Verification for Object Detection – IBP IoU

Noémie Cohen, Mélanie Ducoffe, Ryma Boumazouza, Christophe Gabreau, Claire Pagetti, Xavier Pucel, Audrey Galametz

Verification for Object Detection – IBP IoU

Noémie Cohen[1], Mélanie Ducoffe[1], Ryma Boumazouza[1], Christophe Gabreau[1],
Claire Pagetti[2], Xavier Pucel[2] and Audrey Galametz[1]
[1] Airbus, [2] ONERA

January 26, 2024

Abstract

We introduce a novel Interval Bound Propagation (IBP) approach for the formal verification of object detection models, specifically targeting the Intersection over Union (IoU) metric. The approach has been implemented in an open source code, named IBP IoU, compatible with popular abstract interpretation based verification tools. The resulting verifier is evaluated on landing approach runway detection and handwritten digit recognition case studies. Comparisons against a baseline (Vanilla IBP IoU) highlight the superior performance of IBP IoU in ensuring accuracy and stability, contributing to more secure and robust machine learning applications.

1 Introduction

During the last decade, formal verification has widely been used on machine learning models, especially neural networks, to assess their correctness in behaving as expected in all situations.

1.1 Motivating Example: Aircraft Pose Estimation

We consider a vision-based aircraft pose estimation, that aims at positioning the aircraft with respect to a visible runway [FB81, BFBC+21].

(a) No impact (b) Slight impact (c) Strong impact (d) Strong impact

Figure 1: Impact of a perturbation on the object detection

Most solutions (e.g. two-stage machine learning pipeline proposed by Daedalean [BFBC+21]) rely on the use of an *object detection model* that identifies one or more runways (one runway identified in figure 1a) in an high resolution input image. Single object detection consists in localizing an object by providing *bounding box* candidates surrounding said object.

Let us assume that the majority of the key points are visible (e.g. no obstructions) on the input image and that the object detection is capable of identifying the runway. In this work we focus on the *stability* property which states that adding any domain perturbation considered as plausible by the experts should not degrade the correct behavior of the object detection stage. A correct behavior consists in: 1) localizing the runway and 2) propose a tight bounding box. Figure 1[1] shows several impacts on the detection due to some perturbation: from none (1a), acceptable (1b) to unacceptable (1c, 1d). This property is a minimal requirement to grant some confidence on the ML-based system and that was also highlighted by [KLE+23].

[1] All the figures rely on images extracted from LARD dataset [DCF+23].

1.2 Contributions

To our knowledge, the case of stability of object detection has not yet been fully addressed by formal methods. To verify stability, we need to rely on the *Intersection over Union* (IoU), a common metric for evaluating the performance of object detection. One particularity of IoU computation is that it is non linear, while formal verification approaches focus on linear properties.

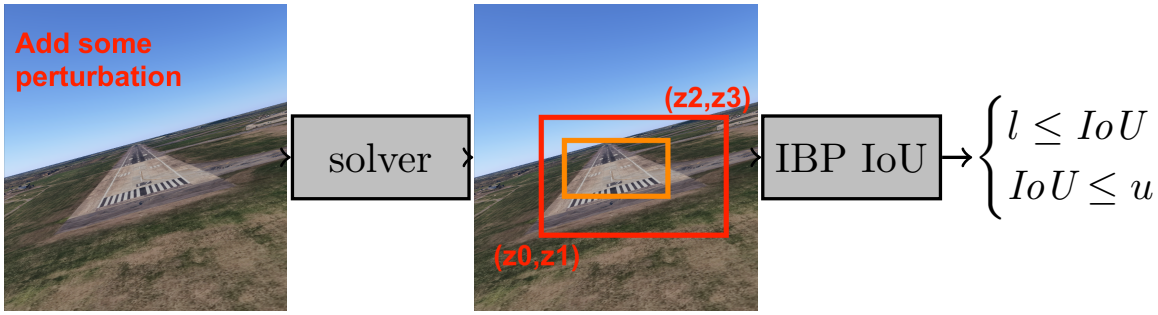


Figure 2: An overview of the IBP IoU approach.

As a result, we propose a two-step approach as shown in Fig. 2. Step 1: we apply a perturbation on the input and rely on classical verification tools such as ERAN [MSS+21], Auto-LIRPA [XSZ+20] or DECOMON [Duc] to obtain the reachable outputs. Instead of having bounding boxes, the output is a list of *extended bounding boxes* that are not defined by their four coordinates but a reachable interval on each coordinate. In Fig. 2, there was a unique candidate and the extended bounding box due to the perturbation are all rectangles that include the orange one and that are included in the red one. In effect, we consider 3 types of perturbation: white-noise, brightness and contrast perturbations. Brightness and contrast were implemented in ERAN [BBS+19] but up to our knowledge, the code was not available. As a result, we propose an implementation for Auto-LiRPA. Step 2: IBP IoU is called to estimate the propagation effect on the IoU. IBP IoU relies on Interval Bound Propagation (IBP) [MGV18, GMDC+18, GDS+18a]. Our method is scalable as the algorithm has a low complexity. Finally, we compare the efficiency against a naive baseline, Vanilla_IBP_IoU. This baseline consists in bounding the primitive operators of the IoU function with IBP.

2 Problem Statement

The problem at hand is to ensure the stability of an object detection in the presence of perturbation.

2.1 Reminder on Object Detection

Single object detection consists in delineating one or more objects in the image using the tightest possible bounding box. There exist many neural network architectures to address this task among which we can mention R-CNN [GDDM14], YOLO (You Only Look Once, [JEL+22]) or FCOS (Fully Convolutional One-Stage object detection, [TSCH19]).

The output —regardless of the model family— is presented as a list containing information about bounding box candidates. More precisely, we can define a single object detection model as a function f_{OD}:

$$\begin{array}{rccl} f_{OD}: & \mathcal{X} \subseteq \mathbb{R}^n & \longmapsto & \mathbb{R}^{k\times 5} \\ & x & \longrightarrow & \mathbf{c} = [c^1, c^2, \dots c^k] \ \textit{such that} \ \forall i,\ c^i = [o^i, b^i] \end{array} \tag{1}$$

where n denotes the dimension of the input space, k is the number of candidate proposals and each candidate $c^i = [o^i, b^i]$ is defined by:

- Objectness $\mathbf{o^i}$: model's probability of an object to be in c^i.
- Location $\mathbf{b^i} = [z_0^i, z_1^i, z_2^i, z_3^i]$: where z_i are the coordinates of the bottom-left and upper-right corners (cf. Fig. 2). We define the set of bounding boxes as $\mathcal{B} = \{[z_0, z_1, z_2, z_3] \in \mathbb{R}_+^4 \mid z_0 \leq z_2, z_1 \leq z_3\}$

2.2 Reminder on Intersection Over Union

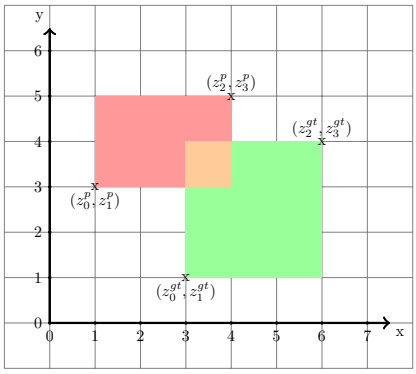
Intersection over Union (IoU, also known as Jaccard Index) [RTG+19] is a similarity metric that quantifies the overlap between two bounding boxes by calculating the ratio of their intersection area to their union area.

Definition 1 (Intersection over Union – *IoU*) *Let a bounding box* $b_0 = [z_0^0, z_1^0, z_2^0, z_3^0]$. *Its area* $a(b_0)$ *is defined by the function* $a : \mathcal{B} \longmapsto \mathbb{R}_+$ *where* $a(b_0) = (z_2^0 - z_0^0) \times (z_3^0 - z_1^0)$.

Let another bounding box $b_1 = [z_0^1, z_1^1, z_2^1, z_3^1]$. *Their intersection* $i(b_0, b_1)$ *is defined by the function* $i : \mathcal{B}^2 \longmapsto \mathcal{B}$, $i(b_0, b_1) = (\max_{j=0,1} z_0^j, \max_{j=0,1} z_1^j, \min_{j=0,1} z_2^j, \min_{j=0,1} z_3^j)$.

The IoU is a function $\mathcal{B}^2 \longmapsto [0, 1]$ *such that:*

$$IoU(b_0, b_1) = \frac{a(i(\mathbf{b_0}, \mathbf{b_1}))}{a(\mathbf{b_0}) + a(\mathbf{b_1}) - a(i(\mathbf{b_0}, \mathbf{b_1}))} \tag{2}$$



Orange bounding box: intersection $\mathbf{b}_{gt \cap p} = i(\mathbf{b}_p, \mathbf{b}_{gt})$.

Areas: $a(\mathbf{b}_p) = 6$, $a(\mathbf{b}_{gt}) = 9$ and $a(\mathbf{b}_{gt \cap p}) = 1$.

$IoU(\mathbf{b}_p, \mathbf{b}_{gt}) = \frac{1}{14}$ (cf. eq. 2)

Figure 3: Ground truth $\mathbf{b}_{gt}$ (green) and predicted $\mathbf{b}_p$ (red) bounding boxes

2.3 Stability Property

We consider an airport featuring a single runway and an object detection model f_{OD}. As detailed in the introduction, we want the model to be stable on plausible perturbations. In the rest of our study, we will assume that the local perturbations studied have no impact on the position of the ground-truth bounding box. We formalise the single box stability property as a minimal threshold on the IoU between the predicted box with highest confidence (i.e. objectness score) and the minimal bounding box containing the runway, known as the ground-truth box $\mathbf{b}_{gt}$. For simplicity of notation, we supposed that f_{OD} proposes candidates ordered by their objectness (i.e. $\forall 1 \leq i < j \leq k, o^i \geq o^j$) and we denote IoU_{gt} as the IoU between a bounding box and the ground-truth bounding box.

Property 1 (Worst Case IoU as a necessary condition for Box Stability) *Consider an input image* $s_0 \in \mathcal{X}$ *with a single runway, a local perturbation domain* $\Omega(s_0)$, *and an object detection model* f_{OD}. *The model reaches* Bounding Box Stability *with threshold* t *if, and only if the candidate box with the highest objectness score (here* b^1 *due to our hypothesis of ordered candidates) overlaps the ground truth:*

$$\begin{aligned} &\min_{s \in \Omega(s_0)} IoU_{gt}(b^1) \geq t \\ &\text{where } f_{OD}(s) = [c^1, \ldots, c^k] \text{ and } c^i = o^i \times b^i \end{aligned} \tag{3}$$

3 Verification for Object Detection

The background being reminded and the stability property being properly defined, let us now formalize the verification strategy based on Interval Bound Propagation that we propose.

3.1 General property

The first step of our approach (cf Fig. 2) consists in using state-of-the-art solvers to compute the effect of the perturbations on the object detection part. As a result, the object detection model must be compatible with available abstract interpretation methods: this restricts in particular the choice of internal operators and how they are composed to obtain the set of candidate proposals. Luckily, more and more operators are supported thanks to initiative such as [XSZ+20]. However, *scalability* still remains an issue meaning the model must not be too large.

Let us consider a given input s_0 and a local perturbation $\Omega(s_0)$ (referred to as Ω for the sake of simplicity). Verification methods yield minimum $\underline{f_{OD}}(\Omega) = [\underline{\mathbf{c^i}}(\Omega)]_{i=1}^k$ and maximum $\overline{f_{OD}}(\Omega) = [\overline{\mathbf{c^i}}(\Omega)]_{i=1}^k$ bounds for every candidates. This entails that $\forall\ s' \in \Omega, \underline{f_{OD}}(\Omega) \leq f_{OD}(s') \leq \overline{f_{OD}}(\Omega)$ where $\leq, \geq$ are element-wise inequalities. With these bounds, we can derive a sufficient condition, defined in theorem 1, to ensure property 1.

Theorem 1 (Certifying Box Stability with IBP) *Consider an input $s_0 \in \mathcal{X}$, a local perturbation domain $\Omega(s_0)$, an object detection model f_{OD}, and a verification method providing a lower and upper bounds of the model given the perturbation domain (respectively $\underline{f_{OD}}(\Omega), \overline{f_{OD}}(\Omega)$). We remind that b^1 is always the best candidate box, then:*

$$\min_{\underline{b^1}(\Omega) \leq \mathbf{b} \leq \overline{b^1}(\Omega)} IoU_{gt}(\mathbf{b}) \geq t \implies \min_{s \in \Omega} IoU_{gt}(\mathbf{b^1}) \geq t \tag{4}$$

Our contribution lies in propagating these intervals through the IoU function thanks to interval extension [S+09]. Indeed, such a bound can then ensure the left part of equation 4. We call such an approach an *IoU interval extension*. To proceed, the perturbation domain on the objectness and location, computed by Step 1, must be represented by intervals.

Definition 2 (*IoU* interval extension) *Given a box $\mathbf{b}$ and a local interval perturbation, an interval extension is any interval that soundly bounds $IoU(b)$:*

$$\forall\ \underline{\mathbf{b}}, \overline{\mathbf{b}} \in \mathbb{R}^4,\ \underline{\mathbf{b}} \leq \mathbf{b} \leq \overline{\mathbf{b}} \implies \underline{IoU_{gt}}([\underline{\mathbf{b}}, \overline{\mathbf{b}}]) \leq IoU_{gt}(\mathbf{b}) \leq \overline{IoU_{gt}}([\underline{\mathbf{b}}, \overline{\mathbf{b}}]) \tag{5}$$

Bounding the IoU is challenging for multiple reasons: (i) it is a multi-dimensional inputs function; (ii) it is neither convex, nor concave (iii) it is not piece-wise linear. To address and solve this problem, we propose two interval extension methods:

- **Vanilla_*IoU*** bounds the primitive operators and composes them using the rules of interval extension arithmetic (see section 3.2);
- **Optimal_*IoU*** computes the highest and lowest IoU values than can be reached by bounding boxes from the input corner intervals (see section 3.3).

3.2 Vanilla_*IoU*- bounding the primitive operators

The fundamental idea is to enhance the traditional arithmetic system by employing closed intervals as basic data types, rather than single point values, on each primitive of the IoU function. Indeed, IoU, from equation 2 can be formulated as a combination of so-called primitive functions: minimum, maximum, addition, subtraction, multiplication and division by a positive scalar. We remind in figure 4 the arithmetic interval for those operators, excerpt from [S+09, HJVE01].

We apply inductively the rules to compute IoU interval extension. Let us consider a box $b = [z_0, z_1, z_2, z_3]$ and a perturbation domain such that $b \in [\underline{b}, \overline{b}]$ with $\underline{b} = [\underline{z_0}, \underline{z_1}, \underline{z_2}, \underline{z_3}]$ and $\overline{b} = [\overline{z_0}, \overline{z_1}, \overline{z_2}, \overline{z_3}]$. Then the area $a(b) \in [\underline{a}, \overline{a}]$ with:

$$\begin{aligned}[\underline{a}, \overline{a}] = [&\max(\underline{z_2} - \overline{z}_0, 0) \times \max(\underline{z_3} - \overline{z}_1, 0),\\ &\max(\overline{z}_2 - \underline{z_0}, 0) \times \max(\overline{z}_3 - \underline{z_1}, 0)]\end{aligned} \tag{6}$$

The intersection of b with $b_{gt} = [z_0^{gt}, z_1^{gt}, z_2^{gt}, z_3^{gt}]$ belongs to $i(b \cap b_{gt}) \in [\underline{i}, \overline{i}]$ with:

$$[\underline{i_k}, \overline{i}_k]_{k=0}^3 = \begin{cases} [\max(\underline{z_k}, z_k^{gt}), \max(\overline{z}_k, z_k^{gt})] & \text{if } k \leq 1 \\ [\min(\underline{z_k}, z_k^{gt}), \min(\overline{z}_k, z_k^{gt})] & \text{else} \end{cases} \tag{7}$$

Operation	Notation	Formula
Addition	$+$	$[a,b]+[c,d]=[a+c,b+d]$
Subtraction	-	$[a,b]-[c,d]=[a-d,b-c]$
Multiplication ($a\geq 0$, $c\geq 0$)	$\times_{\leq 0}$	$[a,b]\cdot[c,d]=[a\cdot c,b\cdot d]$
Division ($a\geq 0$)	$/$	$\frac{1}{[a,b]}=[\frac{1}{b},\frac{1}{a}]$
Maximum	max	$\max([a,b],[c,d])=[\max(a,c),\max(b,d)]$
Minimum	min	$\min([a,b],[c,d])=[\min(a,c),\min(b,d)]$

Figure 4: Summary of interval arithmetic operations used for Vanilla IoU

We denote by $u=a(i(b\cap b_{gt}))$, then $u\in[\underline{u},\overline{u}]$:

$$[\underline{u},\overline{u}]=[\underline{a}([\underline{i},\overline{i}]),\overline{a}([\underline{i},\overline{i}])] \tag{8}$$

Definition 3 (Vanilla_*IoU* extension) *Given a box* $\mathbf{b}$ *and a local interval perturbation, we denote as Vanilla interval extension* $[\underline{IoU}_v,\overline{IoU}_v]$*:*

$$[\underline{IoU}_v,\overline{IoU}_v]=\left[\frac{\underline{u}}{a(\mathbf{b}_{gt})+\overline{a}+\overline{u}},\frac{\overline{u}}{a(\mathbf{b}_{gt})+\underline{a}+\underline{u}}\right] \tag{9}$$

By construction, this interval soundly bounds IoU.

3.3 Optimal_IoU extension - exact bounds

Our main contribution involves an optimal interval extension of IoU, denoted $[\underline{IoU}_{opt},\overline{IoU}_{opt}]$.

Theorem 2 (Optimality) *Considering any sound interval extension of IoU* $[\underline{IoU},\overline{IoU}]$*, the following relation holds:*

$$\underline{IoU}([\underline{\mathbf{b}},\overline{\mathbf{b}}])\leq\underline{IoU}_{opt}([\underline{\mathbf{b}},\overline{\mathbf{b}}])\leq IoU_{gt}(\mathbf{b})\leq\overline{IoU}_{opt}([\underline{\mathbf{b}},\overline{\mathbf{b}}])\leq\overline{IoU}([\underline{\mathbf{b}},\overline{\mathbf{b}}]) \tag{10}$$

Our result is based on the analysis of the variations of IoU according to its four input coordinates in order to deduce its exact bounds. First, we provide the different variations of the IoU function in fig. 5, using the partial derivatives formulated in equations 11 and 12. For the purpose of enhancing readability, we abbreviate certain mathematical notations as follows: $x_{max}=\min(z_2,z_2^{gt})$, $x_{min}=\max(z_0,z_0^{gt})$, $y_{max}=\min(z_3,z_3^{gt})$, $y_{min}=\max(z_1,z_1^{gt})$, $d_{gt}(\mathbf{b})=a(\mathbf{b}_{gt})+a(\mathbf{b})-a\big(i(\mathbf{b},\mathbf{b}_{gt})\big)$ $c_{k=2,3}=-1$, $c_{k=0,1}=1$.

$$\frac{\partial IoU_{gt}(\mathbf{b})}{\partial z_{k=0,2}}=\frac{y_{max}-y_{min}}{d_{gt}(\mathbf{b})^2}\times\begin{cases}-c_k(z_3-z_1)(x_{max}-x_{min}) & \text{if } c_kz_k\leq c_kz_k^{gt}\\ -c_ka(\mathbf{b}_{gt})+(z_3-z_1)(x_{max}-z_2+z_0-x_{min})\end{cases} \tag{11}$$

$$\frac{\partial IoU_{gt}(\mathbf{b})}{\partial z_{k=1,3}}=\frac{x_{max}-x_{min}}{d_{gt}(\mathbf{b})^2}\times\begin{cases}-c_k(z_2-z_0)(y_{max}-y_{min}) & \text{if } c_kz_k\leq c_kz_k^{gt}\\ -c_ka(\mathbf{b}_{gt})+(z_2-z_0)(y_{max}-z_3+z_1-y_{min})\end{cases} \tag{12}$$

The IoU function has the major advantage of having independent variations among its variables. This specificity allows us to optimize the IoU by coordinates and deduce the global optima of the interval extension function. Every variation of IoU is depicted in Fig. 5. The + sign indicates that the function is increasing over the interval, independently of the other coordinates. Conversely, the − sign indicates decreasing parts of the function along specific coordinates.

z	$-\infty$		z_0^{gt}		z_2^{gt}		∞
$\frac{\partial IoU}{\partial z_0}$		$+$	$\dot{0}$	$-$		$-$	
$\frac{\partial IoU}{\partial z_2}$		$+$		$+$	$\dot{0}$	$-$	

z	$-\infty$		z_1^{gt}		z_3^{gt}		∞
$\frac{\partial IoU}{\partial z_1}$		$+$	$\dot{0}$	$-$		$-$	
$\frac{\partial IoU}{\partial z_3}$		$+$		$+$	$\dot{0}$	$-$	

Figure 5: Variation of the function IoU given its partial derivatives

For a fixed ground truth box, the IoU is increasing when the input variables get closer to the ground truth coordinates $\mathbf{b}_{gt}=[z_i^{gt}]_{i=0}^3$. Hence, computing the optimal box coordinates $\mathbf{b}_{\mathbf{u}}^*$ in terms of IoU is

immediate, with the rule:

$$\overline{IoU}_{opt} = \max_{\mathbf{b}\in[\underline{\mathbf{b}},\overline{\mathbf{b}}]} IoU_{gt} = IoU(\mathbf{b}^*_{\mathbf{u}} = [z_i^*]_{i=0}^3 \cap \mathbf{b}_{gt} = [z_i^{gt}]_{i=0}^3)$$
$$\text{with } z_i^* = \begin{cases} z_i^{gt} & \text{if } z_i^{gt} \in [\underline{\mathbf{b}}_i, \overline{\mathbf{b}}_i] \\ \underline{\mathbf{b}}_i & \text{if } z_i^{gt} \leq \underline{\mathbf{b}}_i \\ \overline{\mathbf{b}}_i & \text{else} \end{cases} \tag{13}$$

The interval extension of the box domain $\mathcal{B}$, is naturally defined as the joint product of the interval extension for each coordinate of a box. However, this definition can lead to ill defined boxes, as the coordinates of the upper right corner may be lower than those of the bottom left corner. Those corner cases happen whenever $\underline{z}_2 \leq \overline{z}_0$ or $\underline{z}_3 \leq \overline{z}_1$. Corner cases create an infinite number of collapsed bounding boxes $\{[z_0, z_1, z_0, z_3]\} \cup \{[z_0, z_1, z_2, z_1]\}$ whose IoU_{gt} saturates to 0. When it comes to the lowest IoU than can be reached in the interval, since the IoU is decreasing when an input variable is getting away from the ground truth coordinates, we know that the box with the lowest IoU is one of the vertices of the input domain of the IoU.

$$\underline{IoU}_{opt} = \min_{\mathbf{b}\in[\underline{\mathbf{b}},\overline{\mathbf{b}}]} IoU_{gt} = \begin{cases} 0 & \text{if } \underline{z}_2 \leq \overline{z}_0 \text{ or } \underline{z}_3 \leq \overline{z}_1 \\ \min\limits_{b\in\{\underline{b},\overline{b}\}} IoU_{gt}(\mathbf{b}) & \text{otherwise} \end{cases} \tag{14}$$

4 Evaluation

The experiments were parallelized to a certain extend over a pool of 20 workers, on a Linux machine with Intel® Xeon® processor E5-2660 v3 @ 2.60GHz of 20 cores and 64 GB RAM. For Step 1 of our approach Fig. 2, the perturbed bounding boxes are obtained through the Auto-LiRPA verification tool [XSZ+20]. Within Auto-LiRPA, we consider three verification methods namely IBP [GDS+18b], CROWN-IBP [ZCX+19], and CROWN [ZWC+18]. For Step 2, we have implementation of the two IoU extensions (Vanilla_IoU and Optimal_IoU) in python.

The purpose is to assess property 1 and for that, we consider two datasets, CNN-based object detection models, various perturbation domains and different verification methods. Our code is released on github[2].

4.1 Datasets & Networks

We explore two object detection use cases:

- **MNIST**: We examine the localization of handwritten digit extracted from the renowned grey-scale MNIST dataset [Den12]. The original grayscaled images of size 28×28, are randomly placed on black background images with a size of 90×90. The coordinates of the ground truth box correspond to the position of the original image. The ground truth box has a fixed size on all images. Our object detection network is a CNN, which description is provided in Fig. 6. It outputs 4 values that predicts the four coordinates. The analysis is conducted on 40 images extracted from the training set, which is uniformly composed of images representing digits from 0 to 9.
- **LARD**: We consider the LARD [DCF+23] industrial dataset that comprises high-quality aerial images for the task of runway detection during approach and landing phases. We selected 40 synthetic images from Reykjavík Domestic airport taken into clear weather conditions within a distance range of 0.33 to 1.08 nautical miles (NM) from the runway. The experiments are conducted on resized RGB images of size 256×256, with the ground truth' sizes ranging between 70 and 706 pixels. Our object detection networks is a CNN provided in Fig. 6. This second use case is more challenging for stability verification due to varying ground truth box sizes.

4.2 Perturbations

The plausible perturbations considered in this work are random noise and varying light conditions, leading to alterations in contrast and brightness. We consider three type of local perturbations Ω around an initial input image x_0. The values considered in our experiments are summarized in Fig. 7.

[2] https://github.com/NoCohen66/Verification4ObjectDetection

MNIST CNN	LARD CNN
Conv 16 3×3/1/1 - ReLU	Conv 32 3×3/2/1 - ReLU
Pool 2×2/2 - ReLU	Conv 64 3×3/2/1 - ReLU
Conv 16 3×3/1/1 - ReLU	Conv 128 3×3/2/1 - ReLU
Pool 2×2/2 - ReLU	Flatten
Flatten	Linear 128 - ReLU
Linear 256 - ReLU	Linear 128 - ReLU
Linear 4	Linear 4

Conv c h×w/s/p corresponds to a 2D-convolution with c output channels, h×w kernel size, stride s in both dimensions, padding p. Pooling layers are specified analogously

Figure 6: Overview of network architectures.

Perturbation	Ω	MNIST CNN			LARD CNN		
		min	max	step	min	max	step
White noise ϵ	$\{x \in \mathbb{R}^n \mid \|\|x - x_0\|\|_\infty \leq \epsilon\}$	0	0.002	11	0	0.002	11
Brightness α_b	$\{x \in \mathbb{R}^n \mid x = x_0 + \alpha_b\}$	0	0.002	11	0	0.02	11
Contrast α_c	$\{x \in \mathbb{R}^n \mid x = x_0 \times \alpha_c\}$	0	0.2	11	0	0.1	11

Figure 7: Perturbations types and parameters

4.3 Assessment of the *IoU* Interval Extensions

In order to compare Optimal_*IoU* and Vanilla_*IoU*, we consider the *Certified Box Accuracy* (**CBA**) metric that measures the number of bounding boxes that fulfill the stability property with a threshold $t = 0.5$. The results are compiled in Fig. 8. For MNIST and a white noise of $\epsilon = 2 \times 10^{-4}$, 76.9% of the bounding boxes are stable when considering CROWN-IBP+Vanilla_*IoU* while 97.4% of the bounding boxes are stable when considering CROWN+Vanilla_*IoU*. This means that the tightness of CROWN vs CROWN-IBP in Step 1 has a strong importance. For the same perturbation, Optimal_*IoU* whether with CROWN-IBP or CROWN proves that all bounding boxes are in fact stable. Using IBP in Step 1 always fails to find any stable box in Step 2.

Interval Extension		MNIST (%)			LARD (%)		
White noise $\epsilon =$		2×10^{-4}	4×10^{-4}	6×10^{-4}	0.0004	0.0006	0.0008
IoU_v	CROWN-IBP	76.9	0.0	0.0	0.0	0.0	0.0
	CROWN	97.4	0.0	0.0	25.0	2.8	0.0
IoU_{opt}	CROWN-IBP	100	35.9	2.6	8.3	0.0	0.0
	CROWN	100.0	66.6	7.70	97.2	75.0	27.8
Brightness $\alpha_b =$		2×10^{-4}	4×10^{-4}	0.001	0.004	0.006	0.008
IoU_v	CROWN-IBP	87.2	0.0	0.0	0.0	0.0	0.0
	CROWN	100.0	100.0	17.9	77.8	38.9	11.1
IoU_{opt}	CROWN-IBP	100.0	35.0	0.0	0.0	0.0	0.0
	CROWN	100.0	100.0	92.3	94.4	86.1	66.7
Contrast $\alpha_c =$		2×10^{-4}	0.001	0.0014	0.01	0.02	0.03
IoU_v	CROWN-IBP	31.4	0.0	0.0	0.0	0.0	0.0
	CROWN	100	8.6	0.0	69.1	32.0	13.4
IoU_{opt}	CROWN-IBP	82.9	0.0	0.0	0.0	0.0	0.0
	CROWN	100	88.6	8.6	82.5	58.8	38.1

Figure 8: Perturbation results on MNIST and LARD datasets

Optimal_*IoU* extension outperforms the more naive baseline version and this is particularly observable for LARD where Vanilla_*IoU* can hardly find stable boxes. We highlight this lack of tightness of Vanilla_*IoU* Fig. 9, where the envelope of mean values for the optimal extension (in red) consistently remains visibly higher than that of the baseline bounds (in blue) for all considered perturbation values. This difference translates to a *false positive rate* ranging from 2.6% to 80% for the standard threshold (t=0.5), when considering CROWN method.

Fig. 10 shows the computation time for computing the bounds on *IoU*. Despite a higher computation

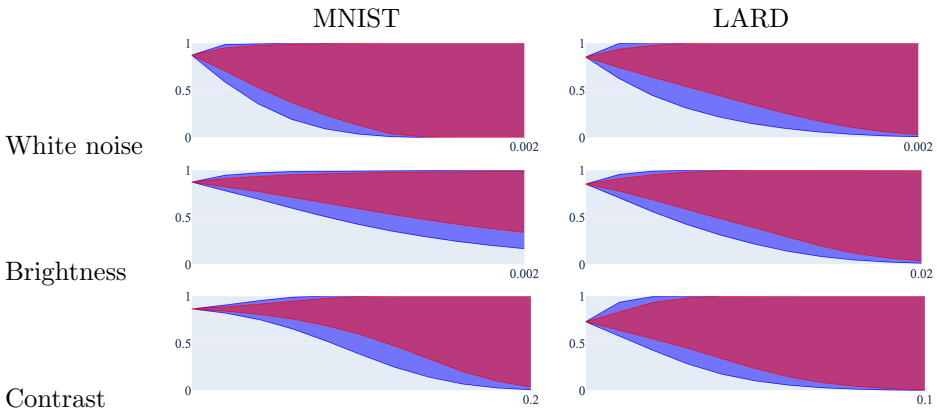


Figure 9: Average IoU bounds across all perturbations and dataset

time of the Optimal_IoU compared to Vanilla_IoU, this computation time is negligible compared to the one of Step 1 (with CROWN-IBP or CROWN).

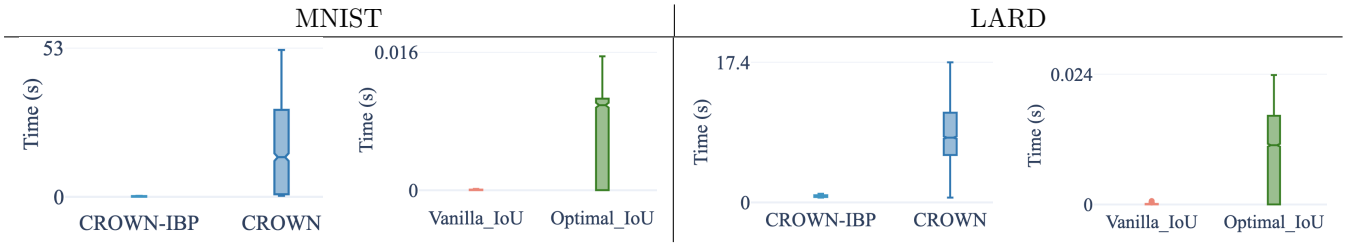
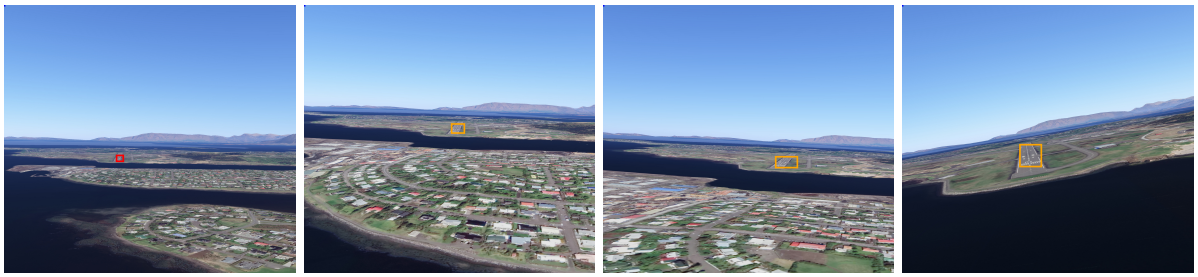


Figure 10: Step 1 and Step 2 computation times

4.4 LARD case study: landing approach runway detection

In this section, we focus deeper on the landing use case. Our experiments show that the distribution of instabilities throughout a trajectory is not uniform. Fig. 11 selects some images during an approach.



(a) 3.15km/1.70NM (b) 2.0km/1.08NM (c) 1.63km/0.88NM (d) 0.78km/0.42NM

Figure 11: Impact of a brightness perturbation $\alpha_b = 0.002$ on a trajectory computed with CROWN method: orange boxes indicate a slight impact while the red box signifies a strong one (with a minimal value of $\underline{IoU}_{opt}$ of 0.34)

We consider brightness perturbation with $\alpha_b = 0.002$. Our verification approach highlights the instability of the toy model when the aircraft is too far away. This design flaw could be addressed either by implementing a safety net, as suggested by [GDG22], or by refining the design through certified learning. It's noteworthy that our extension, with its significantly reduced computational cost, seamlessly integrates into certified training, allowing to balance between a good IoU on the training samples while guaranteeing Property 1. Future work will consider this type of certified training for object detection.

Let us now go further and consider the pose estimation system as a whole. The expected behavior is to always safely estimate the pose. Daedalean in [BFBC+21] proposed a two-stage machine learning

pipeline for this task. In effect, the first stage is an object detection in charge of detecting the runway. The input of the second stage is a cropped image around the bounding box of the input image. The second stage then computes the four corners of the runway. Our work so far only concerns the first stage. The authors of [KLE+23] introduce a ML-based system for verifying the semantic segmentation of neural network for estimating the aircraft pose during landing. In that sense, they focus on the second stage of the pipeline of Daedalean. They employ formal verification to assess U-Nets [RFB15] against white-noise, brightness, and contrast properties. Thus their work is complementary to ours and we can imagine combine our two approaches to assess the verification of the full pose estimation task. Fig. 12 proposes a naive combination of both results. For white noise, Stage 1 can accept a perturbation of 2×10^{-4} while Stage 1 can accept a perturbation of 10^{-8}. This means that we may accept $min(2 \times 10^{-4}, 10^{-8}) = 10^{-8}$ white noise perturbation.

	CBA	Stage 1 Optimal_IoU	Stage 2 [KLE+23]	Stage (1+2) together
White noise	100 %	$\mathbf{2 \times 10^{-4}}$	1×10^{-8}	1×10^{-8}
	0 %	$\mathbf{1 \times 10^{-3}}$	1×10^{-3}	1×10^{-3}
Brightness	100 %	$\mathbf{2 \times 10^{-3}}$	5×10^{-5}	5×10^{-5}
	90 %	$\mathbf{5 \times 10^{-3}}$	5×10^{-4}	5×10^{-4}
	80 %	$\mathbf{6 \times 10^{-3}}$	5×10^{-2}	6×10^{-3}
Contrast	100 %	5×10^{-3}	5×10^{-3}	5×10^{-3}
	90 %	6×10^{-3}	$\mathbf{5 \times 10^{-2}}$	6×10^{-3}

Figure 12: Combination of verification results

5 Conclusion

We presented IBP IoU, a novel Interval Bound Propagation approach for formal verification of object detection models. Our main contribution is the formalisation of non-linear single box stability property, which ensures the stability against local perturbations of the minimal bounding box containing the runway. Our key idea is to bound the challenging Intersection over Union function (Jaccard Index) which is multi-dimensional, non convex/concave and without an inherent property of partial monotony. To enable this, we propagate the perturbation intervals through the IoU function w.r.t to two schemes : (1) bounding the primitive operators, (2) applying interval extension on IoU function (Optimal_IoU). Our experimental evaluation shows the overall benefit of (Optimal_IoU) on an industrial usecase.

As future work, we will continue addressing formal verification of object detection by considering more operators such as Non-maximum Suppression (NMS) and more classical object detection models such as YoLo. These formal methods aim at contributing to the certification in general. Thus, we also would like to define the system expected properties. In particular, looking at the approach of Fig. 1, an open question is: at which distance do we expect the object detection to be stable? and with which threshold? As we already mentioned in the experiments section, we would also like to dig the open question of certified training for object detection.

6 Acknowledgement

This work has partly benefited from the AI Interdisciplinary Institute ANITI, which is funded by the French “Investing for the Future – PIA3” program under the Grant agreement ANR-19-P3IA-0004.

References

[BBS+19] Mislav Balunovic, Maximilian Baader, Gagandeep Singh, Timon Gehr, and Martin Vechev. Certifying geometric robustness of neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[BFBC+21] Giovanni Balduzzi, Martino Ferrari Bravo, Anna Chernova, Calin Cruceru, Luuk van Dijk, Peter de Lange, Juan Jerez, Nathanaël Koehler, Mathias Koerner, Corentin Perret-Gentil,

et al. Neural network based runway landing guidance for general aviation autoland. Technical report, United States. Department of Transportation. Federal Aviation Administration ..., 2021.

[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.

[Den12] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[Duc] Melanie Ducoffe. Decomon: Automatic certified perturbation analysis of neural networks. https://github.com/airbus/decomon. 2023-11-20.

[FB81] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[GDDM14] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[GDG22] Joris Guérin, Kevin Delmas, and Jérémie Guiochet. Evaluation of runtime monitoring for UAV emergency landing. In *2022 International Conference on Robotics and Automation, ICRA 2022, Philadelphia, PA, USA, May 23-27, 2022*, pages 9703–9709. IEEE, 2022.

[GDS+18a] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

[GDS+18b] Sven Gowal, Krishnamurthy Dvijotham, Robert Stanforth, Rudy Bunel, Chongli Qin, Jonathan Uesato, Relja Arandjelovic, Timothy A Mann, and Pushmeet Kohli. On the effectiveness of interval bound propagation for training verifiably robust models. arxiv 2018. *arXiv preprint arXiv:1810.12715*, 2018.

[GMDC+18] 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.

[HJVE01] Timothy Hickey, Qun Ju, and Maarten H Van Emden. Interval arithmetic: From principles to implementation. *Journal of the ACM (JACM)*, 48(5):1038–1068, 2001.

[JEL+22] Peiyuan Jiang, Daji Ergu, Fangyao Liu, Ying Cai, and Bo Ma. A review of yolo algorithm developments. *Procedia Computer Science*, 199:1066–1073, 2022.

[KLE+23] Panagiotis Kouvaros, Francesco Leofante, Blake Edwards, Calvin Chung, Dragos Margineantu, and Alessio Lomuscio. Verification of semantic key point detection for aircraft pose estimation. In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning*, volume 19, pages 757–762, 2023.

[MGV18] Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pages 3578–3586. PMLR, 2018.

[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.

[RFB15] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[RTG+19] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019.

[S+09] Teruo Sunaga et al. Theory of an interval algebra and its application to numerical analysis. *Japan Journal of Industrial and Applied Mathematics*, 26(2):125, 2009.

[TSCH19] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019.

[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.

[ZCX+19] Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. *arXiv preprint arXiv:1906.06316*, 2019.

[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.