Robustness Testing of a Machine Learning-based Road Object Detection System: An Industrial Case

Anne-Laure Wozniak[*]
anne-laure.wozniak@kereval.com
Kereval
Rennes, France

Sergio Segura
sergiosegura@us.es
SCORE Lab, I3US Institute,
Universidad de Sevilla
Seville, Spain

Raúl Mazo[†]
raul.mazo@ensta-bretagne.fr
ENSTA Bretagne
Brest, France

Sarah Leroy
sarah.leroy@kereval.com
Kereval
Rennes, France

ABSTRACT

With the increasing development of critical systems based on artifi-cial intelligence (AI), methods have been proposed and evaluated in academia to assess the reliability of these systems. In the context of computer vision, some approaches use the generation of images altered by common perturbations and realistic transformations to assess the robustness of systems. To better understand the strengths and limitations of these approaches, we report the results obtained on an industrial case of a road object detection system. By compar-ing these results with those of reference models, we identify areas for improvement regarding the robustness of the system and the metrics used for this evaluation.

CCS CONCEPTS

• **Computing methodologies** → *Machine learning.*

KEYWORDS

robustness testing,

machine learning,

object detection,

industrial case

1 INTRODUCTION

With the increasing development of critical systems based on artificial intelligence (AI) the question of the reliability of these systems has become of paramount importance. As such systems usually evolve in dynamic environments, it is important to ensure that they are indeed trustworthy during their entire lifecycle under different operating conditions.

A common approach to assess the reliability of AI system is by testing its *robustness*, this is, assessing "the degree to which a system operates correctly in the presence of invalid inputs or stressful environmental conditions" (ISO/IEC/IEEE 24765:2010). In the context of computer vision—the scope of this paper—this is often done by generating adversarial examples [1]. Adversarial examples are created with perturbations that are invisible to the human eye and that can change the predictions of AI models dramatically. Other approaches resort to generating images altered by common perturbations and realistic transformations, the choice of which may be guided by domain knowledge [2], coverage measures [5], or by properties that the system must verify [4]. These transformations usually concern three aspects of the system under test (SUT): the environment, the sensor, and the digital processing. Thus, various perturbations on images have been proposed to ensure the robustness of the system in novel scenarios (e.g. changes in weather or in the settings of its hardware setup).

Regardless of techniques used, most approaches have been proposed and evaluated in academia and therefore their applicability and usefulness in practice is mostly unknown. However, industrial case studies are required to better understand the strengths and limitations of current software engineering approaches for assessing the reliability of AI systems.

In this paper, we report the results on an industrial case on robustness testing of a safety-critical machine learning-based object detection system. In contrast to related work, mostly interested in assessing the performance under worst-case scenarios (e.g., adversarial examples), we aim to test the performance of the system when facing common perturbations, encountered in its normal functioning. To this end, we use fifteen transformations representative of changes in the sensor and digital processing parameters. By comparing the behaviour of the SUT with that of baseline models, we observe several similarities in their overall behaviour against perturbations, with local differences that may indeed be related to the datasets used and the intrinsic robustness of the models. The benefit of re-training with data augmentation is also studied.

In what follows, we report the system under test (Section 2), experimental evaluation (Section 3), and lessons learned (Section 4).

2 ROAD OBJECT DETECTION SYSTEM

The system under consideration has been developed by an industrial actor in the field of embedded artificial intelligence for the automotive and smart city domains. It falls within the scope of road monitoring and aims to be widely deployed in cities. Using cameras strategically placed in road junctions, it allows traffic regulation at

[*] Also with Lab-STICC, ENSTA Bretagne.
[†] Also with Universidad Eafit (Medellin, Colombia).

two levels: at the macroscopic level for the flow of vehicles throughout the day, and at the microscopic level for the management of specific events such as the crossing of a pedestrian, as in Figure 1. In practice, it consists of a road observation module, linked to a traffic light management module. Thus, when a person approaches the pedestrian crossing, the system detects him and can act on the traffic lights to let him pass.

In particular, we are interested in the subsystem based on machine learning, whose aim is to correctly detect and identify the various road users (vehicles, pedestrians, etc.) on the basis of images from the cameras. This subsystem is critical at the microscopic level, and a failure in detection can have serious, life-threatening consequences, as the decision to change the traffic lights is based on its detections. It is therefore imperative for this subsystem to be reliable.

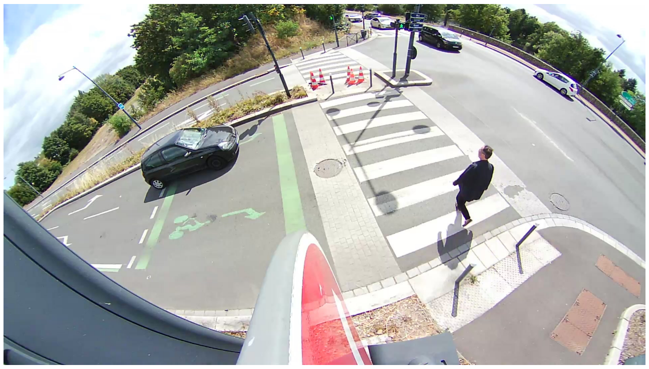

Figure 1: View of the road junction

3 EVALUATION

3.1 Objective and scope

Our objective is to characterise the detection system and produce robustness metrics in order to assess its reliability. Robustness can be assessed against perturbations related to the hardware and software environment (e.g. change in the parameters of the image signal processor), the observed environment (e.g. change in weather), and external adversary (e.g. adversarial attacks). In this paper, we focus on testing the robustness with respect to the hardware and software environment. This is a particularly relevant problem for the system under test since it needs to be usable with different camera technologies.

3.2 Experimental setup

We consider the detection system in a black-box setting. We therefore have no information on its architecture and parameters. Moreover, the interactions with the system are limited to sending an image from a camera as input, and retrieving a list of detected objects as output. Six classes of objects are detected by the system: cars, trucks, buses, people, bicycles, and motorcycles.

The object detection system has been trained and evaluated on a proprietary dataset, called Mango. Each image in the dataset is the view of a real road junction. In practice, it consists of images from the video stream that have already been preprocessed (the details of which are not known) to be saved in jpeg format (1920x1080, 96 dpi). We do not have access to the data used for training and performance evaluation by the development team. However, in order to perform the tests, we have at our disposal 2645 images extracted from this dataset which were not used during the training phase of the detection model.

Following its training, the detection system has been evaluated by the development team of the system under test (SUT), using the COCO detection evaluation metrics[1]. These 12 metrics are widely used in object detection, as they take into account both localisation and classification performance. For the sake of conciseness, we will only use the AP metric in reporting our results. It is the average accuracy over all classes, for Intersection over Union (IoU) thresholds between 0.5 and 0.95. This metric is the primary COCO challenge metric and rewards detectors with better localization. Its values are between 0 and 1, and the better the performance, the higher the value. According to the COCO challenge leaderboard[2], the latest and best performing detection models have an AP around 0.5. Nevertheless, more attention will be paid to the evolution of this metric during the tests than to its intrinsic value (see below).

In the absence of specifications to guide the testing of the detection system, we seek to test the system against plausible scenarios. To do so, we analyse the response of the system to images that have been modified by realistic transformations with respect to the image acquisition and creation process. Thus, the main constraint is that images produced may reflect a change in the settings or a failure of the subsystems upstream of the detection model. In practice, a system that is robust to a change or a failure will maintain the same performance; i.e., it will make the same predictions as in the initial configuration.

In order to define relevant perturbations, we have selected the transformations proposed in [2] that can be applied to our system and that correspond to our objectives. In particular, we set aside all the transformations that reflect meteorological changes. We then mapped these transformations, originally classified into four categories (blur, noise, weather, digital), to the parameters that can affect the different stages of the image acquisition process (see Table 1). This ensures that the transformations are representative of reality. Moreover, this mapping between the real phenomena and the types of transformations on the images allowed us to define new transformations that were not included in [2], such as dead pixels, unsharp mask, and chromatic aberration. This gives a total of 15 transformations. The *blur* category includes the different blurring techniques (Gaussian blur, zoom blur, motion blur, etc.), but also the inverse transformations (e.g. pixelation). The *colour* category includes transformations that affect the histogram of the image in the broadest sens, such as colour, contrast and brightness. The *noise* category includes Gaussian noise, Speckle noise, Salt noise, Pepper noise, and the Salt and Pepper noises combination. Finally, the *faulty pixel* category includes dead pixels and their extension to dead lines.

In order to study the response of the system to perturbations, we first need to make the results comparable. However, not all transformations performed have the same level of severity nor represent

[1] https://cocodataset.org/#detection-eval
[2] https://cocodataset.org/#detection-leaderboard

Table 1: Types of transformations.

Step	Parameter	Category
Acquisition (camera)	Lens material	Blur
	Lens distortion	Image distortion
	Focus	Blur
	Resolution	Blur
	Aperture	Colour
	Shutter speed	Blur
	ISO Sensitivity / Gain	Noise
	Dead pixels/lines	Faulty pixel
Processing (ISP)	Dematrixing	Colour
	Tone mapping (HDR)	Colour
	Sharpening	Blur
	Contrast	Colour
	Brightness	Colour
	Noise reduction	Noise
	White balance	Colour
	Lens shading	Colour
	Color mapping	Color
	Encoding/Compression	Blur
	Black level	Noise

the same difficulties for the system. To overcome this problem and build a common scale, we use the Structural Similarity (SSIM) metric proposed by [6], a full-reference quality metric that measure the similarity between an original image and its transformed counterpart, in terms of luminance, constrast and structure. The SSIM value is between 0 and 1, where a SSIM of 1 corresponds to the case where the two images being compared are identical.

Finally, in order to compare and validate our results, we provide baseline results for a set of common object detection models, including Faster R-CNN, RetinaNet, Cascade R-CNN, FCOS, Deformable ConvNets v2, Libra R-CNN, HRNet, and Sparse R-CNN. Those models were trained and evaluated on the BDD100k dataset [8], which is a dataset consisting of 100,000 videos of driving scenes recorded under different conditions (change of weather, time of day, and city).

3.3 Experimental results

Before going into the details of the results for the SUT over all transformations, a comparison is made with the baseline models. Due to space constraints, we choose to present the results for only one transformation, Gaussian blur, in Figure 2. Nevertheless, the observed behaviour is similar whatever the transformation considered. The models are compared on the basis of the average SSIM value on their respective datasets (BDD100K for the baseline models, Mango for the SUT). The results for the baseline models are roughly identical, which can be explained by a similar architecture, based on R-CNN, and training on the same dataset, BDD100K. In contrast, the results for the SUT differ significantly: the slope of the performance is much lower at high SSIM (i.e. for low image alterations), which reflects a certain robustness, even though the performance of the SUT on its original dataset is worse than that of the baseline models on their original dataset. Note also that the transformation is applied with the same parameter range, but this does not necessarily result in the same SSIM. For example, the first point corresponds to the same transformation parameters, yet the SSIM on BDD100K is much higher than on Mango. This could be explained by the nature of the dataset (variety of situations, format, etc.), but further experimentation is needed to conclude on this subject.

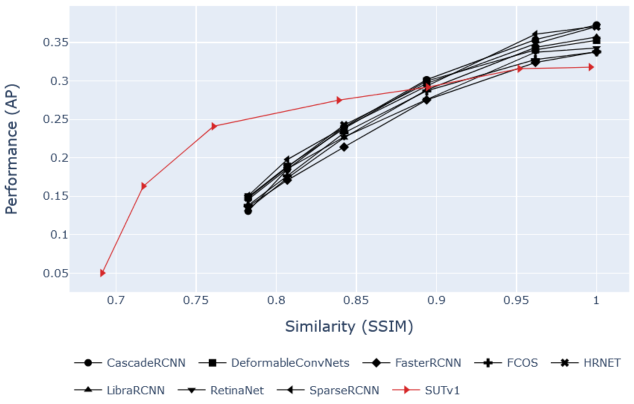

Figure 2: Comparison between the SUT and the baseline models against Gaussian blur.

Figure 3 shows the results obtained for the SUT against the transformations defined in the previous section. This behaviour is similar in the case of a baseline model (Faster R-CNN). Overall, it can be observed that the lower the SSIM, the lower the AP value. However, two areas with significantly different average slopes can be distinguished. The area with the lowest average slope is mostly related to Noise transformations: on average, the detection model is quite robust to electronic noise. On the other hand, the performance degrades rapidly in the case of Blur or Colour transformations. This can be explained by the fact that an object detection model is very sensitive to edges and contours. A Blur transformation has an immediate effect on this characteristic. The same is true of some colour transformations, such as chromatic aberration, which shifts the contours between the RGB layers. In the case of the SUT, it can be seen that the brightness (GammaCorrection and SaltNoise curves) also plays a very important role: this is because the dataset images already have a high exposure, and by increasing the brightness a little bit, the object edges quickly disappear. While brightness and colour issues can be adjusted during the digital processing phase, it is more difficult to correct the blurring created during acquisition: this is where the greatest care needs to be taken.

To overcome these issues, methods exist to improve the intrinsic robustness of detection models. For instance, in the context of adversarial examples, Adversarial Training [3] consists of augmenting the training dataset with adversarial examples and then retraining the model to improve its performance and its robustness against new adversarial examples. In the same way, we are interested in the effect of such data augmentation, based on the transformations, in the case of the SUT. Thus, images altered by the Gamma Correction transformation were added to the training dataset, and the detection model was retrained. We observed an improvement in robustness against the corresponding transformation, which was reflected by a translation of the curve along the y-axis. These results should be

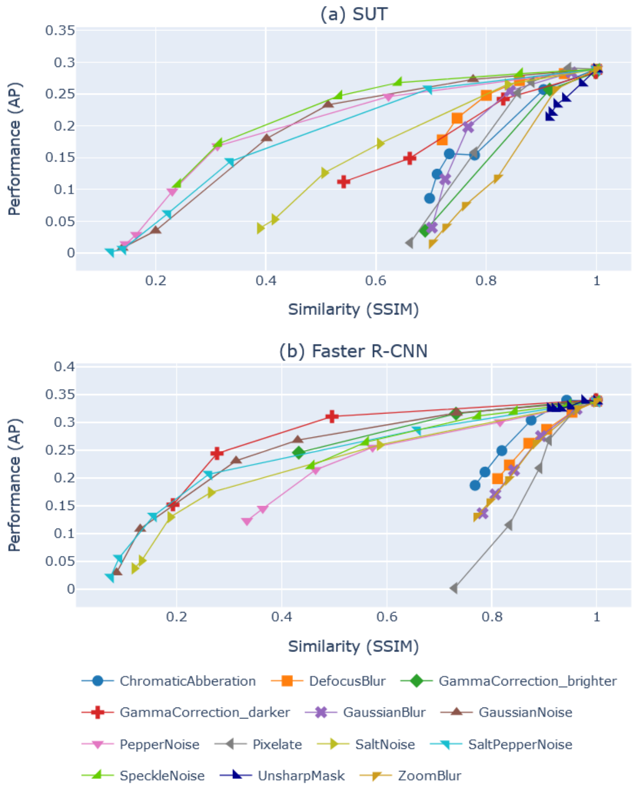

Figure 3: Overall behaviour of (a) SUT, and (b) Faster R-CNN when confronted with the transformations

treated with caution, as two successive trainings can give significantly different results, but they suggest that the data augmentation solution is a good avenue for robustness improvement.

4 LESSONS LEARNED

The main issue when using transformations to test a model is selecting the most appropriate ones. It is necessary to validate that they have a physical reality or that they are representative of a physical change, which often requires domain knowledge. In our experiment, we only test transformations separately, but it would be necessary to also study the behaviour of the system when faced with combinations of transformations, which would be all the more representative of real situations.

Comparing the performance of models trained on different datasets can also be a real challenge. The choice of a full reference metric such as SSIM avoids this problem, as it compares degrees of similarity to the original dataset. However, we observed that if the range of transformations parameters applied on two different datasets is the same, it may not be the case of the average SSIM. This could be related to the nature of the dataset (variety of situations, format, etc.), but further experimentation is needed to conclude.

We used the AP metric to evaluate the performance of the detection models and defined their robustness according to the evolution of this performance. However, it is questionable to what extent this approach is sufficient for the study of robustness. In particular, it would be interesting to study the type of error introduced by a transformation (misclassification, mislocation, etc.). In addition, papers report the study of model quality metrics, initially focused on adversarial examples, but which could be applied to our case study [7].

5 CONCLUSION AND FUTURE WORK

In the context of computer vision, several approaches to evaluate the robustness of a system have been proposed in the literature, to ensure that they are indeed trustworthy under different operating conditions. In this paper, we report the results of robustness testing on an industrial case of a safety-critical object detection system based on machine learning.

The performance of the system against transformations representative of changes in sensor or digital processing parameters is evaluated. The strengths of the system are thus identified, as well as areas where there is still room for improvement, notably through data augmentation.

Future work will focus on further developing of this case with respect to the limitations mentioned, as well as on investigating of new metrics for robustness assessment that provide more information about the errors made by the system.

ACKNOWLEDGMENTS

This work has been supported by the French National Agency of Research and Technology (ANRT) [CIFRE N°2020/0754], by the European Commission (FEDER) and Junta de Andalucia under projects EKIPMENT-PLUS (P18-FR-2895) and APOLO (US-1264651), and by the Spanish Government (FEDER/Ministerio de Ciencia e Innovación – Agencia Estatal de Investigación) under project HORATIO (RTI2018-101204-B-C21).

REFERENCES

[1] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR '15)*. OpenReview.net, San Diego, CA, USA, 303–314.

[2] Dan Hendrycks and Thomas Dietterich. 2019. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the 7th International Conference on Learning Representations (ICLR '19)*. OpenReview.net, New Orleans, LA, USA.

[3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR '18)*. OpenReview.net, Vancouver, BC, Canada.

[4] Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. 2017. Towards Practical Verification of Machine Learning: The Case of Computer Vision Systems. arXiv:1712.01785 [cs.CR]

[5] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. 2018. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. In *Proceedings of the 40th International Conference on Software Engineering (ICSE '18)*. Association for Computing Machinery, New York, NY, USA, 303–314.

[6] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. 2004. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Transactions on Image Processing* 13, 4 (April 2004), 600–612.

[7] Zhou Yang, Jieke Shi, Muhammad Hilmi Asyrofi, and David Lo. 2022. Revisiting Neuron Coverage Metrics and Quality of Deep Neural Networks. arXiv:2201.00191 [cs.SE]

[8] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. 2018. BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning. arXiv:1805.04687 [cs.CV]