

ORIGINAL ARTICLE

Open Access



Diffusion-augmented YOLO26-Swin cascaded framework with hybrid SHAP-CAM for autonomous power grid inspection

Stefano Frizzo Stefenon^{1,2*} , João Pedro Matos-Carvalho^{3,4,5} , Viviana Cocco Mariani^{6,7} ,
Leandro dos Santos Coelho^{6,8} and Kin-Choong Yow²

Abstract

Deep learning-based autonomous inspection of power grid insulators is challenged by data imbalance and model opacity. This paper presents an end-to-end solution integrating advanced data synthesis, detection, classification, and explainability. First, a conditional diffusion model generates realistic synthetic fault images to balance the dataset. A two-stage architecture based on You Only Look Once version 26 (YOLO26) extra-large and Shifted windows (Swin)-V2-B, called YOLO26-Swin, fine-tuned with Bayesian optimization, performs robust insulator detection and then fault classification. Finally, a novel SHapley Additive exPlanations with Class Activation Mapping (SHAP-CAM) method provides intuitive visual explanations for model predictions. Extensive experiments validate our framework's superiority: it achieves an F1-score of 0.98149 and a mean Average Precision (mAP)@[0.5] of 0.98951, exceeding leading detection and classification models. This work highlights the efficacy of diffusion models for data augmentation in critical infrastructure and advances the interpretability of vision-based inspection systems.

Keywords: Bayesian optimization, Diffusion models, Generative artificial intelligence, Explainable artificial intelligence

1 Introduction

The electricity distribution power system has the function of supplying electricity to consumers continuously and stably [1]. To improve the reliability of the power grid, inspections are performed to detect broken insulators and flashovers, indicating that the insulation is compromised and the insulator should be replaced [2]. Inspections can be conducted from the ground or aerially, using aircraft photographs. The use of Unmanned Aerial Vehicles (UAVs) allows the inspection of the electricity network

in difficult-to-access locations, being a solution that is currently being used by electricity utilities [3].

The use of an image database to classify insulator defects allows the use of models based on Deep Learning (DL) [4], such as Convolutional Neural Networks (CNNs), which are increasingly being used in the field of computer vision [5]. Hybrid models are outstanding in these applications, and can be applied in an aggregated way, such as object detection and image classification, even outperforming state-of-the-art models. Despite the growing use of DL-based models, there are challenges in their application, such as unbalanced data or non-generalist images [6].

These challenges, particularly the small data constraint, mean that models become overfitted to specific conditions and cannot generalize effectively to classes with fewer samples. Although overfitting can be mitigated by using

* Correspondence: stefano.stefenon@isel.pt

¹Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Rua Conselheiro Emídio Navarro 1, 1599-007 Lisboa, Portugal

²Faculty of Engineering and Applied Sciences, University of Regina, Regina, Saskatchewan, S4S 0A2, Canada

Full list of author information is available at the end of the article

the early stop criterion, the models may not be generalizable to new samples [7]. The use of synthetic images can be a potential solution to this issue. Generative Adversarial Networks (GANs) stand out for generating synthetic images as a solution for data augmentation and mitigating the problem of unbalanced classes [8].

Diffusion models can be used to generate synthetic images for insulator data augmentation. Diffusion models may be more stable during training, reducing the likelihood of generating unrealistic images compared to GANs [9]. Unlike GANs, which can suffer from mode collapse, diffusion models provide better data distribution coverage, ensuring a more diverse set of generated images. Diffusion models can be guided during the generation process using classifier guidance, allowing controlled image synthesis based on specific attributes or descriptions [10].

Given the challenges of detecting faulty insulators, this paper proposes an optimized cascaded framework that detects and classifies insulators separately. For the detection of insulators, the You Only Look Once (YOLO) version 26 extra-large (YOLO26x) is adopted [11]. This model is tuned by Bayesian optimization based on Gaussian process to ensure adequate hyperparameter values. For the classification task, the dataset containing the bounding boxes is augmented with a conditional diffusion model. Considering this augmented dataset, the Shifted windows (Swin)-V2-B model is used for classification.

This paper has the following contributions:

- To ensure the best hyperparameter setup, the proposed approach employs Bayesian optimization for hyperparameter tuning, ensuring that the detection model operates at its best for insulator identification.
- The dataset augmentation with a conditional diffusion proved better than using a conditional Wasserstein GAN to create synthetic images or the use of only original images.
- The SHapley Additive exPlanations (SHAP) merged with the Class Activation Map (CAM), creating a hybrid structure to provide better explainable results. SHAP-CAM assists operators in decision-making regarding electrical power grid inspections.
- For the insulator detection, the proposed method (hypertuned YOLO26x) outperforms the DETection TRansformer (DETR), Faster Region-based CNN (Faster R-CNN), RetinaNet, Single-Shot Detector (SSD), YOLOv10 to YOLO26, and modified YOLO models (Spatial Transformer Networks YOLO (STN-YOLO) and Squeeze-and-Excitation YOLO (SAE-YOLO)).
- For the classification, the Swin-V2-B model overcomes the Visual Geometry Group (VGG), Residual neural Network (ResNet), EfficientNet, Vision Transformer (ViT), and other Swin models.

Table 1 List of acronyms

Acronym	Full form
Adam	Adaptive moment estimation
AdamW	Adam with Weight decay
CAM	Class Activation Map
CNN	Convolutional Neural Network
cGAN	Conditional GAN
DL	Deep Learning
DETR	DEtection TRansformer
GAN	Generative Adversarial Network
GenAI	Generative Artificial Intelligence
GFLOPs	Giga Floating-point Operations Per second
GLA	Global-Local Attention
GP	Gaussian Process
HiResCAM	High-resolution CAM
IoU	Intersection over Union
MCI	Multiscale Channel Information
mAP	Mean Average Precision
ML	Machine Learning
ReLU	Rectified Linear Unit
ResNet	Residual neural Network
R-CNN	Region-based CNN
RMSprop	Root Mean Square propagation
SAE-YOLO	Squeeze-and-Excitation YOLO
ScoreCAM	Score-weighted CAM
SGD	Stochastic Gradient Descent
SHAP	SHapley Additive exPlanations
SSD	Single Shot Detector
STN-YOLO	Spatial Transformer Networks YOLO
Swin	Hierarchical vision transformers using Shifted windows
UAV	Unmanned Aerial Vehicle
ViT	Vision Transformer
VGG	Visual Geometry Group
XAI	EXplainable Artificial Intelligence
XGradCAM	EXtended Gradient-weighted CAM
YOLO	You Only Look Once

The next sections of this work are organized as follows: Section 2 presents work related to fault identification using object detection methods to improve electrical system inspection. Section 3 explains the hybrid method proposed in this paper. In Section 4, the setup used to carry out the comparisons, the evaluation metrics, and the data set considered are presented. Section 5 focuses on presenting and discussing the results, comparing the proposed method with other state-of-the-art models. Finally, Section 6 presents this paper's conclusion and suggestions for future work. The acronyms used in this work are presented in Table 1.

2 Related works

This section presents an overview of previously conducted studies on insulator defect detection, continuous monitoring, and inspection of insulators for preventing failures in the electrical power grids using YOLO-based models and augmented datasets.

2.1 Overview of YOLO applications

Manual inspections have high operational costs, due to the large geographic areas with adverse weather conditions [12]. In this context, automated insulator detection from aerial images using YOLO serves as the foundational step toward real-time condition assessment and classification, enabling the deployment of UAVs [13]. With the increasing use of UAV systems, insulator defect detection using Machine Learning (ML) and deep neural networks has arisen as a promising and viable solution [14].

Progress has been made in using object detection methods in the power system sector during the last few years, focusing on insulator fault identification and monitoring [15], where YOLO has been frequently used for anomaly detection [16], especially to check the health of insulators. Several versions of YOLO have been modified over the years. The first version was proposed in 2015 by Redmon et al. [17], and Redmon et al. [18], called YOLO or YOLOv1, and currently, in 2025, there is YOLOv12 proposed by Tian et al. [19]. Unlike traditional methods that analyze an image several times in different regions, YOLO processes it in a single pass through the model, making it fast and efficient [20]. Currently, YOLO26 presented by Sapkota et al. [21] is available for similar object detection tasks.

Each version of the YOLO model includes multiple scaled variants, with suffixes indicating differences in model size and complexity [22]. Specifically, *n* denotes the nano version, *s* represents small, *m* corresponds to medium, *l* stands for large, and *x* represents extra-large. Additional suffixes include *t* for tiny, *c* for compact, *b* for balanced, and *e* for extended. These variations provide a range of trade-offs between classification performance and inference speed, enabling the choice of the most appropriate model based on computational constraints and application requirements [23].

Stefenon et al. [24] proposed a hybrid approach for the detection and classification of faulty insulators, integrating a customized YOLOv5u for insulator detection, evaluated against other YOLO variants, with the optimized Quasi-ProtoNet architecture for classification. The model is assessed using six different CNNs baseline architectures. Considering an F1-score equal to 0.95165, the optimized hybrid model using DenseNet-161 stood out in the comparisons to VGG-16, VGG-19, ResNet-34, ResNet-152, and DenseNet-121.

Using a synthetic dataset and an upgraded YOLOv5 model, Song et al. [25] proposed a fault detection method for transmission line components using the convolutional block attention module to raise detection performance. Comparative analysis shows that, in terms of performance, the proposed approach outperforms other commonly used object detection networks. Sadykova et al. [26] evaluated insulator images with the presence of ice, snow, and water using different YOLO models. Insulator fault detec-

tion using the coordinate attention mechanism and feature channel shuffle operation, YOLO was evaluated by Cao et al. [27].

By decreasing feature redundancy and increasing the inference speed of the feature extraction network, the ghost convolution based on YOLOv5s has been introduced to improve precision and recall in detecting insulator flaws [28]. The YOLO-S model was proposed by Yi et al. [29] and compared with YOLOv5s for insulator defect detection. In the proposed YOLO-S, the S-Intersection over Union (IoU) loss function replaced the original CIoU loss function. The Sigmoid-weighted Linear Unit (SiLU) function was replaced by Mish, HardSwish, and a rectified linear unit, among other replacements. The results were promising when compared to existing detection algorithms under the same conditions.

The cross-modality information attention YOLO model was used by Liu et al. [30] to enhance the insulator defect detection capabilities. A meteorological domain synthesis module was built to translate weather-conditioned insulator images into the real domain. Their proposed model outperformed the state-of-the-art UAV-based multi-domain insulator tasks. Stefenon et al. [31] using 240 inspection photographs of power distribution systems, proposed the hypertuned YOLO model, with hyperparameters optimized by a genetic algorithm, to classify insulating components in good condition, in need of repair, or in need of replacement. The performance results were promising considering the results of the eigenvector-based CAM. The eigenvector-based CAM was integrated into the model to provide an explainable framework [32].

Wang et al. [33] proposed a Multiscale Channel Information (MCI) with Global-Local Attention (GLA) for YOLO. While GLA captures local spatial details and global context information, MCI fully extracts and uses multiscale feature map information, improving the network's capacity for learning. According to experimental findings, the MCI-GLA increases the average performance of the YOLOv4 to YOLOv8 models in identifying insulator breakdown defects. Li et al. [34] presented another study considering the multiscale attention alignment adversarial domain adaptation YOLO, integrating domain adaptation and generalization. The results proved that their model significantly improved the performance of insulator defect detection in previously unseen complex cloudy domain scenarios.

He et al. [35] proposed an improved YOLOv8 for detecting multiple insulator conditions, such as normal, self-explosive, damaged, and flashover. The designed model exhibited high detection performance. Jing et al. [36] also considered the YOLOv8 to improve the feature extraction capability to recognize defective insulators from transmission lines. Li et al. [37] considered the wise-IoU to optimize detection performance. Their method applies Con-

vNeXt, a bi-directional feature pyramid network, wise-IoU, and small object detection heads. The model incorporates multi-scale feature fusion and small object detection into YOLOv8 to improve the performance of identifying diverse equipment under varying scales, partial obstruction, and real-time operational constraints

Li et al. [38] presented a modified YOLOv8 model to detect insulator defects in complex backgrounds. The wise-IoU loss function was added to reduce the negative impact of low-quality pictures, and a C2f network was built by integrating the convolutional receptive field coordination attention module. A cross-validation scheme was used to ensure that the subsets completed training and testing, thus minimizing generalization errors. Insulator data with four fault categories (normal, faulty, dirty, and aged) were tested with high precision.

The results from the YOLOv8 network were fused with frequency domain features selected using the extreme gradient boosting algorithm by Wang and Cheng [39]. They applied this method for fast load identification with generalization capability, achieving an identification performance of over 99%. Zhang et al. [40] suggested another model for detecting insulator defects that considers the different backgrounds of power transmission lines. Experiments demonstrated the performance of the model in both general and complex scenery.

To recognize insulators and detect faults, Li et al. [41] proposed a modified hybrid attention mechanism, regularization, and depth-separable convolution for the YOLOX model. The results showed that the algorithm improved the average detection performance, the speed enabled real-time detection, and the performance focused on complex and small background targets. Based on YOLOv8n, Liu et al. [42] proposed a model to support the safe operation of power systems by addressing the processing and lightweight design challenges for the effective real-time detection of transmission line anomalies by drones and other resource-constrained mobile devices.

Hien et al. [43] proposed an architecture to improve the ability to recognize faulty insulators by enhancing the feature extraction network to accurately localize insulators in visually complex backgrounds. This was performed by introducing a new architecture for object detection and incorporating a diffusion model that enables context-aware inference while preserving linear scalability. Recently, an improved tilted insulator detection based on YOLOv8 was proposed by Wang et al. [44] and tested on real power grid insulators with high detection performance.

YOLOv12 has the attention-oriented mechanisms, including area attention and residual efficient layer aggregation, to improve feature representation [45]. The YOLO26 refines label assignment strategies for small object detection and introduces the multi-scale stochastic gradient descent (MuSGD) optimizer, aiming to improve deployment

efficiency [21]. Applications of YOLO26 for insulator detection, as proposed in this paper, are still rare.

2.2 Data augmentation for power system analysis

A major challenge in applying DL-based models to identify faulty insulators is that the fault classes have few samples [46]. Insulators without faults are rare to find in the network, making it difficult to train DL-based models to identify them. One of the solutions that can be employed to improve the ability to deal with unbalanced data and augmentation through generative models [47].

A dataset of 13,000 photos of synthetic foggy insulators was presented by Zhang et al. [48]. By adding the channel attention mechanism in the YOLOv5 network, a reliable detection model for insulators and their flaws is created. Using a unique fully convolutional network architecture, Sampedro et al. [49] proposed an automatic solution to diagnose and recognize insulator strings. Synthetic photos and sequences of actual aerial inspections of high-voltage insulators were used to evaluate the proposed procedure.

To segment insulators at the pixel level in real-time, a compact end-to-end neural network trained within the context of conditional GAN (cGAN) is suggested by Chang et al. [50]. Superior segmentation and real-time performance are demonstrated by the experimental results. To create segmentation samples for the insulators, including positive, empty, and fake ones, Chang et al. [51] proposed a synthetic approach. Three end-to-end segmentation networks were used by Chang et al. [50] to adjust to the generators in an adversarial training framework to validate these generated samples, which are consistent with the earlier studies.

For pixel-level insulator segmentation, a modified cGAN was proposed by Gao et al. [52]. To simplify the network complexity and extract additional types of feature information, the generator is recreated using encoder-decoder layers with an asymmetric convolution kernel. Experiments confirm that the suggested approach outperforms existing state-of-the-art networks in terms of computing efficiency.

Kim et al. [53] suggested using a deep convolutional GAN as a generative method to generate a large number of line-post insulator defect images. Chen et al. [54] employed the diffusion model for perceptual interpretation of images to improve the model's effectiveness in insulator fault identification. The proposed model outperforms Faster R-CNN and DETRs. Li et al. [55] developed an insulator defect detection model based on improved YOLOv5s, which significantly reduces model parameters while maintaining detection performance. Additionally, the visual object classes dataset and the synthetic faulty insulator dataset were used to validate the generalization of the model.

Zhou et al. [56] collected datasets based on composite insulator surface images. Three GANs were used to adapt

to the image data and synthesize virtual representations of the insulator surface. Liu and Huang [57] introduced multi-domain insulator multi-scale spatial augmentation to assist the target model in more correctly locating faults at various sizes without the need for additional inference resources, which is important for transmission lines. Yang et al. [58] proposed a high-precision technique based on the synthetic weather algorithm and enhanced YOLOv7 to detect insulator flaws in extreme weather conditions. The dataset was enhanced with synthetic rain, snow, and fog algorithms. Besides, the original dataset was augmented with affine and color adjustments to boost the generalization of the model's performance in complex power inspection backdrops.

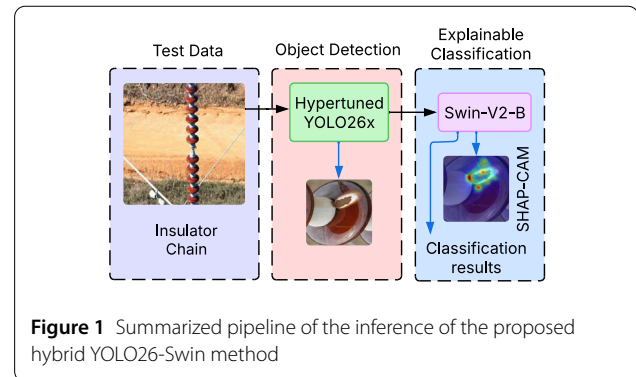
Jiang et al. [59] proposed a weakly-supervised learning-based automatic augmentation method for autonomously generating enhanced aerial pictures using an existing dataset and backdrop images. To ensure diversity and avoid manual costs, the weakly-supervised segmentation mix employs augmentation and segmentation techniques. Zhang et al. [60] employed an enhanced GAN to generate artificial defective insulator samples, which greatly reduces the need for substantial empirical data collection. Several experiments were run to determine the quality of the generated samples, as well as their impact on the detection model.

Ning et al. [61] expanded the glass insulator data using the enhanced denoising diffusion probabilistic models generator. The augmented noise-fitting network produces images of insulator defects with high fidelity and increased quality. Faster R-CNN was selected as the model for defect detection, and ResNet-50 was used instead of the VGG-16 feature extraction network. A DL framework that combines YOLO with deep convolutional GANs and super-resolution GANs was a promising model for the detection and classification of insulator scenarios presented by Akella et al. [62]. In comparison to the state-of-the-art techniques, a lightweight adaptive method for detecting insulator self-explosion faults was presented by Liu and Jiang [63], and it proved to be a promising alternative.

3 Methodology

In this paper, we propose a cascaded framework that separates object detection and classification tasks. Figure 1 presents a summarized structure of the proposed model, called YOLO26-Swin, employed for inference. The proposed structure considers a hypertuned YOLOv26x to identify the insulator position, and a Swin-V2-B to perform classification. In the inference, augmented data, i.e., synthetic images, are not used; only real images recorded during the electrical inspection of the system are considered.

During training, the YOLOv26x has its hyperparameters tuned by Bayesian optimization to ensure its best use. Considering that the dataset has unbalanced classes, synthetic



images generated by conditional diffusion are considered to augment the dataset. The Swin-V2-B is used for classifying the insulator conditions, and a proposed SHAP-CAM is used to provide interpretable results. All the components of this hybrid structure are explained in this section. Figure 2 shows the summarized pipeline of our model considering the training phase.

The major highlights of our proposed model are the following:

- We proposed a YOLO26-Swin to perform identification of insulators and classification of faults separately, as explained in Sect. 3.1.
- To ensure the best use of the model setup, the Bayesian optimization is used for hyperparameter tuning, presented in Sect. 3.2.
- Considering that the dataset is unbalanced, we used the conditional diffusion model to augment the dataset, creating synthetic images of faults (see Sect. 3.3).
- To have more explainability, we proposed a SHAP-CAM, enhancing the explainability of the results; we present more details in Sect. 3.4.

3.1 Proposed YOLO26-Swin

The proposed YOLO26-Swin considers YOLO for insulator detection. The Swin architecture is used based on the bounding boxes found in the object detection phase. Combining YOLO26 with Swin creates a modified model (YOLO26-Swin), which has separate detection and classification tasks. These methods are explained in this subsection.

3.1.1 Object detection using YOLO

DL has revolutionized object detection by enabling highly accurate and efficient recognition of objects in images, using CNN-based architectures like DETR, Faster R-CNN, RetinaNet, SSD, and YOLO. These models leverage large annotated datasets to learn features, allowing them to distinguish objects of varying sizes and orientations. Among these models, the YOLO stands out for its superior performance in object detection [64].

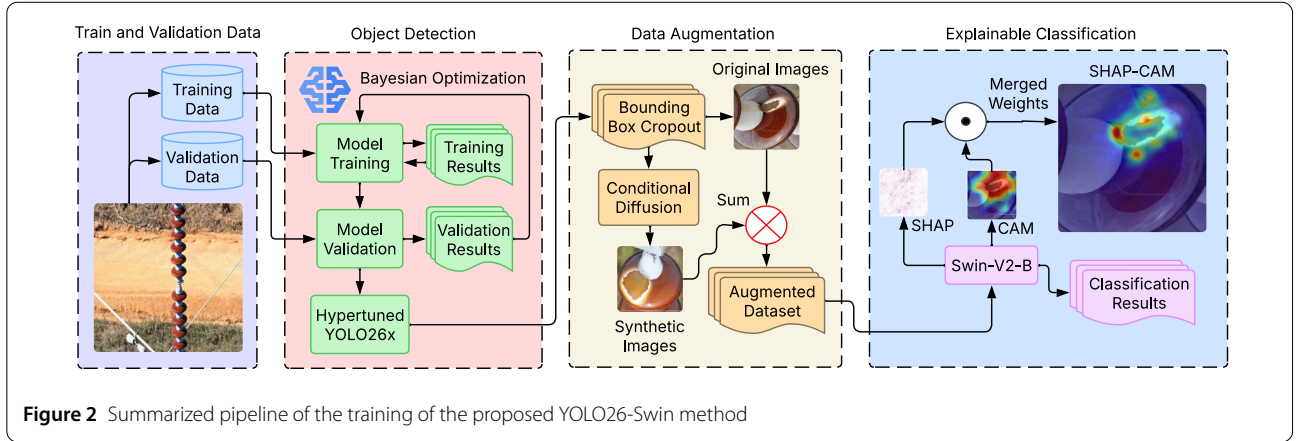


Figure 2 Summarized pipeline of the training of the proposed YOLO26-Swin method

YOLO is a single-stage object detector that reframes object detection as a regression problem. Given an input image, YOLO divides the image into an $S \times S$ grid. Each grid cell is responsible for predicting B bounding boxes, their corresponding confidence scores, and the conditional class probabilities [65].

For each grid cell, the network predicts: The coordinates of the bounding box ($\hat{b} = (x, y, w, h)$), where (x, y) denotes the bounding box center relative to the grid cell boundaries, while w and h correspond to the width and height (often normalized by the image dimensions). The confidence score is given by

$$\hat{c} = \text{Pr}(\text{object}) \cdot \text{IoU}(b, \hat{b}), \quad (1)$$

which represents both the probability that an object exists in the bounding box and the performance of the bounding box prediction as measured by the IoU between the predicted box $\hat{b}$ and the ground truth box b . Conditional class probabilities for each class c are

$$\hat{p}_c = \text{Pr}(\text{class } c \mid \text{object}). \quad (2)$$

The final output tensor of the network has the shape $(S \times S \times (B \cdot 5 + C))$, where factor 5 comes from the four bounding box coordinates plus one confidence score per box, and C is the number of classes [66]. The overall loss function in YOLO is designed to balance localization error, confidence error, and classification error.

These errors are calculated by the localization loss (Eq. (3) and Eq. (4)), confidence loss (Eq. (5)), classification loss (Eq. (6)). Localization loss measures the error in predicting the box center coordinates and dimensions, given by

$$L_{\text{coord}} = \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right], \quad (3)$$

and for the width and height

$$L_{\text{size}} = \lambda_{\text{coord}} \sum_{i=1}^{S^2} \sum_{j=1}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right], \quad (4)$$

where $\mathbb{1}_{ij}^{\text{obj}}$ denotes a binary indicator that takes the value 1 if the j th bounding box predictor in cell i is assigned to detect an object, and 0 otherwise. The square root transformation helps stabilize gradients for larger boxes.

The confidence loss (L_{conf}) penalizes the difference in the confidence score for boxes with and without objects, calculated according to

$$L_{\text{conf}} = \sum_{i=1}^{S^2} \sum_{j=1}^B \left[\mathbb{1}_{ij}^{\text{obj}} (c_i - \hat{c}_i)^2 + \lambda_{\text{noobj}} \mathbb{1}_{ij}^{\text{noobj}} (c_i - \hat{c}_i)^2 \right], \quad (5)$$

where $\mathbb{1}_{ij}^{\text{noobj}}$ equals 1 if the j th bounding box in cell i does not have to any object.

Classification loss (L_{class}) applies only to cells responsible for detecting objects, given by

$$L_{\text{class}} = \sum_{i=1}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2. \quad (6)$$

where λ_{coord} and λ_{noobj} are hyperparameters used to balance the contributions of the localization and confidence errors [67]. Thus, the total loss is

$$L = L_{\text{coord}} + L_{\text{size}} + L_{\text{conf}} + L_{\text{class}}. \quad (7)$$

The localization loss (Eq. (3) and Eq. (4)) measures the error in the bounding box coordinate predictions (with a square root applied to w and h to lessen the impact of large

errors). The confidence loss (Eq. (5)) accounts for the error in the confidence scores when an object is present or absent, respectively. The classification loss (Eq. (6)) measures the classification error only for the grid cells that are responsible for an object.

YOLO produces a tensor that encodes spatial information by means of a grid, and its loss function is a weighted sum of errors in localization, confidence, and classification [68]. This approach allows YOLO to achieve real-time performance with competitive performance. In this paper, YOLO is applied specifically for object detection, and classification is performed by specific CNNs, as explained in the following.

3.1.2 Classification using Swin

The hierarchical vision transformer using Shifted windows [69], known as Swin, builds upon the vision transformer framework by introducing a hierarchical structure that computes self-attention within local windows, significantly reducing computational complexity while preserving the ability to capture both local and global information. Its shifted window mechanism allows for cross-window connections that enable effective multi-scale feature learning.

Unlike traditional vision transformers that operate on fixed-size patches globally, the Swin introduces a shifted windowing scheme, allowing localized self-attention within non-overlapping windows while allowing connections across windows. This design scales better to high-resolution inputs and preserves spatial hierarchies through a patch-merging mechanism, making it highly effective for image classification tasks. By progressively aggregating information from local to global contexts, the Swin achieves promising performance across various benchmarks, outperforming several CNNs models [70].

3.2 Hypertuning by Bayesian optimization

Bayesian optimization constructs a probabilistic model of the objective function and leverages it to identify promising hyperparameter combinations based on a trade-off between exploration and exploitation. By updating the model with each evaluation, it efficiently narrows down the search space and converges toward optimal settings [71]. This makes it well-suited for tuning our proposed model.

In the context of hyperparameter tuning, the goal of Bayesian optimization is to find the hyperparameters that minimize an objective function $f(\mathbf{x})$, calculated according to

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathcal{X}} f(\mathbf{x}). \quad (8)$$

Because each evaluation of f is computationally costly, Bayesian optimization builds a surrogate model to approximate f and intelligently selects the next hyperparameter

configuration to evaluate [72]. A common choice for the surrogate model is a Gaussian process ($\mathcal{GP}$). This process places a priority over functions:

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')), \quad (9)$$

where $m(\mathbf{x})$ is the mean function, $k(\mathbf{x}, \mathbf{x}')$ is a covariance kernel that measures the similarity between inputs. Given n observations

$$\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n \quad \text{with} \quad y_i = f(\mathbf{x}_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma_n^2), \quad (10)$$

where the posterior distribution at a new point $\mathbf{x}$ is Gaussian, given by

$$p(f(\mathbf{x}) | \mathcal{D}_n, \mathbf{x}) = \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x})), \quad (11)$$

with

$$\begin{aligned} \mu(\mathbf{x}) &= \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \\ \sigma^2(\mathbf{x}) &= k(\mathbf{x}, \mathbf{x}) - \mathbf{k}(\mathbf{x})^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}), \end{aligned} \quad (12)$$

where

$$\mathbf{k}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), \dots, k(\mathbf{x}, \mathbf{x}_n)]^\top \quad (13)$$

and $\mathbf{K}$ is the $n \times n$ kernel matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ [73].

The Expected Improvement (EI) guides the selection of the next hyperparameter to evaluate, defined as

$$\text{EI}(\mathbf{x}) = \mathbb{E}[\max(0, f_{\min} - f(\mathbf{x}))], \quad (14)$$

where

$$f_{\min} = \min_{1 \leq i \leq n} f(\mathbf{x}_i) \quad (15)$$

is the best-observed value so far. Since the posterior $f(\mathbf{x}) \sim \mathcal{N}(\mu(\mathbf{x}), \sigma^2(\mathbf{x}))$, EI can be written in closed form:

$$\begin{aligned} \text{EI}(\mathbf{x}) &= (f_{\min} - \mu(\mathbf{x})) \Phi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right) \\ &\quad + \sigma(\mathbf{x}) \phi\left(\frac{f_{\min} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}\right), \end{aligned} \quad (16)$$

where μ is the predicted mean of the objective function, σ is predicted standard deviation (uncertainty), $\Phi(\cdot)$ and $\phi(\cdot)$ denote the cumulative distribution function and the probability density function of the standard normal distribution, respectively [74].

Algorithm 1 shows how Bayesian Optimization is computed considering a Gaussian process ($\mathcal{GP}$).

Algorithm 1 Bayesian optimization**Require:** Domain $\mathcal{X}$, budget N

- 1: Initialize dataset $\mathcal{D}_0$ with n_0 points via space-filling design;
- 2: **for** $n = n_0, n_0 + 1, \dots, N - 1$ **do**
- 3: Fit $\mathcal{GP}$ surrogate to $\mathcal{D}_n$
- 4: Select next point $\mathbf{x}_{n+1} = \arg \max_{\mathbf{x} \in \mathcal{X}} \alpha_n(\mathbf{x})$
- 5: Evaluate $y_{n+1} = f(\mathbf{x}_{n+1})$
- 6: Augment dataset: $\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(\mathbf{x}_{n+1}, y_{n+1})\}$
- 7: **end for**
- 8: **return** $\mathbf{x}^* = \arg \min_{(\mathbf{x}_i, y_i) \in \mathcal{D}_N} y_i$

3.3 Data augmentation using conditional diffusion

Data augmentation using Generative Artificial Intelligence (GenAI) is an approach that enhances ML models by artificially increasing the size and diversity of training datasets [75]. GenAI models, such as GANs [76] and diffusion models, can create synthetic yet realistic data samples, including images of electrical power grid components. Data augmentation helps improve model generalization, especially in scenarios with limited data availability [77]. In this paper, a conditional diffusion model is used for data augmentation. This model is explained in the following.

Diffusion models belong to a class of generative models that generate data by learning to reverse a stochastic diffusion process. The idea is to gradually add noise to data during training and then learn to reverse this process to generate new samples [78]. The forward diffusion process adds Gaussian noise to data over T time steps. Given an initial data sample $x_0 \sim q(x)$, the forward process is defined as

$$q(x_t | x_{t-1}) = \mathcal{N}(x_t; \sqrt{1 - \beta_t} x_{t-1}, \beta_t I), \quad (17)$$

where β_t is a variance schedule controlling the amount of noise added at each step. Using the reparameterization, the noisy sample at any timestep t can be directly computed as

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t} \epsilon, \quad (18)$$

where $\epsilon \sim \mathcal{N}(0, I)$ and $\bar{\alpha}_t = \prod_{s=1}^t (1 - \beta_s)$.

The goal of the model is to learn the reverse process:

$$p_\theta(x_{t-1} | x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t)). \quad (19)$$

Diffusion models provide a framework for generative modeling by learning to reverse a noise-adding process [79]. This paper considers a conditional diffusion model; thus, the model generates images of synthetic insulators of the flashover and broken classes in a single run. The focus is on generating images of faults because the number of faults is much smaller than that of insulators in good condition, so it is possible to have a more balanced dataset.

3.4 Explainability using the proposed SHAP-CAM

In this study, SHAP is merged with CAM, named SHAP-CAM, enabling a more comprehensive and interpretable analysis of model behavior, particularly in complex tasks such as image classification or object detection. SHAP provides global feature importance, while CAM provides spatial insights. Together, they provide both feature-level and region-level explanations. SHAP-CAM is used to highlight the regions of an input image that are most influential to the model's decision [80].

3.4.1 SHAP

To improve the interpretability and transparency of DL models, the SHapley Additive exPlanations approach [81], in short SHAP, is applied. SHAP is based on the concept of Shapley values from cooperative game theory, which attribute the contribution of each player (feature) to the total payoff (model output) by considering all possible feature combinations.

Shapley values measure the individual contribution of each participant within a cooperative system. Variables to classify insulator defects are considered as participants within the SHAP structure. When a new variable is introduced into the model, SHAP calculates its marginal contribution by considering how it influences the outcome across various permutations of the variable order [82]. This process allows for a comprehensive assessment of the significance of each variable, facilitates the analysis of individual responses to changes in influencing factors, and provides insight into how variable interactions influence the final prediction.

Given a model f and an input instance x , SHAP computes the contribution ϕ_i of each feature i by averaging the marginal contributions of that feature over all possible subsets of features. Formally, the Shapley value for feature i is defined as

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} [f(S \cup \{i\}) - f(S)], \quad (20)$$

where F is the set of all features, S is a subset of features excluding i , and $f(S)$ represents the model's output when only the features in S are considered [83].

3.4.2 CAM

CAM methods [84] allow the visualization of the spatial regions in the input that contribute most to the model's predictions. These methods generate saliency maps that highlight relevant features, assisting domain experts in validating the model's focus areas and ensuring that the decisions are based on meaningful patterns rather than artifacts or noise. In this paper, a set of advanced CAM techniques was applied to understand the outputs of our proposed YOLO26-Swin model. The CAM methods evaluated include eXtended Gradient-weighted CAM (XGradCAM),

Algorithm 2 SHAP-CAM visual explanation generation**Require:** Input image I , trained model M , transformations T , target layers L **Ensure:** Visual explanations combining SHAP values and CAM heatmaps

```

1:  $I_{\text{tensor}} \leftarrow T(I)$ 
2:  $y_{\text{pred}} \leftarrow M(I_{\text{tensor}})$ 
3:  $c_{\text{pred}} \leftarrow \arg \max(y_{\text{pred}})$ 
4:  $\text{target} \leftarrow \text{ClassifierOutputTarget}(c_{\text{pred}})$ 
5:  $\text{CAM} \leftarrow \text{XGradCAM}(M, L, \text{reshape\_transform})$ 
6:  $G \leftarrow \text{CAM}(I_{\text{tensor}}, \text{target})$ 
7:  $I_{\text{denorm}} \leftarrow \text{Denormalize}(I_{\text{tensor}})$ 
8:  $I_{\text{denorm}} \leftarrow \text{Clip}(I_{\text{denorm}}, 0, 1)$ 
9: for all images  $i$  and SHAP values  $S_i$  in SHAP explanation set do
10:    $S_+ \leftarrow \text{Clip}(S_i, 0, \infty)$ 
11:    $S_+ \leftarrow \text{GaussianFilter}(S_+, \sigma = 5)$ 
12:    $\text{Mask} \leftarrow S_+ \cdot G$ 
13:    $\text{Mask} \leftarrow \text{Normalize}(\text{Mask})$ 
14:    $\text{Visualization} \leftarrow \text{show\_cam\_on\_image}(I_{\text{denorm}}, \text{Mask})$ 
15: end for

```

High-Resolution CAM (HiResCAM), and Score-weighted CAM (ScoreCAM).

XGradCAM extends the original Grad-CAM [85] formulation by introducing a more stable normalization of the gradient-based weights. Instead of relying solely on the average gradients, XGradCAM applies an alternative normalization scheme to mitigate the impact of vanishing or unstable gradients. Although XGradCAM lacks a formal peer-reviewed publication, it has been incorporated into practical implementations such as the `pytorch-grad-cam` library, where it demonstrates improved stability and consistency over Grad-CAM.

HiResCAM [86] proposes a gradient-free approach that utilizes the raw activations of the convolutional layers to produce high-resolution saliency maps. By bypassing the need for backpropagation, HiResCAM reduces computational overhead and enables finer spatial localization. However, since it does not incorporate gradient information, it may sacrifice some fidelity in representing the true contribution of the highlighted regions to the model's output.

ScoreCAM, introduced by Wang et al. [87], eliminates the reliance on gradients by using the model's output scores to weight the importance of each activation map. This is achieved by masking the input image with individual activation maps and measuring the change in the model's output score for the target class. While ScoreCAM provides high-resolution and robust saliency maps, it is computationally expensive as it requires multiple forward passes through the network for each activation map.

3.4.3 Combining SHAP with CAM

Algorithm 2 shows the hybrid visual explanation technique that combines SHAP values with CAM (our pro-

posed SHAP-CAM) to improve the interpretability of deep neural network predictions. Initially, the input image is transformed and passed through a trained model to obtain the predicted class. Using this class as a target, a CAM heatmap is generated via CAM methods (XGradCAM, HiResCAM, or ScoreCAM).

SHAP values are computed and filtered to retain only positive contributions, which are then smoothed. The final explanation is produced by element-wise multiplication of the CAM and SHAP maps, followed by normalization and overlay on the original image, yielding a combined visualization that highlights the most relevant regions both in terms of class activation and feature importance.

The integration of SHAP and CAM in Algorithm 2 is performed through a complementary fusion mechanism that combines feature attribution with spatial localization. Specifically, the CAM heatmap G encodes class-discriminative spatial regions derived from the network activations, while the SHAP values S_i quantify the contribution of input features to the model output.

To ensure interpretability, only the positive SHAP contributions are retained and smoothed via a Gaussian filter to reduce noise. The fusion is then carried out through an element-wise multiplication between the processed SHAP map S_+ and the CAM heatmap G , i.e., $\text{Mask} = S_+ \odot G$, which emphasizes regions that are simultaneously important in both attribution spaces.

This operation acts as a gating mechanism, preserving only the spatial locations that are supported by both feature importance and activation strength. Finally, the resulting map is normalized to the $[0, 1]$ range to ensure consistent visualization and comparability across samples, and overlaid on the original image. This formulation improves robustness and reproducibility by explicitly defining how

SHAP and CAM are jointly leveraged to produce coherent visual explanations.

4 Experimental setup

The experiments were performed using an NVIDIA RTX 3060 TI graphics processing unit with 120 GB of random-access memory. The algorithms proposed here were written using the Python language. For object detection, a maximum of 500 training epochs was defined, using the early stop criterion when there is no improvement in training after 100 epochs.

For the generation of synthetic images using generative models, a maximum of 5000 epochs was considered, and the performance was evaluated according to the number of epochs for training to define how many epochs are needed to train the model to achieve the best performance. For future comparisons, the proposed algorithm is available at: <https://github.com/jpmcarvalho/Optimized-YOLO>.

4.1 Performance evaluation metrics

For model assessment, the results of precision (21), recall (22), F1-score (23), and mean average precision (mAP) (24), are considered. Specifically, the IoU threshold applied for object detection is 0.5 (mAP@[0.5]) and from 0.5 to 0.95 (mAP@[0.5:0.95]), achieved by calculating the average of the results using these thresholds with a step of 0.05. These metrics use the true positive (tp), false positive (fp), and false negative (fn), calculated according to

$$\text{precision} = \frac{tp}{tp + fp} \quad (21)$$

$$\text{recall} = \frac{tp}{tp + fn} \quad (22)$$

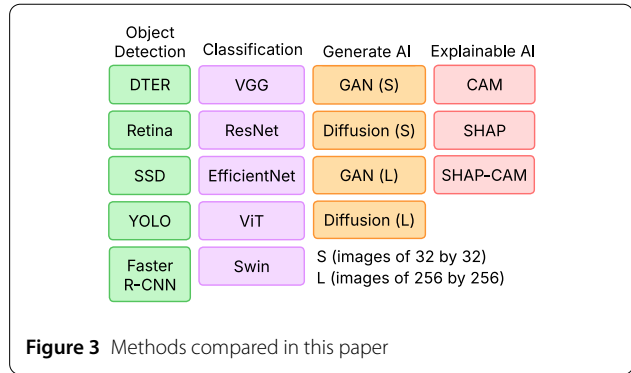
$$\text{F1-score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{recall} + \text{precision}} \quad (23)$$

$$\text{mAP} = \frac{1}{n} \sum_{k=1}^n \left(\sum_{\eta} (\text{recall}_{\eta} - \text{recall}_{\eta-1}) \text{precision}_{\eta} \right)_k \quad (24)$$

where η is the n th threshold and k is the corresponding class of n classes [88].

4.2 Models for comparison

Figure 3 presents all the compared models evaluated in this paper. In terms of object detection, the proposed method is compared to the DETR [89], Faster R-CNN [90], RetinaNet [91], SSD [92], YOLO (10, 11, 12, and 24), STN-YOLO, and SAE-YOLO [22]. Regarding classification, the YOLO26-Swin is compared to the VGG (11, 13, 16, and 19) [93], ResNet (50, 101, and 152) [94], EfficientNet (V2-S, V2-M, and V2-L) [95], ViT (B-16, B-32, L-16, and L-32) [96], Swin (V2-T, V2-S, and V2-B).



These CNN architectures are compared based on the bounding boxes found in object detection. For data augmentation, the conditional diffusion is compared to the conditional Wasserstein GAN [97]. For explainability comparison, the SHAP, XGradCAM, HiResCAM, ScoreCAM, and SHAP-CAM (using all these CAM methods) are evaluated.

4.3 Dataset

Images of insulators with three subclasses of condition make up the dataset: good insulator, broken insulator, and flashover insulator. These are original, high-resolution photos taken from high-voltage inspected transmission power grids. Lewis and Kulkarni's repository [98] contains the original dataset and additional metadata for future comparisons.

High-resolution pictures taken by digital single-lens reflex cameras during power grid inspections in favorable weather circumstances make up the dataset under consideration. The pre-processing involved rescaling the photos to 640×640 pixels, which is the usual size for the models under consideration, and converting the annotations from JSON files to YOLO-compatible readable files (.txt). An example of the considered insulators (highlighting the damaged ones in a red box) is shown in Fig. 4.

The dataset contains 1,596 images and 17,108 annotations, with a consistent density of about 10.7 annotations per image across training, validation, and test splits. The partitioning preserves similar annotation distributions, ensuring no bias in object density between splits. Table 2 summarizes the composition of the dataset used in the experimental evaluation.

The dataset contains 1,596 images and 17,108 annotated bounding-box instances, divided into training, validation, and test subsets with 1117, 159, and 320 images, respectively. The average number of annotations per image remains consistent across the three subsets, ranging from 10.54 to 11.33, indicating a similar object density among the splits. However, the class distribution is highly imbalanced: the *No issues* class represents 78.1% of all annotations, while the *Broken* and *Flashover* fault classes account

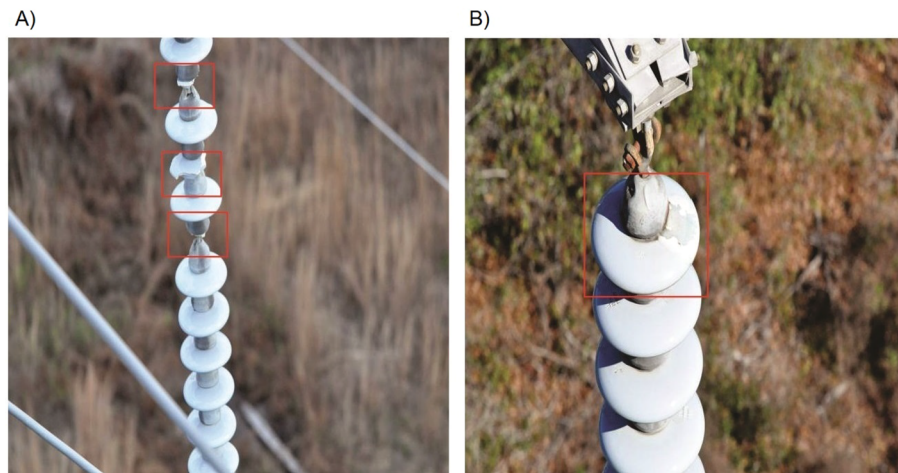


Figure 4 Examples of insulator fault conditions in the original dataset: (A) Broken and (B) flashover cases. Highlighted fault regions are indicated by red bounding boxes

Table 2 Composition of the insulator fault dataset used for training, validation, and testing. Annotation counts refer to individual bounding-box instances per class

Split	Images	Annotations per class			Total ann.	Avg. ann./img
		No issues	Broken	Flashover		
Train	1,117	9,266	818	1,850	11,934	10.68
Validation	159	1,457	109	235	1,801	11.33
Test	320	2,641	253	479	3,373	10.54
Total	1,596	13,364	1,180	2,564	17,108	10.72
Share (%)	—	78.1%	6.9%	15.0%	100%	—

for only 6.9% and 15.0%, respectively. This imbalance highlights the difficulty of learning representative fault patterns, particularly for broken insulators, and motivates the use of synthetic data augmentation to increase the diversity and availability of minority-class samples during training.

5 Results and discussion

This section presents the results and discusses the application of the proposed YOLO26-Swin. Initially, the results are evaluated in relation to the object detection task, then the classifiers are analyzed. Considering the best classifier, synthetic images are used in the training with the focus on mitigating the problem related to unbalanced classes.

Using the best classifier with the augmented dataset, interpretable results are presented with the SHAP, CAM, and the proposed SHAP-CAM to better understand the reasons why the model performed the classification. Finally, benchmarking is carried out to prove that the proposed model is adequate and it outperforms state-of-the-art DL models.

5.1 Object detections results

The proposed YOLO26-Swin uses YOLO26 as an object detection model, focusing on identifying the position of the insulators. Here, we present the results of a comparison of YOLO, considering several versions and other DL models applied specifically for object detection. After this analysis, we present the YOLO hypertuning results.

5.1.1 Object detection model selection

Table 3 compares the insulator detection performance, in which all the conditions of the insulators are of the same class. In this initial experiment, the models are trained for up to 500 epochs using an early stop criteria in which training is finished when there is no improvement in the model after 100 epochs. For comparison purposes, a batch size equal to 4 (a higher batch size restricts the use of available hardware, see Sect. 4) was considered using the default model hyperparameters of each structure. The performance is also evaluated by Giga Floating-point Operations Per second (GFLOPs).

The baseline models, including DETR, Faster R-CNN, RetinaNet, and SSD, present moderate detection performance. Their mAP@[0.5] values are approximately 0.90 for

Table 3 Insulator detection evaluation by standard models

Model	mAP		GFLOPs	Training time (h)	Inference time (ms)
	[0.5]	[0.5:0.95]			
DETR	0.90043	0.64615	234.96	1.85	15.21
Faster R-CNN	0.90212	0.64748	272.14	2.68	80.23
RetinaNet	0.90057	0.62167	245.77	2.98	65.67
SSD	0.83203	0.46371	91.28	1.15	29.26
STN-YOLO	0.00300	0.00074	10.70	3.18	9.15
SAE-YOLO	0.96098	0.80886	9.49	11.53	10.17
YOLOv10n	0.97937	0.88372	8.39	2.26	3.31
YOLOv10s	0.97657	0.89232	24.77	2.76	4.90
YOLOv10m	0.98193	0.90178	63.97	4.32	6.48
YOLOv10l	0.98107	0.91392	127.20	6.15	10.66
YOLOv10x	0.98236	0.91555	171.00	8.62	15.43
YOLOv11n	0.98445	0.89489	6.44	1.76	3.44
YOLOv11s	0.98194	0.89539	21.55	2.15	4.09
YOLOv11m	0.98602	0.90075	68.19	3.93	7.22
YOLOv11l	0.98348	0.89773	87.27	5.01	7.81
YOLOv11x	0.98344	0.90355	195.45	8.32	14.70
YOLOv12n	0.98273	0.91799	6.48	0.77	4.11
YOLOv12s	0.98651	0.90205	21.52	3.03	5.87
YOLOv12m	0.98202	0.89150	67.74	4.78	8.13
YOLOv12l	0.98137	0.89529	89.41	7.83	12.59
YOLOv12x	0.98311	0.89626	199.82	12.48	21.04
YOLOv26n	0.98486	0.91759	5.77	3.44	4.13
YOLOv26s	0.98201	0.91653	22.50	5.61	4.93
YOLOv26m	0.97665	0.89507	74.72	7.24	6.85
YOLOv26l	0.98667	0.90058	93.12	8.95	8.04
YOLOv26x	0.98821	0.90544	208.51	12.15	14.08

Best results in bold

DETR, Faster R-CNN, and RetinaNet, while SSD exhibits a lower value of 0.83203. When considering the more restrictive mAP@[0.5:0.95], all baseline models show a significant reduction, with values ranging from 0.46371 for SSD to approximately 0.64748 for Faster R-CNN. These results indicate limited localization precision under stricter IoU thresholds. In addition, these models present relatively high computational costs, with GFLOPs exceeding 200 for most architectures and inference times reaching up to 80.23 ms for Faster R-CNN, which restricts their applicability in real-time scenarios. SAE-YOLO showed promising results, performing better than all models except the original YOLO models. STN-YOLO did not perform well in this analysis.

The YOLOv10 family introduces a substantial improvement in both detection performance and efficiency. All variants significantly outperform the baseline models, with mAP@[0.5] values exceeding 0.97 and mAP@[0.5:0.95] values above 0.88. The progression from YOLOv10n to YOLOv10x shows a consistent increase in performance, reaching 0.98236 and 0.91555, respectively. This improvement is associated with increased computational complexity, as GFLOPs rise from 8.39 to 171.00 and inference time

increases from 3.31 ms to 15.43 ms. Nevertheless, even the largest YOLOv10 model remains faster than traditional detectors while achieving higher performance.

The YOLOv11 models further refine this trade-off. The smallest variant, YOLOv11n, achieves a mAP@[0.5] of 0.98445 with only 6.44 GFLOPs, representing the lowest computational cost among all evaluated models. The medium and large variants improve performance, with YOLOv11m reaching 0.98602 and YOLOv11x achieving 0.90355 in mAP@[0.5:0.95]. The inference times remain within a practical range, below 15 ms, confirming the suitability of this family for real-time applications.

The YOLOv12 family demonstrates additional improvements in localization performance. The YOLOv12n model achieves the highest mAP@[0.5:0.95] value of 0.91799 among all models with a relatively low computational cost of 6.48 GFLOPs and the shortest training time of 0.77 hours. Larger variants such as YOLOv12s reach a mAP@[0.5] of 0.98651, although at the expense of increased training and inference times. This indicates that YOLOv12 effectively enhances bounding box precision while maintaining computational efficiency in its smaller configurations.

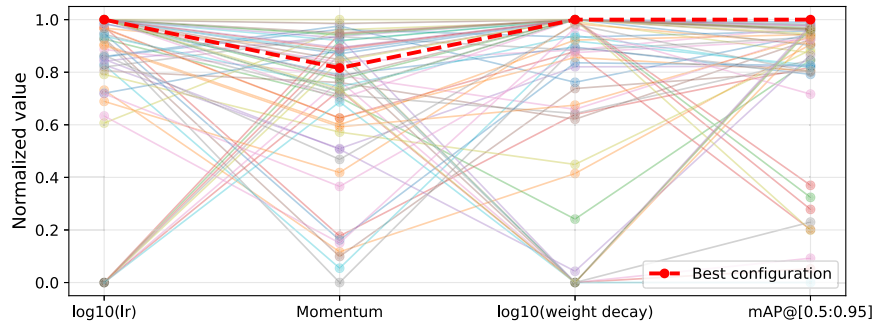


Figure 5 Combination of hyperparameters chosen in the model trials

The YOLOv26 models achieve the highest overall detection performance. In particular, YOLOv26x reaches the best mAP@[0.5] value of 0.98821, while maintaining a competitive mAP@[0.5:0.95] of 0.90544. The YOLOv26n variant presents a favorable balance, with mAP@[0.5] of 0.98486 and mAP@[0.5:0.95] of 0.91759, combined with low GFLOPs of 5.77. However, larger variants such as YOLOv26x require significantly higher computational resources, exceeding 200 GFLOPs, which approaches the complexity of traditional detectors.

From a global perspective, the results indicate that YOLO based architectures consistently outperform traditional detection models in both performance and efficiency. For maximum detection performance, YOLOv26x offers the highest mAP@[0.5] . In the structure definition phase, we considered this model as the most promising to be hypertuned, which is carried out using Bayesian optimization.

5.1.2 Hypertuning YOLO26

Figure 5 presents the results of the multi-objective optimization trials to tune the model hyperparameters. A trial consists of choosing a value for each hyperparameter, and the lines in this plot show different combinations of values. The darker color of the lines represents the trial that performed best in terms of achieving the objective function, which in this case is the optimization of the mAP value (presented on the right side of Fig. 5).

Bayesian optimization runs 50 trials, combining the hyperparameters to achieve the model's best performance (mAP). In this analysis, the evaluated hyperparameters were learning rate [1.0×10^{-5} : 1.0×10^{-1}], momentum [1.0×10^{-9} :0.99], optimizer [Stochastic Gradient Descent (SGD), Adaptive moment estimation (Adam), Root Mean Square propagation (RMSprop), Adam with Weight decay (AdamW), and weight decay [1.0×10^{-9} : 2.0×10^{-1}].

Based on this optimization experiment, the set of hyperparameters is learning rate equal to 0.01, momentum equal to 0.001, SGD optimizer, and weight decay equal to

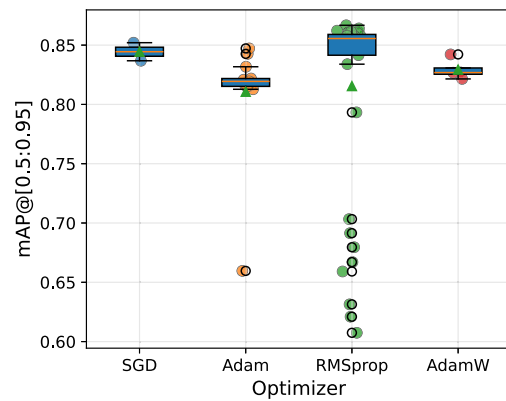


Figure 6 Optimizer comparison in the model hypertuning experiment

0.0359. Using these hyperparameters, the model is trained once more for its best application. The parameter that resulted in greater variability was the choice of the optimizer, shown in Fig. 6.

RMSprop achieves the highest peak values among all optimizers, reaching around 0.86–0.87. However, this comes at the cost of substantial variability. The distribution shows a wide spread, with numerous low-performing outliers extending down to approximately 0.60. Considering that the maximum performance is prioritized and variability can be managed through extensive tuning, RMSprop is used.

Figure 7 shows the values of the mAP@[0.5] and mAP@[0.5:0.95] for insulator detection. It is possible to notice that it takes several epochs for the model to stabilize (after 450), highlighting how a model needs to be trained properly (until stabilized) to achieve its best performance.

5.2 Classification results

We present the classification results for defining the base classification model. A perceived challenge in the classification phase is the low number of samples that repre-

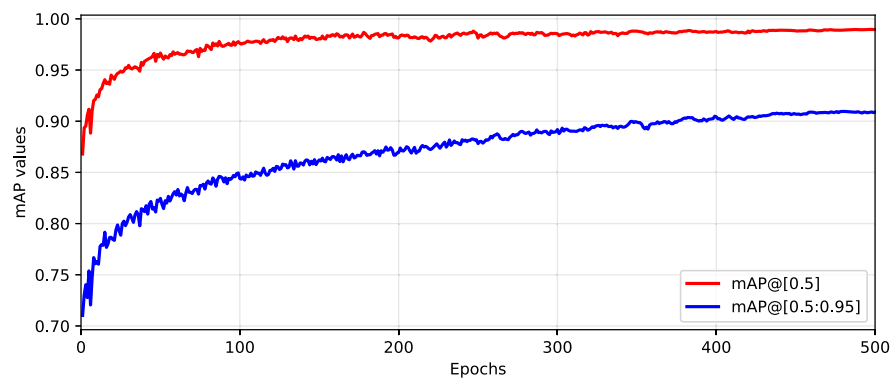


Figure 7 Comparison of mAP metrics over training epochs

Table 4 Selection of the CNN to be used for classification

Model	Precision	Recall	F1-score	GFLOPs	Training time (h)	Inference time (ms)
VGG-11	0.94597	0.90518	0.92457	128.68	2.68	0.52
VGG-13	0.94374	0.91627	0.92939	180.87	3.63	0.62
VGG-16	0.94444	0.91784	0.93070	247.46	4.38	0.84
VGG-19	0.92727	0.90124	0.91361	314.0	5.13	0.97
ResNet-50	0.97127	0.91866	0.94339	66.11	2.24	3.24
ResNet-101	0.96159	0.91532	0.93722	125.8	3.50	6.19
ResNet-152	0.96620	0.92493	0.94454	185.6	6.15	9.37
EfficientNet-V2-S	0.98158	0.95686	0.96886	46.4	2.69	9.31
EfficientNet-V2-M	0.96603	0.97169	0.96884	87.1	4.31	13.63
EfficientNet-V2-L	0.96601	0.94024	0.95261	198.05	5.45	22.54
ViT-B-16	0.97250	0.91986	0.94428	180.56	8.08	3.20
ViT-B-32	0.92127	0.89710	0.90872	47.2	4.80	3.09
ViT-L-16	0.95272	0.91692	0.93384	637.68	25.67	6.52
ViT-L-32	0.93065	0.90953	0.91875	163.68	14.48	6.07
Swin-V2-T	0.97511	0.94841	0.96135	47.6	4.10	8.68
Swin-V2-S	0.97584	0.95871	0.96712	92.2	6.93	16.79
Swin-V2-B	0.97692	0.96116	0.96887	163.58	9.90	17.51

Best results in bold

sent faults, considering that most of the insulators are in good condition. To mitigate this problem, the conditional Wasserstein GAN and the conditional diffusion model were compared to augment the dataset, creating synthetic images of broken and flashover insulators.

5.2.1 Selection of baseline classifier

After using the optimized model to detect the insulators, the classification of faults in relation to the bounding boxes and those found is carried out. Table 4 shows the comparative results of several CNN models to select the most suitable model to be used in the classification phase. The F1-score is the main measure considered here to evaluate the classification performance.

The most promising results were found using EfficientNet and the second-generation Swin. Specifically, using EfficientNet-V2-S, a precision equal to 0.98158 was

achieved, and using EfficientNet-V2-S, a recall equal to 0.97169 was achieved, these being the best performance values found for the models evaluated. Considering that the F1-score takes into account both precision and recall values, this metric was chosen to determine the best model, which had a high precision and recall value simultaneously, being Swin-V2-B with an F1-score equal to 0.96887.

Observing that the images are resized to 640×640 to be used in the object detection model, two analyses were evaluated for classifying insulators. The first analysis was using bounding boxes equal to 256×256 ; in this case, considering the original high-resolution images, the second analysis was carried out considering the use of pre-processed (compressed) images, in this case with bounding boxes equal to 32×32 .

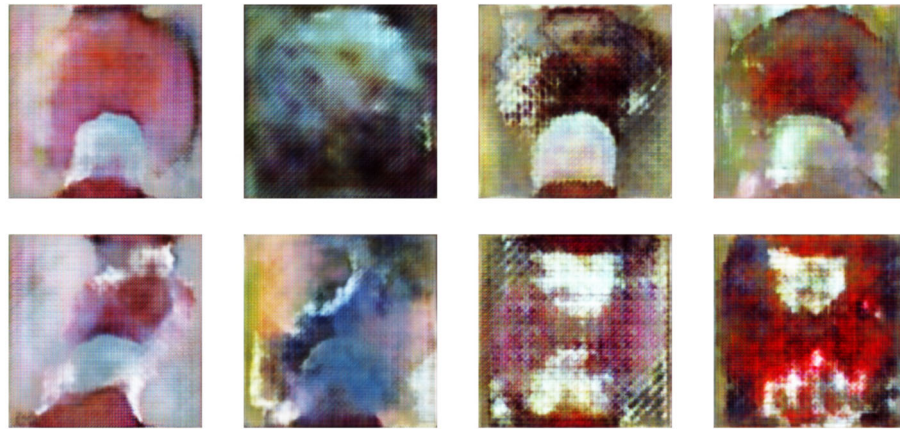


Figure 8 Broken insulators generated by the conditional Wasserstein GAN with a size of 256×256 . The columns represent 500, 1000, 2500, and 5000 training epochs

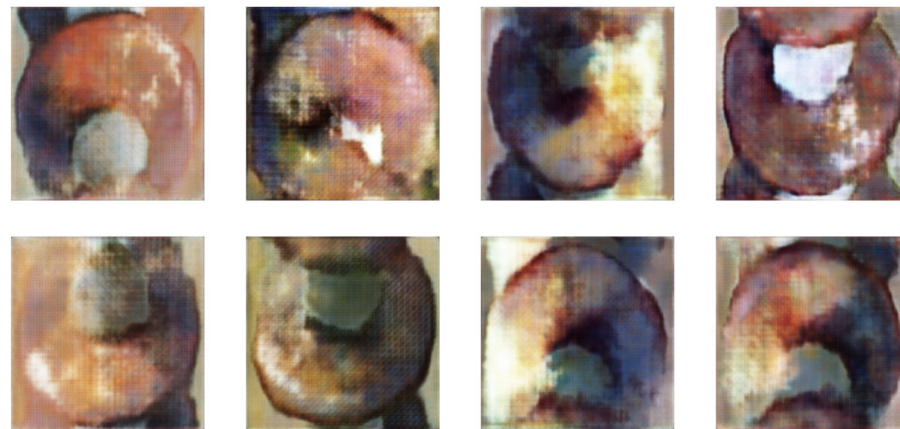


Figure 9 Flashover damage insulators generated by the conditional Wasserstein GAN with a size of 256×256 . The columns represent 500, 1000, 2500, and 5000 training epochs

5.2.2 High resolution synthetic images

This subsection evaluates two methods of synthetic image generation, the conditional Wasserstein GAN and conditional diffusion. The synthetic images are used to augment the training dataset. The comparison considers images of 256×256 . Figure 8 and Fig. 9 present examples of synthetic images of 256×256 generated by the conditional Wasserstein GAN, respectively of broken and flashover insulators. Comparatively, Fig. 10 and Fig. 11 present examples of images generated by conditional diffusion.

These images are presented according to the number of training epochs by each model. Qualitative results show that diffusion models are considerably superior to the conditional Wasserstein GAN. For diffusion models, it is difficult to define which images are synthetic after 2500 training epochs, which is considerably promising given that this is the aim of the augmentation carried out in this work.

Table 5 shows the quantitative results using synthetic images for the model training. This analysis considers images of 256×256 pixels. The performance results are evaluated, given the number of epochs used to generate the images and the variation in the number of images used. The training time is based on the classifier model, which can vary depending on the use of the early stop.

The best result found using synthetic images was using the conditional diffusion model, trained up to 2500 epochs, using 5000 images. This result was 2.1% better than the classification model using only real images, thus justifying the use of synthetic images to train the classifier. In terms of computational effort for classification, for images of 256×256 , all the models had GFLOPs equal to 163.58, considering that the same batch size (16) was used and the same classifier (Swin-V2-B). The inference time is also the same, equal to 17.51 milliseconds. The training

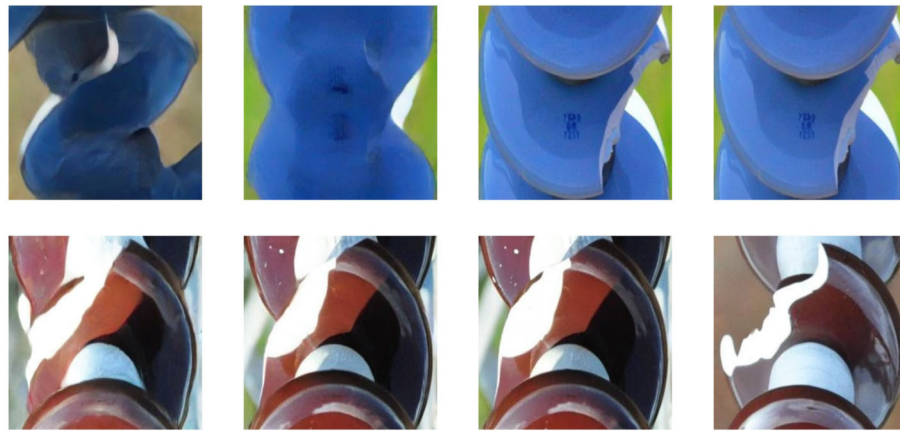


Figure 10 Broken insulators generated by a conditional diffusion model with a size of 256×256 . The columns represent 500, 1000, 2500, and 5000 training epochs

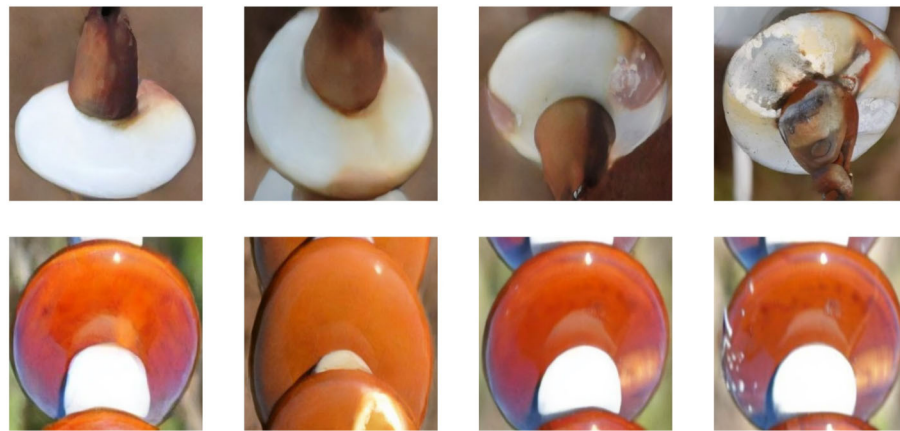


Figure 11 Flashover damage insulators generated by a conditional diffusion model with a size of 256×256 . The columns represent 500, 1000, 2500, and 5000 training epochs

time varied mainly due to the use of the early stop criterion.

Having the best result considering images trained up to 2500 epochs shows that training beyond this value is not necessary to achieve better results, and in this experiment, when the model was trained up to 5000 epochs, there were no major visual or numerical differences. Most of the results using the conditional diffusion model were higher than 0.98, even when a smaller number of images were used (1000 synthetic images), proving their visual and numerical superiority compared to the conditional Wasserstein GAN.

Although the images generated by the conditional Wasserstein GAN are visually different from the original images, there was still a performance improvement when these images were used. This shows that this improvement is due to the intrinsic characteristics of the classification

model, since in models based on conditional Wasserstein GAN, the generator is trained to fool the classifier as to whether the images are synthetic or not.

The results of using the conditional diffusion model were promising for both broken insulators and insulators damaged by flashover. Especially for broken insulators, there was variability in the generation of different colored components with different backgrounds. This result is considerably promising since the aim of augmentation is to create a dataset that represents greater variability in the samples, so that the classification model can be used for different types of insulators.

5.2.3 Low resolution synthetic images

The purpose of using smaller images in this analysis is to compare whether the preprocessing phase impairs classification by reducing the quality of the images, noting that

Table 5 Classification results using augmented data (images of 256×256 pixels)

Model	Synthetic images	Epochs ^a	Precision	Recall	F1-score	Training time (h)
Original	-	-	0.97692	0.96116	0.96887	9.90
Conditional Wasserstein GAN	500	500	0.97972	0.97746	0.97857	24.04
		1000	0.97234	0.96703	0.96957	36.47
		2500	0.97877	0.96894	0.97376	11.74
		5000	0.98652	0.96132	0.97346	14.40
	1000	500	0.98511	0.97406	0.97949	29.41
		1000	0.98507	0.97193	0.97838	32.85
		2500	0.98117	0.97819	0.97966	20.15
		5000	0.98552	0.96759	0.97633	23.70
	2500	500	0.98343	0.97837	0.98086	35.27
		1000	0.98580	0.97500	0.98028	37.37
		2500	0.98240	0.98075	0.98157	23.28
		5000	0.98943	0.98415	0.98676	33.03
	5000	500	0.98925	0.98543	0.98731	27.24
		1000	0.98944	0.98754	0.98848	30.77
		2500	0.98939	0.98830	0.98884	21.23
		5000	0.98979	0.98928	0.98953	42.84
Conditional Diffusion	500	500	0.97354	0.98232	0.97786	20.16
		1000	0.98150	0.97883	0.98014	20.27
		2500	0.98243	0.97633	0.97933	20.25
		5000	0.96989	0.97837	0.97398	9.72
	1000	500	0.98339	0.98197	0.98267	22.98
		1000	0.97841	0.98554	0.98192	27.72
		2500	0.98776	0.97540	0.98147	28.09
		5000	0.99005	0.98060	0.98526	10.00
	2500	500	0.98186	0.97857	0.98017	50.10
		1000	0.98994	0.98494	0.98741	30.94
		2500	0.98647	0.97529	0.98076	22.91
		5000	0.98333	0.98841	0.98583	13.07
	5000	500	0.98855	0.98492	0.98669	45.98
		1000	0.98729	0.98515	0.98621	31.82
		2500	0.99028	0.98944	0.98986	18.50
		5000	0.98830	0.98837	0.98833	14.54

^aEpochs used to create synthetic images

Best results in bold

current YOLO models perform this operation to reduce computational complexity.

Figure 12 (broken insulators) and Fig. 13 (flashover insulators) show examples of synthetic images of 32×32 generated by the conditional Wasserstein GAN. Figure 14 (broken insulators) and Fig. 15 (flashover insulators) show examples of conditional diffusion-generated images. Here, the results are also presented according to the number of training epochs by each model.

Qualitatively, the images generated by conditional Wasserstein GAN had promising results, as did the results of diffusion models after a large number of training epochs. With a low number of training epochs, the synthetic images generated by the conditional Wasserstein GAN were superior to the images generated by the conditional diffusion model, but when more than 2500 epochs are consid-

ered for generating synthetic images, the diffusion-based models show better results.

Table 6 shows the classification results using 32×32 images. The best result was found using a conditional diffusion model, achieving an F1-score of 0.98149, which is 1.26% better than the classification model (Swin-V2-B) using only original images for training. This shows that the use of synthetic images for class balancing improves the classification model's capacity also for smaller images. In terms of computational effort, all the models had an equal GFLOPs for processing images of 32×32 , considering that the same classifier (Swin-V2-B) and batch size (16) were used.

An interesting result observed using 32×32 synthetic images is that when 500 epochs are used to train the generative model, the classification results are inferior to using



Figure 12 Broken insulators generated by the conditional Wasserstein GAN with a size of 32×32 . The columns represent 500, 1000, 2500, and 5000 training epochs

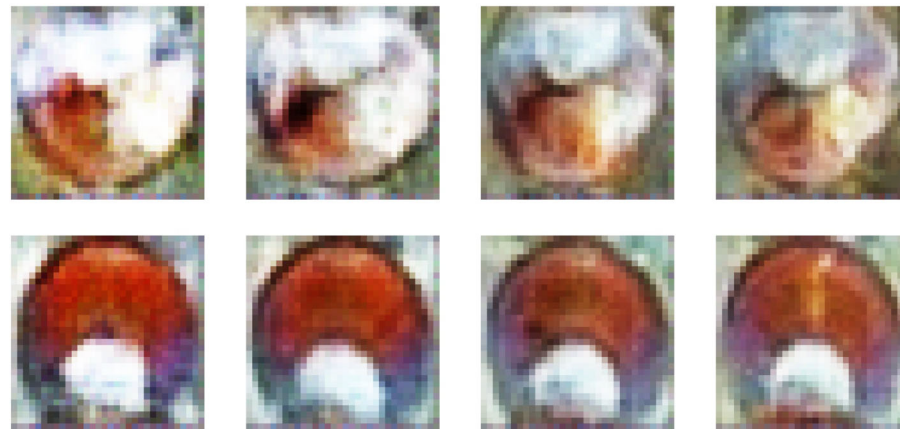


Figure 13 Flashover damage insulators generated by the conditional Wasserstein GAN with a size of 32×32 . The columns represent 500, 1000, 2500, and 5000 training epochs

the original images in most cases. This shows that when a model is not trained properly, it can impair the ability to classify insulators. The only superior result using 500 training epochs to generate images was when 5000 images from the conditional Wasserstein GAN were considered, and in this case, the large number of images meant that the model's performance was superior. The qualitative results also had this outcome, observing that when few epochs are considered, synthetic images can have worse quality. This result reinforces the importance of proper training.

In general, the use of smaller images resulted in lower performance (best F1-score equal to 0.98149) when compared to higher resolution images (best F1-score equal to 0.98986). In terms of training time, the conditional diffusion model for 256×256 images took 607.19 hours (≈ 25 days) to complete 5000 training epochs, and for 32×32 images it took 36.91 hours. In comparison, the conditional

Wasserstein GAN took 120 hours (≈ 5 days) to complete 5000 training epochs, while for 32×32 images it took 22.15 hours.

A cost-benefit analysis shows that the conditional diffusion model is advantageous only when the target is maximum classification performance and the augmentation process is performed offline, whereas conditional Wasserstein GAN offers a more economical alternative when computational budget is tighter. For high resolution 256×256 images, the baseline classifier trained only on real data achieved an F1-score of 0.96887 in 9.90 h, while the best conditional Wasserstein GAN setting increased the F1-score to 0.98953 with 5000 synthetic images generated at 5000 epochs, requiring 42.84 h of classifier training; the best conditional diffusion setting further improved the F1-score to 0.98986 with 5000 synthetic images generated

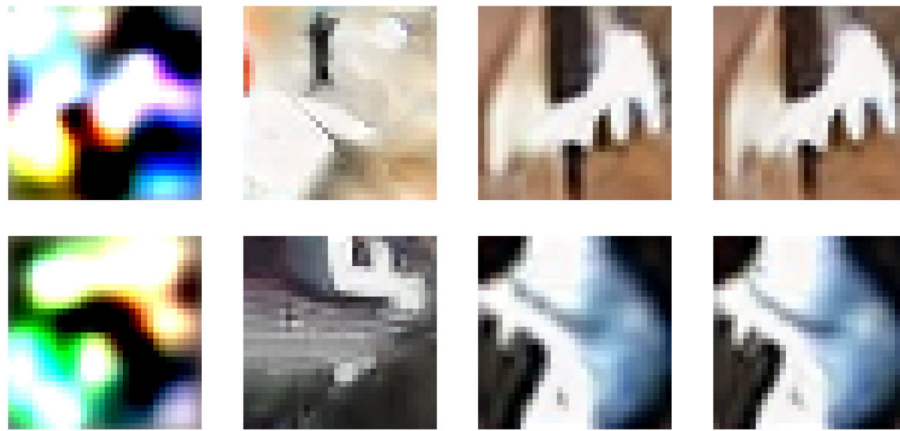


Figure 14 Broken insulators generated by a conditional diffusion model with a size of 32×32 . The columns represent 500, 1000, 2500, and 5000 training epochs

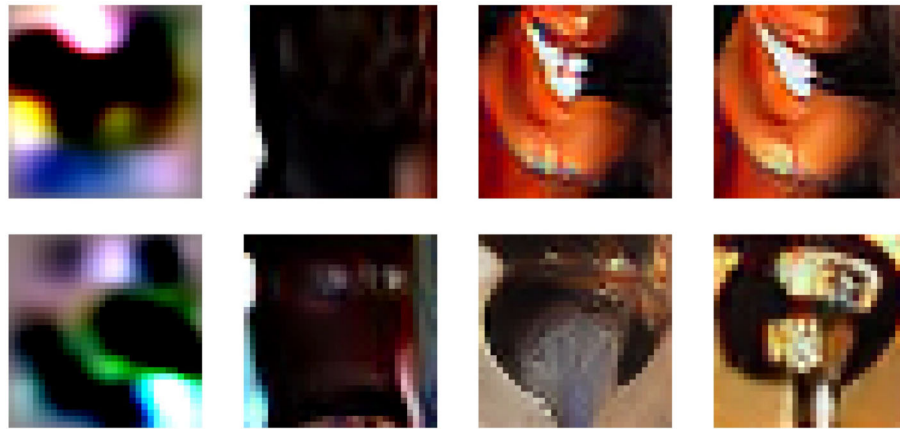


Figure 15 Flashover damage insulators generated by a conditional diffusion model with a size of 32×32 . The columns represent 500, 1000, 2500, and 5000 training epochs

at 2500 epochs, while reducing classifier training time to 18.50 h.

Diffusion provided only a marginal absolute gain of 0.00033 in F1 over the best Wasserstein GAN result, but reached that best point with substantially lower downstream training time for the classifier. For low-resolution 32×32 images, however, the advantage of diffusion becomes clearer in effectiveness but not in generation cost: the original model obtained an F1-score of 0.96887, the best Wasserstein GAN result reached 0.97999, and the best diffusion result reached 0.98149, corresponding to gains of about 1.11 and 1.26 percentage points, respectively, over the non-augmented baseline. At the same time, the manuscript explicitly states that training the conditional diffusion generator required 5.06 times longer than the conditional Wasserstein GAN for 256×256 images and

16.45 times longer for 32×32 images, which confirms that the generative stage is the main computational bottleneck.

The practical conclusion is that conditional diffusion is justified when the objective is the highest possible predictive performance and augmentation is performed once off-line, while conditional Wasserstein GAN remains a more favorable trade-off for adaptive or frequently retrained pipelines where lower augmentation cost is more important than the final fraction of a percentage point in F1-score.

5.2.4 Explainable results

While DL architectures such as CNNs and transformer-based models like Swin models have achieved promising performance, their decision-making processes often remain a black box. This lack of interpretability raises concerns about the reliability and trustworthiness of these

Table 6 Classification results using augmented data (images of 32 × 32 pixels)

Model	Synthetic images	Epochs ^a	Precision	Recall	F1-score	Training time (h)
Original	-	-	0.97692	0.96116	0.96887	9.90
Conditional Wasserstein GAN	500	500	0.95045	0.93275	0.94102	30.44
		1000	0.96351	0.93951	0.95112	36.25
		2500	0.97064	0.93759	0.95343	34.03
		5000	0.96063	0.93014	0.94476	40.34
	1000	500	0.95930	0.94886	0.95395	21.55
		1000	0.96443	0.94472	0.95415	32.47
		2500	0.96140	0.94940	0.95529	31.58
		5000	0.96751	0.95005	0.95853	42.25
	2500	500	0.96461	0.94856	0.95628	20.97
		1000	0.97813	0.95975	0.96856	36.90
		2500	0.97643	0.96523	0.97070	30.82
		5000	0.98064	0.96197	0.97096	32.48
	5000	500	0.98182	0.97309	0.97727	15.56
		1000	0.97834	0.97306	0.97562	30.95
		2500	0.98105	0.97899	0.97999	37.59
		5000	0.97907	0.97203	0.97537	18.94
Conditional Diffusion	500	500	0.91446	0.91227	0.91297	10.13
		1000	0.97684	0.95210	0.96409	20.43
		2500	0.96817	0.95069	0.95921	20.48
		5000	0.96259	0.93374	0.94762	10.07
	1000	500	0.92285	0.90712	0.91390	11.17
		1000	0.96938	0.95186	0.96040	32.71
		2500	0.96758	0.95010	0.95861	32.89
		5000	0.96613	0.95480	0.96037	10.11
	2500	500	0.92375	0.91887	0.92125	26.77
		1000	0.97690	0.96959	0.97319	37.18
		2500	0.97647	0.96975	0.97306	37.26
		5000	0.96604	0.96510	0.96556	23.28
	5000	500	0.93681	0.93332	0.93441	36.72
		1000	0.98531	0.97797	0.98149	44.88
		2500	0.98036	0.97496	0.97758	44.91
		5000	0.98114	0.97606	0.97853	42.02

^aEpochs used to create synthetic images

Best results in bold

models, particularly in high-stakes domains such as power grid infrastructure monitoring, where incorrect decisions can lead to costly operational failures or safety hazards.

To address these challenges, eXplainable Artificial Intelligence (XAI) techniques have been adopted to provide insight into model behavior. Figure 16 (for broken insulators) and Fig. 17 (flashover damage insulators) present results of different CAM methods. In these figures, the CAM methods highlight in red the areas identified by the model as activation maps, thus creating this fault interpretability.

Although the results are promising, the XGradCAM and ScoreCAM methods go beyond the limits of the fault by highlighting areas larger than the focus of analysis, so results using SHAP are evaluated. Figures 18 and 19 show the results obtained using SHAP. The blue results mean that they are features that move away from the class under

study. Values coloured red mean that they are features that are closer to the class under study.

Based on the CAM and SHAP methods, this work proposed a combination of both to improve the visualization of the fault observed in the data provided. Figure 20 (broken insulator) and Fig. 21 (flashover damage class) show the results of this fusion, where visual improvement is noticeable and, for the most part, only the focus on the class to be detected is increased.

In the proposed hybrid SHAP-CAM method, there is greater clarity in highlighting the regions of evidence, which results in a better interpretation of the results. This hybrid SHAP-CAM method outperforms other methods, as it is possible to observe the exact location of the fault, rather than the large observation area as in other CAM methods, or the difficulty in interpreting SHAP.

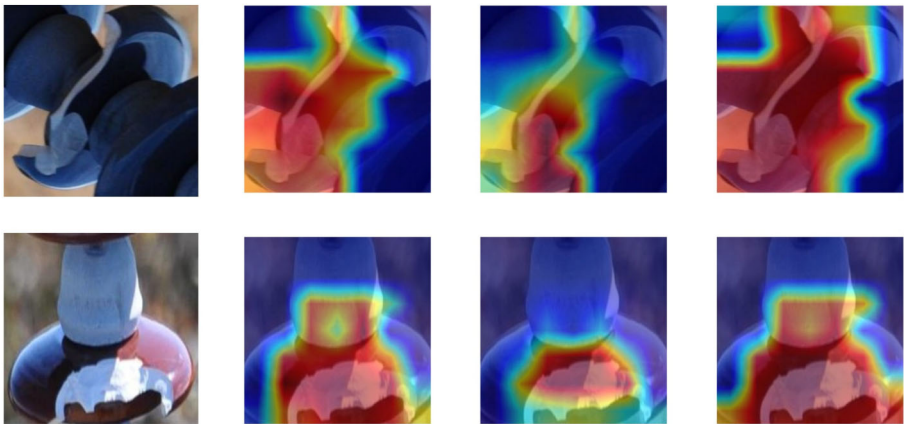


Figure 16 CAM results for broken insulators: The columns are the raw images, XGradCAM, HiResCAM, and ScoreCAM, respectively

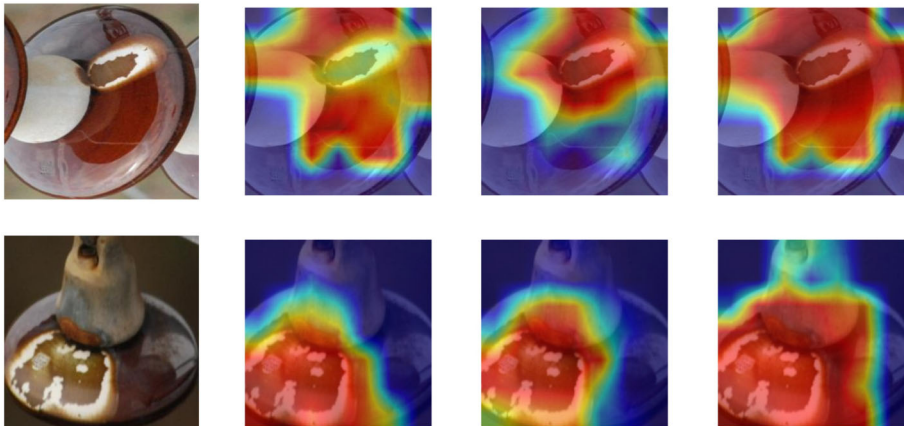


Figure 17 CAM results for flashover damage insulators: the columns are the raw images, XGradCAM, HiResCAM, and ScoreCAM, respectively

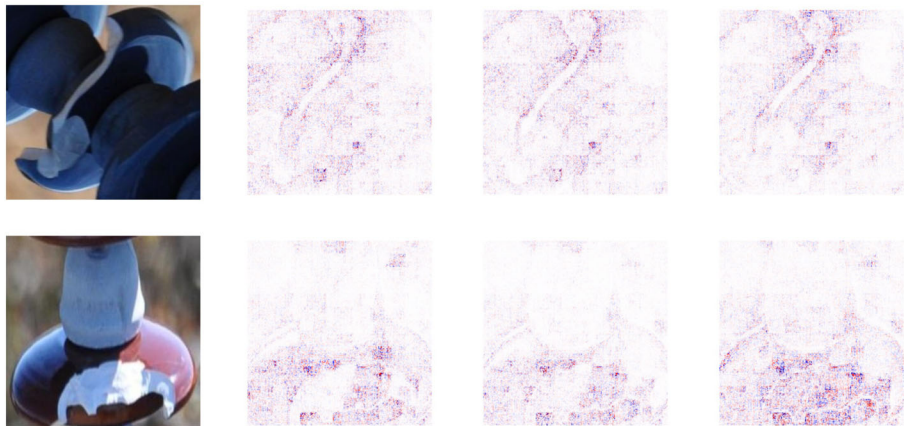


Figure 18 SHAP results for broken insulators: the columns are the raw images, the SHAP results for the broken class, the SHAP results for the flashover damage class, and the SHAP results for the no issues class

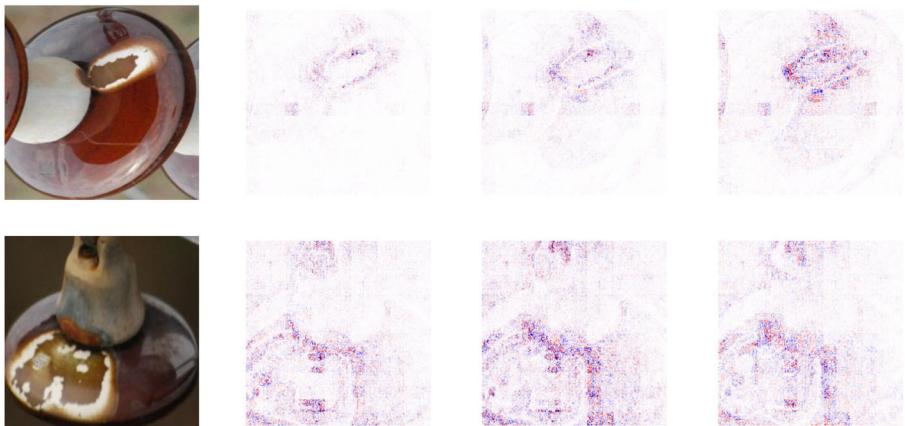


Figure 19 SHAP for flashover damage insulators: the columns are the raw images, the SHAP results for the broken class, the SHAP results for the flashover damage class and the SHAP results for the no issues class

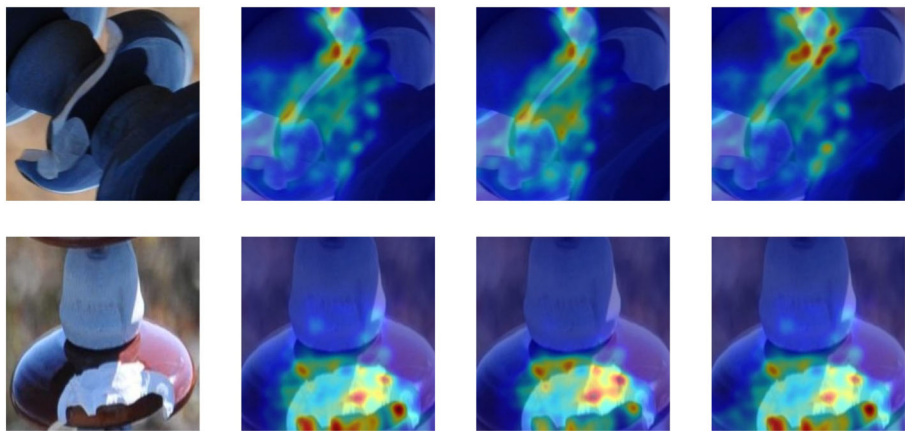


Figure 20 Proposed hybrid SHAP-CAM results for broken insulators: the columns are the raw images, SHAP-XGradCAM, SHAP-HiResCAM, and SHAP-ScoreCAM, respectively

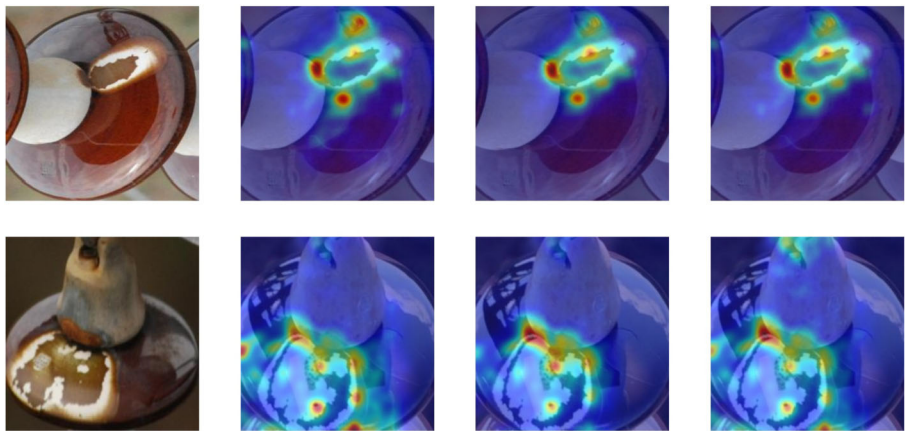


Figure 21 Proposed hybrid SHAP-CAM results for flashover damage insulators: the columns are the raw images, SHAP-XGradCAM, SHAP-HiResCAM, and SHAP-ScoreCAM, respectively

Table 7 Insulator fault detection evaluation models' prediction performance

Model	Precision	Recall	F1-score	mAP	
				[0.5]	[0.5:0.95]
DETR	0.69900	0.85619	0.76965	0.65116	0.46523
Faster R-CNN	0.41826	0.85230	0.56114	0.41211	0.26569
RetinaNet	0.27802	0.91727	0.42671	0.34637	0.23005
SSD	0.66123	0.58523	0.62091	0.32116	0.16968
STN-YOLO	0.90778	0.83811	0.87155	0.92821	0.78253
SAE-YOLO	0.88339	0.85990	0.87149	0.92579	0.79439
YOLOv10n	0.95897	0.95107	0.95452	0.97674	0.89722
YOLOv10s	0.95415	0.92386	0.93881	0.96839	0.90620
YOLOv10m	0.95047	0.95245	0.95177	0.97355	0.90680
YOLOv10l	0.94928	0.94527	0.94768	0.97151	0.90878
YOLOv10x	0.96555	0.94079	0.95315	0.97544	0.92064
YOLOv11n	0.97209	0.95530	0.96375	0.98379	0.90439
YOLOv11s	0.97219	0.95546	0.96376	0.98319	0.91813
YOLOv11m	0.97012	0.95904	0.96322	0.98219	0.92031
YOLOv11l	0.97068	0.96973	0.96996	0.98385	0.92990
YOLOv11x	0.96319	0.94591	0.95488	0.97249	0.89402
YOLOv12n	0.97189	0.95852	0.96474	0.97793	0.90716
YOLOv12s	0.96819	0.96520	0.96677	0.98173	0.92474
YOLOv12m	0.96348	0.96025	0.96258	0.97741	0.90668
YOLOv12l	0.95698	0.94239	0.95045	0.97730	0.91488
YOLOv12x	0.97624	0.94793	0.96201	0.97709	0.89832
YOLOv26n	0.97093	0.96164	0.96627	0.98018	0.91333
YOLOv26s	0.97028	0.95332	0.96173	0.98019	0.93533
YOLOv26m	0.97378	0.94267	0.95797	0.97497	0.92984
YOLOv26l	0.96329	0.96466	0.96398	0.97949	0.93174
YOLOv26x	0.96197	0.96491	0.96344	0.98340	0.93236
Ours	0.98531	0.97797	0.98149	0.98951	0.90891

Best results in bold

The results are promising both when broken insulators are analyzed and when the evaluation is carried out on insulators with the presence of surface discharges (flashover). This highlighting indication can help operators to be more assertive in classifying faults when compared to good insulators or those that have small signs of contamination.

5.3 Benchmarking

To carry out a comparative analysis, the DETR, Faster R-CNN, RetinaNet, SDD, YOLO (versions 10 to 24, sizes nano, small, medium, large, and extra large), STN-YOLO, and SAE-YOLO models are evaluated. These models are used to detect and classify the insulators simultaneously (multi-classification). Table 7 presents the performance results of this comparison; the default hyperparameters of each model are used so that our model can be compared with versions used in the literature.

In general, the YOLO versions are more promising than other models, confirming their suitability for our baseline model. All standard versions of YOLO achieve an

Table 8 Insulator fault detection evaluation of computational cost

Model	GFLOPs	Training time (h)	Inference time (ms)
DETR	234.96	2.15	14.84
Faster R-CNN	272.14	2.74	80.37
RetinaNet	245.77	2.86	65.41
SSD	91.28	1.15	22.09
STN-YOLO	10.70	7.06	5.51
SAE-YOLO	9.49	6.49	4.59
YOLOv10n	8.39	4.54	5.02
YOLOv10s	24.77	6.02	6.56
YOLOv10m	63.97	8.85	13.27
YOLOv10l	127.20	11.25	25.55
YOLOv10x	171.00	16.23	27.06
YOLOv11n	6.44	3.84	3.78
YOLOv11s	21.55	4.31	6.11
YOLOv11m	68.19	6.97	12.78
YOLOv11l	87.27	8.48	14.02
YOLOv11x	195.45	13.42	24.40
YOLOv12n	6.48	4.85	6.22
YOLOv12s	21.52	5.81	9.00
YOLOv12m	67.74	9.09	14.80
YOLOv12l	89.41	13.39	24.97
YOLOv12x	199.82	21.28	37.63
YOLOv26n	5.77	3.46	4.61
YOLOv26s	22.50	5.61	4.69
YOLOv26m	74.72	7.28	6.08
YOLOv26l	93.12	8.95	7.67
YOLOv26x	208.51	12.17	10.76
Ours	208.50	23.41	32.18

Best results in bold

mAP@[0.5] greater than 0.96 and an mAP@[0.5:0.95] greater than 0.89, showing that they are indeed suitable for the task being analyzed. The simulation time can vary as the early stop is used to avoid overfitting the model, which means that training is stopped early to ensure that the model has been trained properly.

With an F1-score of 0.98149 and mAP@[0.5] of 0.98951, our method outperforms all compared models. In terms of computational effort, models that use a greater number of parameters have higher GFLOPs, which is not a limitation, considering that the inference time is acceptable for real-time applications.

Table 8 compares various models for insulator fault detection in terms of computational cost, training time, and inference speed. Classical detectors such as DETR, Faster R-CNN, RetinaNet, and SSD show moderate to high GFLOPs, with SSD achieving the fastest training (1.15 h) and DETR the lowest inference time (14.84 ms), though Faster R-CNN and RetinaNet are significantly slower during inference.

The YOLO series demonstrates a clear trade-off between model size and efficiency, with YOLOv11n achieving the

best balance of lowest GFLOPs (6.44) and fastest inference (3.78 ms), while larger variants of YOLOv11 and YOLOv12 increase computational cost and latency.

The proposed model exhibits high complexity (208.50 GFLOPs) and long training time (23.41 h, considering object detection and classification), yet maintains competitive inference speed (32.18 ms, considering object detection and classification), suggesting that it prioritizes detection performance and robustness over training efficiency. These results indicate that lightweight YOLO variants are optimal for real-time applications, whereas the proposed model suits scenarios where performance is more critical than training cost.

Although the full model integrates diffusion-based augmentation, transformer-based classification, and explainability modules, these components are primarily confined to the training phase, while the inference pipeline relies only on the hypertuned YOLO26x detector and the Swin-V2-B classifier operating on real images. Empirical results demonstrate that the detection stage achieves inference times on the order of a few milliseconds (e.g., 14.67 ms for YOLO26x), while classification requires approximately 17.51 ms, indicating that near real-time performance is already attainable on standard hardware.

The modular architecture enables straightforward lightweighting strategies, including replacing Swin-V2-B with smaller variants such as Swin-V2-T, reducing input resolution (e.g., 32×32 crops), and selecting compact YOLO variants (nano or small), which offer favorable trade-offs between performance and latency.

Additional optimizations such as model pruning, quantization, and edge-oriented deployment (e.g., TensorRT acceleration) can further reduce computational overhead. These results, combined with the demonstrated effectiveness of low-resolution inputs and smaller models, confirm that the proposed framework is adaptable to low-power UAV edge devices while preserving high detection and classification performance, supporting its practical viability in real-time engineering applications.

5.4 Limitations

The conditional diffusion model needed 5.06 times longer to be trained than the conditional Wasserstein GAN to generate 256×256 images, which can be a problem for adaptive models. For images of 32×32 , the conditional diffusion model needed 16.45 times longer to be trained than the conditional Wasserstein GAN. For offline applications such as those considered in this work, this limitation does not represent a problem since the model's inference time is considerably faster once it has been trained.

6 Conclusion and future research

This paper has presented a novel, end-to-end approach for fault detection in high-voltage insulators by integrating

data augmentation, object detection, classification, and interpretability techniques. To address the challenge of the unbalanced dataset, the conditional diffusion model has been used to generate synthetic images of faulty insulators, achieving superior results compared to the conditional Wasserstein GAN. The model has enhanced transparency through a hybrid SHAP-CAM approach, providing intuitive visual explanations directly to field engineers, and helping the designers to better understand the strengths of the DL models.

The proposed YOLO-Swin architecture (hypertuned via Bayesian optimization) proved to have better results at both localizing insulators (mAP@[0.5] equal to 0.98951) and classifying fault conditions (F1-score=0.98149), outperforming the state-of-the-art detectors (DETR, Faster R-CNN, RetinaNet, SSD, YOLOv10–12, STN-YOLO, and SAE-YOLO) and CNN classifiers (VGG, ResNet, EfficientNet, ViT, Swin). Beyond these quantitative achievements, the proposed approach reduces the reliance on scarce real-fault images, enabling retraining to cover new insulator designs or damage conditions.

The evaluation under complex environmental conditions, such as rain, fog, and occlusion, remains an important direction for further validation of the proposed framework. However, the availability of publicly accessible datasets that comprehensively represent such challenging scenarios is currently limited, which restricts a systematic assessment in this context.

As future work, efforts could focus on incorporating more diverse and realistic environmental variations, either through the collection of new field data or through advanced data augmentation strategies, including physics-informed simulation and domain adaptation techniques. This would enable a more rigorous evaluation of model robustness and generalization capability under adverse inspection conditions, further strengthening the practical applicability of the proposed approach in real-world power grid environments.

To further strengthen the solution's real-world applicability, future work may focus on expanding generative diversity by integrating multi-source weather and imaging conditions to capture a wider array of fault manifestations. Real-time deployment on edge devices, optimizing for latency and energy efficiency so that inspections can be performed via drones or on-site cameras. Adaptive learning through online fine-tuning allows the model to incorporate newly acquired fault instances and continuously improve detection performance. Cross-domain validation across different grid infrastructures and insulator materials to ensure broad generalizability and robustness.

Acknowledgements

We would like to thank the University of Regina for providing us with the hardware needed to carry out the experiments. This work was partially

supported by FCT Fundação para a Ciência e Tecnologia, I.P. under Grants 2023.15441.TENURE.051/CP00003/CT00029 and the LASIGE Research Unit (ref. UID/00408/2025, <https://doi.org/10.54499/UID/00408/2025>), and co-financed by FCT and the PRR - Recovery and Resilience Plan by the European Union under the LASIGE Research Unit (UID/PRR/00408/2025, <https://doi.org/10.54499/UID/PRR/00408/2025>).

Author contributions

SFS and JPM-C - wrote the introduction, method, results, and conclusion sections, drew the figures, and computed the experiments; VCM, LSC, and K-CY - proofread and supervised.

Funding information

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant DDG-2024-00035, and in part by the Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) under Grant DDG-2024-00035.

Data availability

For future comparisons, the dataset is available at: <https://github.com/jpmcarvalho/Optimized-YOLO>.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details

¹Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa, Rua Conselheiro Emídio Navarro 1, 1959-007 Lisboa, Portugal. ²Faculty of Engineering and Applied Sciences, University of Regina, Regina, Saskatchewan, S4S 0A2, Canada. ³LASIGE, Departamento de Informática, Faculdade de Ciências, Universidade de Lisboa, 1749-016 Lisboa, Portugal. ⁴COPELABS, Lusófona University, Campo Grande 376, 1749-024 Lisbon, Portugal. ⁵Center of Technology and Systems (UNINOVA-CTS) and LASI, 2829-516 Caparica, Portugal. ⁶Department of Electrical Engineering, Federal University of Parana, Curitiba, PR, Brazil. ⁷Graduate Program in Mechanical Engineering, Federal University of Parana, Curitiba, PR, Brazil. ⁸Graduate Program in Electrical Engineering, Federal University of Parana, Curitiba, PR, Brazil.

Received: 7 January 2026 Revised: 30 April 2026 Accepted: 26 May 2026
Published online: 09 June 2026

References

- L.O. Seman, W.G. Buratto, G.V. Gonzalez, V.R.Q. Leithardt, A. Nied, S.F. Stefenon, Differentiable neural search architecture with zero-cost metrics for insulator fault prediction. *Results Eng.* **29**, 109716 (2026). <https://doi.org/10.1016/j.rineng.2026.109716>
- S.F. Stefenon, L.O. Seman, N.F. Sopelsa Neto, L.H. Meyer, V.C. Mariani, L.d.S. Coelho, Group method of data handling using Christiano-Fitzgerald random walk filter for insulator fault prediction. *Sensors* **23**(13), 6118 (2023). <https://doi.org/10.3390/s23136118>
- Y. Xuan, X.Y. Zhang, C. Li, H. Wang, M. Chaoxu, LAM-YOLOv11 for UAV transmission line inspection: overcoming environmental challenges with enhanced detection efficiency. *Multimed. Syst.* **31**(110), 1–13 (2025). <https://doi.org/10.1007/s00530-025-01702-y>
- Q. Zhang, J. Zhang, Y. Li, C. Zhu, G. Wang, ID-YOLO: a multimodule optimized algorithm for insulator defect detection in power transmission lines. *IEEE Trans. Instrum. Meas.* **74**, 1–11 (2025). <https://doi.org/10.1109/TIM.2025.3527530>
- H. Manninen, C.J. Ramlal, A. Singh, S. Rocke, J. Kilter, M. Landsberg, Toward automatic condition assessment of high-voltage transmission infrastructure using deep learning techniques. *Int. J. Electr. Power Energy Syst.* **128**, 106726 (2021). <https://doi.org/10.1016/j.ijepes.2020.106726>
- M. Mishra, J.G. Singh, A comprehensive review on deep learning techniques in power system protection: trends, challenges, applications and future directions. *Results Eng.* **25**, 103884 (2025). <https://doi.org/10.1016/j.rineng.2024.103884>
- R. Aitelhaj, B.-E. Benelmostafa, H. Medromi, Exploring the generalizability of recent object detection models in identifying defective glass insulators for UAV power line inspection a case study in Morocco, in *International Conference on Advanced Intelligent Systems for Sustainable Development* (Springer, Marrakech, 2023), pp. 301–311. https://doi.org/10.1007/978-3-031-54288-6_29
- S. Falahatnejad, A. Karami, H. Nezamabadi-pour, PTSRGAN: power transmission lines single image super-resolution using a generative adversarial network. *Int. J. Electr. Power Energy Syst.* **155**, 109607 (2024). <https://doi.org/10.1016/j.ijepes.2023.109607>
- S.F. Stefenon, M. Cristoforetti, A. Cimatti, Conditional diffusion to enhance performance of object detection in unbalanced data engineering drawings. *Neural Comput. Appl.* **37**, 20335–20364 (2025). <https://doi.org/10.1007/s00521-025-11458-1>
- P. Dhariwal, A. Nichol, Diffusion models beat GANs on image synthesis. *Adv. Neural Inf. Process. Syst.* **34**, 8780–8794 (2021)
- Y. Tian, Q. Ye, D. Doermann, YOLOv12: attention-centric real-time object detectors, **1**, 1–13 (2025). <https://doi.org/10.48550/arXiv.2502.12524>. Preprint. [arXiv:2502.12524](https://arxiv.org/abs/2502.12524)
- Z. Huang, H. Wang, Y. Tang, F. Gao, B. Du, J. Wang, A two-stage location-allocation optimization method for fixed UAV nests in power inspection considering node failure scenarios. *Sensors* **25**(4), 1089 (2025). <https://doi.org/10.3390/s25041089>
- H.-G. Yeh, R. Nguyen, S.-C. Kwon, Object classification with YOLOv5 for electric utility asset inspection using UAVs, in *International Conference on Artificial Intelligence in Information and Communication* (IEEE, Osaka, 2024), pp. 820–825. <https://doi.org/10.1109/ICAICI60209.2024.10463428>
- Y.-J. Cha, R. Ali, J. Lewis, O. Büyükoztürk, Deep learning-based structural health monitoring. *Autom. Constr.* **161**, 105328 (2024). <https://doi.org/10.1016/j.autcon.2024.105328>
- M. Zhou, T. Wu, Z. Xia, B. He, L. Bing Kong, H. Su, Research progress in deep learning for ceramics surface defect detection. *Measurement* **242**, 115956 (2025). <https://doi.org/10.1016/j.measurement.2024.115956>
- P. Jiang, D. Ergu, F. Liu, Y. Cai, B. Ma, A review of YOLO algorithm developments. *Proc. Comput. Sci.* **199**, 1066–1073 (2022). <https://doi.org/10.1016/j.procs.2022.01.135>
- J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection (2015). <https://doi.org/10.48550/arXiv.1506.02640>. arXiv preprint. [arXiv:1506.02640](https://arxiv.org/abs/1506.02640)
- J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, Las Vegas, 2016), pp. 779–788. <https://doi.org/10.1109/CVPR.2016.91>
- Y. Tian, Q. Ye, D. Doermann, YOLOv12: attention-centric real-time object detectors (2025). <https://doi.org/10.48550/arXiv.2502.12524>. arXiv preprint. [arXiv:2502.12524](https://arxiv.org/abs/2502.12524)
- J. Liu, M. Hu, J. Dong, X. Lu, Summary of insulator defect detection based on deep learning. *Electr. Power Syst. Res.* **224**, 109688 (2023). <https://doi.org/10.1016/j.epsr.2023.109688>
- R. Sapkota, R.H. Cheppally, A. Sharda, M. Karkee, YOLO26: key architectural enhancements and performance benchmarking for real-time object detection, arXiv preprint (2026). <https://doi.org/10.48550/arXiv.2509.25164>. [arXiv:2509.25164](https://arxiv.org/abs/2509.25164)
- J.P.M. Carvalho, S.F. Stefenon, V.R.Q. Leithardt, L.O. Seman, K.-C. Yow, J.F.D.P. Santana, Input attention, squeeze and excitation, and spatial transformer of YOLO for fault detection using UAV. *Ain Shams Eng. J.* **17**(3), 104067 (2026). <https://doi.org/10.1016/j.jasej.2026.104067>
- N. Jegham, C.Y. Koh, M. Abdelatti, A. Hendawi, YOLO Evolution: a comprehensive benchmark and architectural review of YOLOv12, YOLOv11, and their previous versions, arXiv preprint (2024). <https://doi.org/10.48550/arXiv.2411.0020>. [arXiv:2411.0020](https://arxiv.org/abs/2411.0020)
- S.F. Stefenon, G. Singh, B.J. Souza, R.Z. Freire, K.-C. Yow, Optimized hybrid YOLOu-quasi-ProtoPNet for insulators classification. *IET Gener. Transm. Distrib.* **17**(15), 3501–3511 (2023). <https://doi.org/10.1049/gtd2.12886>
- J. Song, X. Qin, J. Lei, J. Zhang, Y. Wang, Y. Zeng, A fault detection method for transmission line components based on synthetic dataset and improved YOLOv5. *Int. J. Electr. Power Energy Syst.* **157**, 109852 (2024). <https://doi.org/10.1016/j.ijepes.2024.109852>
- D. Sadykova, D. Pernebayeva, M. Bagheri, A. James, IN-YOLO: real-time detection of outdoor high voltage insulators using UAV imaging. *IEEE*

- Trans. Power Deliv. **35**(3), 1599–1601 (2020). <https://doi.org/10.1109/TPWRD.2019.2944741>
27. Z. Cao, K. Chen, J. Chen, Z. Chen, M. Zhang, CACS-YOLO: a lightweight model for insulator defect detection based on improved YOLOv8m. *IEEE Trans. Instrum. Meas.* **73**, 1–10 (2024). <https://doi.org/10.1109/TIM.2024.3453332>
 28. L. Ding, Z.Q. Rao, B. Ding, S.J. Li, Research on defect detection method of railway transmission line insulators based on GC-YOLO. *IEEE Access* **11**, 102635–102642 (2023). <https://doi.org/10.1109/ACCESS.2023.3316266>
 29. W. Yi, S. Ma, R. Li, Insulator and defect detection model based on improved YOLO-S. *IEEE Access* **11**, 93215–93226 (2023). <https://doi.org/10.1109/ACCESS.2023.3309693>
 30. Y. Liu, X. Huang, D. Liu, Weather-domain transfer-based attention YOLO for multi-domain insulator defect detection and classification in UAV images. *Entropy* **26**(2), 136 (2024). <https://doi.org/10.3390/e26020136>
 31. S.F. Stefenon, L.O. Seman, A.C.R. Klaar, R.G. Ovejero, V.R.Q. Leithardt, Hypertuned-YOLO for interpretable distribution power grid fault location based on EigenCAM. *Ain Shams Eng. J.* **15**(6), 102722 (2024). <https://doi.org/10.1016/j.jasej.2024.102722>
 32. R. Jiao, J. Liu, K. Li, R. Qiao, Y. Liu, W. Zhang, YOLO-DTAD: dynamic task alignment detection model for multi-category power defects image. *IEEE Trans. Instrum. Meas.* **74**, 1–14 (2025). <https://doi.org/10.1109/TIM.2025.3541692>
 33. Y. Wang, X. Song, L. Feng, Y. Zhai, Z. Zhao, S. Zhang, Q. Wang, MCI-GLA plug-in suitable for YOLO series models for transmission line insulator defect detection. *IEEE Trans. Instrum. Meas.* **73**, 1–12 (2024). <https://doi.org/10.1109/TIM.2024.3385817>
 34. J. Li, H. Zhou, G. Lv, J. Chen, A2MADA-YOLO: attention alignment multiscale adversarial domain adaptation YOLO for insulator defect detection in generalized foggy scenario. *IEEE Trans. Instrum. Meas.* **74**, 1–19 (2025). <https://doi.org/10.1109/TIM.2025.3541814>
 35. M. He, L. Qin, X. Deng, K. Liu, MFI-YOLO: multi-fault insulator detection based on an improved YOLOv8. *IEEE Trans. Power Deliv.* **39**(1), 168–179 (2024). <https://doi.org/10.1109/TPWRD.2023.3328178>
 36. Y. Jing, T. Huang, L. Gao, J. Deng, Insulator detection based on FA-YOLO network with improved feature extraction ability. *IET Image Process.* **18**(12), 3600–3616 (2024). <https://doi.org/10.1049/ipr2.13197>
 37. Z. Li, Q. Qin, Y. Yang, X. Mai, Y. Ieiri, O. Yoshie, An enhanced substation equipment detection method based on distributed federated learning. *Int. J. Electr. Power Energy Syst.* **166**, 110547 (2025). <https://doi.org/10.1016/j.jepes.2025.110547>
 38. S. Li, M. Wang, Y. Zhou, Q. Su, L. Liu, T. Wu, IG-YOLOv8: insulator guardian based on YOLO for insulator fault detection. *IEEJ Trans. Electr. Electron. Eng.* **20**(4), 537–547 (2025). <https://doi.org/10.1002/tee.24221>
 39. J. Wang, L. Cheng, An improved YOLOv8-XGBoost load rapid identification method based on multi-feature fusion. *Int. J. Electr. Power Energy Syst.* **166**, 110573 (2025). <https://doi.org/10.1016/j.jepes.2025.110573>
 40. Q. Zhang, J. Zhang, Y. Li, C. Zhu, G. Wang, IL-YOLO: an efficient detection algorithm for insulator defects in complex backgrounds of transmission lines. *IEEE Access* **12**, 14532–14546 (2024). <https://doi.org/10.1109/ACCESS.2024.3358205>
 41. Y. Li, D. Feng, Q. Zhang, S. Li, HRD-YOLOX based insulator identification and defect detection method for transmission lines. *IEEE Access* **12**, 22649–22661 (2024). <https://doi.org/10.1109/ACCESS.2024.3363430>
 42. C. Liu, S. Wei, S. Zhong, F. Yu, YOLO-PowerLite: a lightweight YOLO model for transmission line abnormal target detection. *IEEE Access* **12**, 105004–105015 (2024). <https://doi.org/10.1109/ACCESS.2024.3434687>
 43. A.T. Hien, A.D. Tran, D.C. Viet, Q.D.T. Thuy, Q.N. Huu, Insulator defect detection based on feature pyramid network and diffusion model. *Signal Image Video Process.* **19**(5), 1–11 (2025). <https://doi.org/10.1007/s11760-025-03960-9>
 44. T. Wang, N. Zhang, W. Zhang, W. Yang, W. Zhang, Smart grid insulator detection network improved based on YOLOv8. *IEEE Latin Am. Trans.* **23**(2), 125–134 (2025). <https://doi.org/10.1109/TLA.2025.10851364>
 45. Y. Ji, T. Ma, H. Shen, H. Feng, Z. Zhang, D. Li, Y. He, Transmission line defect detection algorithm based on improved YOLOv12. *Electronics* **14**(12), 2432 (2025). <https://doi.org/10.3390/electronics14122432>
 46. S.F. Stefenon, L.O. Seman, G. Singh, K.-C. Yow, Enhanced insulator fault detection using optimized ensemble of deep learning models based on weighted boxes fusion. *Int. J. Electr. Power Energy Syst.* **168**, 110682 (2025). <https://doi.org/10.1016/j.jepes.2025.110682>
 47. R. Ye, A. Boukerche, X.-S. Yu, C. Zhang, B. Yan, X.-J. Zhou, Data augmentation method for insulators based on cycle GAN. *J. Electron. Sci. Technol.* **22**(2), 100250 (2024). <https://doi.org/10.1016/j.jnlest.2024.100250>
 48. Z.-D. Zhang, B. Zhang, Z.-C. Lan, H.-C. Liu, D.-Y. Li, L. Pei, W.-X. Yu, FINet: an insulator dataset and detection benchmark based on synthetic fog and improved YOLOv5. *IEEE Trans. Instrum. Meas.* **71**, 1–8 (2022). <https://doi.org/10.1109/TIM.2022.3194909>
 49. C. Sampedro, J. Rodriguez-Vazquez, A. Rodriguez-Ramos, A. Carrio, P. Campoy, Deep learning-based system for automatic recognition and diagnosis of electrical insulator strings. *IEEE Access* **7**, 101283–101308 (2019). <https://doi.org/10.1109/ACCESS.2019.2931144>
 50. W. Chang, G. Yang, J. Yu, Z. Liang, Real-time segmentation of various insulators using generative adversarial networks. *IET Comput. Vis.* **12**(5), 596–602 (2018). <https://doi.org/10.1049/iet-cvi.2017.0591>
 51. W. Chang, G. Yang, Z. Wu, Z. Liang, Learning insulators segmentation from synthetic samples, in *International Joint Conference on Neural Networks (IJCNN)*, vol. 1 (IEEE, Rio de Janeiro, 2018), pp. 1–7. <https://doi.org/10.1109/IJCNN.2018.8489142>
 52. Z. Gao, G. Yang, E. Li, T. Shen, Z. Wang, Y. Tian, H. Wang, Z. Liang, Insulator segmentation for power line inspection based on modified conditional generative adversarial network. *J. Sens.* **2019**(1), 4245329 (2019). <https://doi.org/10.1155/2019/4245329>
 53. D. Kim, S. Majumder, L. Xie, Line-post insulator fault classification model using deep convolutional gan-based synthetic images, in *2023 North American Power Symposium (NAPS)*, vol. 1 (IEEE, Asheville, 2023), pp. 1–6. <https://doi.org/10.1109/NAPS58826.2023.10318616>
 54. Z. Chen, H. Wang, D. Mao, Y. Yan, H. Rao, J. Wang, Insulator defect detection in power transmission and transformation based on the diffusion model, in *2023 IEEE 7th Conference on Energy Internet and Energy System Integration (EI2)* (IEEE, Hangzhou, 2023), pp. 5248–5252. <https://doi.org/10.1109/EI259745.2023.10512748>
 55. D. Li, Y. Lu, Q. Gao, X. Li, X. Yu, Y. Song, LiteYOLO-ID: a lightweight object detection network for insulator defect detection. *IEEE Trans. Instrum. Meas.* **73**, 1–12 (2024). <https://doi.org/10.1109/TIM.2024.3418082>
 56. Z. Zhou, C. Zhang, M. Xie, B. Cao, Classification method of composite insulator surface image based on GAN and CNN. *IEEE Trans. Dielectr. Electr. Insul.* **31**(4), 2242–2251 (2024). <https://doi.org/10.1109/TDEI.2024.3355031>
 57. Y. Liu, X. Huang, Efficient cross-modality insulator augmentation for multi-domain insulator defect detection in UAV images. *Sensors* **24**(2), 428 (2024). <https://doi.org/10.3390/s24020428>
 58. Y. Yang, S. Yang, C. Li, Y. Wang, X. Pi, Y. Lu, R. Wu, Insulator defect detection under extreme weather based on synthetic weather algorithm and improved YOLOv7. *High Volt.* **10**(1), 69–77 (2024). <https://doi.org/10.1049/hve2.12513>
 59. D. Jiang, Y. Cao, Q. Yang, Weakly-supervised learning based automatic augmentation of aerial insulator images. *Expert Syst. Appl.* **242**, 122739 (2024). <https://doi.org/10.1016/j.eswa.2023.122739>
 60. L. Zhang, L. Wang, Z. Yan, Z. Jia, H. Wang, X. Tang, Star generative adversarial VGG network-based sample augmentation for insulator defect detection. *Int. J. Comput. Intell. Syst.* **17**(1), 141 (2024). <https://doi.org/10.1007/s44196-024-00524-6>
 61. P. Ning, J. Jin, Y. Xu, C. Kong, C. Zhang, D. Tang, J. Huang, Z. Xu, T. Li, Enhanced detection of glass insulator defects using improved generative modeling and faster RCNN. *Proc. CIRP* **129**, 31–36 (2024). <https://doi.org/10.1016/j.procir.2024.10.007>
 62. R. Akella, S.K. Gunturi, D. Sarkar, Enhancing power line insulator health monitoring with a hybrid generative adversarial network and YOLO3 solution. *Tsinghua Sci. Technol.* **29**(6), 1796–1809 (2024). <https://doi.org/10.26599/TST.2023.9010137>
 63. B. Liu, W. Jiang, LA-YOLO: bidirectional adaptive feature fusion approach for small object detection of insulator self-explosion defects. *IEEE Trans. Power Deliv.* **39**(6), 3387–3397 (2024). <https://doi.org/10.1109/TPWRD.2024.3467915>
 64. J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González, A comprehensive review of YOLO architectures in computer vision: from YOLOV1 to YOLOV8 and YOLO-NAS. *Mach. Learn. Knowl. Extr.* **5**(4), 1680–1716 (2023). <https://doi.org/10.3390/make5040083>
 65. S.F. Stefenon, M. Cristoforetti, A. Cimatti, Towards automatic digitalization of railway engineering schematics, in *AlxIA 2023 – Advances in Artificial Intelligence*, vol. 22 (Springer, Rome, 2023), pp. 453–466. https://doi.org/10.1007/978-3-031-47546-7_31

66. Y. Liu, P. Luo, YOLO-TS: a lightweight YOLO model for traffic sign detection. *IEEE Access* **12**, 169013–169023 (2024). <https://doi.org/10.1109/ACCESS.2024.3498057>
67. X. Huang, J. Zhu, Y. Huo, SSA-YOLO: an improved YOLO for hot-rolled strip steel surface defect detection. *IEEE Trans. Instrum. Meas.* **73**, 1–17 (2024). <https://doi.org/10.1109/TIM.2024.3488136>
68. S.F. Stefenon, M. Cristoforetti, A. Cimatti, Automatic digitalization of railway interlocking systems engineering drawings based on hybrid machine learning methods. *Expert Syst. Appl.* **281**, 127532 (2025). <https://doi.org/10.1016/j.eswa.2025.127532>
69. Y. Xi, K. Zhou, L.-W. Meng, B. Chen, H.-M. Chen, J.-Y. Zhang, Transmission line insulator defect detection based on Swin transformer and context. *Mach. Intell. Res.* **20**(5), 729–740 (2023). <https://doi.org/10.1007/s11633-022-1355-y>
70. K. Pinasthika, B.S.P. Laksono, R.B.P. Irsal, S. Shabiyya, N. Yudistira, Sparseswin: Swin transformer with sparse transformer block. *Neurocomputing* **580**, 127433 (2024). <https://doi.org/10.1016/j.neucom.2024.127433>
71. H. Geng, T. Chen, Y. Ma, B. Zhu, B. Yu, Ptp: physical design tool parameter tuning via multi-objective Bayesian optimization. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **42**(1), 178–189 (2023). <https://doi.org/10.1109/TCAD.2022.3167858>
72. J. Nayak, B. Naik, P.B. Dash, S. Vimal, S. Kadry, Hybrid Bayesian optimization hypertuned catboost approach for malicious access and anomaly detection in IoT nomalyframework. *Sustain. Comput. Inform. Syst.* **36**, 100805 (2022). <https://doi.org/10.1016/j.suscom.2022.100805>
73. X. Wang, Y. Jin, S. Schmitt, M. Olhofer, Recent advances in Bayesian optimization. *ACM Comput. Surv.* **55**(13s), 1–36 (2023). <https://doi.org/10.1145/3582078>
74. R. Guo, B. Yang, Y. Guo, H. Li, Z. Li, B. Zhou, B. Hong, F. Wang, Machine learning-based prediction of outdoor thermal comfort: combining Bayesian optimization and the SHAP model. *Build. Environ.* **254**, 111301 (2024). <https://doi.org/10.1016/j.buildenv.2024.111301>
75. B.A. Kyem, J.K. Asamoah, Y. Huang, A. Aboah, Weather-adaptive synthetic data generation for enhanced power line inspection using StarGAN. *IEEE Access* **12**, 193882–193901 (2024). <https://doi.org/10.1109/ACCESS.2024.3520120>
76. C. Dewi, Generative adversarial network for synthetic image generation method: review, analysis, and perspective. *Appl. Gen. AI* **1**, 91–116 (2024). https://doi.org/10.1007/978-3-031-46238-2_5
77. L. She, Y. Fan, M. Xu, J. Wang, J. Xue, J. Ou, Insulator breakage detection utilizing a convolutional neural network ensemble implemented with small sample data augmentation and transfer learning. *IEEE Trans. Power Deliv.* **37**(4), 2787–2796 (2022). <https://doi.org/10.1109/TPWRD.2021.3116600>
78. F.-A. Croitoru, V. Hondru, R.T. Ionescu, M. Shah, Diffusion models in vision: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(9), 10850–10869 (2023). <https://doi.org/10.1109/TPAMI.2023.3261988>
79. L. Yang, Z. Zhang, Y. Song, S. Hong, R. Xu, Y. Zhao, W. Zhang, B. Cui, M.-H. Yang, Diffusion models: a comprehensive survey of methods and applications. *ACM Comput. Surv.* **56**(4), 1–39 (2023). <https://doi.org/10.1145/3626235>
80. Z. Xu, S. Wang, X.-L. Zhang, G. He, Optimal sensor placement for ensemble-based data assimilation using gradient-weighted class activation mapping. *J. Comput. Phys.* **514**, 113224 (2024). <https://doi.org/10.1016/j.jcp.2024.113224>
81. S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in *Advances in Neural Information Processing Systems*, vol. 31, ed. by I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, R. Garnett (2017), pp. 4765–4774
82. J. Kwak, Y. Lee, M. Choi, S. Lee, Deep learning based approaches to enhance energy efficiency in autonomous driving systems. *Energy* **307**, 132625 (2024). <https://doi.org/10.1016/j.energy.2024.132625>
83. Y. Kim, Y. Kim, Explainable heat-related mortality with random forest and Shapley additive explanations (SHAP) models. *Sustain. Cities Soc.* **79**, 103677 (2022). <https://doi.org/10.1016/j.scs.2022.103677>
84. K.H. Sun, H. Huh, B.A. Tama, S.Y. Lee, J.H. Jung, S. Lee, Vision-based fault diagnostics using explainable deep learning with class activation maps. *IEEE Access* **8**, 129169–129179 (2020). <https://doi.org/10.1109/ACCESS.2020.3009852>
85. R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, GRAD-CAM: visual explanations from deep networks via gradient-based localization. *Int. J. Comput. Vis.* **128**, 336–359 (2020). <https://doi.org/10.1007/s11263-019-01228-7>
86. R.L. Draelos, L. Carin, Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks, arXiv preprint (2021). <https://doi.org/10.48550/arXiv.2011.08891>. arXiv:2011.08891
87. H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, X.H. Ding, P. Mardziel, X.L. Hu, Score-cam: score-weighted visual explanations for convolutional neural networks, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (IEEE, Seattle, 2020), pp. 24–25. <https://doi.org/10.1109/CVPRW50498.2020.00020>
88. W. Zhao, J. Chen, X. Liu, J. Liu, Enhancing object detection through global collaborative learning. *Auton. Intell. Syst.* **5**(1), 29 (2025). <https://doi.org/10.1007/s43684-025-00114-z>
89. Y. Li, N. Miao, L. Ma, F. Shuang, X. Huang, Transformer for object detection: review and benchmark. *Eng. Appl. Artif. Intell.* **126**, 107021 (2023). <https://doi.org/10.1016/j.engappai.2023.107021>
90. J. Ou, J. Wang, J. Xue, J. Wang, X. Zhou, L. She, Y. Fan, Infrared image target detection of substation electrical equipment using an improved faster R-CNN. *IEEE Trans. Power Deliv.* **38**(1), 387–396 (2023). <https://doi.org/10.1109/TPWRD.2022.3191694>
91. V. Teju, K.V. Sowmya, S.R. Kandula, A. Stan, O.P. Stan, A hybrid retina net classifier for thermal imaging. *Appl. Sci.* **13**(14), 8525 (2023). <https://doi.org/10.3390/app13148525>
92. X. Miao, X. Liu, J. Chen, S. Zhuang, J. Fan, H. Jiang, Insulator detection in aerial images for transmission line inspection using single shot multibox detector. *IEEE Access* **7**, 9945–9956 (2019). <https://doi.org/10.1109/ACCESS.2019.2891123>
93. B. Vigneshwaran, R. Maheswari, L. Kalaivani, V. Shanmuganathan, S. Rho, S. Kadry, M.Y. Lee, Recognition of pollution layer location in 11 kv polymer insulators used in smart power grid using dual-input VGG convolutional neural network. *Energy Rep.* **7**, 7878–7889 (2021). <https://doi.org/10.1016/j.jegyr.2020.12.044>
94. S. Wang, Y. Liu, Y. Qing, C. Wang, T. Lan, R. Yao, Detection of insulator defects with improved ResNeSt and region proposal network. *IEEE Access* **8**, 184841–184850 (2020). <https://doi.org/10.1109/ACCESS.2020.3029857>
95. A. Swetapadma, A. Yadav, Transfer learning based EfficientNet method for transmission line insulator flaw detection. *e-Prime - Adv. Electr. Eng. Electron. Energy* **10**, 100873 (2024). <https://doi.org/10.1016/j.prime.2024.100873>
96. G.A.S. Surek, L.O. Seman, S.F. Stefenon, V.C. Mariani, L.d.S. Coelho, Video-based human activity recognition using deep learning approaches. *Sensors* **23**(14), 6384 (2023). <https://doi.org/10.3390/s23146384>
97. M.M. Saad, R. O'Reilly, M.H. Rehmani, A survey on training challenges in generative adversarial networks for biomedical image analysis. *Artif. Intell. Rev.* **57**(2), 19 (2024). <https://doi.org/10.1007/s10462-023-10624-y>
98. D. Lewis, P. Kulkarni, Insulator Defect Detection. *IEEE Dataport* (2021). <https://ieee-dataport.org/competitions/insulator-defect-detection>. Accessed on March 15, 2025

Publisher's note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.