

Article

A Novel Auto-Synthesis Dataset Approach for Fitting Recognition Using Prior Series Data

Jie Zhang, Xinyan Qin, Jin Lei *, Bo Jia, Bo Li, Zhaojun Li, Huidong Li, Yujie Zeng and Jie Song

College of Mechanical and Electrical Engineering, Shihezi University, Shihezi 832003, China; zhangjie77@stu.shzu.edu.cn (J.Z.); qinxy@shzu.edu.cn (X.Q.); 20192109044@stu.shzu.edu.cn (B.J.); 20202109065@stu.shzu.edu.cn (B.L.); 20202009021@stu.shzu.edu.cn (Z.L.); lihuidong@stu.shzu.edu.cn (H.L.); 20212009014@stu.shzu.edu.cn (Y.Z.); 20212009017@stu.shzu.edu.cn (J.S.)

* Correspondence: jinlei@shzu.edu.cn

Abstract: To address power transmission line (PTL) traversing complex environments leading to data collection being difficult and costly, we propose a novel auto-synthesis dataset approach for fitting recognition using prior series data. The approach mainly includes three steps: (1) formulates synthesis rules by the prior series data; (2) renders 2D images based on the synthesis rules utilizing advanced virtual 3D techniques; (3) generates the synthetic dataset with images and annotations obtained by processing images using the OpenCV. The trained model using the synthetic dataset was tested by the real dataset (including images and annotations) with a mean average precision (mAP) of 0.98, verifying the feasibility and effectiveness of the proposed approach. The recognition accuracy by the test is comparable with training by real samples and the cost is greatly reduced to generate synthetic datasets. The proposed approach improves the efficiency of establishing a dataset, providing a training data basis for deep learning (DL) of fitting recognition.



Citation: Zhang, J.; Qin, X.; Lei, J.; Jia, B.; Li, B.; Li, Z.; Li, H.; Zeng, Y.; Song, J. A Novel Auto-Synthesis Dataset Approach for Fitting Recognition Using Prior Series Data. *Sensors* **2022**, *22*, 4364. <https://doi.org/10.3390/s22124364>

Academic Editors: Simone Bianco, Marco Buzzelli and Jean Baptiste Thomas

Received: 15 May 2022

Accepted: 6 June 2022

Published: 9 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: synthesis dataset; prior series data; fitting recognition; YOLOX; Blender; inspection robot

1. Introduction

A power transmission line (PTL) system is the infrastructure of a country, necessary for industrial development and people's livelihood [1]. The key components of a PTL system consist of PTLs, fittings, towers, etc. Once one of these components of a PTL system fails, it would cause significant economic loss [2], therefore it is necessary to carry out planned inspections for early fault detection and maintenance [3]. At present, the inspection methods mainly include two kinds: human inspection and robot inspection [4].

With the development of inspection technology, people strive to develop advanced inspection robots to replace humans. There are mainly three types of advanced inspection robots: flying-mode robots (e.g., UAVs), walking-mode robots (e.g., dual-arm inspection robots), and hybrid-mode robots (e.g., flying-walking PTL inspection robots—FPTLIR) [5]. The flying-mode robots fly in the air and can cross obstacles with a flexible inspection view. The walking-mode robots walk along the PTLs with a stable inspection view. The hybrid-mode inspection robots can fly in the sky and walk along the PTLs, with the advantages of the previous two kinds of robots. However, the key technology of inspection robots to identify and position various fittings on the PTLs is the primary support for autonomous inspection, defect detection, and power maintenance [6,7]. Object detection methods based on deep learning (DL) can automatically extract features without processing complex images [8], showing a great potential in image recognition [9]. Moreover, the influences of light intensity, target scale, and environment on image recognition can be effectively reduced. DL has two research directions on image recognition: one-stage detection and two-stage detection. The representative methods of one-stage detection have YOLO series, SSD, etc. The representative methods of two-stage detection have R-CNN, Fast R-CNN, Faster R-CNN, etc.

Many research efforts to detect fittings using DL have been reported in recent decades [10,11]. The SSD-based mode could achieve automatic multi-level feature extraction to identify aerial images with complex backgrounds [12]. Exact region-based convolutional neural network (Exact R-CNN) and cascade mask extraction and exact region-based convolutional neural network (CME-CNN) were proposed in the literature [13]. These methods could solve the accuracies of existing detection methods limited by the complex background interference and small target detection. Rahman et al., [14] used YoloV4 to identify porcelain insulators on the towers and achieve good performance.

It is well known that training a DL model needs a large dataset. However, the volume of the dataset is often inadequate while training DL models for fitting recognition. The volume of a dataset is one of the key factors affecting the accuracy of a model [15]. These methods on the expanding dataset can be mainly divided into two categories. One is to expand using the real dataset. Song et al., [16] investigated the impacts of data augmentation methods on the accuracy of target recognition, in which the impacts of the data augmentation methods were significant such as Gaussian blur, scaling, and rotation. Chang et al., [17] focused on the training characteristics of the neural network using towers and aerial images as a background. Images of insulator segments were randomly stitched to simulate different lengths and types of insulators and RGB channels, lighting, and color were randomly varied to generate the dataset. Rahman et al., [14] combined various image processing techniques (e.g., data expansion, shimmer enhancement) to increase the volume of the training dataset and to improve the generalization capability of the insulator detection model. Another direction is to expand by building a synthesis dataset. Zidek et al., [18] generated a DL training dataset from virtual 3D models. The advantage of the approach was the swift preparation of a 2D sample training dataset from virtual 3D models to train models. The recognition accuracy was comparable with the trained model by real samples. Chen et al., [19] proposed a sample expansion method based on a combination of statute and 3D modeling technology. The high accuracy results were obtained when the ratio of real samples to simulated samples was 2.0. Madaan et al., [20] rendered synthetic wires using a ray tracing engine and overlaid them on 67 K images from flight videos to generate a synthesis dataset, with an average precision score of 0.73 on the USF dataset. Table 1 shows the key points of these methods.

Table 1. Summary of the literature for expanding dataset.

Characteristic	Core Idea	Method	Pros and Cons
Expanding a real dataset	Data augmentation [16]	Histogram equalization, gaussian blur, random translation, scaling, cutout, and rotation	Histogram equalization, random translation, and cutout have a low impact on network accuracy. Faster RCNN cannot deal with the issue of target rotation well.
	Random splicing [17]	—	The trained model used synthetic samples have a good generalization for the real images, the transparent samples made the networks difficult to converge.
	Image processing [14]	A deep Laplacian pyramid-based super-resolution network and a low-light image enhancement technique	The quality and volume of the training dataset are improved, but the images need to be collected.
Generating a synthesis dataset	Generating dataset using virtual 3D models [18,21,22]	Blender script	Dataset generation is easy and fast in a single background working condition for standard parts.
	Generating dataset based on images [19,23,24]	Mixing real and synthetic data, Generative Adversarial Networks (GANs)	The appropriate ratio is favorable for neural network learning; however, it is difficult to collect real images.
	Overlaying synthetic data on videos [20]	A ray tracing engine	Semantic segmentation datasets can be quickly generated, but the background changes are not flexible.

Data acquisition is hard and costly for a complex environment around PTLs and it relies on some specialized equipment (e.g., UAVs, inspection robots). In addition, the weather is changeable. It is challenging to include all working conditions in the data (e.g., the image of a lake on a sunny evening with a backlight). To address these difficulties, we propose a novel auto-synthesis dataset approach for fitting recognition using prior series data. The synthesis rules are formulated based on prior series data (including type and position of fitting, viewpoint position, topography, season, and irradiance). The 2D images are rendered by the synthesis rules using advanced virtual 3D techniques. Finally, a synthetic dataset can be generated with images and annotations obtained by processing images using OpenCV (a standard image processing technique). The various attributes in the synthesis dataset (e.g., type and pixel coordinates of fitting, background environment) would be adjusted and expanded. The coverage of the synthesis dataset is more comprehensive than the real dataset. The proposed approach allows to extract the prior series data according to the working conditions and synthesizes the dataset with the characteristic properties of the working conditions.

The main contributions of this paper are as follows:

1. Formulating the data synthesis rules based on prior series data. The prior series data are derived from complex working conditions; therefore, the synthesis rule possesses the characteristics of these working conditions.
2. Integrating advanced virtual 3D technology to render 2D images. Synthetic images would not be subject to topographic restrictions such as real images, reducing the difficulty of acquiring image.
3. Proposing a novel auto-synthesis dataset approach for fitting recognition using prior series data. The synthetic dataset with space–time properties generated by this approach is used to train the DL model, making the network easier to learn important features. The training results are biased toward identifying the fittings at the source of the prior information, improving the accuracy rate of target recognition.

This paper is organized as follows: Section 2 describes the prior series data and the approach for generating the synthetic dataset. Section 3 trains a DL model by the synthetic dataset and verifies the effectiveness and feasibility of this approach by tests. Section 4 provides the discussion. Conclusions are drawn in the last section.

2. Proposed Approach

Training a DL model to identify a fitting needs a large dataset. However, it is difficult to collect the fitting dataset on the PTLs, requiring manual upline or manipulating inspection robots. Some fittings for a PTL system are shown in Figure 1. Moreover, the cost of collecting the fitting dataset is more than a general dataset collection and manual annotation is time consuming and imprecise. To address these problems, we propose a novel auto-synthesis dataset approach based on the prior series data.

We take a section of a 35 kV PTL system in Xinjiang as our research object. The PTL system is distributed in Yanqi and Kuerle, passing through flatland, wetland, and gobi. The average ratio of cloudy–sunny–cloudy is 52%, 30%, 9% in these areas from 2011 to 2021, respectively. The percentage of severe weather (unsuitable for inspection) is small at 9%. Therefore, the impact of severe weather might be ignored. The details of the auto-synthesis dataset approach are shown in Figure 2. The dataset can be automatically generated based on the prior series data using Blender software and OpenCV library.

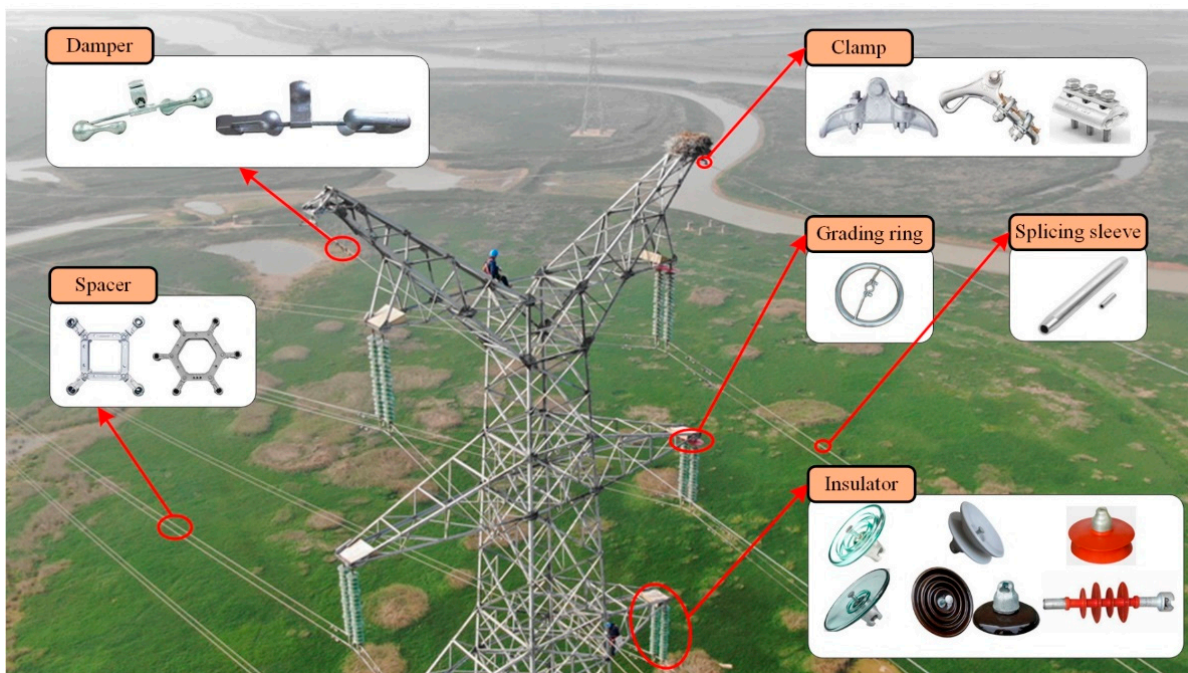


Figure 1. Illustration of fitting type.

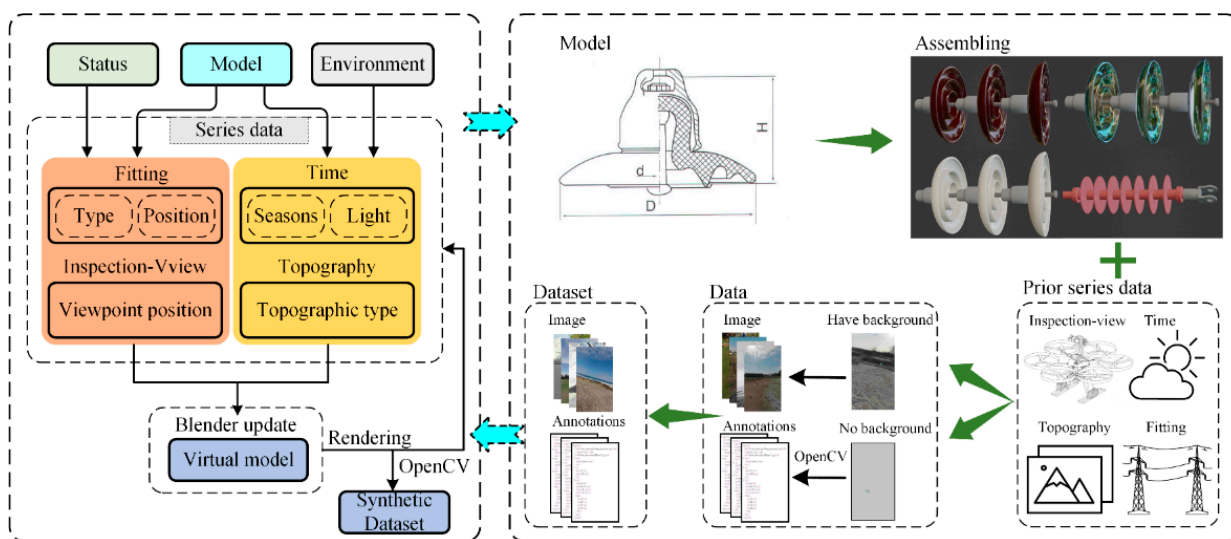


Figure 2. The details of the dataset auto-synthesis approach.

2.1. Prior Series Data

The synthesis rules are formulated from the prior series data, which reduce the generation of inspection-independent images in the dataset and assign space–time properties to the dataset. Therefore, the quality of the synthesized dataset can be enhanced and the amount of non-essential features are reduced.

When a robot inspects the PTLs, the motion trajectory of the robot is planned, and the collected images depend on this trajectory. Figure 3 illustrates the flying and walking trajectories of the FPTLIR; therefore, the prior series data would restrict the viewpoints of collected images on the trajectory. The images whose viewpoints are outside the trajectory can be covered to avoid the DL model learning the unrelated features.

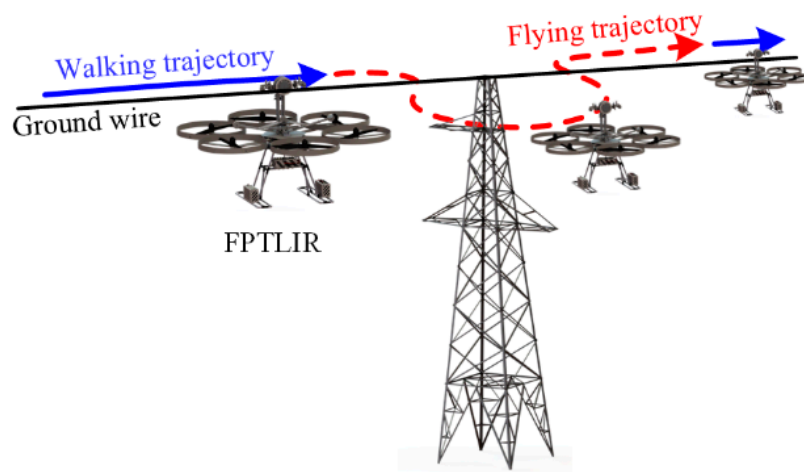


Figure 3. Illustration of the motion trajectory of the FPTLIR.

In general, if there is a PTL system with j segments (a segment refers to two adjacent towers and the PTLs between them), the set of a prior series data can be expressed as

$$S = \{ S^i | i = 1, 2, 3, \dots, j \}, \tag{1}$$

where S^i is the set of a prior series data for the i th segment, it can be obtained as

$$S^i = \{ P^m N | m = 1, 2, \dots, N \in N+ \}, \tag{2}$$

where P^m denotes kinds of prior series data, $P^m N$ represents the N th prior series data in P^m , and the set of prior series data of each segment can be described as

$$S^i = P^1 N_1 / P^2 N_1 / P^3 N_1 / P^4 N_4 (N_1, N_2, N_3, N_4 \in N). \tag{3}$$

The structure of the prior series data is shown in Figure 4. Figure 5 illustrates the meanings of prior series data. Each segment of the PTL system includes four prior series data, namely: prior fitting series (P^1), prior inspection-view series (P^2), prior topography series (P^3), and prior time series (P^4). Prior fitting series (P^1) includes: prior information, i.e., insulator type ($P^1 T$), insulator position ($P^1 P$), damper type ($P^1 TD$), damper position ($P^1 PD$), and other various fitting types and positions. Prior inspection-view series (P^2) includes: viewpoint position ($P^2 V$). Prior topography series (P^3) includes: the prior information, i.e., topographic type ($P^3 R$). Prior time series (P^4) includes: season ($P^4 S$) and irradiance ($P^4 L$).

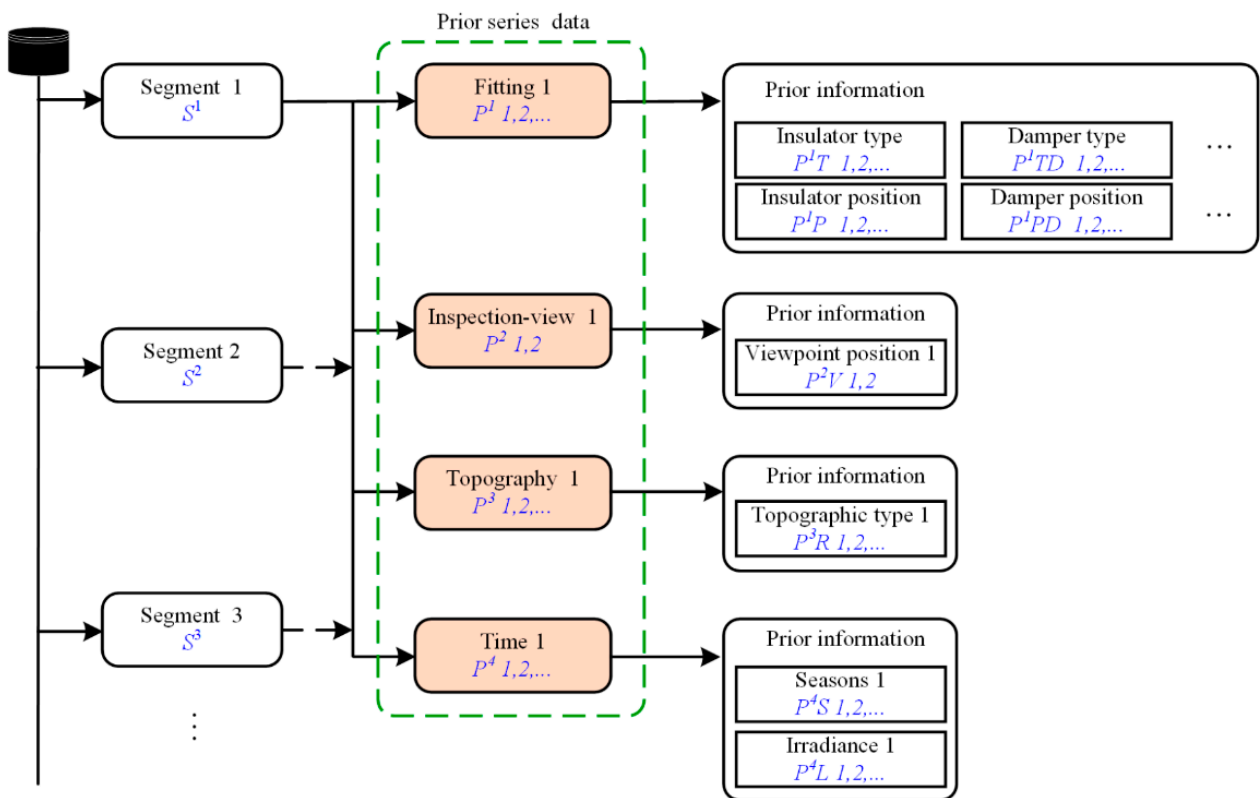


Figure 4. Illustration of the structure of the prior series data.

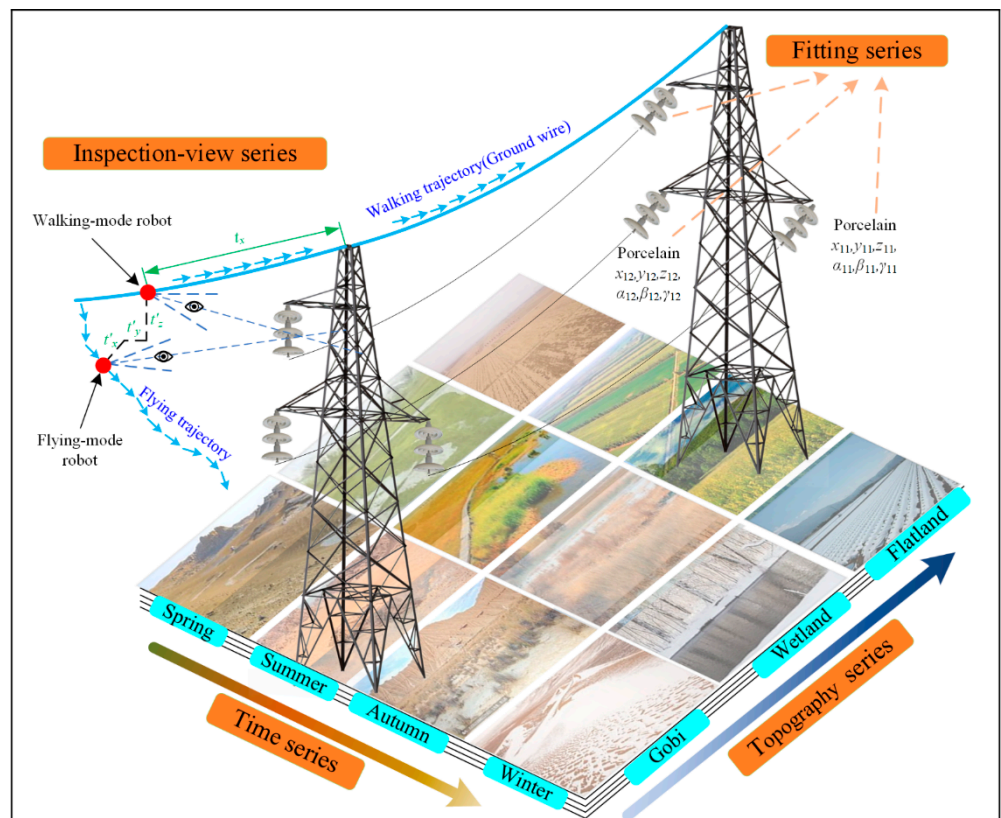


Figure 5. Illustration of the meanings of a prior series data in a segment.

The prior fitting series includes the prior information of many types of fittings with a high similarity; thus, the insulator is used as an example in these fittings.

Some sets of prior series data are shown in Table 2. The four prior series data are independent in that all series data have derived prior information. In addition, they are also interrelated in that multiple series data can influence the synthetic dataset property. For example, the position relationship between the insulator and viewpoint is influenced by the prior fitting series(P^1) and the prior inspection-view series(P^2). The background of images is determined by the prior topography series(P^3) and the prior time series(P^4). Insulator type, insulator-viewpoint position relationship, and background are the three key parameters for constituting the synthetic dataset.

Table 2. Example of the set of prior series data.

Tower Number	Date	Time	Set of Prior Series Data
1#	2 January	10:00	S^1-P^1 $1/P^2$ $1/P^3$ $1/P^4$ 1
		15:00	S^1-P^1 $1/P^2$ $1/P^3$ $1/P^4$ 2
	3 March	13:00	S^1-P^1 $1/P^2$ $1/P^3$ $1/P^4$ 3
		18:00	S^1-P^1 $1/P^2$ $1/P^3$ $1/P^4$ 4
2#	4 June	9:00	S^2-P^1 $2/P^2$ $1/P^3$ $2/P^4$ 5
		11:00	S^2-P^1 $2/P^2$ $1/P^3$ $2/P^4$ 6

2.1.1. Prior Fitting Series

The prior fitting series includes the insulator position P^1P and the insulator type P^1T . The insulator position includes the position coordinate and the direction of the insulator.

The voltage of the PTL system of our study is 35 kV. The PTL system is composed of several tension segments. Two tension towers are constituted into one tension segment and tangent towers are constructed between the tension segments. Furthermore, angle towers to substitute tension towers are used for PTL system steering with an angle of 0° to 90° . Angle towers and tension towers are equipped with suspension insulator strings and tension insulator strings, and the horizontal angle of the tension insulator string is β ($0^\circ < \beta < 90^\circ$). Each tangent tower also has suspension insulator strings with three or four pieces.

Insulators can be divided into disk insulators and bar composite insulators. The component of disk insulator includes a steel cap, an insulator, and a steel foot. The insulator of disk insulators is available in various materials: porcelain, tempered glass, and rubber. The bar composite insulators are composed of an umbrella sleeve, a core bar, and metal at both ends.

2.1.2. Prior Inspection-View Series

The prior inspection-view series includes the viewpoint position P^2V of the robot, which includes the position coordinates and the direction of the viewpoint.

The inspection-view is related to the inspection mode of the robot. The inspection robots include three modes: flying-mode, walking-mode, and hybrid-mode. The flying-mode robot performs the fixed-point air inspection paralleling the detection objects, and the walking-mode robot walks along the PTLs with a fixed inspection viewpoint. The hybrid-mode robot has the view of the walking-mode robot and the flying-mode robot. The view angle relationship is shown in the upper left of Figure 5.

2.1.3. Prior Topography Series

The prior topography series includes the topographic type P^3R .

The PTL passes through flatland, wetland, and gobi. The flatland is mainly planted with cotton, apricot, red date, wheat, corn, sugar beet, tomato, pepper, melon, and grape. The main plants of the wetlands are phytoplankton and aquatic vascular plants. The main topography of the gobi is saline and dunes.

2.1.4. Prior Time Series

The prior time series includes the irradiance P^4L and the four seasons P^4S . This series data are mainly used to constrain the season and the irradiance of the prior topography series. Figure 6a shows the variation of average irradiance from sunrise to sunset on the first day of every month over the decade. The strongest irradiance is 1025 W/m^2 . The peak is usually at 2:00 pm each day in January, March, and June. Figure 6b depicts the daily distribution of the average strongest irradiance over the decade. The stronger irradiance mainly occurred around June.

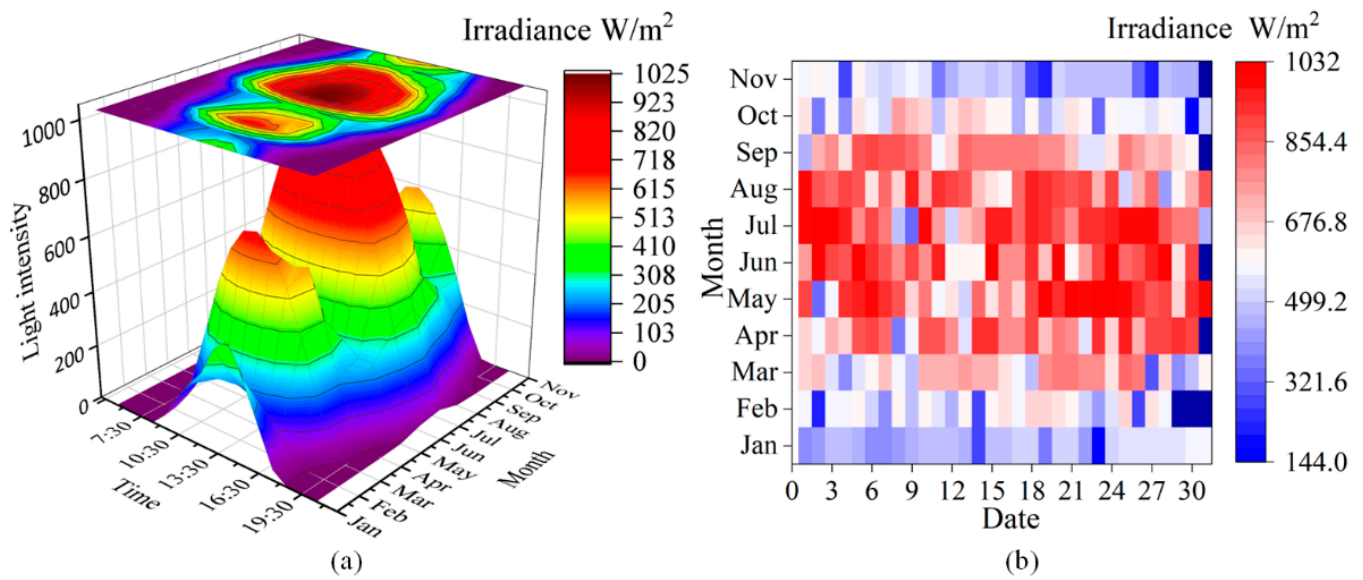


Figure 6. (a) Variation of day irradiance from light to no light on the first day of each month, (b) daily distribution of the strongest irradiance.

2.2. Auto-Synthesis Dataset

An auto-synthesis dataset is generated, including the following steps:

- The 3D model of the insulator pieces with the material information is imported into Blender and the insulator pieces are assembled into insulator strings;
- A camera is created so that the coordinate system of the camera viewpoint coincides with the world coordinate system. The position and the angle of the insulator strings are corrected by the prior inspection-view series and the prior fitting series, and the view angle of the camera is also adjusted;
- Background-free images are outputted, then the high dynamic range images (HDRIs) are imported and adjusted to render images based on the prior time series and the prior topography series;
- The background-free image is processed using standard image processing techniques (Python OpenCV library) to generate object bounding boxes and annotate files with basic parameters (position/dimensions);
- A synthetic dataset is generated with images and annotations. The synthetic dataset is divided into a training set and a validation set.

Figure 7 depicts the flowchart of the auto-synthesis dataset approach.

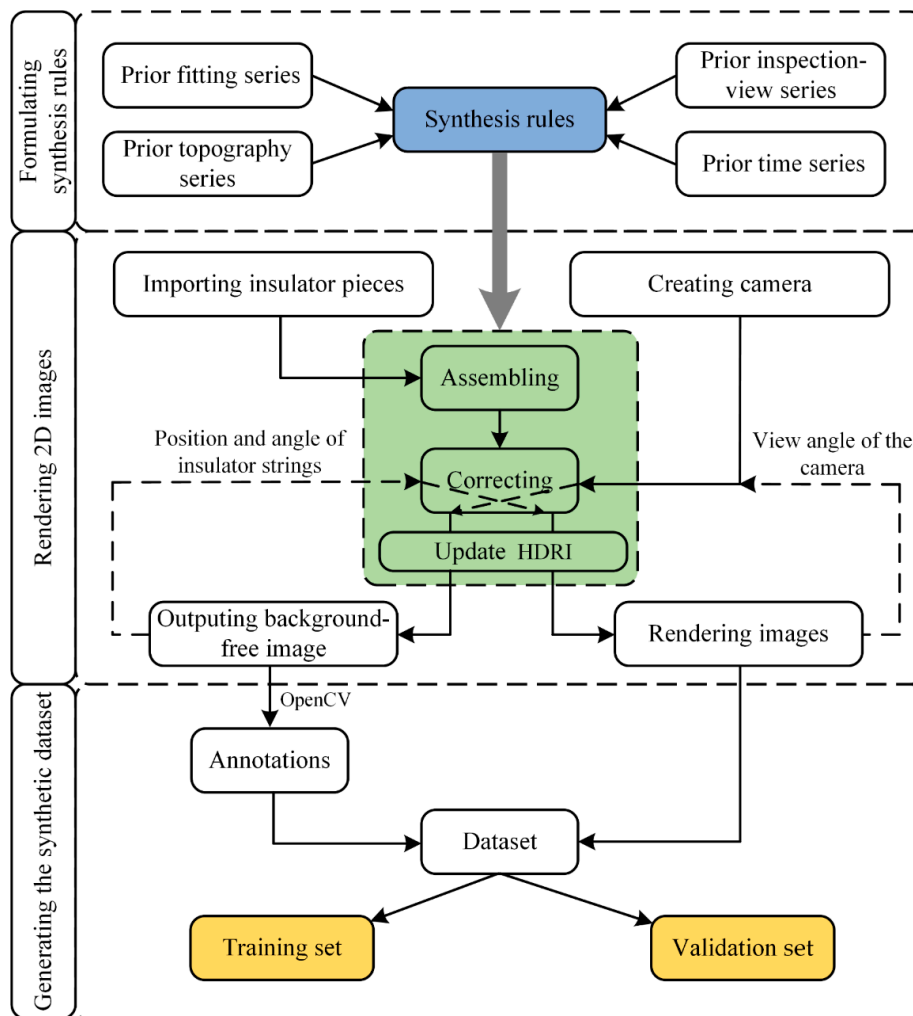


Figure 7. Flowchart of the auto-synthesis dataset approach.

The insulator piece and insulator strings are shown in Figure 8. According to the prior fitting series and national standard GB/T 7253-2005, the 3D model of the insulator piece is drawn and imported into Blender as the basic model. Insulators include porcelain insulators, tempered glass insulators, and composite insulators.

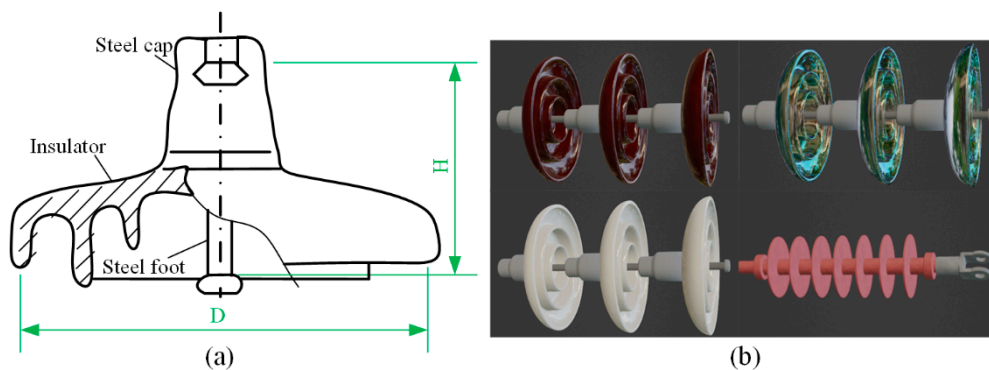


Figure 8. (a) Structure of insulator piece. (b) Four types of insulator strings in Blender.

The position relationship between the FPTLIR’s viewpoint and the insulator is shown in Figure 9. The viewpoint coordinate system ($o_r-x_r y_r z_r$) coincides with the world coordinate system. The insulator piece is assembled based on the Blender scripting language and the insulator string is translated and rotated relative to the viewpoint. The movement

of the insulator string is described by the rotation matrix given in Equation (4) and the translation matrix given in Equation (5). The camera position remains unchanged and only the view angle is changed.

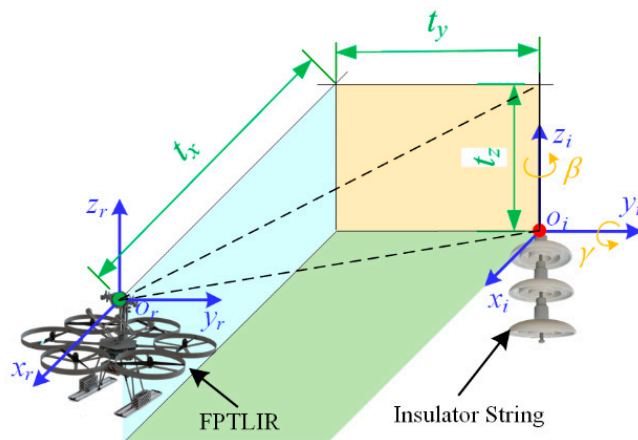


Figure 9. The position relationship between the FPTLIR’s viewpoint and the insulator.

$$R_y = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}; R_z = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{4}$$

where

R_y R_z represents the four-dimensional rotation matrix of insulators, β, γ is the rotation angle of each additional generated view.

$$T_{xyz} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where

T_{xyz} represents the four-dimensional translation matrix of the insulators, t_x, t_y, t_z is the translation distance of each additional generated view.

For the walking mode of the FPTLIR, β is related to the insulator angle, $0 < \beta < 35^\circ$; γ is related to the angle of the PTL system steering, $0 < \gamma < 90^\circ$. The values of t_y, t_z are shown in Table 3 and relate to the tower type. $t_x > 0.3$ m is related to the distance of the robot from the tower.

Table 3. The key parameters of the synthetic dataset.

Description	Symbol	Unit	Value
x_i axis conversion distance of insulator	t_x	m	[0.3, 30]
y_i axis conversion distance of insulator	t_y	m	{1.9, 2.2, 2.3}
z_i axis conversion distance of insulator	t_z	m	[0, 4.6]
Translation distance x_r axis increase in flight	t'_x	m	[-30, 30]
Translation distance y_r axis increase in flight	t'_y	m	[-10, 10]
Translation distance z_r axis increase in flight	t'_z	m	[-10, 10]
y_i axis rotation angle of insulator	β	°	[-90, 90]
z_i axis rotation angle of insulator	γ	°	[-60, 60]
Irradiance	E	W/m ²	[0, 1032]
HDRI environmental horizontal rotation angle	θ_h	°	[0, 360]
HDRI environmental vertical rotation angle	θ_v	°	[-45, 45]
x_i axis conversion distance of insulator	t_x	m	[0.3, 30]

For the flying mode of the FPTLIR, the translation matrix can be added to the translation matrix of the walking mode, whose translation matrix is described by Equation (6)

$$T_{F-xyz} = \begin{bmatrix} 1 & 0 & 0 & t_x + t'_x \\ 0 & 1 & 0 & t_y + t'_y \\ 0 & 0 & 1 & t_z + t'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (6)$$

where

T_{F-xyz} represents the four-dimensional translation matrix of the insulator in the flying mode. t'_x, t'_y, t'_z is the translation distance of each additional generated view relative to the walking mode.

The view angle of the camera is adjusted to ensure that the insulator is within the output image. The view angles for rotation along the z_r and x_r axes are described by Equation (7). The synthetic dataset parameters are listed in Table 3.

$$\begin{cases} \theta_z = \arccos \frac{t_z^2 + t_y \sqrt{t_x^2 + t_y^2}}{t_x^2 + t_y^2 + t_z^2} \\ \theta_x = \arctan \frac{\sqrt{t_x^2 + t_y^2}}{-t_z} (t_x \geq 0) \\ \theta_x = -\arctan \frac{\sqrt{t_x^2 + t_y^2}}{-t_z} (t_x < 0) \end{cases} \quad (7)$$

Blender is an open-source modeling and rendering software providing an internal python language to control object angles, object positions, object materials, camera views, and HDRIs. The background-free 2D images are generated based on the Eevee engine. The environmental background of four seasons and irradiance are added in Blender based on the HDRIs file. The images are rendered based on the Cycles engine and all images are generated at 720×1280 resolution.

The bounding box (including width, height, and center coordinate of the bounding box) of the insulator is detected using the OpenCV based on the background-free images. It is written to the XML files that the object classification, the position and the size of the object bounding box, and the image height/width. Figure 10 shows the image generated by Blender and the process of generating the XML from the 2D image. The synthetic pseudo codes are listed in Algorithm 1.

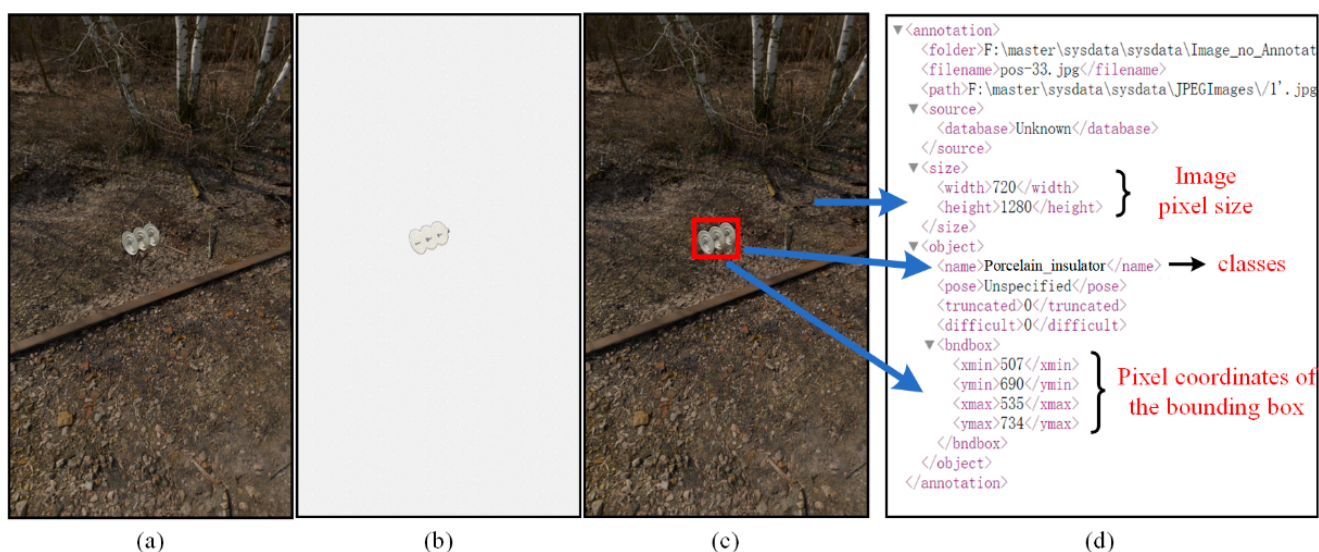


Figure 10. (a) Rendered image in Blender. (b) Generating backgrounded-free image in Blender. (c) Drawing bounding boxes on the rendered image, processing background-free image by OpenCV to get the bounding box. (d) Key image parameters in the XML file.

Algorithm 1. Auto-synthesis dataset algorithm

```

def main():
    # Parameter initialization
    _init_0
    # Read hdri files
    hdri_folder = bpy.path.abspath("//hdri")
    hdri_file = [os.path.join(hdri_folder, f) for f in os.listdir(hdri_folder) if f.endswith(".exr")]
    # Synthetic dataset
    for num_insulators in range(starnum_insulators, endnum_insulators):
        # Assemble insulators according to the number of insulator strings
        zoom(num_insulators)
        for hdris in hdri_file:
            for k in range(1, num_hdri):
                # Counting
                num_image += 1
                # Loading environment
                node_environment, node_background, link1 = hdri(hdris)
                # Adjustment of environmental parameters
                hdri_adjust()
                # Mobile insulator
                move()
                # Adjusting ambient light
                light()
            # Switching the CYCLES rendering engine
            bpy.context.scene.render.engine = 'CYCLES'
            bpy.context.scene.cycles.device = 'GPU'
            save(Fi0 + str(num_image) + ".png")
            # Output background-free image
            bpy.context.scene.render.engine = 'BLENDER_EEVEE'
            bpy.data.worlds["World"].node_tree.nodes["Background"].inputs[1].default_value = 1
            clear(node_environment, node_background, link1)
            save(Fi2 + str(num_image) + ".png")
            # Clear nodes and cache images, unload hdri
            bpy.context.scene.world.node_tree.nodes.clear()
            img_remove()
            hdri_reload()

        #Automatic generation of annotation files
        cv_label()

```

2.3. Dataset Evaluation

The selected HDRI scenes include flatland, wetland, gobi, and snow with an equalization ratio. The irradiance of the images is also equal.

The costs of a generated dataset mainly include the rendering cost and the annotation cost. Generating a synthesis dataset of 10,800 images takes 35 h, including 6.8 min of annotation. The costs of rendering images are counted in Figure 11. The synthetic images are rendered using the Cycles engine on the GTX 1660, rendering one image per 8.69 s on average. The background-free images are generated using the Eevee engine on the AMD R7 3700X CPUs, generating one background-free image per 3.07 s on average. There are 328,000 images of 91 classes of objects in the MS COCO dataset. However, the task of annotating this dataset is arduous. For example, it takes more than 20,000 h [25] to determine object classes presented in images of the MS COCO and the average is 16.4 s per annotation. However, the automatic annotation takes only 38 ms per piece, reducing time costs compared with the manually annotated dataset.

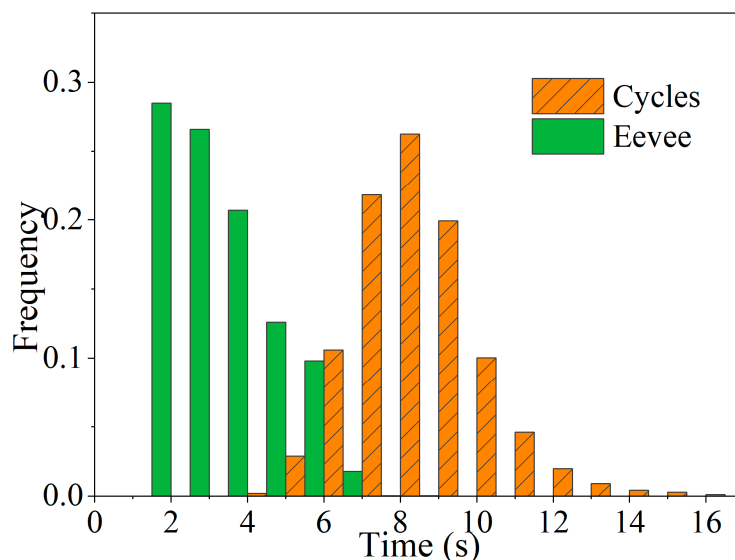


Figure 11. Costs of rendering images.

3. Experiments

3.1. Experiment Description

- (1) Dataset: A synthetic dataset of 10,800 images with 720×1280 pixels is divided into a training set and a validation set to train a DL model. A dataset of 1200 real images with 2160×3840 pixels is used to test the trained DL model. The ratio of the training set, validation set, and test set is 8:1:1. The format of these datasets is COCO2017. Note that the format of our generated synthetic dataset is VOC format and needs to be converted for the COCO format. The object classification is a porcelain insulator.
- (2) Experimental configuration: The experiments are conducted based on the DL framework YOLOX. The computer configuration is AMD R7 3700X CPU, NVIDIA GTX-1660 with 8 GB of video memory, and 16 GB RAM. The operating system is ubuntu 18.04.
- (3) Defect detection criteria (evaluation criteria): Three widely used indexes are used to quantitatively assess the performance of defect detection methods: precision (P), recall (R), and achieved mean average precision (mAP). P is the percentage of true samples among all the samples that the system determines to be “true”. R is the percentage of “true” samples found among all true samples. AP represents the detection accuracy of a single category, and mAP is the average of AP for each category.

$$P = \frac{T_P}{T_P + F_P}, \quad (8)$$

$$R = \frac{T_P}{T_P + F_N}, \quad (9)$$

$$AP = \int_0^1 P(R) dR, \quad (10)$$

where T_P and F_P are the number of correctly and incorrectly positioned defects, respectively. $T_P + F_P$ is the total number of located defects, and $T_P + F_N$ is the total number of actual defects.

3.2. Training Results with YOLOX

YOLOX is a released model for target detection, exceeding the YOLO series in 2021. YOLOX has a fast detection speed and meets inspection robots’ performance needs.

The YOLOX network architecture has three components: backbone, neck, and head. The backbone based on the CSPDarknet53 is used for feature extraction. The neck is used for extracting some more complex features. The head is mainly used to predict the classification and location of targets (bounding boxes). Figure 12 illustrates the YOLOX

network architecture. The structure is input–backbone–neck–head–output. The input is a 640×640 pixel image obtained by the warpAffine conversion of a 720×1280 pixel image. And the output is the object classification, classification probability, and localizations of bounding boxes.

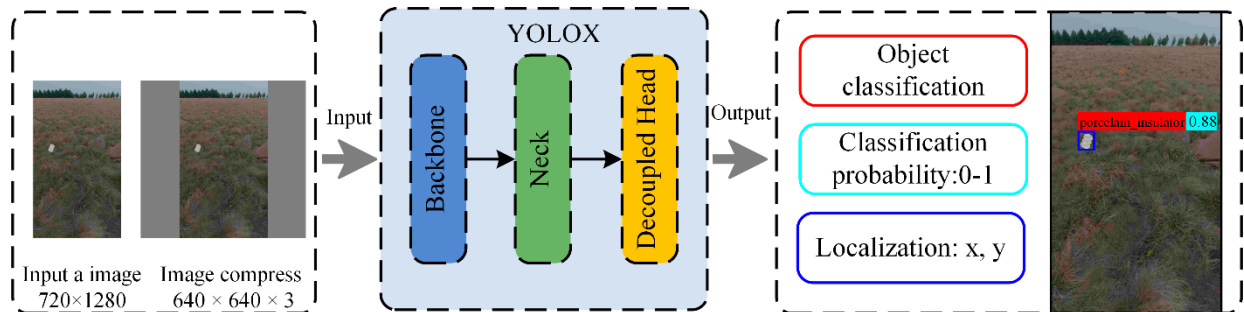


Figure 12. Illustration of the YOLOX network architecture.

The pre-trained model (YOLOX-s) is used to reduce the training time. Verification is performed every 1 epoch. All other parameters use the default official configuration. The model is trained with 300 epochs and the training time is 36 h 48 min. The mAP of the validation set is shown in Figure 13. The mAP_{val5095} on the validation set is 97.22%.

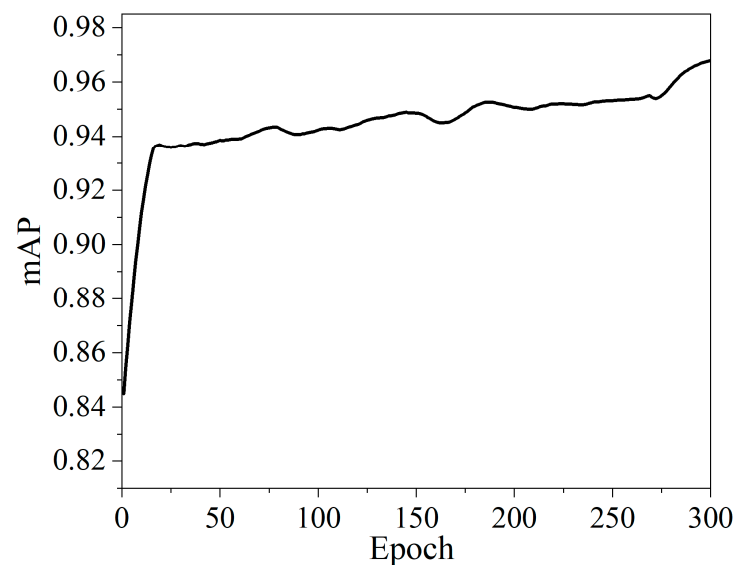


Figure 13. Schemes follow the same formatting.

3.3. Comprehensive Performance Analysis

The trained model is tested on the test set with 1200 images and the images of the test set are taken from 60 video streams of the camera shots. The position relationship of the camera insulator is consistent with the position relationship of the viewpoint insulator. Few images in the test set are stitched by adding insulators from cropped shooting images to the aerial images. The graphical image annotation tool for annotating the test set is LabelImg. Some examples of the images with annotations are shown in Figure 14.

Experiments show that R, P, and F1 are 95%, 96.18%, and 0.96, respectively, and the mAP_{test5095} reached 98.38% when the score threshold is 0.5. The evaluation process of the YOLOX-s detection model is shown in Figure 15. The results show that the trained model using a synthetic dataset can effectively recognize real samples.

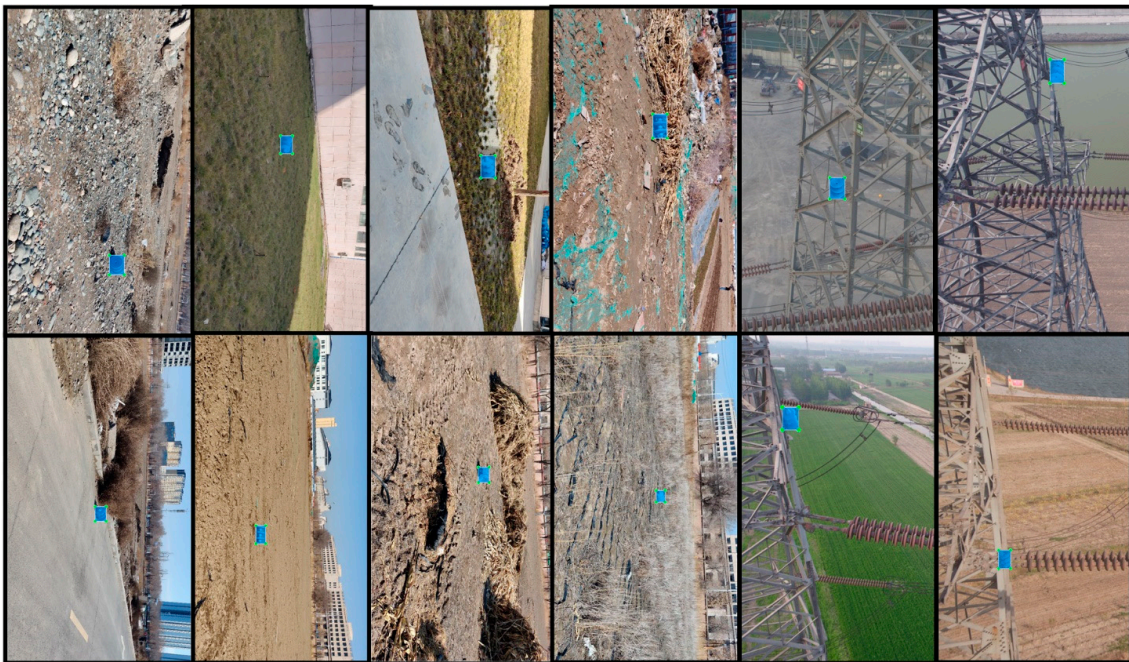


Figure 14. Example of the test set of images with annotations.

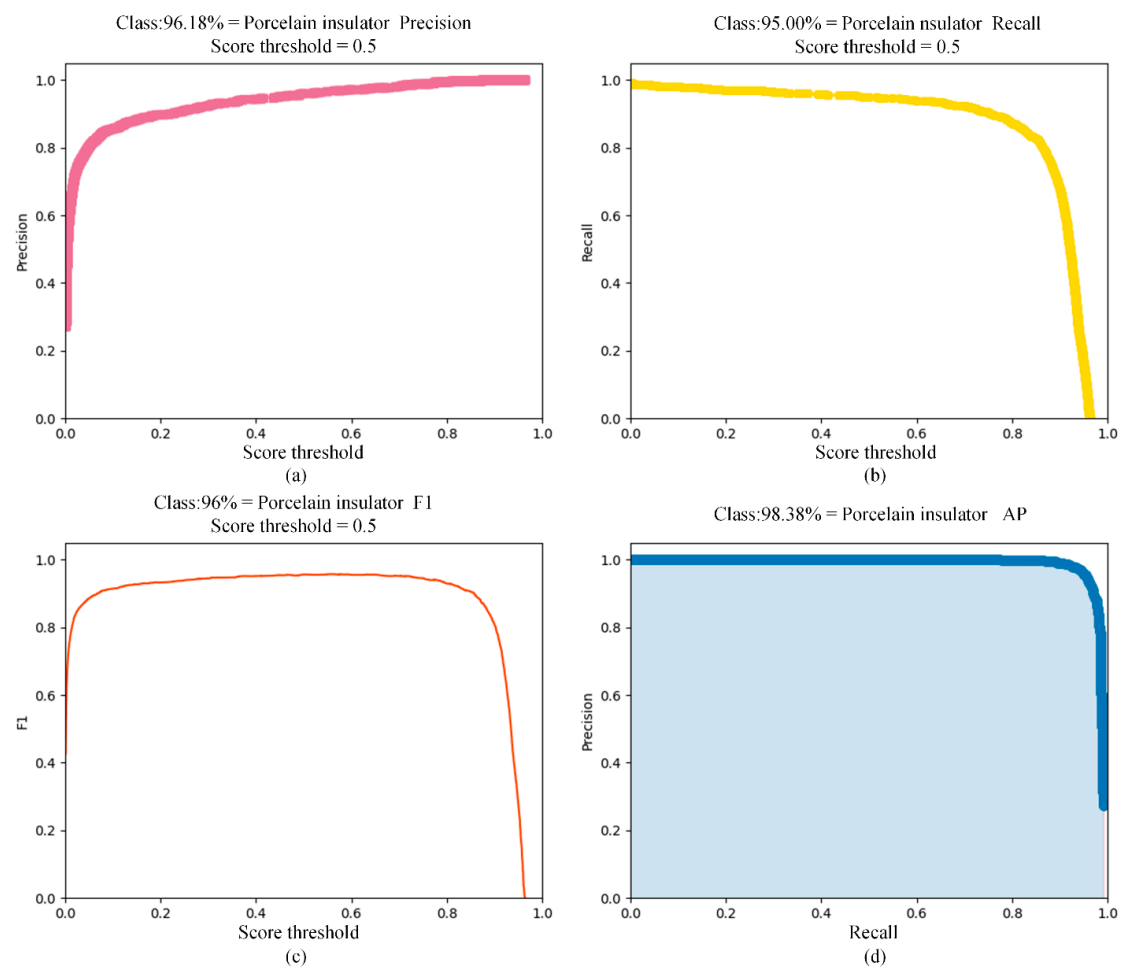


Figure 15. Evaluation results include P, R, F1, and AP. (a) P; (b) R; (c) F1; (d) AP.

4. Discussions

4.1. Synthetic Dataset vs. Real Dataset

It takes 11.6 s to synthesize an image and annotation for a dataset based on virtual synthesis techniques; this would take less time if better hardware were used. The average time to annotate an image is 38 ms in the synthetic dataset, which is much lower than the average annotation time of 16.4 s for the real dataset. Therefore, Synthetic datasets are easily obtained without considering the difficulty of the generation and annotation. However, it is costly and difficult to expand a fitting dataset of a PTLs. Once the volume of real data increases, the time and the economic cost also substantially increase relative to the synthetic dataset. In addition, dataset annotation can also output depth maps and instance segmentation labels with much better speed and accuracy than manual. When generating synthetic data, there is no human subjective error in annotation accuracy and the scale of synthetic data would be easily expanded with scalability and operability.

4.2. Potential for the Synthetic Dataset Based on the Prior Series Data

The synthesis of the dataset can contain any pose of the model and any environment, synthesizing some unimportant images if the dataset synthesis is not constrained. For fitting recognition, the synthetic dataset covering too wide a range is not conducive to DL training because the collect images with viewpoints outside the motion trajectory are unrelated features. For example, the walking-mode robots walk along a ground wire and the robots cannot collect images where the viewpoint is not on the trajectory. The viewpoints of images are not on their trajectory, causing the DL model to learn unrelated features, taking more datasets to train the network to improve accuracy. However, prior series data can achieve better training results with a smaller dataset. The disorderly and random datasets are endowed with space–time properties along the ground wire, showing the great scalability and tunability.

4.3. False Detections and Omissions

The trained model is more likely to detect the round-like white objects as porcelain insulators when the volume of the synthetic dataset is 6 K to train the DL model. When the volume of the synthetic dataset increases to 10.8 K, this false detection is significantly reduced and the missed detections also decrease. This false detection would also be easily solved during the deployment of the trained model. One reason is that the viewpoint of the test set images is not on the robot trajectory and the actual viewpoint is located in the sky. All objects are shrunk in the image except for the PTLs, reducing the number of round-like white objects. Second, the identifications of insulators in the video stream are continuous and false detections are often intermittent, which would be shielded from false detections by this feature. The trained model can more accurately detect the images of the background with vegetation and bare soil and almost no missed detection. Some images in the front view of insulators on the tarmac are missed detection because the series information of these images does not belong to the prior series data. However, there are few missed detections in the elevation view and top view directions of insulators on the tarmac, indicating good generalization of the model.

4.4. Comparison with Other Dataset Expanding Methods

A summary of the methods on the expanding dataset is shown in Table 4. The main comparisons with the related studies are as follows: (1) Our proposed approach of the synthetic dataset based on the prior series data could obtain a dataset with a large number of important features. The trained model using this synthetic dataset could identify targets in complex backgrounds, showing good generalization and accuracy. The synthesis dataset for these studies [18–20] contains unimportant features that may be learned by the DL model. (2) The expanding dataset based on real images [14,16,17,20] can significantly increase the volume of the dataset, but collecting real images is difficult, especially for the fitting dataset. Our dataset synthesis is completely virtual; therefore, the synthetic images

would not be subject to topographic restrictions like real images, reducing the difficulty of generating images. (3) The trained model using the synthetic dataset for the study [18] has great accuracy for identifying regular objects in the single-color background. Our synthetic dataset based on prior series data could improve the recognition accuracy of the DL model in complex backgrounds since the synthetic dataset has a lot of important features.

Table 4. Summary of the methods for expanding dataset.

Core Idea	Method	Recognition Target	Data Volume	Sensor Type	Vehicle Type	Performance Metrics
Data augmentation [16]	Faster RCNN	Insulator	—	RGB Camera	NVIDIA RTX2080ti	Highest accuracy: 90.2%
Random splicing [17]	cGAN	Insulator	8 K	Infrared camera and RGB camera	NVIDIA GTX1080	Highest accuracy: 85%
Image processing [14]	YOLO v4	Insulator	15 K	RGB Camera	NVIDIA GTX1060	AP: 82.9%
Generating dataset using virtual 3D models [18]	CNN	Screw, nut, and washer	—	RGB Camera	Samsung S7, Epson Moverio M350	Accuracy: 91% to 99%
Mixing real and simulated samples [19]	CNN	Insulator	18 K	RGB Camera	Omnisky SCW4750 workstation	Best mixing ratio: 2.0 Accuracy: 97.9%
Overlaying synthetic wires on flight videos [20]	CNN	Wire	68 K	RGB Camera	NVIDIA Jetson TX2	AP: 73%
Generating dataset base prior series data	YOLOX	Insulator	10.8 K	D 435i	NVIDIA GTX1660	AP: 98%

5. Conclusions

In this study, a novel auto-synthesis dataset approach is proposed for fitting recognition using the prior series data. The generated synthetic dataset by this approach is validated on the YOLOX model. The main conclusions are as follows:

1. The synthetic dataset is generated by the Blender script using the prior series data. Per 720×1280 pixel image and its annotation are generated in only 11.6 s. The efficiency of synthesizing the dataset is substantially improved compared with humans collecting images and annotating.
2. The formulation of synthesis rules based on prior series data can control the properties of the dataset. The synthesis dataset has strong scalability, operability, and tunability.
3. Training the YOLOX model using a synthesis dataset without real samples can obtain good models. The trained model achieves an mAP of 0.98 on the test set of real samples, indicating that the trained model on the synthetic dataset has a great generalization to recognize real samples. The research results suggest that training the DL model using a synthesis dataset is promising.

The limitations of the proposed approach mainly focus on two aspects. One is that the performance of our GPU device is not excellent and the time cost of generating a dataset will be very high if the volume of dataset needed is very large. We could replace the GPU with more excellent performance and optimize the algorithms in the future to reduce the time cost of the generated dataset. The other is that the depth maps and instance segmentation labels are not generated in this paper, and we will refine our program to generate them in the following work.

Author Contributions: Conceptualization, J.Z. and J.L.; methodology, J.Z., J.L. and X.Q.; software, J.Z. and B.J.; validation, B.L., Y.Z. and J.S.; formal analysis, H.L.; investigation, J.Z., X.Q., J.L., B.J., B.L., H.L., Z.L., J.S. and Y.Z.; resources, J.Z., B.J. and B.L.; data curation, J.Z., X.Q., J.L., B.J., B.L., H.L., Z.L., J.S. and Y.Z.; writing—original draft preparation, J.Z. and J.L.; writing—review and editing, J.Z., J.L. and X.Q. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (grant nos. 62063030 and 62163032), the Financial Science and Technology Program of the XPCC (grant no. 2021DB003), and the High-level Talent Project of Shihezi University (grant nos. RCZK2018C31 and RCZK2018C32).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We published our dataset at https://github.com/zjlanthe/synthetic_data_insulator_blender, accessed on 12 April 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Surya, A.S.; Harsono, B.B.S.D.A.; Tambunan, H.B.; Mangunkusumo, K.G.H. Review of Aerial Vehicle Technology for Transmission Line Inspection in Indonesia. In Proceedings of the Electrical Power Electronics, Malang, Indonesia, 26–28 August 2020. [CrossRef]
2. Zhang, W.; Ning, Y.; Suo, C. A Method Based on Multi-Sensor Data Fusion for UAV Safety Distance Diagnosis. *Electronics* **2019**, *8*, 1467. [CrossRef]
3. Qin, X.; Wu, G.; Lei, J.; Fan, F.; Ye, X.; Mei, Q. A Novel Method of Autonomous Inspection for Transmission Line based on Cable Inspection Robot LiDAR Data. *Sensors* **2018**, *18*, 596. [CrossRef] [PubMed]
4. Huang, L.; Xu, D.; Zhai, D. Research and Design of Space-Sky-Ground Integrated Transmission Line Inspection Platform Based on Artificial Intelligence. In Proceedings of the 2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2), Beijing, China, 20–22 October 2018. [CrossRef]
5. Alhassan, A.B.; Zhang, X.; Shen, H.; Xu, H. Power transmission line inspection robots: A review, trends and challenges for future research. *Int. J. Electr. Power* **2020**, *118*, 105862. [CrossRef]
6. Bian, J.; Hui, X.; Zhao, X.; Tan, M. A Novel Monocular-Based Navigation Approach for UAV Autonomous Transmission-Line Inspection. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Madrid, Spain, 1–5 October 2018. [CrossRef]
7. Hui, X.; Bian, J.; Zhao, X.; Tan, M. Vision-based autonomous navigation approach for unmanned aerial vehicle transmission-line inspection. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1729881417752821. [CrossRef]
8. Côté-Allard, U.; Campbell, E.; Phinyomark, A.; Laviolette, F.; Gosselin, B.; Scheme, E. Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features. *Front. Bioeng. Biotechnol.* **2020**, *8*, 158. [CrossRef] [PubMed]
9. Titov, E.; Limanovskaya, O.; Lemekh, A.; Volkova, D. The Deep Learning Based Power Line Defect Detection System Built on Data Collected by the Cablewalker Drone. In Proceedings of the 2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), Novosibirsk, Russia, 21–27 October 2019. [CrossRef]
10. Yao, N.; Zhu, L. A Novel Foreign Object Detection Algorithm Based on GMM and K-Means for Power Transmission Line Inspection. *J. Phys. Conf. Ser.* **2020**, *1607*, 012014. [CrossRef]
11. Mukherjee, A.; Kundu, P.K.; Das, A. Transmission Line Faults in Power System and the Different Algorithms for Identification, Classification and Localization: A Brief Review of Methods. *J. Inst. Eng. Ser. B* **2021**, *102*, 855–877. [CrossRef]
12. Miao, X.; Liu, X.; Chen, J.; Zhuang, S.; Fan, J.; Jiang, H. Insulator Detection in Aerial Images for Transmission Line Inspection Using Single Shot Multibox Detector. *IEEE Access* **2019**, *7*, 9945–9956. [CrossRef]
13. Wen, Q.; Luo, Z.; Chen, R.; Yang, Y.; Li, G. Deep Learning Approaches on Defect Detection in High Resolution Aerial Images of Insulators. *Sensors* **2021**, *21*, 1033. [CrossRef] [PubMed]
14. Rahman, E.U.; Zhang, Y.; Ahmad, S.; Ahmad, H.I.; Jobaer, S. Autonomous Vision-Based Primary Distribution Systems Porcelain Insulators Inspection Using UAVs. *Sensors* **2021**, *21*, 974. [CrossRef] [PubMed]
15. Mill, L.; Wolff, D.; Gerrits, N.; Philipp, P.; Kling, L.; Vollnhals, F.; Ignatenko, A.; Jaremenko, C.; Huang, Y.; De Castro, O.; et al. Synthetic Image Rendering Solves Annotation Problem in Deep Learning Nanoparticle Segmentation. *Small Methods* **2021**, *5*, 2100223. [CrossRef] [PubMed]
16. Song, C.; Xu, W.; Wang, Z.; Yu, S.; Zeng, P.; Ju, Z. Analysis on the Impact of Data Augmentation on Target Recognition for UAV-Based Transmission Line Inspection. *Complexity* **2020**, *2020*, 3107450. [CrossRef]
17. Chang, W.; Yang, G.; Wu, Z.; Liang, Z. Learning Insulators Segmentation from Synthetic Samples. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018. [CrossRef]
18. Židek, K.; Lazorič, P.; Pitel, J.; Hošovský, A. An Automated Training of Deep Learning Networks by 3D Virtual Models for Object Recognition. *Symmetry* **2019**, *11*, 496. [CrossRef]
19. Cheng, H.; Chen, R.; Wang, J.; Liu, X.; Zhang, M.; Zhai, Y. Study on insulator recognition method based on simulated samples expansion. In Proceedings of the 2018 Chinese Control and Decision Conference (CCDC), Shenyang, China, 9–11 June 2018. [CrossRef]
20. Madaan, R.; Maturana, D.; Scherer, S. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017. [CrossRef]

21. Barth, R.; Ijsselmuiden, J.; Hemming, J.; Henten, E.J.V. Data synthesis methods for semantic segmentation in agriculture: A Capsicum annum dataset. *Comput. Electron. Agric.* **2018**, *144*, 284–296. [[CrossRef](#)]
22. Boikov, A.; Payor, V.; Savelev, R.; Kolesnikov, A. Synthetic Data Generation for Steel Defect Detection and Classification Using Deep Learning. *Symmetry* **2021**, *13*, 1176. [[CrossRef](#)]
23. Cheng, K.; Tahir, R.; Eric, L.K.; Li, M. An analysis of generative adversarial networks and variants for image synthesis on MNIST dataset. *Multimed. Tools Appl.* **2020**, *79*, 13725–13752. [[CrossRef](#)]
24. Benitez-Garcia, G.; Yanai, K. Ketchup GAN: A New Dataset for Realistic Synthesis of Letters on Food. In Proceedings of the International Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia, Taipei, Taiwan, 16–19 November 2021; pp. 8–12. [[CrossRef](#)]
25. Tian, Y.; Li, X.; Wang, K.; Wang, F. Training and testing object detectors with virtual images. *IEEE/CAA J. Autom. Sin.* **2018**, *5*, 539–546. [[CrossRef](#)]