16th International Conference on Information Science and Technology (ICIST 2026)
Coimbra, Portugal, October 1-6, 2026

Formal Verification of Deep Neural Networks for Image Classification in Industrial Environments

Manuel Abreu
Dept. of Electrical and Computer Engineering
Institute of Systems and Robotics, University of Coimbra
Coimbra, Portugal
manuel.abreu@isr.uc.pt

Luís Garrote
Dept. of Electrical and Computer Engineering
Institute of Systems and Robotics, University of Coimbra
Coimbra, Portugal
garrote@isr.uc.pt

Luís Conde Bento
Institute of Systems and Robotics, University of Coimbra
Polytechnic of Leiria
Leiria, Portugal
conde@isr.uc.pt

Bradley Schmerl
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
schmerl@andrew.cmu.edu

Urbano J. Nunes
Dept. of Electrical and Computer Engineering
Institute of Systems and Robotics, University of Coimbra
Coimbra, Portugal
urbano@isr.uc.pt

David Garlan
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
garlan@andrew.cmu.edu

2026 16th International Conference on Information Science and Technology (ICIST) | 979-8-3195-4311-0/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICIST71072.2026.11727805

***Abstract*—Deep Learning methods, particularly Convolutional Neural Networks (CNNs), have become widely adopted in industrial computer vision applications due to their strong image classification capabilities. Industrial autonomous systems, such as autonomous mobile robots (AMRs), rely on accurate object perception to ensure safe operation and reliable decision-making in complex environments. However, despite their high performance, CNN-based models remain vulnerable to adversarial perturbations and sensor-related degradations, which may lead to incorrect classifications and compromise system reliability in safety-critical applications.**

This work presents the robustness of CNN-based image classification models for industrial applications by analyzing their performance under increasing perturbation levels and identifying the limits at which misclassification occurs. An industrial object dataset containing seven classes was constructed using an object detection and extraction pipeline, resulting in object-centered RGB and depth image samples. To perform this analysis, two CNN-based architectures were developed and evaluated: a baseline RGB model and a multimodal RGB-D model incorporating depth information from multiple sources.

Robustness was assessed through two complementary approaches. Formal verification was performed using the alpha-beta-CROWN framework under an L_∞-bounded threat model, providing mathematical guarantees on model predictions under bounded perturbations. In addition, corruption-based experiments following the ImageNet-C protocol were conducted to evaluate model behavior under common image degradations, with a depth-suitable corruption subset introduced for metric depth modalities.

***Index Terms*—Image Classification, Deep Learning, Formal Verification, Robustness Evaluation, Autonomous Mobile Robots**

I. Introduction

Recent advances in Deep Learning (DL), particularly in Convolutional Neural Networks (CNNs), have improved computer vision systems across several domains, including industrial automation. In industrial environments, image classification models are integrated into automated processes such as warehouse automation and object sorting systems, where accurate object recognition is essential for operational efficiency and safe operation.

Despite their strong performance, CNN-based models remain vulnerable to adversarial perturbations, where small and often imperceptible input modifications can lead to incorrect predictions [1], as well as to common perception degradations such as illumination changes, blur, and noise. In industrial scenarios, such failures may result in operational interruptions, material waste, or safety risks to human operators, making classification accuracy alone insufficient to guarantee reliable system behavior.

To address these concerns, two complementary robustness evaluation strategies are considered in this work. Formal verification provides mathematical guarantees on model behavior by analyzing whether specific properties hold for all inputs within a bounded perturbation space [2], going beyond the finite-sample evaluation of traditional testing. Corruption-based robustness evaluation offers a complementary empirical perspective, assessing model behavior under structured image degradations representative of real-world conditions [3].

This work evaluates the robustness of RGB and RGB-

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

D image classification models for industrial perception, investigating whether depth information improves robustness against both adversarial perturbations and common image corruptions. Lightweight CNN-based architectures were selected as the classification backbone, as their low parameter count makes them computationally efficient for deployment on Autonomous Mobile Robots (AMRs) with limited onboard resources, while still being well-suited for robustness analysis in robotic perception pipelines [4]. An industrial object dataset was constructed through a YOLO-based [5] object extraction pipeline, yielding object-centered samples across seven industrial classes: industrial drums, boxes, box slots, fire extinguishers, forklifts, people, and toolboxes. The work contributes with a multimodal RGB-D dataset collected from a real warehouse environment, a comparative evaluation of CNN classifiers across multiple depth input configurations, a formal robustness analysis using alpha-beta-CROWN under bounded adversarial perturbations, and a corruption-based evaluation extended with a depth-suitable corruption subset designed for metric depth maps.

Fig. 1 illustrates the overall framework: an industrial dataset is constructed and used to train CNN-based classifiers, whose robustness is then evaluated through formal verification and corruption-based testing. Experimental results show that RGB-D models generally improve robustness under several corruption settings when compared to RGB-only models.

II. Related Work

The deployment of vision-based systems in industrial environments has grown with the use of deep learning, particularly CNNs. AMRs are increasingly used in logistics and warehouse automation, where reliable object recognition is central to safe and efficient operation. Recent work has demonstrated that YOLO-based detection models can be effectively integrated with robotic platforms for real-time inspection and object handling in unstructured industrial settings, showing strong potential for use across sectors such as logistics, ports, and manufacturing [6]. Despite these advances, industrial perception systems remain sensitive to environmental factors such as variable illumination, occlusions, and visually similar objects, motivating the development of more-robust recognition pipelines.

To address the inherent limitations of purely appearance-based methods, multimodal approaches combining RGB and depth data have been explored for object recognition in robotics. Eitel et al. [7] proposed a two-stream CNN architecture for RGB-D object recognition, demonstrating that depth information provides complementary geometric cues that improve classification robustness, particularly under imperfect sensing conditions. More recent work on industrial anomaly detection has further confirmed that fusing RGB with depth modalities reduces misclassification under challenging conditions such as cluttered scenes or specular surfaces [8]. These findings support the adoption of multimodal inputs in safety-critical perception pipelines.

Formal verification of neural networks has emerged as a principled methodology for providing mathematical guarantees on model behavior, going beyond empirical testing. Foundational work in this area includes the Reluplex algorithm [9], which formulated the neural network verification problem as a Satisfiability Modulo Theory query and demonstrated its feasibility for ReLU-activated networks.

A complementary line of work focuses on bound propagation methods. The alpha-beta-CROWN verifier [10], which combines incomplete verification via alpha-CROWN with complete verification through Branch-and-Bound (BaB), has consistently ranked as the top-performing verifier in the International Verification of Neural Networks Competition (VNN-COMP) [11-13].

Application-oriented verification studies have extended these tools to safety-critical domains. Huang et al. [2] introduced a Satisfiability Modulo Theory-based framework for verifying image classification decisions against perturbations representing realistic image manipulations. More recently, Elboher et al. [14] extended formal verification to object detection networks, demonstrating that verification can assess robustness properties beyond classification. Together, these works help to establish formal verification as an increasingly effective tool for assessing neural network trustworthiness in real-world applications.

Beyond worst-case adversarial robustness, the behavior of neural networks under naturally occurring image degradations is of significant practical relevance for deployment in industrial settings. Goodfellow et al. [1] demonstrated that small, structured input perturbations are sufficient to induce misclassification in deep networks, motivating systematic robustness analysis. Szegedy et al. [15] had earlier shown that such adversarial examples transfer across different model architectures, highlighting their fundamental nature.

To the best of our knowledge, formal verification methods have not yet been applied to multimodal RGB-D perception models, nor have formally verified robustness and corruption robustness been jointly evaluated in these systems. This work addresses that gap by combining formal verification with a corruption-based robustness evaluation framework adapted from the ImageNet-C methodology. In addition to the standard RGB corruptions, the proposed framework introduces a depth-suitable corruption subset specifically designed for metric depth maps, excluding appearance-based transformations that are not physically meaningful for depth data.

III. Methodology

This section describes the methodology adopted throughout this work. The image classification approach, including the CNN architectures and training setup, is presented in Section III-A. The formal verification framework is detailed in Section III-B, followed by the corruption-based robustness evaluation protocol in Section III-C.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

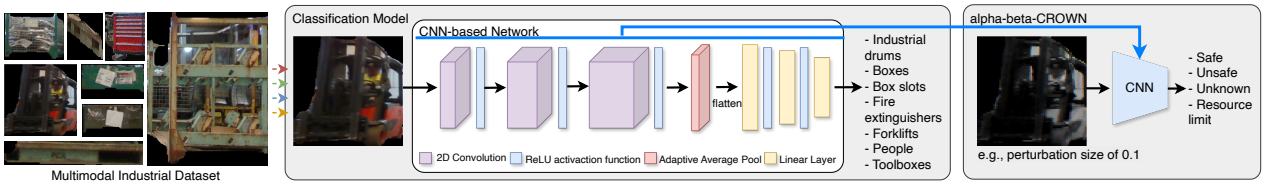


Fig. 1. Overview of the proposed pipeline for robustness evaluation of CNN-based image classification models. The framework includes dataset construction, model training, and formal verification under perturbations.

A. Image Classification

Our approach is centered around a custom CNN designed to process multi-modal data, combining RGB images with depth information.

All images are resized to a fixed low resolution of 64×64 pixels to ensure uniform input size for the network, to reduce memory requirements during formal verification, and to maintain computational efficiency suitable for deployment on AMR platforms with limited onboard resources. The depth images, which are provided in a 16-bit single-channel format, represent the projection of the 3D point cloud onto the camera frame, encoding the distance of objects from the sensor. They provide spatial information about the scene by explicitly indicating the distance between each point and the camera.

1) Architecture: Two variations of a CNN-based architecture were developed and evaluated to assess the impact of multimodal data, based on the networks proposed in [4]. The first model, a baseline architecture, processes standard 3-channel RGB images. Its structure consists of a backbone of three convolutional layers ($32 \rightarrow 64 \rightarrow 128$), followed by an adaptive average pooling layer and three fully connected layers ($128 \rightarrow 128 \rightarrow N_C$) to produce the final class logits. This RGB-based model contains 127,175 trainable parameters. To explore the benefits of incorporating geometric information, a second model was developed. This version considers RGB-D multimodality by processing a 4-channel input tensor, which is created by concatenating the 3-channel RGB image with its corresponding single-channel depth map. The resulting RGB-D architecture contains 127,463 trainable parameters.

2) Loss Function: The training process is optimized using a weighted Cross-Entropy Loss function. This approach was chosen to effectively address the class imbalance inherent in the dataset. By computing weights for each class in a balanced mode, minority classes are assigned a higher weight, thereby increasing the penalty for misclassifying them. These weights are integrated directly into the loss function, compelling the model to pay more attention to under-represented classes.

B. Formal Verification

To assess the robustness of the proposed CNN architectures against adversarial perturbations, we employ formal verification techniques. In particular, we formulate robustness properties as logical constraints over the network inputs and outputs, and verify whether these properties hold for all admissible perturbations.

Given a neural network $f(\cdot)$, an input image x, and its ground-truth class k, the verification problem consists of determining whether there exists a perturbed input $\hat{x}$ such that the network prediction changes under bounded perturbations.

For the verification experiments, we use the alpha-beta-CROWN [10] verifier, a state-of-the-art neural network verification framework [11-13] that supports convolutional architectures, multiple output classes, and various activation functions.

1) Input Constraints: Robustness is evaluated under bounded pixel-wise perturbations using an L_∞-norm threat model. Given an original image x and a perturbation magnitude ϵ, the admissible perturbed input $\hat{x}$ is constrained such that each pixel value remains within the interval:

$$x - \epsilon \leq \hat{x} \leq x + \epsilon$$

while ensuring that all pixel intensities remain within the valid image range $[0, 1]$. This formulation defines the set of all admissible adversarial perturbations explored during verification.

2) Output Constraints: Given an input sample x, the neural network produces an output logit vector

$$y = f(x) \in \mathbb{R}^{N_C},$$

where N_C represents the number of classes. In our experiments, $N_C = 7$, resulting in:

$$y = [y_0, y_1, \ldots, y_6]$$

For a correctly classified sample belonging to class k, robustness is verified by ensuring that the logit corresponding to the true class remains higher than the logits of all other classes under every admissible perturbation:

$$y_k > y_j, \quad \forall j \neq k$$

Equivalently, the predicted class must remain unchanged:

$$\arg\max_i y_i = k$$

If the verifier proves that these constraints hold for all perturbed inputs satisfying the input constraints, then the sample is considered formally robust for the specified perturbation radius ϵ. Otherwise, the verifier returns a counterexample corresponding to an adversarial example.

C. Corruption-Based Robustness Evaluation

In addition to formal verification, we evaluate model robustness under common image corruptions that simulate realistic perception degradations. Unlike formal verification, this analysis is empirical and aims to assess how model predictions are

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

affected by structured input perturbations commonly encountered in real-world conditions.

The evaluation follows the ImageNet-C corruption protocol [3] and includes the following corruption types: brightness, contrast, defocus blur, elastic transform, fog, frost, Gaussian blur, Gaussian noise, glass blur, impulse noise, JPEG compression, motion blur, pixelate, saturate, shot noise, snow, spatter, speckle noise, and zoom blur. Each corruption is applied at five severity levels, ranging from 1 (lowest severity) to 5 (highest severity).

This evaluation complements the formal verification analysis: while formal verification provides guarantees within bounded perturbation regions, corruption-based evaluation analyzes model behavior under diverse and semantically meaningful degradations that may occur in industrial perception systems.

IV. Experimental Results

A. Implementation Details

All experiments were implemented in Python using PyTorch. Training was conducted on a computer equipped with a NVIDIA GeForce RTX 3060 GPU, AMD Ryzen 7 3800X CPU, and 64GB DDR4 memory.

1) Image Classification Setup: The models were trained for 300 epochs using the Adam optimizer, with weight decay of 1×10^{-5} and an initial learning rate of 1×10^{-4}. To ensure stable convergence, a linear learning rate schedule with a warmup phase was employed. During the initial 20% of the training the learning rate was gradually increased linearly from zero to its target initial value.

The RGB and depth images were normalized and resized to 64×64. The input is constructed by concatenating the normalized RGB image (3 channels) with its corresponding normalized depth map (1 channel). To prevent overfitting, data augmentation was applied to all input modalities. For RGB images, Gaussian noise was added independently to the hue, saturation, and value channels, introducing small variations in color and brightness. Depth maps were augmented by adding Gaussian noise with a standard deviation of 0.05.

In this work, three different sources of input depth data were explored. The first depth map is obtained from the camera-native RGB-D depth map. The second is generated from LiDAR data projected into the camera frame, and interpolated using Delaunay triangulation [16], presenting an improvement over a sparse depth map with limited structural detail, at the cost of introducing depth errors in areas not represented by LiDAR points. The last depth source proposed in [17], where RGB images, sparse LiDAR points projected onto the camera frame, and segmentation cues are fused to generate a refined depth map that leverages the complementary strengths of each modality: the dense semantic context of RGB images, the metric precision of LiDAR, and the geometric structure encoded by segmentation maps.

Based on these depth representations, four input configurations were evaluated: baseline model using only RGB images (RGB), RGB combined with the camera-native depth map (RGB-D$_{Cam}$), RGB combined with the LiDAR depth map interpolated with Delaunay triangulation (RGB-D$_{LiDAR}$), and RGB combined with the depth map proposed in [17] (RGB-D$_{Fused}$). For all multimodal configurations, the depth map is concatenated with the RGB image, thereby enabling the evaluation of how different depth representations affect the network's classification performance.

2) Formal Verification Setup: Formal robustness verification was performed using the alpha-beta-CROWN verifier [1]. Verification queries were evaluated under an L_∞-bounded adversarial threat model, where each input pixel was allowed to vary within a perturbation radius ϵ, while remaining constrained to the valid image range $[0, 1]$.

For each correctly classified sample, the verifier checks whether there exists an adversarial perturbation capable of changing the predicted class. If such perturbation exists, the sample is considered non-robust. Otherwise, robustness is formally certified for the specified perturbation radius.

The verification pipeline follows the standard alpha-beta-CROWN workflow. First, a projected gradient descent (PGD) attack is applied to efficiently search for adversarial counterexamples. Samples for which no adversarial example is found are then subjected to formal verification using alpha-CROWN bound propagation, followed by complete verification through the BaB procedure when required. The complete verifier was configured using the BaB backend together with the kfsb branching heuristic.

PGD adversarial search was performed using 100 optimization steps and 30 random restarts. Complete verification was executed with a timeout of 3600 seconds per sample. Verification experiments were conducted for perturbation radius of $\epsilon \in \{0.1, 0.05, 0.01, 0.005, 0.001, 5\times10^{-4}, 10^{-4}, 5\times10^{-5}, 10^{-5}\}$. The same L_∞ perturbation radius ϵ was applied uniformly to all RGB-D input channels during formal verification.

For the RGB-D models, adversarial perturbations were applied across all four input channels, including both RGB and depth information.

B. Industrial Dataset

The dataset was collected in a Stellantis warehouse located in Mangualde, Portugal, using the Interbot platform [18], equipped with a Velodyne VLP-16 LiDAR and an IntelRealSense D435 RGB-D camera. It comprises 5859 synchronized RGB-D images and LiDAR point clouds.

This dataset has a set of seven abundant classes in industrial environments, namely industrial drums, boxes, box slots, fire extinguishers, forklifts, people, and toolboxes (SAM [19] was used to create the segmentation masks). The resulting detection and segmentation dataset, of this data collection, comprises 5859 frames containing a total of 121159 annotated objects. To evaluate detection and segmentation performance, we used a set of YOLO models (from version 8 to 12 [20]), with their results presented in Table I. Some YOLO variants were evaluated only for detection or segmentation, as not all versions provide

[1] https://github.com/Verified-Intelligence/alpha-beta-CROWN

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

TABLE I

COMPARISON OF DETECTION (BOUNDING BOX) AND INSTANCE-SEGMENTATION PERFORMANCE ACROSS YOLO VARIANTS (YOLOV8M-YOLOV12M) ON THE TEST SET. DETECTION RESULTS REPORT PRECISION, RECALL, AND MEAN AVERAGE PRECISION (MAP@50). SEGMENTATION RESULTS REPORT THE SAME METRICS FOR BOUNDING BOXES (B) AND MASKS (M). UNAVAILABLE TASKS ARE MARKED WITH '-'.

Version	Detection			Segmentation					
	Precision	Recall	mAP@50	Precision (B)	Recall (B)	mAP@50 (B)	Precision (M)	Recall (M)	mAP@50 (M)
YOLOv8m	0.93	0.48	0.58	0.98	0.98	0.99	0.96	0.96	0.97
YOLOv9m	0.91	0.47	0.59	-	-	-	-	-	-
YOLOv9c	-	-	-	0.98	0.98	0.99	0.96	0.96	0.98
YOLOv10m	0.80	0.52	0.62	-	-	-	-	-	-
YOLOv11m	0.98	0.98	0.99	0.98	0.98	0.99	0.96	0.96	0.97
YOLOv12m	0.98	0.98	0.99	-	-	-	-	-	-

implementations for both tasks. Among the tested models, YOLOv11m achieves the best overall performance, exhibiting high precision, recall, and mAP@50 for both detection and segmentation, and was therefore selected as the segmentation model used in the subsequent stages of this work.

After creating the detection and segmentation datasets and selecting the best-performing model, the generated bounding boxes were used to extract regions of interest from the RGB images and depth images. These regions correspond to the target objects and serve as the basis for the construction of a new dataset aimed at image classification. The RGB and depth images were cropped according to the bounding boxes, and a segmentation mask was applied to isolate the object and ensure that only relevant data are retained. RGB and depth images were normalized to a common $[0, 1]$ input range. RGB images were normalized using standard min-max scaling, and 16-bit depth images were normalized by first converting them to metric depth values and then linearly scaling them by the camera maximum range.

A total of 203 images, containing 3558 objects, comprise the dataset used for the classification task, split into 80% for training and 20% for testing. The resulting dataset is imbalanced across seven classes, with box slots (1742) and boxes (1270) predominant, followed by people (249), industrial drums (144), toolboxes (61), fire extinguishers (51) and forklifts (41), reflecting the natural distribution of objects in the warehouse environment.

1) Corruption-Based Setup: Our corruption pipeline is implemented from the robustness codebase presented in [3][2] and adapted to our multimodal dataset structure composed of RGB, camera depth, LiDAR depth from 3D points projected onto the camera frame, and a depth model variant [17]. For each original object crop, we generate corrupted samples at 5 severity levels that represent image corruption operations (from level 1 to level 5), and for each level, we generate 10 stochastic variants. For RGB images, we apply all 19 CIFAR-C-style corruptions (noise, blur, weather, digital, and color transformations). For depth modalities, we apply only a depth-suitable subset of 10 corruptions: Gaussian noise, shot noise, impulse noise, speckle noise, Gaussian blur, defocus blur, motion blur, zoom blur, elastic transform, and pixelate. Corruptions such as fog, frost, snow, spatter, brightness, contrast, and saturate were excluded for depth inputs because they are color/appearance effects not physically meaningful for metric depth maps.

Each generated sample uses a deterministic seed, guaranteeing repeatable augmentations while preserving per-sample randomness. All inputs are resized to 64×64, and all modality inputs are independently normalized to $[0, 1]$ range. Depth images are stored as 16-bit PNGs, temporarily mapped to 8-bit for corruption operators, and then mapped back to 16-bit. At evaluation time, raw network outputs (7 logits, without softmax/argmax) are stored for each crop file.

[2]https://github.com/hendrycks/robustness

C. Image Classification Metrics

The performance of the classification models was evaluated on the test set. A confusion matrix was generated for each input configuration to provide a detailed visualization of the model's performance, offering a clear view of correct and incorrect predictions for each class. The RGB configuration was used as the baseline for comparison. For the remaining configurations, the confusion matrices were represented relative to the RGB baseline, where green indicates an improvement and red indicates a degradation in class-wise performance.

From these matrices, several key performance indicators were calculated. Overall accuracy, representing the ratio of all correct predictions to the total number of samples, was used as a primary measure of the model's general performance. Precision, which measures the accuracy of positive predictions for each class, and recall, which measures the model's ability to identify all relevant instances of a class, were also considered.

D. Corruption-Based Robustness Metrics

To evaluate robustness under image corruptions, each corruption type was applied at five different severity levels. For each corruption type c and severity level l, the unsafe rate was computed as:

$$\mathrm{UnsafeRate}(c, l) = \frac{N_{\mathrm{unsafe}}(c, l)}{N_{\mathrm{total}}(c, l)} \times 100,$$

where $N_{\mathrm{unsafe}}(c, l)$ represents the number of corrupted samples that were incorrectly classified, and $N_{\mathrm{total}}(c, l)$ corresponds to the total number of evaluated samples for the respective corruption configuration.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

TABLE II
CLASSIFICATION PERFORMANCE COMPARISON BETWEEN THE EVALUATED MODELS. REPORTED PRECISION, RECALL, AND F1-SCORE CORRESPOND TO THE AVERAGE ACROSS ALL CLASSES (%).

Model	Accuracy	Precision	Recall	F1-score
RGB	80.43	72.83	92.08	77.90
RGB-D$_{Cam}$	83.39	71.83	93.76	77.61
RGB-D$_{LiDAR}$	**86.35**	**78.99**	**95.29**	**84.96**
RGB-D$_{Fused}$	80.92	70.44	91.16	77.04

The unsafe rate represents the percentage of corrupted samples that lead to misclassification. In addition, aggregated unsafe rates were computed by averaging the results across corruption types and severity levels, enabling a direct robustness comparison between the evaluated models.

E. Image Classification Results

Table II summarizes the classification performance of the evaluated models in terms of accuracy, precision, recall, and F1-score.

The inclusion of depth information generally improved the classification performance when compared to the RGB baseline model. Among all evaluated configurations, RGB-D$_{LiDAR}$ achieves the best overall performance, obtaining the highest accuracy, recall, and F1-score values.

The RGB-D$_{Cam}$ configuration also demonstrated a noticeable improvement over the RGB baseline, indicating that the incorporation of geometric depth information contributes positively to the classification task. In contrast, RGB-D$_{Fused}$ achieves performance comparable to the RGB baseline, suggesting that the fused depth representation did not provide additional discriminative information for the current experimental setup.

Fig. 2 presents the normalized confusion matrices obtained for the evaluated models. The RGB configuration is shown as the reference baseline, while the remaining matrices highlight the class-wise differences with respect to RGB.

F. Formal Verification Results

Before applying verification, misclassified samples must be identified, which correspond to inputs that are already incorrectly classified by the network prior to verification. Since robustness verification assumes a correctly classified sample, these cases are reported separately and are not considered valid robustness certifications.

For the remaining samples, the robustness verification results produced by alpha-beta-CROWN are categorized into three possible outcomes: safe, unsafe, and unknown. A sample is considered safe when the verifier proves that no adversarial perturbation exists within the specified perturbation bound. Conversely, a sample is labeled as unsafe when an adversarial example is successfully found, demonstrating that the robustness property does not hold. The unknown category corresponds to cases where the verifier cannot determine robustness within the imposed computational limits, typically due to timeout or branch-and-bound resource constraints.

Fig. 3 summarizes the verification outcomes for different perturbations of the four models. In general, increasing the perturbation bound leads to a reduction in the number of certified robust samples and an increase in adversarial counterexamples.

1) Quantitative Results: Quantitative analysis of the results shows that, for low values of ϵ, all evaluated models exhibit high adversarial robustness, with virtually all examples classified as safe. As the value of ϵ increases, a progressive degradation of robustness is observed, reflected by the growing proportion of examples classified as unsafe and by the increase in cases where the verifier reaches computational resource limits. Among the evaluated models, RGB-D$_{LiDAR}$ achieves the lowest misclassification error of 13.65%, corresponding to the grey zone in Fig. 3, while maintaining good robustness for small values of ϵ, whereas RGB shows the highest misclassification error rate of 19.57%, indicating a lower classification capability. Overall, the results demonstrate that, although all models are robust to small perturbations, adversarial robustness decreases significantly for larger perturbations.

2) Qualitative Results: An adversarial example corresponds to a perturbed input capable of altering the prediction of the neural network. In the context of formal verification, the existence of such perturbation demonstrates that the robustness property does not hold for the specified perturbation radius ϵ.

Fig. 4 presents both adversarial examples generated during the verification process using alpha-beta-CROWN and samples generated under common corruption scenarios. Some perturbations are nearly imperceptible to the human eye, yet they are sufficient to change the predicted class of the model. This highlights the sensitivity of the classifier to input perturbations and illustrates the importance of formally verifying robustness properties under bounded adversarial attacks.

G. Corruption-Based Robustness Results

Fig. 5 presents the corruption-based robustness evaluation for all model configurations: RGB, RGB-D$_{Cam}$, RGB-D$_{Fused}$, and RGB-D$_{LiDAR}$. Each subplot corresponds to a corruption type and reports the unsafe rate across severity levels, ordered from level 5 to level 1. Corruptions shared across all multimodal configurations are presented first to facilitate direct comparison between input modalities.

Overall, the RGB baseline exhibits higher sensitivity to image corruptions than the RGB-D configurations. For most corruption types and severity levels, multimodal models maintain lower unsafe rates, indicating robustness under degraded visual conditions.

The impact of corruption severity is consistently observable, with unsafe rates generally increasing as perturbation strength increases. Noise-related corruptions such as impulse noise and Gaussian noise, as well as deformation-based corruptions such as elastic transform, produce noticeable performance degradation across all RGB-D configurations. In contrast, corruptions such as pixelate, defocus blur, and Gaussian blur tend to produce comparatively smaller performance variations.

Across all corrupted RGB samples, the RGB model achieves an overall unsafe rate of 33.42%, indicating substantial sen-

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

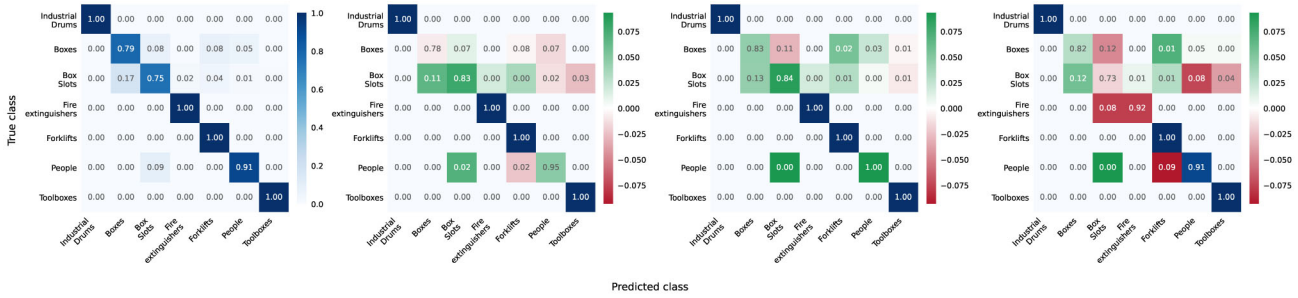


Fig. 2. Confusion matrices for the evaluated input configurations, shown from left to right: RGB, RGB-D$_{Cam}$, RGB-D$_{LiDAR}$, and RGB-D$_{Fused}$. The RGB matrix serves as the baseline, while the remaining matrices highlight the variation with respect to RGB: green indicates an improvement and red indicates a degradation.

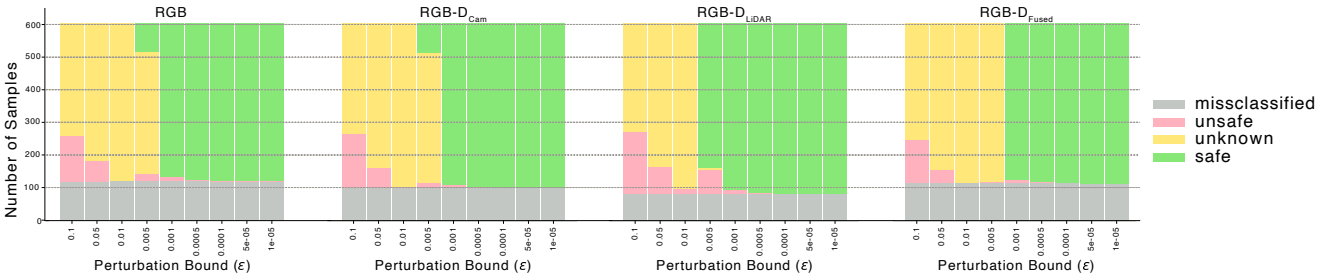


Fig. 3. Formal verification results for the evaluated models: RGB, RGB-D$_{Cam}$, RGB-D$_{LiDAR}$, and RGB-D$_{Fused}$, from left to right.

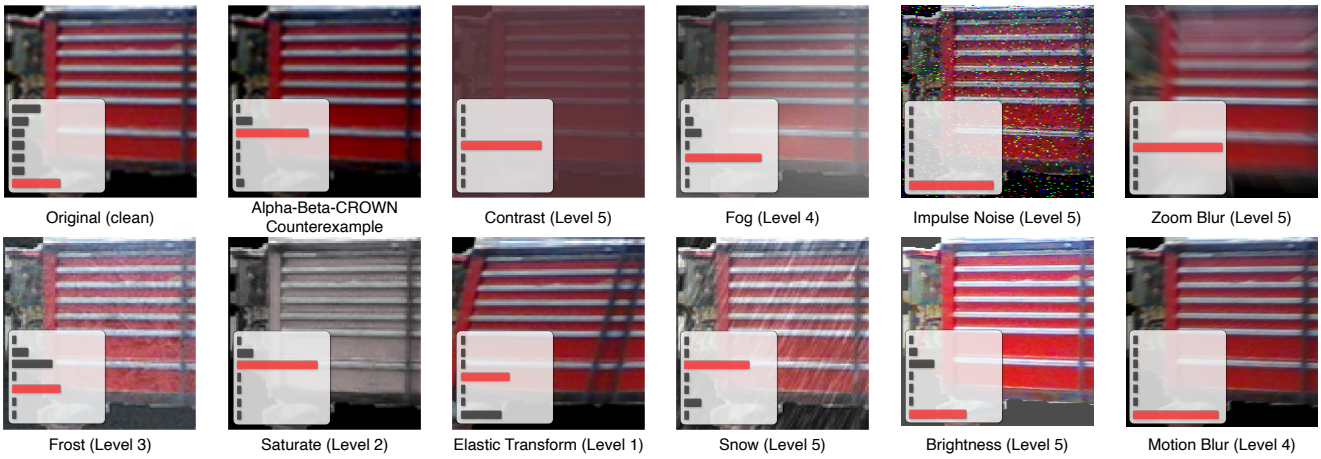


Fig. 4. Adversarial example generated during the formal verification process using alpha-beta-CROWN for the toolbox class, along with samples generated for common corruptions (same class order as in Fig. 2).

sitivity to image degradations. The highest unsafe rates are observed for frost (65.79%), fog (63.63%), and brightness (56.22%). Lower unsafe rates are obtained for JPEG compression (19.56%), speckle noise (19.89%), and shot noise (20.25%).

For RGB-D models, lower unsafe rates are obtained in all depth sources evaluated: 18.94% for RGB-D$_{Cam}$, 22.16% for RGB-D$_{Fused}$, and 19.91% for RGB-D$_{LiDAR}$. Within the aggregated RGB-D setting, impulse noise (26.25%), elastic transform (23.29%), and Gaussian noise (21.75%) produce the highest unsafe rates, whereas pixelate (17.76%), defocus blur (18.02%), and Gaussian blur (18.29%) have a comparatively smaller impact on classification performance.

Overall, the quantitative results indicate that incorporating depth information improves robustness under common image corruptions when compared to RGB-only inputs, although performance degradation remains noticeable under noise-related and geometric distortions.

V. Conclusion

This work presented a robustness analysis of CNN-based image classification models for industrial perception, evaluating the impact of incorporating depth information through formal verification and corruption-based robustness evaluation.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

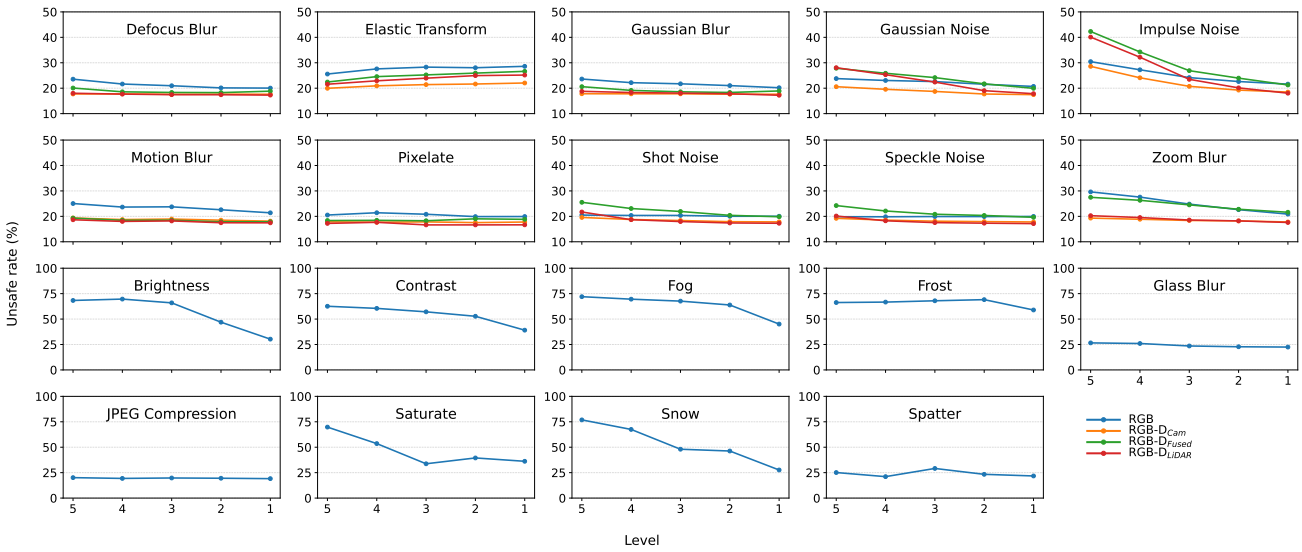


Fig. 5. Model robustness under corruption severity levels (from highest to lowest), comparing RGB, RGB-D$_{Cam}$, RGB-D$_{Fused}$, and RGB-D$_{LiDAR}$. Each subplot corresponds to one corruption type and reports the unsafe rate (%) across severity levels.

An industrial object dataset was constructed from real warehouse data using an object detection and extraction pipeline, yielding object-centered RGB and RGB-D samples across seven industrial classes. Lightweight CNN-based classifiers were trained and evaluated under four input configurations, combining RGB images with depth maps from different sources.

Classification results demonstrated that incorporating depth information generally improves performance over the RGB baseline, with multimodal configurations achieving up to a 5.92 percentage point increase in accuracy and notable gains in F1-score. Beyond nominal performance, formal verification using the alpha-beta-CROWN framework indicated that multimodal RGB-D models generally maintain higher certified robustness than the RGB baseline across perturbation bounds, suggesting that depth information contributes not only to classification accuracy but also to more stable and reliable feature representations. Complementing this analysis, corruption-based evaluation following the ImageNet-C methodology further confirmed that multimodal models consistently outperform RGB-only perception under common image degradations, with the proposed depth-suitable corruption subset enabling a more meaningful robustness assessment for metric depth modalities. Overall, these results indicate that multimodal RGB-D perception, combined with formal verification, provides a more reliable foundation for trustworthy object classification than RGB-only pipelines in industrial robotic systems.

Future work will explore the integration of these verification methods into AMR systems, the extension of verification to object detection architectures, and the combined analysis of corruption robustness and formal verification by evaluating certified robustness on corruption-degraded samples. Additionally, future research will investigate the use of the alpha-beta-CROWN framework to guide neural network retraining to improve robustness against adversarial perturbations in safety-critical industrial applications.

Acknowledgments

This work has been supported by the Portuguese Foundation for Science and Technology (FCT) through grant ISR-UC UID/00048/2025 (DOI: 10.54499/UID/00048/2025) and by Agenda "GreenAuto: Green innovation for the Automotive Industry", with reference 02/C05-i01.01/2022.PC644867037-00000013.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations (ICLR)*, 2015.

[2] X. Huang, M. Kwiatkowska, S. Wang, and M. Wu, "Safety verification of deep neural networks," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds. Springer International Publishing, 2017, pp. 3–29.

[3] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *Proceedings of the International Conference on Learning Representations*, 2019.

[4] N. Cohen, M. Ducoffe, R. Boumazouza, C. Gabrea, C. Pagetti, X. Pucel, and A. Galametz, "Verifiou – robustness of object detection to perturbations," in *2025 AIAA DATC/IEEE 44th Digital Avionics Systems Conference (DASC)*, 2025.

[5] R. Khanam and M. Hussain, "YOLOv11: An overview of the key architectural enhancements," *arXiv:2410.17725*, 2024.

[6] J. Sanchez-Cubillo, J. Del Ser, and J. L. Martin, "Toward fully automated inspection of critical assets supported by autonomous mobile robots, vision sensors, and artificial intelligence," *Sensors*, vol. 24, no. 12, 2024.

[7] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, "Multimodal deep learning for robust RGB-D object recognition," in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015.

[8] D.-C. Hoang, P. X. Tan, A.-N. Nguyen, T. H. A. Duong, T.-M. Huynh, D.-M. Nguyen, M.-D. Cao, D.-H. Ngo, T.-U. Nguyen, K.-T. Phan, M.-Q. Do, X.-T. Dinh, V.-H. Duong, N.-A. Hoang, and V.-T. Nguyen, "Accurate industrial anomaly detection with efficient multimodal fusion," *Array*, vol. 28, p. 100512, 2025.

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.

[9] G. Katz, C. Barrett, D. L. Dill, K. Julian, and M. J. Kochenderfer, "Reluplex: An efficient SMT solver for verifying deep neural networks," in *Computer Aided Verification*, R. Majumdar and V. Kunčak, Eds. Springer International Publishing, 2017, pp. 97–117.
[10] K. Xu, H. Zhang, S. Wang, Y. Wang, S. Jana, X. Lin, and C.-J. Hsieh, "Fast and Complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers," in *International Conference on Learning Representations*, 2021.
[11] C. Brix, S. Bak, C. Liu, and T. T. Johnson, "The fourth international verification of neural networks competition (VNN-COMP 2023): Summary and results," 2023.
[12] C. Brix, S. Bak, T. T. Johnson, and H. Wu, "The fifth international verification of neural networks competition (VNN-COMP 2024): Summary and results," 2024.
[13] K. Kaulen, T. Ladner, S. Bak, C. Brix, H. Duong, T. Flinkow, T. T. Johnson, L. Koller, E. Manino, T. H. Nguyen, and H. Wu, "The 6th international verification of neural networks competition (VNN-COMP 2025): Summary and results," 2025.
[14] Y. Y. Elboher, A. Raviv, Y. L. Weiss, O. Cohen, R. Assa, G. Katz, and H. Kugler, "Formal verification of deep neural networks for object detection," 2024.
[15] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 12 2013.
[16] L. Garrote, J. Perdiz, L. A. da Silva Cruz, and U. J. Nunes, "Point cloud compression: Impact on object detection in outdoor contexts," *Sensors*, 2022.
[17] M. Abreu, L. Garrote, and U. J. Nunes, "Distilling Apple DepthPro for RGB-LiDAR depth estimation," *Robotics and Autonomous Systems*, p. 105437, 2026.
[18] R. Cruz, L. Garrote, A. Lopes, and U. J. Nunes, "Modular software architecture for human-robot interaction applied to the InterBot mobile robot," in *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018.
[19] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.
[20] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO," May 2026. [Online]. Available: https://doi.org/10.5281/zenodo.20399943

Authorized licensed use limited to: Carnegie Mellon Libraries. Downloaded on October 22,2026 at 17:27:40 UTC from IEEE Xplore. Restrictions apply.