



HAL
open science

Towards the use of formal methods in the certification of vision-based landing systems using high-dimensional ML algorithms

Noemie Cohen, Christophe Gabreau, Mélanie Ducoffe, Claire Pagetti, Xavier Pucel

► **To cite this version:**

Noemie Cohen, Christophe Gabreau, Mélanie Ducoffe, Claire Pagetti, Xavier Pucel. Towards the use of formal methods in the certification of vision-based landing systems using high-dimensional ML algorithms. 13th European Congress of Embedded Real Time Systems (ERTS), Feb 2026, Toulouse, France. <10.82331/ERTS.2026.36>. <hal-05513973>

HAL Id: hal-05513973

<https://hal.science/hal-05513973v1>

Submitted on 17 Feb 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Towards the use of formal methods in the certification of vision-based landing systems using high-dimensional ML algorithms

Noemie Cohen¹, Christophe Gabreau¹, Melanie Ducoffe¹, Claire Pagetti², Xavier Pucel²

¹Airbus, ²ONERA

Abstract: *Formal methods show great promise for supporting the verification process within the avionics certification context. This paper focuses on an industrial use case: runway detection during the landing phase. It presents a methodology for formally verifying this neural network: moving from property definition, through input-domain modelling, to automated proof execution. The result is both a concrete blueprint for this specific aviation use case and a guide to the open problems that must be solved for certifiable, vision-based machine learning.*

Keywords: *Machine learning, Formal verification, Certification.*

1. INTRODUCTION

Avionics is a traditionally cautious domain when it is about adopting new software methodologies, demanding a high level of maturity before implementation. Nonetheless, the considerable potential of machine learning (ML) to augment avionic functionalities and drive innovations that will be instrumental to address 21-st-century environmental challenges (e.g. sustainability, zero emissions), has prompted aeronautical industries to consider integrating ML-based functions. Still this is not the case yet because the Artificial Intelligence (AI) is disrupting the design methodologies of the civil aerospace industry. Indeed embedding AI technics in a safety-critical system will modify the transformation (from design to manufacturing and maintenance) and operating chains. In particular, the paradigm of the existing design standards (e.g., ED12/DO-178) that was used for years to develop safety-related aeronautical systems is impacted. [SE21] pointed out that existing methodologies are requirement based, developing understandable design and traceable code. Though, when it comes to design embedded systems using ML technics, the development becomes also data-driven, leads to abstruse design and untraceable code. Four years later, this worldwide working group is about to release (in 2026) the first industry standard (ED-324/ARP6983) that will fulfill these methodological gaps.

In parallel, by August 2024, the EU council has mandated the European Union Aviation Safety Agency (EASA) to apply the EU AI act in order to guarantee a high level of trustworthiness for the High-Risk systems of the avionic world that will embed AI technologies. This will impact the regulatory text and impose new guidance on the aeronautical industry to certify their

Noemie et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

products integrating ML-based systems. EASA has recently published (March 2024) the issue 2 of *Concept Paper: guidance for Level 1 & 2 machine learning applications* [EAS23] that proposes novel objectives and anticipated means of compliance for certifying such new products in order to guarantee trustworthy AI. Among these objectives, the embedded ML algorithms will be required to properly hold generalization and stability properties so that their outcomes do not challenge the safety of the system they support.

While massive testing can be costly, time-consuming, or even intractable, formal methods offer one of the most promising ways to demonstrate these properties. Although their applications have been explored for classification or regression use cases today, this is less true for object detection models with specific industrial requirements. We focus on a case study in a safety-critical avionics application: a vision-based system for estimating parameters during an aircraft's landing phase. This system identifies a runway in high-resolution images using a single-object detection neural network (Figure 1).

This work proposes the use of formal methods to demonstrate that this ML model verifies the stability property, which defines a level of trustworthiness required for embedded safety-critical systems. Our contributions are twofold:

- A formalization of the stability property tailored specifically to the runway detection use case.
- A state-of-the-art analysis that identifies the critical gaps preventing the end-to-end application of formal methods to this problem.

2. PROPERTY FORMALIZATION

One of the challenges in applying formal methods is to identify and formalize the exact property relevant to the specific application. Stability analysis aims to verify that the object detection ML model's output remains correct for small, real-world variations around a known reference image $x^* \in \mathbb{R}^n$. These variations are called *perturbations*.

Practically, consider a single-object detection model (a feed-forward neural network) $f : \mathbb{R}^n \rightarrow \mathbb{R}^4$ that returns a predicted bounding box (a rectangle encapsulating the object of interest) b defined by 4 coordinates. Let b^* be the ground truth bounding box associated with the image x^* . Let an input domain $\mathcal{C} \subseteq \mathbb{R}^n$ represent the *perturbations* around a reference input. The reachable image of f under \mathcal{C} is $f(\mathcal{C}) = \{f(x) \in \mathbb{R}^4 \mid x \in \mathcal{C}\}$. The stability property \mathcal{P} holds if:

$$f(\mathcal{C}) \text{ satisfies } \mathcal{P} \tag{1}$$

Solving Equation (1) involves three tasks:

1. **Input-domain definition** (\mathcal{C}). Defining \mathcal{C} , the input domain that encompasses all possible images available through plausible perturbations. For an aircraft on approach, geometric transformations induced by yaw, pitch, roll viewpoint shifts are especially relevant. The input domain \mathcal{C} must be tightly defined, so that it accurately reflects the intended geometric perturbations. In the literature, the input domain \mathcal{C} is often represented as a norm-based perturbation region around a fixed input x^* . *White-noise* perturbation for instance is defined by the l_∞ norm ball, i.e., $\mathcal{C} = \{x_0 \in \mathbb{R}^n \mid \|x^* - x_0\|_\infty \leq \epsilon\}$. Unlike norm-based perturbations, which alter pixel intensities directly while keeping their positions fixed, these transformations shift and interpolate pixel values, complicating the definition of the input domain \mathcal{C} .
2. **Formal verification** ($f(\mathcal{C})$). Computing $f(\mathcal{C})$ using state-of-the-art verifiers such as MARABOU [KHI⁺19].
3. **Stability property** (\mathcal{P}). Formalizing the stability property for runway localization: *For all images in the domain \mathcal{C} , the runway must remain correctly localized.* In fact, the requirement is to determine the set of detected bounding box that remains sufficiently close to the ground truth bounding box b^* . A formal-method strategy can be used to verify these properties using the Intersection over Union (IoU) metric, the most widely adopted in object detection.

To enable compatibility with a state-of-the-art verifier, \mathcal{C} , $f(\mathcal{C})$ and \mathcal{P} should be implemented in a way that integrates seamlessly with the solver framework. We begin by specifying the input domain \mathcal{C} , then we compute $f(\mathcal{C})$ and finally formalise the stability property \mathcal{P} .

3. \mathcal{C} : GEOMETRICAL PERTURBATIONS STUDY

During landing approach, the plausible perturbations are the ones possible within the *approach cone*. The latter corresponds to a realistic aircraft trajectory during landing, as well as an envelope for the aircraft attitude that encompasses typical aircraft orientations during approaches on a runway. The approach cone is defined in the Operational Design Domain (ODD), the operating conditions under which a given aircraft system is specifically designed to function. In this methodology, we focus on image perturbations induced by geometric transformations (e.g., rotations). We do not consider all other types of perturbations (occlusion, scene perturbations, etc.)

3.1. Problem formulation

The key steps for defining \mathcal{C} are as follows:

1. **ODD definition.** First, ODD must be clearly characterized [CCD⁺24]. During the final approach phase (landing), the aircraft follows a narrow approach cone that converges toward the runway threshold. The viewpoint changes between consecutive frames are mainly due to small yaw, pitch, and roll rotations (typically within $\pm 3^\circ$), moderate forward translation, and gradual scale increase as the runway gets closer in view. These conditions define the set of admissible perturbations that the network must tolerate.
2. **Geometric transformations.** Second, these physical vari-

ations must be translated into a set of geometric transformations, such as:

$$\mathcal{T} = \{T_{\text{yaw}}, T_{\text{pitch}}, T_{\text{roll}}, T_{\text{scale}}, T_{\text{shift}}\}$$

Each transformation $T_\kappa \in \mathcal{T}$ is parameterised by realistic intervals, and acts on the spatial structure of the image, displacing pixels and introducing interpolation. To reproduce such viewpoint changes, the 3-D scene must be projected to a 2-D image. Then, the aircraft's movement must be translated into a composition of elementary operators intervals on the image of the transformations: rotation, translation, scaling, shearing, projection. We group all d transformation parameters into a single vector $\kappa \in \mathbb{R}^d$. We consider realistic bounds for each parameter denoted as $\kappa \in [\underline{\kappa}, \overline{\kappa}]$.

3. **Input domain definition.** Third, an input domain must be defined to encompass all possible transformed images within the ODD. A geometric image transformation is usually defined as a function $g : \mathbb{R}^d \times \mathbb{R}^n \rightarrow \mathbb{R}^n$, where $\kappa \in \mathbb{R}^d$ is the transformation parameter (e.g. rotation angle). A transformed image is $g_x(\kappa) \in \mathbb{R}^n$. Geometric stability, derived from Equation (1), is to verify whether a neural network f satisfies the stability property \mathcal{P} for all transformed images $g_x^{[\underline{\kappa}, \overline{\kappa}]} = \bigcup_{\kappa \in [\underline{\kappa}, \overline{\kappa}]} g_x(\kappa)$. Due to the highly non-linear nature of geometric transformations, computing the exact set $g_x^{[\underline{\kappa}, \overline{\kappa}]}$ is not tractable. To address this, we apply over-approximation and define an input domain \mathcal{C} that over-approximates $g_x^{[\underline{\kappa}, \overline{\kappa}]}$ such that:

$$\mathcal{C} \supseteq g_x^{[\underline{\kappa}, \overline{\kappa}]} \quad (2)$$

3.2. State of the art

ODD definition. [CCD⁺24] presents the design and the validation of a dataset [DCF⁺23] that would comply with the ODD of the approach cone. Other works focus on the methodology for defining the ODD, for instance through a model-based approach [TGR⁺24] or via data-centric characterization [KAPE23]. Furthermore, reviews of vision-based landing scenarios, provide a practical taxonomy of ODD parameters [XTGL22].

Geometric transformations. [HZ03] establish the basics of geometry for images and is a good basis for determining some properties of the position/shape and other geometric considerations of the runway (or any other scenery element) depending on the range of attitude of the aircraft. This geometry is practically applied in Perspective-n-Point (PnP) algorithms, which estimate the 6-DOF (Degrees of Freedom) camera pose from a set of known 3D world points and their corresponding 2D image projections. Such methods provide the exact mathematical model to link 3D aircraft motion (e.g., yaw, pitch, roll) to the 2D transformations seen by the vision system [WZC22].

Input domain definition. Several methods exist to over-approximate \mathcal{C} in Equation (2), necessitating a trade-off between abstraction precision or computational efficiency. Interval-based methods (Semantify-NN [MWC⁺20], FGV [YLMS22]) prioritize scalability and GPU efficiency but suffer from lower precision. [YLMS22] provides a specific geometric robust training for FGV. Conversely, optimization-based approaches

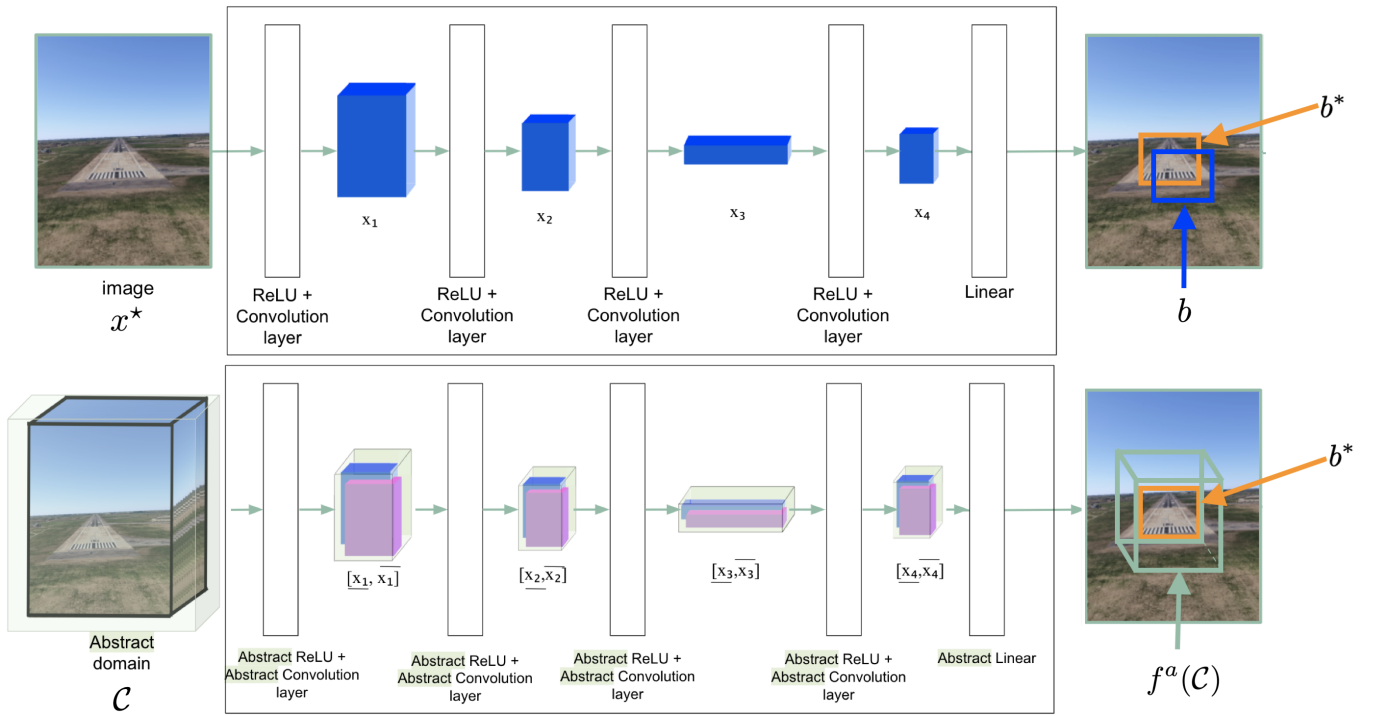


Figure 1. Inference versus abstract interpretation. Top: a single image x^* is forwarded through the convolutional network, producing a bounding-box prediction b . Bottom: the same network is executed on an *input domain* \mathcal{C} that encloses all admissible perturbations. Each layer is replaced by its abstract counterpart, propagating an interval $[x_k, \bar{x}_k]$ instead of a single matrix x_k . The output is $f^a(\mathcal{C})$.

(DeepG, PWL) leverage Lipschitz continuity and solvers (LP or MILP) to construct tighter domains; however, they are computationally expensive and struggle to scale to large networks. Alternative approaches, such as randomized smoothing [FBV21] [LWX⁺21] or asymptotic Lipschitz bounding [WXRH23], are excluded as they fail to provide the strict deterministic guarantees required for formal verification.

3.3. Limitations

State-of-the-art methods already handle compositions of rotations, translations, scaling, and shearing (Figure 2 (b)), but none have yet incorporated the perspective-projection operator as in Figure 2 (c) needed to represent aircraft orientations during the approaches on a runway.

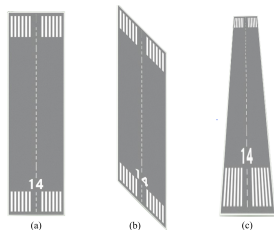


Figure 2. Images of a runway under different transformations: (a) original image; (b) shearing transformation; (c) projective transformation.

4. $f(\mathcal{C})$: FORMAL VERIFICATION

4.1. Problem statement

Given \mathcal{C} , the objective is to compute $f(\mathcal{C})$, i.e., all bounding boxes the model could predict from inputs in \mathcal{C} .

4.2. State of the art

Formal verification methods such as Satisfiability Modulo Theory, MILP, and abstract interpretation are widely used for computing $f(\mathcal{C})$ and then verify Equation (1) [LAL⁺21, UM21, MBT⁺22]. Among these, abstract interpretation is particularly suitable for scaling up to large networks with high-dimensional inputs such as images (millions of pixels). It relies on *over-approximations* of $f(\mathcal{C})$, denoted by $f^a(\mathcal{C})$ where f^a is a *relaxation* of f such that $f(\mathcal{C}) \subseteq f^a(\mathcal{C})$. The stability property holds if:

$$f^a(\mathcal{C}) \text{ satisfies } \mathcal{P} \quad (3)$$

Hence, Equation (3) \implies Equation (1). Each layer is replaced by its abstract counterpart as illustrated in Figure 1. More precise abstractions come at the cost of increased computation but are more likely to verify the property. Linear relaxation-based perturbation analysis (LiRPA) is state-of-the-art as its implementation in `auto-LiRPA` [XSZ⁺20] won the International Competition on Verification of Neural Networks (VNN-Comp) [BBJW24, BMB⁺23]. Methods such as CROWN [ZWC⁺18], DeepPoly [SGPV19], α -CROWN [XZW⁺20] propagate *linear* bounds:

$$f^a(\mathcal{C}) = \{(b, x^*) \in \mathbb{R}^4 \times \mathcal{C} \mid \underline{A} \cdot x^* + \underline{B} \leq b \leq \overline{A} \cdot x^* + \overline{B}\} \quad (4)$$

Where $\underline{A}, \overline{A} \in \mathbb{R}^{4 \times n}$ and $\underline{B}, \overline{B} \in \mathbb{R}^4$ are relaxation coefficient. Interval Bound Propagation (IBP) [Moo66] can be seen as the special case where $\underline{A} = \overline{A} = 0$ in Equation (4), as illustrated in Figure 1. Existing tools are ERAN [MSS⁺21], DECOMON [Duc22] or `auto-LiRPA`.

4.3. Limitation

Even with abstract interpretation, scalability remains a bottleneck: keeping the over-approximation $f^a(\mathcal{C})$ sufficiently tight to verify Equation (3) requires high-precision abstractions whose computational cost grows rapidly with network size and input dimensionality. Solvers are being updated to support new models, but there is often a time lag between the release of new machine learning models and the solvers' ability to support them. Only specific versions of YOLO [RDGF16], like a modified YOLOv5 nano [Joc20] (without SiLU layers), have been adapted to work with solvers like `Auto-LiRPA`.

5. \mathcal{P} : IOU METRIC STUDY

Given $f^a(\mathcal{C})$, we aim to define the stability property in the case of object detection. The goal is to ensure that plausible perturbations do not degrade the object detection model's correct behavior, i. e. accurately localizing the runway.

5.1. Problem formulation

IoU is commonly used in computer vision to evaluate object detection models because it directly measures the overlap between predicted and ground-truth bounding boxes, providing a clear assessment of localization accuracy. This simplicity and effectiveness make it a standard metric in the field [EVGW⁺10]. This work concentrates on the localization property, ensuring that plausible domain perturbations do not degrade the object detection model's correct behavior, defined as accurately localizing the runway and proposing a tight bounding box. The predicted bounding box b is in orange in Figure 1 (top) and the ground-truth box b^* is in blue. For two boxes b, b^* , define the IoU:

$$\text{IoU}(b, b^*) = \frac{|b \cap b^*|}{|b \cup b^*|}, \quad 0 \leq \text{IoU} \leq 1$$

Fix a threshold $\tau \in (0, 1]$. The set of outputs that meet the localization criterion is:

$$\mathcal{P}_\tau = \{b \in \mathbb{R}^4 \mid \text{IoU}(b, b^*) \geq \tau\} \quad (5)$$

Given b^* fixed, it corresponds to all existing bounding boxes that have an IoU above a threshold τ . The property in Equation (3) can be specified for this use case. Let $f^a(\mathcal{C})$ be the approximation of all bounding boxes the model could predict through \mathcal{C} . The property checks that $f^a(\mathcal{C})$ is a subset of \mathcal{P}_τ :

$$\text{IoU}(f^a(\mathcal{C}), b^*) \subseteq \mathcal{P}_\tau? \quad (6)$$

To verify Equation (6), the goal is to bound the minimum IoU value for all boxes in $f^a(\mathcal{C})$ defined in Equation (4). Verifiers like `auto-LiRPA` [XSZ⁺20] and its extensions do not inherently support IoU. This is achieved by first computing $f^a(\mathcal{C})$ using abstract interpretation, as detailed in Section 4, then propagating this set through the IoU function. The IoU metric itself is non-convex and multi-dimensional, complicating formal verification.

5.2. State of the art

Some works have recently emerged later for formal verification in object detection. The authors of [REAM⁺24] encoded the IoU as a neural network using operators already

supported by LiRPA solvers. This approach adapts existing formal methods to the unique challenges posed by metrics like IoU, which are critical for evaluating object detection model performance. The IoU function is treated as a latent layer within the network, which simplifies the verification process. However, [CDB⁺24] approached the IoU as a metric rather than a network layer, which requires a more nuanced analysis of its extreme (minimum and maximum) values. This metric-centered approach is closer to the verification challenges seen in classification tasks, where metrics like cross-entropy are central. [NS25] uses previous verification method for a YOLOv2 model.

5.3. Limitations

In standard machine learning tasks, a threshold τ of 0.5 is typically used for the IoU to consider a detection correct. However, depending on the specific certification problems, this value must be found and justified to ensure it is sufficient for safety.

6. DISCUSSION AND NEXT STEPS

To achieve an end-to-end formal verification for runway detection, we propose the following roadmap to address the identified limitations:

- **Geometric projection (Addressing Sec. 3):** We plan to develop a tool that links aircraft attitude (yaw, pitch, roll) directly to the input domain \mathcal{C} , capturing the specific distortions of an approach cone.
- **Scalability (Addressing Sec. 4):** To handle state-of-the-art object detectors, we must enhance the verification tools directly. We plan to contribute to libraries like `auto-LiRPA`, `DECOMON` by implementing currently unsupported layers. Furthermore, we will pursue research on algorithmic optimizations to ensure abstract interpretation scales efficiently to large, high-dimensional networks.
- **IoU threshold (Addressing Sec. 5):** We plan to conduct a study with domain experts to determine the exact IoU threshold value τ required to satisfy safety requirements. This follows the approach of [Pai25], which proposes a method to translate quantitative safety objectives into concrete performance metrics for the machine learning component.

7. CONCLUSIONS

From a certification standpoint, the ultimate aim is to demonstrate that high-dimensional ML algorithms are trustworthy and can be integrated into safety-critical systems, meeting stringent regulatory requirements for properties such as generalization, stability, and robustness. The strategies proposed in this methodology advance this goal by highlighting how formal methods can ensure the stability of ML algorithms for runway detection. Through a literature review, we identified the necessary building blocks, both existing and missing, for applying formal methods in this scenario. While substantial research exists, to date, no single tool can accept an input domain based on geometric transformations and perform stability verification against the IoU metric. Consequently, achieving full, certifiable trustworthiness will require sustained effort in formal methods research.

8. ACKNOWLEDGEMENT

This work has benefited from the AI cluster ANITI2 funded by the French government through the ANR under the France 2030 program (grant ANR-23-IACL-0002).

REFERENCES

- [BBJW24] Christopher Brix, Stanley Bak, Taylor T Johnson, and Haoze Wu. The 5th international verification of neural networks competition (vnn-comp 2024): Summary and results. *arXiv preprint arXiv:2412.19985*, 2024.
- [BMB⁺23] Christopher Brix, Mark Niklas Müller, Stanley Bak, Taylor T Johnson, and Changliu Liu. First three years of the international verification of neural networks competition (vnn-comp). *International Journal on Software Tools for Technology Transfer*, 25(3):329–339, 2023.
- [CCD⁺24] Cyril Cappi, Noémie Cohen, Mélanie Ducoffe, Christophe Gabreau, Laurent Gardes, Adrien Gauffriau, Jean-Brice Ginestet, Franck Mamalet, Vincent Mussot, Claire Pagetti, et al. How to design a dataset compliant with an ml-based system odd? *arXiv preprint arXiv:2406.14027*, 2024.
- [CDB⁺24] Noémie Cohen, Mélanie Ducoffe, Ryma Boumazouza, Christophe Gabreau, Claire Pagetti, Xavier Pucel, and Audrey Galametz. Verification for object detection–ibp iou. *arXiv preprint arXiv:2403.08788*, 2024.
- [DCF⁺23] Mélanie Ducoffe, Maxime Carrere, Léo Féliers, Adrien Gauffriau, Vincent Mussot, Claire Pagetti, and Thierry Sammour. Lard–landing approach runway detection–dataset for vision based landing. *arXiv preprint arXiv:2304.09938*, 2023.
- [Duc22] Melanie Ducoffe. Decomon: Automatic certified perturbation analysis of neural networks, 2022. 2023-11-20.
- [EAS23] EASA. Easa concept paper: First usable guidance for level 1 and 2 machine learning applications, 2023.
- [EVGW⁺10] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.
- [FBV21] Marc Fischer, Maximilian Baader, and Martin Vechev. Scalable certified segmentation via randomized smoothing. In *International Conference on Machine Learning*, pages 3340–3351. PMLR, 2021.
- [HZ03] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [Joc20] Glenn Jocher. Ultralytics YOLOv5. Available at <https://github.com/ultralytics/yolov5>, 2020. Version 7.0.
- [KAPE23] Fateh Kaakai, Sridhar Adibhatla, Ganesh Pai, and Emmanuelle Escorihuela. Data-centric operational design domain characterization for machine learning-based aeronautical products. In *International Conference on Computer Safety, Reliability, and Security*, pages 227–242. Springer, 2023.
- [KHI⁺19] Guy Katz, Derek A Huang, Duligur Ibeling, Kyle Julian, Christopher Lazarus, Rachel Lim, Parth Shah, Shantanu Thakoor, Haoze Wu, Aleksandar Zeljić, et al. The marabou framework for verification and analysis of deep neural networks. In *Computer Aided Verification: 31st International Conference, CAV 2019, New York City, NY, USA, July 15-18, 2019, Proceedings, Part I 31*, pages 443–452. Springer, 2019.
- [LAL⁺21] Changliu Liu, Tomer Arnon, Christopher Lazarus, Christopher Strong, Clark Barrett, Mykel J Kochenderfer, et al. Algorithms for verifying deep neural networks. *Foundations and Trends® in Optimization*, 4(3-4):244–404, 2021.
- [LWX⁺21] Linyi Li, Maurice Weber, Xiaojun Xu, Luka Rimanic, Bhavya Kailkhura, Tao Xie, Ce Zhang, and Bo Li. Tss: Transformation-specific smoothing for robustness certification. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 535–557, 2021.
- [MBT⁺22] Mark Huasong Meng, Guangdong Bai, Sin Gee Teo, Zhe Hou, Yan Xiao, Yun Lin, and Jin Song Dong. Adversarial robustness of deep neural networks: A survey from a formal verification perspective. *IEEE Transactions on Dependable and Secure Computing*, 2022.
- [Moo66] Ramon E Moore. *Interval analysis*. Prentice-Hall, 1966.
- [MSS⁺21] Christoph Müller, François Serre, Gagandeep Singh, Markus Püschel, and Martin Vechev. Scaling polyhedral neural network verification on gpus. *Proceedings of Machine Learning and Systems*, 3:733–746, 2021.
- [MWC⁺20] Jeet Mohapatra, Tsui-Wei Weng, Pin-Yu Chen, Sijia Liu, and Luca Daniel. Towards verifying robustness of neural networks against a family of semantic perturbations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 244–252, 2020.

- [NS25] Ashutosh Kumar Nirala and Soumalya Sarkar. Towards certified object detectors: Certified runway detection using yolo. In *2025 IEEE International Conference on Image Processing (ICIP)*, pages 827–832. IEEE, 2025.
- [Pai25] Ganesh J Pai. Deriving safety-related performance requirements for machine learnt aeronautical applications. In *Digital Avionics Systems Conference (DASC) 2025*, 2025.
- [RDGF16] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [REAM⁺24] Avraham Raviv, Yizhak Y Elboher, Michelle Aluf-Medina, Yael Leibovich Weiss, Omer Cohen, Roy Assa, Guy Katz, and Hillel Kugler. Formal verification of object detection. *arXiv preprint arXiv:2407.01295*, 2024.
- [SE21] SAE International and EUROCAE. Artificial Intelligence in Aeronautical Systems: Statement of Concerns. Report SAE AIR6988 / EUROCAE ER-022, SAE International and EUROCAE, April 2021.
- [SGPV19] 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):1–30, 2019.
- [TGR⁺24] Christoph Torens, Siddhartha Gupta, Nirmal Roy, Jasper Sprockhoff, and Umut Durak. From operational design domain to runtime monitoring of ai-based aviation systems. In *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, pages 1–9. IEEE, 2024.
- [UM21] Caterina Urban and Antoine Miné. A review of formal methods applied to machine learning. *CoRR*, abs/2104.02466, 2021.
- [WXRH23] Fu Wang, Peipei Xu, Wenjie Ruan, and Xiaowei Huang. Towards verifying the geometric robustness of large-scale neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 15197–15205, 2023.
- [WZC22] Zhaoyang Wang, Dan Zhao, and Yunfeng Cao. Visual navigation algorithm for night landing of fixed-wing unmanned aerial vehicle. *Aerospace*, 9(10):615, 2022.
- [XSZ⁺20] Kaidi Xu, Zhouxing Shi, Huan Zhang, Yihan Wang, Kai-Wei Chang, Minlie Huang, Bhavya Kailkhura, Xue Lin, and Cho-Jui Hsieh. Automatic perturbation analysis for scalable certified robustness and beyond. *Advances in Neural Information Processing Systems*, 33, 2020.
- [XTGL22] Long Xin, Zimu Tang, Weiqi Gai, and Haobo Liu. Vision-based autonomous landing for the uav: A review. *Aerospace*, 9(11):634, 2022.
- [XZW⁺20] Kaidi Xu, Huan Zhang, Shiqi Wang, Yihan Wang, Suman Jana, Xue Lin, and Cho-Jui Hsieh. Fast and complete: Enabling complete neural network verification with rapid and massively parallel incomplete verifiers. *arXiv preprint arXiv:2011.13824*, 2020.
- [YLMS22] Rem Yang, Jacob Laurel, Sasa Misailovic, and Gagandeep Singh. Provable defense against geometric transformations. *arXiv preprint arXiv:2207.11177*, 2022.
- [ZWC⁺18] Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. *Advances in neural information processing systems*, 31, 2018.