

## APPLIED RESEARCH

# Tolerate Failures of the Visual Camera With Robust Image Classifiers

MUHAMMAD ATIF<sup>ID</sup>, ANDREA CECCARELLI<sup>ID</sup>, TOMMASO ZOPPI<sup>ID</sup>,  
AND ANDREA BONDAVALLI<sup>ID</sup>, (Senior Member, IEEE)

Department of Mathematics and Informatics, University of Florence, 50142 Florence, Italy

Corresponding author: Tommaso Zoppi (tommaso.zoppi@unifi.it)

This work was supported in part by the H2020 Programme under the MarieSkłodowska-Curie (ADVANCE) Project 823788, and in part by the Regione Toscana SPaCe Project POR FESR 2014-2020.

**ABSTRACT** Deep Neural Networks (DNNs) have become an enabling technology for building accurate image classifiers, and are increasingly being applied in many ICT systems such as autonomous vehicles. Unfortunately, classifiers can be deceived by images that are altered due to failures of the visual camera, preventing the proper execution of the classification process. Therefore, it is of utmost importance to build image classifiers that can guarantee accurate classification even in the presence of such camera failures. This study crafts classifiers that are robust to failures of the visual camera by augmenting the training set with artificially altered images that simulate the effects of such failures. Such a data augmentation approach improves classification accuracy with respect to the most common data augmentation approaches, even in the absence of camera failures. To provide experimental evidence for our claims, we exercise three DNN image classifiers on three image datasets, in which we inject the effects of many failures into the visual camera. Finally, we applied eXplainable AI to debate why classifiers trained with the data augmentation approach proposed in this study can tolerate failures of the visual camera.

**INDEX TERMS** Visual camera failures, deep learning, data augmentation, robustness, traffic sign recognition.

## I. INTRODUCTION

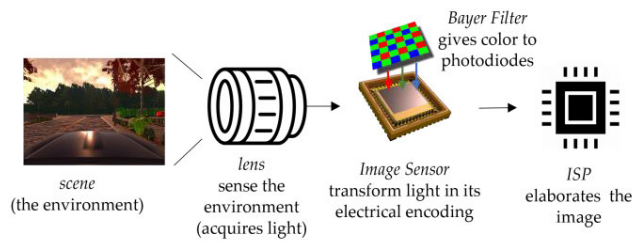
The popularity of Deep Neural Networks (DNNs) for image classification has grown enormously in the last decade, and they have been successfully applied in multiple domains such as computer vision [21], autonomous driving [12], bioinformatics [25], medical image analysis [1], and climate science [24]. The excellent performance of image classifiers depends heavily on i) the training phase, where the classifier learns how to classify images [27], and ii) the validity of the assumption that the data to be classified at test time comes from the same distribution as that of the training set [36].

When deployed in a real system, DNNs may face unexpected operational conditions that lead to out-of-distribution data due to unexpected or unknown behaviors. The images captured by the visual camera may be altered due to a

multitude of reasons [32]: this paper accounts for possible failures of the visual camera, which may be due to internal (e.g., failures of the electrical parts), external (e.g., dirt or scratched lenses), or environmental (e.g., rain or icing on lenses) factors. We consider a complete set of visual camera failures resulting from a Failure Mode and Effect Analysis [22] to embrace a complete set of visual camera failures known to date, according to the literature.

Visual camera failures may occur frequently when a visual camera is placed in its operational setup, particularly when it is placed outside. Therefore, this study shows how to build image classifiers that are robust to visual camera failures, so that altered images delivered to the DNN will hardly compromise the image classification task. To achieve these goals, we developed a methodology based on data augmentation [2] that builds image classifiers whose classification performance suffers only from minimal degradation when processing altered images. We trained DNNs using a clean

The associate editor coordinating the review of this manuscript and approving it for publication was Wenming Cao<sup>ID</sup>.



**FIGURE 1. Main components of a Visual Camera: the Lens, and the camera body composed of Bayer Filter, Image Sensor, and Image Signal Processor.**

training set that contains only images from the original datasets (we refer to them as regular classifiers, which are built using the usual scaling and translation approaches for data augmentation) and an augmented training set that includes images from the original datasets plus altered images in which we injected the effects of visual camera failures (we refer to the resulting classifiers as augmented classifiers).

We assessed this methodology by considering the deployment of image classifiers in autonomous vehicles for Traffic Sign Recognition (TSR), which may be significantly affected by visual camera failures [22], [28], [33]. We gathered three well-known TSR datasets, namely the German Traffic Sign Recognition Benchmark (GTSRB, [10]), the Belgium Traffic Sign (BelgiumTSC, [11]), and the Dataset of Italian Traffic Signs (DITS, [12]), applied AlexNet [7], MobileNetV2 [9], and Inceptionv3 [8] DNNs, which have wide application in the TSR domain [3], [4], [5], and injected a total of 13 visual camera failures under different configurations.

Our experimental campaign revealed that augmented classifiers have far better classification accuracy than regular classifiers when processing altered images and also on clean images; for example, the accuracy of the clean images increased in the three datasets from 0.994 to 1 (from 82 to no misclassifications out of 12570 images), from 0.96 to 0.992 (from 46 to 9 misclassifications out of 1159 images), and from 0.997 to 0.999 (from 7 to 3 misclassifications out of 2505 images). Furthermore, we explain our results using the LIME [30] framework, which explains why our augmented classifiers output fewer misclassifications than the regular classifiers. Briefly, augmented classifiers select a few strong features compared to regular classifiers, which instead select many weak features that individually do not contribute much to classification.

The remainder of this paper is organized as follows. Section II provides background information on visual camera failures, letting Section III review related works on DNN robustness. Section IV describes the experimental methodology. Section V presents and discusses the results of the experimental campaign and elaborates on the robustness of regular and augmented image classifiers. Section VII details the limitations of this study, and Section VIII concludes the paper and presents future work.

## II. BACKGROUND ON VISUAL CAMERA FAILURES

Misbehavior(s) of the visual camera may generate altered images that are delivered to the image classifier. Fig. 1 depicts the main components of a visual camera [23], where camera failures may occur. The lens senses a scene from the environment in the form of light. This light is processed by an Image Sensor, whose photodiodes transform the light in its electrical encoding to produce a raw file [29]. The Bayer Filter, which acts on top of the Image Sensor, colors the photodiodes into a red-green-blue (RGB) pattern. The Image Signal Processor (ISP) processes the raw file to produce a digital image; it has multiple functions, including demosaicing, noise reduction, image sharpness correction, lens distortion correction, chromatic aberration correction, image compression, and JPEG encoding [23].

While many works, such as [35] elaborate on the effect of modified images on the classification process, only a few studies in the literature focus on the effects that failures of the visual camera may have on the produced image and consequently on the image classification. Even if the risk of accidental alterations of the output image of the camera is acknowledged as realistic [37], this consideration is usually ancillary to the main contribution of this study. Examples are [6], where the authors focus on environmental conditions and build a DNN that implements an attention mechanism for performance improvement, and [38], which explores how sensors respond when used in real circumstances, as well as to confirm the impacts of environmental conditions on driving scenarios.

The most comprehensive work on visual camera failures is [22], where the authors systematically identified the failure modes and effects of a visual camera through the application of a Failure Mode and Effects Analysis (FMEA). The effects of camera failures on the output images are summarized in Fig. 2. Failures are caused by malfunctions of the lens, Image Sensor, Bayer Filter, or ISP, and belong to the following categories.

- *Banding* (Fig. 2b). A banded image contains many vertical and/or horizontal lines that are visible in the background.
- *Black Pixels* (Fig. 2c). The frame delivered to the camera body may contain dark spots due to the anomalous behavior of the Image Sensor. The effect is the visualization of black or black spots on the image.
- *Blurring* (Fig. 2d). If the focus of the lens is incorrect, the image may have visible blur.
- *Brightness* (Fig. 2e). The brightness of the image can be altered, ranging from a fully black frame (no brightness) to a fully white frame (maximum brightness).
- *Broken Lens* (Fig. 2f). Scratches, lines, or black areas may appear in the image when one or more lenses of the camera break.
- *Condensation* (COND, Fig. 2g). Halos may appear in images affected by the condensation failure.



**FIGURE 2.** Visual effects of camera failures on a sample traffic sign image (best viewed in color).

- *Dirt (Fig. 2h)*. Dirt on the external or internal lens of the camera may create a wide variety of alterations in the images; most likely, the image will have scattered black spots or areas in which the colors differ from the clean image.
- *Ice (Fig. 2i)*. Ice crystals may be visible in images when the temperature drops to the freezing point.
- *No Bayer Filter (NOBF, Fig. 2j)*. When the Bayer filter does not work properly, the image will likely end up having altered colors or even being colorless at all.
- *No Chromatic Aberration Correction (NCAC, Fig. 2k)*. Halos appear at the corners and edges of the image, and blur the outer edges.
- *No Demosaicing (NDEMOS, Fig. 2l)*. If the demosaicing process fails, the image becomes pixelated: in this case, each pixel is painted as a mosaic of RGB colors.
- *No Noise Reduction (NNR, Fig. 2m)*. If the noise-removal component is malfunctioning, the image may include excessive noise.
- *Rain (Fig. 2n)*. Drops of water on external or internal lenses can produce images with small circles at random positions.

### III. RELATED WORKS ON ROBUST IMAGE CLASSIFIERS

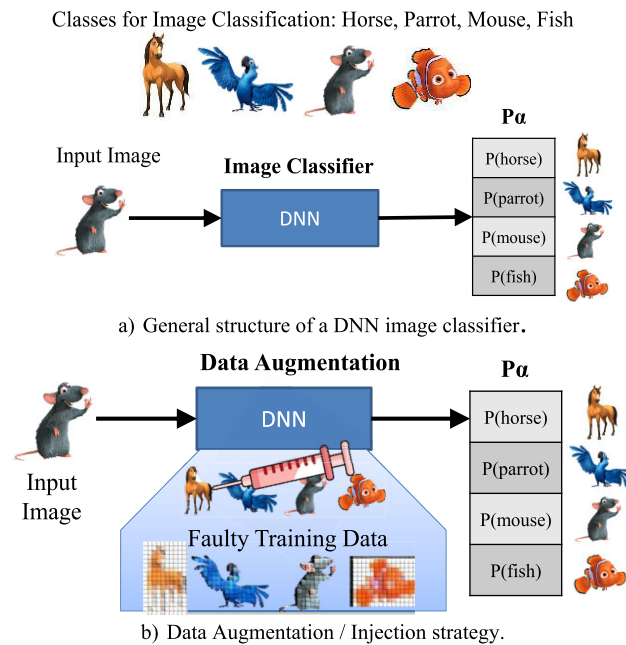
The robustness of a DNN image classifier aims at *sustaining the performance of the model under various image corruptions or alterations* [40]. In our study, image corruption

and alterations were due to visual camera failures, leading to the effects discussed in Section II. We now review the robustness approaches organized in architectural solutions for building robust DNNs, out-of-distribution detection, and data augmentation strategies for robustness and adversarial defense, explaining how this paper differs from each category.

#### A. ARCHITECTURAL SOLUTIONS FOR ROBUST DNNs

Several recent proposals have been made for robust image classifiers. In [14], the authors analyzed the classification performance of several DNNs for images degraded by Gaussian noise, blurring, and compression. They proposed a two-step (master-slave) process, in which the master classifies the quality of the degraded input image, which is then used to select the most suitable slave DNN for classification. In [18], the authors provided a review of methods that combine two or more images and pre-process them to delete specific regions of the images, which may cause the DNN to lean towards misclassifications. Experimental results show that this method builds image classifiers that are more robust and have better classification performance, even when dealing with altered images. In [33], the authors estimated the confidence of a DNN in response to unexpected execution contexts.

With respect to the surveyed works, we do not add any additional component in the architecture for the purpose of detecting outliers in the images.



**FIGURE 3.** Visual explanation of a DNN image classifier, and data augmentation strategy to improve robustness.

### B. OUT-OF-DISTRIBUTION DETECTION

Many studies, such as [34] and [36] have aimed to detect out-of-distribution samples without knowledge of the type of image alteration in the training data. In particular, [34] trained four different DNNs with three different supervisors at various stages of training to detect the point at which supervisors' performance began to decline during training.

This study assumes knowledge of the failure effects at the training time, as discussed in Section II.

### C. DATA AUGMENTATION FOR ROBUSTNESS

Another group of studies aimed to achieve DNN robustness through data augmentation strategies [13]. The authors of [15] used data augmentation to improve the generalization capability of DNNs using smoothness regularization against perturbations to improve the classification performance. In another study [16], the authors employed a Pixel Mask to diminish the sensitivity of DNNs to image corruption. Moreover, a previous study [17] proposed a data augmentation pipeline to accelerate MRI reconstruction.

Instead, our study focuses on ICT systems that employ one or more visual cameras. To the best of our knowledge, no study has applied data augmentation to tolerate many failures of the visual camera. To address this gap in the literature, we consider a complete set of failures that may occur due to malfunctions in the visual camera.

### D. DATA AUGMENTATION FOR ADVERSARIAL DEFENSE

Many recent studies on data augmentation have set the goal of adversarial defenses, that is, defending the encompassing system from images explicitly designed by an attacker to fool

a classifier, and output a wrong class. Briefly, these studies increased the robustness of the target classifier, which was trained using genuine and adversarial images [39].

Even if the approach is conceptually similar to ours, our study does not consider adversarial activity as a potential source of altered images; therefore, these are only marginally related works and are interesting to this study only in terms of the approach they follow.

## IV. METHODOLOGY FOR A CLASSIFIER ROBUST TO CAMERA FAILURES

We describe our data augmentation approach with the aid of Fig. 3. Traditionally, an image classifier outputs a set  $P\alpha$  of  $\alpha$  probabilities, where each  $P_i$  represents the probability of the input image belonging to class  $i$ ,  $0 \leq i < \alpha$ . The graphical example in Fig. 3a sketches a 4-class classification problem ( $\alpha = 4$ ), where the image classifier processes the input and outputs an array  $P\alpha$  of 4 probabilities, the highest of which points to the class the classifier will predict for the animal in the image. Fig. 3b improves the training phase of the DNN, aiming at a more robust model. The training set is augmented with perturbations of the clean images from the training set, providing a broader set of images to the DNN. Perturbations are obtained by injecting the effects of visual camera failures into clean images, thereby saving the corrupted image as a separate item.










This approach was implemented as follows. In Section IV-A, we gathered three datasets for classification, which were images of the traffic signs. In Section IV-B we describe how we created altered images from the clean images of the three datasets. Section IV-D reports the three DNNs used to perform image classification (TSR). The training and testing processes are described in detail in Section IV-C. All the code we developed is available at [44]; all the altered images we created are available at [45].

### A. DATASETS FOR TRAFFIC SIGN RECOGNITION (TSR)

We select the GTSRB [10], BelgiumTSC [11], and DITS [12] which are public datasets that contain sequences of images of different categories of traffic signs from different countries. The images in the GTSRB dataset have heterogeneous lighting conditions and distance from the camera. The DITS dataset is considered more difficult than others, because it contains images captured under different lighting and environmental conditions. Instead, images of the BelgiumTSC dataset were captured by multiple visual cameras from different viewpoints.

Each dataset has its own categorization of traffic signs based on the color, shape, and content of the traffic sign, which changes depending on the country from which the images were sampled. Overall, we identified 9 overlapping categories of traffic signs, as listed in Table 1. The GTSRB dataset contains images belonging to 8 categories out of the 9 in Table 1, missing the blue rectangular traffic sign images (Class 8). The BelgiumTSC dataset has 8 categories of traffic signs, as shown in Table 1, with missing white

TABLE 1. Categorization of traffic signs in nine categories.

Category	1	2	3	4	5	6	7	8	9
Traffic Signs									
	Stop Sign	Red Triangular	Inverted Triangular	Speed Limit	Red Circular	Blue Circular	Diamond	Blue Rectangular	White Circular
BelgiumTSC								✓	
GTSRB	✓	✓	✓	✓	✓	✓	✓		✓
DITS								✓	✓

circular images (Class 9), whereas DITS contains traffic signs for all categories.

**B. INJECTION OF VISUAL CAMERA FAILURES**

We injected 13 different visual camera failures into the images contained in the three datasets. Each failure had multiple configurations for a total of 103. We set up a Python script to inject each failure, with the configurations planned below.

- *Banding*. We created 3 different configurations by overlapping the banding images using a blend function.
- *Black Pixels*. Different configurations, such as a single black line, horizontal and vertical black lines, and sets of [50, 200, 500, 1000] pixels, were randomly selected and turned black in the traffic sign image.
- *Blurring*. We employed 12 different configurations to produce blurred images with varying degrees of blur using the OpenCV [41] blur function, with settings in the range [25], [35].
- *Brightness*. We simulated 8 different levels of brightness, from a very dark (brightness 0) to an almost white image (brightness value 15). We used the brightness values as [15, 0.1, 0.3, 0.6, 1.5, 6, 7.5, 10].
- *Broken Lens*. We simulated the broken-lens effect using 15 superimposed images from [19].
- *Condensation (COND)*. We overlaid three different condensation images to clean images and simulate the condensation effect.
- *Dirt*. We used 36 dirt images from [19] that we overlaid to clean images and simulate the dirt effect.
- *Ice*. We overlaid different ice images onto clean images to create four different altered images for each clean image.
- *No Bayer Filter (NOBF)*. We simulated this failure by converting each image into grayscale.
- *No Chromatic Aberration Correction (NCAC)*. We created chromatic aberration failures using the code in [26].
- *No Demosaicing (NDEMOS)*. The image was processed using cv2 [43] and resized using the PIL [42] module.

- *No Noise Reduction (NNR)*. We used 10 different configurations of speckle noise with different levels of noise.
- *Rain*. We simulated the rain effect by overlaying 5 different rain images with traffic sign images.

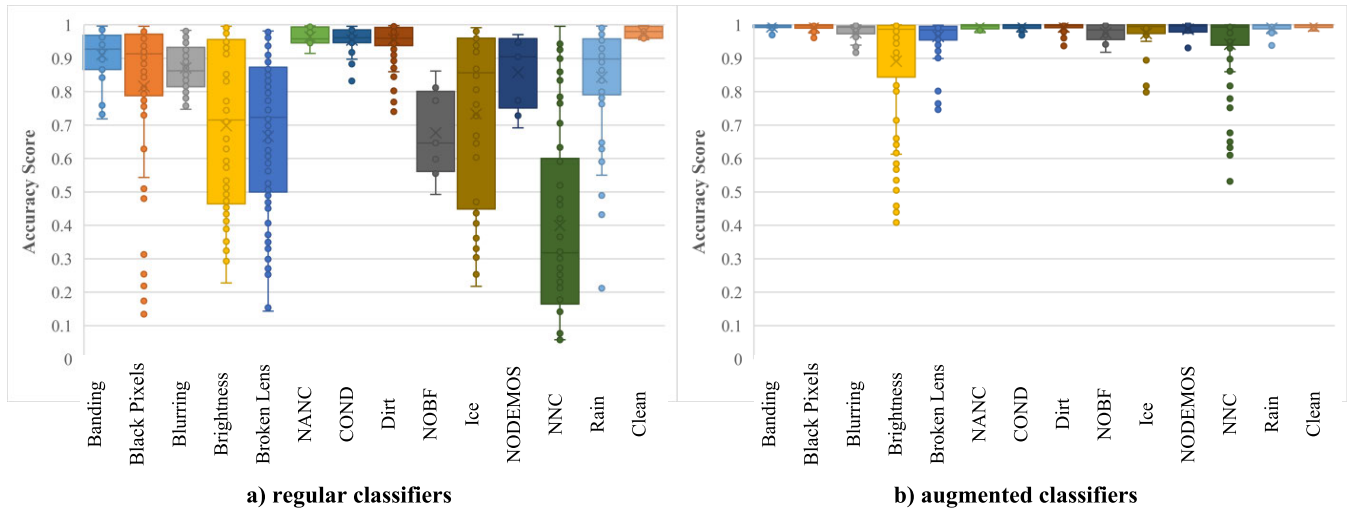
**C. BUILDING TRAIN AND TEST SETS FOR CLASSIFIERS**

We collected the GTSRB, DITS, and BelgiumTSC datasets from their repositories, and processed each image to inject 103 failure configurations from Section IV-B. This created 103 variants in addition to the original dataset, for a total of 104 datasets. Each of these 104 datasets was split into training and test sets; the split was identical for all datasets.

We refer to *train\_clean/test\_clean* as the sets containing the train/test split of the original dataset; both *train\_clean* and *test\_clean* contain only clean images. We call *train\_altered/test\_altered* the train/test set, which contains *train\_clean/test\_clean* and train/test splits of all 103 variants of the original dataset. For the sake of our analysis, we have to understand how each of the six classifiers deals with images corrupted with different configurations of the same visual camera failure. As such, we grouped the 103 variants based on the 13 failures we used to generate them. This allowed the creation of 13 groups of test splits of variants, as follows:

- Failures *NOBF*, *NCAC*, and *NDEMOS* were each used to create a single variant because they had only a single configuration of failure.
- Failures *Banding* and *COND* were used to create three variants each.
- *Ice*, *Rain*, *Black Pixels*, *Brightness*, *NNR*, *Blurring*, *Broken Lens*, and *Dirt* failures were responsible for generating 4, 5, 6, 8, 9, 11, 15, and 36 variants, respectively.

Each of the 13 groups above contains all the variants generated using each of the 13 visual camera failures. Different groups differ in the number of variants and images they contain, which depends on the number of failure configurations. Noticeably, testing a classifier for each group provides a set of classification metric scores, one for each test split of a variant in the group. For example, testing a classifier with variants in the *Ice* group will output 4 metric scores,



**FIGURE 4.** Box Plots showing the accuracy of the three regular classifiers (Fig. 4a) and the three augmented classifiers (Fig. 4b) on the three datasets, for the *test\_clean* and the 13 groups of variants related to single failures.

whereas testing a classifier with variants in the *Banding* group will generate only 3 metric scores because our study considers 3 configurations of the *Banding* failure.

#### D. DNNs FOR TRAFFIC SIGN RECOGNITION

The image classifiers in this study are three DNNs that were pre-trained on the ImageNet dataset: AlexNet (AN, [7]), InceptionV3 (IC3, [8]), and MobileNetV2 (MN2, [9]). These three DNNs have different characteristics: AN is a rather old DNN but is considered a milestone, as in 2012 it achieved relevant results in the ImageNet Large Scale Visual Recognition Challenge [31]. IC3 stems from GoogleNet and outperforms AN on the ImageNet dataset used for training [31], whereas MN2 is based on an inverted residual structure [9] as opposed to traditional residual models that use expanded input representations.

We adapted these pre-trained image classifiers to our TSR case study using transfer learning with a batch size of 32 and 10 epochs, utilizing the stochastic gradient descent with momentum (*sgdm*, [20]) optimizer and cross-entropy loss at the softmax layer on each of the GTSRB, DITS, and BelgiumTSC datasets. We retrained each DNN (AN, IC3, and MN2) twice on each of the three datasets as follows:

- *regular classifier*: We performed transfer learning using the *train\_clean* set of each of the three datasets. We employed classical techniques such as image scaling and translation to limit overfitting.
- *augmented classifier*: created using the *train\_altered* set as the data baseline for transfer learning, without image scaling or translation, to compare the robustness of camera failure data augmentation against traditional data augmentation techniques such as scaling and translation.

This process creates a total of 18 classifiers, and for each of the 3 datasets, we created a regular and augmented classifier for AN, MN2, and IC3 DNNs. We then used the *test\_clean*

and *test\_altered* sets to test each classifier and quantify their classification performance.

#### V. EXPERIMENTAL RESULTS

We analyzed the classification performance of regular and augmented classifiers and measured the extent to which the augmented image classifiers were more robust to visual camera failures than regular classifiers. Experiments were performed on an Intel(R) Core (TM) i5-8350U CPU@1.7 GHz 1.9 GHz, using an NVIDIA Quadro RTX 5000 GPU.

##### A. IMPROVED ROBUSTNESS TO CAMERA FAILURES

We compared the classification performance of regular classifiers against augmented classifiers, as shown in Fig. 4. The figure is composed of two series of box plots. On the left (Fig. 4a), we show the average accuracy of regular classifiers such as AN, MN2, and IC3 on all three datasets against each failure. On the right (Fig. 4b), the accuracy of the augmented classifiers is presented; each figure contains 14 box plots (*test\_clean* and 13 groups of variants from Section IV-C). On the vertical axis, we have accuracy scores: the higher the accuracy, the better is the classification performance.

The two plots in Fig. 4 appear very different from each other: the boxes in Fig. 4a span across a wider range of accuracy scores with respect to Fig. 4b. This means that regular classifiers have more variability in their classification performance than augmented classifiers. Most importantly, augmented classifiers have almost maximum accuracy, which is desirable for any classification task.

Two additional trends are evident in Fig. 4. First, the accuracy improvement of the augmented against regular classifiers is not constant across different test sets (and visual camera failures). The accuracy is almost perfect ( $\sim 1$ ) when using augmented classifiers for *test\_banding*, *test\_blackpixels*, *test\_NDEMOS*, and *test\_rain*, whereas

**TABLE 2.** Accuracy achieved using regular classifiers on the three datasets. (Accuracy scores report ranges, when a failure is injected using multiple configurations).

Test Set or Group of Variants (# Config.)	GTSRB		DITS		BelgiumTSC	
	Accuracy	Best DNN	Accuracy	Best DNN	Accuracy	Best DNN
Clean	0.994	MN2	0.96	IC3	0.997	AN
Banding (3)	[0.947, 0.963]	AN	[0.902, 0.952]	IC3	[0.991, 0.996]	AN
Black Pixels (6)	[0.875, 0.993]	MN2	[0.849, 0.956]	MN2	[0.940, 0.995]	AN
Blurring (11)	[0.946, 0.983]	MN2	[0.780, 0.837]	IC3	[0.862, 0.970]	AN
Brightness (8)	[0.592, 0.959]	AN	[0.359, 0.941]	AN	[0.324, 0.996]	AN
Broken Lens (15)	[0.469, 0.836]	AN	[0.607, 0.852]	AN	[0.465, 0.981]	AN
COND (3)	[0.974, 0.993]	MN2	[0.932, 0.955]	IC3	[0.988, 0.995]	AN
Dirt (36)	[0.970, 0.994]	MN2	[0.845, 0.959]	IC3	[0.978, 0.997]	IC3
Ice (4)	[0.729, 0.960]	AN	[0.362, 0.938]	AN	[0.603, 0.988]	AN
NOBF (1)	0.861	MN2	0.812	IC3	0.598	MN2
NCAC (1)	0.993	MN2	0.949	IC3	0.996	IC3
NDEMOS (1)	0.963	AN	0.774	IC3	0.97	IC3
NNR (9)	[0.419, 0.959]	AN	[0.317, 0.903]	AN	[0.228, 0.995]	AN
Rain (5)	[0.864, 0.958]	AN	[0.790, 0.939]	AN	[0.864, 0.992]	AN

**TABLE 3.** Accuracy achieved using augmented classifiers on the three datasets. (Accuracy scores report ranges, when a failure is injected using multiple configurations).

Test Set or Group of Variants (# Config.)	GTSRB		DITS		BelgiumTSC	
	Accuracy	Best DNN	Accuracy	Best DNN	Accuracy	Best DNN
Clean	1.000	MN2	0.992	MN2	0.999	AN
Banding (3)	[0.999, 0.999]	IC3	[0.991, 0.993]	MN2	[0.998, 0.999]	IC3
Black Pixels (6)	1.000	MN2	[0.987, 0.994]	MN2	[0.997, 0.999]	IC3
Blurring (11)	[0.996, 0.999]	IC3	[0.975, 0.983]	MN2	[0.997, 0.999]	IC3
Brightness (8)	[0.990, 1.000]	MN2	[0.458, 0.992]	MN2	[0.668, 0.999]	IC3
Broken Lens (15)	[0.902, 0.997]	IC3	[0.948, 0.991]	MN2	[0.996, 0.998]	IC3
COND (3)	1.000	MN2	[0.988, 0.994]	MN2	[0.998, 0.999]	IC3
Dirt (36)	1.000	MN2	[0.972, 0.995]	MN2	[0.997, 0.999]	IC3
Ice (4)	1.000	MN2	[0.894, 0.993]	MN2	[0.976, 0.999]	IC3
NOBF (1)	1.000	MN2	0.982	MN2	0.986	MN2
NCAC (1)	1.000	MN2	0.992	MN2	0.998	IC3
NDEMOS (1)	1.000	MN2	0.983	MN2	0.998	AN
NNR (9)	[0.999, 1.000]	MN2	[0.649, 0.993]	MN2	[0.859, 0.998]	AN
Rain (5)	1.000	MN2	[0.988, 0.993]	MN2	[0.997, 0.998]	IC3

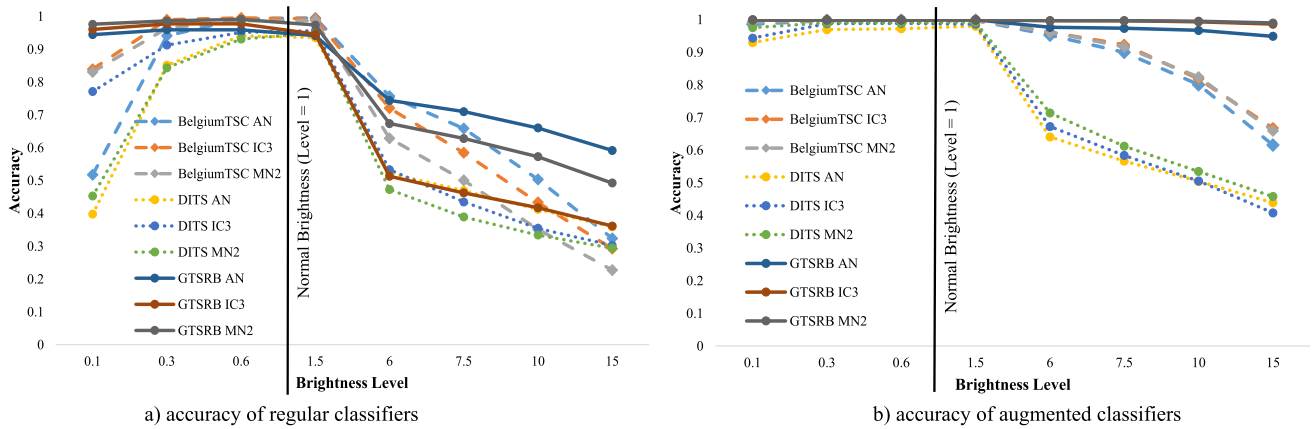
regular classifiers struggle. Overall, the boxes in Fig. 4b are narrow and close to the top of the plot. For most of the other test sets, there was still a clear improvement when using the augmented classifiers, whose accuracy was close to 1.0.

Second, the accuracy improved when classifying the *test\_clean* set with no visual camera failures. The last box in Fig. 4a (*test\_clean*) was wider than the corresponding box in Fig. 4b. All augmented classifiers are excellent when processing clean images; instead, the accuracy scores of the regular classifiers have more variance than the augmented classifiers.

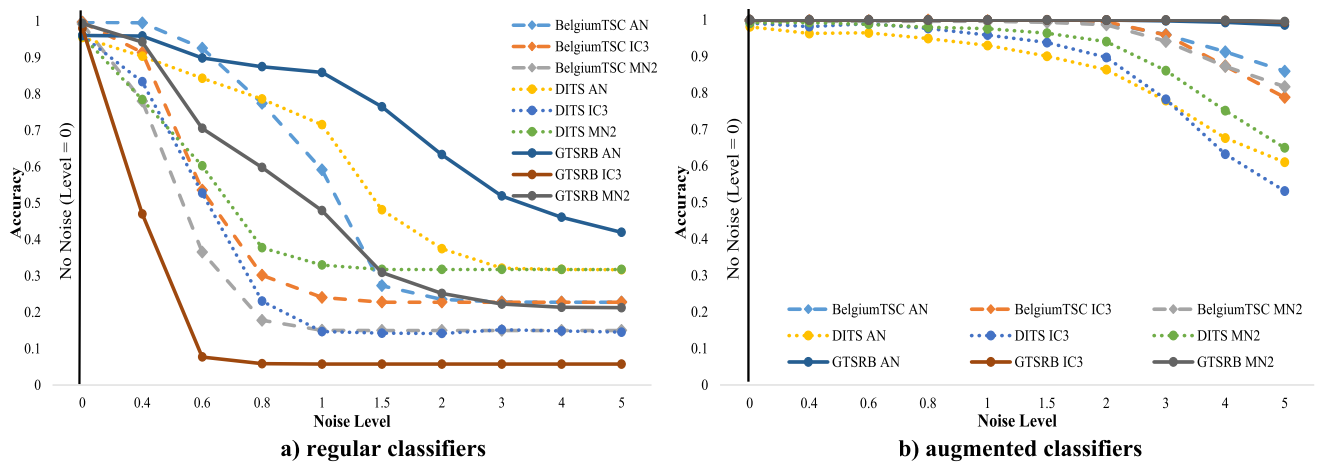
We further explored the accuracy scores in Table 2 and Table 3. Both tables report a row for each test set and three groups of columns: one for each dataset. For each dataset, we reported the highest accuracy and the DNN that achieved it. Accuracy scores in the table often report ranges, which occur when a failure is injected using multiple configurations, thus accounting for many dataset variants (see Section IV-C). Instead, *test\_clean* contains no failures, and *test\_NCAC*, *test\_NOBF*, and *test\_NDEMOS* contain a single configuration for a failure; they have a single accuracy value in both Table 2 and Table 3.

The regular classifiers in Table 2 always have inferior accuracy compared with the corresponding augmented classifiers in Table 3. Looking at the GTSRB column in Table 3, we can see that in many cases, even in the presence of visual camera failures, the classification accuracy is perfect (1.0). However, this trend does not apply to the other two datasets, which are harder to classify; neither regular nor augmented classifiers reach 1.0 accuracy on *test\_clean* (first row of Table 3). As such, we cannot expect perfect accuracy for altered images.

Some test sets resulted in low accuracy, regardless of the training dataset and DNN. This is the case for *test\_brightness*, *test\_NNR*, *test\_ice*, and to a lesser extent, *test\_brokenlens*. The range of accuracy of *test\_ice* on the DITS and BelgiumTSC datasets is broad, even when adopting augmented classifiers; for some configurations of the ice visual camera failure, accuracy in DITS drops as low as 0.894 even when using the augmented classifiers. This is because the ice image overlays the image of the traffic sign image; in some cases, it covers significant parts of the image, whereas sometimes it does not. On BelgiumTSC, the Ice failure makes drop as low as 0.976 for specific configurations, which is a decent



**FIGURE 5.** Accuracy achieved on test\_brightness by regular classifiers (left) and augmented classifiers (right) using the 3 DNNs AN, MN2, IC3 on the 3 datasets DITS, BelgiumTSC, GTSRB and brightness levels in the range [0.1 to 15].



**FIGURE 6.** Accuracy achieved on test\_NNR by regular classifiers (Fig. 6a, on the left) and augmented classifiers (Fig. 6b, on the right) using different DNNs on different datasets with different levels of noise (0 to 5).

accuracy in general terms, but still lower than that achieved in the other test sets, except for test\_brightness and test\_NNR.

Furthermore, Table 2 and Table 3 show which DNN results in the highest accuracy value for each test set using regular and augmented classifiers, respectively (i.e., columns Best DNN in the tables). There was no clear winner for the regular classifiers listed in Table 2. Considering the augmented classifiers in Table 3, the best performing DNN is MN2, which is always the preferred DNN for the DITS dataset and its variants, as well as for the clean (initial) dataset and 10 out of 13 groups of variants of GTSRB (see Table 3). For BelgiumTSC, it appears to be more beneficial to adopt IC3. The regular AN often achieves better classification performance than regular MN2 and IC3; however, the augmented AN classifier shows only marginal improvement in accuracy, whereas augmented MN2 and IC3 outperform regular MN2 and IC3, as well as augmented AN.

Overall, we conclude that the augmented classifiers have significantly better classification performance than the regular classifiers when processing clean and altered images. Their accuracy remains high when dealing with images

containing visual camera failures, making them robust to altered images.

### B. ON BRIGHTNESS AND NOISE FAILURES

We explore the impact of brightness and noise camera failures that have a major effect on the classification performance of both regular and augmented classifiers.

**Brightness Analysis.** Fig. 5 plots accuracy achieved by regular (Fig. 5a) and augmented (Fig. 5b) classifiers when varying brightness levels in the range [0.1; 15], where 0.1 is an image almost entirely black, 1 is the clean image, and 15 is an image that is almost entirely white. Fig. 5a shows how brightness levels in the range [0.3; 1.5] do not have a major impact on the accuracy of regular classifiers, whereas brightness levels lower than 0.3 and greater than 1.5, dramatically reduce accuracy. As shown in Fig. 5b, augmented classifiers are more robust to brightness failures: their accuracy does not decrease significantly with low brightness, and it suffers serious degradation only with DITS and BelgiumTSC, and brightness levels greater than 1.5. Even



**TABLE 4.** Minimum accuracy of regular classifier MN2 on DITS, compared to classifiers trained on clean data plus one failure type. Improvement is shown as difference from the accuracy of the regular MN2. Augmented MN2 is reported for completeness.

Test Set	Regular Classifier	Clean + Banding	Clean + Black Pixel	Clean + Blurring	Clean + Brightness	Clean + Broken Lens	Clean + COND	Clean + Dirt	Clean + Ice	Clean + NOBF	Clean + NCAC	Clean + NDEMOS	Clean + NNR	Clean + Rain	Augmented Classifier
	Accuracy	Difference from accuracy of the regular classifier													
Clean	0.96	0.019	0.02	0.027	0.024	<u>0.032</u>	0.022	0.03	0.016	0.018	0.019	0.024	0.002	0.023	0.033
Banding	0.719	<u>0.254</u>	0.145		0.064	0.13			0.018			0.018	0.222	0.139	0.273
Black Pixels	0.849		<u>0.118</u>			0.067								0.009	0.135
Blurring	0.747			<u>0.211</u>		0.014									0.228
Brightness	0.294		0.046	0.046	<u>0.16</u>	0.049	0.067		0.04	0.06	0.037	0.01	0.027	0.055	0.164
Broken Lens	0.569					<u>0.356</u>			0.019						0.38
COND	0.883		0.047			<u>0.074</u>	0.072	0.039	0.066				0.038	0.06	0.105
Dirt	0.74	0.009	0.055	0.104	0.07	0.121	0.037	<u>0.22</u>	0.021			0.05		0.06	0.232
Ice	0.304		0.028	0.027	0.082			0.027	<u>0.541</u>				0.107	0.124	0.59
NOBF	0.79	0.091	0.091	0.017	0.047	0.079	0.089	0.063		<u>0.181</u>	0.093	0.072	0.086		0.192
NCAC	0.947		0.015	0.009	0.022	0.016	0.018	0.027	0.009	0.011	<u>0.028</u>	0.02	0.009	0.016	0.046
NDEMOS	0.728	0.062	0.148	0.111	0.065	0.059	0.1	0.167	0.053	0.126	0.084	<u>0.229</u>	0.12	0.106	0.255
NNR	0.318												<u>0.355</u>		0.332
Rain	0.763		0.108			0.078							0.076	<u>0.187</u>	0.225

in these cases, the degradation of accuracy is much less evident than that in regular classifiers.

**Noise Analysis.** Fig. 6 is analogous to Fig. 5 but refers to an NNR failure. The clean image has a noise level of 0; the higher the noise level, the more degraded the image. Fig. 6a shows that regular classifiers struggle to classify noisy images. Specifically, with InceptionV3 (IC3) on GTSRB, accuracy drops to random guessing when the noise level reaches or exceeds 0.6 (in a balanced 8-class classification problem, random guessing quantifies as 12.5% of accuracy). This is visible in the figure by looking at the solid brown line that immediately falls at the bottom of the plot.

Fig. 6b shows that augmented classifiers are, to some extent, robust to NNR; when the noise level is below 1, the accuracy still exceeds 0.90. In general, we observe only a very slight performance degradation: we hypothesize that this is due to a high number of images in the training data and the high resolution of the images.

**C. IMPACT OF INDIVIDUAL FAILURES ON TRAINING**

We investigated which visual camera failure contributed the most to the improved classification accuracy. For this purpose, we trained a DNN on 13 train variants, each composed of a *train\_clean* set plus images altered with a single visual camera failure. This creates 13 *intermediate* classifiers which employ a minimal data augmentation process.

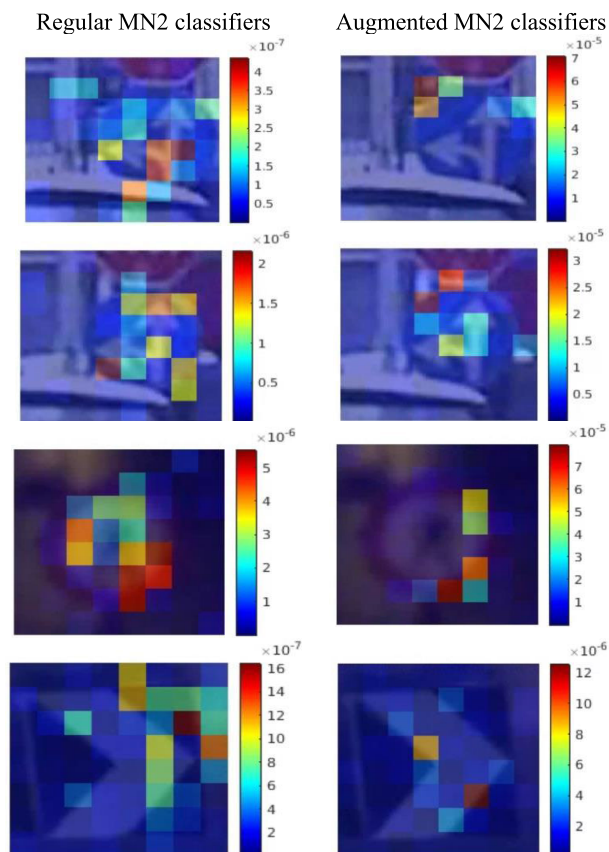
We discuss the behavior of MN2 on DITS. We deem it more interesting to explore the DITS dataset instead of GTSRB and BelgiumTSC, because classification on DITS has the lowest accuracy scores; thus, we believe it is the most challenging. MN2 is the DNN that always achieves the highest accuracy on DITS (see Table 2).

Table 4 contains a row for each test set. On the columns, we have 15 classifiers: the regular, the 13 obtained using the train variants above, and the augmented. The second

column of the table shows the minimum accuracy achieved by the regular MN2 classifier for any configuration of a specific failure to be used as a reference. The quantities shown in the rest of the table are the differences between the minimum accuracy achieved with the 13 intermediate classifiers and that achieved by the regular classifier. Only positive differences are reported; these are cases in which training with a specific train variant creates a better classifier than the regular one. The highest accuracy gains for each test set (each row in the table) are underlined; they show the train variant that improves classification the most with respect to training on *train\_clean*. Finally, the last column of the table reports the accuracy difference between *augmented* MN2 and *regular* MN2.

For all test sets, except *test\_clean* and *test\_COND*, the greatest improvement in accuracy was obtained by adding images altered with the corresponding failure in the training set. Adopting a training set composed of clean images and images with dirt failure increased the accuracy of MN2 from 0.74 to 0.96 (thus a difference of 0.22, see 8<sup>th</sup> row, 8<sup>th</sup> column in Table 4) when processing *test\_dirt*. The data augmentation process clearly provides the DNN with information about the effects of a specific visual camera failure, allowing MN2 to learn how to classify even in the presence of that specific failure. Interestingly, the greatest improvement in accuracy for *test\_clean* and *test\_COND* was when adding images altered with broken lens failure to the training set. This shows that a failure may be used to augment the training set and build a classifier that is robust to that specific failure and to others that may be correlated.

Another interesting observation is the accuracy of the clean images (first row of Table 4). The augmented classifier scored 0.993 accuracy with an increase of 0.033 with respect to the regular classifier, which was 0.960. Augmenting the training set with broken lens failure leads to a gain in accuracy of 0.032, whereas training with dirt failure provides an accuracy



**FIGURE 7.** Explanation of predictions of regular (on the left) and augmented (on the right) classifiers using LIME. (best viewed in colors).

improvement of 0.030 for *test\_clean*: this shows that specific failures make MN2 learn a model that also has enhanced classification performance with respect to clean images.

#### D. EXPLAINING PERFORMANCE OF REGULAR AND AUGMENTED CLASSIFIERS ON CLEAN IMAGES

While it is intuitive that the augmented classifiers have better classification accuracy than the regular classifiers on altered images, it is worth exploring why the classification accuracy on clean images improves as well. We rely on the LIME tool for eXplainable AI (XAI) [30] which allows us to examine how a classifier builds its prediction and explains the process behind the outputs of the image classifiers. LIME provides a graphical interface that shows the areas of the input image that have the highest relevance when calculating the output prediction.

Fig. 7 shows four images from DITS that were misclassified by the regular MN2 classifier (on the left) but were correctly classified by the augmented MN2 classifier (on the right). Each image in Fig. 7 was processed using LIME, which visualizes a heatmap of the most relevant features used for prediction. Red areas correspond to features that contribute the most to classification, whereas blue areas have negligible to no impact.

The regular MN2 classifier on the left side of Fig. 7 selects relevant features from many areas of the image, while the augmented MN2 classifier selects only a few stronger features. The color scales on the right of each image show the contribution of each feature in the order of  $10^{-6}$  to  $10^{-7}$  for regular classifiers, whereas the features for the augmented classifier have a larger absolute weight, with a magnitude of  $10^{-5}$ . These two observations pair well: the augmented classifier selects a few strong features, whereas the regular classifier selects many weak features from different parts of the image. This difference in the models learned from regular and augmented classifiers improves the image classification accuracy; relying on fewer features provides a clear advantage in our experiments.

#### VI. LIMITATIONS TO VALIDITY

Here, we report possible limitations to the validity and applicability of our study. These are not to be intended as showstoppers when considering the conclusions of this study. Instead, they should be interpreted as boundaries or possible future implications that may affect the validity of this study.

##### A. USAGE OF PUBLIC DATA AND LIBRARIES

The use of public image datasets and tools to inject visual camera failure algorithms enables the reproducibility of our analysis (see also scripts and corrupted data in [44] and [45]). However, the heterogeneity of data sources and potential lack of documentation may limit the understandability of the data. In addition, public datasets are not under our control; therefore, possible actions, such as changing the way the data are generated, are out of consideration. For example, it is not possible to create longer sequences of traffic signs for DITS nor creating a time-sequenced version of the BelgiumTSC.

##### B. PARAMETERS OF DEEP IMAGE CLASSIFIERS

Each deep classifier relies on parameters that are applicable to any deep neural network or specific to a DNN model. Finding the optimal values of the parameters is a substantial process that requires sensitive analyses and is directly linked to the scenario in which the classifier is to be exercised. We tried our best to precisely tune these parameters through grid searches, which ran a classifier with different parameter values and chose the parameter that maximized the accuracy. This does not guarantee to find the absolute optimum value of a parameter for a given classifier on a given dataset but constitutes a good approximation [46].

##### C. GENERALIZATION BEYOND TRAFFIC SIGN RECOGNITION

External validity is usually concerned with the extent to which the results of a study can be generalized. The idea of injecting the effects of camera failures into images to perform data augmentation goes beyond Traffic Sign Recognition, which is the domain from which our experimental data comes. Therefore, the methodology we used and assessed in this study can be used to build robust image classifiers for

other domains in which the data to be provided to the DNNs are captured by visual cameras, such as security cameras, camera trap images, and OCR for car plate recognition.

## VII. CONCLUSION AND FUTURE WORKS

This study investigated methods for improving the robustness of image classifiers against visual camera failures, either due to internal faults or adverse environmental conditions. The image classifiers built using our data augmentation technique tolerated most failures of the visual camera. Such data augmentation approach is not only improving robustness but is also improving classification accuracy. We showed that images altered using specific failures contribute more than others to improving the accuracy and robustness of classifiers. However, to ensure robustness with respect to the entire failure set, it was necessary to perform data augmentation by enriching the training set with altered images due to multiple visual camera failures.

Our future plan is to train a failure detector to exclude images that are corrupted beyond the extent to which our augmented classifier can properly classify them. In other words, we are aiming for an architectural approach in which the images are either discarded by the failure detector or deemed of adequate quality for processing using a robust classifier. Furthermore, we will explore whether our composite approach, which merges a failure detector and data augmentation, is sufficiently general to build safer image classifiers in safety-critical domains other than Traffic Sign Recognition.

## REFERENCES

- R. Gargeya and T. Leng, "Automated identification of diabetic retinopathy using deep learning," *Ophthalmology*, vol. 124, no. 7, pp. 962–969, 2017.
- D. A. Van Dyk and X.-L. Meng, "The art of data augmentation," *J. Comput. Graph. Statist.*, vol. 10, no. 1, pp. 1–50, 2001.
- M. Atif, T. Zoppi, M. Gharib, and A. Bondavalli, "Quantitative comparison of supervised algorithms and feature sets for traffic sign recognition," in *Proc. 36th Annu. ACM Symp. Appl. Comput.*, Mar. 2021, pp. 174–177.
- J. Li and Z. Wang, "Real-time traffic sign recognition based on efficient CNNs in the wild," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 3, pp. 975–984, Mar. 2019.
- M. Atif, T. Zoppi, M. Gharib, and A. Bondavalli, "Towards enhancing traffic sign recognition through sliding windows," *Sensors*, vol. 22, no. 7, p. 2683, Mar. 2022.
- J. H. Chung, D. W. Kim, T. K. Kang, and M. T. Lim, "Traffic sign recognition in harsh environment using attention based convolutional pooling neural network," *Neural Process. Lett.*, vol. 51, no. 3, pp. 2551–2573, Jun. 2020.
- A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2012.
- C. Szegegy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.
- M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
- J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. Int. Joint Conf. Neural Netw.*, 2011, pp. 1453–1460.
- R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 633–647, Apr. 2014.
- A. Youssef, D. Albani, D. Nardi, and D. D. Bloisi, "Fast traffic sign recognition using color segmentation and deep convolutional networks," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, Cham, Switzerland: Springer, Oct. 2016, pp. 205–216.
- U. S. Shanthamallu, A. Spanias, C. Tepedelenlioglu, and M. Stanley, "A brief survey of machine learning methods and their sensor and IoT applications," in *Proc. 8th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Aug. 2017, pp. 1–8.
- S. Ghosh, R. Shet, P. Amon, A. Hutter, and A. Kaup, "Robustness of deep convolutional neural networks for image degradations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 2916–2920.
- G. Gong, T. Ren, M. Ye, and Q. Liu, "MaxUp: Lightweight adversarial training with data augmentation improves neural network training," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2474–2483.
- A. Agarwal, M. Vatsa, R. Singh, and N. Ratha, "Cognitive data augmentation for adversarial defense via pixel masking," *Pattern Recognit. Lett.*, vol. 146, pp. 244–251, Jun. 2021.
- Z. Fabian, R. Heckel, and M. Soltanolkotabi, "Data augmentation for deep learning based accelerated MRI reconstruction with limited data," in *Proc. Int. Conf. Mach. Learn.*, Jul. 2021, pp. 3057–3067.
- H. Naveed, "Survey: Image mixing and deleting for data augmentation," 2021, *arXiv:2106.07085*.
- ActionVFX*. Accessed: Dec. 22, 2022. [Online]. Available: <https://www.actionvfx.com/>
- S. Postalçglu, "Performance analysis of different optimizers for deep learning-based image recognition," *Int. J. Pattern Recognit. Artif. Intell.*, vol. 34, no. 2, Feb. 2020, Art. no. 2051003.
- N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
- A. Ceccarelli and F. Secci, "RGB cameras failures and their effects in autonomous driving applications," *IEEE Trans. Dependable Secure Comput.*, early access, Mar. 7, 2022, doi: 10.1109/TDSC.2022.3156941.
- J. B. Phillips and H. Eliasson, *Camera Image Quality Benchmarking*. Hoboken, NJ, USA: Wiley, 2018.
- Y. Liu, E. Racah, J. Correa, A. Khosrowshahi, D. Lavers, K. Kunkel, M. Wehner, and W. Collins, "Application of deep convolutional neural networks for detecting extreme weather in climate datasets," 2016, *arXiv:1605.01156*.
- S. Min, B. Lee, and S. Yoon, "Deep learning in bioinformatics," *Briefings Bioinform.*, vol. 18, no. 5, pp. 851–869, 2017.
- Realistic Lens Blur/Chromatic Aberration Filter*. Accessed: Sep. 15, 2022. [Online]. Available: <https://github.com/yoonsikp/kromo>
- J. G. Carbonell, R. S. Michalski, and T. M. Mitchell, "An overview of machine learning," in *Machine Learning, An Artificial Intelligence Approach*, vol. 1, 1983, pp. 3–23.
- C. Premevida, G. Melotti, and A. Asvadi, "RGB-D object classification for autonomous driving perception," in *RGB-D Image Analysis and Processing*. Cham, Switzerland: Springer, 2019, pp. 377–395.
- K. Hogan, "Options for camera raw in the digital workflow," in *Proc. SMPTE Annu. Tech. Conf. Exhib.*, Oct. 2014, pp. 1–18.
- M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?': Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 1135–1144.
- O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, and A. C. Berg, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
- N. Humbatova, G. Jahangirova, G. Bavota, V. Riccio, A. Stocco, and P. Tonella, "Taxonomy of real faults in deep learning systems," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, Jun. 2020, pp. 1110–1121.
- A. Stocco, M. Weiss, M. Calzana, and P. Tonella, "Misbehaviour prediction for autonomous driving systems," in *Proc. ACM/IEEE 42nd Int. Conf. Softw. Eng.*, Jun. 2020, pp. 359–371.
- J. Henriksson, C. Berger, M. Borg, L. Tornberg, S. R. Sathyamoorthy, and C. Englund, "Performance analysis of out-of-distribution detection on various trained neural networks," in *Proc. 45th Euromicro Conf. Softw. Eng. Adv. Appl. (SEAA)*, Aug. 2019, pp. 113–120.
- D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019, *arXiv:1903.12261*.
- Y.-C. Hsu, Y. Shen, H. Jin, and Z. Kira, "Generalized ODIN: Detecting out-of-distribution image without learning from out-of-distribution data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10951–10960.

- [37] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing image artifacts due to dirty camera lenses and thin occluders," in *Proc. ACM SIGGRAPH Asia Papers*, Dec. 2009, pp. 1–10.
- [38] S. Hossain, A. R. Fayjie, O. Doukhi, and D. J. Lee, "CAIAS simulator: Self-driving vehicle simulator for AI research," in *Proc. Int. Conf. Intell. Comput. Optim.* Cham, Switzerland: Springer, Oct. 2018, pp. 187–195.
- [39] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
- [40] N. Drenkow, N. Sani, I. Shpitser, and M. Unberath, "A systematic review of robustness in deep learning for computer vision: Mind the gap?" 2021, *arXiv:2112.00639*.
- [41] G. Bradski and A. Kaehler, "Dr Dobb's journal of software tools," *OpenCV Library*, vol. 25, no. 11, p. 120, 2000.
- [42] *PIL 3.0*. Accessed: Dec. 22, 2022. [Online]. Available: <https://pillow.readthedocs.io/en/3.0.x/>
- [43] *CV2 Filtering*. Accessed: Dec. 22, 2022. [Online]. Available: <https://docs.opencv.org/2.4/modules/imgproc/doc/filtering.html>
- [44] *Developed Software (GitHub)*. Accessed: Dec. 22, 2022. [Online]. Available: <https://github.com/muhammadatif11081992/Robust-TSR>
- [45] *Dataset of Altered TSR Images (Google Drive-Approximately 100GB Compressed)*. Accessed: Dec. 22, 2022. [Online]. Available: <https://drive.google.com/drive/folders/12HA1xV7Zb9XHxI-BwzdcZB3yzCM2tho>
- [46] A. B. Jiménez, J. L. Lázaro, and J. R. Dorronsoro, "Finding optimal model parameters by discrete grid search," in *Innovations in Hybrid Intelligent Systems*. Berlin, Germany: Springer, 2007.



**ANDREA CECCARELLI** is currently an Associate Professor in computer science at the University of Florence, Florence, Italy. His primary research interests include the design, monitoring, experimental evaluation of dependable and secure systems, and systems-of-systems. His scientific activities originated more than 100 papers that appeared in international conferences, workshops, and journals. He is also a member of the IFIP WG 10.4 on "Dependable Computing and Fault-Tolerance." He has been the PC co-chair of the conferences SRDS and LADC.



**TOMMASO ZOPPI** is currently a Research Associate at the University of Florence, Brazil. He is involved in several European and nationally funded and even industrial projects. His research interests include anomaly detection, security and safety, often applies standards to plan, design, develop, and implement appropriate architectures or software in the domain of critical systems. He serves as a member of the program committee at several international conferences.



**ANDREA BONDAVALLI** (Senior Member, IEEE) is currently a Full Professor in computer science at the University of Florence. His research interests include the design and evaluation of resilient and secure systems and infrastructures. His scientific activities originated more than 220 papers that appeared in international journals and conferences. He led various national and European projects and has been chairing the program committee in several international conferences. He is a member of the IFIP W.G. 10.4 Working Group on "Dependable Computing and Fault-Tolerance."

...



**MUHAMMAD ATIF** received the B.E. degree in IT from the Dr. A. Q. Khan Institute of Computer Science and Information Technology, Pakistan, in 2014, and the M.S. degree in computer system engineering from GIKI, Pakistan, in 2016. He is currently pursuing the Ph.D. degree with the University of Florence, Italy. From 2016 to 2019, he worked as a Lecturer with the Department of Computer Science, FAST NUCES, Pakistan. His research interests include data mining, computer vision, and computational intelligence.

Open Access funding provided by 'Università degli Studi di Firenze' within the CRUI CARE Agreement