Proceedings of the 20th International Conference on Principles of Knowledge Representation and Reasoning
Main Track

Verification of Semantic Key Point Detection for Aircraft Pose Estimation

Panagiotis Kouvaros[1,3] , **Francesco Leofante**[1] , **Blake Edwards**[2] ,
Calvin Chung[2] , **Dragos Margineantu**[2] , **Alessio Lomuscio**[1,3]
[1]Imperial College London, London, UK
[2]Boeing Research & Technology, Seattle, USA
[3]Safe Intelligence, London, UK
{p.kouvaros, f.leofante, a.lomuscio}@imperial.ac.uk,
{blake.t.edwards, chunkit.chung, dragos.d.margineantu}@boeing.com

Abstract

We analyse Semantic Segmentation Neural Networks running on an autonomous aircraft to estimate its pose during landing. We show that automated reasoning techniques from neural network verification can be used to analyse the conditions under which the networks can operate safely, thus providing enhanced assurance guarantees on the behaviour of the overall pose estimation systems.

1 Introduction

Over the years the area of knowledge representation and reasoning has achieved a stream of deep theoretical results relevant to Machine Learning (ML), including, but not limited to, description logics and ontologies (de Sousa Ribeiro and Leite 2021), argumentation (Potyka 2021), belief revision (Coste-Marquis and Marquis 2021) and beyond. In the context of automated reasoning, the formal verification of ML-based systems has emerged as an area of interest. Verification can help assess the safety of these systems before deployment, thereby mitigating the possibilities of AI causing harm. Considerable work has recently been devoted to the problem of formally verifying that an AI system realised by neural networks meets a given specification, e.g. robustness to input perturbations and adversarial attacks (Bak et al. 2020; Ehlers 2017; Balunovic et al. 2019; Botoeva et al. 2020; Bunel et al. 2020; Cheng, Nuhrenberg, and Ruess 2017; Dvijotham et al. 2018; Henriksen and Lomuscio 2020; Henriksen and Lomuscio 2021; Henriksen et al. 2021; Katz et al. 2017; Katz et al. 2019; Kouvaros and Lomuscio 2021; Guidotti, Pulina, and Tacchella 2021; Singh et al. 2019; Tjeng, Xiao, and Tedrake 2019; Tjandraatmadja et al. 2020; Tran et al. 2020; Wang et al. 2018a; Wang et al. 2021). While much progress has been made over the past five years, two key problems remain. Firstly, very few applications of industrial relevance have been tackled by the resulting methods and tools; secondly, large neural networks, such as those used in applications, are seldom formally analysed. In this paper we make a contribution towards both challenges. Firstly, we extend present methods and tools to deal with a complex neural system of millions of tunable parameters; secondly, we tailor verification methods to a novel application of industrial relevance, i.e. semantic key point detection for aircraft pose estimation in the context of an airplane autoland system, as described in the following.

Accurate ego-pose estimation is a central building block for the successful development of fully autonomous aviation systems. Indeed, for an autonomous system to operate safely, a sense of local awareness is required to effectively navigate and adapt to the surrounding environment. Various solutions for pose estimation have been developed based on, e.g. GPS technology, Inertial Navigation Systems or laser sensors (Cadena et al. 2016). Among these, vision-based approaches stand out for being particularly accurate and versatile, but also affordable. A prominent example is the Perspective-n-Point method (Fischler and Bolles 1981), which estimates the pose of a calibrated camera (tightly attached to the autonomous system) leveraging point correspondences between 3D coordinates of real world points and their 2D projections on an image.

A key requirement for Perspective-n-Point (PnP) methods to produce satisfactory results is the precise identification of 3D-to-2D point correspondences. While 3D points typically correspond to objects of known coordinates, the same does not hold for their 2D projections, which need to be identified from the image. This step is of crucial importance as PnP is sensitive to spurious correspondences, which may result in poor pose estimates.

In this paper we consider an ML-powered PnP system where U-Nets (Ronneberger, Fischer, and Brox 2015), a special class of Semantic Segmentation Networks (SSNNs), are used to identify the 2D coordinates of key 3D points within an image. The prototypical system, developed by Boeing, is used to estimate the 6 Degrees of Freedom (DOF) pose of an autonomous aircraft and plays an important role during landing.

Clearly, the safety-critical nature of the application requires that U-Nets satisfy stringent safety requirements. However, neural networks are known to be susceptible to adversarial attacks (Goodfellow, Shlens, and Szegedy 2014; Szegedy et al. 2014) which may disrupt operations if undetected (Kouvaros et al. 2021). This may have catastrophic consequences in aviation, where human lives are at stake.

In this paper we consider the verification of an industrial-scale U-Net for PnP designed and trained by Boeing. The technical contributions of the paper are as follows.

- Firstly, and differently from previous work, U-Nets operate on large input/output spaces, which dramatically increases the computational requirements of verification. Additionally, U-Nets feature complex layers, presently not supported by other techniques. We overcome these by extending VENUS (Kouvaros and Lomuscio 2021), a complete verifier based on constrained optimisation and symbolic interval arithmetic (Zhang et al. 2018), to fully support U-Nets and enable their verification.
- Secondly, previous verification work has mostly focused on academic datasets and classification challenges. Instead, we here study a concrete industrial problem for autonomous systems, firstly by considering standard local robustness specifications (Katz et al. 2017), and then extend it to more sophisticated photometric adjustments and a novel finer-grained analysis in the pixel space.
- Lastly, beside the technical contributions above in automated reasoning, the work enabled the discovery of areas of fragilities in the U-Net detector, indicating areas of robustness and accuracy, but also concrete fragilities so that they can be further mitigated before any deployment.

2 Background

We formally define the verification problem for neural networks and outline VENUS, a state-of-the-art neural network verifier that we use for resolving the verification queries considered in Section 4.

U-Net. A *neural network* $\mathbf{f}$ is a directed rooted tree with exactly one leaf node. Each i-th vertex (also called a *layer*) of $\mathbf{f}$, denoted $\mathbf{f}^{(i)}$, is a vector-valued function on the concatenation of the outputs of its parent nodes. That is, $\mathbf{f}^{(i)} : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{n_j}$, where $m_i \triangleq \sum_{k \in \mathcal{P}(i)} n_k$, $n_j > 0$, and $\mathcal{P}(i)$ is the set of parent layers of $\mathbf{f}^{(i)}$. Thus, a neural network is equivalent to a function $\mathbf{f} \colon \mathbb{R}^{m_0} \rightarrow \mathbb{R}^{n_K}$, where $\mathbf{f}^{(0)}$, $\mathbf{f}^{(K)}$ are the root and leaf layers, m_0 and n_K are the input and output dimensions of the root and leaf layers, and $\mathbf{f}(\mathbf{x}) = \mathbf{f}^{(K)}(\bigoplus_{i \in \mathcal{P}(K)} \mathbf{f}^{(i)}(\ldots \mathbf{f}^{(0)}(\mathbf{x}) \ldots))$, where $\bigoplus$ denotes vector concatenation.

In this paper we focus on U-Nets (Ronneberger, Fischer, and Brox 2015), a special type of neural networks where every layer implements one of the following operations: (i) a convolution; (ii) a transposed convolution; (iii) concatenation; (iv) the ReLU activation function; (v) batch normalisation; (vi) max-pooling. We refer to (Goodfellow, Bengio, and Courville 2016) for the formal definitions of these operations. We also restrict our attention to *semantic segmentation* tasks, where each element of the output of $\mathbf{f}$ expresses the likelihood of an input pixel belonging to a certain class.

Verification problem. Given a neural network $\mathbf{f}$, a set $\mathcal{X} \subseteq \mathbb{R}^{m_0}$ of possible inputs for the network, and a linear function $h \colon \mathbb{R}^{n_K} \rightarrow \mathbb{R}$, the verification problem is to determine whether

$$\forall \mathbf{x} \in \mathcal{X} \colon h(\mathbf{f}(\mathbf{x})) > 0. \tag{1}$$

The *local adversarial robustness* instantiation of the problem is widely considered in neural network safety analyses. It aims at ascertaining whether a network is vulnerable to *adversarial attacks*, imperceptible perturbations to the input that cause the network to miss-classify. More formally, given an input $\mathbf{x}_0$, the local adversarial robustness problem sets $\mathcal{X}$ to be a set of "similar" inputs to $\mathbf{x}_0$ and defines h as $h \triangleq \mathbf{f}(\mathbf{x})_i - \mathbf{f}(\mathbf{x})_j$ for $\forall \mathbf{x} \in \mathcal{X}$, where i and j are the true and adversarial classes for $\mathbf{x}_0$ respectively. Establishing that condition 1 holds in this context is a proof that the network is robust for the set of perturbations encoded in $\mathcal{X}$. The latter is typically defined either as an ℓ_∞ ball around the input, i.e. $\mathcal{X} = \{\mathbf{x} \colon \|\mathbf{x} - \mathbf{x}_0\|_\infty \leq \epsilon\}$, where ϵ is the radius of the ball, or as a set of brightness/contrast adjustments, i.e. $\mathcal{X} = \{\alpha \,\square\, \mathbf{x}_0 \colon l \leq \alpha \leq u\}$, where l and u are the lower and upper bounds of the adjustment and $\square = +$ (brightness) or $\square = \cdot$ (contrast). We refer to (Kouvaros and Lomuscio 2018) for a more detailed discussion.

VENUS. VENUS is a neural network verifier that implements a branch-and-bound procedure to solve the verification problem (Kouvaros and Lomuscio 2021). At each branch of the process two key steps are performed. First, the verification problem is divided into two sub-problems by splitting the operational semantics of a ReLU unit into its two linear segments. Second, a lower bound of $h(\mathbf{f}(\cdot))$ is computed for each branch using symbolic interval arithmetic methods (Zhang et al. 2018). The steps are repeated until a user-defined threshold on the branch-and-bound tree is reached. The sub-problems that have not been resolved up to this point are encoded as Mixed Integer Linear Programs and solved using the MILP-solver GUROBI (Gu, Rothberg, and Bixby 2020). We refer to (Botoeva et al. 2020; Kouvaros and Lomuscio 2021) a for a more detailed exposition on VENUS.

3 Estimating Aircraft Pose from Images

In this section, we introduce the aircraft pose estimation problem at the core of this work and describe the AI-based pipeline developed by Boeing.

Overview. We are interested in estimating the 6DOF pose of an autonomous aircraft during landing. We use PnP to estimate the pose of a calibrated camera tightly mounted on the aircraft by leveraging 3D-to-2D point correspondences between 3D coordinates of real world points and their 2D projections on an image captured from the aircraft. For this case study, we use 16 key points whose 3D coordinates are known beforehand in the world reference frame: the corners of the runway (4), the threshold marking (4) and touchdown markings (8) on the runway (Figure 1 (top)). As for the 2D images, we start from high-resolution images (4000×3096 pixels) and apply standard pre-processing techniques (downsampling to 112×112 pixels and gray scale conversion) before feeding them to a U-Net trained to perform semantic segmentation. The network has 75 layers, around 2 million trainable parameters and around 1.6 million non-linear elements. It produces 16 semantic segmentation heatmaps corresponding to the 16 ground truth points. Pixel values in each heatmap indicate the likelihood of each pixel being a key point; thus we extract the 2D coordinates of key points by identifying the pixel that has the highest value in each heatmap (Figure 1 (top)). Once the 16 pairs of points have

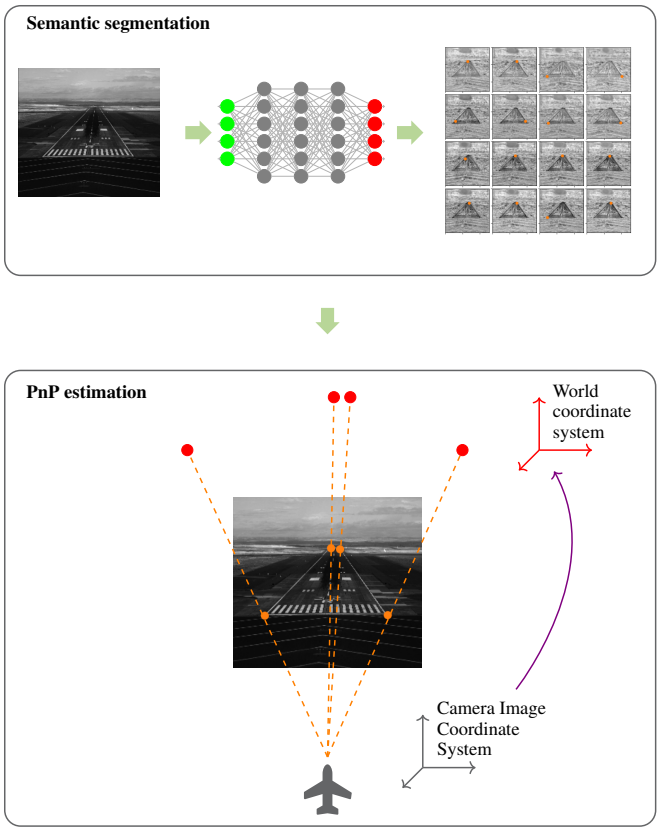

Figure 1: Pictorial representation of the 3D-to-2D matching problem studied in this paper. Given a 2D image of the runway, a U-Net is used to detect 16 key points (orange dots) whose ground truth 3D coordinates are known (top). These points are then fed to the PnP pipeline, which uses them to estimate the 6DOF of the aircraft (bottom).

been created, we use off-the-shelf PnP algorithms to draw correspondences between 2D and 3D coordinates and derive the 6DOF pose of the aircraft (Figure 1 (bottom)).

Safety concerns. The above pipeline heavily relies on the U-Net to identify meaningful key points in the image. Despite being trained to high-accuracy, U-Nets are susceptible to adversarial attacks that may result in erroneous estimates of the 2D coordinates of key points, ultimately jeopardising the pose estimate produced by PnP. In this work we argue that automated reasoning techniques from formal verification can be used to quantify the error that U-Nets may inject and provide formal guarantees on their behaviour.

Challenges. The analysis of U-Nets deployed by Boeing poses a number of challenges to state-of-the-art neural network verifiers. To begin with, U-Net's tree-like architectures allows for *skip connections* between layers, which deviate from more standard linear architectures that are most commonly studied in the VNN literature (Bak, Liu, and Johnson 2021). This, together with operations such as transposed convolutions and concatenations, restricts considerably the pool of verification tools that are able to handle U-Nets. Finally, the size of the U-Net, together with the dimensionality of its input/output spaces, poses the biggest challenge to the application of formal verification techniques whose computational cost is known to increase with these two parameters (Katz et al. 2017). Indeed, the U-Nets we consider here are at least an order of magnitude bigger than previously considered. For reference, (Tran et al. 2021) consider SSNNs with 22 layers and operating on inputs/outputs spaces of dimension up to 64×84. In contrast, U-Nets considered here have 75 layers and operate on an input space of dimension 112×112 and output space of dimension $20 \times 112 \times 112$.

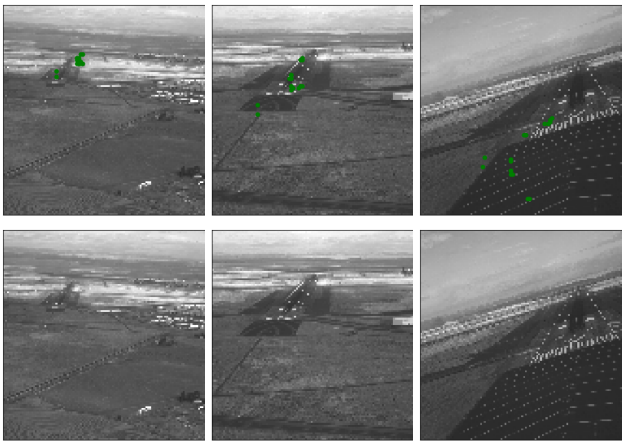

Figure 2: **First row**: Input images and key point detection (green dots). **Second row** (from left to right): white noise, brightness and contrast counterexamples.

4 Experimental Evaluation

We conduct a formal analysis of the U-Net presented in the previous section to estimate its level of fragility against adversarial perturbations.

Specifications. The operational domain of the U-Net inevitably includes inputs reflecting the variability of the environment, as often expressed by random noise and different lighting conditions generating contrast and brightness transformations. To define the robustness of the U-Net against these perturbations, denote the U-Net and its i-th key point prediction by $\mathbf{f}$ and $key_i(\mathbf{f}(\cdot))$, assume an input image $\mathbf{x}_0$, a perturbation radius $\epsilon > 0$, and ranges $[l_b, u_b]$, $[l_c, u_c]$ for the levels of brightness and contrast adjustments.

(i) **White-noise robustness.** For all $\mathbf{x}$ such that $\|\mathbf{x} - \mathbf{x}_0\|_\infty \leq \epsilon$ and for all $i \in [1, 16]$, we have that $key_i(\mathbf{f}(\mathbf{x})) = key_i(\mathbf{f}(\mathbf{x}_0))$.

(ii) **Brightness robustness.** For all $\alpha \in [l_b, u_b]$ and for all $i \in [1, 16]$, we have that $key_i(\mathbf{f}(\mathbf{x}_0 + \alpha)) = key_i(\mathbf{f}(\mathbf{x}_0))$.

(iii) **Contrast robustness.** For all $\alpha \in [l_c, u_c]$ and for all $i \in [1, 16]$, we have that $key_i(\mathbf{f}(\alpha \cdot \mathbf{x}_0)) = key_i(\mathbf{f}(\mathbf{x}_0))$.

Informally, the specifications above require that the coordinates of each key point as predicted by $\mathbf{f}$ on $\mathbf{x}_0$ are not affected by variability in the input.

Our solution. As mentioned in the previous section, verifying U-Nets against the above properties poses a number of

Perturbation	ϵ/α	#Robust	#Non-robust	#Timeouts
Brightness	$[-5e-5, 5e-5]$	20	0	0
	$[-5e-4, 5e-4]$	18	2	0
	$[-5e-2, 5e-2]$	16	4	0
Contrast	$[-5e-4, 5e-4]$	20	0	0
	$[-5e-3, 5e-3]$	20	0	0
	$[-5e-2, 5e-2]$	18	2	0
White noise	1e-13	20	0	0
	1e-8	20	0	0
	1e-3	0	8	12

Table 1: Robustness results for brightness, contrast and white noise perturbations.

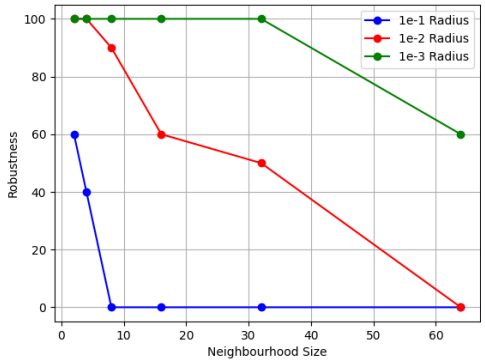

Figure 3: Robustness results for localised white noise perturbations for neighbourhoods of different sizes.

challenges to modern verification tools. Several extensions to the neural network verifier VENUS had to be implemented to overcome them. In particular, since VENUS had only support for fully-connected, feed-forward architectures, the extension of its different components (including bound propagation and MILP-encoding methods) to arbitrary computational graphs required a complete overhaul of the tool. Additionally, the size of the U-Net presently considered posed challenges in the efficient execution of the bound propagation method that VENUS implements (Singh et al. 2019). The method computes bounds for each of the units in the network which not only help to strengthen the MILP encoding of the verification problem but can also at times be used to prove safety. To improve the efficiency of the procedure we implemented looser and faster propagation methods, including interval bound propagation (Wang et al. 2018c) and symbolic bound propagation (Wang et al. 2018b), which we synthesised in a procedure that considers first the application of the faster methods. These are used to filter out ReLU units that are stably operating in a linear manner when applying the preciser methods. Effectively, this reduces the overall floating-point operations required (Henriksen and Lomuscio 2021), thus enabling the verification of industrial scale U-Nets as we demonstrate in the following.

Experiments. We used a machine with an Intel Core i9 10920X 3.5 GHz 12-core CPU, 128 GB RAM, equipped with a GeForce RTX 2080 graphics card, and running Fedora 35 with Linux kernel 5.19. We randomly selected 20 images from the data set and used the extended version of VENUS to solve the robustness verification problems (i) - (iii) with a timeout limit of three hours. We instantiated the parameters of the white-noise, contrast and brightness perturbations to ranges that were found to contain the transition points from proving robustness to identifying fragilities.

Table 1 reports the results. We observe that VENUS established that the U-Net is robust to minor brightness changes ($\alpha \in [-5e-5, 5e-5]$) but found vulnerabilities for larger changes ($\alpha \in [-5e-2, 5e-2]$. Similar observations can be made with respect to contrast changes. In comparison to brightness however, the model was found to be robust to contrast for larger ranges for α than to contrast. As for white noise, the model was proven to be robust for very small noise patterns ($\epsilon = 1e-8$) but fragile for perturbations of $\epsilon \geq 1e-3$. We also considered more localised perturbations around the key points, corresponding to more realistic scenarios where only parts of the 2D scenes are not clearly visible. We thus constrained the robustness verification problems (i) - (iii) to only consider neighbourhoods around the key points (where only the values of the pixels in those regions are altered as opposed to the values of all the pixels). Verification results show increased robustness for the model. For instance, Figure 3 shows that the U-net is robust for a 100% of the inputs for rectangular neighbourhoods of size 2 and perturbation radius of 0.01, but is not robust for none of the inputs for neighbourhoods of size 64.

The existence of counter-examples produced by VENUS identifies concrete concerns regarding the deployment of the U-Net. Our experiments show that the U-Net is likely to introduce errors in the PnP estimation during deployment, given that counter-examples observed are often perceptually indistinguishable from the original input (see Figure 2).

5 Conclusions

We considered the problem of verifying Semantic Segmentation Neural Networks used by Boeing to estimate the 6DOF pose of an autonomous aircraft during landing. The VENUS verification toolkit was extended to handle the increased architectural complexity of U-Nets and was successfully deployed to generate proofs of safety or counter-examples to show when safety could not be guaranteed. While scalability remains the main challenge, in this paper we have shown for the first time that U-Nets of up to two million parameters can be handled by modern verifiers. To the best of our knowledge, this is the first study to report such results, demonstrating that automated reasoning techniques can play a crucial role in building safe and trustworthy AI systems. Despite the positive results reported in this paper, challenges remain for the wider application of KRR techniques to industry-scale neural networks. However, we highlight that should scalability become a barrier to the safe deployment of neural network-based systems, automated reasoning and verification could be paired with more lightweight runtime assurance methods during execution. Initial proposals have been made within the formal verification community (Cheng, Nührenberg, and Yasuoka 2019; Henzinger, Lukina, and Schilling 2020); we plan to explore the complementarities of these approaches in future work.

Acknowledgements

Work partially supported by the DARPA Assured Autonomy programme (FA8750-18-C-0095), the UK Royal Academy of Engineering (CiET17/18-26) and an Imperial College Research Fellowship awarded to Leofante. The views, opinions and/or findings expressed are those of the author and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government. Approved for Public Release, Distribution Unlimited.

References

Bak, S.; Tran, H.; Hobbs, K.; and Johnson, T. 2020. Improved geometric path enumeration for verifying ReLU neural networks. In *Proceedings of the 32nd International Conference on Computer Aided Verification (CAV20)*, volume 12224 of *LNCS*, 66–96. Springer.

Bak, S.; Liu, C.; and Johnson, T. 2021. The second international verification of neural networks competition (vnn-comp 2021): Summary and results. *arXiv preprint arXiv:2103.06624*.

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 (NeurIPS19)*. Curran Associates, Inc. 15313–15323.

Botoeva, E.; Kouvaros, P.; Kronqvist, J.; Lomuscio, A.; and Misener, R. 2020. Efficient verification of neural networks via dependency analysis. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence (AAAI20)*, 3291–3299. AAAI Press.

Bunel, R.; Lu, J.; Turkaslan, I.; Kohli, P.; Torr, P.; and Mudigonda, P. 2020. Branch and bound for piecewise linear neural network verification. *Journal of Machine Learning Research* 21(42):1–39.

Cadena, C.; Carlone, L.; Carrillo, H.; Latif, Y.; Scaramuzza, D.; Neira, J.; Reid, I.; and Leonard, J. 2016. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Trans. Robotics* 32(6):1309–1332.

Cheng, C.; Nuhrenberg, G.; and Ruess, H. 2017. Verification of binarized neural networks. *arXiv preprint arXiv:1710.03107*.

Cheng, C.; Nührenberg, G.; and Yasuoka, H. 2019. Runtime monitoring neuron activation patterns. In *Design, Automation & Test in Europe Conference & Exhibition (DATE19)*, 300–303. IEEE.

Coste-Marquis, S., and Marquis, P. 2021. On belief change for multi-label classifier encodings. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 1829–1836. ijcai.org.

de Sousa Ribeiro, M., and Leite, J. 2021. Aligning artificial neural networks and ontologies towards explainable AI. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI21)*, 4932–4940. AAAI Press.

Dvijotham, K.; Stanforth, R.; Gowal, S.; Mann, T.; and Kohli, P. 2018. A dual approach to scalable verification of deep networks. *arXiv preprint arXiv:1803.06567*.

Ehlers, R. 2017. Formal verification of piece-wise linear feed-forward neural networks. In *Proceedings of the 15th International Symposium on Automated Technology for Verification and Analysis (ATVA17)*, volume 10482 of *Lecture Notes in Computer Science*, 269–286. Springer.

Fischler, M., and Bolles, R. 1981. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* 24(6):381–395.

Goodfellow, A.; Bengio, Y.; and Courville, A. 2016. *Deep learning*, volume 1. MIT press Cambridge.

Goodfellow, I.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gu, Z.; Rothberg, E.; and Bixby, R. 2020. Gurobi optimizer reference manual. http://www.gurobi.com. Accessed: 2023-06-21.

Guidotti, D.; Pulina, L.; and Tacchella, A. 2021. pynever: A framework for learning and verification of neural networks. In *Proceedings of the 19th International Symposium on Automated Technology for Verification and Analysis (ATVA21)*, volume 12971 of *Lecture Notes in Computer Science*, 357–363. Springer.

Henriksen, P., and Lomuscio, A. 2020. Efficient neural network verification via adaptive refinement and adversarial search. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI20)*, 2513–2520. IOS Press.

Henriksen, P., and Lomuscio, A. 2021. DEEPSPLIT: an efficient splitting method for neural network verification via indirect effect analysis. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 2549–2555. ijcai.org.

Henriksen, P.; Hammernik, K.; Rueckert, D.; and Lomuscio, A. 2021. Bias field robustness verification of large neural image classifiers. In *Proceedings of the 32nd British Machine Vision Conference (BMVC21)*. BMVA Press.

Henzinger, T.; Lukina, A.; and Schilling, C. 2020. Outside the box: Abstraction-based monitoring of neural networks. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI20)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 2433–2440. IOS Press.

Katz, G.; Barrett, C.; Dill, D.; Julian, K.; and Kochenderfer, M. 2017. Reluplex: An efficient SMT solver for verifying deep neural networks. In *Proceedings of the 29th International Conference on Computer Aided Verification (CAV17)*, volume 10426 of *Lecture Notes in Computer Science*, 97–117. Springer.

Katz, G.; Huang, D.; Ibeling, D.; Julian, K.; Lazarus, C.; Lim, R.; Shah, P.; Thakoor, S.; Wu, H.; Zeljic, A.; Dill, D.; Kochenderfer, M.; and Barrett, C. 2019. The Marabou framework for verification and analysis of deep neural networks. In *Proceedings of the 31st International Conference on Computer Aided Verification (CAV19)*, 443–452.

Kouvaros, P., and Lomuscio, A. 2018. Formal verification of cnn-based perception systems. *arXiv preprint arXiv:1811.11373*.

Kouvaros, P., and Lomuscio, A. 2021. Towards scalable complete verification of relu neural networks via dependency-based branching. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI21)*, 2643–2650. ijcai.org.

Kouvaros, P.; Kyono, T.; Leofante, F.; Lomuscio, A.; Margineantu, D.; Osipychev, D.; and Zheng, Y. 2021. Formal analysis of neural network-based systems in the aircraft domain. In *Proceedings of the 24th International Symposium on Formal Methods (FM21)*, volume 13047 of *Lecture Notes in Computer Science*, 730–740. Springer.

Potyka, N. 2021. Interpreting neural networks as quantitative argumentation frameworks. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI21)*, 6463–6470. AAAI Press.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the 18th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI15)*, volume 9351 of *Lecture Notes in Computer Science*, 234–241. Springer.

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

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *Proceedings of the 2nd International Conference on Learning Representations (ICLR14)*.

Tjandraatmadja, C.; Anderson, R.; Huchette, J.; Ma, W.; PATEL, K.; and Vielma, J. 2020. The convex relaxation barrier, revisited: Tightened single-neuron relaxations for neural network verification. *Advances in Neural Information Processing Systems (NeurIPS20)*.

Tjeng, V.; Xiao, K.; and Tedrake, R. 2019. Evaluating robustness of neural networks with mixed integer programming. In *Proceedings of the 7th International Conference on Learning Representations (ICLR19)*.

Tran, H.; Yang, X.; Lopez, D. M.; Musau, P.; Nguyen, L.; Xiang, W.; Bak, S.; and Johnson, T. 2020. NNV: the neural network verification tool for deep neural networks and learning-enabled cyber-physical systems. In *CAV20*, volume 12224 of *Lecture Notes in Computer Science*, 3–17. Springer.

Tran, H.; Pal, N.; Musau, P.; Lopez, D. M.; Hamilton, N.; Yang, X.; Bak, S.; and Johnson, T. 2021. Robustness verification of semantic segmentation neural networks using relaxed reachability. In *Proceedings of the 33rd International Conference on Computer Aided Verification (CAV21)*, volume 12759 of *Lecture Notes in Computer Science*, 263–286. Springer.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018a. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS18)*, 6367–6377.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018b. Efficient formal safety analysis of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS18)*, 6367–6377. Curran Associates, Inc.

Wang, S.; Pei, K.; Whitehouse, J.; Yang, J.; and Jana, S. 2018c. Formal security analysis of neural networks using symbolic intervals. In *Proceedings of the 27th USENIX Security Symposium (USENIX18)*.

Wang, S.; Zhang, H.; Xu, K.; Lin, X.; Jana, S.; Hsieh, C.; and Kolter, J. 2021. Beta-crown: Efficient bound propagation with per-neuron split constraints for complete and incomplete neural network verification. *arXiv preprint arXiv:2103.06624*.

Zhang, H.; Weng, T.-W.; Chen, P.-Y.; Hsieh, C.-J.; and Daniel, L. 2018. Efficient neural network robustness certification with general activation functions. *Advances in Neural Information Processing Systems* 31:4939–4948.