Abstract interpretation limitations for deep neural network robustness evaluation

Faouzi Adjed, Mallek Mziou Sallami

Abstract Interpretation Limitations for Deep Neural Network Robustness Evaluation

Conference Paper · June 2022

CITATIONS
0

READS
25

3 authors:

Faouzi Adjed
IRT System X
18 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Mallek Mziou Sallami
Atomic Energy and Alternative Energies Commission
58 PUBLICATIONS **118** CITATIONS
SEE PROFILE

Actes Taima
Ecole Nationale des Sciences de l'Informatique
38 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Abstract Interpretation Limitations for Deep Neural Network Robustness Evaluation

Faouzi ADJED[1] and Mallek MZIOU SALLAMI[2]

[1] IRT-SystemX,
8 Avenue de la Vauve, Palaiseau 91120, France
faouzi.adjed@irt-systemx.fr
[2] CEA, The French Alternative Energies and Atomic Energy Commission, France.
mallek.mziou@cea.fr

Résumé La vérification formelle des réseaux de neurones a été largement étudiée ces dernières années pour estimer la robustesse des réseaux de neurones face aux attaques.

Nous abordons le problème de spécifier des attaques génériques et composées sous forme de domaines abstraits pour mieux caractériser les entrées d'un réseau de neurones et pouvoir évaluer l'impact de leurs éventuelles perturbations. Nous considérons le cas réaliste avec des attaques géométriques en perspective appliquées à des images couleurs et des effets d'occultation sur les bords que nous corrigeons avec la méthode "inpainting". Nous étudions l'effet de l'inpainting sur l'éxactitude de l'évaluation de la robustesse et estimer la différence entre les attaques réélles et les attaques simulées.

Mots clés Rotation 3D, Interpretation Abstraite, Robustesse des réseaux de neurones profonds

Abstract The formal verification of neural networks has been widely studied in recent years to estimate the robustness of networks against attacks. We address the problem of specifying generic and compound attacks in the form of abstract domains to better characterize the inputs of a neural network and be able to evaluate the impact of its possible disturbances. We consider the realistic case with perspective geometric attacks applied to colour images and occlusion effects on the edges that we correct with inpainting. We study the effect of the digital inpainting on the robustness performance and evaluate the existing difference between real and simulated attacks.

Key words 3D rotations, Abstract Interpretation, DNN robustness.

1 Introduction

Robustness of deep neural networks (DNN) is one of the most difficult tasks in machine learning domain. In the literature, DNN were often considered as an opaque box and their instability against natural and synthetic disturbances called adversarial attacks has been proven [17],[12]. Thus, several recent work has focused on studying the robustness of deep neural network architectures in order to propose verification methods. In the literature, Ruan et al. [13] summarize an overview of the developed approaches for deep learning robustness, such as Reluplex [6], DeepSymbol [8] and abstract interpretation [5]. Among these methods, we focus on those based on abstract

interpretation theory ([1], [15], [16]). Indeed, Singh et al. [16] proposed a DNN verifier called DeepPoly to evaluate brightness and 2D rotation attacks on deep images classifiers. In [14] and [10], DeepPoly was extended to estimate the robustness of a DNN against 3D rotation and convolution on images. In a recent work, 2D closed contours data was considered [7]. All the proposed method which generalizes the notions of Upper Bound and Lower Bound and support the following attacks: 3D rotation, filtering and occlusion, was tested on simulated attacks and often on images with a single object on a black background such as the MNIST database. However, in the real case, other constraints can be added. Indeed, images are colored and the presence of several objects on a single image is inevitable. Perspective effects and stretching may cause also appearance or disappearance of regions. For these reasons, a more realistic case with compound attacks must be considered.
In this paper, we study how exact is the DNN robustness estimation using abstract interpretation. The proposed work is aiming towards the applicability study of this certification method in the real case and investigate the effect of the synthetic gap on robustness evaluation. As a use case, we study the effects of the difference between simulated 3D rotation and realistic 3D rotation on robustness value. In the last case, we use impainting method to fill the black pixels generated by the 3D rotation. It is important to highlight that the same effect is visible for 2D rotation.

The remaining of this paper is organized as follows. Section 2 is dedicated to state the fundamental concepts of the Abstract Interpretation for robustness evaluation, 3D rotation theory and the implemented approach for the inpainting. We demonstrate the different 3D rotation and inpainting attacks on colored images within a perception context. In section 3, we present the new algorithm to evaluate the robustness against each of these attacks. Our experimentation settings and results are given in Section 4. Finally, in Section 5, we draw our conclusions and we discuss some future perspectives.

2 Background

2.1 Abstract interpretation

Abstract Interpretation was developed by Cousot and Cousot [2] and adapted for logic functions for computer programs by the same authors [3]. Recently, the abstract interpretation was adapted for neural networks to create a new function which approximates and overestimates each step of a given DNN architectures such as ReLU and convolution functions [5,9]. The approximation of each step is called abstract transformer. Thus, for each function f in the DNN, an abstract transformer T_f which overestimates the behaviour of f is constructed. In what follows, we recall in a synthetic way some notions and notations as detailed by Gerh et al. [5].
Let $\bar{X}$ be a given input. The original inputs perturbed by ε are denoted by $R_{\bar{X},\varepsilon}$. Let C_L be the robustness condition that defines the output ensemble with the same label L, i.e the set of outputs y describing the same label L. We denote $\bar{Y}$ the set of each prediction for each element in $R_{\bar{X},\varepsilon}$.

$$C_L = \{\bar{y} \in \bar{Y} | \arg\max \bar{y}_i = L\} \tag{1}$$

The ($R_{\bar{X},\varepsilon}$,C_L) property is verified only if the outputs O_R of $R_{\bar{X},\epsilon}$ are included in C_L. However, in reality, we have no knowledge about O_R. The Abstract Interpretation is a proposed alternative to face this shortcoming. In fact, it allows to determine an abstract domain thought transformers and verifies the inclusion condition in new abstract domains α_R, which is an abstraction of $\bar{X}$. We denote the output abstract domain α_R^O. The ($R_{\bar{X},\varepsilon}$,C_L) property is checked if the outputs α_R^O of α_R (the abstraction of $R_{\bar{X},\varepsilon}$) are included in C_L. In other words, the verification of ε perturbation proves all perturbations smaller than ε.

2.2 3D rotation

The relationship between the object displacement and its projection on the image, can be explained referring to the pinhole camera model described in [19]. Using the Pinhole model, one can easily prove that there exist two elements $\pm k$ for every rotation r of the symetric space of the rotation group $\mathbf{SO}(3)$ [18] such that:

$$k = \pm \begin{pmatrix} \bar{a} & -\bar{b} \\ b & a \end{pmatrix}$$

with,

$$a = \pm \cos(\frac{\phi}{2}) e^{\frac{i(\psi_1+\psi_2)}{2}}, \quad b = \pm i \sin(\frac{\phi}{2}) e^{\frac{i(\psi_1-\psi_2)}{2}}$$

where ψ_1 and ψ_2 are the rotation angle in image plane, and ϕ is the rotation angle out of the image plane [14]. Consequently, the rotation of the point z is denoted by:

$$k.z = \frac{az+b}{-\bar{b}z+\bar{a}}$$

2.3 Inpainting

The MNIST database, which contains only binary images, is not affected by the created black parts in the image due to the rotation or translation. However, the colored images, such as CIFAR database, which is closer to the real world, is more affected by the black parts added by the rotation transformation, as illustrated in Fig. 1-b (Rgb image rotated with generated black regions). The obtained images after rotation becomes not smooth and the evaluation of the robustness of DNNs will may decrease by these black generated parts. To overcome this issue, an inpainting and completion based on biharmonic interpolation method [4] is implemented. The biharmonic interpolation is based on finding a function u describing the known parts in the image to fill missed information by solving the following equation:

$$\begin{aligned} \Delta^2 u &= 0 \quad \text{on } \Omega \setminus \Omega_K \\ \partial_n u &= 0 \quad \text{on } \partial\Omega \\ u &= f \quad \text{on } \Omega_k \end{aligned}$$

where, Δ denotes the Laplacian operator, Ω is the rectangular image domain, Ω_k is the known part in the image. ∂_n denotes the derivatives normal to the Boundary, and $f : \Omega \to \mathbf{R}$ is smooth function on a Bounded domain Ω with regular Boundary $\partial\Omega$.

To define the missed regions in the image generated by the rotation, a white mask (matrix of 32×32, where each value is equal to 255) is rotated by the same angles. Then, a binary mask is created separating a know and unknown region after rotation of the original image. Fig1-c illustrates the three steps of inpainting of missed regions, which are, rotation, mask generation and filling using the biharmonic interpolation.

3 Proposed approach

The workflow of the approach is divided into three main steps which are : (i) 3D rotation on the three channels (RGB) of the colored image. (ii) Inpainting of missed regions in the rotated image. (iii) Computation of Lower and Upper Bounds of each rotated image with inpainting using

(a) (b) (c)

Figure 1 – Example of 3D rotation of images from CIFAR database: (a): original image, (b) image rotated using 3D rotation with dark regions generated by the rotation, (c) the final result with rotation and inpainting.

abstract interpretation. Algorithms 1, 2 and 3 details the main stages of 3D rotation, implemented in [14], bi-harmonic inpainting and abstract interpretation approaches, respectively.

Algorithm 1 shows the main steps of the rotation procedure for RGB image using the method presented in section 2.2, followed by the Algorithm 2 for filling the missed region due to the rotation. Thus, to estimate the robustness of a DNN against a 3D rotation R $(\phi, \psi_1, \psi_2) \in [-\pi, \pi]$, one have to estimate the abstract domain expressed by Upper and Lower Bound and denoted by T^I_{LB} and T^I_{UB} as presented in Algorithm 3. For this purpose, we begin by splitting the three intervals to batches as described in Algorithm 3. Moreover, for each batch, we call Algorithm 1 coupled with Algorithm 2 (if we apply impainting) to compute the Lower and Upper Bound T_{LB} and T_{UB} for a given 3D rotation. By enumerating all possible integer values of c_{low}, c_{high}, v_{low} and v_{high}, we can identify a polygon where the pixel (x', y') (transformation of the pixel (x, y)) is located. If the pixel is in the center of the image, we do not perform special processing. Else, if the pixel is in the edges of the image, we apply the interpolation to know the three RBG values. To verify that for any image $I \in X$, for any angle $\phi \in [\phi_{min}, \phi_{max}]$ and any angle $\psi_1 \in [\psi^1_{min}, \psi^1_{max}]$ the neural network N classifies $I_{\epsilon=(\phi,\psi_1)}$ to the class of I, we cannot simply enumerate all possible rotations as done for simpler rotation algorithms and concrete images. [11]

4 Results

Our experimentation is achieved on CIFAR 10 dataset which consists of 32×32 color images categorized into ten classes. The proposed approach is implemented using DeepPoly [1] method developed by [16], where six architectures are implemented and tested for different angles. Table 1 summarizes the type, number of layers and the activation function of each architecture.

1. https://github.com/eth-sri/eran

Algorithm 1 Rotate Image Im by 3D rotation

procedure PROCEDURE IMAGE_ROTATION
Input: $Im \in [0,255]^{m\times n\times 3}$; $\phi, \psi_1, \psi_2 \in [-\pi, \pi]$; T, T_{LB}, $T_{UB} \in [0,255]^{m\times n\times 3}$
$(a,b) = (\cos(\frac{\phi}{2})e^{\frac{i(\psi_1+\psi_2)}{2}}, i\sin(\frac{\phi}{2})e^{\frac{i(\psi_1-\psi_2)}{2}})$
for $d \in \{1,2,3\}$ **do**
  I = Im[:,:,d]
  for $c \in \{1,\dots,m\}; v \in \{1,\dots,n\}$ **do**
    $(x,y,z) = (c - \frac{m+1}{2}, \frac{n+1}{2} - v, x + iy)$
    $z = \frac{(az+b)}{-\bar{b}z+\bar{a}}$
    $(y', x') = (Im(z), Re(z))$
    $(c'_{low}, c'_{high}) \leftarrow (\max(1, \frac{m+1}{2} - y'), \min(m, \frac{m+1}{2} - y')$
    $(v'_{low}, v'_{high}) \leftarrow (\max(1, x' + \frac{n+1}{2}), \min(n, x' + \frac{n+1}{2})$
    $R^{Low}_{c,v} \leftarrow \min(255, \min_{c'\in[c'_{low},c'_{high}],v'\in[v'_{low},v'_{high}]} I[c', v']$
    $R^{Hight}_{c,v} \leftarrow \max(0, \max_{c'\in[c'_{low},c'_{high}],v'\in[v'_{low},v'_{high}]} I[c', v']$
    $t \leftarrow \sum_{c'=c'_{low},v'=v'_{low}}^{c'=c'_{high},v'=v'_{high}} \max(0, 1 - \sqrt{(v'-x')^2 + (c'-y')^2})$
    $t' \leftarrow \sum_{c'=c'_{low},v'=v'_{low}}^{c'=c'_{high},v'=v'_{high}} (\max(0, 1 - \sqrt{(v'-x')^2 + (c'-y')^2}) \times I[c', v'])$
    if $t \neq 0$ **then**
      $T[c,v] \leftarrow \frac{1}{t} \times t'$
      $T_{LB}[c,v] \leftarrow min(T_{LB}[c,v], R^{Low}_{c,v})$
      $T_{UB}[c,v] \leftarrow max(T_{UB}[c,v], R^{Hight}_{c,v})$
    else
      $T[c,v], T_{LB}[c,v], T_{UB}[c,v] \leftarrow 0$
Return T, T_{LB}, T_{UB}

Algorithm 2 Biharmonic Inpainting Algorithm

procedure BIHARMONIC_INPAINTING
Input: rot_image $\in [0,255]^{m\times n\times 3}$, mask $\in [0,1]^{m\times n}$
for $i \in \{1,...,m\}$; $j \in \{1,..,n\}$ **do**
  pixel_mask = mask(i, j)
  if pixel_mask == 0 **then**
    inpaint_image(i,j) $\leftarrow$ biharmonic_interpolation(i,j)
  else
    inpaint_image(i,j) $\rightarrow$ rot_image(i,j)
Return inpaint_image

Algorithm 3 Lower and Upper Bound for 3D Rotation (on a rotation interval)

procedure PROCEDURE ROTATION_LOWER_UPPER_BOUND
Input: $I, T^I_{LB}, T^I_{UB} \in [0,255]^{m\times n\times 3}; bs_\phi, bs_{\psi_1}, bs_{\psi_2} \in \mathbb{N}$
  $\phi_{min}, \phi_{max} \psi^1_{min}, \psi^1_{max}, \psi^2_{mmin}, \psi^2_{max}, \in [-\pi, \pi]$
$(step_\phi,\ step_{\psi 1},\ step_{\psi 2}) = (\frac{|\phi_{max}-\phi_{min}|}{bs_\phi}, \frac{|\psi^1_{max}-\psi^1_{min}|}{bs_{\psi_1}}, \frac{|\psi^2_{max}-\psi^2_{min}|}{bs_{\psi_2}})$
Compute lists $\phi_{all}, \psi^1_{all}, \psi^2_{all}$ of all values using their respective steps
for $(\phi_0, \psi^1_0, \psi^2_0) \in (\phi_{all}, \psi^1_{all}, \psi^2_{all})$ **do**
  $(T, T_{LB}, T_{UB}) \leftarrow IMAGE\_ROTATION(I, \phi_0, \psi^1_0, \psi^2_0)$
  $T^I_{LB} = \min(T^I_{LB}, T_{LB})$
  $T^I_{UB} = \max(T^I_{UB}, T_{UB})$
Return T^I_{LB}, T^I_{UB}

Table 1 – Implemented neural network architectures for robustness evaluation

Model	Type	Layers	Activation
4×100	fully connected	4	ReLu
6×100	fully connected	6	ReLu
9×200	fully connected	9	ReLu
ConvSmall	convolutional	3	ReLu
ConvMaxpool	convolutional	9	ReLu

For each architecture, a metric performance of robustness (Rts) is computed following the equation given below:

$$Rts = \frac{\#VI}{\#CI}$$

where $\#VI$ and $\#CI$ denotes the number of verified images and well classified image respectively.

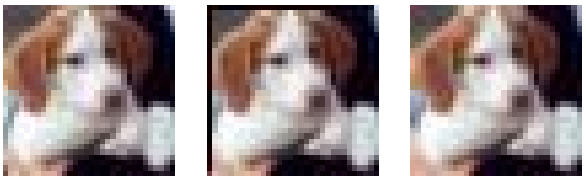
The parameters of the the three angles are given below:

— Plane rotation (ψ_1 and ψ_2):
  — the angles ψ_1 and ψ_2 are equal.
  — ψ_1, ψ_2 include to $[0, 30°]$
  — the interval of each angle is split into 30 batches (1 batch = $1°$)
— Spatial rotation (the third dimension ϕ)
  — ψ_1, ψ_2 include to $[0, 10°]$
  — the interval is split into 50 batches (1 batch = $0.2°$

The results of our method encompass the effect of two controlled attacks, which are the plane and the spatial rotations, and one unwanted and uncontrollable attack generated by the inpaiting and the filling of missed regions. Indeed, as illustrated in figure 2, where only the top line and the left column are missed (2-b) and filled by the biharmonic interpolation (2-c), the image is missclassified by 4-fully connected layers network (the first model in the table 1). To isolate the effect of the inpainting attack, we evaluated the attack between the original images and the oriented images with null angles (occlusion of the top line and the left column then inpainting). Figure 3 shows that all models are sensitive to inpaiting step. We can see that the most sensitive and unstable model is the ConvSmall with 41% of robustness followed by 4×100 model with 88%. The 6×600, 9×200 and ConvMaxpool models exceed the 90% of robustness with 98%, 92% and 97% respectively. According to the discussion above, the inpaiting step could affect randomly the robustness of the model for large angles.

5 Conclusion

The estimation of DNN robustness is very important step towards the a validation of a perception system. The abstract interpretation theory and other robustness evaluation approaches could be helpful. However, the gap between the simulated perturbation and the real perturbation impact overly the robustness value. By applying the inpainting, it is impossible to evaluate the effect of the perturbation and the effect of the restoration due to the inpainting step. Therefore, even if the attacks are realistic and physically interpretable, the robustness evaluation of the model cannot be proven only for the perturbation. The obtained results show the additional generated synthetic perturbation such as the inpaiting in the current case.



(a) original image

(b) oriented image with zero degree for all angles (ϕ and ψ_1, ψ_2)

(c) oriented image with inpainting. Image classified as *horse*-class instead of *dog*-class

Figure 2 – Effect of the occlusion due to 3D rotation: (a) classified as *dog* and (c) classified as *horse*

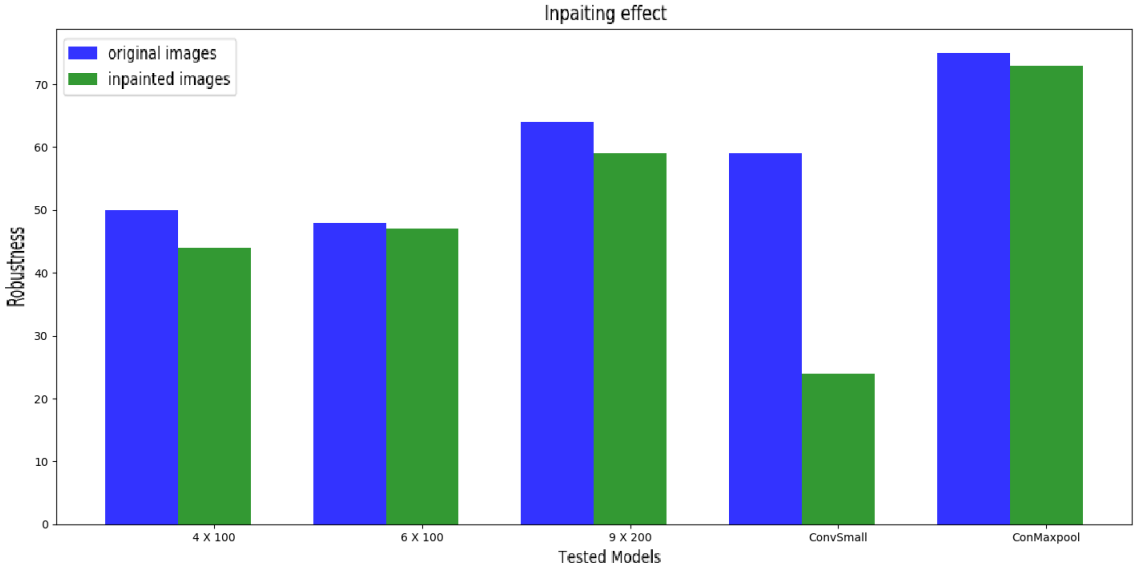


Figure 3 – Inpainting effect and model comparison

Aknowledgement

This research work has been carried out in the framework of IRT SystemX, Paris-Saclay, France, and therefore granted with public funds within the scope of the French Program Investissements d'Avenir. This work is a part of the project EPI project (EPI for "Evaluation des Performances de l'Intelligence artificielle").

References

1. Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. In *Advances in Neural Information Processing Systems*, pages 15287–15297, 2019.
2. Patrick Cousot and Radhia Cousot. Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *Proceedings of the 4th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 238–252. ACM, 1977.
3. Patrick Cousot and Radhia Cousot. Abstract interpretation and application to logic programs. *The Journal of Logic Programming*, 13(2-3):103–179, 1992.
4. SB Damelin and NS Hoang. On surface completion and image inpainting by biharmonic functions: Numerical aspects. *International Journal of Mathematics and Mathematical Sciences*, 2018, 2018.
5. Timon Gehr, Matthew Mirman, Dana Drachsler-Cohen, Petar Tsankov, Swarat Chaudhuri, and Martin Vechev. Ai2: Safety and robustness certification of neural networks with abstract interpretation. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2018.
6. Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International conference on computer aided verification*, pages 97–117. Springer, 2017.
7. R. Khalsi, M. Sallami, I. Smati, and F. Ghorbel. Contourverifier: A novel system for the robustness evaluation of deep contour classifiers. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence*, volume 3, pages 1003–1010, 2022.
8. Jianlin Li, Jiangchao Liu, Pengfei Yang, Liqian Chen, Xiaowei Huang, and Lijun Zhang. Analyzing deep neural networks with symbolic propagation: Towards higher precision and faster verification. In *International Static Analysis Symposium*, pages 296–319. Springer, 2019.
9. Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3575–3583, 2018.
10. M. Mziou-Sallami and F. Adjed. Towards a certification of deep image classifiers against convolutional attacks. In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence*, volume 2, pages 419–428, 2022.
11. Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Deepxplore: Automated whitebox testing of deep learning systems. In *proceedings of the 26th Symposium on Operating Systems Principles*, pages 1–18. ACM, 2017.
12. Kexin Pei, Yinzhi Cao, Junfeng Yang, and Suman Jana. Towards practical verification of machine learning: The case of computer vision systems. *arXiv preprint arXiv:1712.01785*, 2017.
13. Wenjie Ruan, Xinping Yi, and Xiaowei Huang. Adversarial robustness of deep learning: Theory, algorithms, and applications. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pages 4866–4869, 2021.
14. Mallek Mziou Sallami, Mohamed Ibn Khedher, Asma Trabelsi, Samy Kerboua-Benlarbi, and Dimitri Bettebghor. Safety and robustness of deep neural networks object recognition under generic attacks. In *International Conference on Neural Information Processing*, pages 274–286. Springer, 2019.

15. Gagandeep Singh, Rupanshu Ganvir, Markus Püschel, and Martin Vechev. Beyond the single neuron convex barrier for neural network certification. In *Advances in Neural Information Processing Systems*, pages 15072–15083, 2019.
16. Gagandeep Singh, Timon Gehr, Markus Püschel, and Martin Vechev. An abstract domain for certifying neural networks. *Proceedings of the ACM on Programming Languages*, 3(POPL):41, 2019.
17. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
18. Jacek Turski. Harmonic analysis on sl (2, c) and projectively adapted pattern representation) and projectively adapted pattern representation. *Journal of Fourier Analysis and Applications*, 4(1):67–91, 1998.
19. Jacek Turski. Projective fourier analysis for patterns. *Pattern Recognition*, 33(12):2033–2043, 2000.