Dealing with Robustness of Convolutional Neural Networks for Image Classification

Paolo Arcaini
National Institute of Informatics
Tokyo, Japan
arcaini@nii.ac.jp

Andrea Bombarda
University of Bergamo
Bergamo, Italy
andrea.bombarda@unibg.it

Silvia Bonfanti
University of Bergamo
Bergamo, Italy
silvia.bonfanti@unibg.it

Angelo Gargantini
University of Bergamo
Bergamo, Italy
angelo.gargantini@unibg.it

***Abstract*—SW-based systems depend more and more on AI also for critical tasks. For instance, the use of machine learning, especially for image recognition, is increasing ever more. As state-of-the-art, Convolutional Neural Networks (CNNs) are the most adopted techniques for image classification. Although they are proved to have optimal results, it is not clear what happens when unforeseen modifications during the image acquisition and elaboration occur. Thus, it is very important to assess the robustness of a CNN, especially when it is used in a safety critical system, as, e.g., in the medical domain or in automated driving systems. Most of the analyses made about the robustness of CNNs are focused on adversarial examples which are created by exploiting the CNN internal structure; however, these are not the only problems we can encounter with CNNs and, moreover, they may be unlikely in some fields. This is why, in this paper, we focus on the robustness analysis when plausible alterations caused by an error during the acquisition of the input images occur. We give a novel definition of robustness w.r.t. possible input alterations for a CNN and we propose a framework to compute it. Moreover, we analyse four methods (data augmentation, limited data augmentation, network parallelization, and limited network parallelization) which can be used to improve the robustness of a CNN for image classification. Analyses are conducted over a dataset of histologic images.**

***Index Terms*—Convolutional Neural Networks, robustness, alteration, image classification**

I. Introduction

Machine Learning (ML) and especially Convolutional Neural Networks (CNNs) are used in image classification, also for performing critical tasks [13], [23]. Although extensive learning can be applied, it is not clear what happens when unforeseen modifications during the image acquisition process occur. This is due to the fact that machine learning is almost a "black box" technique [6]. For example, it is possible that, for some errors in the acquisition and elaboration process, input images are blurred, contain some noise, are underexposed or overexposed. Moreover, the image classification can be used over a JPEG compressed image, which has a lower quality w.r.t. the images used in the training phase. For crucial tasks, like diagnosis in the medical sector, or for automated driving systems, the accuracy of CNNs over altered images can be a critical factor.

P. Arcaini is supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST. Funding Reference number: 10.13039/501100009024 ERATO. We would like to thank Michele Zanchi for the preliminary work done for his master thesis.

Many studies have been conducted on the *robustness* of a neural network versus input alterations. Most of them put their focus on the generation of *adversarial examples*, by pursuing the most common approach in software testing that aims at finding particular inputs that show the failure of the system. For instance, [29] defines robustness as the ability to classify in the same category similar inputs, even if they are adversarial examples, and it suggests several solutions that can improve the robustness of a network versus adversarial examples.

However, since adversarial examples are created by exploiting the CNN internal structure [25], they may be unlikely to occur during the CNN operation (as also noted in [20], [21], [33]); on the other hand, they may not include *plausible* alterations that could really happen during the CNN usage. Our position is that a robustness measure should take into account all the plausible domain specific alterations.

For this reason, in this paper, we propose a testing approach (complementary to that pursued by adversarial examples) that aims at gaining confidence on the *reliability* of the image recognition system in its *typical working condition* [12]. In particular, (i) we define a way to formalize plausible *alterations*, (ii) we propose a measure of *robustness* w.r.t. these alterations, and (iii) we study how robustness can be improved, using 4 techniques for CNN retraining. Since most researches in CNN testing focus on adversarial examples, we also study the relation between our alterations and adversarial examples; to this aim, we introduce a measure of *adversariability* that can be compared with our definition of robustness.

As case study, we consider image classification in the medical domain. Namely, we define plausible alterations that can occur during image acquisition in medical practice, we measure robustness for a CNN trained to recognize cancers, and we assess how much robustness improves with the four CNN retraining techniques. We have found that CNN robustness can be not optimal, but it can be improved without losing accuracy over the unaltered images. We have discovered that the best way to improve robustness is to train the network by applying a variant of *Data Augmentation* (DA), called *Limited Data Augmentation* (LDA), that adds only the alterations which are proven to lead to an accuracy lower than 100% with the original network. A less effective but lighter alternative is the *Network Parallelization* technique, which trains a parallel network using only the alterations which are proven to lead

to an accuracy less than 100%, as in LDA.

Paper structure. Sec. II provides the necessary background. Then, Sec. III gives our novel definitions of *alteration* and *robustness*, and also introduces the notion of *adversariability* to assess how robustness relates to approaches based on adversarial examples. Sec. IV presents the running case study, the dataset used, and the trained CNN. Sec. V shows how to measure robustness, and Sec. VI analyses four ways that can be used to improve the robustness of a CNN, and shows their application to the case study. Sec. VII reviews some works related to the analysis of the robustness of the CNNs used to classify images, and Sec. VIII concludes the paper.

II. Background

A *convolutional neural network* (CNN) is a type of deep neural network mainly used to analyse images. CNNs use the linear mathematical operation *convolution* (instead of the regular matrix multiplication) in at least one of their layers [23].

CNNs can be trained to be used as binary classifier to assess, e.g., whether an image contains a given element of interest.

Definition 1 (Binary classifier)**.** *A* binary classifier C *for images can be seen as a Boolean function that tells whether an image p has the characteristic of interest, i.e.,*

$$C(p) \iff C \text{ classifies } p \text{ as belonging to a set of interest}$$

In order to train and test the CNN C to be used as classifier, different sets of labelled images are used: IS is the input set of images divided in training set TR, and validation set VA (i.e., IS =TR $\cup$VA); TE is the test set.

Since we use the CNN as a binary classifier, we measure its *quality* in terms of *accuracy*, as done in [34] and [28]. However, any other quality measure could be used.

Definition 2 (Accuracy)**.** *The* accuracy *of a binary classifier C w.r.t. a set of images P is defined as the ratio of correctly evaluated images in P, i.e.,*

$$acc(C,P) = \frac{|\{p \in P \mid C(p) = label(p)\}|}{|P|}$$

where label gives the correct evaluation of an image p.

Although the accuracy can be computed w.r.t. any set of images, in the following, we will do it w.r.t. the test set TE.

III. Proposed Robustness Measure

We expect that, given a classifier C, by altering an image p, the trustworthiness of the response of C will change and it will likely decrease by increasing the alteration level. Therefore, the accuracy of the classifier also depends on the quality of the images used in testing. Fig. 1 shows how the classifier accuracy diminishes when changing the brightness of the images in TE.

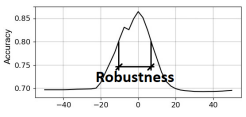


Fig. 1. Accuracy change when brightness is altered

How can we define and measure the *robustness* of a classifier when an *alteration* occurs? First, we define what we mean with *alteration* and then we provide a definition of *robustness*.

Definition 3 (Alteration)**.** *An* alteration *of type A of a digital image is a transformation of the image that mimics the possible effect over the image when a problem in image acquisition, or in its elaboration, occurs in reality. In the following, we identify with $[L_A, U_A]$ the range of plausible alterations of type A; moreover, we identify with P^{A_i} the set of images obtained by altering all the images in P with an alteration of type A of level $i \in [L_A, U_A]$. We require one element $I_A \in [L_A, U_A]$ to be the* unaltered value*, i.e., $P^{A_{I_A}} = P$.*

Typical alterations are translation, blur, noise, compression, and zoom. For each alteration A, the user defines a suitable interval $[L_A, U_A]$ by analysing the risks that may occur during image processing. For example, the camera used for acquisition could have a damaged sensor, causing an alteration of the image brightness: we can set the alteration value to U_A=50% for an overexposed image, and L_A=-50% for an underexposed image. Note that these artificial alterations must mimic real one, e.g., in translation, no black border should be inserted[1].

Given a set of alterations, we define robustness as follows.

Definition 4 (Robustness)**.** *Let Θ be a threshold representing the minimum accepted accuracy. The* robustness *of a classifier C w.r.t. transformation of type A in the range $[L_A, U_A]$ (using a set of images P) is defined as the percentage of alteration values for which the accuracy is above Θ. Formally:*

$$rob_A(C,P) = \frac{\int_{L_A}^{U_A} H(acc(C,P^{A_i}) - \Theta)di}{U_A - L_A} \quad \text{where} \quad H(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

As said before, we have decided to compute the accuracy and the robustness by using the test set, so P = TE.

Fig. 1 shows robustness representation with Θ = 80%. Since computing robustness is not feasible (as it requires to use an infinite number of alterations), we approximate it as follows.

Definition 5 (Approximate robustness)**.** *Given n equi-distributed points $SP = \{i_1, \ldots, i_n\}$ sampled in the interval $[L_A, U_A]$ of all the possible alterations of type A, the* approximate robustness *is defined as:*

$$rob_A(C,P) = \frac{|\{i \in SP \mid acc(C,P^{A_i}) \geq \Theta\}|}{|SP|} \qquad (1)$$

The previous definitions use the *accuracy* to compute the CNN robustness. However, the definitions could be adapted to use recall, precision, or F1-score, depending on the context.

Adversariability: Although the alterations considered by our robustness function are those that are more likely to occur in reality, they may not consider some rare cases that could be particular difficult for the network. A well-established line of research aims at finding these *adversarial examples* [25] that can mislead the classifier. Differently from our alterations, they are built starting from the network internal structure; namely, given an image p correctly classified, an adversarial example is a modification p' of p, computed considering the network

[1]Note that black borders removal is automatically done by the CNN preprocessing.

structure, so that p' is wrongly classified. Different approaches have been proposed for generating adversarial examples, each one considering different aspects of the network; please refer to [35] for a survey. Even if adversarial examples could be less likely to occur in many fields [20], they can still occur; therefore, we also consider them in our analyses. To this aim, in the following we propose a notion of *adversariability* which will be used to evaluate to what extent adversarial examples and our alterations are related, i.e., whether there is a relation between adversariability and robustness.

Differently from alterations, the generation of adversarial examples does not directly provide us a measure of difference between the starting image and the modified one. Therefore, we use the classical definition of *structural similarity index* taken from [5], defined as follows.

Definition 6 (Structural similarity index)**.** *The* structural similarity index *between the two images p and q is defined as:*

$$S(p,q) = \frac{(2\mu_p\mu_q + c_1)(2\sigma_{pq} + c_2)}{(\mu_p^2 + \mu_q^2 + c_1)(\sigma_p^2 + \sigma_q^2 + c_2)}$$

where μ_p and μ_q are the averages of the pixel values in p and q, σ_p^2 and σ_q^2 are the variances of p and q, σ_{pq} is the covariance of p and q, and c_1 and c_2 are two constants.

The value $S(p,q)$ is in interval $[0,1]$, where 1 means that the images are identical.

Let's $ADVEX(C,p)$ be the set of adversarial examples (generated by a given technique) for a classifier C and an image p. $ADVEX(C,p)$ is empty if p cannot be modified in a way to mislead C or the generation technique of $ADVEX$ is not powerful enough. Among all adversarial examples (which can be many), we select the *most adversarial* one:

Definition 7 (Most adversarial example)**.** *Let C be a binary classifier, and p an image correctly classified, i.e., $C(p) = label(p)$. We define the* most adversarial example *as the most similar image to p that is misclassified (if it exists), formally:*

$$p^{ae} = \underset{p' \in ADVEX(C,p)}{\arg\max} S(p,p')$$

If $ADVEX(C,p)$ is empty, we say that p^{ae} does not exist.

By using Def. 6 and 7, we define the vulnerability w.r.t. adversarial examples as follows.

Definition 8 (Adversariability)**.** *Let C be a binary classifier, and P a set of images. We define the* adversariability *of C, as the percentage of correctly evaluated images p of P for which there exists an adversarial example p^{ae},* weighted *by the similarity index between p and p^{ae}. Formally:*

$$adv(C,P) = \frac{\sum_{p \in CE} \hat{S}(p, p^{ae})}{|CE|}$$

where $\hat{S}$ is equal to the similarity S if the adversarial example p^{ae} exists, 0 otherwise; and $CE = \{p \in P \mid C(p) = label(p)\}$ is the subset of P of images correctly evaluated.

$adv(C,P)$ is in the range $[0,1)$, where higher values mean that C is more vulnerable to adversarial examples. Note that adversarial examples p^{ae} that are more similar to the original image p (i.e., those having higher similarity index $\hat{S}(p,p^{ae})$) are those that contribute the most to the adversariability: indeed, they represent the most insidious cases in which an imperceptible modification misleads the classification.

Similarly to robustness, in the following, the adversariability will be always computed w.r.t. the test set TE.

IV. Case study

Breast cancer (particularly, Invasive Ductal Carcinoma (IDC)) is one of the main causes of cancer death in women ($\sim$12% in 2019) and one of the most diagnosed cancers (1/3 of all cancers) [4]. All diagnoses are based on analysing images of histological features of tissue or cells removed with surgery or biopsy. These images are captured by a microscope and examined by pathologists to make a decision about the benignity or the malignity of the suspected cancer.

In this paper, we used the publicly available dataset curated by [16]: it consists of 162 images of slides scanned at 40×, from which a total of 277,524 labelled patches of 50×50 pixels were extracted (198,738 benign examples, and 78,786 malign examples). Examples of these images are reported online [1].

For the analysis proposed in this paper, we train and test a CNN C_o supposed to identify whether the input image comes from a patient with IDC or not. For the implementation, we use Python and its library Keras. For the network structure, we have been inspired by [24] that describes a CNN for breast cancer identification. The first layer in C_o is a convolutional layer, with 32 filters, 3×3 kernels, followed by a rectified linear unit (ReLU) activation function. Then, we insert a batch-normalization layer, a max-pooling layer and a drop-out of 0.3 to prevent over-fitting. After these layers, we put a double couple of convolutional layers (64 filters, with 3×3 kernels) and ReLU activation functions. To further prevent over-fitting, we insert another batch normalization layer followed by a max-pooling layer. The last block of layers is composed of a fully-connected-layer with ReLU activation, a batch normalization with a drop-out of 0.5, and a sigmoid classifier.

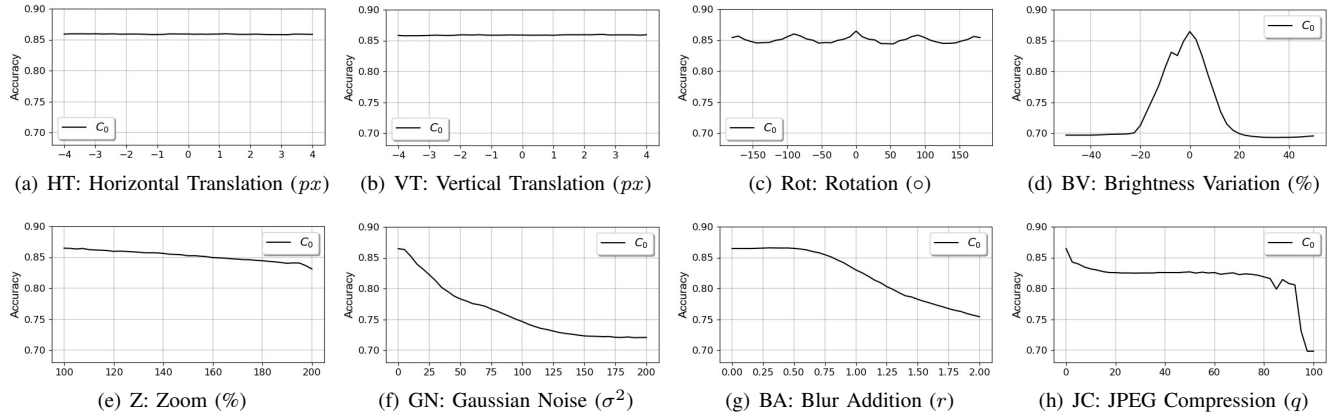
C_o has 63,106 parameters and its training took 2h08m. We used an input set IS_{C_o} of 210,593 images taken from the original dataset [16]: 70% (i.e., 147,415 images) have been used as training set TR_{C_o}, and the remaining 30% (i.e., 63,178 images) as validation set VA_{C_o}; C_o achieved an accuracy of 86,46% on the test set TE_{C_o} composed of the remaining 66,931 images.

The Python scripts used to achieve the results presented in this paper, the plots of the accuracy, and all the instructions to replicate the results can be retrieved online [1].

V. Measuring robustness

In order to investigate the robustness of the CNN C_o under study, we consider the alterations displayed in Table I, i.e., the most common alterations that can occur when working on digital images in the medical sector using a microscope:

- **Horizontal** (HT) and **Vertical Translation** (VT) and **Rotation** (Rot) may occur when the microscopic slides are incorrectly placed.



(a) HT: Horizontal Translation (px) (b) VT: Vertical Translation (px) (c) Rot: Rotation ($\circ$) (d) BV: Brightness Variation (%)

(e) Z: Zoom (%) (f) GN: Gaussian Noise (σ^2) (g) BA: Blur Addition (r) (h) JC: JPEG Compression (q)

Fig. 2. Accuracy modification using the altered data input over the original CNN C_o

TABLE I
ALTERATIONS CONSIDERED IN THE ROBUSTNESS ANALYSIS

A	HT	VT	Rot	BV	Z	GN	BA	JC
$[L_A, U_A]$	$[-4px, 4px]$	$[-4px, 4px]$	$[-180°, 180°]$	$[-50\%, 50\%]$	$[100\%, 200\%]$	$[0, 200]$	$[0, 2]$	$[0, 100]$
I_A	$0px$	$0px$	$0°$	0%	100%	0	0	0

Brightness Variation (BV) may occur because each microscope has a different brightness for the produced images.

Zoom (Z) is chosen by the user working with the microscope.

Gaussian Noise (GN) simulates the possible effect of a wrong manipulation of the microscopic slide (e.g., too much dye has been used for contrast) [8]. We considered different values for the variance σ^2 of the noise.

Blur Addition (BA) may occur due to a small move of the tool causing a focus loss. We vary the radius r of blurring.

JPEG Compression (JC) may occur when images are transferred in a lossy manner. We vary compression value q.

Following Def. 5, we compute the (approximate) robustness of the CNN C_o, by using its train set TE_{C_o}. We use n=40 total points SP, and a minimum accepted accuracy Θ =80.00%. By applying the alterations in Table I to TE_{C_o} and classifying all the altered images with C_o, we obtained the results in Table II reporting the values of $rob_A(C_o, TE_{C_o})$ for the different alteration types A. They confirm the invariance property of the network with respect to geometric transformations [19] (i.e., the classification does not change when some particular geometric transformations are applied): indeed, we have a robustness of 100% in translation, rotation, and zoom with the given minimum accepted accuracy Θ. The changes of $acc(C_o, TE^{A_i}_{C_o})$ obtained by applying the alterations of type A can be observed in the plots in Fig. 2. Apart from the alterations achieving 100% robustness, we observe that some of the other alterations (e.g., JC) keep the accuracy value greater than Θ for most of their alteration interval, so leading to higher robustness values. For other alterations (e.g., GN), instead, the accuracy is lower than Θ for most of their alteration level, and so the robustness is lower.

We have also computed, by using test set TE_{C_o}, the *adversariability* of the classifier C_o (see Def. 8). We have obtained a set of adversarial examples (with L-BFGS Attack [35]) that lead to an adversariability value $adv(C_o, TE_{C_o})$=0.38.

VI. How to improve robustness

We have previously shown that altering the images to be classified leads to a decrement of the classifier accuracy. In safety critical systems (as the medical application of our case study), we want to have a system as robust as possible.

The first applicable solution is to create a more complex CNN, which is able to guarantee a better robustness; however, this solution could be too costly and, moreover, the designer could also not know how to modify the network to increase robustness. Therefore, in this paper, we consider additions to the training data or automatic extensions of the network that do not require the intervention of the designer.

In the following, we analyse the robustness improvement obtained with four techniques for CNN retraining. A good technique should not only improve the robustness, but also not degrade the classification of the unaltered images; therefore, we will also compute the accuracy of the retrained network.

We first consider the well-known solution of *data augmentation* (and one of its variant we have devised), and then a technique based on the training of a *parallel network* (and one of its variant). The techniques are compared in Table III.

A. Data Augmentation (DA)

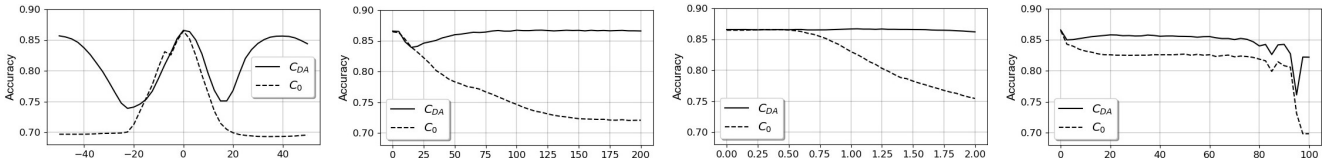
Data augmentation (DA) is a wide subject [30]. In our context, DA consists in training the CNN by using both the original and the altered images. The major con of this solution is that we have to retrain the whole network and this can require a lot of time because it is not possible to retrain the original model using only the new augmented dataset, as this would cause a *catastrophic forgetting*, i.e., the loss of the knowledge learnt during the original training phase [17].

We tried DA by applying some alteration values[2] of each alteration type A to half of the images randomly selected,

[2]To reduce the training time, we applied a limited number of alteration values to each image. We took 4 samples for each image when the alteration can be positive and negative, and only 2 samples when it can only be positive.

TABLE II
ROBUSTNESS

Alt.	C_o rob_A	C_{DA} ×n	rob_A	Δ_{C_o}	C_{LDA} ×n	rob_A	Δ_{C_o}	$\Delta_{C_{DA}}$	C_{NP} ×n	rob_A	Δ_{C_o}	$\Delta_{C_{DA}}$	C_{LNP} ×n	rob_A	Δ_{C_o}	$\Delta_{C_{NP}}$
HT	100.00%	4	100.00%	0.00%	0	100.00%	0.00%	0.00%	4	100.00%	0.00%	0.00%	0	100.00%	0.00%	0.00%
VT	100.00%	4	100.00%	0.00%	0	100.00%	0.00%	0.00%	4	100.00%	0.00%	0.00%	0	100.00%	0.00%	0.00%
Rot	100.00%	2	100.00%	0.00%	0	92.96%	−7.04%	−7.04%	2	100.00%	0.00%	0.00%	0	100.00%	0.00%	0.00%
BV	17.07%	4	60.97%	43.90%	4	87.80%	70.73%	26.83%	4	17.07%	0.00%	−43.90%	4	24.39%	7.32%	7.32%
Z	100.00%	2	100.00%	0.00%	0	100.00%	0.00%	0.00%	2	100.00%	0.00%	0.00%	0	100.00%	0.00%	0.00%
GN	19.51%	2	100.00%	80.49%	2	100.00%	80.49%	0.00%	2	34.14%	14.63%	−65.86%	2	34.14%	14.63%	0.00%
BA	63.41%	2	100.00%	36.59%	2	100.00%	36.59%	0.00%	2	100.00%	36.59%	0.00%	2	100.00%	36.59%	0.00%
JC	87.80%	2	97.56%	9.76%	2	97.56%	9.76%	0.00%	2	90.24%	2.44%	−7.32%	2	90.24%	2.44%	0.00%
AVG	73.47%		94.81%			97.29%				80.18%				81.09%		



(a) BV: Brightness Variation (%) (b) GN: Gaussian Noise (σ^2) (c) BA: Blur Addition (r) (d) JC: JPEG Compression (q)

Fig. 3. Accuracy modification using the CNN C_{DA} (original CNN with the data-augmentation technique)

obtaining a total of 2,526,281 input images, i.e., we used the set $IS_{C_{DA}}$=IS_{C_o}∪S_{DA}, where S_{DA} is the set of selected altered images and IS_{C_o} the input set used for the original classifier C_o. We obtained the new classifier C_{DA} with a training duration of 14h54m and a reached accuracy of 86.57% on the test set TE_{C_o} (the same used for classifier C_o). Its robustness values $rob_A(C_{DA}, TE_{C_o})$ are shown in Table II. The value $\times n$ in Table II indicates the number of altered images of alteration type A generated from each single standard image and Δ_{C_o} represents the robustness improvement w.r.t. the robustness of the original CNN C_o. The changes of $acc(C_{DA}, TE_{C_o}^{A_i})$ obtained by applying the alterations of type A (only for the alterations that lead to a robustness less than 100% with the original CNN C_o) can be observed in Fig. 3 (other plots are reported online [1]). We observe that the performances of C_{DA} (continuous lines) are much better than those of the original network C_o trained only with unaltered data (dashed lines). For BV, we observe that the accuracy is improved more for larger alteration values than for small ones: this may due to the fact that the retraining does not get enough knowledge from small alterations, while it can learn larger alteration values because they are more distinguishable. Note that a similar effect (although less evident) can also be observed for GN and JC.

To conclude, we observe that although the overall accuracy (i.e., of unaltered images) improved only a little (86.57%), the robustness is much better.

Moreover, we have computed the *adversariability* of C_{DA} (over the test set TE_{C_o}), obtaining a value $adv(C_{DA}, TE_{C_o})$=0.64; the value is increased w.r.t. the value of C_o.

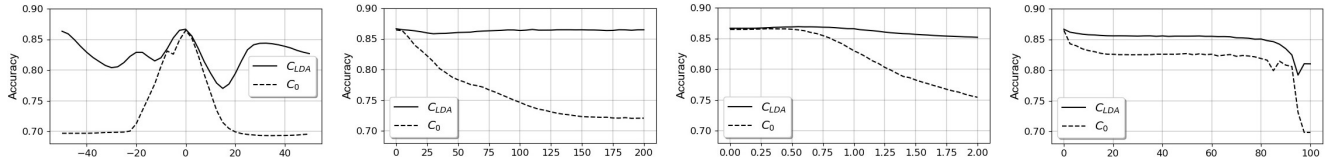
B. Limited Data Augmentation (LDA)

The main con of DA is that the training process is very time expensive. So, we here propose an alternative version to mitigate this problem. Exploiting the CNNs invariance properties, we propose to use augmented images only for the alterations that lead to a robustness less than 100% on C_o.

We trained the new CNN C_{LDA} with images obtained by applying the selected alterations (BV, GN, BA, and JC) that obtain less than 100% robustness on C_o (see Table II), obtaining in total additional 2,316,523 images, i.e., we used the input set $IS_{C_{LDA}}$=IS_{C_o}∪S_{LDA}, where S_{LDA} is the set of selected altered images and IS_{C_o} the input set used for C_o. Using this dataset, we obtained the new classifier C_{LDA} with a training duration of 13h39m and a reached accuracy of 86.64% on the test set TE_{C_o} (the same used for the other classifiers). Its robustness values $rob_A(C_{LDA}, TE_{C_o})$ are shown in Table II. The value $\times n$ in the table indicates the number of altered images generated from each single standard image and Δ_{C_o} represents the robustness improvement w.r.t. C_o, while $\Delta_{C_{DA}}$ is the improvement w.r.t. C_{DA}. The changes of $acc(C_{LDA}, TE_{C_o}^{A_i})$ obtained by applying the alterations of type A on the images in the test set TE_{C_o} can be observed in the plots in Fig. 4. Other plots are reported online [1].

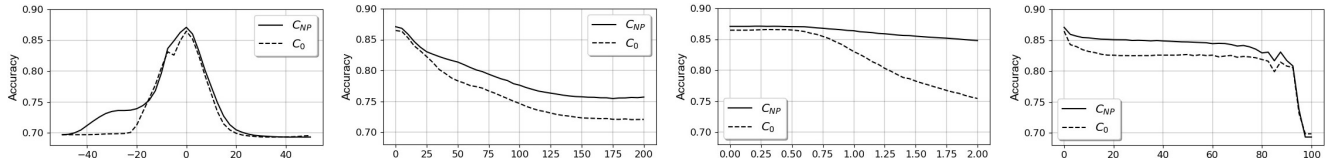
We observe that the average robustness of C_{LDA} is better than C_{DA} and C_o; this may be due to the fact that the retraining focuses on the weaknesses of the network. Considering the robustness of each single alteration, the only decrease happens with Rot, for which we did not augment input images; all the other alterations are classified better or equally than the previous solutions. So, LDA is a good choice, it is faster than the full data-augmentation training and provides better results.

Moreover, we have computed, by using the test set TE_{C_o}, the *adversariability* of the classifier C_{LDA}, obtaining a value



(a) BV: Brightness Variation (%) (b) GN: Gaussian Noise (σ^2) (c) BA: Blur Addition (r) (d) JC: JPEG Compression (q)

Fig. 4. Accuracy modification using CNN C_{LDA} (original CNN with limited-data-augmentation technique)



(a) BV: Brightness Variation (%) (b) GN: Gaussian Noise (σ^2) (c) BA: Blur Addition (r) (d) JC: JPEG Compression (q)

Fig. 5. Accuracy modification using C_{NP} (original CNN with parallel network)

$adv(C_{LDA}, TE_{C_o})$= 0.39. We observe that its value is decreased w.r.t. the value of C_{DA}.

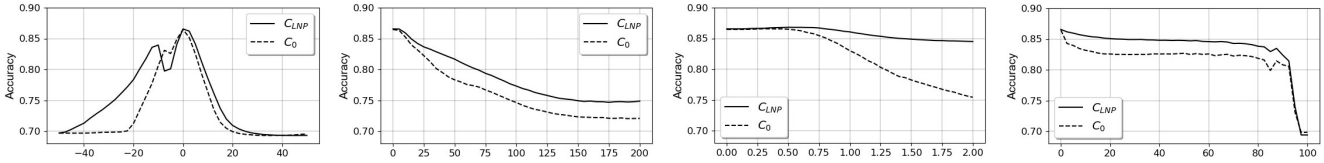
C. Network Parallelization (NP)

To overcome the computational cost of DA, we propose a different technique that adds to C_o a parallel network C_{Par}, trained with altered images only. The outcome of the whole network C_{NP}=$C_o \| C_{Par}$ is given by the combination (using a max-layer) of the outputs of C_o and C_{Par}.

We trained C_{Par} with only the 2,315,688 images obtained by applying the alterations to the original ones in the input set IS_{C_o} (i.e., only using the set S_{DA} that we also used in DA). We obtained C_{Par} with a training duration of 13h26m; the reached accuracy of the total CNN C_{NP} is 87.10% over the test set TE_{C_o}. Its robustness values $rob_A(C_{NP}, TE_{C_o})$ are shown in Table II. The value $\times n$ in Table II indicates the number of altered images generated from each single standard image, Δ_{C_o} represents the robustness improvement w.r.t. C_o, while $\Delta_{C_{DA}}$ w.r.t. C_{DA}. The changes of $acc(C_{NP}, TE^{A_i}_{C_o})$ obtained by applying the alterations of type A (for the alterations that lead to a robustness less than 100% with C_o) can be observed in the plot in Fig. 5. We observe that the performance of the network C_{NP} (continuous lines in the graphs) is slightly better than the original network C_o trained with unaltered data (dashed lines in the graphs). The other plots are reported online [1]. We observe that Δ_{C_o} in Table II is always greater than or equal to zero, so we have a better overall robustness w.r.t. C_o. Despite C_{NP} has a better accuracy than C_{DA}, it has worse robustness (see $\Delta_{C_{DA}}$). This is because we still have a part of the network trained with only the standard data, so it is possible that, taking the maximum probability from each branch of the network, some images could be classified in the wrong way but with a higher probability by the original branch of the network. For C_{NP}, we cannot compute the adversariability because no adversarial attack has been proposed for parallel networks (to the best of our knowledge), and existing attacks are not suitable because images that can be adversarial for the frozen net C_o may not be adversarial for the parallel net C_{Par} and the other way round.

D. Limited Network Parallelization (LNP)

As we did for data-augmentation in which we applied the limited-data augmentation (obtaining C_{LDA}), we apply the limitation technique also to NP, obtaining the limited-network-parallelization technique. We trained the parallel net C_{LimPar} with the 2,105,930 images obtained by applying the selected alterations (BV, GN, BA, and JC) to the original images in the input set IS_{C_o} that lead to a robustness decrease (i.e., images S_{LDA} used also in LDA). The final network C_{LNP} =$C_o \| C_{LimPar}$ has been obtained with a training duration of 12h02m, and achieved an accuracy of 86.51% over the test set TE_{C_o}; the value is smaller than that obtained by C_{NP}. Its robustness values $rob_A(C_{LNP}, TE_{C_o})$ are shown in Table II. The value $\times n$ in Table II indicates the number of altered images generated from each single standard image and Δ_{C_o} represents the improvement with respect to the robustness of the original CNN C_o, while $\Delta_{C_{NP}}$ indicates the robustness improvement w.r.t. the classifier C_{NP}. The changes of $acc(C_{LNP}, TE^{A_i}_{C_o})$ obtained by applying the alterations of type A on the images in the test set TE_{C_o} can be observed in the plot in Fig. 6. The other plots are reported online [1]. We observe that the performance of the network C_{LNP} (continuous lines in the graphs) is better than the original network C_o trained with unaltered data (dashed lines). Moreover, if we compare the results of C_{LNP} to those of C_{NP} (see Table II), we observe that it has the same performance of C_{NP} for all the alterations, except for brightness variation for which it behaves better. Note that its training takes almost one hour less than the training of C_{NP}. Therefore, the only disadvantage is that the accuracy is slightly smaller. As for C_{NP}, also for C_{LNP} it is not possible to compute the adversariability.



(a) BV: Brightness Variation (%) (b) GN: Gaussian Noise (σ^2) (c) BA: Blur Addition (r) (d) JC: JPEG Compression (q)

Fig. 6. Accuracy modification using C_{LNP} (original CNN with limited-network-parallelization)

TABLE III
SUMMARY OF THE MAIN INFORMATION OF ALL THE METHODS USED TO IMPROVE THE ROBUSTNESS OF A CNN

C	Input size	Train. time	Accuracy	Avg rob.	Adversariability
C_o	210,593	2h08m	86.46%	73.47%	0.38
C_{DA}	2,526,281	14h54m	86.57%	94.81%	0.64
C_{LDA}	2,316,523	13h39m	86.64%	97.29%	0.39
C_{NP}	2,315,688	13h26m	87.10%	80.18%	N/A
C_{LNP}	2,105,930	12h02m	86.51%	81.09%	N/A

E. Discussion

Table III[3] reports a brief summary of the main relevant information about the four methods presented in the previous subsections. We observe that the best solution is to retrain the whole model using LDA technique, because we have a very high resulting average robustness using less input images than the standard DA. In both methods, the training phase requires a lot of time (also considering that we are working on a medium-small dataset). Even the use of the network parallelization technique can lead to an improvement of the robustness of our CNN and, also in this case, the use of the limitation technique can slightly improve the performances in terms of both training time and average robustness.

Looking at Figs. 3, 4, 5 and 6, we notice that generally the higher the alteration is, the better our methods perform. This is reasonable because, including in the input set a image that is only slightly altered, does not give enough increment of the knowledge to the network to significantly improve the accuracy at a certain level of alteration.

Table III also reports the summary of the adversariability data for the classifiers C for which it is possible to compute it: we do not see any correlation between robustness and adversariability, as we obtain good values of adversariability for both C_o and C_{LDA} that are very different in robustness. This seems to confirm that the testing based on the proposed alterations (aiming at increasing robustness) is complementary to that based on adversarial examples (aiming at reducing adversariability); however, further experiments with other case studies are needed to generalize the results.

VII. RELATED WORK

Different testing approaches have been proposed for CNNs. For example, the effectiveness of the traditional mutation tools in the context of neural networks testing has been analyzed in [9]. In other papers, e.g., [10], [36], authors validate a deep neural network using metamorphic testing.

Some of the alterations presented in this paper have already been studied, but no rigorous definition of robustness has been given. In [7], the sensitivity of a CNN when blur or noise alteration occurs is studied. [2] and [14] have studied the robustness of a network where a JPEG compression is applied to the input samples, and [27] analyses many alterations on images acquired by an autonomous driving system. A study of the robustness of CNNs to appearance variability in biomedical images is presented [26]. The authors introduce a new type of layer, called neighbourhood similarity layer (NSL), to improve the robustness w.r.t. changes in the appearance of objects that are not well represented by the training data. [3] introduces CNN-Cert, a general framework capable of certifying robustness on general convolutional neural networks, composed of convolutional layers, max-pooling layers, batch normalization layer, residual blocks, as well as general activation functions.

A methodology similar to the one used in this paper is used in image manipulation detection [18] but, also in this case, a formal definition of robustness has not been introduced.

Authors in [20] made the same as our observation that the adversarial condition is unlikely in many real contexts. They propose a definition of *probabilistic robustness* that guarantees that a neural network is robust with at least (1-ε) probability, given an input probability distribution; however, differently from us, they do not focus on any particular input alteration.

As stated in Sec. I, most of the papers regarding the robustness of the CNNs (and other kinds of neural networks) focus on the adversarial examples. [31] gives a theoretical analysis of the robustness of a NN with respect to adversarial examples. The authors propose a method to find the key reasons why an adversarial example can fool a classifier and to add these oracles to make the network immune to a specific kind of adversarial example. [22] analyzes the lower bound on the robustness to adversarial perturbations, verifying the results on the MNIST and CIFAR-10 data sets. In [15], the authors propose the Cross-Lipschitz regularization functional to improve the robustness of the network against adversarial manipulations. [11] studies the robustness w.r.t. adversarial examples of a generic classifier by using semi-random noise, that has proven to generalize in a simple way the adversarial examples and the random noise.

Studies have also been conducted over the robustness (for adversarial examples) of compressed CNNs models [32] created for mobile apps, where a full CNN cannot be used, be-

[3]Experiments have been run on a server with 264GB of RAM and a Intel® Xeon® E5-2620 CPU.

cause of the limited computational capacity of mobile devices.

VIII. Conclusions

In this paper, we have introduced a novel definition of robustness, for a CNN, w.r.t. unforeseen (but plausible) input modifications. The definition has proven to be simple to be computed and effective to evaluate the behaviour of a CNN with respect to input image alterations. It appears to be applicable in any field and not only in medical image classification. We have also introduced the definition of adversariability, i.e. the vulnerability of a binary classifier w.r.t. adversarial examples. Moreover, we have analyzed different methods to improve the robustness of a network, showing that the best solution is to adopt the *limited-data-augmentation* (LDA) technique.

In the experiments, we have observed that, for some alterations, it is easier to improve the robustness for large alteration values rather than for small ones; as future work, we want to try different methods to improve the performance of a CNN also for small values of alteration.

References

[1] CNN Robustness Evaluation. https://github.com/fmselab/CNNRobustnessEvaluation, 2019.

[2] B. Bayar and M. C. Stamm. On the robustness of constrained convolutional neural networks to JPEG post-compression for image resampling detection. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, mar 2017.

[3] A. Boopathy, T.-W. Weng, P.-Y. Chen, S. Liu, and L. Daniel. CNN-cert: An efficient framework for certifying robustness of convolutional neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3240–3247, jul 2019.

[4] Breastcancer.org. U.S. breast cancer statistics. https://www.breastcancer.org/symptoms/understand_bc/statistics, 2019.

[5] D. Brunet, E. R. Vrscay, and Z. Wang. On the mathematical properties of the structural similarity index. *IEEE Transactions on Image Processing*, 21(4):1488–1499, apr 2012.

[6] D. Castelvecchi. Can we open the black box of AI? *Nature*, 538(7623):20–23, oct 2016.

[7] X. Chai, Q. Ba, and G. Yang. Characterizing robustness and sensitivity of convolutional neural networks for quantitative analysis of mitochondrial morphology. *Quantitative Biology*, 6(4):344–358, dec 2018.

[8] S. Chatterjee. Artefacts in histopathology. *Journal of Oral and Maxillofacial Pathology*, 18(4):111, 2014.

[9] N. Chetouane, L. Klampfl, and F. Wotawa. Investigating the effectiveness of mutation testing tools in the context of deep neural networks. In *Advances in Computational Intelligence*, pages 766–777. Springer International Publishing, 2019.

[10] J. Ding, X. Kang, and X.-H. Hu. Validating a deep learning framework by metamorphic testing. In *2017 IEEE/ACM 2nd International Workshop on Metamorphic Testing (MET)*. IEEE, may 2017.

[11] A. Fawzi, S.-M. Moosavi-Dezfooli, and P. Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems 29*, pages 1632–1640. 2016.

[12] P. Frankl, R. Hamlet, B. Littlewood, and L. Strigini. Evaluating testing methods by delivered reliability [software]. *IEEE Transactions on Software Engineering*, 24(8):586–601, 1998.

[13] Q. Fu, F. Yang, J. Zhao, X. Yang, T. Xiang, G. Huai, J. Zhang, L. Wei, S. Deng, and H. Yang. Bioinformatical identification of key pathways and genes in human hepatocellular carcinoma after CSN5 depletion. *Cellular Signalling*, 49:79–86, sep 2018.

[14] S. Ghosh, R. Shet, P. Amon, A. Hutter, and A. Kaup. Robustness of deep convolutional neural networks for image degradations. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, apr 2018.

[15] M. Hein and M. Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems 30*, pages 2266–2276. 2017.

[16] A. Janowczyk and A. Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, 7(1):29, July 2016.

[17] R. Kemker, M. McClure, A. Abitino, T. L. Hayes, and C. Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings AAAI-18, IAAI-18, and EAAI-18*, pages 3390–3398. AAAI Press, 2018.

[18] D.-H. Kim and H.-Y. Lee. Image manipulation detection using convolutional neural network. *International Journal of Applied Engineering Research*, 12(21):11640–11646, 2017.

[19] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. *International Journal of Computer Vision*, 127(5):456–476, May 2019.

[20] R. Mangal, A. V. Nori, and A. Orso. Robustness of neural networks: A probabilistic and practical approach. In *Proceedings of the 41st International Conference on Software Engineering: New Ideas and Emerging Results*, ICSE-NIER '19, pages 93–96, Piscataway, NJ, USA, 2019. IEEE Press.

[21] D. Marijan, A. Gotlieb, and M. K. Ahuja. Challenges of testing machine learning based systems. In *2019 IEEE International Conference On Artificial Intelligence Testing (AITest)*, pages 101–102, April 2019.

[22] J. Peck, J. Roels, B. Goossens, and Y. Saeys. Lower bounds on the robustness to adversarial perturbations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 804–813. USA, 2017.

[23] W. Rawat and Z. Wang. Deep convolutional neural networks for image classification: A comprehensive review. *Neural Computation*, 29(9):2352–2449, sep 2017.

[24] A. Rosebrock. Breast cancer classification with keras and deep learning. https://www.pyimagesearch.com/2019/02/18/breast-cancer-classification-with-keras-and-deep-learning/, 2019.

[25] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Conference Track Proceedings*, 2014.

[26] T. Tasdizen, M. Sajjadi, M. Javanmardi, and N. Ramesh. Improving the robustness of convolutional networks to appearance variability in biomedical images. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*. IEEE, apr 2018.

[27] Y. Tian, K. Pei, S. Jana, and B. Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th International Conference on Software Engineering*, ICSE '18, pages 303–314, New York, NY, USA, 2018. ACM.

[28] S. Uchida, S. Ide, B. K. Iwana, and A. Zhu. A further step to perfect accuracy by training CNN with larger data. In *2016 15th Int. Conference on Frontiers in Handwriting Recognition (ICFHR)*. IEEE, oct 2016.

[29] M. Uličný, J. Lundström, and S. Byttner. Robustness of deep convolutional neural networks for image recognition. In *Intelligent Computing Systems*, pages 16–30. Springer International Publishing, Cham, 2016.

[30] D. A. van Dyk and X.-L. Meng. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50, mar 2001.

[31] B. Wang, J. Gao, and Y. Qi. A theoretical framework for robustness of (deep) classifiers against adversarial samples. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*, 2017.

[32] A. W. Wijayanto, C. J. Jin, K. Madhawa, and T. Murata. Robustness of compressed convolutional neural networks. In *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, dec 2018.

[33] X. Xie, L. Ma, F. Juefei-Xu, M. Xue, H. Chen, Y. Liu, J. Zhao, B. Li, J. Yin, and S. See. DeepHunter: A coverage-guided fuzz testing framework for deep neural networks. In *Proceedings of the 28th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ISSTA 2019, pages 146–157, New York, NY, USA, 2019. ACM.

[34] H. Ye, Z. Wu, R.-W. Zhao, X. Wang, Y.-G. Jiang, and X. Xue. Evaluating two-stream CNN for video classification. In *Proc. of the 5th ACM on International Conference on Multimedia Retrieval*. ACM Press, 2015.

[35] X. Yuan, P. He, Q. Zhu, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, 30(9):2805–2824, Sep. 2019.

[36] M. Zhang, Y. Zhang, L. Zhang, C. Liu, and S. Khurshid. DeepRoad: GAN-based metamorphic testing and input validation framework for autonomous driving systems. In *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering - ASE 2018*. ACM Press, 2018.