# Towards a Certification of Deep Image Classifiers against Convolutional Attacks

Mallek Mziou-Sallami[1,3] and Faouzi Adjed[1,2]
[1]IRT SystemX, Palaiseau, France
[2]Expleo Group, Montigny-le-Bretonneux, France
[3]CEA, Evry, France

Keywords: NN Robustness, Uncertainty in AI, Perception, Abstract Interpretation.

Abstract: Deep learning models do not achieve sufficient confidence, explainability and transparency levels to be integrated into safety-critical systems. In the context of DNN-based image classifier, robustness have been first studied under simple image attacks (2D rotation, brightness), and then, subsequently, under other geometrical perturbations. In this paper, we intend to introduce a new method to certify deep image classifiers against convolutional attacks. Using the abstract interpretation theory, we formulate the lower and upper bounds with abstract intervals to support other classes of advanced attacks including image filtering. We experiment the proposed method on MNIST and CIFAR10 databases and several DNN architectures. The obtained results show that convolutional neural networks are more robust against filtering attacks. Multilayered perceptron robustness decreases when increasing number of neurons and hidden layers. These results prove that the complexity of DNN models improves prediction's accuracy but often impacts robustness.

## 1 INTRODUCTION

Experiments showed that DNNs are overly sensitive to small disturbances in their input data. It is well known that one can generate adversarial examples by applying geometrical transformation (Biggio et al., 2013; Szegedy et al., 2013). In the other hand, to embed DNN-based algorithm into safety-critical systems such as aircraft systems or autonomous vehicles, robustness proof remains essential. Moreover, the non-transparency nature of DNNs makes robustness verification a major challenge facing the many different possibilities of disturbances. For example, to evaluate the lightness effects, we have to verify whether if a NN-based image classifier is robust against brightness shift or not. In other words, whether the classification results are invariant under affine variation of pixels intensity or not. This assumption hugely simplifies the reality. If we use an hypothesis closer to the reality, a more complicated modeling have to be considered, such as, the case where, views taken are linked by perspective transformations in the presence of thick fog inducing a blur on the images. The question to be addressed therefore is how to guarantee the robustness of DNN model? To overcome this challenge and come up with a solution, many DNN robustness verification frameworks have been proposed such as DeepSymbol (Li et al., 2019), ERAN (Singh et al., 2018), DeepG (Balunovic et al., 2019), Reluplex (Katz et al., 2017), PLANET (Bunel et al., 2018) and PRODeep (Li et al., 2020). The common idea behind all these verification tools is the prediction of disturbed input model using an approximate neural network behavior. In another approach, researchers focus on robustness study during the design phase to build more accurate models against such transformations (Xiao et al., 2018; Jaderberg et al., 2015).

It should be noted that robustness verification is a different approach from the empirical evaluation of neural networks. Table [1] summarize some research in the two approaches classified according to image's attacks. Despite the multitude of methods dealing empirically with the evaluation of neural networks (Xiao et al., 2018; Engstrom et al., 2019; Kanbak et al., 2018; Goodfellow et al., 2009; Fawzi et al., 2017; Alaifari et al., 2018), robustness certification still a remaining challenge. In this context, formal methods are widely applied to assess the local and global robustness of deep neural networks. Typically, static analysis with abstract interpretation or SAT solvers approaches are applied to neural networks and leverages the important progress of formal

Mziou-Sallami, M. and Adjed, F.
Towards a Certification of Deep Image Classifiers against Convolutional Attacks.
DOI: 10.5220/0010870400003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 2*, pages 419-428
ISBN: 978-989-758-547-0; ISSN: 2184-433X

Table 1: Summary of research work related to DNNs verification against image's attacks or deformations.

| DNNs Certification | | | | DNNs Evaluation |
|---|---|---|---|---|
| DeepG | | DeepPoly | | (Xiao et al., 2018) |
| Attacks | References | Attacks | References | (Engstrom et al., 2019) |
| Translation | (Balunovic et al., 2019) | Brightness,$L_\infty$ | (Singh et al., 2018; Gehr et al., 2018; Singh et al., 2019b; Singh et al., 2019a) | (Kanbak et al., 2018) |
| Rotation 2D | (Balunovic et al., 2019) | FGSM | (Singh et al., 2018; Gehr et al., 2018; Singh et al., 2019b; Singh et al., 2019a) | (Goodfellow et al., 2009) |
| Scaling | (Balunovic et al., 2019) | Rotation 2D | (Singh et al., 2019a) | (Fawzi et al., 2017) |
| Sheering | (Balunovic et al., 2019) | Rotation 3D | (Sallami et al., 2019) | (Alaifari et al., 2018) |
| Vector Fields | (Ruoss et al., 2020) | Convolution | (Sallami et al., 2019) | |

methods over the last decades. Such approaches estimate bounds on the perturbation of the inputs and formally guarantee the same DNN prediction within these bounds. However, formal methods over DNN for image perception system have often been applied to simple image attacks. Existing robustness verification tools often consider norm based robustness or brightness robustness. For example, authors, in (Singh et al., 2018; Gehr et al., 2018; Singh et al., 2019b), have introduced a neural network certification method based on the abstract interpretation. Experimental results on MNIST and CIFAR databases have proven the capability of a such system to certify the robustness against attacks including simple contrast, FGSM (Fast Gradient Signed Method) noise and $L_\infty$ attacks. Other works have explored certification against geometric transformation such as 2D rotation (Singh et al., 2019a), scaling (Balunovic et al., 2019) and 3D rotations (Sallami et al., 2019). This paper is the continuity of the earlier work proposed in (Sallami et al., 2019). We focus on formal methods for NN-based object recognition systems and we introduce a new method to assess the robustness of a given NN-based image classifier under convolutional attacks. We propose a new algorithm to compute lower and upper bounds abstract elements that allow us to verify the robustness of a DNN against filtering attacks. To the best of our knowledge, the proposed method is currently the state-of-the-art system for certifying robustness of neural networks under filtering and convolutional attacks. Fig.1 illustrates this fact by briefly describing the proposed system.

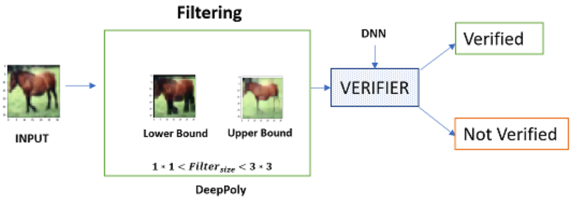


Figure 1: End-to-end DNN robustness verification against filtering.

The following article is organized as follows: In section 2, a brief review of verification approaches is given with a focus on the abstract interpretation theory and its distinctive contribution to neural network verification. We present in Section 3 the system designed to certify filtering robustness. Our experimental settings and results will be given respectively in section 4 and 5. Finally, in Section 6, we present our conclusions and we discuss some future perspectives.

## 2 BACKGROUND AND RELATED WORKS

Several techniques allowing the verification of neural networks are developed in the literature. The abstract interpretation, which is one among these techniques will be presented in more details in the next sections. We will recall the formulation of the lower bound and upper bound for both contrast and geometric attacks.

### 2.1 Abstract Interpretations for Neural Network Certification

Abstract interpretation is an approach to infer semantic properties from computer programs and to demonstrate their soundness (Cousot and Cousot, 1992). Static analysis by abstract interpretation allows to automatically extract information about all possible execution states of a computer program. It is used for automatic debugging, compilers optimizing, code execution and the certification of programs against some classes of bugs. One of the first application of static analysis with abstract interpretation for neural networks is implemented by Pulina and Tacchella (Pulina and Tacchella, 2010) in 2010. However, their work was focused on shallow NN (MLP).

Recently, several scientific contributions adapted this method for verifying the robustness property of larger neural networks by proposing abstract transformers[1] for each type of activation function (Singh et al., 2018; Gehr et al., 2018; Singh et al., 2019b; Singh et al., 2019a). In what follows, we recall in a synthetic way some concepts.
Let $\bar{X}$ be a given input. $\bar{X}$ may undergo a deformation or even an attack. In such a case, $\bar{x} \in \bar{X}$ will be

[1] Abstract transformer is a step of abstract interpretation construction which is a abstract set that includes all concrete outputs corresponding to the real data

transformed into $\bar{x}_{\varepsilon}$. The original inputs perturbed by $\varepsilon$ are denoted by $R_{\bar{X},\varepsilon}$. Verifying the robustness property for $R_{\bar{X},\varepsilon}$ consists of checking the property over the whole possible perturbation of $\bar{X}$.

Let $C_L$ be the output sets with the same label $L$. We denote $\bar{Y}$ as the set of each prediction for each element in $R_{\bar{X},\varepsilon}$.

$$C_L = \{\bar{y} \in \bar{Y} \,|\, \arg\max \bar{y}_i = L\} \qquad (1)$$

The ($R_{\bar{X},\varepsilon}$,$C_L$) robustness property is verified only if the outputs $O_R$ of $R_{\bar{X},\varepsilon}$ are included in $C_L$. However, in reality, we are not able to control the behavior of hidden layers. Accordingly, we have no knowledge about $O_R$. The abstract interpretation is an alternative to face this defect. In fact, it allows to determine an abstract domain thought transformers and verifies the inclusion condition in new abstract domains $\alpha_R$, which is an abstraction of $\bar{X}$. We denote the output abstract domain $\alpha_R^O$. The ($R_{\bar{X},\varepsilon}$,$C_L$) property is checked:

- If the outputs $O_R$ of $R_{\bar{X},\varepsilon}$ are included in $C_L$.
- If the outputs $\alpha_R^O$ of the abstraction of $R_{\bar{X},\varepsilon}$ ($\alpha_R$ ) are included in $C_L$.

It seems necessary to define abstract transformers that are precise for the different existing activation functions. However, the sacalability is one of the major shortcoming in the implementation of this approach. Therefore, Singh et al. (Singh et al., 2018) proposed an alternative solution, called DeepZ analyzer, to deal with the scalability problem. DeepZ allows also to certify the robustness of neural network. Another analyzer, called DeepPoly, have been introduced by Gerhr et al. (Gehr et al., 2018). This approach relies on a novel abstract domain that merge polyhedron with floating point and intervals. The approach is denoted AI$^2$ (Gehr et al., 2018). This analyzer may automatically prove the robustness of different neural networks architectures, including convolutional neural networks. The test results demonstrate that $AI^2$ analyzer is fully accurate and may be used to certify the most recent defense efficiency for neural networks. It is characterized by its high precision arithmetic in floating point and it manages several activation functions, including **ReLU** (*REctified Linear Unit*),**TanH**, and **Sigmoid**. It is worth mentioning that DeepZ is based on the abstract domains and more particularly the zonotopes (Ghorbal et al., 2009). DeepPoly analyser supports refine transformation as well as modern activation functions such as **ReLU**, **sigmoid**, **TanH** and **maxpool**. According to authors, DeepPoly is the most precise analyser compared to AI$^2$ and DeepZ and manage also deep convolutional neural networks. This method has been used to check complex perturbation, including 2D rotation.

## 2.2 Lower and Upper Bound for Contrast and Geometrical Attacks

In the abstract interpretation for computer science (Henry, 2014), lower bound and upper bound are defined as longest execution time case. Similarly, for AI$^2$, the lower and upper bounds are the limits of accepted disturbance. In the case of luminosity disturbance, the lower bound (LB) and the upper bound (UB) are respectively the minimum and maximum brightness values. We can approximate it to a brightness shift. Indeed, these two values allow us to define the abstract intervals that we need. In the case of plane rotation, the contribution of the neighboring pixels to the intensity of the disturbed pixel is proportional to its distance from the initial pixel. This approximation lets us estimate the possible LB and UB, which give us the polytopes in which each rotated pixel is going to end. Combined with abstract intervals, they allow us to compute the needed abstract domain. It is recommended to add a tracing algorithm which split the rotation interval into sub-intervals. Such procedure validate whether the neural network is able to recognize the object when it changes orientation in the image.

# 3 PROPOSED METHOD

Image attacks have been extensively studied over the last few years for solving real world problems in several areas. Different attacks were explored on images and videos as summarized by Vassaux et al. (Vassaux et al., 2002). We recall here the three following categories of image attacks:

- **Common Signal Processing:** It is necessary that the neural network recognize the object despite common signal processing operations being applied to the input image. Among these operations, we name few: filtering, re-sampling, requantization, compression, color contrast and enhancement.
- **Occlusion Attack:** is defined by masking some parts of the images like cropping.
- **Common Geometric Distortions:** which are geometric deformation of the image, such as rotation, translation and scaling.

We focus our exploration on the first category, specifically on convolutional (filtering) attacks. With regard to the geometric image distortions, the basic ones include rotation, uniform scale change, reflection and shearing are studied in Sallami et al. (Sallami et al., 2019), and by Balunovic et al. (Balunovic et al., 2019) by proposing different and more precise abstract domains for NN geometrical certification.

The particularity of this work is to avoid the specificity of the disturbance. As a matter of fact, the proposed method allows the evaluation of the model depending on the internal structural data and not by adding an ε-perturbation. The filtering attack depends on the structural variability of the neighborhood of each pixel encompassing all disturbances. Using the convolution filter with different kernel size, the proposed method may certify all the possible values of the filter as illustrated in figure 2.

## 3.1 Filtering Attack

Filtering is a practice for enhancing images. Mathematically, the filtering is the result of the convolution of a function (image) with a kernel. Suppose that we have a filter $H$ with $d \times d$ size applied to the image $I$. For every pixel in the filtered image $I'$ will have the following value:

$$I'(i,j) = (f*h)(i,j) = \sum_{n=-\frac{d-1}{2}}^{n=\frac{d-1}{2}} \sum_{m=-\frac{d-1}{2}}^{m=\frac{d-1}{2}} f(i-n,j-m)h(n,m) \tag{2}$$

In image processing field, the resulting image depends on the choice of the kernel. In fact, it can be used for blurring, enhancement, smoothing and filtering, etc. For example, the Gaussian filter is used for noising and denoising depending on the variance of the kernel. In other words, the Gaussian distribution is approximated by a convolution kernel to build a convolution matrix (Gedraite and Hadad, 2011). In the real world case, the noise is applied randomly like a fog or snow. Therefore, when the image is captured, some pixels will be masked. Consequently, the recognition of an object in the image depends strongly on the size of the mask applied. With the convolution, locally (pixel by pixel), we can fit the weights of the kernel to reproduce the same noise. However, it will not be possible to create a kernel for each pixel, therefore we suggest to build an interval for every pixel and to verify it formally by abstract interpretation approach. It can be seen as a 3D image with variable voxels.

## 3.2 Lower and Upper Bound for Convolutional Attacks

Our approach consists in defining a lower bound (LB) and an upper bound (UB) independently from the applied filter coefficients. The pixel on the filtered image is estimated according to the size of the filter. The LB and the UB of each pixel are computed using the pixel's neighborhood. Indeed, the value of pixel after the convolution will have the minimum value of its neighborhood for the LB and the maximum for the UB. i.e,

$$UB_{I(i,j)} = \sum_{n=-\frac{d-1}{2}}^{n=\frac{d-1}{2}} \sum_{m=-\frac{d-1}{2}}^{m=\frac{d-1}{2}} f(i-n,j-m)\mathbb{1}_{\max \mathcal{N}(f(i,j))} \tag{3}$$

$$LB_{I(i,j)} = \sum_{n=-\frac{d-1}{2}}^{n=\frac{d-1}{2}} \sum_{m=-\frac{d-1}{2}}^{m=\frac{d-1}{2}} f(i-n,j-m)\mathbb{1}_{\min \mathcal{N}(f(i,j))} \tag{4}$$

where $\mathcal{N}(f(i,j))$ defines the neighborhood of the pixel $(i,j)$, $f$ defines the original image and $h$ is the filter. The figure 2 illustrates an example of a selection of the upper and the lower bounds. It illustrates also the indicative minimum neighborhood ($\mathbb{1}_{\min \mathcal{N}(f(i,j))}$) and the indicative maximum neighborhood ($\mathbb{1}_{\max \mathcal{N}(f(i,j))}$) by respectively the convolution kernels up and down. The final LB and UB correspond respectively to the min and the max values between the LB and the UB images related to the minimum and the maximum filter size. Algorithms 1 and 2 describe in more details the different steps, where $p_h$ and $p_v$ are the horizontal and vertical position of the pixel in the image, $w$ and $h$ are the filters and dim defines its size. In algorithm 1, depending on the filter size, we extract for each pixel its neighborhood, whereas algorithm 2 computes the lower and upper bounds of the selected neighborhood.

# 4 EXPERIMENTAL SETTINGS AND RESULTS

This section is dedicated to highlight our experimental settings and results for evaluating the effectiveness of our approach used to verify the robustness properties against the convolution attacks.

## 4.1 Experimentation Settings

Herein, we point out the two main settings that allow us to carry out our experiment. The first one is the used datasets and the second one is the a set of the neural networks, to evaluate, pre-trained on the two datasets. The details of the implementation are presented in the subsection 4.2.

### 4.1.1 Datasets

Well known datasets, MNIST and CIFAR, are used to evaluate the impact of filter's size on the robustness of the selected neural networks models.

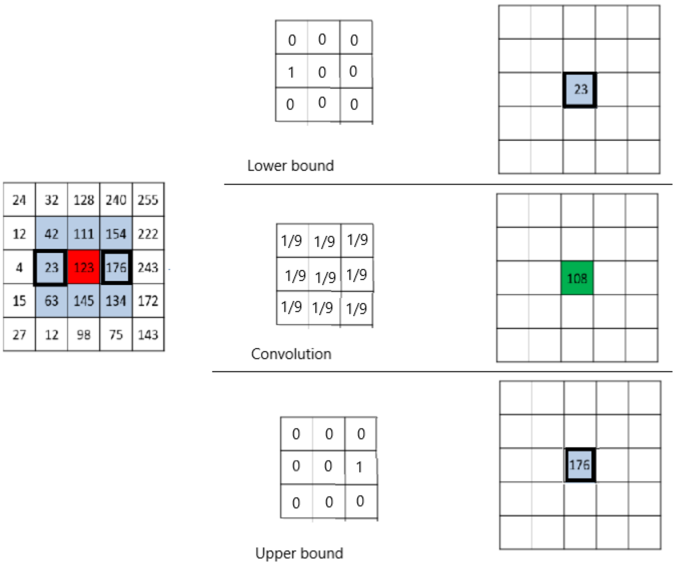


Figure 2: Lower and upper bound for convolution. The first line convolution illustrates the lower bound kernel, the second line convolution represents an example of convolution (Box blur), whereas the third line illustrates the upper bound kernel.

Algorithm 1: Neighbors.

```
procedure PROCEDURE NEIGHBORS
Require:Image $\in [0,255]^{m\times n}$, dim $\in [1,N]$
Require:$p_h \in [1,m]$, $p_v \in [1,n]$
    $Neighbors \leftarrow []$
    for $t_1 \in \{-\mathrm{dim},\ldots,\mathrm{dim}\}$; $t_2 \in \{-\mathrm{dim},\ldots,\mathrm{dim}\}$ do
        if $\Big((0 < p_h - t_1 < m)$ & $(0 < p_v - t_2 < n)\Big)$ then
            $Neighbors \leftarrow Neighbors \cup \mathrm{Image}\big[p_h - t_1{:}p_h + t_1;\ p_v - t_2{:}\ p_v + t_2\big]$
        end if
    end for
    Return $Neighbors$
end procedure
```

Algorithm 2: Lower and Upper Bound for convolution.

```
procedure PROCEDURE CONVOLUTION_LOWER_UPPER_BOUND
Input: $I \in [0,255]^{m\times n}; h, w \in [\![1,T]\!]$, $T$ size of the filter
    $I_{LB}, I_{UB} \leftarrow I$
    for $p_h \in \{1,\ldots,m\}; p_v \in \{1,\ldots,n\}$ do
        $L_1[p_h,p_v] \leftarrow \min(Neighbors(I,h,p_h,p_v))$
        $U_1[p_h,p_v] \leftarrow \max(Neighbors(I,h,p_h,p_v))$
        $L_2[p_h,p_v] \leftarrow \min(Neighbors(I,w,p_h,p_v))$
        $U_2[p_h,p_v] \leftarrow \max(Neighbors(I,w,p_h,p_v))$
        $I_{LB}[p_h,p_v] \leftarrow \min(L_1[p_h,p_v], L_2[p_h,p_v])$
        $I_{UB}[p_h,p_v] \leftarrow \max(U_1[p_h,p_v], U_2[p_h,p_v])$
    end for
    Return $I_{LB}, I_{UB}$
end procedure
```

1. MNIST Dataset (LeCun, 1998; LeCun et al., 1998) contains grayscale images of size 28 × 28 pixels. It consists of a handwritten digits size normalized and centred in fixed-size image where 50 images are selected for the evaluation.
2. CIFAR 10 dataset (Krizhevsky et al., 2009) contains a color images of size 32 × 32 pixels. It contains 10 different and exclusive classes where 100 images are selected for the evaluation.

The robustness criterion of the evaluation is the fraction between the number of verified images $(VI)$ under the attack and the total number of well classified ones $(WCI)$. Then, the robustness metric $R_{ess}$ is set to:

$$R_{ess} = \frac{\#VI}{\#WCI} \quad (5)$$

The equation 5 values vary between 0 and 1. The more the results of the equation is closer to 1, the more the combination, neural network model and filter size is robust. The use of abstract interpretation (see section 2.1) certify formally the robustness of the given combination.

#### 4.1.2 Evaluated Neural Networks

Four fully connected neural network models with 3, 6 and 9 layers and one convolutional are selected for the MNIST dataset. Whereas, for CIFAR-10, four models are used: three of them are fully connected layers with 4, 6 and 9 layers respectively and a last one convolutional with 3 layers. Table 2 resumes the characteristic of the evaluated models giving more details, such as the activation function used and the number parameters estimated (#units).

### 4.2 Implementation

The abstract domain for convolutional attacks is implemented in Python. As abstract interpretation analyzer, DeepPoly solution (see section 2.1) has been used. This latter is based on two main libraries[2] ERAN and ELINA, coded in respectively Python and C programming languages. The pre-trained models presented in table 2 are implemented, where fully connected layers and convolutional models are evaluated using MNIST and CIFAR10 datasets. The size of the filters varies from 2 × 2 to 18 × 18 which is equivalent to dim $\in \{1, 2, \ldots, 9\}$ in algorithm 1.

## 5 RESULTS

To investigate the efficacity and scalability of our certification method, we evaluate it on a different size of filters. The effect of the convolution attack, based on the size of the filter, is illustrated in the examples given in figures 5 and 7 for MNIST dataset and figures 6 and 8 for CIFAR10 dataset. For MNIST examples, it is obviously visible that the lower bound attack reduces the white (clear) pixels and replace it with dark ones, and the upper bound attack increases the white pixels. These examples highlight the effect of the borders and object structure in the classification. The CIFAR10 examples highlights this effect on the RGB images making darker and clearer the image for respectively lower and upper bounds. In RGB images, the convolution attack shows also a blurring of the object. From the examples given in figures 7 and 8, where the size of the convolution kernel is larger, the attacks damages further the object in the image compared to figures 5 and 6.

The lower and upper bounds are also highlighted attacking only one channel of the RGB images given in CIFAR10. Figure 9 illustrates respectively the attack of red, green and blue channels. The lower bound reduces the effect of the channel and the upper one increases it. This attack could simulate the failure of the channel sensors of a camera.

As described in earlier sections, we applied our proposed method to prove a neural network robustness against filtering attacks. Specifically, our analysis can prove that the MNIST network can classify a given image of a digit correctly even if every pixel intensity is an aggregation of its neighbors intensity. Filtering according to $x$ and $y$ generate a blurry appearance on the image. So, to test the robustness of the neural network, just consider a two dimensional filter with a size between $dim_1$ and $dim_2$. Figure 3 (resp 4) show an example of robustness function for $dim_1 \in [0, 9]$, $dim_2 \in [0, 9]$ on MNIST dataset (resp CIFAR dataset). The results of MNIST robustness, using the equation 5, are given in the figures 3. The results show the square filter i.e 1 in the x-axis is equivalent to the filter 3 × 3, in other words $dim1 = 1$ and $dim2 = 1$. The y-axis graphs represents the robustness metric. For example, in figure 3 the robustness of *convMaxPool* model is equal to 30% when MNIST images are filtered with a filter 3 × 3 (x-axis = 1). From MNIST results, we can see that the convolutional layers model is more robust than fully connected layers models especially for filters with size not exceeding 13 × 13. This is expected since the convolutional models capture shapes using the convolution operator. Consequently, filtering has a low

[2] All needed libraries for the implementation are given in the following Github https://github.com/eth-sri/eran

Table 2: Neural Networks used for the evaluation of convolutional attacks.

| Dataset | Model | Type | #units | #Layers | Activation function |
|---|---|---|---|---|---|
| MNIST | $3\times50$ | fully connected | 110 | 3 | ReLU |
| | $3\times100$ | fully connected | 210 | 3 | ReLU |
| | $6\times100$ | fully connected | 510 | 6 | ReLU |
| | $9\times200$ | fully connected | 1,610 | 9 | ReLU |
| | convMaxPool | convolutional | 13,798 | 9 | ReLU |
| CIFAR10 | $4\times100$ | fully connected | 140 | 4 | ReLU |
| | $6\times100$ | fully connected | 610 | 6 | ReLU |
| | $9\times200$ | fully connected | 1,810 | 9 | ReLU |
| | convMaxPool | convolutional | 53,938 | 9 | ReLU |

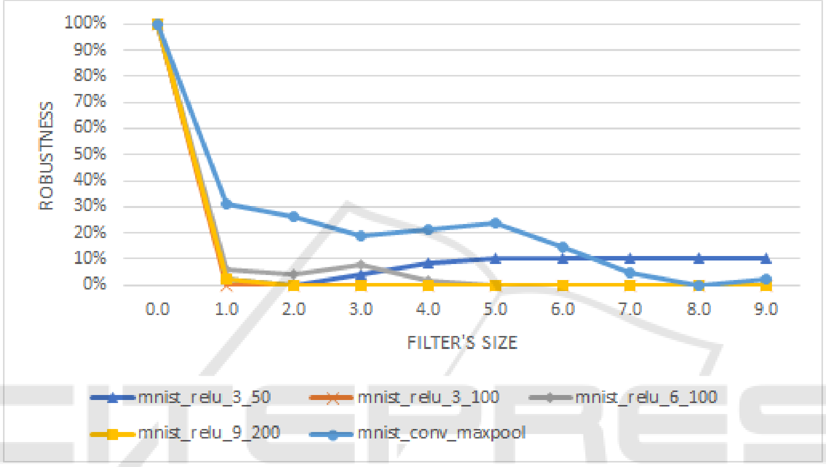


Figure 3: MNIST robustness variation according to the filer's size.

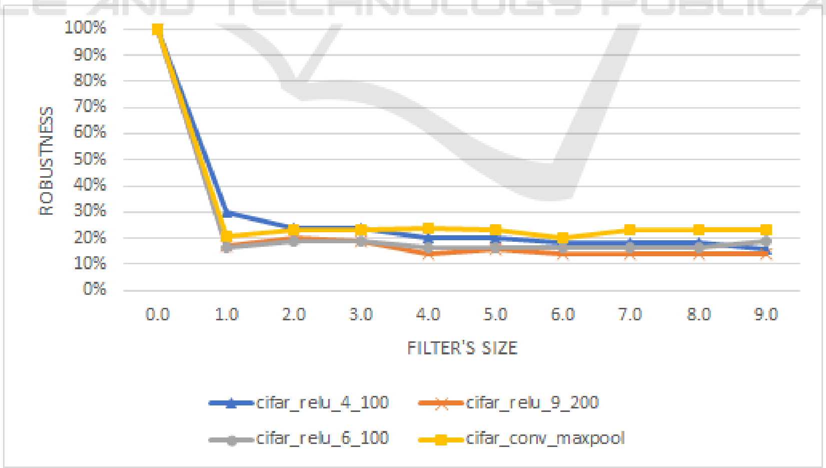
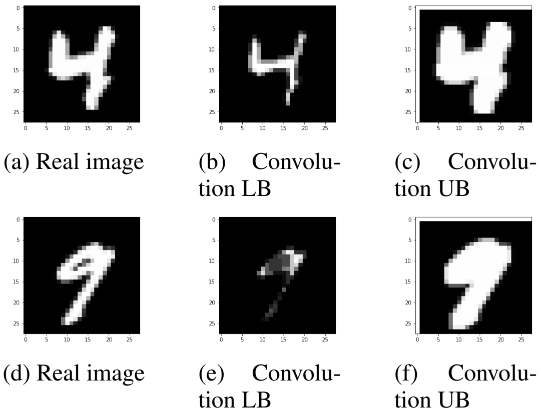


Figure 4: CIFAR robustness variation according to the filer's size.

impact on robustness although it modifies the images content. Beyond this size ($13\times13$), model robustness decrease, same for fully connected models with the exception of the *mnist_relu_3_50* model. Indeed, filtering the images with a large filter greatly modifies the initial information in the images. *mnist_relu_3_50* model did not capture details on the images during training stage. Therefore, it is invariant with respect to the filtering operation.

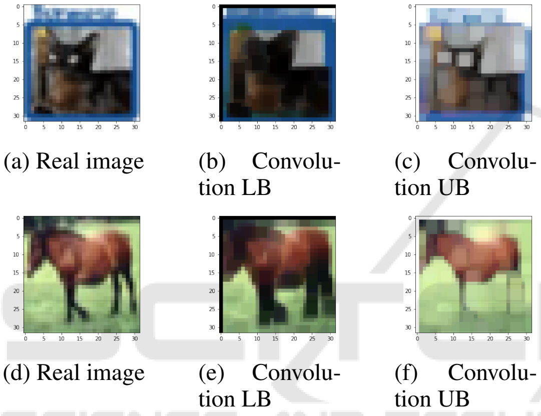
Result obtained using CIFAR10 dataset, illustrated in figure 4, confirm that multilayer perceptron models are more sensitive to filtering regardless of the dataset. However, the robustness on CIFAR is overall more important by comparing it with robustness on



(a) Real image (b) Convolution LB (c) Convolution UB

(d) Real image (e) Convolution LB (f) Convolution UB

Figure 5: MNIST Database: Lower and Upper bounds in Convolutional attacks (for dim1=0 and dim2=1).



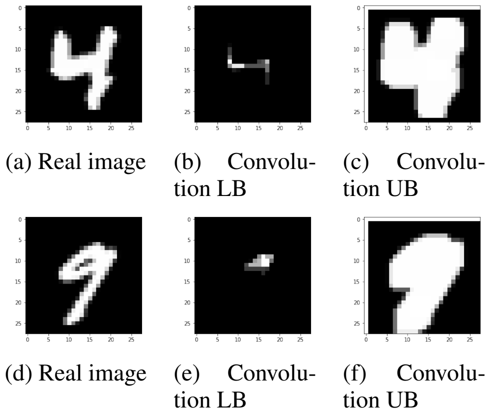
(a) Real image (b) Convolution LB (c) Convolution UB

(d) Real image (e) Convolution LB (f) Convolution UB

Figure 6: CIFAR10 Database:Lower and Upper bounds in Convolutional attacks (for dim1=0 and dim2=1).

MNIST. This is due to several factors. First, the CIFAR images are larger which changes the proportion between the filter size and the image size. Second, in the presence of a convolution-type disturbance, part of the information will be deleted from the initial image. Cifar models may persist because images contain more information and more texture. Such interpretation can be deduced by comparing image 5e in figure 5 with image 6e in figure 6. For this reason the NN models which are trained on CIFAR are, overall, more robust against convolutional attacks.

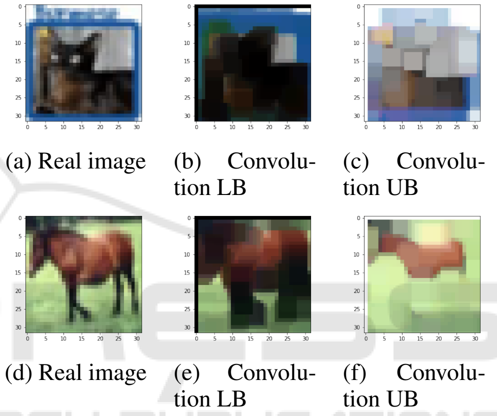
Nevertheless, all models are sensitive to convolution attacks with a maximum of 30% of robustness. This could be interpreted as the model learns more on the object texture than on the object structure, this makes the model lose (dramatically) on robustness even with a small disturbance of the structure.



(a) Real image (b) Convolution LB (c) Convolution UB

(d) Real image (e) Convolution LB (f) Convolution UB

Figure 7: Lower and Upper bounds in Convolutional attacks (for dim1=0 and dim2=2).



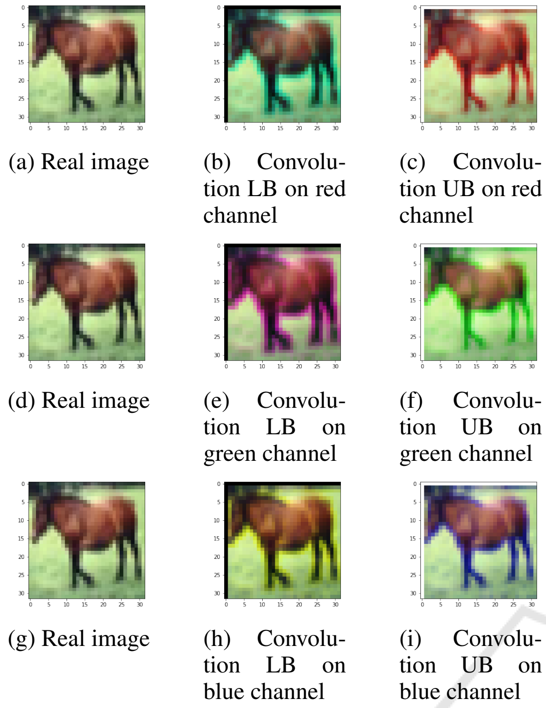
(a) Real image (b) Convolution LB (c) Convolution UB

(d) Real image (e) Convolution LB (f) Convolution UB

Figure 8: Lower and Upper bounds in Convolutional attacks (for dim1=0 and dim2=2).

## 6 CONCLUSIONS

We introduced a new method for certifying deep neural networks robustness against filtering attacks. The core idea of this work is the extension of the abstract interpretation based certification method, which is an abstract domain suitable to compute the LB and the UB in the presence of convolution. To the best of our knowledge this is the first study that tries to prove, by the proposed method we showed, for the first time, how to prove the robustness of a neural network when the input image is convoluted by a random kernel. We tested our method with DeepPoly analyzer, and evaluated it extensively on a wide range of networks of different sizes and different architectures. We believe this work is a promising step towards more effective evaluation of deep neural networks against convolutional attacks such as blurring, enhancement, smoothing and filtering. In a future study, we propose to optimize our abstract domain to achieve more precision.



(a) Real image

(b) Convolution LB on red channel

(c) Convolution UB on red channel

(d) Real image

(e) Convolution LB on green channel

(f) Convolution UB on green channel

(g) Real image

(h) Convolution LB on blue channel

(i) Convolution UB on blue channel

Figure 9: Lower and Upper bounds in Convolutional attacks (for dim1=0 and dim2=1).

We will also consider the optimisation of neural network architecture for training neural networks to be provably robust against convolutional attacks.

## REFERENCES

Alaifari, R., Alberti, G. S., and Gauksson, T. (2018). Adef: an iterative algorithm to construct adversarial deformations. *arXiv preprint arXiv:1804.07729*.

Balunovic, M., Baader, M., Singh, G., Gehr, T., and Vechev, M. (2019). Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems*, pages 15313–15323.

Biggio, B., Corona, I., Maiorca, D., Nelson, B., Šrndić, N., Laskov, P., Giacinto, G., and Roli, F. (2013). Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer.

Bunel, R. R., Turkaslan, I., Torr, P., Kohli, P., and Mudigonda, P. K. (2018). A unified view of piecewise linear neural network verification. In *Advances in Neural Information Processing Systems*, pages 4790–4799.

Cousot, P. and Cousot, R. (1992). Abstract interpretation and application to logic programs. *The Journal of Logic Programming*, 13(2-3):103–179.

Engstrom, L., Tran, B., Tsipras, D., Schmidt, L., and Madry, A. (2019). Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR.

Fawzi, A., Moosavi-Dezfooli, S.-M., and Frossard, P. (2017). The robustness of deep networks: A geometrical perspective. *IEEE Signal Processing Magazine*, 34(6):50–62.

Gedraite, E. S. and Hadad, M. (2011). Investigation on the effect of a gaussian blur in image filtering and segmentation. In *Proceedings ELMAR-2011*, pages 393–396. IEEE.

Gehr, T., Mirman, M., Drachsler-Cohen, D., Tsankov, P., Chaudhuri, S., and Vechev, M. (2018). Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE.

Ghorbal, K., Goubault, E., and Putot, S. (2009). The zonotope abstract domain taylor1+. In *International Conference on Computer Aided Verification*, pages 627–633. Springer.

Goodfellow, I., Lee, H., Le, Q., Saxe, A., and Ng, A. (2009). Measuring invariances in deep networks. *Advances in neural information processing systems*, 22:646–654.

Henry, J. (2014). *Static Analysis by Abstract Interpretation and Decision Procedures*. PhD thesis, Université de Grenoble.

Jaderberg, M., Simonyan, K., Zisserman, A., et al. (2015). Spatial transformer networks. *Advances in neural information processing systems*, 28:2017–2025.

Kanbak, C., Moosavi-Dezfooli, S.-M., and Frossard, P. (2018). Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449.

Katz, G., Barrett, C., Dill, D. L., Julian, K., and Kochenderfer, M. J. (2017). Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer.

Krizhevsky, A., Hinton, G., et al. (2009). Learning multiple layers of features from tiny images.

LeCun, Y. (1998). The mnist database of handwritten digits. *http://yann. lecun. com/exdb/mnist/*.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, J., Liu, J., Yang, P., Chen, L., Huang, X., and Zhang, L. (2019). Analyzing deep neural networks with symbolic propagation: towards higher precision and faster verification. In *International Static Analysis Symposium*, pages 296–319. Springer.

Li, R., Li, J., Huang, C.-C., Yang, P., Huang, X., Zhang, L., Xue, B., and Hermanns, H. (2020). Prodeep: a platform for robustness verification of deep neural networks. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1630–1634.

Pulina, L. and Tacchella, A. (2010). An abstraction-refinement approach to verification of artificial neural networks. In *International Conference on Computer Aided Verification*, pages 243–257. Springer.

Ruoss, A., Baader, M., Balunović, M., and Vechev, M. (2020). Efficient certification of spatial robustness. *arXiv preprint arXiv:2009.09318*.

Sallami, M. M., Khedher, M. I., Trabelsi, A., Kerboua-Benlarbi, S., and Bettebghor, D. (2019). Safety and robustness of deep neural networks object recognition under generic attacks. In *International Conference on Neural Information Processing*, pages 274–286. Springer.

Singh, G., Gehr, T., Mirman, M., Püschel, M., and Vechev, M. (2018). Fast and effective robustness certification. In *Advances in Neural Information Processing Systems*, pages 10825–10836.

Singh, G., Gehr, T., Püschel, M., and Vechev, M. (2019a). An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41.

Singh, G., Gehr, T., Püschel, M., and Vechev, M. T. (2019b). Boosting robustness certification of neural networks. In *ICLR (Poster)*.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. (2013). Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Vassaux, B., Nguyen, P., Baudry, S., Bas, P., and Chassery, J.-M. (2002). Survey on attacks in image and video watermarking. In Tescher, A. G., editor, *Applications of Digital Image Processing XXV*, volume 4790, pages 169 – 179. International Society for Optics and Photonics, SPIE.

Xiao, C., Zhu, J.-Y., Li, B., He, W., Liu, M., and Song, D. (2018). Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*.