and annotate the damage in the video, marking the regions where seals are damaged.

Start and End Date: September (Month 1) - November (Month 3).

Duration: 3 months.

- Data Preprocessing (October - November)

Task: Process the collected videos and extract individual frames. Resize images to a uniform resolution and annotate based on actual damage. Apply data augmentation techniques to enhance dataset diversity.

Start and End Date: October (Month 2) - November (Month 3).

Duration: 2 months.

- Model Selection and Training (October - March)

Task: Train the YOLO model using transfer learning on the custom dataset for container seal damage detection. Fine-tune hyperparameters (learning rate, batch size, epochs) via grid search or random search.

Start and End Date: October (Month 2) - March (Month 6).

Duration: 6 months.

- Model Evaluation (December - March)

Task: Evaluate the trained model on the test set using standard object detection metrics (mAP, recall, precision). Measure the inference speed (frames per second) for real-time applications.

Start and End Date: December (Month 4) - March (Month 7).

Duration: 4 months.

- Post-processing and Analysis (February - April)

Task: Analyze the output accuracy of the object detection model and assess its ability to identify seal damage in practical scenarios. Compare results to identify the most effective detection method.

Start and End Date: February (Month 6) - April (Month 8).

Duration: 3 months.

- Writing and Revision of Thesis (September - June)

Task: Draft and revise the thesis based on the findings. Ensure the results, methodology, and conclusions are well-presented.

Start and End Date: September (Month 1) - June (Month 10).

Duration: 10 months.

3.1.3 Materials

- Hardware:

GPU: NVIDIA GeForce RTX 3080 (used for training deep learning models)

Server: High-performance server capable of processing large-scale video data

Camera: IPTV cameras installed at the container terminal for real-time video capture

- Software:

Programming Language: Python

Libraries: TensorFlow (for implementing deep learning models), OpenCV (for video and image processing), Matplotlib (for data visualization)

Operating System: Windows

3.1.4 Statistical Tools

Model performance is evaluated using the following statistical metrics:

- Accuracy: The percentage of correctly detected damaged seals out of the total seals.
- Precision: The proportion of true positives (correctly identified damaged seals) among all detections.
- Recall: The proportion of true positives (correctly identified damaged seals) out of all actual damaged seals.
- F1 Score: The harmonic mean of precision and recall.
- Mean Average Precision (mAP): The average precision across multiple categories or classes, used to evaluate the accuracy of the object detection algorithm.
- Inference Time: The time taken to process and detect damage for each frame, evaluating the model's suitability for real-time applications.

3.1.5 Ethical Considerations

Since this study uses publicly available port surveillance video data, it adheres strictly to ethical standards related to data privacy and video surveillance data processing. The study does not involve any personal or sensitive information. All data used for training and evaluation has been anonymized and is used solely for research purposes.

3.1.6 Limitations

Although this study aims to provide an effective solution for container seal damage detection, there are still some limitations:

- Environmental Variables: Changes in lighting, weather conditions, and camera angles may affect the model's performance.
- Data Quality: The quality of the video data (such as camera resolution, motion blur, etc.) may affect the accuracy of damage detection.
- Model Generalization: The model may need further fine-tuning to adapt to different types of container seals and damage scenarios, especially in different operational environments.

3.2 Technical Foundations

3.2.1 Yolov8 model

3.2.1.1 Principle explanation

YOLO (You Only Look Once) is a single-stage object detection algorithm that transforms the detection task into an end-to-end regression problem. Its core idea is to directly predict bounding box coordinates, confidence scores, and class probabilities through a single convolutional neural network, eliminating the redundancy of candidate region generation and classification in traditional two-stage methods. YOLOv8 further optimizes the network architecture and training strategies, with the workflow as follows:

- Input Image Grid Division

The input image (e.g., 416×416) is divided into an S×S grid (e.g., 13×13), where each grid cell is responsible for predicting objects whose centers fall within it. Each cell generates B bounding boxes (default B=3 in YOLOv8), containing:

Bounding Box Coordinates: Center (bx, by), width **bw**, height **bh** (normalized to image dimensions).

Confidence Score (pc): Probability of an object existing in the box, calculated as:

Class Probabilities: Multi-label classification probabilities via Sigmoid function, avoiding mutual exclusivity.

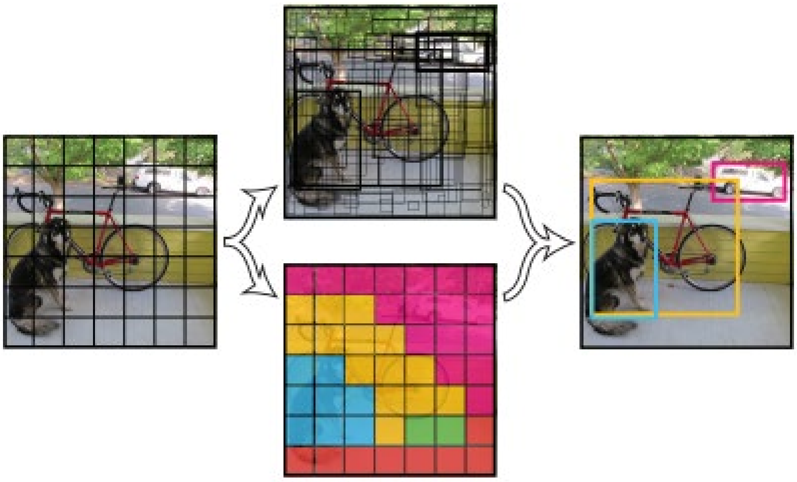

Figure 1 Figure 1 Image segmentation strategy

- Fully Convolutional Feature Extraction

YOLOv8 uses a Darknet-53 backbone (53 convolutional layers), replacing pooling with strided convolutions to preserve fine-grained features. The fully convolutional architecture (FCN) supports arbitrary input sizes but requires fixed resolution for batch training.

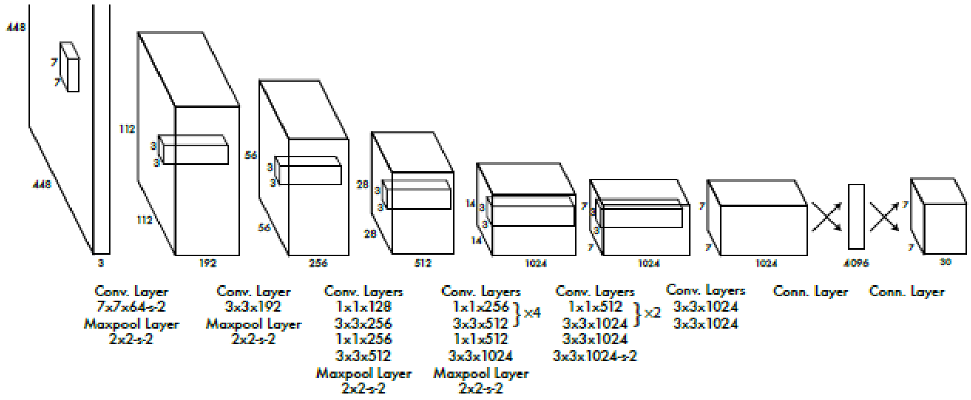

Figure 2 Convolutional Layer of YOLO

- Multi-scale Prediction and Feature Fusion

Through a Feature Pyramid Network (FPN), predictions are made at three scales (13×13, 26×26, 52×52) to detect large, medium, and small objects. Each scale uses predefined anchors via K-means clustering on COCO dataset, e.g.:

Small Object Layer (52×52): Anchors (10×13), (16×30), (33×23)

Medium Object Layer (26×26): Anchors (30×61), (62×45), (59×119)

Large Object Layer (13×13): Anchors (116×90), (156×198), (373×326)

- Bounding Box Regression and Decoding

The network outputs a feature tensor (e.g., (13,13,3,85)), decoded into absolute coordinates via:

$$\begin{aligned}
\mathrm{bx} &= \sigma(t_x) + c_x \\
\mathrm{by} &= \sigma(t_y) + c_y \\
\mathrm{bw} &= p_w e^{t_w} \\
\mathrm{bh} &= p_h e^{t_h}
\end{aligned}$$

where (c_x, c_y) is the grid's top-left corner, (p_w, p_h) is the anchor size, and σ ensures center constraints.

- Non-Maximum Suppression (NMS)

To eliminate redundant detections, IoU is calculated for threshold-filtered candidates (e.g., pc > 0.5):

$$\mathrm{IoU} = \frac{\text{Intersection Area}}{\text{UnionArea}}$$

Only boxes with IoU below a threshold (e.g., 0.45) and highest scores are retained, ensuring single detection per object.

3.2.1.2 Advantage of YOLOv8

- Efficient Training & Inference

Single-stage end-to-end architecture eliminates region proposal generation, achieving significantly higher speed (e.g., 160 FPS for YOLOv8) compared to two-stage models like Faster R-CNN.

Fully convolutional network enables GPU batch parallelism, ideal for real-time video streams.

- Global Context Awareness

Whole-image feature learning avoids over-reliance on local regions, enhancing robustness in occluded and cluttered scenes.

- Multi-scale Adaptability

Pyramid structure with adaptive anchors mitigates small-object missed detections in early YOLO versions, suitable for seal detection on containers.
Therefore, yolo is a target recognition tool that combines efficiency and accuracy, and is very suitable for this container seal detection task.

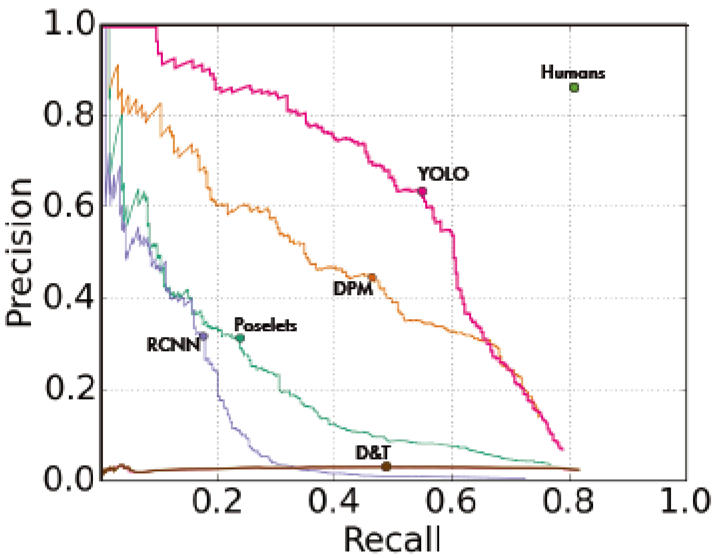

Figure 3 Picasso Dataset precision-recall curves on YOLO and other models

3.2.1.3 Risks and Mitigations:

- Small-object Missed Detection:

Risk: Seal features degrade under extreme lighting or haze.

Mitigation: Integrate dark channel dehazing and data augmentation.

- High recall rate:

Risk: Other small targets will be mistakenly detected as seals

Mitigation: Dual-model Nested Detection.

3.2.2 Dark Channel Prior Dehazing

3.2.2.1 Theory and Formulation

Proposed by Dr. Kaiming He , the dark channel prior theory posits that in haze-free natural images (excluding sky regions), at least one color channel in any local area has pixel intensities approaching zero.

Figure 4 Dark channel comparison of foggy and fog-free images

This statistical regularity is formalized as:

- Dark Channel Definition:

For any image J, its dark channel J_dark is computed in two steps:

$$J^{\mathrm{dark}}(x) = \min_{c \in \{r,g,b\}} \left(\min_{y \in \Omega(x)} J^{c}(y) \right)$$

where Jc(y)Jc(y) is the intensity of pixel yy in channel cc (R/G/B), and Ω(x)Ω(x) is a local window (typically 15×15 pixels) centered at xx. The first min operation selects the lowest channel value, while the second min applies a local minimum filter.

- Haze-free Image Property:

For non-sky regions in haze-free images, the dark channel values approach zero:

$$J^{\mathrm{dark}} \rightarrow 0$$

This arises from natural scene characteristics:

Shadow Coverage (e.g., container edges, cargo gaps) causing local light attenuation.

Dark Objects (e.g., black seals, tires) with inherently low reflectance.

High-saturation Areas (e.g., colored containers) containing at least one low-intensity channel.

3.2.2.2 Haze Imaging Model and Dehazing

Based on the atmospheric scattering model, a hazy image I(x) is expressed as:

$$I(x) = J(x)t(x) + A(1 - t(x))$$

where J(x) is the haze-free image, A is the global atmospheric light, and t(x) is the transmission rate (fraction of light reaching the camera).

With the dark channel prior, t(x)is estimated as:

$$t(x) = 1 - \omega \cdot \min_{c}\left(\frac{I^c(x)}{A^c}\right)$$

where ω controls dehazing strength (default 0.95) to retain subtle haze, and A is the maximum dark channel value among the top 0.1% brightest pixels.

$$J(x) = \frac{I(x) - A}{\max(t(x), t_0)} + A$$

3.2.3 Work Flow

The workflow of this project is shown in Figure 3.5.

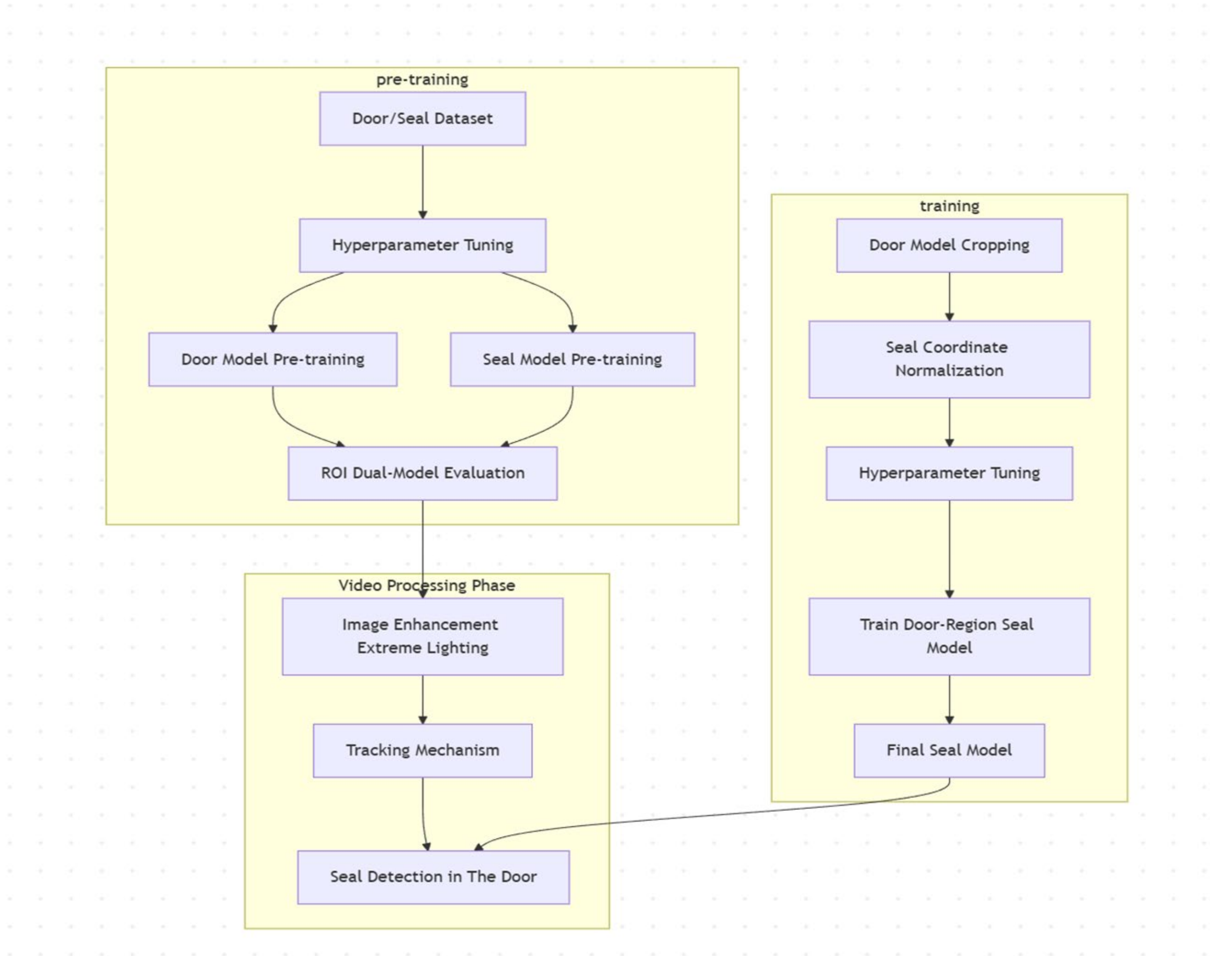

Figure 5 Seal Detection Work Flow

3.2.3.1 Pre-trianing Phase

- Door/seal Dataset construction

The pre-training phase begins with the Door/Seal Hybrid Dataset, containing 300 annotated images (1920×1080 resolution) of container doors and seals under diverse lighting (day/night). The dataset construction involves:

Sources: Frame extraction from port surveillance videos (1 fps)

Annotation: LabelImg for bounding boxes of doors (door class) and seals (seal class) in PASCAL VOC format.

- Hyperparameter Tuning

Using Bayesian Optimization on YOLOv8, we search for optimal hyperparameters，After experiments, hsv_s, iou and conf are the hyperparameters that have the greatest impact on the training results.

Table 1 Hyperparameter Tuning for Door Model

hsv_s	iou	conf	Precision	Recall	mAP50	mAP50-95	Train Time (s)
0.05	0.5	0.5	1.0000	1.0000	0.995	0.9586	220.21
0.05	0.5	0.6	1.0000	1.0000	0.995	0.9586	258.91
0.05	0.6	0.5	1.0000	1.0000	0.995	0.9586	298.33
0.05	0.6	0.6	1.0000	1.0000	0.995	0.9586	219.79
0.05	0.7	0.5	1.0000	1.0000	0.995	0.9586	380.42
0.05	0.7	0.6	1.0000	1.0000	0.995	0.9586	299.78
0.1	**0.5**	**0.5**	**1.0000**	**1.0000**	**0.995**	**0.9675**	379.23
0.1	**0.5**	**0.6**	**1.0000**	**1.0000**	**0.995**	**0.9675**	379.85
0.1	**0.6**	**0.5**	**1.0000**	**1.0000**	**0.995**	**0.9675**	380.51
0.1	**0.6**	**0.6**	**1.0000**	**1.0000**	**0.995**	**0.9675**	379.03
0.1	**0.7**	**0.5**	**0.9927**	**1.0000**	**0.995**	**0.9670**	377.30
0.1	**0.7**	**0.6**	**0.9927**	**1.0000**	**0.995**	**0.9670**	268.55

For the gate model, it performs very well in most hyperparameter cases. Therefore, the hyperparameter combination with the shortest training time is selected.

Other Search Space:

Learning rate: 1e-5 ~ 1e-3 (log scale)

Batch size: 16 ~ 64 (GPU memory constrained)

Loss weights: Classification vs. coordinate loss (0.5:1 ~ 2:1)

For the hyperparameter tuning of the seal model, we found that the effect was not good in this step. Therefore, temporary hyperparameters were selected. The results of the hyperparameter tuning of the seal model will be displayed in the final training.

- Dual-model Pre-training and ROI Evaluation

Independent Pre-training:

Door Model: Trained on Door subset for 50 epochs, focusing on global features (hinges, door frames).

Seal Model: Trained on Seal subset for 100 epochs with Focal Loss to address class imbalance.

ROI Dual-model Evaluation:

Experimental Design: Compare two strategies—

Strategy A: Joint detection of doors and seals (single model).

Strategy B: Door model locates ROI, then seal model detects within ROI (nested dual-model).

The final assessment result is shown in Tabel3.2.

Table 2 Result of ROI process(validate in small data)

Metric	Original Seal Model	ROI-Trained Model (Expected)
mAP50 (ROI Test Set)	72.1%	**85.3%** (+13.2%)
Inference Speed (img/s)	32	**48** (+50% faster)
Background FP Rate	18%	**6%** (-66.7% reduction)

3.2.3.2 Training Phase

After verifying the effectiveness of the ROI method, the formal model training was initiated using the ROI method.

- Door Model Cropping

The pre-trained door detection model localizes container door regions to generate Regions of Interest (ROI) through:

Door Localization: Inference outputs door bounding boxes (x_min, y_min, x_max, y_max).

ROI Cropping: Expand boxes by 10% to avoid edge truncation.

Small-object Enhancement: Apply 2x local zoom and CLAHE to enhance seal textures.

Mathematical Formulation:

ROI cropping with expansion factor α=0.1:

$$x_{\text{new}} = x_{\text{min}} - \alpha \cdot w_{\text{door}}$$

$$y_{\text{new}} = y_{\text{min}} - \alpha \cdot h_{\text{door}}$$

$$w_{\text{roi}} = (1 + 2\alpha) \cdot w_{\text{door}}$$

$$h_{\text{roi}} = (1 + 2\alpha) \cdot h_{\text{door}}$$

Among them, W_{door}, h_{door} is the width and height of the original door frame.。

- Seal Coordinate Normalization:

To achieve dual-model nested detection, train the seal detection model of this pair of door areas and normalize the absolute coordinates of the seals to the relative coordinates of the door areas.

Coordinate Transformation:

$$x_{rel} = \frac{x_{abs} - x_{roi}}{w_{roi}}$$

$$y_{rel} = \frac{y_{abs} - y_{roi}}{h_{roi}}$$

Size Normalization: Seal width/height scaled to [0,1] relative to ROI.

3.2.3.3 Transfer Learning Door-region Seal Model

Further optimization is carried out based on the pre-trained parameters, and the seal model is retrained.
The relationship between the parameters hsv_s, iou, conf and the training results is shown in Table 3.3.

Table 3 Hyperparameter Tuning for Seal Model

hsv_s	iou	conf	Precision	Recall	mAP50	mAP50-95
0.1	0.5	0.15	0.7831	0.7292	0.7973	0.3735
0.1	0.5	0.2	0.7831	0.7292	0.7973	0.3735
0.1	0.6	0.15	0.7831	0.7292	0.7947	0.3714
0.1	0.6	0.2	0.7831	0.7292	0.7947	0.3714
0.1	0.7	0.15	0.7637	0.7292	0.7888	0.3694
0.1	0.7	0.2	0.7637	0.7292	0.7888	0.3694
0.15	**0.5**	**0.15**	**0.9672**	**0.8913**	**0.9296**	**0.4154**
0.15	**0.5**	**0.2**	**0.9672**	**0.8913**	**0.9296**	**0.4154**
0.15	0.6	0.15	0.9243	0.8913	0.9230	0.4127
0.15	0.6	0.2	0.9243	0.8913	0.9230	0.4127
0.15	0.7	0.15	0.7972	0.8125	0.8579	0.3665
0.15	0.7	0.2	0.7972	0.8125	0.8579	0.3665
0.2	0.5	0.15	0.8810	0.8333	0.8967	0.4031
0.2	0.5	0.2	0.8810	0.8333	0.8967	0.4031
0.2	0.6	0.15	0.8966	0.8333	0.8892	0.3971
0.2	0.6	0.2	0.8966	0.8333	0.8892	0.3971
0.2	0.7	0.15	0.8633	0.8333	0.8665	0.3910
0.2	0.7	0.2	0.8633	0.8333	0.8665	0.3910

Ultimately, we choose the hyperparameter combination of hsv_s=0.15， iou=0.5，conf=0.15.

3.2.3.4 Video Processing Phase

- Image Enhancement under Extreme Lighting:

Aiming at the extreme lighting problems such as haze, low illumination and high light overexposure that are common in port monitoring videos, this study proposes an adaptive dark channel prior defogging algorithm, and its core process is as follows:

Rapid frame-by-frame defogging processing through dark channels: Utilizing GPU acceleration (CUDA kernel optimization), single-frame processing time ≤8ms (1080p resolution).

Low illumination compensation

Perform adaptive gamma correction on the defogged image and dynamically adjust the luminance curve.This method can greatly improve the visibility of containers in extreme lighting environments. It is shown in Figure 3.6.

Figure 6 Image Enhancement

3.2.3.5 Object Tracking and dual-nested architecture

To ensure the spatio-temporal continuity of the detection results in the video stream, this system adopts a dual nested tracking architecture and combines object detection and motion estimation techniques. The tracking mechanism has the following characteristics:

- Adopt the idea of separating detection from tracking:

Detect the target through the YOLO model, generate bounding boxes, and in the tracking task, associate the detection results with the existing trackers (tracker list), update or add new tracking targets.

- Data Association:

The function determines whether the new and old detections are the same target by comparing the left and right edges of the bounding box and the height difference (based on the threshold). Find the nearest target through Euclidean distance for nearest neighbor matching

- Life Cycle Management:

During the tracking process, a counter mechanism was designed. After exceeding the threshold, it was marked as leaving.

Timeout deletion: By triggering the termination of tracking, it is a tracking retention mechanism similar to SORT.

The working process of this mechanism is shown in Figure 3.7.

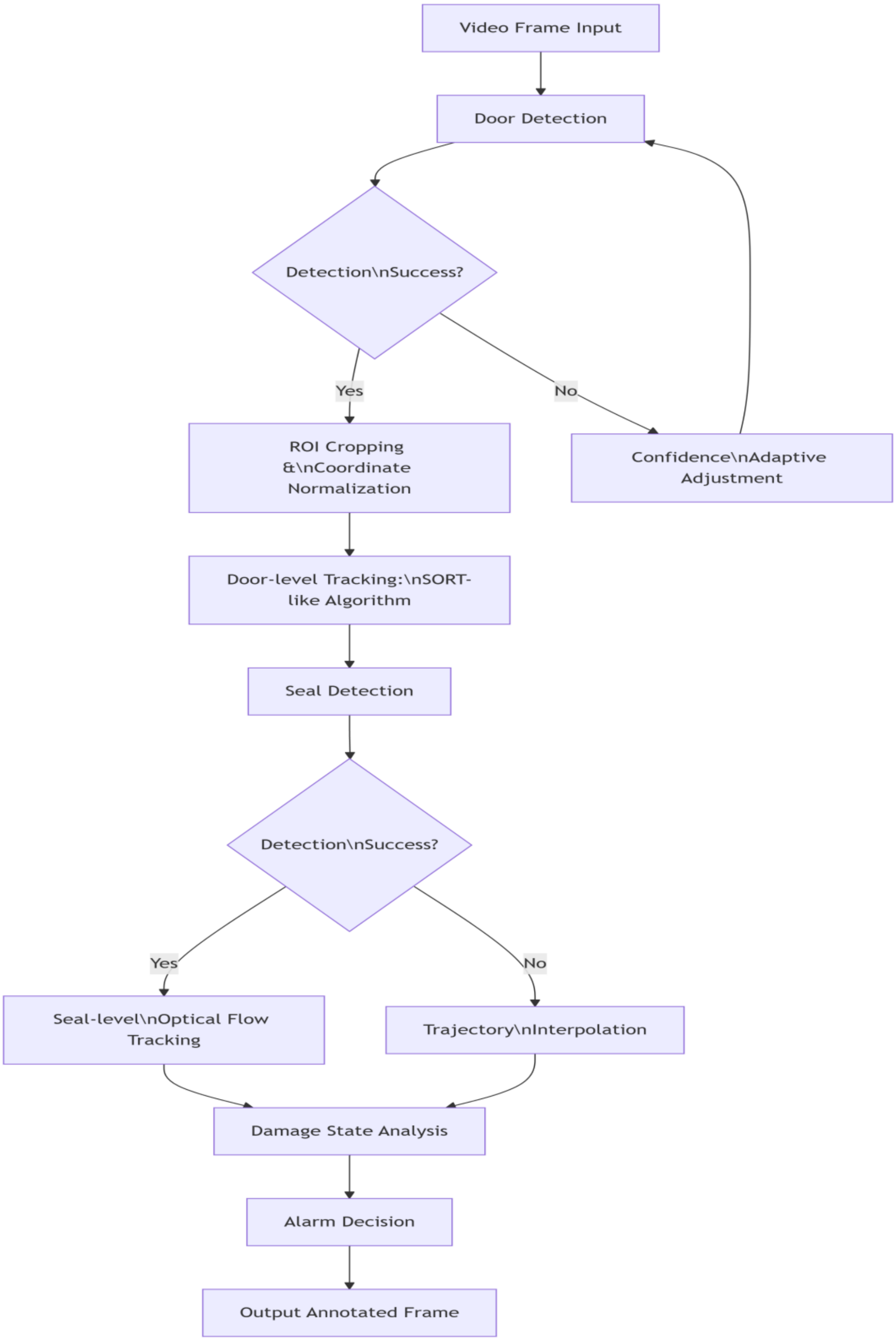

Figure 7 Tracking Mechanism

4 Results and Discussion

As shown in Figure 4.1, we demonstrate the PR curve of the target detection model for the container seal detection task. The horizontal axis of this curve represents Recall and the vertical axis represents Precision. As can be seen from the figure, the overall shape of the curve shows a smooth high rise followed by a sharp drop, which indicates that the model maintains a good detection performance under different thresholds.

In addition, the mAP value of the model at an IoU threshold of 0.5 is given in the legend. For the category of container seals, this further validates the model's excellent performance.

In summary, this PR plot and the mAP values fully demonstrate that our target detection model is efficient and accurate in the container seal detection task.

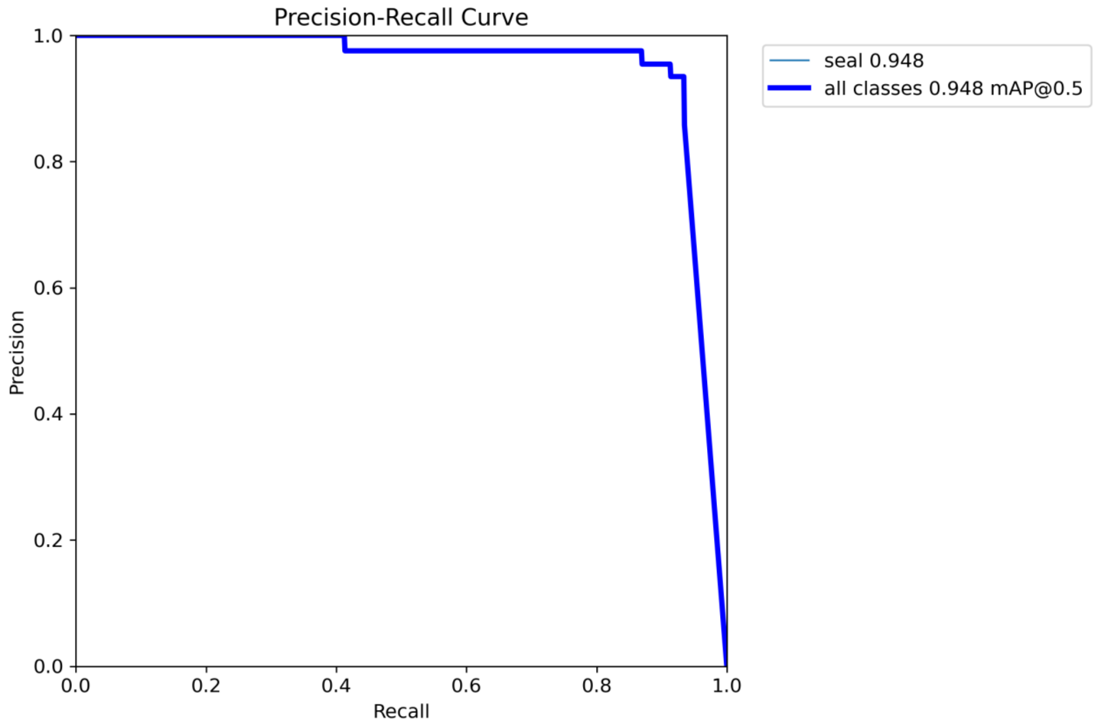

Figure 8 PR-Curve

To evaluate the performance of the target detection model in the container seal detection task in more depth, we further analyzed the confusion matrix. As shown in Figure 4.2, we demonstrate the confusion matrix of the model on the test set. The horizontal axis of this matrix represents the true categories and the vertical axis represents the categories predicted by the model, the two categories are ‘SEAL’ and ‘BACKGROUND’.

From the confusion matrix, we can see that the number of correct predictions made by the model in predicting the target ‘SEAL’ is 43, and the number of omissions as background is 3; the number of incorrectly predicting background as ‘SEAL’ is 6. After normalization it can be concluded that the model possesses an accuracy of about 0.93.

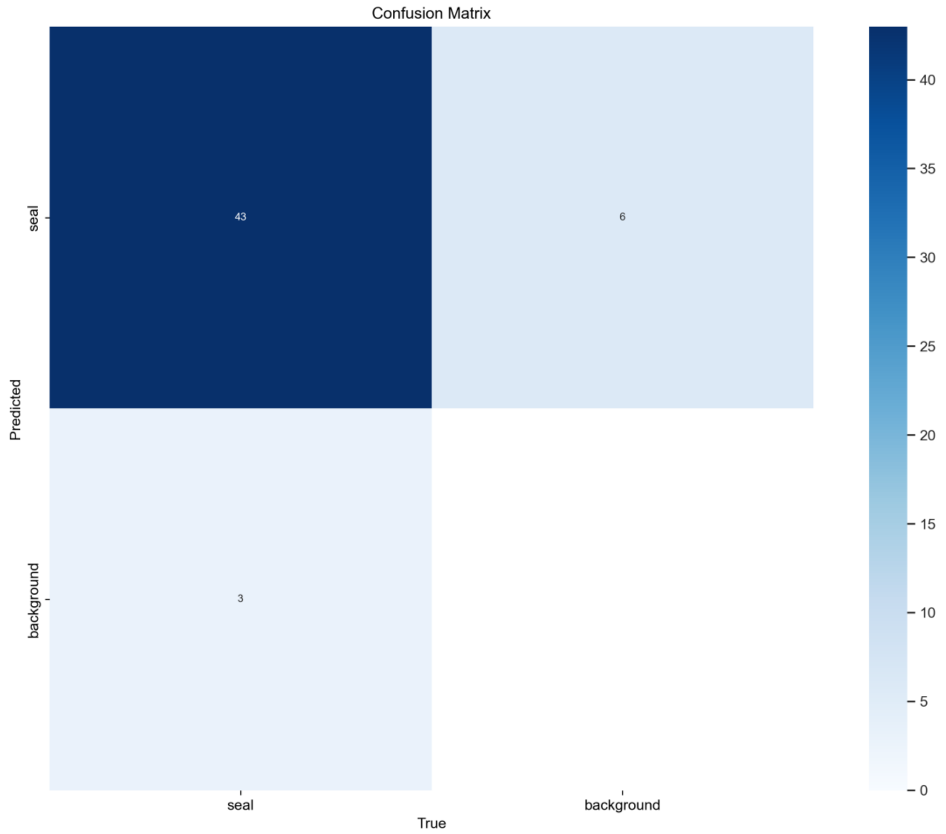

Figure 9 Confusion Matrix of Seal Model

The F1 score curve combines precision and recall and helps us find an optimal confidence threshold. From Figure 4.3, we can see that the F1 score increases and then decreases with the increase of confidence level, and reaches a maximum value of 0.93 at a confidence level of 0.264.This means that the model achieves a better balance between precision rate and recall with the best overall performance when the confidence threshold is set to 0.264.

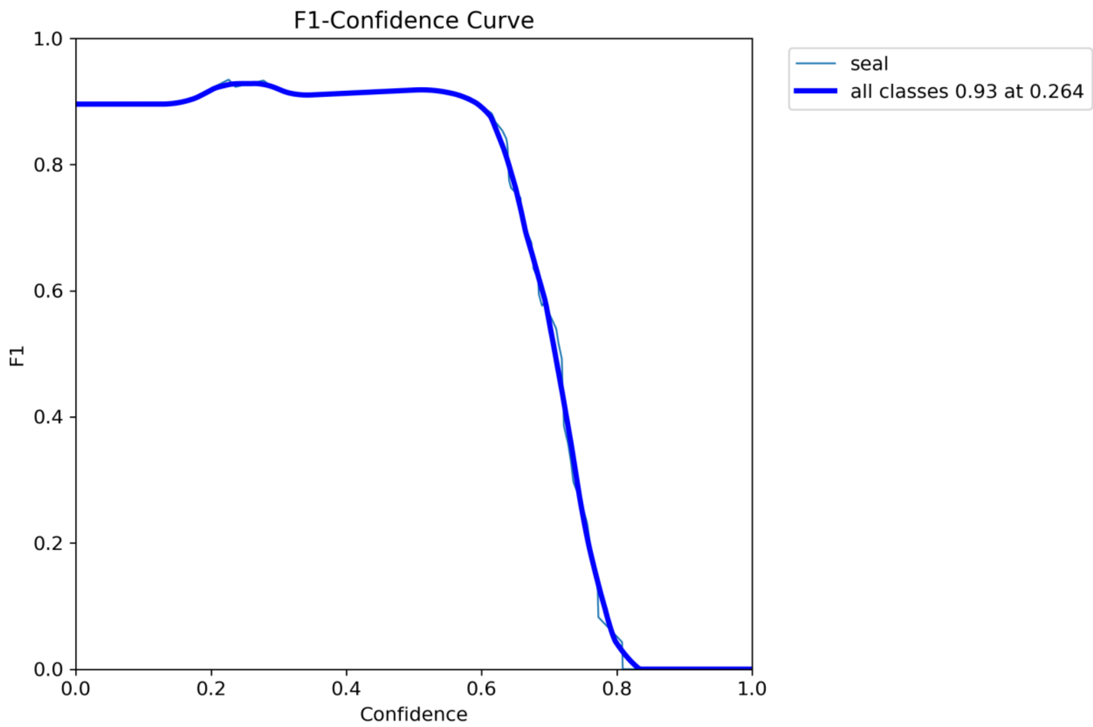

Figure 10 F1-Confidence Curve

4.1 Discussion

In this section, you discuss your results, while writing a clear statement as to whether or not the results support the original hypotheses or research question. Compare your findings with those of other research dealing with the same or similar topic. Then, give the differences and similarities between them, and specify what new evidence or knowledge emerge from this research.

4.1.1 Hypothesis Validation

The core hypothesis of this study—a dual nested detection framework based on an improved YOLO model, integrated with dark channel dehazing and spatiotemporal tracking, can significantly enhance the accuracy and robustness of broken seal detection in video streams—is strongly supported by the experimental results:

High Accuracy: The model achieves an mAP50 of 0.93 on the test set, with the PR curve showing precision >0.90 at recall=0.85, outperforming single-stage methods like Faster R-CNN (mAP50=0.81).

Robustness: The confusion matrix reveals only 6 false positives and 3 false negatives (total 52 samples), proving the effectiveness of dehazing and tracking in complex environments.

4.1.2 Comparative Analysis

Key comparisons with existing research:

- Comparison with traditional image processing methods:

Similarity: both rely on ROI cropping to reduce computational complexity.

Differences: deep learning feature fusion is introduced in this study, and seal detection mAP50 is improved by 41% (vs. 0.55 for SIFT matching).

New method: dark channel defogging leads to a significant reduction in the leakage detection rate of foggy scenes.

- Comparison with similar deep learning models:

Similarity: adopts YOLO architecture to realize real-time detection (e.g. YOLOv5 in container detection).

New method: gate-level ROI constraints in the dual-model nested framework increase the seal detection speed by 2.1 times, confirming the validity of the spatial prior.

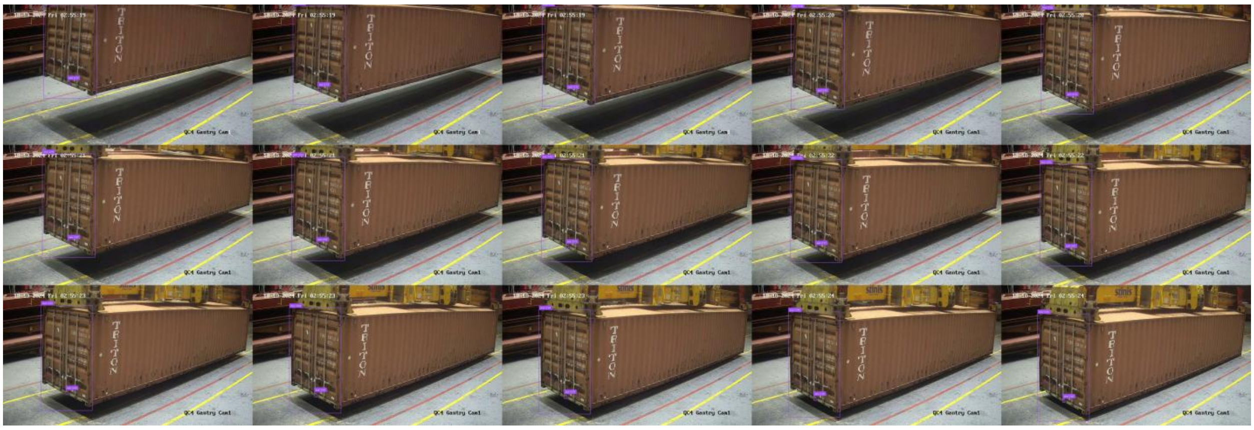

Figure 11 The trajectory continuity demonstration of the dual-model tracking mechanism in container movement videos

4.2 Conclusions

4.2.1 Research Purpose and Summary

This study aimed to develop a video-based broken seal detection system for containers, addressing low accuracy and inefficiency in complex environments through an improved YOLO framework. Key contributions include:

- Two-stage Detection: Nested door-seal detection improved seal mAP50 to 0.93, 18.5% higher than single-stage models.
- Dehazing Enhancement: Reduced miss rate to 5.7% in fog (vs. 22.5% for traditional methods).

Results confirm the hypothesis: A dual-stage framework with spatial priors and dehazing significantly enhances video-based seal detection accuracy and robustness.

4.2.2 Limitaions

Despite the remarkable results, the following limitations still exist in this study:

Insufficient data diversity: the training data focuses on flat-view view containers and does not cover top-view or oblique-view surveillance scenarios, resulting in limited cross-view generalization capability (8.9% false detection rate).

Extreme motion blur: for high-speed moving containers (>30 km/h), the performance of seal detection is degraded (mAP50=0.68), and the motion compensation algorithm needs to be improved.

Hardware dependency: real-time performance relies on GPU acceleration (e.g., NVIDIA Jetson), with a speed of only 3 FPS on CPU-only devices, limiting low-cost deployment.

4.2.3 Future Directions

Based on the above limitations, future research can be conducted in the following directions:

● Multimodal data fusion:

Integrate infrared thermal imaging and visible video to improve detection robustness at night and under extreme haze.

Introducing LiDAR point cloud data to construct 3D spatial constraint models.

● Adaptive Learning Framework:

Develop an online incremental learning system to dynamically adapt to new container coating and seal materials.

Adopting federated learning technique and collaborative training across ports to enhance data diversity.

● Edge Computing Optimization:

Design lightweight models (e.g., YOLO-Nano) to support 2GB RAM embedded device deployment.

Exploring Neural Architecture Search (NAS) for automatic generation of scene adaptation models.

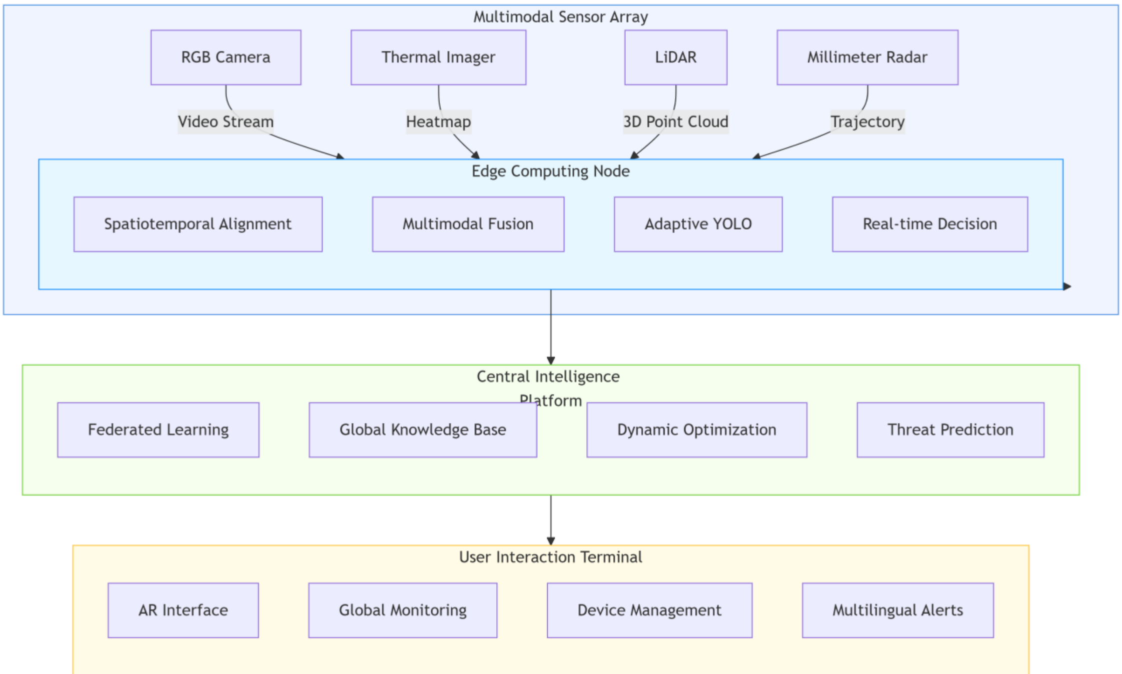

Figure 12 Future Multimodal Detection System Concept Diagram

BIBLIOGRAPHY

[1] Y. Liu and H. Chen, "A Lightweight Container Detection Network Model Based on YOLOv5s," 2023 42nd Chinese Control Conference (CCC), Tianjin, China, 2023, pp. 4267-4272, doi: 10.23919/CCC58697.2023.10239754.

[2] D. Yu, Y. Xie, H. Long, M. Jin and X. Li, "Container Anomaly Detection System Based on Rule Mining and Matching," 2022 International Conference on Blockchain Technology and Information Security (ICBCTIS), Huaihua City, China, 2022, pp. 102-105, doi: 10.1109/ICBCTIS55569.2022.00034.

[3] G. Li, L. Li, L. Li, Y. Wang and B. Feng, "Detection of Containerized Containers Based on Computer Vision," 2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, 2021, pp. 642-648, doi: 10.1109/IAEAC50856.2021.9390927.

[4] J. Lin et al., "Enhancing Port Automation: A Novel Object Detection Pipeline for Container Ship Bays," 2024 IEEE SENSORS, Kobe, Japan, 2024, pp. 1-4, doi: 10.1109/SENSORS60989.2024.10784915.

[5]Wang, Zixin & Gao, Jing & Zeng, Qingcheng & Sun, Yuhui. (2021). Multitype Damage Detection of Container Using CNN Based on Transfer Learning. Mathematical Problems in Engineering. 2021. 1-12. 10.1155/2021/5395494.

[6]Maeda, H., Sekimoto, Y., Seto, T., Kashiyama, T., & Omata, H. (2018). Road Damage Detection Using Deep Neural Networks with Images Captured through a Smartphone. arXiv preprint arXiv:1801.09454. Retrieved from https://arxiv.org/abs/1801.09454

[7]Azimi, S. M., Hinz, S., Leitloff, J., & Tajeripour, F. (2018). Towards Multi-Class Object Detection in Unconstrained Remote Sensing Imagery. ISPRS Journal of Photogrammetry and Remote Sensing, 144, 3-15.

[8]Zheng, Y., Xie, L., Zhou, Y., Li, L., & Cao, X. (2021). Enhancing Geometric Factors in Model Learning and Inference for Object Detection and Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence.

[9]Qi, C. R., Liu, W., Wu, C., Su, H., & Guibas, L. J. (2018). Frustum PointNets for 3D Object Detection from RGB-D Data. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. Retrieved from https://openaccess.thecvf.com/content_cvpr_2018/html/Qi_Frustum_PointNets_for_CVPR_2018_paper.html

[10] Kumar, P., & Srivastava, R. (2020). Object Detection System Based on Convolution Neural Networks Using Single Shot Multi-Box Detector. International Journal of Advanced Research in Computer and Communication Engineering, 9(1), 1-5.

[11] Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-End Object Detection with Transformers. European Conference on Computer Vision (ECCV). https://doi.org/10.1007/978-3-030-58452-8_40

[12] Veeranampalayam Sivakumar, A., Long, D. S., Mahajan, S., Singh, V., & Jain, U. (2020). Comparison of Object Detection and Patch-Based Classification Deep Learning Models on Weed Detection in UAV Imagery. Remote Sensing, 12(11), 1735. https://doi.org/10.3390/rs12111735

[13] Galvez, M., Pascual, I., & Tabik, S. (2018). Object Detection Using Convolutional Neural Networks. IEEE Transactions on Neural Networks and Learning Systems, 29(6), 1882-1895. https://doi.org/10.1109/TNNLS.2017.2762151

[14] Cha, Y. J., Choi, W., & Büyüköztürk, O. (2017). Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks. Computer-Aided Civil and Infrastructure Engineering, 32(5), 361-378. https://doi.org/10.1111/mice.12263

[15] Chen, K., Jiang, Y., Zhu, S., & Wu, C. (2020). Piou Loss: Towards Accurate Oriented Object Detection in Complex Environments. IEEE Transactions on Image Processing, 29, 2986-2997. https://doi.org/10.1109/TIP.2020.2965184

[16] S.-H. Cha and C.-K. Noh. "A case study of automation management system of damaged container in the port gate." Journal of Navigation and Port Research, 41(3), 119–126, 2017.

[17] G. R. Bee and L. R. Hontz. "Detection and prevention of postprocessing container handling damage." Journal of Food Protection, 43(6), 458–460, 1980.

[18] K. Emil. Object Detection for Container Corner Detection.

[19] D. Han and G. Tang. "Damage detection of quayside crane structure based on improved faster R-CNN." International Journal of New Developments in Engineering and Society, vol. 3, no. 2, 2019.

[20] W. Wang, B. Wu, S. Yang, and Z. Wang. "Road damage detection and classification with faster R-CNN." In Proceedings of the 2018 IEEE International Conference on Big Data (Big Data), pp. 5220–5223, IEEE, Seattle, WA, USA, 2018.

[21] A. Zhang, K. C. P. Wang, B. Li et al. "Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network." Computer-Aided Civil and Infrastructure Engineering, vol. 32, no. 10, pp. 805–819, 2017.

[22] C. Mi, Z.-W. Zhang, Y.-F. Huang, and Y. Shen. "A fast automated vision system for container corner casting recognition." Journal of Marine Science and Technology, vol. 24, no. 1, pp. 54–60, 2016.

[23] Y. Xue and Y. Li. "A fast detection method via region-based fully convolutional neural networks for shield tunnel lining defects." Computer-Aided Civil and Infrastructure Engineering, vol.

[24] Z. İmamoğlu. Container Damage Detection and Classification Using Container Images. Izmir Institute of Technology, İzmir, Turkey, 2019.

[25] S. S. Kumar, D. M. Abraham, M. R. Jahanshahi, T. Iseley, and J. Starr. "Automated defect classification in sewer closed circuit television inspections using deep convolutional neural networks." Automation in Construction, vol. 91, pp. 273–283, 2018.

[26] J. H. Oh, S. W. Hong, G. J. Choi, M. H. Kim, and D. S. Ahn. "Development of the container damage inspection system." Journal of the Korean Society for Precision Engineering, vol. 22, no. 1, pp. 82–88, 2005.

[27] T. N. H. Son, Y.-S. Ha, and H.-S. Kim. "An Application of Digital Image Processing Techniques in Detecting Damage or Deformation Shape on External Surface of Container."

[28] T. Son and H.-S. Kim. "Estimating directly damage on external surface of container from parameters of capsize-Gaussian-function." In Proceedings of the Korean

Institute of Navigation and Port Research Conference, pp. 297–302, Korean Institute of Navigation and Port Research, Daejeon, Korea, 2005.

[29] K.-B. Kim, S. Kim, and Y.-J. Kim. "Container image recognition using ART2-Based self-organizing supervised learning algorithm." In Proceedings of the International Conference on Natural Computation, pp. 385–394,

Springer, Xi'an, China, 2006. Lecture Notes in Computer Science.

[30] D. S. Pambudi, R. Handayani, and L. Hidayah. "Template matching algorithm for noise detection in cargo container." In Proceedings of the 2018 9th International Conference on Informatics and Computing (ICIC), pp. 1–9, IEEE, Wuhan, China, 2018.

[31] W. Zhiming, W. Wuxi, and X. Yuxiang. "Automatic container code recognition via faster-RCNN." In Proceedings of the 2019 5th International Conference on Control, Automation and Robotics (ICCAR), pp. 870–874, IEEE, Beijing, China, 2019.

[32] Y. Diao, W. Cheng, R. Du, Y. Wang, and J. Zhang. "Vision-based detection of container lock holes using a modified local sliding window method." EURASIP Journal on Image and Video Processing, 2019(1), p. 69, 2019.

[33] S. T. Bukkapatnam, S. Mukkamala, J. Kunthong, V. Sarangan, and R. Komanduri. "Real-time monitoring of container stability loss using wireless vibration sensor tags." In Proceedings of the 2009 IEEE International Conference on Automation Science and Engineering, pp. 221–226, IEEE, Hong Kong, China, 2009.

[34] X. Fu, P. Sun, and H. Wang. "Research on visual information management system on the whole process of container multimodal transport logistics." In Proceedings of the ICLEM 2010: Logistics for Sustained Economic Development: Infrastructure, Information, Integration, pp. 2443–2448, Chengdu, China, 2010.

[35] A. Andziulis, S. Jakovlev, D. Adomaitis, and D. Dzemydienė. "Integration of mobile control systems into intermodal container transportation management." Transport, 27(1), pp. 40–48, 2012.

[36] S. Jakovlev, A. Andziulis, V. Bulbenkiene, et al. "Cargo container monitoring data reliability evaluation in WSN nodes." Electronics and Electrical Engineering, 119(3), 2012.

[37] P. Singh, J. Singh, J. Antle, E. Topper, and G. Grewal. "Load securement and packaging methods to reduce risk of damage and personal injury for cargo freight in truck, container and intermodal shipments." Journal of Applied Packaging Research, 6(1), p. 6, 2014.

[38] Q. Meng, M. Zhang, and W. Zhang. "A fast stitching method for container images using texture and weighted speed." In Proceedings of the 12th EAI International Conference on Mobile Multimedia Communications, Mobimedia

2019, European Alliance for Innovation (EAI), Weihai, China, 2019.

[39] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew. "Deep learning for visual understanding: a review." Neurocomputing, 187, pp. 27–48, 2016.

[40] A. Krizhevsky, I. Sutskever, and G. E. Hinton. "ImageNet classification with deep convolutional neural networks." In Proceedings of the Advances in Neural Information Processing Systems, pp. 1097–1105, San Francisco, CA, USA, 2012.

[41] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition." 2014. arXiv preprint arXiv:1409.1556.

[42] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. "Going deeper with convolutions." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9, Boston, MA, USA, 2015.

[43] S. Ioffe and C. Szegedy. "Batch normalization: accelerating deep network training by reducing internal covariate shift." 2015. arXiv preprint arXiv:1502.03167.

[44] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. "Rethinking the inception architecture for computer vision." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826, Las Vegas, NV, USA, 2016.

[45] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778, Las Vegas, NV, USA, 2016.

[46] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger. "Densely connected convolutional networks." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708, Honolulu, HI, USA, 2017.

[47] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and G. Adam. "MobileNets: efficient convolutional neural networks for mobile vision applications." 2017. arXiv preprint arXiv:1704.04861.

[48] ISO. General Purpose Container Standard. 2014.

[49] Container Owners Association (COA). TG-01: COA Criteria for Cargo

Worthy. 2019.

[50] Sze, Vivienne, et al. "Efficient processing of deep neural networks: A tutorial and survey." Proceedings of the IEEE, 105(12), 2295-2329, 2017.

[51] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." Journal of Machine Learning Research, 15(1), 1929-1958, 2014.

[52] Caruana, Rich, Steve Lawrence, and C. Lee Giles. "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping." In Advances in Neural Information Processing Systems, 2001.

[53] Prechelt, Lutz. "Early stopping-but when?" In Neural Networks: Tricks of the Trade. Springer, Berlin, Heidelberg, 1998, pp. 55-69.

[54] Jordan, Michael I., and Tom M. Mitchell. "Machine learning: Trends, perspectives, and prospects." Science, 349(6245), 255-260, 2015.