Electrical insulator defect detection with incomplete annotations and imbalanced samples

Fengqian Pang¹, Chunyue Lei¹, and jingsheng zeng¹

¹North China University of Technology

August 25, 2023

Abstract

Insulators are one of the key components in high-voltage power systems that prevent transmission lines from grounding. Since they are exposed to different kinds of harsh environments and climates, periodic inspection is indispensable for the safety and high quality of power grid. Nowadays, Unmanned Aerial Vehicle (UAV) inspection is more widely used, facilitating incorporation of CNN-based detectors in the insulator detection task. However, these methods are generally based on the assumption that the image samples are balanced among different categories and possess completely ideal annotations. The problem of sample imbalance or incomplete annotation is rarely investigated in depth for insulator defect detection. In this paper, we focus on insulator defect detection with imbalanced data and incomplete annotations. Our proposed framework, named Pi-Index, introduces Positive Unlabeled (PU) learning to solve the problem of incomplete annotation and designs a novel index the class prior, which is a key parameter in PU learning. Moreover, focal loss is integrated in our framework to alleviate the effect of sample imbalance. Experiment results demonstrate that the proposed framework achieves better performance than the baseline methods in situations of sample imbalance and missing annotation.

DOI: xxx

Type: xxx

Electrical insulator defect detection with incomplete annotations and imbalanced samples

Fengqian Pang^{1, *}, Chunyue Lei¹, and Jingsheng Zeng¹

¹School of Information Science and Technology, North China University of Technology, Beijing, China

*Corresponding Author: fqpang@ncut.edu.cn

Received: XX Month 202X; Accepted: XX Month 202X

Abstract: Insulators are one of the key components in high-voltage power systems that prevent transmission lines from grounding. Since they are exposed to different kinds of harsh environments and climates, periodic inspection is indispensable for the safety and high quality of power grid. Nowadays, Unmanned Aerial Vehicle (UAV) inspection is more widely used, facilitating incorporation of CNN-based detectors in the insulator detection task. However, these methods are generally based on the assumption that the image samples are balanced among different categories and possess completely ideal annotations. The problem of sample imbalance or incomplete annotation is rarely investigated in depth for insulator defect detection. In this paper, we focus on insulator defect detection with imbalanced data and incomplete annotations. Our proposed framework, named Pi-Index, introduces Positive Unlabeled (PU) learning to solve the problem of incomplete annotation and designs a novel index the class prior, which is a key parameter in PU learning. Moreover, focal loss is integrated in our framework to alleviate the effect of sample imbalance. Experiment results demonstrate that the proposed framework achieves better performance than the baseline methods in situations of sample imbalance and missing annotation. **Keywords:** Insulator defect detection; Transmission line; Power system; Sample imbalance; Incomplete annotation

1 Introduction

Transmission of electricity throughout the power grid is accomplished with the assistance of high-voltage transmission lines, electrical insulators, and power towers. Although the insulators therein are not directly responsible for delivering electricity, they suspend the overhead transmission lines and prevent the transmission lines' grounding [1]–[5]. Furthermore, they must constantly withstand the supply voltage and bear the load of the transmission lines' gravity. Therefore, the insulator is an essential component in a high-voltage power system, which facilitates maintaining the safety and stabilization of the power grid.

Because of the extensive deployment of the power grid, the insulators need to be adaptable to different kinds of natural environments and geographic conditions. A fairly large number of them are exposed to an outdoor environment throughout the entire year, which is more susceptible to the erosion caused by harsh climates such as sunlight, rain, and snow [6]. Besides, overvoltage shocks from lightning and on-off operations, mechanical load, weight of wires, as well as metal accessories may also make the insulators more easily prone to self-blast or breakage [7]. The above factors inevitably give rise to the insulators' defects and reduce their service lifetimes. Moreover, aged and defective insulators may cause regional grid failures and enormous economic losses if they are not periodically inspected [2].

One solution is manual inspection of power equipment, but this traditional method is considered lowefficiency, labor-intensive, and unsafe. It is incapable of providing quick feedback on the condition of insulators and meeting the inspection needs of a modern smart grid, namely more frequent inspection, repair, and maintenance [8]. Due to the low cost, miniaturization, and high mobility of Unmanned Aerial Vehicle (UAV) inspection, it has replaced manual inspection and become the mainstream inspection scheme for power equipment [9]–[12]. For insulator defect detection, the UAV inspection collects a large number of aerial images with insulators. The collected insulator images not only enable the automatic identification of the insulator defects, but also make it an urgent problem to be solved.

Early algorithms for insulator defect detection are based on handcrafted features and machine learning technologies [1], [13], [14]. However, feature design is time-consuming and costly, and requires the assistance of experienced experts. In recent years, deep learning technology has made a breakthrough in image classification [6], object detection [15], image segmentation [16], etc. The deep learning-based detectors were introduced in this field [17]–[20], including both You Only Look Once (YOLO) family [21]–[24] and Region-based CNN (RCNN) series [25]–[27] detectors. On the one hand, the YOLO-based detectors were investigated to enhance the speed of insulator detection by incorporating a lightweight backbone, such as, replacing the backbone by MobileNet [28] in YOLO v3 [23] and YOLO v4 [24], and choosing suitable backbone among the four versions of YOLO v5 [9]. Simultaneously, attention mechanisms have been introduced into insulator detection research. In detail, channel-wise self-attention was merged with TinyYOLO v4 to facilitate feature representation [10], while YOLO v5 pipeline was incorporated with a triplet attention module in [9] and a Convolutional Block Attention Module (CBAM) in [29] for providing more context information for insulator defect detection.

On the other hand, Faster RCNN, a two-stage detector, was introduced into the insulator detection community [17], [30]. It first roughly generates the proposals of insulators and then refines the proposals for locating the insulators' defects. Moreover, Zhong et al. modified the standard Faster RCNN pipeline to consider arbitrarily oriented insulator localization [31]. In [32], the attention mechanism was introduced in Faster RCNN for self-explosion insulator defects. In summary, the aforementioned approaches focused on the following problems: 1) A complex background and various types of insulators make insulator defect detection difficult; 2) some severely damaged insulators are too small to recognize; 3) detection speed and accuracy needs to be balanced.

In most fields of object detection, "perfect annotation" indicates that the labeled bounding boxes are close to the targets' true boundaries and there are no missing labels. Therefore, the acquisition of the perfect annotation is time-consuming and labor-intensive. To decrease the dependency on perfect annotation, numerous research works have studied the different scenarios of imperfect annotation, such as incomplete annotation [33], [34], unreliable labels [35], and incremental new categories [36]. However, the incomplete annotation problem of insulator defect detection is not thoroughly investigated. Besides, there is a sample imbalance among different categories in our task.

In this paper, we propose a novel framework for incomplete annotation and sample imbalance in insulator defect detection. The proposed framework is based on the Faster RCNN detector and integrates Positive Unlabeled (PU) learning and focal loss. It is termed Pi-Index that follows the name of the proposed algorithm for estimating the class prior, a key parameter in PU loss, to improve PU learning. The algorithm is designed to generate a continuous value (Pi-Index) for each anchor as its probability of being positive. In the aspect of network architecture, Region Proposal Network (RPN) is improved by introducing PU learning to overcome the problem of incomplete annotation, whereas the focal loss strategy is applied to Region Of Interest (ROI) Head, to alleviate the impairment caused by sample imbalance.

The contributions of this paper are summarized in the following aspects: 1) A novel estimation strategy for the class prior, termed Pi-Index, is proposed to improve vanilla PU learning; 2) The PU learning strategy and focal loss are separately incorporated with RPN and ROI Head, which are responsible for the above two problems; and 3) Experiment results show that the proposed framework achieves better performance compared with baseline methods when missing labels or sample imbalance scenarios occur.

The rest of this paper is organized as follows. Section 2 describes the proposed framework that contains FPN backbone, RPN with PU learning and ROI Head with focal loss. Section 3 is about experimental results that prove the effectiveness of the proposed framework. Section 4 gives a summary of this paper.

Hosted file

image1.emf available at https://authorea.com/users/655376/articles/661212-electricalinsulator-defect-detection-with-incomplete-annotations-and-imbalanced-samples

2 Related work

In this section, we introduce the related work that is most relevant to our study. Firstly, a series of references about insulator detection are reviewed in Section 2.1. Then, Section 2.2 contains scientific literature on insulator segmentation, which is viewed as pixel-level detection of the insulators.

2.1 Insulator detection

Object detection is to predict a bounding box as an indication of the target's category and location. Similarly, insulator detection or insulator defect detection aims to locate the insulators by surrounding them with bounding boxes and identifying their categories or defect categories. Early methods about insulator defect detection adopted a combination of computer vision and machine learning technologies [1], [13], [14]. These methods heavily relied on hand-crafted features, which were time-consuming to design and required the assistance of experienced experts.

In recent years, deep learning-based detectors are introduced in the application of insulator detection [17]–[20]. The studies can be classified into one-stage and two-stage detectors. One-stage detectors typically correlate to the You Only Look Once (YOLO) family of deep neural networks [21]–[24], whereas two-stage detectors include Region-based CNN (RCNN) and its variations [25]–[27].

Various one-stage detectors are used to identify the insulator's defective regions. Yang et al. incorporated a lightweight backbone into the vanilla architecture of YOLO v3 to identify missing-cap insulators [19]. The lightweight backbone is based on MobileNet [28] with spatial pyramid pooling [37]. Similarly, a lightweight YOLO v4 is also proposed in [20] to balance detection accuracy and detection speed for insulator detection. Their lightweight techniques are analogous, with MobileNet replacing the original backbone. Furthermore, Han et al. presented TinyYOLO v4 that merged the self-attention module into the Feature Pyramid Network (FPN) [38] to enhance channel-level feature fusion [10]. This channel-wise self-attention facilitates learning better feature representation. With the release of YOLO v5, its pipeline was introduced into insulator detection detection research. In [39], four versions of YOLO v5 were explored for the localization of the insulator defect. As a result, the more suitable network architecture was chosen through contrast experiments. Gao et al. modified the YOLO v5 pipeline by incorporating a triplet attention module in order to enhance the detection performance of small insulator defects [9]. Then, another attempt to incorporate attention mechanisms with the YOLO v5 was reported in [29]. Lan et al. introduced the Convolutional Block Attention Module (CBAM) to provide more channel and spatial context information for insulator defect detection.

The methods listed above rely on one-stage detectors. Furthermore, two-stage object detection frameworks were introduced into the insulator detection community. In [17], [30], Faster RCNN was used to first roughly localize the regions where insulators are most likely to exist, referred to as "proposals" in the framework. Then, these proposals are fed into the second stage network, a multitask head, to refine the localization of the insulators' defects. Moreover, Tao et al. model insulator defect detection as a two-level task that includes insulator localization as well as defect detection [18]. The framework is made up of two concatenated Faster RCNNs: one with a VGG16 backbone for localization and another with the original Faster RCNN for detecting defective regions. Zhong et al. modified the standard Faster RCNN pipeline to consider arbitrarily oriented insulator localization [31]. The proposed framework introduced an oriented Region Proposal Network (RPN) to implement arbitrarily oriented localization for insulators. In [32], the attention mechanism was introduced in Faster RCNN for self-explosion insulator defects. In detail, an adaptive receptive field network is proposed and inserted into the FPN backbone.

2.2 Insulator segmentation

In other research works, the focus of the studies was to segment the insulators or defective regions from the background. In [40], a framework with two cascaded networks were proposed by Li et al. to detect the insulators globally and segment the local defect objects. The segmentation model was designed to incorporate an attention mechanism in an improved version of U-Net [41]. Efficient Channel Attention Networks (ECA-Net) was also introduced as the U-Net encoder, providing an example of fusing an attention mechanism for insulator segmentation [42]. Yu et al. focused on introducing fine-grained texture into the SINet architecture and simultaneously improved a positioning network to segment defective regions for insulators [2]. The insulator segmentation problem was solved by Antwi-Bekoe et al. using a common instance segmentation framework [43], in which the detection and mask branches implemented instance-level segmentation. Xuan et al. used a squeeze-excitation module to improve the backbone and a spatial attention module to forecast the insulator mask to produce excellent results in insulator defect segmentation [44].

3 Method

Our proposed framework intends to locate insulator areas, determine if these insulators are faulty, and identify which categories these insulators belong to. It needs to be emphasized that the framework explores the feasibility of this task under the conditions that a portion of annotations are inaccessible and sample sizes of various classes are imbalanced. In detail, our framework is based on the Faster RCNN pipeline [27], illustrated in Figure 1. We improved the Faster RCNN by combining PU learning with Region Proposal Network (RPN) and incorporating focal loss into the Region Of Interest (ROI) Head.

Hosted file

image3.emf available at https://authorea.com/users/655376/articles/661212-electricalinsulator-defect-detection-with-incomplete-annotations-and-imbalanced-samples

According to the network architecture depicted in Figure 1, our modified Faster RCNN consists of three components: Module A (the Feature Pyramid Network (FPN) backbone), Module B (the PU-RPN), and Module C (the ROI Head). Firstly, FPN serves as a feature extractor in charge of computing the feature maps, which are the input of subsequent Modules B and C. Secondly, RPN, the first stage of the detector, focuses on a binary classification, i.e., insulator regions, and background. The generated insulator regions are denoted as proposals in the Faster RCNN framework, and they will be refined into good insulators or different types of defective insulators in ROI Head. To solve the problem of incomplete annotation, we introduce the PU learning strategy [45] into the vanilla RPN, denoted as PU-RPN. Finally, Module C (ROI Head), the second stage of the detector, utilizes the proposals to further refine the predicted insulator's category and bounding box's localization. We applied focal loss to the ROI Head in order to mitigate the effect of sample imbalance. The details of the above components are described in the following subsections.

3.1 FPN as feature extractor

There are large or small targets in scenes to be recognized for object detection. Likewise, insulator defect detection also possesses the insulator strings in large size and the small insulators. The larger ones tend to be detected in high-level feature maps, which have low resolution and rich semantic information. But the smaller damaged insulators correspond to too few pixels to be distinguished in high-level feature maps. Therefore, a multiple-scale strategy is key to insulator defect detection.

Hosted file

image6.emf available at https://authorea.com/users/655376/articles/661212-electricalinsulator-defect-detection-with-incomplete-annotations-and-imbalanced-samples

The Feature Pyramid Network (FPN) is a famous architecture that applies the multiple-scale strategy to a base feature extractor . FPN follows the idea of the images' pyramid, and extends it to the pyramid of feature maps. The goal of FPN is to combine the advantages of both high-level and low-level feature maps. As shown in Figure 2, FPN consists of two inverse pathways, a bottom-up and atop-down pathway. The bottom-up pathway is the base feature extractor mentioned above (on the left in Figure 2), and usually employs a convolutional neural network (CNN) classifier. Along the direction of the dataflow in the bottom-up pathway, the base feature extractor is separated into five stages, and a downsampling operation is applied to each block. The top layers export the feature maps with more semantic information, while the output of the low layers possesses a higher spatial resolution. Following the architecture in [27], the base feature extractor adopts a Residual Neural Network (ResNet) [45]. Concretely, ResNet-50 is chosen to balance performance and computational complexity. The architecture of the adopted ResNet-50 is displayed on the left side of Figure 2. The learnable convolutional layers are organized into 5 stages, with **Stage1** as a convolutional layer (out-channel=64 and stride=2) and **Stage2~Stage5** as several stacked convolutional blocks. Each convolutional block in Stage2[~]Stage5 has three convolutional layers which match to the three lines11Each line contains settings for the kernel size, input channel, and output channel. The first line consists of two input channels. The former is a parameter of the initial convolutional layer, whereas the latter is a parameter of the two subsequent convolutional layers. in "Stage" boxes in Figure 2.

As depicted in Figure 2, FPN also provides a *top-down* pathway that contains top-down and lateral connections. The top-down connections are responsible for upsampling the higher-level feature maps to the same size as their lower-level counterparts. Specifically, an upsampling operation is based on nearest-neighbor interpolation. Meanwhile, lateral connections use a convolutional layer to increase the channel dimension of the bottom feature maps according to the top ones. The upsampled and channel-increased feature maps are then merged and fed into a convolutional layer to generate pyramid feature maps (**P5, P4, P3**, and **P2**).

). From top to bottom, the top-down and lateral connections cooperate to handle the original feature maps from ResNet-50 stage-by-stage.

3.2 RPN with PU learning for incomplete annotations

In insulator detection, incomplete annotation will lead to some unlabeled insulators treated as background during the training process, which causes the ambiguity between targets and background [34]. Therefore, we introduce PU learning as the new loss of Region Proposal Network (RPN).

A. Region Proposal Network for insulator proposals

RPN is a typical anchor-based detector, which implies targets are detected from anchor regions. The anchors are obtained by partitioning input images. RPN is in charge of determining whether an insulator exists and locating the target's offsets in each anchor. The centers of the anchors correspond to the centers of the receptive field and, more specifically, to the pixels in the top feature map **P5**. The anchor boxes at the center of each anchor have variable height-to-width ratios to accommodate targets of various shapes. According to [27], the Faster RCNN pipeline has nine anchor boxes with varying height-to-width ratios.

Our PU-RPN inherits the architecture and supervision method of the vanilla RPN. As seen in Figure 3 (or *Module B* in Figure 1), the feature maps from FPN are fed into PU-RPN, which generates proposals and crops the feature maps based on the proposals. The cropped feature maps serve as the ROI Head's inputs. PU-RPN is comprised of convolutional layer and two separated convolutional layers. The former convolutional layer learns from the pyramid feature maps (**P5**, **P4**, **P3**, and**P2**), which expand the input channel (256) to the output channel (512). In Figure 3, the upper classification branch uses a convolutional layer is *eighteen*, which implies *two* categories and *nine* anchors. Similarly, the other regressor branch aims to predict the coordinate offsets of the insulators (offsets for and). The output dimension of regressor branch is 36 (offsets).

Before the training processing, the ground-truth bounding boxes need to be converted to the supervision information of the anchors. A positive label is assigned to an anchor when Intersection over Union (IoU) is greater than 0.7 with any ground-truth box, whereas a negative label corresponds to IoU values below 0.3. The coordinate offsets are determined using the difference between the annotated bounding boxes and the positive anchors. The coordinate offsets of negative anchors are set in a random way.

The loss functions of the original RPN can be summarized in two parts: Positive-Negative (PN) classification and smooth L1 regression. The PN classification of insulators predicts the good insulators and defective insulators as positive samples, while the background is regarded as negative samples. The loss function for this PN classification is computed as follows:

where and separately represent the total number of a specific class and the predicted classification score of a particular anchor. The subscripts and stand for positive and negative class, respectively. The superscripts and are the indices of positive and negative anchors, respectively. is usually set to a cross-entropy loss that calculates the error between the anchors' prediction classification probability and the corresponding ground-truth labels.

When it comes to the localization loss for insulator defect detection, a typical choice is the *smooth L1-loss* function [46]. The predicted bounding-box is denoted as , while the ground-truth bounding box is represented as . Hence, the localization loss is defined as

In this equation, and is the same as Equation . The complete loss function for insulator defect detection is based on the combination of the PU classification loss and the localization loss .

The loss in Equation is used to train the original RPN in the Faster RCNN. Our proposed PU-RPN replaces the PN loss with the PU loss, and the details are given in the following sections.

B. PU learning for incomplete annotations

For insulator defect detection from images, manual annotations need to overcome the problems derived from the varied insulator appearances and the complicated background. In the scenario of incomplete annotations, the missing-labeled regions with insulators are treated as the background. If PU-RPN is trained with the loss defined in Equation , the PN loss will lead to semantic ambiguity. To solve this issue, PU learning is introduced in PU-RPN as an alternative to PN loss. Furthermore, PU learning can mitigate the effect that unlabeled insulators are treated as background.

In the framework of PU learning [47], the class prior π is usually introduced to represent the proportion of the actual positive samples in the dataset. The loss function of PU learning can be defined as:

where and therein stand for the number of labeled positive samples and unlabeled samples, respectively. and represent the indices of unlabeled anchors and the corresponding classification probability, respectively. The remaining symbols refer to Equation . The first term in Equation estimates approximately the loss from predicting true-positive samples as positive. The second term is the difference in loss between all anchors and true-positive anchors, which are both predicted to be negative. Then a non-negative operation is applied to the second term as suggested in [47], which leads to

The estimation of the class prior is crucial for the PU classification loss. The approach to determine the class prior is described in Section 2.4. Based on PU classification loss, Equation is rewritten as:

3.3 ROI Head with focal loss for sample imbalance

During identifying the categories of the insulators, different categories are with various quantities of samples. Therefore, the categories' contributions to the loss are not equal, which makes the category with fewer samples inclines to obtain a worse performance. Therefore, we introduce focal loss into the Region of Interest (ROI) Head to relieve this issue.

A. Region of Interest (ROI) Head for insulator detection

The ROI Head is the second-stage detector at the end of the Faster RCNN. The schematic diagram of ROI Head refers to Figure 4 (or *Module C* in Figure 1). It follows the FPN and RPN modules, which reserve the top k proposal regions as ROIs. The ROI Head refines the classification and regression results predicted by RPN. Its network architecture is composed of ROI pooling layers and several fully-connected (FC) layers. ROI pooling projects ROI spatial dimensions to fixed-size feature maps. Those FC layers imitate the VGG classifier head, which possesses two shared FC layers and two parallel separate FC layers as the classification and detection branches. The goal of the classification branch is to identify the good or defective insulators from those ROIs' feature maps.

During the training process, ROI pooling initially receives a lot of ROIs in different sizes. Each ROI's feature map can be partitioned roughly equal bins along the spatial dimensions. Then ROI pooling employs maxpooling to handle the values in the bins. As a result, each bin generates one maximum as its replacement, ensuring that all ROI's feature maps have the same size. Furthermore, the ROI-pooled feature maps are reshaped as a feature vector. The vector passes the shared FC layers for the enhancement of the semantic information. The last two separated FC layers finish the classification and detection tasks.

The loss functions for the ROI Head contain a multi-class cross entropy and smooth L1 localization loss defined in Equation . The multi-class cross entropy is defined as

where is the predicted vectors, and stands for the one-hot vector of the label. The and correspond to the indices of ROIs and elements in predicted vectors, respectively.

B. Focal loss for sample imbalances

The original Faster RCNN employs the multi-class cross entropy that penalizes the samples equally for all classes. This leads to the drawback that the classes with more samples are weighted by a larger for all classes. This leads to the drawback that the classes with more samples are weighted by a larger factor.

Focal loss is a better alternative for cross-entropy when the problem of sample imbalance exists [48]. In our framework, focal loss is incorporated into ROI Head to alleviate the effect of the sample imbalance. The focal loss for multi-class can be defined as follows:

Hosted file

image46.emf available at https://authorea.com/users/655376/articles/661212-electricalinsulator-defect-detection-with-incomplete-annotations-and-imbalanced-samples

Hosted file

image48.emf available at https://authorea.com/users/655376/articles/661212-electricalinsulator-defect-detection-with-incomplete-annotations-and-imbalanced-samples

where and represent the weighing factor and focusing parameter, respectively. The other notations refer to Equation . In practice, and are separately set to 0.25 and 2, according to the parameter setting in [48]. The final loss function for ROI Head is obtained by adding the above loss and :

The losses of ROI Head are accumulated and back-propagated to train the proposed framework.

3.4 Pi-Index: the strategy of estimating class prior

In the Faster RCNN framework, the PU loss is usually applied to the anchor-based RPN [34], [49]. In other words, classification and regression are performed separately for each anchor in RPN. Incomplete annotations should thus be converted from annotated boxes to anchors. There are lots of positive anchors assigned negative labels as background because a part of the targets lacks their annotations. For the PU classification loss of RPN, the class prior is defined as the percentage of both correctly and incorrectly labeled positive anchors.

A. Background: the existing methods

Zhao et al. considered the class prior as a hyper-parameter and determined it by grid search based on a validation set [49]. The estimation granularity of the class prior relies on the interval of the grid search.

Table 1: Performance (AP) of the method in [34] with different confidence thresholds (0.1-0.5) and annotation percents (0.3, 0.5, 0.7 and 1).

APC	0.1	0.2	0.3	0.4	0.5
1	87.60%	87.93%	87.58%	86.98%	87.56%
0.7	76.74%	76.66%	77.01%	76.67%	76.71%
0.5	70.17%	70.45%	70.48%	70.64%	69.95%
0.3	59.16%	$\mathbf{59.38\%}$	58.70%	59.23%	59.13%

The more accurate class prior is needed to be estimated, the smaller interval needs to be set. In other words, class prior with high precision leads to high computation complexity of the grid search method. For example, If the class prior is searched with an interval of 0.1, the computation complexity will increase tenfold from

the original.

In [34], the computation of class prior is transformed from the grid search of the class prior to determining a confidence threshold, which is used to predict whether the anchors are positive. Suppose an anchors' set, where and stand for the -th anchor box and its probability of being positive. is the number of anchors and is obtained by RPN. The class prior is computed with Equation .

where denotes the confident threshold. In conclusion, a reasonable threshold directly determines the number of positive anchors, which affects the estimation of the class prior. is also viewed as a hyper-parameter and needs to be estimated by the grid search.

Table 1 provided experiment results for the grid search of the confident threshold. Each column stands for AP metrics (details in Section 4.3) with the confident thresholds, while the different rows correspond to various annotation percents. When the Annotation PerCent (APC, refer to Section 4.2.1) varies from 1 to 0.3, the number of annotated labels decreases during the training process. From Table 1, it is concluded that the best confidence thresholds are inconsistent with different APCs. In [34], the confident threshold is fixed, and therefore it should be set to 0.2. The parameter selection is based on the fact that the confident threshold makes the model achieve more best performance. To sum up, this fixed threshold strategy (denoted as Pi-FT) also needs compute-intensive optimization of hyper-parameter.

B. A novel index for class prior

In this section, we offer a novel estimation technique for the class prior in our PU-RPN. As shown in Figure 5, the predicted results from two stages, the RPN and ROI Head, are fused to compute the class prior .

Suppose an anchor from a set of . The probability of predicted to be positive is symbolized by . Therefore, is a set of anchors with their probability of positive class. A predicted box is denoted as , which is output by ROI Head. Then the predicted boxes are collected as . The and indicate the number of anchors and predicted boxes, respectively. For an arbitrary anchor , we first match it with the predicted boxes and then determine the matched box using Equation .

where is a function of computing IoU between two boxes. We propose a class prior index for each anchor, i.e., the index for the anchor is defined as

The indices of anchors can be expressed as .The class prior is calculated by

Inspired by [34], an Exponential Moving Average (EMA) strategy to stabilize the class prior . The momentum is set to 0.9. The EMA class prior denotes the class prior after updates. The initialization of () is specified as the class prior of the first batch. Assuming the current batch's class prior is , and the EMA class prior is updated base on the Equation .

Algorithm 1: Class prior estimation based on Pi-Index for one batch.

Algorithm 1: estimation process

Input: Anchor Set: ; Image Batch; Total number of anchors: ; The number of classes ; FPN network ; RPN network; ROIH Output: Class prior;

- 3 Compute the probabilities predicted by RPN
- 4 # Filter proposals with a fixed threshold (0.05 in *detectron2*) for the probabilities.
- 5 Select proposals
- 6 # Compute the probabilities and bounding boxes predicted by ROIHead.

¹ for Image do

² Compute FPN features

Algorithm 1: estimation process
7;
8~# Filter the predicted boxes based on a configured confidence threshold .
9
10 Determine the index of positive anchors with and according to Equation (12) .
11 Collect the set of as
12 Compute the class prior of one image
13
14 end
Return

4 Experiment results

In this section, we conduct a series of comparative experiments to evaluate our proposed framework and the baseline method. First, we describe the details of the dataset in Section 3.1 and the experimental setup in Section 3.2. Then, the evaluation metrics are presented in Section 3.3. Finally, we show and analyze the experimental results in Section 3.4.



4.1 Dataset

4.1.1 Dataset description

The Insulator Defect Image Dataset (IDID)11The Insulator Defect Image Dataset refers to: https://ieeedataport.org/competitions/insulator-defect-detection, a widely-used insulator dataset, is utilized for model assessment. As indicated in Figure 6, the IDID comprises several high-resolution aerial images with insulators and the corresponding annotations. Each annotation consists of a category label and a bounding box. Insulators are classified as "Good," "Broken," and "FlashDamaged." Additionally, there exists a fourth category termed as "Insulator String", which refers to a cluster of insulators depicted in the images. Figure 6 depicts the many categories of bounding boxes with a variety of colors. The IDID training set comprises a total of 1600 aerial images, encompassing 2636 "Good", 1140 "Broken", and 2004 "FlashDamaged" insulator shells. These insulator shells collectively form 1788 insulator strings.

However, there exists a portion of images with incomplete annotations in IDID. Figure 7 displays several aerial images in which some insulators are incorrectly annotated. Specifically, the yellow boxes in the second row of images represent unlabeled insulators. The incomplete annotations could potentially stem from the dense arrangement of insulators and the oversight on part of the annotators. In many studies [], the IDID dataset has been utilized as a perfectly annotated dataset despite the presence of missing annotations. Therefore, regarding IDID as a partially annotated dataset is more reasonable and this partially annotated scenario studied in this paper has practical significance.

4.1.2 Dataset split

In order to eliminate interference from the sample imbalance, the model evaluation is performed on validation and test sets with category balance. The number of broken insulator shells is the lowest among the four classes. The samples from this category are distributed into the training, validation, and test sets in a ratio of 5:2:3. The validation and test sets for each category contain 228 and 342 samples, respectively. Therefore, we randomly select 228 and 342 samples from each category to form the validation and test sets, respectively. The remaining samples from each category are mixed together to form the training set, which consists of 1218 insulator strings, 1756 good insulator shells, 570 broken insulator shells, and 1434 flashover damaged insulator shells. In all our experiments, we have set the random seed to 1.

4.2 Experimental setup

4.2.1 Implementation details

To verify our method with different proportions of annotations, we randomly remove a portion of the annotations from the training set. The Annotation PerCent (APC) represents the percentage of annotations that remain after the aforementioned removal procedure. The other significant hyper-parameters are delineated as follows: The batch size is set to 16, the learning rate is set to 0.02, the total number of iterations is 10,000, and the evaluation interval for the validation set is 200 iterations. The best model is selected according to the AP metric on the validation set, which is then applied to the test set. Finally, the data augmentation in our framework contains the horizontal and vertical flips as well as the default data augmentation strategy in Detectron222The github repository of Detectron2: https://github.com/facebookresearch/detectron2.

4.2.2 Software platform

This experiment was conducted on a server with the Linux system and used Visual Studio Code (VSCode) as our Integrated Development Environment (IDE). PyTorch and Python were selected as the deep learning toolkit and the programming language, respectively. The hardware of the server is mainly composed of two Intel(R) Xeon(R) E5-2680 v4 CPUs with 14 cores each running at 2.4 GHz, 256 GB memory, and two Nvidia GeForce RTX 3090 GPUs.

4.2.3 Adopted baselines

To verify the effectiveness of our proposed method on incomplete annotation data, we conducted experiments to compare our method with other mainstream methods under different APCs (1, 0.7, 0.5, and 0.3). Our proposed framework is a Positive-Unlabeled (PU) framework, which is viewed as a combination of a Positive-Negative (PN) pipeline and PU loss. Therefore, we first selected existing mainstream Positive-Negative (PN) learning object detection algorithms [31], [32] to ablate the influence of PU loss. Furthermore, we also introduced several PU-based detectors [34], [49] as contrast methods. PN-based object detection algorithms typically include two frameworks: one-stage and two-stage. The existing one-stage frameworks for insulator detection are usually based on YOLO v3 and YOLO v4 with MobileNet backbone, abbreviated as M-YOLO v3 [19] and M-YOLO v4 [20], respectively. Since our study does not focus on lightweight computing, we combined DarkNet53 with the aforementioned YOLO frameworks (D-YOLO v3 [23] and D-YOLO v4 [24]), as well as YOLO v5 (D-YOLO v5) [39]. Because

 Table 2: Detection results of our method and other methods based on the complete annotation supported by IDID.

Methods	\mathbf{AP}	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
YOLO v3 (MobileNet)	42.92%	46.18%	72.94%	45.60%	50.90%	25.70%	49.40%
YOLO v3 (DarkNet53)	62.20%	76.30%	94.80%	59.80%	65.60%	55.70%	67.70%
YOLO v4 (MobileNet)	62.40%	77.10%	92.90%	64.60%	66.40%	54.60%	63.80%
YOLO v4 (DarkNet53)	66.10%	82.50%	95.40%	64.30%	69.90%	60.50%	69.80%
YOLO v5 (DarkNet53)	84.50%	96.20%	$\mathbf{98.30\%}$	84.60%	87.30%	79.10%	86.80%
Faster RCNN	87.19%	95.12%	97.23%	91.77%	87.08%	80.13%	89.79%
Pi-GS[38]	87.70%	95.33%	97.28%	92.07%	87.51%	80.60~%	90.62%
Pi-FT[24]	87.93%	95.59%	97.54%	92.37%	88.08%	80.72%	90.56%
Ours	88.11%	$\mathbf{96.35\%}$	97.98%	$\boldsymbol{93.14\%}$	88.06%	81.03%	90.20%

the proposed method is based on the Faster RCNN, this section also selected Faster RCNN as the baseline of PN-based two-stage detectors.

For the fair comparison, we chosen two PU-based detectors that adopted Faster RCNN as the base model. These two methods are detailed in Section 3.4. The first one is Pi-GS (Grid Search) [49], which estimates the class prior probability by conducting a grid search on the validation set with a search interval of 0.1 and repeating training 10 times. The second method is Pi-FT (Fixed Threshold) [34], which utilizes a fixed threshold to filter out the positive anchors by comparing their confidence scores with this threshold. In summary, we use M-YOLO v3, D-YOLO v3, M-YOLO v4, D-YOLO v4, D-YOLOv5, Pi-GS, and Pi-FT as the comparison methods.

4.3 Evaluation metrics

In this paper, we introduce the COCO evaluation metrics, a popular metrics for object detection, from the COCO challenge [50]. The COCO metrics is designed based on the principal metric, mAP@T, which

stands for the mAP with IoU threshold equaling T. For example, AP@0.5 and AP@0.75 are the typical mAP metrics, provided by the COCO metrics. AP in COCO metrics represents the average of mAP with IoU threshold varying from 0.5 to 0.95 (interval 0.05). Finally, AP-category is the AP being applied to one particular category, such as AP-String, APGood, AP-Broken, and AP-Flashover-Damage (shorted as AP-FlashoverD) in our experiments.

4.4 Detection results

This section presents the detection results of different methods under 1.0, 0.7, 0.5, and 0.3 Annotation PerCent (APC), as shown in Tables 2-5. Meanwhile, Figures 8-9 visualize the detection results of different methods, providing a more intuitive display of the results. In detail, Figure 8 (I) and (II) depict the detection results under 1.0 and 0.7 APCs, respectively. Figures 9 (I) and (II) individually present the detection results under 0.5 and 0.3 APCs.

4.4.1 Detection results with IDID's annotations

The IDID dataset is used as a fully annotated dataset in reality. But there are some samples with missing annotations due to the dense arrangement of insulators and oversights by annotators. Some examples of missing annotations are shown in Section 4.1.1. Therefore, even though all the IDID's annotations are used, it still belongs to incomplete annotation setting. We constructed the first experiment under all annotations provided by the IDID dataset. The detection results are summarized in Table 2, and the visualization of prediction results are shown in Figure 8 (I).



Rows 1-6 of Table 2 display the detection results of PN-based, primarily from one-stage (rows 1-5) and two-stage (row 6) networks. YOLO v5 significantly outperforms other one-stage networks with an Average Precision (AP) of 84.50%, an AP@0.75 of 96.20%, and an AP@0.5 of 98.30%. However, the two-stage Faster RCNN achieved an AP of 87.19%, which is approximately 2.69% higher than YOLO v5. This demonstrates that Faster RCNN exhibits the best performance among the above PN learning frameworks.

PU-based methods with Faster RCNN as the backbone are represented in rows 7-9 of Table 2. They have improved detection performance in comparison with PN-based detectors. Among these methods, our proposed Pi-Index obtained the highest performance with an AP metric of 88.25%, which to some extent

indicates the advantage of Pi-Index in estimating the class prior. Moreover, the AP values of Pi-GS [49] and Pi-FT [34] are 0.41% and 0.18% lower than Pi-Index, respectively. The reason may be that their class prior estimates are based on a hyper-parameter or a fixed threshold. This parameter or threshold is determined via grid search, but the search interval of 0.1 may stride over the optimal value and further lead to performance deterioration.

Figure 8 (I) shows the detection results of various methods with APC=1. The different columns correspond to different insulator detection scenarios, while each row is the detection result of a particular method. Rows (a)-(h) display manual labels and prediction results of M-YOLO v3, D-YOLO v3, M-YOLO v4, D-YOLO v4, YOLOv5, Pi-FT, and our Pi -Index. The green solid and red dashed boxes in the figure represent the Ground-Truth bounding Boxes (GT-Boxes) and predicted boxes, respectively. The text in the upper left corner of each GT-Box and predicted box individually indicates the annotated and predicted category, including: "String", "Good", "Broken", and "FlashDamged". To facilitate the analysis of prediction results, yellow arrows are employed to number those small and densely arranged insulators. The analysis of detection results is divided into two aspects: 1) The count of correctly detected insulators, i.e., whether any insulator strings or insulators were missed or detected repeatedly; 2) The localization accuracy of predicted boxes for the insulator strings or insulators.

In the first column of row (a), there is one insulator string and 13 insulators. In detail, the insulators at positions (1)-(4) and (9)-(12) are labeled as "Good", while those at positions (5), (7), and (13) are labeled as "Broken". However, the insulators at positions (6) and (8) are not labeled. All methods successfully identified the insulator string. Furthermore, the comparison of (b)-(e) and (f)-(h) reveals that M/D-YOLO v3 and v4 exhibit larger errors than YOLO v5 and two-stage algorithms in locating the string.

For the small-scale insulators, all methods correctly identify and accurately locate those at positions (1)-(3) and (7). The detection errors of each method are mainly concentrated on positions (5) and (13). Primarily, the "Broken" insulator at position (5) is misidentified as "Good" by M-YOLO v3, D-YOLO v3, and YOLO v5, misclassed as "FlashDamaged" by D-YOLO v4, and repeatedly identified as both "Good" and "Broken" by Pi-FT. Only M-YOLO v4 and the method proposed in this paper manage to correctly identify it. Next, the "Broken" insulator at position (13) was missed by M-YOLO v3, M-YOLO v4, and YOLO v5. It was misidentified as "Good" by D-YOLO v3 and "FlashDamaged" by D-YOLO v4. Only Pi-FT and our proposed method correctly identified it. By comprehensively considering the detection results of position (5) and (13), it reveals that M-YOLO v4, Pi-FT, and Pi-Index demonstrate superior performance.

Compared with Pi-Index, M-YOLO v4 not only misclassed the "Good" at position (9) as "Broken" but also missed the insulators at positions (10)-(12). In the detection results of Pi-FT, false alarms occurred at positions (4) and (5). They individually belong to the categories of "Broken" and "Good", but are simultaneously identified as both "Good" and "Broken". This further substantiates the superiority of the Pi-Index over the other two algorithms. There are another two unlabeled targets at positions (6) and (8). Neither M-YOLOv4 nor YOLO v5 recognized them, while M-YOLO v3 and D-YOLOv4 only recognized positions (6) and (8), respectively. By contrast, D-YOLO v3, Pi-FT, and ours recognized the insulators at positions (6) and (8). In summary, our proposed method not only achieves the overall best performance for the labels provided by IDID but also yields competitive results on unlabeled data.

The second column in Figure 8 (I) contains one insulator string that is composed of 12 insulators. However, seven insulators therein were annotated in IDID as either "Good" or "Broken". The former category includes six insulators at positions (1)-(4), (8), and (9), while the latter category includes only one insulator at position (5). It is worth noting that the insulator at position (9) was repeatedly annotated as "Good".

Table 3: Detection results of our method and the baseline method with (APC=0.7) in the training process.

Methods	AP	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashE
YOLO v3 (MobileNet)	35.50%	35.20%	64.30%	39.50%	49.10%	18.40%	35.10%
YOLO v3 (DarkNet53)	58.30%	68.90%	90.70%	54.00%	66.20%	50.90%	62.20%

Methods	AP	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
YOLO v4 (MobileNet)	59.70%	72.70%	90.20%	62.60%	63.50%	50.80%	61.80%
YOLO v4 (DarkNet53)	64.10%	79.10%	92.70%	60.70%	68.60%	58.10%	69.10%
YOLO v5 (DarkNet53)	72.20%	87.00%	91.00%	71.70%	79.30%	63.00%	75.00%
Faster RCNN	76.04~%	88.08~%	92.66~%	84.24~%	77.04~%	66.15%	76.72~%
Pi-GS	77.01%	87.25%	91.43%	82.61%	78.14%	67.13%	80.14%
Pi-FT	76.66%	87.41%	90.92%	81.15%	76.84%	67.89%	80.74%
Ours	79.57%	$\mathbf{89.45\%}$	$\mathbf{92.64\%}$	83.85%	$\mathbf{82.13\%}$	$\boldsymbol{68.96\%}$	$\mathbf{83.36\%}$

The detection results of (b)-(h) show that all methods have identified the insulator string. However, there was a false alarm in M-YOLO v3, and three more insulator strings were detected in detail. Meanwhile, the predicted boxes from D-YOLO v3, M-YOLO v4, and D-YOLO v4 exhibit a relatively larger error on the right border. On the contrary, YOLO v5, Pi-FT, and Pi-index provide the more accurate localization for the insulator string.

In the detection of insulators, YOLO v5, Pi-FT and our proposed Pi-Index correctly identified seven insulators (annotated in IDID) and the other methods had missed insulators to varying extent. This demonstrates that the Pi-Index achieved competitive results on the annotated insulators in IDID. Furthermore, the IDID dataset lacks annotations at positions (6), (7), (10), (11), and (12). Based on the aforementioned analysis, we further compared the performance of YOLO v5, Faster RCNN, and Pi-Index on unannotated insulators. recognized the unannotated insulators at positions (7) and (10). Therefore, both annotated and unannotated experimental results proved the effectiveness of Pi-Index.

4.4.2 Detection results under 0.7 annotation percentage

A decrease in data annotations inevitably results in a decline in the performance of the model. This circumstance facilitates a more comprehensive exploring the performance of different algorithms under imperfect labeling conditions. In this section, we further increase the proportion of unlabeled targets in the dataset. In detail, we randomly delete 30% of the labels in the IDID dataset. The detection results of different methods are organized in Table 3, and the visualization of annotated and predicted boxes is shown in Figure 8 (II).

From Tables 2 and 3 (1.0 vs. 0.7 APCs), it can be seen that each method's performance in Table 3 has declined compared with the counterpart in Table 2. Meanwhile, the performance differences between the various methods become more significant while maintaining consistency in trends. As detailed in Table 3, YOLO v5 surpassed other one-stage detectors (rows 1-4) with an AP value of 72.2%. Nevertheless, the AP, AP@0.75, and AP@0.5 of Faster RCNN are 3.84%, 1.08%, and 1.66% higher than those of YOLO v5, respectively.

Compared with Faster RCNN, the AP value of Pi-GS, Pi-FT, and Pi-Index have shown improvements of 2.56%, 2.91%, and 3.53%, respectively. This indicates that the addition of PU loss can effectively alleviate the effect of missing labels. From the last three rows in Table 3, Pi-Index achieved the best performance with an AP value of 79.57%, an AP@0.75 of 89.45%, and an AP@0.5 of 92.64%, which also outperformed other mainstream algorithms in each category's AP. As depicted in Table 2, when compared to Pi-GS and Pi-FT, Pi-Index exhibits a marginal improvement in the AP metric by less than 0.50%. But in Table 3, the AP of Pi-Index increased by 2.56% and 2.91% compared to those of Pi-GS and Pi-FT, respectively. This proves that the Pi-Index can achieve more significant performance improvements as the amount of unlabeled data increases.

Figure 8 (II) displays the detection results of seven methods, which are identical to those depicted in Figure 8 (I), under an APC of 0.7. The two columns in Figure 8 (II) correspond to two images with insulators. The first image or column contains a total of two insulator strings and 12 insulators. The insulators within the left-hand string are distributed evenly and appear relatively large in the image. Whereas the insulators in the

right insulator string are densely arranged, and some insulators therein are occluded due to shooting reasons. Consequently, distinguishing the insulators located on the right side poses a greater challenge compared to those on the left side. According to the IDID dataset, the insulators at positions (2), (3), (5), (6), and (7)-(12) have been annotated as "Good", while the insulator at position (4) has been labeled as "Broken".

From (b)-(h) in the first column of Figure 8 (II), all methods have successfully identified the insulator string on the left, whereas they failed to identify the insulator string on the right. This discrepancy is presumably due to a greater degree of occlusion affecting the insulator string on the right. Both M/D-YOLO v3 and M/D-YOLO v4 exhibited larger localization errors for the left insulator string compared to other methods. Specifically, the lower bounds of the predicted boxes for M/D-YOLO v3 and D-YOLO v4 exceeded the ground-truth lower bound by a large amount. The localization error of M-YOLO v4 was reflected in the larger predicted box, which enclosed the GT-Box. Furthermore, the Pi-Index possessed the highest overlap between its predicted box and the GT-Box than YOLO v5 and Pi-FT. These results indicate that the Pi-Index is more accurate for localizing insulator strings.

For small target insulators, M-YOLO v3 and YOLO v5 exhibited subpar performance with an accuracy rate below 50%. D-YOLO v3, M-YOLO v4, and Pi-FT successfully detected seven insulators. The first six insulators detected by all of them are positions (3)-(5), (7), (9), and (10). The seventh insulator detected by D-YOLO v3, M-YOLO v4, and Pi-FT is position (1), (2), and (12), respectively.

Furthermore, D-YOLO v4 and Pi-Index have correctly identified 8 insulators, indicating that these two methods outperformed the previously mentioned methods. For the localization of the left insulator strings, the predicted box of D-YOLO v4 was shifted to the right side of the GT-Box. Additionally, the lower boundary of the predicted box extended beyond the GT-Box. Contrastingly, Pi-Index generated a predicted bounding box that exhibited a large overlap with the GT-Box. Furthermore, we conducted a comparative analysis between D-YOLO v4 and Pi-Index based on their respective performance in detecting insulators within insulator strings. In the case of the left insulator string, both D-YOLO v4 and Pi-Index correctly identified the insulators at positions (1)-(4). However, D-YOLO v4 exhibited a large offset when locating insulators at positions (1) and (2). Within the right insulator string, both D-YOLO v4 and Pi-Index detected four insulators. Despite these detected insulators being located at different positions, there was a significantly greater overlap between GT-Boxes and the predicted bounding boxes produced by Pi-Index than those produced by D-YOLO v4. In conclusion, the proposed Pi-Index exhibits superior performance to the comparison algorithm in terms of both the number of correct detections and localization accuracy.

The second column of both Figure 8 (I) and (II) shared the same image or detection scenario. This scenario consisted of one insulator string and 12 insulators. It is worth noting that the same scenario was applied with different data APCs during the training process, which led to different detection results. The following analysis is based on APC = 0.7.

In Figure 8 (II), with the exception of M-YOLO v3, all other methods failed to successfully identify the insulator string. This may be attributed to the fact that during the process of reducing the APC, labels for insulator strings within similar scenes were randomly eliminated, resulting in their being interpreted by the network as background. However, M-YOLO v3 misclassified the insulators at positions (2)-(7) as an insulator string. Combining the detection results of M-YOLO v3 under APC=1 (in the second column of Figure 8 (I)), it can be seen that M-YOLO v3 identified up to four insulator strings. These identified insulator strings only encompassed a portion of the insulators present within the annotated insulator string. Therefore, we may conclude that M-YOLO v3 has a tendency to identify a series of consecutive insulators as forming an insulator string, rather than detecting the entire insulator string. Contrastingly, the remaining methods failed to detect the insulator string.

Table 4: Detection results of our method and the baseline method with (APC=0.5) in the training process.

Methods	AP	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
YOLO v3 (MobileNet)	31.90%	29.80%	59.80%	35.70%	47.90%	14.30%	29.60%

Methods	AP	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
YOLO v3 (DarkNet53)	52.30%	57.00%	86.60%	49.60%	59.70%	45.30%	54.60%
YOLO v4 (MobileNet)	54.50%	62.40%	87.20%	54.20%	62.40%	48.50%	52.90%
YOLO v4 (DarkNet53)	59.80%	70.80%	88.50%	54.10%	68.40%	53.80%	62.90%
YOLO v5 (DarkNet53)	62.20%	76.90%	85.50%	59.00%	71.40%	56.40%	61.90%
Faster RCNN	69.35~%	80.88~%	88.85~%	75.42~%	71.04~%	59.34~%	71.60~%
Pi-GS	69.90%	81.86%	88.56%	74.23%	73.26%	59.40%	72.69%
Pi-FT	70.45%	83.12%	91.06%	73.86%	74.34%	59.73%	$\mathbf{73.85\%}$
Ours	73.49%	84.43%	89.22%	79.74%	$\mathbf{78.33\%}$	$\boldsymbol{62.45\%}$	73.42%

For the insulators, M-YOLO v3, D-YOLO v3, and M-YOLO v4 each detected no more than two insulators, indicating their poor detection performance. D-YOLO v4 correctly detected the majority of the insulators, with the exception of those at positions (4) and (9). In contrast, YOLO v5, Pi-FT, and ours Pi-Index were able to accurately detect all of the insulators. These results show that Pi-Index is capable of achieving high detection performance under labeled conditions.

Furthermore, we compared the details in the detection results of YOLO v5, Pi-FT, and Pi-Index. Firstly, the detection results at positions (3)-(5) indicate that both Pi-FT and Pi-Index were significantly more accurate in locating insulators than YOLO v5. Subsequently, Pi-Index also identified the unannotated insulator at position (7). This further demonstrated the effectiveness of the proposed method when applied to unannotated data. Although D-YOLO v4 was able to detect two unannotated insulators, its failure to detect some annotated insulators resulted in an overall detection performance that was inferior to that of the Pi-Index.

4.4.2 Detection results under 0.5 annotation percentage

This section provides a summary of the detection results obtained using an APC value of 0.5. Table 4 presents quantitative results, including values for AP, AP@0.75, and AP@0.5. The visualized prediction results are illustrated in Figure 9 (I).

Rows 1-5 of Table 4 present results for one-stage detectors based on PN learning. YOLO v5 achieved an AP value of 62.2%, demonstrating a significant advantage over the other one-stage detectors. In contrast, the two-stage Faster RCNN method outperformed the aforementioned one-stage methods, achieving an AP value of 69.35%, an AP@0.75 value of 80.88%, and an AP@0.5 value of 88.85%.

The PU-based methods in rows 7-9 can be viewed as the combination of Faster RCNN and PU loss. These PU-based methods have achieved better performance than Faster RCNN, as shown in the sixth row. The AP metric of Pi-Index is 3.59% higher than that of Pi-GS and 3.04% higher than that of Pi-FT. A comparison of Tables 2-4 reveals that the performance of each method deteriorated to a certain extent as the labeling ratio decreased. However, the degree of deterioration for Pi-Index is relatively small compared to other mainstream methods. In other words, the performance improvement of Pi-Index is more significant than other mainstream algorithms.

Figure 9 (I) shows the detection results of the seven above methods with an APC of 0.5. As shown in (a) of the first column, the image contains one insulator string and a total of 11 insulators. The insulators at positions (1), (3)-(5), and (7)-(11) are labeled as "Good", whereas the insulator at position (6) is labeled as "FlashDamaged".



In the first column, from (b) to (h), all methods except for M/D-YOLO v4 and YOLO v5 correctly identified the insulator string. However, compared to Pi-FT and Pi-Index, M/D-YOLO v3 had a larger localization error for insulator strings. Specifically, the right boundary of the predicted box by M-YOLO v3 was inaccurate, while the upper boundary of the predicted box by D-YOLO v3 exceeded the GTBox.

For the relatively small insulators, all methods correctly classified the insulators at positions (1), (3), and (10). The positions that are easily undetected or misclassified by these methods are (6), (7), and (11). According to the IDID's annotations, M-YOLO v3 missed the insulators at positions (4), (6), and (7). D-YOLO v3 misclassified the "FlashDamaged" insulator at position (6) as "Good" and missed the insulators at positions (7) and (11). M-YOLO v4 missed the insulators at positions (6) and (7), and further misidentified the "Good" insulator at position (11) as "FlashDamaged". D-YOLO v4 missed 3 insulators at positions (5), (7), and (8). YOLO v5 missed the insulator at position (9) and had false alarms at positions (8) and (11), in which the "Good" insulators were identified as "Good" and "FlashDamaged". Both Pi-FT and Pi-Index detected the "Good" insulator at position (11) as both "Good" and "FlashDamaged".

In summary, Pi-FT and Pi-Index were the most effective algorithms for labeled data. It is worth noting that

the insulator at position (2) was not labeled in the IDID dataset, yet only Pi-Index was able to identify it. This demonstrates that Pi-Index is more effective for unlabeled data than other mainstream algorithms.

In the second column of Figure 9 (I), there is one insulator string containing 12 insulators. The insulators at positions (1) and (12) were only partially visible in the image, while the rest were entirely visible with clear edges. Specifically, positions (1)-(4), (6), and (8)-(12) were labeled as "Good", whereas positions (5) and (7) were labeled as "Broken".

The insulator string was identified only by D-YOLO v4, Pi-FT, and Pi-Index, whereas the rest of the methods missed it. Moreover, we focused on the detection details of the insulator string by D-YOLO v4, Pi-FT, and Pi-Index. D-YOLO v4 predicted a large error in locating the insulator string, and specifically its upper boundary exceeded that of the GT-Box. Pi-FT had false alarms on the prediction of the insulator string. The insulators at positions (1)-(5) and (3)-(12) were also misclassified as two separate insulator strings. However, Pi-Index has the best performance on the localization of the insulator strings. It correctly detected the insulators at positions (1)-(12) as one insulator string, and the boundary box basically overlaps with the GT-Box.

For the small-size insulators, all methods correctly detected the insulators at positions (2)-(4), (6), and (9)-(11). The differences of detection results from the above methods are mainly concentrated at positions (1), (5), (7), (8), and (12). Both M-YOLO v3 and D-YOLO v3 missed three insulators at positions (1), (5), and (12), but M-YOLO v3 further misidentified the "Good" insulator at position (7) as "Broken". M/D-YOLO v4 missed the insulators at positions (1) and (12). Pi-FT missed the insulators at positions (1) and (8), while our Pi-Index only missed the insulator at position (1). YOLO v5 successfully detected all insulators, performing better than other mainstream methods. In summary, Pi-Index is significantly better than other mainstream algorithms except for YOLO v5.

When comparing Pi-Index and YOLO v5, we can be concluded that Pi-Index is significantly better than YOLO v5 in terms of the large-scale insulator string. However, Pi-Index has a higher miss rate than YOLOv5 in terms of the small-size insulators. Besides, Pi-Index outperforms YOLOv5 in terms of locating small targets. This is specifically reflected in the overlap rate between the predicted box and the GT-Box for the insulators at positions (5) and (7), where Pi-Index has a higher rate than that of YOLOv5.

4.4.2 Detection results under 0.3 annotation percentage

In this section, the APC is set to 0.3, which means that 70% of the labels in the IDID dataset are randomly removed. The aim is to observe the detection performance of each method when labels are severely missing and to verify the effectiveness of the algorithm proposed in this paper under such conditions.

In Tables 2-5, each detector's performance deteriorated as the APC decreased. Contrary to the results of the previous experiment, Table 5 shows that the AP metric of YOLO v5 is lower than that of D-YOLO v4. It indicates that YOLO v5 is not suitable for the situation where there is a severe lack of labels. Nevertheless, the two-stage Faster RCNN surpasses other PN-based methods with an AP of 58.57%, an AP@0.75 of 70.41%, and an AP@0.5 of 81.02%, which is consistent with previous experiments. According

Methods	AP	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
YOLO v3 (MobileNet)	25.30%	23.10%	49.39%	26.90%	45.10%	7.50%	21.70%
YOLO v3 (DarkNet53)	47.20%	52.30%	78.00%	48.10%	59.00%	36.90%	44.80%
YOLO v4 (MobileNet)	46.20%	51.90%	77.80%	47.40%	58.20%	34.40%	44.90%
YOLO v4 (DarkNet53)	53.40%	63.20%	80.40%	50.80%	65.50%	40.40%	56.90%
YOLO v5 (DarkNet53)	50.00%	58.60%	74.90%	41.40%	66.00%	42.50%	50.00%
Faster RCNN	58.57~%	70.41~%	81.02~%	64.52~%	62.98~%	41.92~%	62.30~%
Pi-GS	60.74%	72.15%	81.72%	65.24%	67.82%	48.37%	61.53%
Pi-FT	59.38%	70.00%	80.13%	66.25%	68.22%	44.25%	58.81%

Table 5: Detection results of our method and the baseline method with (APC=0.3) in the training process.

Methods	\mathbf{AP}	AP@0.75	AP@0.5	AP-String	AP-Good	AP-Broken	AP-FlashD
Ours	63.82%	75.37%	83.16%	72.2%	71.71%	47.55%	$\boldsymbol{63.81\%}$

to Table 5, Pi-GS, Pi-FT, and Pi-Index are individually 2.17%, 0.81%, and 5.25% higher than Faster RCNN's AP metric, indicating that PU learning can effectively improve the detection performance under severe label missing conditions.

Figure 9 (II) displays the detection results of the seven methods with an APC value of 0.3. The first scenario, shown in the first column of Figure 9 (II), contains an insulator string that consists of six insulators. The insulator at position (6) is partially exposed, while the rest are clearly visible in the image. The insulators, except for the one labeled as "Broken" in position (3), are labeled as "Good".

For the insulator string, M/D-YOLO v3 did not recognize the insulator string, as shown in (b) and (c). Meanwhile, the other methods were able to identify it successfully. However, D-YOLO v4 and YOLO v5 had large localization errors when detecting the insulator string. Specifically, the left and right boundaries of the predicted box by D-YOLO v4 both exceeded the boundaries of the GTBox. The predicted box by YOLO v5 was completely enclosed within the GTBox. For small-size insulators, M-YOLO v3 and M-YOLO v4 missed all insulators. D-YOLO v3 and D-YOLO v4 detected only two insulators, while YOLO

v5 and Pi-FT detected three insulators. Therefore, Pi-Index outperformed the other methods with four insulators detected correctly.

There is one insulator string and seven insulators in the second scenario, shown in the second column of Figure 9 (II). The insulators are arranged closely and the positions (1)-(5) labeled as "Good", while the insulator at position (7) labeled as "FlashDamaged". From the detection results of the insulator string, it can be seen that the results from all the methods are poor. Only Pi-FT and Pi-Index were able to successfully recognize the insulator string. For the insulators, M/D-YOLO v3 and M-YOLO v4 did not detect any insulators, while D-YOLO v4 only detected the insulator at position (3). This indicates that these methods have a high rate of missed detections. The performance of YOLO v5 and Pi-FT is slightly better. In detail, they only missed the insulator at position (7). Pi-Index further successfully identified the insulator at position (7). The detection results demonstrated that Pi-Index performed best for detecting insulators in contrast to the other detectors.

4 Conclusion

In this paper, we propose a framework to explore insulator defect detection under circumstances that combine incomplete annotation and sample imbalance. The framework introduces a PU-RPN that integrates improved PU learning with RPN and incorporates focal loss into the ROI Head. On one hand, the improved PU loss is used to address the problem of incomplete annotation by appropriately calibrating the losses of different samples. In addition, the proposed Pi-Index strategy is responsible for estimating a more accurate class prior by combining classification confidence scores from RPN and predicted boxes from ROIHead. On the other hand, focal loss is incorporated into the ROI Head to alleviate performance degradation caused by sample imbalance. To verify the effectiveness of our proposed framework, we conducted two groups of experiments. The experimental results demonstrate that our method outperforms not only the baseline method or Faster RCNN, but also other mainstream methods. Specifically, our method achieved the highest AP metrics (88.11% for 1.0 APC, 79.57% for 0.7 APC, 73.49% for 0.5 APC, and 63.82% for 0.3 APC) with different proportions of annotations when compared to mainstream methods.

Funding Statement: This research is partially supported by the R&D Program of Beijing Municipal Education Commission (KM202210009003), the National Natural Science Foundation of China (62001009), and the Scientific Research Foundation of North China University of Technology.

Author Contributions: Author Contributions: Conceptualization, F. Q. P.; Methodology, F. Q. P., C.Y. L. and J. S. Z.; Validation, J. S. Z., C. Y. L.; Writing—original draft preparation, F. Q. P., and C. Y. L; Writing—review and editing, J. S. Z., and C. Y. L. All authors have read and agreed to the published version of the manuscript. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials: The Insulator Defect Image Dataset refers to: https://ieeedataport.org/competitions/insulator-defect-detection

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. Z. Fortes, V. H. Ferreira, and R. Zanghi, "Fault diagnosis in transmission lines: trends and main research areas," *IEEE Lat. Am. Trans.*, vol. 13, no. 10, pp. 3324–3332, 2015.

[2] J. Yu, K. Liu, M. He, and L. Qin, "Insulator defect detection: A detection method of target search and cascade recognition," *Energy Rep.*, vol. 7, pp. 750–759, 2021.

[3] Y. Liu, D. Liu, X. Huang, and C. Li, "Insulator defect detection with deep learning: A survey," *IET Gener. Transm. Distrib.*, vol. 17, no. 16, pp. 3541–3558, Aug. 2023, doi: 10.1049/gtd2.12916.

[4] S. F. Stefenon, G. Singh, B. J. Souza, R. Z. Freire, and K. Yow, "Optimized hybrid YOLOu-Quasi-ProtoPNet for insulators classification," *IET Gener. Transm. Distrib.*, vol. 17, no. 15, pp. 3501–3511, Aug. 2023, doi: 10.1049/gtd2.12886.

[5] S. F. Stefenon *et al.*, "Classification of insulators using neural network based on computer vision," *IET Gener. Transm. Distrib.*, vol. 16, no. 6, pp. 1096–1107, Mar. 2022, doi: 10.1049/gtd2.12353.

[6] Y. Hao, W. Liang, L. Yang, J. He, and J. Wu, "Methods of image recognition of overhead power line insulators and ice types based on deep weakly-supervised and transfer learning," *IET Gener. Transm. Distrib.*, vol. 16, no. 11, pp. 2140–2153, Jun. 2022, doi: 10.1049/gtd2.12428.

[7] S. Peng, L. Ding, W. Li, W. Sun, and Q. Li, "Research on intelligent recognition method for self-blast state of glass insulator based on mixed data augmentation," *High Volt.*, p. hve2.12296, Dec. 2022, doi: 10.1049/hve2.12296.

[8] W. Godoi, K. De Geus, R. Da Silva, and others, "Automated flaw detection in X-ray tomography of polymer insulators," *Insight-Non-Destr. Test. Cond. Monit.*, vol. 52, no. 10, pp. 533–539, 2010.

[9] J. Gao, X. Chen, and D. Lin, "Insulator Defect Detection Based on improved YOLOv5," in 2021 5th Asian Conference on Artificial Intelligence Technology (ACAIT), IEEE, 2021, pp. 53–58.

[10] G. Han, M. He, F. Zhao, Z. Xu, M. Zhang, and L. Qin, "Insulator detection and damage identification based on improved lightweight YOLOv4 network," *Energy Rep.*, vol. 7, pp. 187–197, 2021.

[11] F. Deng, W. Luo, B. Wei, Y. Zuo, H. Zeng, and Y. He, "A novel insulator defect detection scheme based on Deep Convolutional Auto-Encoder for small negative samples," *High Volt.*, vol. 7, no. 5, pp. 925–935, Oct. 2022, doi: 10.1049/hve2.12210.

[12] C. Dong, K. Zhang, Z. Xie, and C. Shi, "An improved cascade RCNN detection method for key components and defects of transmission lines," *IET Gener. Transm. Distrib.*, vol. n/a, no. n/a, doi: https://doi.org/10.1049/gtd2.12948.

[13] L. Guo, Y. Liao, H. Yao, J. Chen, and M. Wang, "An electrical insulator defects detection method combined human receptive field model," *J. Control Sci. Eng.*, vol. 2018, 2018.

[14] D. Zuo, H. Hu, R. Qian, and Z. Liu, "An insulator defect detection algorithm based on computer vision," in 2017 IEEE International Conference on Information and Automation (ICIA), IEEE, 2017, pp. 361–365.

[15] J. Fang, Y. Wang, and W. Chen, "End-to-end power equipment detection and localization with RM transformer," *IET Gener. Transm. Distrib.*, vol. 16, no. 19, pp. 3941–3950, 2022, doi: https://doi.org/10.1049/gtd2.12578.

[16] A. Amini Badr, A. Safari, and S. Najafi Ravadanegh, "Segmentation of interconnected power systems considering microgrids and the uncertainty of renewable energy sources," *IET Gener. Transm. Distrib.*, vol. n/a, no. n/a, doi: https://doi.org/10.1049/gtd2.12934.

[17] G. Kang, S. Gao, L. Yu, and D. Zhang, "Deep architecture for high-speed railway insulator surface defect detection: Denoising autoencoder with multitask learning," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 8, pp. 2679–2690, 2018.

[18] X. Tao, D. Zhang, Z. Wang, X. Liu, H. Zhang, and D. Xu, "Detection of power line insulator defects using aerial images analyzed with convolutional neural networks," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 50, no. 4, pp. 1486–1498, 2018.

[19] L. Yang, J. Fan, S. Song, and Y. Liu, "A light defect detection algorithm of power insulators from aerial images for power inspection," *Neural Comput. Appl.*, pp. 1–11, 2022.

[20] Z. Xing and X. Chen, "Lightweight algorithm of insulator identification applicable to electric power engineering," *Energy Rep.*, vol. 8, pp. 353–362, 2022.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[22] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[23] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," ArXiv Prepr. ArXiv180402767 , 2018.

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *ArXiv Prepr. ArXiv200410934*, 2020.

[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, 2015.

[26] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[27] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, vol. 28, 2015.

[28] A. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *ArXiv Prepr. ArXiv170404861*, 2017.

[29] Y. Lan and W. Xu, "Insulator defect detection algorithm based on a lightweight network," in *Journal of Physics: Conference Series*, IOP Publishing, 2022, p. 012007.

[30] W. Zhao, M. Xu, X. Cheng, and Z. Zhao, "An Insulator in Transmission Lines Recognition and Fault Detection Model Based on Improved Faster RCNN," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–8, 2021, doi: 10.1109/TIM.2021.3112227. [31] J. Zhong, Z. Liu, C. Yang, H. Wang, S. Gao, and A. Núñez, "Adversarial reconstruction based on tighter oriented localization for catenary insulator defect detection in high-speed railways," *IEEE Trans. Intell. Transp. Syst.*, 2020.

[32] Y. Zhang, W. Cai, S. Fan, R. Song, and J. Jin, "Object Detection Based on YOLOv5 and GhostNet for Orchard Pests," *Information*, vol. 13, no. 11, 2022, doi: 10.3390/info13110548.

[33] M. Xu et al., "Missing Labels in Object Detection.," in CVPR Workshops, 2019, p. 5.

[34] Y. Yang, K. J. Liang, and L. Carin, "Object detection as a positive-unlabeled problem," ArXiv Prepr. ArXiv200204672, 2020.

[35] K. Fujii, D. Suehiro, K. Nishimura, and R. Bise, "Cell Detection from Imperfect Annotation by Pseudo Label Selection Using P-classification," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2021, pp. 425–434.

[36] F. Cermelli, A. Geraci, D. Fontanel, and B. Caputo, "Modeling Missing Annotations for Incremental Learning in Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 3700–3710.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, 2015.

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[39] Z. Feng, L. Guo, D. Huang, and R. Li, "Electrical insulator defects detection method based on yolov5," in 2021 IEEE 10th Data Driven Control and Learning Systems Conference (DDCLS), IEEE, 2021, pp. 979–984.

[40] X. Li, H. Su, and G. Liu, "Insulator defect recognition based on global detection and local segmentation," *IEEE Access*, vol. 8, pp. 59934–59946, 2020.

[41] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, Springer, 2015, pp. 234–241.

[42] G. Han *et al.*, "Improved U-Net based insulator image segmentation method based on attention mechanism," *Energy Rep.*, vol. 7, pp. 210–217, 2021.

[43] E. Antwi-Bekoe, G. Liu, J.-P. Ainam, G. Sun, and X. Xie, "A deep learning approach for insulator instance segmentation and defect detection," *Neural Comput. Appl.*, vol. 34, no. 9, pp. 7253–7269, 2022.

[44] Z. Xuan, J. Ding, and J. Mao, "Intelligent Identification Method of Insulator Defects Based on Center-Mask," *IEEE Access*, vol. 10, 2022.

[45] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[46] H. Li *et al.*, "A novel loss calibration strategy for object detection networks training on sparsely annotated pathological datasets," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2020, pp. 320–329.

[47] R. Kiryo, G. Niu, M. C. Du Plessis, and M. Sugiyama, "Positive-unlabeled learning with non-negative risk estimator," *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[48] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[49] Z. Zhao, F. Pang, Z. Liu, and C. Ye, "Positive-unlabeled learning for cell detection in histopathology images with incomplete annotations," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2021, pp. 509–518.

[50] "(2014). The COCO Object Detection Challenge. Available: https://cocodataset.org/detection-eval,"