

PoleXAI: An Explainable Neuro-Symbolic Framework for Visual Damage Assessment of Telecommunication Poles

Thai Le[1], Thanh Ma[1(✉)], Phu-An Thai[1], Quoc-Khang Tran[1], Cang Dang[2], Long Huynh[2], Bui Xuan Tung[3], and Umberto Straccia[4]

[1] Can Tho University, Can Tho, Vietnam
{lnthai,mtthanh,tqkhang}@ctu.edu.vn
[2] Vietnam Posts and Telecommunications Group, Can Tho City, Vietnam
[3] Tay Do University, Can Tho City, Vietnam
bxtung@tdu.edu.vn
[4] CNR-ISTI, Pisa, Italy
umberto.straccia@isti.cnr.it

Abstract. Visual assessment of telecommunication poles requires detecting surface defects, quantifying their spatial extent relative to the visible pole structure, and translating visual evidence into actionable maintenance guidance in a form that field operators can verify. Purely data-driven approaches address detection well but provide no mechanism for inspecting the reasoning that connects a pixel-level observation to a maintenance decision. We propose PoleXAI, a neuro-symbolic framework that couples deep visual perception with explicit symbolic reasoning to produce an auditable inspection output. Instance segmentation localizes visible damage regions; a lightweight convolutional classifier distinguishes four degradation types; a pole-relative quantification step measures the affected fraction of the visible pole area; and a defect-specific fuzzy inference stage maps the resulting percentage to a continuous severity score, which a constraint-based layer then resolves to a discrete severity level and a maintenance recommendation. Every intermediate value, from damage and pole masks to fuzzy score and severity, is exposed to the operator rather than collapsed into a single opaque label. Evaluated on a field dataset of Mekong Delta infrastructure images, the YOLO11l segmentation model achieves a Mask mAP50 of 0.408, and a MobileNet classifier attains 98.53% aggregate accuracy on four damage categories. The results demonstrate that integrating deep visual perception with explicit symbolic rules yields an inspection system whose decisions are both accurate and traceable.

Keywords: Telecommunication pole inspection · Visual damage assessment · Instance segmentation · Neuro-symbolic AI · Fuzzy reasoning · Explainable decision support

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2027
N. Thai-Nghe et al. (Eds.): ISDS 2026, CCIS 3253, pp. 370–385, 2027.
https://doi.org/10.1007/978-981-92-7408-6_25

1 Introduction

Hundreds of thousands of steel telecommunication poles in Southeast Asia face tropical corrosion, coating failure, and cracking, risking collapse, service interruption, and maintenance costs [15,26]. VNPT technicians photograph defects, prepare reports, and escalate maintenance decisions to engineers; these manual surveys are labour-intensive, inspector-dependent, and difficult to scale across the Mekong Delta.

CNN-based inspection of bridges and pipelines [1,3,13] can approach or exceed human corrosion assessment on held-out data [2,4]. However, boxes, masks, and labels alone neither establish pole-relative severity nor explain warranted maintenance actions. Operators need this connection to justify decisions under asset-management policies and avoid premature or delayed intervention.

Neuro-symbolic methods combine neural evidence extraction with inspectable domain rules and constraints [9,12,14,18,21]. Overlapping fuzzy memberships represent gradual degradation around ISO 4628-informed thresholds[1], while constraints connect assessment to discrete maintenance guidance.

PoleXAI integrates damage segmentation, four-class recognition, visible-pole-relative quantification, and defect-specific fuzzy and constraint reasoning. Its web interface exposes masks, percentages, fuzzy scores, and severities so operators can inspect the complete decision path.

This paper makes the following contributions.

1. A visible-pole-relative damage ratio invariant to uniform image scaling.
2. Independent ISNet [10] and SAM2 [20] pole-segmentation routes whose area-estimate differences are exposed to operators.
3. Defect-specific, ISO-4628-informed triangular fuzzy memberships for rust, peeling paint, and cracks, coupled with constraint-based severity and recorded maintenance recommendations.
4. An inspectable end-to-end system evaluated on Mekong Delta field data, with quantitative perception-component evaluation and interface-based demonstration of downstream reasoning.

The remainder of the paper is organized as follows. Section 2 establishes the theoretical foundations. Section 3 describes the PoleXAI framework in detail. Section 4 presents the experimental setup and results. Section 5 discusses findings and limitations, and Sect. 6 concludes.

2 Background

2.1 Instance Segmentation for Damage Detection

Instance segmentation adds per-instance pixel masks to detection boxes and labels [19], providing the spatial support for damage-area ratios without assumptions about within-box damage distribution.

[1] https://www.iso.org/standard/83324.html.

Definition 1 (Damage Mask). *Let* Ω *denote the (finite) pixel domain of an image* I*. A damage mask* $M_d : \Omega \to \{0, 1\}$ *assigns* $M_d(x) = 1$ *to pixel* x *if that pixel belongs to a detected damage region and* $M_d(x) = 0$ *otherwise. The support of* M_d *is* $\mathrm{supp}(M_d) = \{x \in \Omega : M_d(x) = 1\}$.

YOLO combines a shared convolutional backbone with detection and mask heads to predict boxes, class scores, and prototype-mask coefficients in one pass [11]. We evaluate YOLOv8m, YOLOv9c [28], YOLO11s/YOLO11l, and YOLO12 [27]. Mask mAP50, measured at mask IoU 0.50, is primary because masks drive downstream quantification.

2.2 Fuzzy Sets and Inference

Fuzzy logic [29] extends binary membership to $[0, 1]$. Membership functions over a universe U define linguistic terms for graded conditions such as moderate corrosion or severe peeling.

Definition 2 (Triangular Membership Function). *A triangular membership function with parameters* $a \le b \le c$ *is defined over* $U \subseteq \mathbb{R}$ *as*

$$\mathrm{trimf}(u; a, b, c) = \max\left(0,\ \min\left(\frac{u-a}{b-a},\ \frac{c-u}{c-b}\right)\right), \tag{1}$$

where the convention $0/0 = 0$ *applies at degenerate endpoints. The function achieves its maximum value of* 1 *at* $u = b$ *and vanishes outside* (a, c).

Fuzzy inference propagates input-membership activations through rules and aggregates their output memberships. Centroid defuzzification gives the scalar severity score

$$z = \frac{\int_U u\, \mu_{\mathrm{agg}}(u)\, \mathrm{d}u}{\int_U \mu_{\mathrm{agg}}(u)\, \mathrm{d}u}, \tag{2}$$

provided the denominator is positive.

2.3 Constraint Programming

Constraint programming (CP) solves combinatorial and mixed problems by expressing requirements as explicit constraints over decision variables [21]. CP-SAT, the constraint solver provided by Google OR-Tools [23], supports integer and Boolean decision variables and propagates constraints to prune infeasible assignments. In PoleXAI, a single integer variable $s \in \{0, \ldots, 5\}$ represents the discrete severity level; interval constraints on the damage percentage p_d determine which assignment is feasible, yielding a crisp severity in $O(1)$ time while preserving an explicit and auditable mapping.

3 The PoleXAI Framework

PoleXAI organizes the inspection pipeline into five successive stages, illustrated in Fig. 1: visual damage localization, damage-type recognition, pole-relative damage quantification, defect-specific fuzzy severity assessment, and constraint-based maintenance recommendation. Each stage produces an intermediate value that is visible to the operator, so the complete decision path from raw pixels to recommended action can be traced and verified.

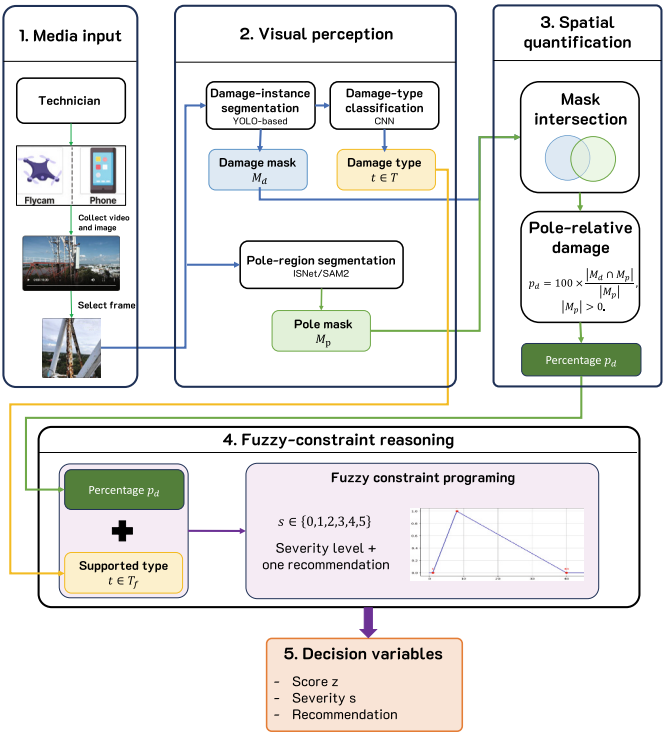

Fig. 1. Overview of PoleXAI. Five stages process an operator-acquired image or selected video frame: damage-mask prediction, damage-type classification, pole-relative damage percentage calculation, fuzzy inference, and constraint-based mapping to a severity level and maintenance recommendation. The web interface displays all intermediate values

3.1 Visual Damage Localization

The first stage applies YOLO-based instance segmentation [17] to the input image. For each detected damage instance, the model returns a bounding box, a class confidence score, and the binary damage mask M_d (Definition 1). The

mask retains the pixel-level shape of the damage region, which the quantification stage requires to compute an area ratio.

Formally, given input image I defined over domain Ω, the segmentation model produces a set of N instances $\{(M_d^{(i)}, c^{(i)}, \sigma^{(i)})\}_{i=1}^{N}$, where $M_d^{(i)} : \Omega \to \{0,1\}$ is the i-th damage mask, $c^{(i)} \in T = \{\text{rust, peeling paint, crack, rusted cable}\}$ is the predicted damage type, and $\sigma^{(i)} \in [0,1]$ is the detection confidence. In the current implementation the segmentation model uses the single class label `corrosion`, so the per-instance type $c^{(i)}$ is resolved by the dedicated classifier described in Sect. 3.2.

3.2 Damage-Type Recognition

Despite similar colour or texture anomalies, the four types in T require different maintenance, from monitoring surface rust to immediate intervention for load-bearing cracks. A lightweight CNN classifies each Stage 1 bounding-box crop to select the appropriate assessment family. We evaluate MobileNetV1 [7], MobileNetV2 [22], MobileNetV3 Small [6], VGG16 [24], ResNet50 [5], InceptionV3 [25], and DenseNet121 [8,16]. Aggregate four-class accuracy is primary; model size is secondary for edge or mobile deployment.

3.3 Pole-Relative Damage Quantification

Full-image normalization confounds damage extent with framing and camera distance. Normalizing by the visible pole area instead measures the affected fraction of the observed pole surface.

Definition 3 (Visible Pole Region). *The visible pole mask $M_p : \Omega \to \{0,1\}$ assigns $M_p(x) = 1$ to pixel x if that pixel belongs to the segmented pole region visible in the image and $M_p(x) = 0$ otherwise. The visible pole area is $A(M_p) = |\{x \in \Omega : M_p(x) = 1\}|$.*

Definition 4 (Relative Damage Extent). *Given a damage mask M_d and a visible pole mask M_p defined on the same domain Ω with $A(M_p) > 0$, the relative damage extent is*

$$r = \frac{A(M_d \cap M_p)}{A(M_p)}, \tag{3}$$

where $A(M_d \cap M_p) = |\{x \in \Omega : M_d(x) = 1 \wedge M_p(x) = 1\}|$ counts pixels assigned to both the damage region and the pole region. The percentage form is $p_d = 100\, r$.

The intersection $M_d \cap M_p$ excludes damage pixels that fall outside the returned pole mask, preventing background detections from inflating the numerator. The denominator $A(M_p)$ is the pole area rather than the total image area, so r measures what fraction of the visible pole surface is affected. By construction, $r \in [0,1]$ whenever $A(M_p) > 0$, and $p_d = 100\, r \in [0,100]$.

Equation (3) is undefined when $A(M_p) = 0$, a condition that arises if the pole segmentation produces an empty mask. In that case, PoleXAI reports no

damage estimate and suppresses the downstream fuzzy and constraint stages, flagging the empty-mask event to the operator.

The pole mask M_p is produced by two independent modules. ISNet applies a Dichotomous Image Segmentation model at 1024×1024 resolution to extract the salient foreground object (the pole) from the background. SAM2 [20] uses the full image as a bounding-box prompt to its segment-anything predictor. The two routes return independent estimates of $A(M_p)$, yielding two values of p_d that can be compared and both displayed to the operator.

Proposition 1 (Scale Invariance of r). *If M_d and M_p are uniformly scaled by a factor $\lambda > 0$, that is, $A(\lambda M_d \cap \lambda M_p) = \lambda^2 A(M_d \cap M_p)$ and $A(\lambda M_p) = \lambda^2 A(M_p)$, then r is invariant under the scaling: $r(\lambda M_d, \lambda M_p) = r(M_d, M_p)$.*

Proof. Direct substitution into Eq. (3): $r(\lambda M_d, \lambda M_p) = \lambda^2 A(M_d \cap M_p)/(\lambda^2 A(M_p)) = r(M_d, M_p)$.

Proposition 1 confirms that r is independent of image resolution and geometric scaling, so the same threshold values in the fuzzy stage apply regardless of whether the input was captured at close range or at a distance.

3.4 Defect-Specific Fuzzy Severity Assessment

Fuzzy inference lets p_d activate adjacent grades simultaneously, reflecting gradual degradation and ISO 4628's graded language: rust coverage of 4.9% and 5.1% should not imply an abrupt condition change.

Input Membership Functions. For each supported damage type $t \in T_f$, where $T_f = \{\text{rust, peeling paint, crack}\}$, the input variable p_d is partitioned into six linguistic grades $G_{t,0}, \ldots, G_{t,5}$ using triangular membership functions (Definition 2). The grade k membership function for type t is

$$\mu_{t,k}(p_d) = \text{trimf}(p_d;\, a_{t,k},\, b_{t,k},\, c_{t,k}), \tag{4}$$

with parameters $(a_{t,k}, b_{t,k}, c_{t,k})$ listed in Table 1. The notation $G_{t,k}$ corresponds to Ri_k, VN_k, and BT_k for t = rust, crack, and peeling paint respectively, following ISO 4628 area-fraction anchors adapted through expert consultation with VNPT field personnel. The damage type t = rusted cable is classified but has no documented fuzzy family in the current system; a detected rusted-cable region is reported without a fuzzy severity estimate.

Rule Base and Inference. The rule base contains six Mamdani rules, one per grade:

$$R_{t,k}: \quad \text{IF } p_d \text{ is } G_{t,k} \text{ THEN } Z \text{ is } H_k, \tag{5}$$

where H_k denotes the k-th output term. The output variable Z is defined over $[0, 100]$ and partitioned into six terms, *Excellent*, *Good*, *Fair*, *Poor*, *Severe*, and *Critical*, using automatic membership functions provided by the `scikit-fuzzy`

Table 1. Triangular input-membership parameters for the three supported defect-specific assessment families. Equal endpoint parameters produce shoulder or degenerate sets at the boundaries of the universe $[0, 100]$

k	Rust (Ri_k)	Crack (VN_k)	Peeling paint (BT_k)
0	(0, 0, 5)	(0, 0, 1)	(0, 0, 0)
1	(0, 5, 15)	(1, 5, 10)	(0, 0.5, 1)
2	(10, 20, 30)	(20, 35, 50)	(1, 2, 5)
3	(25, 40, 55)	(50, 75, 100)	(6, 10, 20)
4	(50, 70, 85)	(100, 200, 500)	(10, 20, 30)
5	(80, 100, 100)	(200, 1000, 1000)	(30, 50, 100)

library. The activation strength of rule $R_{t,k}$ is $\alpha_k = \mu_{t,k}(p_d)$, and the activated consequent is clipped at α_k. The aggregated output function is $\mu_{\text{agg}}(u) = \max_k \alpha_k \cdot \mu_{H_k}(u)$, and centroid defuzzification (Eq. (2)) yields the continuous severity score $z \in [0, 100]$ whenever at least one rule is activated and the centroid denominator is positive.

The activation domain for type t is

$$D_t = \left\{ p \in [0, 100] \;\middle|\; \max_k \mu_{t,k}(p) > 0 \right\}. \tag{6}$$

For $p_d \notin D_t$, no rule is activated and no fuzzy score is reported.

3.5 Constraint-Based Maintenance Recommendation

Let $\mathcal{M}$ denote the maintenance recommendation retrieved from Table 2 by severity index; its range is {No action, Monitor, Minor maintenance, Urgent maintenance, Repair or replace, Full replacement}. From p_d, two parallel reasoning paths are computed: the fuzzy path $p_d \to z$ (Sect. 3.4) and the constraint path $p_d \to s \to \mathcal{M}$ described below. Both outputs are exposed to the operator.

Definition 5 (Severity State). *The severity state $s \in \{0, 1, 2, 3, 4, 5\}$ is the unique integer satisfying the constraint*

$$s = \begin{cases} 0, & p_d < 5, \\ 1, & 5 \le p_d < 15, \\ 2, & 15 \le p_d < 30, \\ 3, & 30 \le p_d < 50, \\ 4, & 50 \le p_d < 70, \\ 5, & p_d \ge 70, \end{cases} \tag{7}$$

obtained by solving a CP-SAT model with one integer variable and interval constraints on p_d.

The six severity levels map to the maintenance recommendations listed in Table 2.

Table 2. Severity levels and corresponding maintenance recommendations

Severity s	Label	Maintenance recommendation
0	Excellent	No action required
1	Good	Monitor and record
2	Fair	Minor maintenance
3	Poor	Urgent maintenance
4	Severe	Repair or replace
5	Critical	Full replacement

3.6 Inspectable Inspection Workflow

After image upload, PoleXAI runs damage segmentation, both pole-mask routes, classification, and conditional assessment. It computes z and s when $A(M_p) > 0$, $t \in T_f$, and $p_d \in D_t$. The interface displays the original image, damage overlay, both pole masks, predicted type, branch-specific p_d, z, s, and $\mathcal{M}$. Stage labels let operators trace recommendations to masks without source-code inspection. Uploaded videos are sampled and processed identically, with overlays and assessment fields for each selected frame.

4 Experimental Results

4.1 Experimental Environment

Table 3 distinguishes Colab training and component evaluation from local application execution and runtime measurement. Reported timings apply to the recorded local hardware. The segmentation stack used Ultralytics 8.3.199 and PyTorch 2.8.0; the classification experiments used TensorFlow 2.19.0 and Keras 3.10.0. The decision-support and application stages used Scikit-fuzzy 0.5.0, OR-Tools 9.14.6206, Flask 3.1.2, OpenCV-python 4.12.0.88, NumPy 1.26.4, and Scikit-image 0.25.2.

Table 3. Recorded experimental environments and their roles

Environment	Hardware	Role
Google Colab	4 vCPU Intel Xeon E2-8276 at 2.50 GHz; 16 GB RAM; one Tesla T4 GPU with 16 GB HBM	Training and evaluation of the recorded deep-learning configurations
Local application	Dell Inspiron N5502; Intel Core i7-1165G7; 8 GB RAM; 512 GB SSD; Windows 11; Python 3.11.9	Integrated application execution and collection of the reported runtime observations

4.2 Dataset Preparation

Segmentation Corpus. VNPT An Giang field images, Mekong Delta pole photographs, and selected Internet images formed a 736-image corpus. Horizontal/vertical flips, 90° clockwise/counter-clockwise and 180° rotations, saturation (±25%), and brightness (±15%) augmentation expanded it to 1,764 images. Roboflow polygon masks labelled `corrosion` covered over 5,000 instances (4–10 per image). The train/validation/test split was 70%/10%/20%.

Classification Corpus. The separate corpus contained 337 rust, 335 peeling-paint, 300 crack, and 380 rusted-cable images (1,352 total). Rotation, shift, zoom, and flip augmentation increased these counts to 812, 823, 842, and 805, respectively (3,282 total). Allocation was 70%/15%/15% for training/validation/test; validation and test images were normalized without spatial augmentation. Results therefore assess aggregate four-class recognition, without establishing end-to-end application accuracy.

4.3 RQ1: Damage-Instance Segmentation

Table 4 reports all 30 configurations using Box and Mask mAP50 (IoU 0.50) and mAP50–95 (averaged over progressively stricter IoU thresholds). Mask mAP50 is the primary integration criterion because masks support Eq. (3).

The leaders differ by metric: YOLO11s (batch 8, 40 epochs) for Box mAP50 (0.431); YOLO11l (16, 50) for Box mAP50–95 (0.229); YOLO11l (8, 40) for Mask mAP50 (0.408); and YOLO11l (18, 22) for Mask mAP50–95 (0.195). YOLOv9c (8, 70) approaches the best box scores but trails on masks. YOLO12 remains below the best values; greater data or training requirements for its attention-centric architecture [27] are a possible explanation, untested by controlled ablation or repeated runs.

Answer to RQ1. The integrated YOLO11l (batch 8, 40 epochs) combines the highest Mask mAP50 with near-best remaining metrics: 0.423/0.225/0.408/0.191 in table-column order. This supports a balanced component choice, without establishing statistical superiority or a universal winner.

4.4 RQ2: Aggregate Four-Class Classification

Table 5 reports seven architectures under three settings on the separate classification corpus. Panel (a) defines settings and panel (b) gives aggregate accuracy, precision, recall, and F1-score. The undocumented averaging convention precludes macro, micro, weighted, or per-class interpretation.

MobileNet under S3 (input 128×128×3, batch 8, patience 15, epoch 35/80) records loss 0.1174 and 98.53% on all four measures, with the smallest model (24.69 MB; other models: 40.35–188 MB). DenseNet201 leads S1 and DenseNet121 leads S2, both at 97.07% accuracy. Resolution, batch size, and patience vary jointly, preventing attribution to individual settings. Persistently high ResNet50/101 losses suggest underfitting; absent repeated-run uncertainty and per-class results limit broader claims.

Table 4. All 30 recorded YOLO segmentation configurations. Bold denotes the best value per column across all configurations

Model	Batch	Ep.	Box mAP50	Box mAP50–95	Mask mAP50	Mask mAP50–95
YOLO11s	16	30	0.410	0.220	0.399	0.182
YOLO11s	16	50	0.430	0.223	0.404	0.187
YOLO11s	32	70	0.421	0.224	0.391	0.187
YOLO11s	32	22	0.405	0.216	0.389	0.182
YOLO11s	8	40	**0.431**	0.227	0.392	0.187
YOLO11s	8	20	0.383	0.199	0.385	0.166
YOLO11l	16	30	0.427	0.222	0.407	0.189
YOLO11l	16	50	0.426	**0.229**	0.404	0.192
YOLO11l	12	70	0.428	0.227	0.405	0.189
YOLO11l	18	22	0.421	0.226	0.405	**0.195**
YOLO11l	8	40	0.423	0.225	**0.408**	0.191
YOLO11l	8	20	0.385	0.200	0.371	0.169
YOLOv8m	16	30	0.406	0.223	0.391	0.185
YOLOv8m	16	50	0.403	0.213	0.385	0.175
YOLOv8m	32	70	0.402	0.209	0.385	0.174
YOLOv8m	32	22	0.418	0.224	0.402	0.187
YOLOv8m	8	40	0.402	0.222	0.389	0.187
YOLOv8m	8	20	0.407	0.212	0.382	0.179
YOLOv9c	16	30	0.412	0.209	0.391	0.182
YOLOv9c	16	50	0.404	0.213	0.390	0.178
YOLOv9c	8	70	0.429	0.227	0.393	0.188
YOLOv9c	20	22	0.390	0.204	0.361	0.171
YOLOv9c	8	40	0.417	0.213	0.396	0.182
YOLOv9c	12	100	0.398	0.207	0.395	0.184
YOLO12	16	30	0.184	0.085	0.167	0.076
YOLO12	16	120	0.326	0.167	0.321	0.144
YOLO12	32	80	0.315	0.156	0.300	0.135
YOLO12	42	100	0.317	0.151	0.302	0.133
YOLO12	8	60	0.263	0.127	0.238	0.109
YOLO12	8	130	0.341	0.169	0.319	0.145

Answer to RQ2. Across the complete recorded comparison, MobileNet under S3 provides the most favourable observed aggregate performance–model-size trade-off: 98.53% on all four reported measures at 24.69 MB. This answer is bounded to the separate classification corpus and the three evaluated settings.

4.5 Application Demonstration and Runtime Observations

PoleXAI accepts operator-acquired images and videos from phones, cameras, or camera-equipped UAVs and displays original media and intermediate assessment evidence. Figure 2 shows YOLO damage masking and alternative ISNet/SAM2 pole masks; intersecting each with the damage mask yields branch-specific p_d. Without annotated pole-mask ground truth, the example demonstrates implemented processing, not pole-mask accuracy.

Figure 3 shows the corresponding colour-coded overlays and assessment outputs. Red denotes the YOLO damage region; blue denotes the pole region returned by ISNet or SAM2. These overlays make it possible to inspect the

Table 5. Complete recorded aggregate four-class classification results. Panel (a) defines the settings used in panel (b)

(a) Recorded settings

Setting	Input	Batch	Patience
S1	224×224×3	32	5
S2	128×128×3	16	10
S3	128×128×3	8	15

(b) Recorded results

Set.	Model	Epoch	Loss	Acc. (%)	Prec. (%)	Rec. (%)	F1 (%)	Size (MB)
S1	MobileNet	9/80	0.1386	95.60	95.64	95.60	95.59	24.69
S1	DenseNet201	18/80	0.1525	97.07	97.14	97.07	97.06	94.83
S1	DenseNet121	14/80	0.1230	95.60	95.69	95.60	95.59	40.35
S1	VGG16	23/80	0.2450	89.75	90.17	89.75	89.81	62.28
S1	InceptionV3	22/80	0.1814	93.17	93.97	93.17	93.20	108.28
S1	ResNet50	32/80	0.6879	70.73	70.51	70.73	69.23	114.66
S1	ResNet101	9/80	0.8223	62.43	69.95	62.43	58.29	188.00
S2	MobileNet	22/80	0.1149	96.09	96.50	96.09	96.08	24.69
S2	DenseNet201	17/80	0.1687	95.12	95.19	95.12	95.07	94.83
S2	DenseNet121	54/80	0.1422	97.07	97.12	97.07	97.05	40.35
S2	VGG16	61/80	0.1729	93.65	93.65	93.65	93.62	62.28
S2	InceptionV3	19/80	0.4377	85.85	87.09	85.85	85.79	108.28
S2	ResNet50	54/80	0.6954	67.31	66.88	67.31	66.83	114.66
S2	ResNet101	20/80	0.7364	63.90	64.15	63.90	61.67	188.00
S3	**MobileNet**	35/80	**0.1174**	**98.53**	**98.53**	**98.53**	**98.53**	**24.69**
S3	DenseNet201	29/80	0.1179	96.09	96.42	96.09	96.10	94.83
S3	DenseNet121	54/80	0.1853	95.60	95.70	95.60	95.58	40.35
S3	VGG16	78/80	0.1810	93.17	93.37	93.17	93.21	62.28
S3	InceptionV3	61/80	0.3224	87.80	87.94	87.80	87.74	108.28
S3	ResNet50	80/80	0.6972	70.73	70.08	70.73	70.15	114.66
S3	ResNet101	54/80	0.6432	68.78	68.16	68.78	68.34	188.00

spatial evidence used to derive p_d before interpreting the downstream assessment.

Video mode displays the original sequence, generated overlays, and selected-frame measurements; it provides frame-wise analysis without continuous tracking or autonomous inspection. Table 6 reports local execution observations. Stage times are not summed because a uniform timing boundary is undocumented; the observations are specific to the recorded implementation and hardware.

5 Discussion

YOLO11l's leading mask-level performance suggests that greater capacity benefits pixel-level prediction, whereas YOLO12's lower scores may

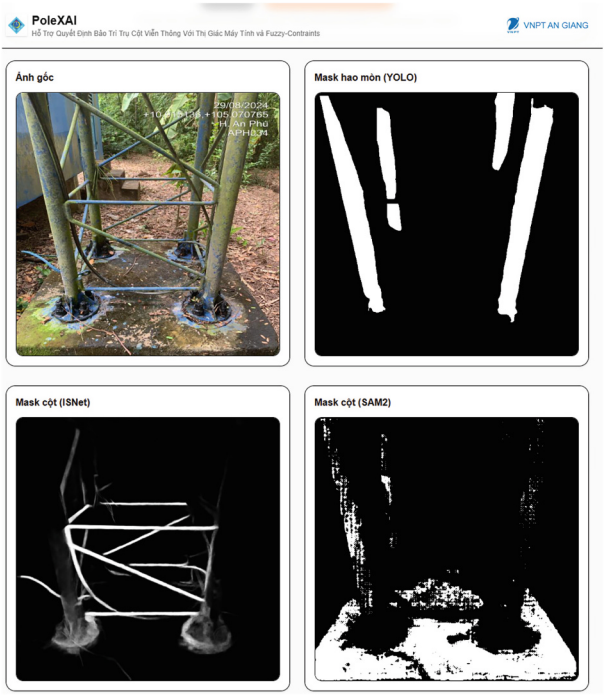

Fig. 2. Still-image analysis in PoleXAI, showing the uploaded field image, YOLO damage mask, and ISNet/SAM2 pole masks. This example illustrates implemented processing without quantitatively evaluating pole-mask accuracy

reflect the data requirements of attention-based models, pending controlled evaluation. MobileNet provides a favourable accuracy–size trade-off through depthwise-separable convolutions. Visible-pole-area normalization yields framing-independent damage fractions that are scale-invariant by Proposition 1, while dual ISNet/SAM2 estimates expose pole-mask-dependent variation.

Separate corpora preclude end-to-end evaluation under realistic crops, while single runs on one split provide no variance estimates (internal validity). Generalization beyond the limited Mekong Delta conditions remains untested (external validity). ISNet/SAM2 lack ground-truth benchmarking, and Mask mAP50 does not directly measure r, which also depends on mask shape and damage–pole overlap (construct validity). The fuzzy parameters lack independent structural validation, rusted cable has no documented fuzzy family, and sensitivity near the thresholds in Eq. (7) remains unknown.

Priority directions include: constructing jointly annotated end-to-end data linking segmentation outputs to classifier inputs and maintenance ground truth; quantitatively evaluating pole-mask accuracy; completing fuzzy routing for rusted cable; defining fail-safe behaviour for empty pole masks and out-of-domain



(a) ISNet–YOLO overlay (b) SAM2–YOLO overlay

(c) Branch-specific assessment outputs

Fig. 3. Inspectable PoleXAI outputs for the same field image. Panels (a) and (b) show the alternative pole-mask overlays. Panel (c) reports the pole-relative percentage p_d, fuzzy score z, constraint-derived severity s, and recommendation for each branch. In the recorded example the ISNet route reports $p_d = 27.23\%$, $z = 46.79$, $s = 3$, and an emergency–maintenance recommendation

inputs; calibrating membership parameters against expert records; and conducting controlled field studies with VNPT technicians to assess decision quality and usability.

Table 6. Recorded execution-time observations on the local application environment (Dell Inspiron N5502, Intel Core i7-1165G7, 8 GB RAM)

Recorded operation	Time (s)
Damage-instance segmentation	1.258
Four-class damage classification	0.384
Constraint reasoning, fuzzy inference, and calculation	0.875
Integrated system (mean of 10 runs)	2.583 ± 0.63

6 Conclusion

We presented PoleXAI, a neuro-symbolic framework combining deep visual perception with explicit symbolic reasoning for auditable telecommunication-pole damage assessment through five stages: instance segmentation, damage-type classification, pole-relative quantification, defect-specific fuzzy severity inference, and constraint-based maintenance mapping, with all intermediate outputs visible to operators. Its pole-relative ratio (Eq. (3)) is scale-invariant by Proposition 1, while ISNet/SAM2 expose independent pole-area estimates. On the Mekong Delta segmentation corpus, YOLO11l (batch 8, 40 epochs) achieves the highest Mask mAP50 (0.408) among evaluated configurations, while MobileNetV3 Small achieves 98.53% aggregate accuracy at 24.69 MB on the classification corpus, providing component-level evidence for the feasibility of interpretable neuro-symbolic pole inspection. Full end-to-end validation, pole-mask benchmarking, and field deployment remain priorities for future work.

Acknowledgements. This study is supported by the European Union's Horizon research and innovation program under the MSCA-SE (Marie Skłodowska-Curie Actions Staff Exchange) grant agreement 101086252; Call: HORIZON-MSCA-2021-SE-01; Project title: STARWARS (STormwAteR and WastewAteR networkS heterogeneous data AI-driven management).

References

1. Atha, D.J., Jahanshahi, M.R.: Evaluation of deep learning approaches based on convolutional neural networks for corrosion detection. Struct. Health Monit. **17**(5), 1110–1128 (2018)
2. Burton, B., Nash, W.T., Birbilis, N.: Rustseg-automated segmentation of corrosion using deep learning, (2022). arXiv:2205.05426 arXiv preprint
3. Cha, Y.J., Choi, W., Suh, G., Mahmoudkhani, S., Büyüköztürk, O.: Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput.-Aided Civil Infrastruct. Eng. **33**(9), 731–747 (2018)
4. Das, A., Dorafshan, S., Kaabouch, N.: Autonomous image-based corrosion detection in steel structures using deep learning. Sensors **24**(11), 3630 (2024)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
6. Howard, A., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 1314–1324 (2019)
7. Howard, A.G., et al.: MobileNets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017). https://doi.org/10.48550/arXiv.1704.04861
8. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4700–4708 (2017). https://doi.org/10.1109/CVPR.2017.243

9. Ibrahimi, M.A., Straccia, U.: A neuro-symbolic fuzzy rule-based approach for melanoma classification. Tech. rep., Research Square Repository (2026). https://doi.org/10.21203/rs.3.rs-10142900/v1
10. Jin, Z., Liu, B., Chu, Q., Yu, N.: Isnet: Integrate image-level and semantic-level context for semantic segmentation. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 7169–7178. IEEE (2021)
11. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLOv8 (2023). https://github.com/ultralytics/ultralytics, software, version 8.0.0
12. Kaiser, B., Caiado, R.G.G., Rios, M.P., Vignon, Y.R., Scavarda, L.F., Corseuil, E.T.: Multi-level fuzzy decision support framework for asset maintenance criticality: corrosion risk assessment in piping systems. Expert Syst. Appl. **299**, 130024 (2026)
13. Katsamenis, I., Doulamis, N., Doulamis, A., Protopapadakis, E., Voulodimos, A.: Simultaneous precise localization and classification of metal rust defects for robotic-driven maintenance and prefabrication using residual attention u-net. Autom. Constr. **137**, 104182 (2022)
14. Le, N.T., Ma, T., Straccia, U.: A fuzzy rule-based neuro-symbolic approach for pipe severity prediction in sewer networks. Tech. rep., Computing Research Repository (2026). https://doi.org/10.48550/arXiv.2607.28481
15. Lekidis, A., Anastasiadis, A.G., Vokas, G.A.: Electricity infrastructure inspection using ai and edge platform-based UAVs. Energy Rep. **8**, 1394–1411 (2022)
16. Ma, T., Benferhat, S., Tran-Nguyen, M.T., Do, T.N.: Aquanetct: Detecting and representing urban drainage networks from video analysis. In: Thai-Nghe, N., Do, T.N., Benferhat, S. (eds.) International Conference on Intelligent Systems and Data Science, pp. 31–45. Springer, Cham (2025). https://doi.org/10.1007/978-981-95-3358-9_3
17. Ma, T., Nguyen, H., Nguyen, X., Nguyen, K., Tran-Nguyen, M.T.: Vitip-ext: Ai-powered 3d visualization for vietnamese instrument knowledge preservation. SN Comput. Sci. **6**(7), 896 (2025)
18. Mendel, J.M.: Fuzzy logic systems for engineering: a tutorial. Proc. IEEE **83**(3), 345–377 (1995)
19. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **44**(7), 3523–3542 (2021)
20. Ravi, N., et al.: Sam 2: Segment anything in images and videos. In: International Conference on Learning Representations, vol. 2025, pp. 28085–28128 (2025)
21. Rossi, F., Van Beek, P., Walsh, T.: Constraint programming. In: Van Harmelen, F., Lifschitz, V., Porter, B. (eds.) Handbook of Knowledge Representation, Foundations of Artificial Intelligence, vol. 3, pp. 181–211. Elsevier (2008). https://doi.org/10.1016/S1574-6526(07)03004-0
22. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: Mobilenetv 2: Inverted residuals and linear bottlenecks. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520. IEEE (2018)
23. Sciau, J.B., Goyon, A., Sarazin, A., Bascans, J., Prud'homme, C., Lorca, X.: Using constraint programming to address the operational aircraft line maintenance scheduling problem. J. Air Transp. Manag. **115**, 102537 (2024)
24. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
25. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)

26. Taheri, P., Mansouri, A., Bachour, B., Ahuja, N., Zamanzadeh, M.: Inspection and mitigation of underground corrosion at anchor shafts of telecommunication towers. In: CORROSION 2017, pp. 1–15. Association for Materials Protection and Performance (2017)
27. Tian, Y., Ye, Q., Doermann, D.: YOLOv12: Attention-centric real-time object detectors. arXiv preprint arXiv:2502.12524 (2025). https://doi.org/10.48550/arXiv.2502.12524
28. Wang, C.Y., Yeh, I.H., Liao, H.Y.M.: YOLOv9: Learning what you want to learn using programmable gradient information. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision - ECCV 2024, pp. 1–21. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-72751-1_1
29. Zadeh, L.A.: Fuzzy sets. Inf. Control **8**(3), 338–353 (1965)