Automating Construction Safety Inspections using Robots and Unsupervised Deep Domain Adaptation by Backpropagation

Vimal Bharathi [1], Samuel A. Prieto [2], Borja Garcia de Soto [2], Jochen Teizer [1]

[1] Department of Civil and Mechanical Engineering, Technical University of Denmark
[2] S.M.A.R.T. Construction Research Group, Division of Engineering, New York University Abu Dhabi
vvebh@dtu.dk, samuel.prieto@nyu.edu, garcia.de.soto@nyu.edu, teizerj@dtu.dk

Abstract

Due to the dynamic aspect of construction sites, constant implementation and removal of safety equipment is a required practice. This leads to frequent manual and time-consuming inspections to make sure the safety measures are in place. There is the potential to automate the inspection process using robots and Deep Learning. Such an approach can save time and cost while improving safety. Using images collected by an Autonomous Ground Vehicle, a Deep Learning model with Domain Adaptation techniques is trained to detect and segment safety guardrails. The results of the model indicate a promising method to assist in automating site safety inspection that can make construction sites safer. Further work is necessary to validate this effort under more realistic and harsh construction site conditions.

Keywords –

Construction safety measures, deep learning, domain adaptation, prevention through design and planning, site inspection, robot, YOLOv8.

1. Introduction

In the dynamic realm of construction safety, technological advancements are reshaping traditional practices. The construction industry stands out as one of the riskiest sectors due to the ever-changing work environment. Ensuring the safety of construction workers is a vital aspect of a construction business, leading to the development of comprehensive regulations and guidelines. These measures are designed to protect workers during construction activities. Given the demanding nature of crafting a secure construction plan, it remains a top priority. Consequently, substantial time and effort are dedicated to promoting the health and well-being of workers, preventing fatalities, severe and minor injuries, and close-call incidents, also known as prevention through design (PtD) [1].

An analysis of industries and their associated hazards in the UK [2] reveals that the private construction sector is the second leading sector in workplace injuries and fatalities. A recent report summarizing construction safety statistics for 2023 shows that falls make up 35% of fatalities in construction sites [3]. This underscores the significance of the focus on fall hazards and, more specifically, protective guardrails.

Another critical aspect of ensuring safety in construction operations involves inspecting and pinpointing missing or subpar safety equipment, such as protective guardrails. Inspecting collective safety equipment is also a labor-intensive task, given the dynamic nature of construction sites. Therefore, inspections must occur frequently. Additionally, the actual installation of safety measures often deviates from the intended quality outlined in digital models.

Hence, in this study we propose a preliminary stage of a more automated inspection that improves the current human-based solutions onsite. The data collection process is based on the work by Gopee et al. [4], which uses the existent BIM to generate waypoints of interest for the navigation of an Autonomous Ground Vehicle (AGV) to collect images (e.g., RGB information) as it traverses the designated areas. The collected data can be processed and used to train Deep Learning models designed for real-time object detection of the elements of interest, such as protective guardrails. In this study, a YOLOv8 network enhanced by a Domain Adaptation technique is used. The differences in construction sites due to factors like weather, sunlight, and location result in numerous scenarios for the deep learning model to analyze and predict outcomes. Each of these cases could be called a "Domain," as the images from these cases could have different features for the model to identify and learn. Due to the lack of available data and time, the technique of domain adaptation has been used to achieve the maximum possible results in predicting the guardrails with less unlabeled data from any target domain. The presented model provides a binary classification (i.e., safe or unsafe) as an output for the guardrails detected. Evaluation metrics such as Precision, Recall and Mean Average Precision (mAP) are used to quantify the improvements in detection capabilities and choose the right model configuration.

The rest of the paper is structured as follows. Section

2 provides a background on segmentation in the construction field. Section 3 presents the methodology used for this study. Section 4 uses a case study to illustrate the main elements of the methodology. Section 5 summarizes key results from the case study and, finally, Section 6 includes the conclusions and future work.

2. Background

There have been many previous works on the usage of different methods and models for the detection of guardrails using data in the form of construction images, synthetic images, and point cloud data (from laser scanners or photogrammetry).

Kolar et al. [5] used synthetic data for training a model with VGG16 as the feature extractor. The construction images with guardrails were used as the validation dataset, while augmented synthetic data was used for training the model. Their method had a 96.5% accuracy but failed in cases with low-light images of guardrails, which shows that the model would work poorly in domain change (i.e., changes in the physical conditions of the environment) situations.

The lack of data available for training an effective and efficient Deep Learning model is challenging. The construction sector is widely diverse, with different types of visual features around the world, including different seasons and weather conditions. Each difference is a different domain of data, which presents an extra challenge to train Deep Learning models to work in all these different conditions. To overcome this, there is a need for a technique that could help to train a generalized Deep Learning model, to work in multiple domains and with low amounts of data by applying Transfer Learning. Ganin et al. [6] showed that domain adaptation could be used to learn Domain Invariant features, which could predict and segment objects across different domains with few or no labeled target domain data. Using the unlabeled target domain data in the training process is called "Unsupervised Domain Adaptation".

Li et al. [7] employed a Domain Adaptation Technique in combination with the YOLOv5 architecture [8], applying it separately to the source and target domains. In this approach, they used CSPDarknet-53 [9] as the feature extractor, which forms the backbone, neck, and head components of the YOLOv5 structure. In their study, a significant amount of source labeled data was used, along with small amounts of target labeled data, to train the YOLOv5 pipeline. Features from 3 different resolutions from the backbone of the source and target pipeline were used to calculate the transfer loss (Maximum Mean Loss). The transfer loss shows the distance between the features extracted by both the source and target data, which is added to the overall loss function as a regularization term. The drawback is that the sample target data that has been used for training has to be labeled, which is a labor-intensive task given the number of different domains.

The performance of YOLOv8 on specific datasets [10] and its anchor-free detection technique that increases the model's ability to detect objects of various shapes and sizes without the constraints imposed by predefined anchors has made it an ideal choice for our task. This work shows how the method of Unsupervised Domain Adaptation with backpropagation [6] can be used with a YOLOv8 [11] model to improve the detection and segmentation of objects with labeled source domain and unlabeled target domain datasets.

3. Methodology

The overall process used in this study is shown in Figure 1. It can be divided into two sections: (1) data collection and (2) data processing/model development. For the data collection, given a BIM of the environment, it is possible to extract a set of waypoints for an AGV to stop and collect data. While the AGV autonomously moves towards the waypoints, it can collect data with multiple sensors (i.e., an RGBD camera and a 360 camera). This study focuses on data processing and model development. For more information about data collection, readers are referred to previous work by the authors (e.g., [12-14]).

3.1 Data Pre-processing and Labeling

To build a Deep Learning model, a set of training data needs to be used. Different labeling tools can be used for the collected data.

3.1.1 Data Augmentation

In the case when the amount of training data is not enough to ensure good results in the segmentation stage, data augmentation can be used to increase the size of the training dataset. Once the data has been labeled, various types of data augmentation are used during the training process, which greatly increases the chances for the model to explicitly learn the guardrail features.

3.2 Build Deep Learning Model

3.2.1 Deep Learning Network

Deep-feedforward architectures have brought significant advances to state-of-the-art models across a wide variety of machine-learning tasks and applications.

3.2.2 Deep Domain Adaptation

A Deep Learning architecture trained on one to work for another domain with a shift in the distribution needs features that are common to both domains. Learning a label classifier in the presence of a shift between source and target distribution is known as domain adaptation. There are several methods to perform Domain Adaptation (DA). Here, we use DA by backpropagation.

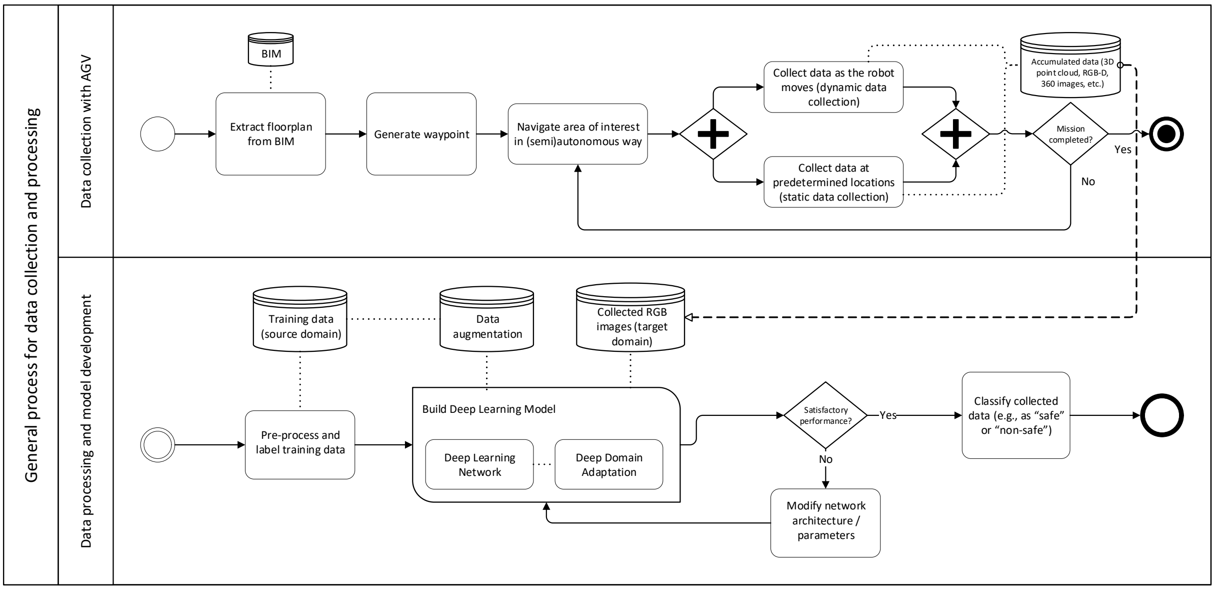
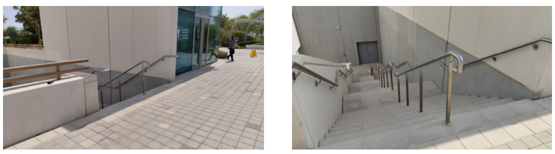


Figure 1: Main elements of the proposed methodology for data collection, processing, and model development.

4. Case Study

The proposed methodology has been tested on a condition with an abrupt change in elevation, representing conditions that could lead to fall hazards if proper protection was not in place during a construction project. The scenario used was a staircase on a university campus (Figure 2). A mock-up of a fall protection guardrail and an AGV equipped with different sensors were used to collect data.

The outdoor experiment consisted of different case studies, considering all the possible cases that could be present in a real construction site. These cases consider the proper installation of the guardrail (with all the elements installed correctly), missing elements of the guardrail (i.e., missing mid and toe board), and the presence of clutter that could potentially be a trip hazard.



(a) (b)

Figure 2: (a) Overall view of the scenario used for the experimentation, and (b) view of the stairs representing the fall hazard.

4.1 Fall Protection Guardrail Mock-up

A wooden mock-up was built to resemble the most common safety guardrails typically used in construction sites as fall protection [15]. It was built according to the requirements specified by the Employer's Liability Insurance Association for the Construction Industry in Germany (BG BAU) [16]. The mock-up consisted of a modular system with three vertical poles (1m height) with top, mid and toe boards (in total 6 horizontal boards of 20cm height and 1.5m width). An overview of the mock-up is shown in Figure 3.

Figure 3: View of fall protection guardrail mock-up developed and used for this study.

4.2 Autonomous Ground Vehicle (AGV)

The AGV used was a SUMMIT-XL platform by Robotnik Automation. The robot has holonomic locomotion (i.e., mecanum wheels). This allows the robot to move in all directions, providing a more accurate and reliable data collection in highly dynamic environments such as construction sites, where narrow passages are common. The AGV is shown in Figure 4, and a recording of the AGV collecting data can be watched in [17].

In terms of sensors for the data collection, the robot is equipped with a mid-range 3D scanner (BLK360) suitable for high-resolution dense point cloud acquisition, a long-range LiDAR (OUSTER OS1) suitable for low-resolution point cloud acquisition used for the navigation, an RGB-D camera (Orbbec Astra) used to collect both RGB and depth information of the robot front view, and a 360 camera (GoPro MAX 360) aimed to collect extra RGB data surrounding the robot as it moves through the environment. Key characteristics of the different equipment used are summarized in Table 1.

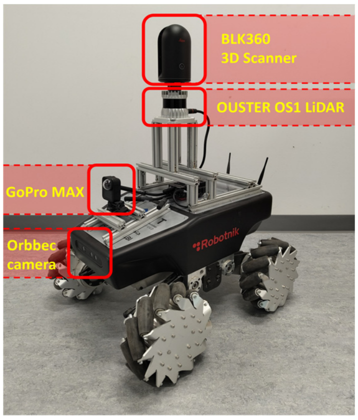


Figure 4: AGV and related payload used.

Table 1: Key specifications of the sensors used.

Sensor	Key specifications
Leica Geosystems BLK360 (Laser scanner)	Accuracy: 6mm@10m Range: 60m Speed of capture: 360,000 pts/s FOV: 360ºH x 300ºV
OUSTER OS1 (LiDAR)	Accuracy: 2.5cm@10m Range: 170m Speed of capture: 2,621,440 pts/s FOV: 360ºH x 45ºV
Orbbec Astra (RGBD camera)	Accuracy: 3mm@1m Range: 0.6m-8m Image resolution: 640x480 @30fps FOV: 60°H x 49.5°V x 73°D
GoPro MAX (Action camera)	Image resolution: 4992 x 2496 x 360º

4.3 Conditions Evaluated

4.3.1 Full Guardrail (Safe Condition)

For this condition, the ideal scenario is tested. This means that the guardrail contains all the horizontal boards, fulfilling all the safety standards (Figure 5).

(a) (b)

Figure 5: (a) View of the mock-up placed on the stairs entrance, and (b) AGV collecting data.

4.3.2 Missing Boards (Unsafe Condition)

For this condition, multiple boards of the guardrail are missing. This reflects a potential fall hazard since the space between the top board and the floor is wide enough for a person to fall through (Figure 6). In addition, multiple objects (clutter) were added, presenting additional trip hazards that can potentially lead to a worker falling through the faulty installation. These elements also present occlusion for the Deep Learning detection algorithm, ensuring that the approach is tested under non-ideal conditions.

(a) (b)

Figure 6: (a) View of the scenario where the mock-up misses the mid and toe boards and has clutter on the floor, and (b) same condition with only vertical poles installed.

4.4 Data Pre-processing and Labeling

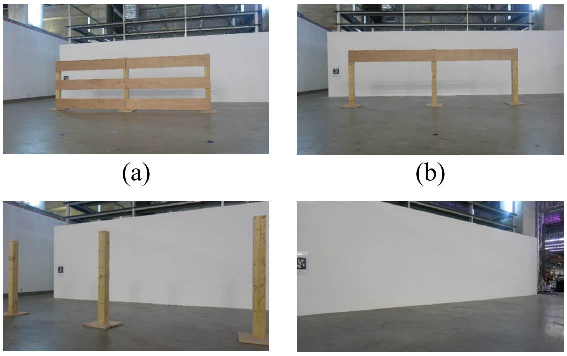
To train the Deep Learning model, RGB data was collected manually in an indoor environment with controlled lighting conditions (source domain). The training data was labeled using "Label Studio", an open-source data labeling tool [14]. For the specifics of a segmentation model, the labeled data needs to be in the form of masks (i.e., vertices of a polygon surrounding the segmented object). To further enhance the robustness and performance of the Deep Learning model under the dynamic conditions of the construction site, Unsupervised Deep Domain Adaptation is also used with indoor training images as the source domain and the outdoor collected images as the target domain. The target domain does not require any data processing or augmentation.

4.4.1 Data Collection

The data collected can be split into two categories: indoor and outdoor. To train the YOLOv8 network, a set of pictures of the fall protection guardrail taken in an indoor controlled environment was used. The source domain corresponds to the labeled training indoor data, and the target domain corresponds to the sample unlabeled data that needs to be classified. The scenarios were classified into safe (Figure 7 (a)) and unsafe (Figure 7(b-d)).

4.4.2 Source Data Collection

A total of 56 images in the indoor setting were taken. The source images were split into training and validation images with a ratio of 3:1. The pictures were taken from different points of view and two sets of distances (Figure 8) and different cases (i.e., all horizontal boards, only the top board, only the vertical poles, and with no guardrail) (Figure 7) to collect as many features as possible.



(a) (b)

(c) (d)

Figure 7: Source images used for training with guardrails: (a) full board, (b) top board and vertical poles, (c) poles only, and (d) none present.

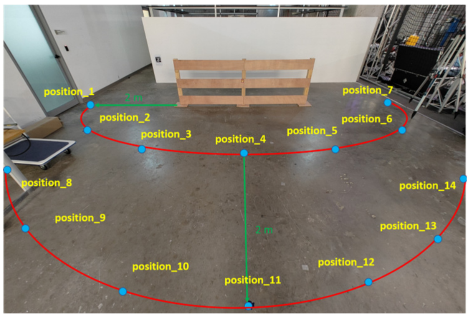


Figure 8: Setup for indoor data (images) collection.

4.4.3 Target Data Collection

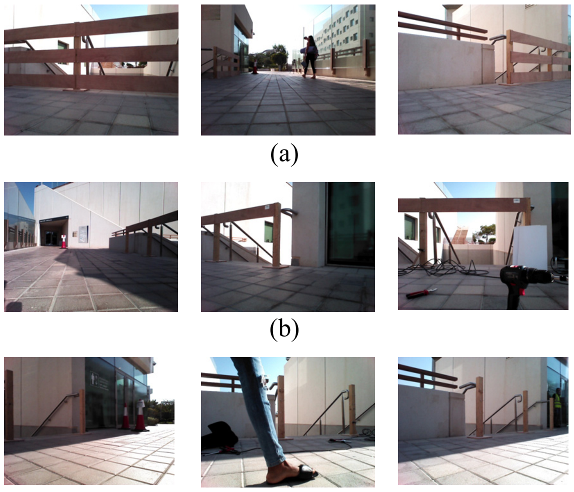
The outdoor images, which correspond to the target domain, were collected with the AGV in different conditions. Examples of the collected data with the AGV are shown in Figure 9.

4.4.4 Data Augmentation

Since the amount of training data was not enough to train a robust object detection model, data augmentation was used to increase the training data tenfold. Continuous and random selection of various techniques of augmented data is used during the training of the Deep Learning segmentation model. The data augmentation techniques used in this study are summarized in Table 2.

Table 2: Augmentation techniques used on the original training dataset.

Type of augmentation	Value
Rotation	[0-30] (+/- deg)
Translation	0.1 (fraction)
Scaling	0.5(+/- gain)
Shear	0.5 (+/- deg)
Flip left/right	0.5 (probability)
Mosaic	[0.1-0.9] (probability)
Mix-up	[0.1–0.9] (probability)



(a)

(b)

(c)

Figure 9: Target images: (a) Full board guardrail, (b) top board with and without clutter, and (c) only vertical poles, with and without clutter.

4.5 Model Development (YOLOv8 Model)

The main goal of this model is to classify images into safe and unsafe categories, as indicated in Section 4.3, by segmenting the presence (or lack of) and state (i.e., if all the components are present) of protection guardrails. For this study, we used YOLOv8 as the base model for the training, coupling it with other techniques, such as data augmentation and Unsupervised Deep Domain Adaptation (DA), to enhance guardrail detection in varying environments (i.e., indoor, outdoor, sunny, dark, etc.). We focus on DA using backpropagation. We use a feed-forward network or feature extractor to extract the domain invariant features and use them to train both the label classifier and Domain Adaptation Network. The label predictor uses the features to detect the guardrails and if it is a safe or unsafe condition. Whereas the domain classifier is used to predict whether the input belongs to the source or target domain.

The YOLOv8 network can be built with different architecture configurations by modifying the amount and type of layers, and the different hyperparameters that affect the weights of the network. To properly choose the best architecture, several tests need to be performed to assess the performance of the network. The domain predictor uses the same features from the last layers of the network to predict whether the input belongs to the source or target domain. The domain predictor layer consists of the gradient reversal layer, which backpropagates the loss for the optimization of the weights of the network with a negative value. This makes the Domain Adaptation network and feature extractor (Deep Learning network backbone) oppose each other, making it possible to learn domain invariant features.

The weights of the underlying deep feature extractor (i.e., YOLOv8 network backbone) are optimized to minimize the loss of the label classifier and to maximize the loss of the domain classifier. The latter encourages domain-invariant features to emerge while training or optimization. The three different networks (namely the feature extractor, label predictor and domain classifier) can be embedded into a single deep feed-forward network using standard layers and loss functions and can be trained using standard backpropagation algorithm [6]. The crucial layer of this architecture for the task of domain adaptation is the gradient reversal layer, which leaves the input unchanged during forward propagation and reverses the gradient by multiplying it by a negative scalar (-1 × *lambda*) during backpropagation.

To achieve the best possible results, different characteristics of the YOLOv8 model architecture, target data, and changes in hyperparameters, such as *lambda* in the Gradient Reversal layer, needed to be fine-tuned.

A summary of the different tested models and their corresponding results can be seen in Table 3 and Figure 10, respectively. The results include metrics: Precision, Recall, and Mean Average Precision (mAP) for the segmented guardrails in the images. These metrics show how well the model identifies the different configurations of guardrail elements in the images.

As seen in Figure 10, the results from training the YOLOv8 only with the source images with and without augmentation (Model#1 and Model #2) are not good on the validation Target data. The training "Model #3" with only 1 layer Domain Adaptation Layer (P5) performs poorer than the previous training as it was given only very few variations in Target data for training. The training of "Model #4" (same as "Model #3" but with more variation in target data for training) performs much better than "Model #3". This shows that even though the Target data are fed into training without the ground truth labels, the model can learn more Target domain features with the Domain Adaptation Network. The best model training was achieved with "Model 5," built with 3-Layers of DAN with features from P3, P4, P5 and more variation in the Target Training Data. "Model #6" is the same as "Model #5" except for the value of lambda for the Gradient reversal layer. "Model #5" was trained with lambda 1, and "Model #6" was trained with lambda 5, which has obtained slightly poor results. This shows that "Model #6" with lambda 5 punishes the rest of the model more to predict the correct domain class during backpropagation and yields worst results on label prediction. Hence, a better value of lambda for this training is 1, as it punishes the model the right amount during training.

Table 3: Training Model description and parameters.

Model	Details
#1	Training only source images without Data Augmentation
#2	Training only source images with Data Augmentation
#3	Training with source images and fewer target images and only the P5 DAN layer
#4	Training with source images and more target images and only the P5 DAN layer
#5	Training with source images and more target images and P3, P4, P5 DAN layers and lambda=1
#6	Training with source images and more target images and P3, P4, P5 DAN layers and lambda=5

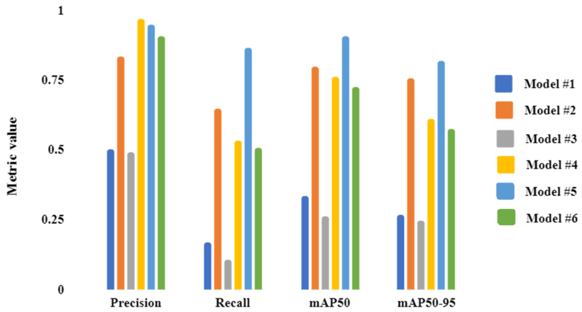


Figure 10: Results for different metrics of each model.

The selected YOLOv8-m architecture is configured with varying channel sizes for the P3, P4, and P5 layers, specifically 192, 384, and 576 channels, respectively. This configuration is dependent on the type of YOLOv8 variant used, which includes n, s, m, l, x models. In this architecture, the extracted features from the training phase are employed by the up-sampling process in the YOLOv8 head. This up-sampling is crucial for reconstructing the image back to its original size, and it is particularly instrumental in the segmentation tasks, where maintaining image resolution and detail is essential. The overall network structure, with the YOLOv8 classifier network and the Domain Adaptation, can be seen in Figure 11.

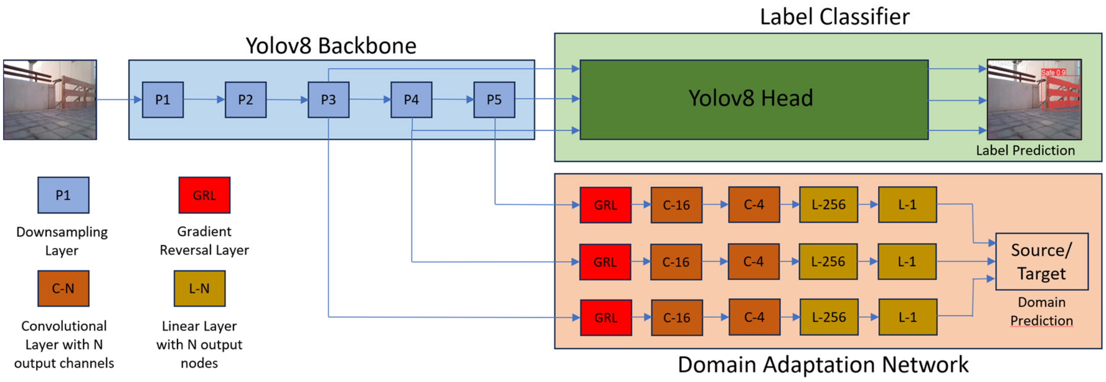


Figure 11: YOLOv8 with domain adaptation network for selected model (Model #5).

5. Results

5.1 Classification Results

The results show that Model #5, with a 3-layer Domain Adaptation Network, has achieved the highest accuracy on the validation target data. The values for the evaluation metrics for Model #5 are summarized in Table 4, and representative results to classify the target images are shown in Figure 12.

Table 4. Evaluation metrics for Model #5.

Precision	Recall	mAP50	mAP50-95
0.947	0.868	0.909	0.818

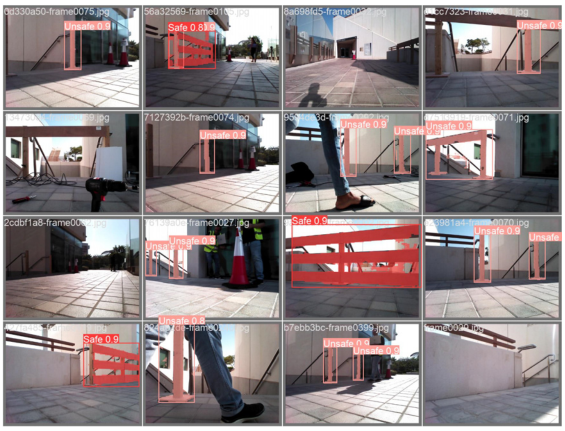


Figure 12: Validation results of Model #5.

The Confusion matrix (Figure 13) shows that Model #5 performs well in identifying the 'Unsafe' class with a high number of true positives (14). This indicates that the model is effective at detecting 'Unsafe' scenarios. There are no instances where 'Safe' is confused with 'Unsafe' or vice versa, which is positive as it suggests that the model can distinguish between these two classes effectively.

Overall, it can be said that the model has a high precision for the 'Unsafe' class since there is only one false positive; however, its recall is affected by the five false negatives. The presence of false negatives for 'Unsafe' (5 instances misclassified as 'Background') suggests that there may be room for improvement in distinguishing between 'Unsafe' and 'Background'. This could be attributed to the debris occluding the view, dark shadows, or being far from the camera, which is acceptable as the model has not been trained enough for such harsh cases. This could be improved in the future by training with more diverse and harsher data so the model learns more features.

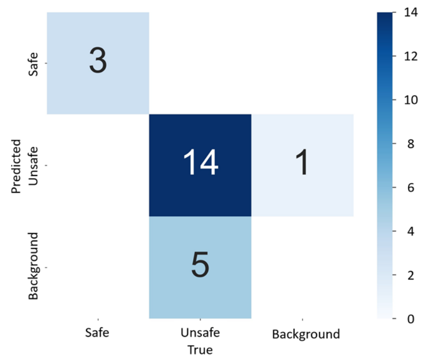


Figure 13: Confusion matrix of Model #5.

5.2 Failed cases

The guardrail detection by the best model is not always accurate. The failed cases, as shown in Figure 14, either have some debris lying in front of the guardrails, the guardrails are at a far distance, or the guardrails are too close and partially captured in the images. These are reasonable failed cases since the training data had no such cases and was not the objective of this work. The guardrails prediction with the AGV was designated to work with the guardrails being at a reasonable distance (i.e., from 2 meters to 4 meters) as can be seen in Figure 8, showing the setup for indoor data collection. Therefore, the failed cases in the target images with guardrails at distances less than 2 meters or more than 4 meters are understandable. In addition, the training data did not contain guardrails with debris lying in front of them. In future work, we will solve these shortcomings with more data, capturing all these situations and using depth information to overcome these.

Figure 14: Failed predictions by Model #5.

6. Conclusion and Future Work

From this work, we have achieved detection and segmentation of guardrails with YOLOv8 trained on source domain (i.e., indoor training data) and predicted on target domain (i.e., outdoor collected data) using Domain Adaptation with Backpropagation. This indicates that the model, needing fewer target domain images, can be used for guardrail detection across different domains without having to label the ground truth for each one of the images.

The model had some limitations that could be overcome by using images from failed cases in the target data during training, allowing the model to learn extra features. The significance of the unlabeled target training data is crucial, as it determines the performance of the model prediction on the target domain. The same method, along with transfer learning, can be effectively used to retrain the model repeatedly onto different domains with less unlabeled training data. This can save a significant amount of time while not sacrificing performance.

Some challenges faced by a vision-based approach can be overcome by adapting the same techniques to RGB-D images or by using point cloud data. The use of depth information might lead to better results. Another future aspect of this work lies in the real-time use of safety information to deploy predictive models in an online Digital Twin. Further work can also target alternative data collection and inspection methods, for example, unmanned aerial vehicles [18] for tailored human-assisted safety management software [19].

Acknowledgment

The research presented in this paper was in part funded by the European Union Horizon 2020 research and innovation program under grant agreement no. 101058548. This work also benefited from the NYUAD (CITIES) and NYUAD (SHORES) funded by Tamkeen under Awards CG013 and CG001.

References

[1] Teizer, J. Right-time vs real-time pro-active construction safety and health system architecture, *Constr. Innov.*, 16(3): 253–280, 2016, https://doi.org/10.1108/ci-10-2015-0049

[2] HSE. Statistics - Industries. 30.11.2023, https://www.hse.gov.uk/statistics/industry

[3] Finity, J. 36 Construction Safety Statistics for 2023," Procore. Accessed: 30.11.2023. https://www. procore.com/library/construction-safety-statistics

[4] Gopee M.A., Prieto S.A., García de Soto B. Improving autonomous robotic navigation using IFC files. *Constr. Robot.*, 2023, https://doi.org/10.1007/s41693-023-00112-8.

[5] Kolar Z., Chen H., Luo X. Transfer learning and deep convolutional neural networks for safety guardrail detection in 2D images. *Autom. Constr.*, 89:8–70, 2018.

[6] Ganin Y., Lempitsky V. Unsupervised Domain Adaptation by Backpropagation. arXiv, 2015.

[7] Li C., *et al.* A domain adaptation YOLOv5 model for industrial defect inspection. *Measurement*, 213, 112725, 2023.

[8] Jocher G., *et al.* ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation. Zenodo, 2022. doi: https://doi.org/10.5281/zenodo.7347926.

[9] Wang C.-Y. et. al. CSPNet: A New Backbone that can Enhance Learning Capability of CNN, 2019.

[10] Solawetz J. What is YOLOv8? The Ultimate Guide. https://blog.roboflow.com/whats-new-in-yolov8/

[11] Ultralytics. YOLOv8. https://docs.ultralytics.com/

[12] Prieto S., García de Soto B., Adan, A. A Methodology to Monitor Construction Progress Using Autonomous Robots. 37th ISARC https://doi.org/10.22260/ISARC2020/0210.

[13] Prieto S.A., Giakoumidis N., García de Soto, B. AutoCIS: An Automated Construction Inspection System for Quality Inspection of Buildings, *38th ISARC,* 2021.

[14] Prieto S.A., Giakoumidis, N., García de Soto B. Multi-agent robotic systems and exploration algorithms: Applications for data collection in construction sites. arXiv, 02, 2023.

[15] Johansen, K.W., de Figueiredo R.P., Golovina O., Teizer J. Autonomous Safety Barrier Inspection in Construction: An Approach Using Unmanned Aerial Vehicles and Safe BIM. 38th ISARC, 2021. https://doi.org/10.22260/ISARC2021/0085.

[16] BG Bau. B 100, Absturzsicherungen auf Baustellen, 2023. https://www.bgbau-medien.de/

[17] *Guardrail video data*, 2023. https://drive.google. com/drive/search?q= SMART%40NYUAD

[18] Siebert S., Teizer J. Mobile 3D mapping for surveying earthwork projects using an Unmanned Aerial Vehicle (UAV) system. *Autom. Constr,* 41:1-14, 2014, https://doi.org/10.1016/j.autcon.2014.01.004

[19] Golovina O., Teizer J., Johansen K.W., König, M. Towards autonomous cloud-based close call data management for construction equipment safety. *Autom. Constr.*, 132, 103962, 2021, https://doi.org/10.1016/j.autcon.2021.103962