

Article

# A Real-Time Strand Breakage Detection Method for Power Line Inspection with UAVs

Jichen Yan <sup>1,2</sup>, Xiaoguang Zhang <sup>1,2</sup>, Siyang Shen <sup>1,2,3</sup>, Xing He <sup>1,2</sup>, Xuan Xia <sup>1,2</sup>, Nan Li <sup>1,2,4,\*</sup>,  
Song Wang <sup>5</sup>, Yuxuan Yang <sup>5</sup> and Ning Ding <sup>1,2,4</sup>

<sup>1</sup> Institute of Robotics and Intelligent Manufacturing, The Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China

<sup>2</sup> Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen 518129, China

<sup>3</sup> School of Data Science, The Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China

<sup>4</sup> Joint Laboratory for Electric Power Robots of China Southern Power Grid Co., Ltd., Shenzhen 518129, China

<sup>5</sup> CSG Electric Power Research Institute Co., Ltd., Guangzhou 510663, China

\* Correspondence: [linan@cuhk.edu.cn](mailto:linan@cuhk.edu.cn)

**Abstract:** Power lines are critical infrastructure components in power grid systems. Strand breakage is a kind of serious defect of power lines that can directly impact the reliability and safety of power supply. Due to the slender morphology of power lines and the difficulty in acquiring sufficient sample data, strand breakage detection remains a challenging task. Moreover, power grid corporations prefer to detect these defects on-site during power line inspection using unmanned aerial vehicles (UAVs), rather than transmitting all of the inspection data to the central server for offline processing which causes sluggish response and huge communication burden. According to the above challenges and requirements, this paper proposes a novel method for detecting broken strands on power lines in images captured by UAVs. The method features a multi-stage light-weight pipeline that includes power line segmentation, power line local image patch cropping, and patch classification. A power line segmentation network is designed to segment power lines from the background; thus, local image patches can be cropped along the power lines which preserve the detailed features of power lines. Subsequently, the patch classification network recognizes broken strands in the image patches. Both the power line segmentation network and the patch classification network are designed to be light-weight, enabling efficient online processing. Since the power line segmentation network can be trained with normal power line images that are easy to obtain and the compact patch classification network can be trained with relatively few positive samples using a multi-task learning strategy, the proposed method is relatively data efficient. Experimental results show that, trained on limited sample data, the proposed method can achieve an F1-score of 0.8, which is superior to current state-of-the-art object detectors. The average inference speed on an embedded computer is about 11.5 images per second. Therefore, the proposed method offers a promising solution for conducting real-time on-site power line defect detection with computing sources carried by UAVs.

**Keywords:** power line inspection; strand breakage detection; power line segmentation; multi-task learning; unmanned aerial vehicle; real-time detection



**Citation:** Yan, J.; Zhang, X.; Shen, S.; He, X.; Xia, X.; Li, N.; Wang, S.; Yang, Y.; Ding, N. A Real-Time Strand Breakage Detection Method for Power Line Inspection with UAVs. *Drones* **2023**, *7*, 574. <https://doi.org/10.3390/drones7090574>

Academic Editor: Abdessattar Abdelkefi

Received: 20 July 2023

Revised: 19 August 2023

Accepted: 5 September 2023

Published: 10 September 2023

Corrected: 7 November 2023



**Copyright:** © 2023 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/>).

## 1. Introduction

Electric power lines are critical infrastructure components that transport electrical energy from power generation plants to users. Since most power lines are implemented in an outdoor environments, they are vulnerable to various types of damage that can impact the reliability and safety of the electricity supply. One typical type of damage is the breakage of strands, which can lead to electrical faults and power outages if they are not eliminated in a timely manner. Thus, detecting broken strands on power lines is essential for ensuring uninterrupted power supply and preventing accidents.

In recent years, due to the high maneuverability, good economic property, and capability of high-quality image acquisition of unmanned aerial vehicles (UAVs), they have been more and more widely applied in power line inspection [1,2]. They have significantly eased the process of capturing high-resolution images of power lines from different angles. However, finding the defects in power lines and other components in the obtained images still relies on manual interpretation or centralized processing on the cloud server with large deep learning models. Due to the rise in labor costs and the rapid growth of data volume, inefficient manual interpretation is gradually being phased out. The centralized processing manner suffers from a huge communication and computational burden, and the delay of offline processing [3]. Therefore, detecting the power line defects in real-time on site with algorithms and computational resources carried by UAVs is a preferable method. There are already many researchers developing online image processing systems for UAV-based power line inspection [3,4]. However, the existing technology cannot meet the requirements of practical applications in terms of accuracy and efficiency.

Furthermore, developing an accurate and robust strand breakage detection method is challenging. On the one hand, the power lines and broken strands are thin objects; their features become indistinct in images captured by UAVs from a distance. On the other hand, obtaining sufficient sample data for training a deep learning model is extremely hard since strand breakage occurs rarely and, once discovered, it is immediately eliminated to prevent serious consequences.

In response to the above requirements and challenges, this paper proposes a novel method for detecting broken strands on power lines within images captured by UAVs. The proposed method features a multi-stage pipeline consisting of power line segmentation, power line patch cropping, and patch classification. Since normal power line images are easy to obtain, a robust power line segmentation model can be trained. As strand breakage only occurs on power lines, image patches are cropped out along each segmented power line and then fed to the patch classification network to recognize whether there is strand breakage in each patch. Since the power line patches are cropped from high-resolution power line images, they contain rich details that benefit the recognition of thin broken strands. In addition, both the segmentation network and patch classification network are designed to be light-weight. As a result, the proposed method can achieve high accuracy and efficiency in detecting power line strand breakage, while it can be trained with relatively few defect samples. It is capable of running in real-time when deployed on an embedded computing device carried by an UAV.

The contribution of this paper can be summarized as follows.

- A multi-stage pipeline for real-time strand breakage detection in power line inspection images is proposed, which consists of power line segmentation, power line patch cropping, and patch classification. Compared to conventional end-to-end pipeline, the proposed pipeline reduces the need for strand breakage samples that are hard to obtain and makes full use of the detailed features within the power line areas in original images.
- An efficient power line segmentation network is proposed, which exploits a shallow and slim backbone and multi-scale feature fusion branches. The network achieves superior segmentation accuracy and efficiency over its counterparts.
- A power line fitting method is proposed based on connected component analysis and the least squares method to fit each power line in the power line segmentation results. Thus, image patch cropping can be conducted to extract local image patches along each power line.
- A light-weight classification network is devised for recognizing strand breakage in power line image patches. Multi-task learning is applied to better train the network with limited data.

The rest of the paper is organized as follows. Section 2 provides an overview of related works on power line segmentation and strand breakage detection. Section 3 provides detailed description of the proposed strand breakage method and its components. Section 4

presents the experimental setup and results. Finally, Section 5 concludes this research and introduces future works.

## 2. Related Works

As intelligent inspection and defect detection technologies show great potential to significantly reduce the cost and risk of manual operation in power system inspection, related research has been conducted for more than a decade. In this section, we will present a literature review on related works which covers both traditional computer vision methods and deep learning-based approaches for power line segmentation and detection of broken strands.

### 2.1. Power Line Segmentation

In the past, power line segmentation tasks often relied on purely hand-crafted models. These hand-crafted models were typically constructed based on low-level local features such as gradients, brightness, texture, and other prior information from wire images. Chen et al. [5] developed the Cluster Radon Transform to extract linear features of power lines from remote sensing imagery and devised a set of rules to distinguish power lines from other linear features like roads. Zhou et al. [6] developed an edge detection method for power line detection, which selects optimal parameters for changing backgrounds and, hence, overcomes the threshold problem in other methods. Du et al. [7] used Hough Transform (HT) butterflies to prove that the HT is not only effective for detecting and locating linear-shaped targets but also for curved wire objects. In the aforementioned works, it has been demonstrated that hand-crafted models constructed using traditional computer vision methods are feasible for power line segmentation tasks. However, these methods still suffer from issues such as low detection accuracy and limited generalizability.

In comparison, significant progress has been made with convolutional neural networks (CNNs) [8,9]. Building upon these advancements, CNNs have also been applied on power line segmentation. However, power line segmentation tasks based on neural networks often face challenges in achieving high-quality feature extraction due to the slender morphology of power lines. The network architecture proposed by Chang et al. [10] suggests fusing shallow and deep features within the neural network. They introduced a compact neural network composed of a generator and a discriminator within the conditional generative adversarial network (cGAN) framework [11]. Additionally, skip connections were incorporated between each encoder and decoder in the network architecture. In the work of Zhang et al. [12], a method of multi-level feature map fusion was introduced. They proposed a convolutional neural network based on the VGG16 architecture [13], which is capable of obtaining hierarchical predictions from different convolutional layers. By leveraging multiple levels of information, the network can automatically learn how to combine them and generate satisfactory fused outputs. In recent work, Choi et al. [14] attempted to generate the location information of power lines in input images by introducing attention into a two-stage semi-supervised learning framework. In the first stage of their method, they utilized the information from various layers of the VGG network to form an Attention Localization Mask (ALM); in the second stage, the mask and sub-network were used to generate the contour information of the power lines. However, their proposed method exhibits a significant increase in computational complexity compared to conventional one-stage semantic segmentation networks, making it challenging to deploy in practical applications. On the other hand, He et al. [15] explored the use of a more powerful baseline and network light-weight design in power line segmentation tasks. They employed a light-weight backbone structure (DFC-GhostNet [16]) for feature extraction and combined it with contextual information features to enhance the U-Net algorithm [17]. Furthermore, they designed a hybrid feature extraction module based on convolution and transformers to optimize deep semantic features, improving the model's ability to locate towers and transmission lines in complex environments.

## 2.2. Strand Breakage Detection

Similar to power line segmentation tasks, early methods for detecting broken strands in power lines often relied on handcrafted models. Researchers modeled the presence or absence of defects in power lines by utilizing low-level local features such as gradients, brightness, texture, and other prior information derived from power line images. These handcrafted models were then used for defect detection tasks related to power lines. Ishino et al. [18] constructed handcrafted models for defect-free power line images using statistical information such as brightness, texture, and morphology. They utilized this model to perform simple classification of broken strand power lines. On the other hand, Mao et al. [19] employed the Histogram of Oriented Gradients (HOG) algorithm to extract gradient features from power line images and used a hybrid classifier composed of the Support Vector Machine (SVM) algorithm to classify normal power lines, broken strand power lines, and obstacles. In the study conducted by Jalil et al. [20], the Canny edge detector and HT were exploited for power line detection. Then, within the corresponding IR image, they computed the histogram of the image, and performed Otsu's thresholding to identify the faults or hot spots.

In recent studies, deep learning-based object detectors have been widely applied in industrial scenarios related to the power system [21,22]. Existing methods usually exploit a two-stage process to locate the regions of power line strand breakage. These detectors typically employ a sliding window approach in the first stage to capture candidate regions that may contain faults and, in the second stage, they discriminate the regions where actual faults occur. In the study by Wang et al. [23], a CNN-based power line fault detection method was proposed. In the first step, a CNN is used in conjunction with the sliding window method to predict all parts of the input image and generate an output map. In the second step, the output map is preprocessed to enhance its localization characteristics. Finally, the target detection is completed based on the preprocessed output map information. On the other hand, Xu et al. [24] applied Faster R-CNN [25] to detect fracture areas in power lines. However, what distinguishes their work is the introduction of an attention mechanism into the feature extraction network of Faster R-CNN. This mechanism guides the network to focus specifically on the parts of the input image directly relevant to fractured regions, thereby enhancing the model's training effectiveness and robustness.

## 2.3. The Advancement of Our Approach

It is worth mentioning that, in order to initially locate the defects in electrical components or power lines, the methods proposed by Xu et al. [24] and Wang et al. [23] employed a sliding window approach to capture candidate regions of defects from the entire input image. However, their sliding window approach involves extracting local image patches from the entire image as targets. Such a method generates a large number of invalid candidate regions and only a few of them actually contain true defects. As a result, the network often needs to recognize the image patches with actual defects from a large number of candidates, which leads to inefficient defect detection. In contrast, in our proposed method, the sliding window is applied only within the power line regions to capture local images of the power lines. Therefore, the detector only needs to identify defects in a small number of candidate regions. On the other hand, the method proposed by Xu et al. [24] attempts to use Faster R-CNN to detect broken strand regions in each sliding window's obtained sub-image patch, which is computationally expensive and not suitable for deployment in low-power UAVs.

Our approach has three improvements compared to previous works: (1) both the candidate region acquisition and the following classification of candidate regions are based on deep neural networks, which benefits the overall accuracy and generalization; (2) the candidate regions are only extracted from the segmented power line areas so as to screen out most of the backgrounds and significantly reduce redundant computing in the following defect recognition procedure; (3) we proposed a novel light-weight power line segmentation network and an image patch classification network that feature both

high accuracy and efficiency, making the proposed method run at an inference speed of 11.5 images per second.

### 3. Materials and Methods

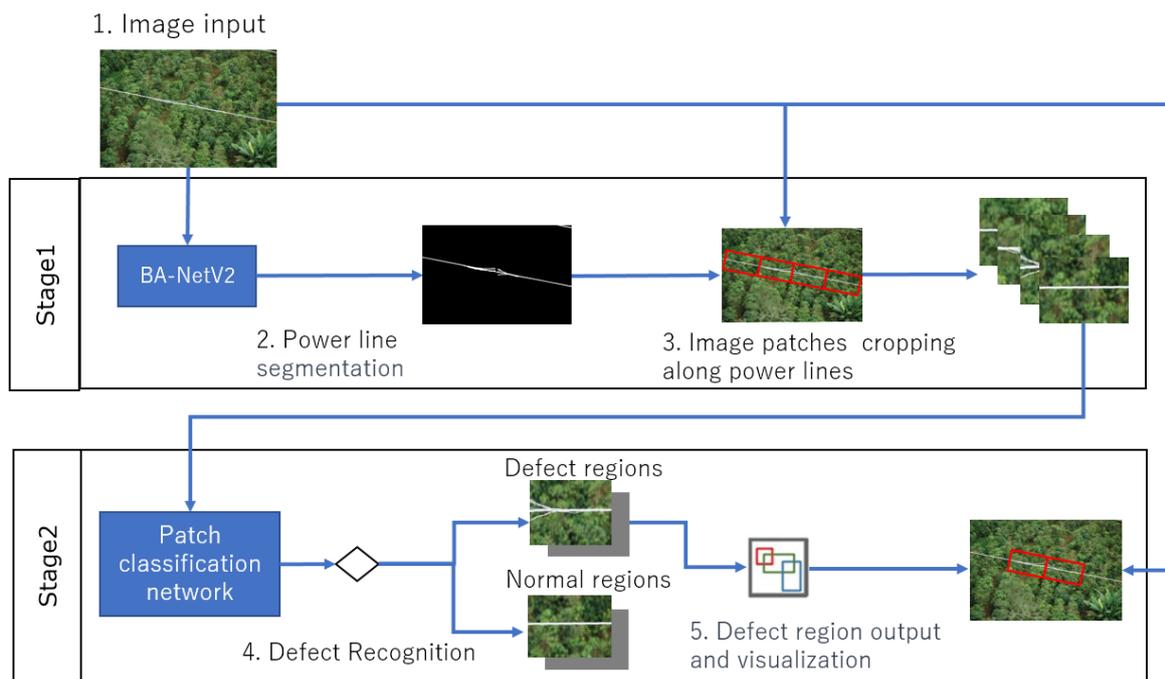
In this section, we will provide a detailed description of the architecture of the proposed power line strand breakage defect detector. Additionally, we will provide a comprehensive explanation of the methods involved and the neural networks employed in this detector.

#### 3.1. Overall Pipeline

In object detection tasks, the detection and segmentation of small objects pose significant challenges due to low object resolution and small object size [26,27]. In the power line inspection images captured by UAVs, the segmentation of power lines faces similar challenges. Therefore, to address these challenges and achieve accurate and efficient localization of power line strand breakage areas using remote sensing imagery, we propose an advanced two-stage defect detector. This detector operates in two stages: In the first stage, it performs power line segmentation to extract the power lines from diverse backgrounds and conducts local image cropping of all regions containing power lines in the input image. These cropped images are then passed to the second stage. In the second stage, the detector performs classification of the local power line images from the first stage. It identifies the presence of defects and visualizes the position information of the defective regions within the original image. This two-stage defect detector aims to precisely locate and identify strand breakage areas in power lines, enabling effective monitoring and maintenance of power lines.

As shown in Figure 1, the diagram illustrates the specific workflow of the power line strand breakage detector. In the first stage, the detector utilizes a semantic segmentation network (BA-NetV2) to perform pixel-level segmentation on the power line images captured by UAVs. Subsequently, the detector utilizes the binary segmentation result to extract essential information regarding the power lines, including the coordinates of the starting and ending points, as well as their length and width. Finally, the detector employs a sliding window approach along the direction of the power lines, starting from the starting point and moving towards the ending point. It scans the power line regions encountered by the sliding window and performs local image cropping to cover the entire length of the power line. It is worth noting that the dimensions (length and width) of the sliding window and the stride used during the sliding process can be adjusted. This sliding window approach ensures comprehensive coverage of the power lines and captures detailed information about any defective regions along the power lines.

Once the detector completes the cropping of local image patches from the power line images, each set of image patches are passed to the second stage, where the patch classification network filters the image patches and identifies the regions related to strand breakage along the power lines. If the detector identifies an image patch as containing a strand breakage, the corresponding region in the original image is recognized as a defect area and a bounding box is generated to enclose this region. Subsequently, the bounding box is annotated on the original image to visually indicate the detected defect area.



**Figure 1.** Overview of the proposed method’s pipeline. The proposed method consists of two key stages: power line local image generator (Stage 1) and defect recognition (Stage 2). Stage 1 takes the entire power line image as input and utilizes a sliding window approach to capture local images of the power lines, based on power line segmentation results. Stage 2 takes the power line local images generated by Stage 1 as input and employs a patch classification network to classify them into normal and defect regions. For the defect regions, Stage 2 visualizes them on the original image.

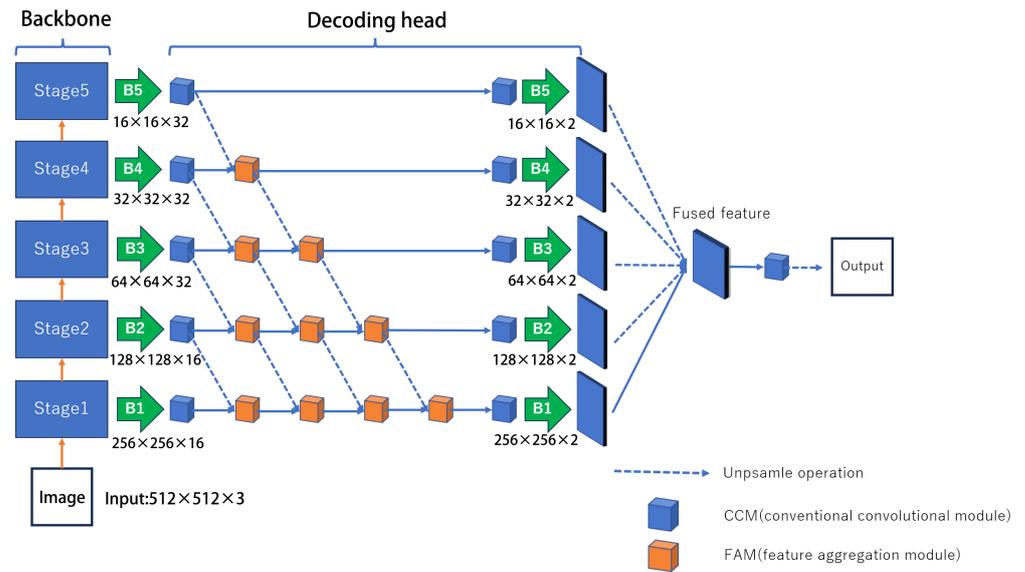
### 3.2. Power Line Segmentation Network

The power line segmentation task differs from common segmentation scenarios, as traditional segmentation models struggle to accurately predict the contours of power lines due to their slender shape characteristics. Moreover, aerial images of power lines are often contaminated with significant amounts of background noise and pseudo-targets that resemble the morphology of power lines, such as wires, branches, and weeds. These sources of interference can increase the false positive rate of the segmentation network for power lines.

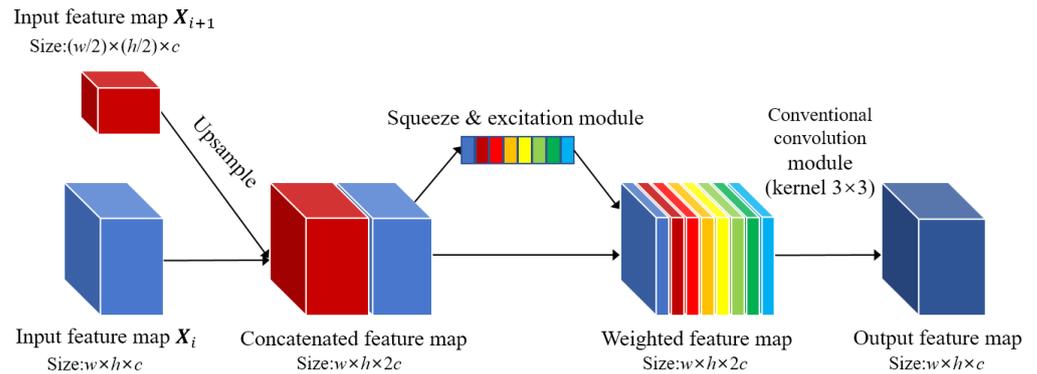
#### 3.2.1. Baseline Network

BA-Net is a light-weight segmentation network proposed in the previous work of our team [28]. It has previously been used for image segmentation tasks in the field of agriculture and has also demonstrated effectiveness in other scenarios. It achieves high efficiency while maintaining good accuracy in image segmentation. In this work, we use BA-Net as the baseline and improve it to better adapt to the power line segmentation task.

As illustrated in Figure 2, BA-Net is composed of a light-weight backbone with five convolutional stages and five parallel branches. Each of the stages in the backbone involves two inverted residual blocks (IRB) as used in MobileNetV2 [29], except for the first stage which only contains a  $5 \times 5$  convolutional layer. When the feature maps pass through each stage, their height and width are reduced by half. The five parallel branches feature a bi-path fusion tree structure to perform efficient multi-scale feature fusion, by building up connections over adjacent side outputs through feature aggregation modules (FAMs). A conventional convolutional module (CCM) is used at the beginning and the end of each branch. Each CCM consists of a  $3 \times 3$  convolutional layer, followed by a normalization operation and Rectified Linear Unit (ReLU) activation. The detailed structure of FAM is illustrated in Figure 3.



**Figure 2.** Brief illustration of the structure of BA-Net.  $w \times h \times c$  means the image or the feature map has a resolution of  $w \times h$  and a channel number of  $c$ . For detailed structure illustration, please refer to [28].



**Figure 3.** Illustration of the feature aggregation module (FAM).  $X_i$  and  $X_{i+1}$  denote feature maps in the  $i$ th and  $(i + 1)$ th branch, respectively.  $w \times h \times c$  means the feature map has a resolution of  $w \times h$  and a channel number of  $c$ .

The FAM concatenate feature maps from adjacent branches and dynamically assigns different weights to different channels of the concatenated feature maps with the Squeeze and Excitation (SE) modules [30]. Each FAM has a CCM connected to the end of the SE module. At the end of the branches, the predictions of all the branches are fused with a CCM to produce the final prediction. For detailed information, please refer to [28].

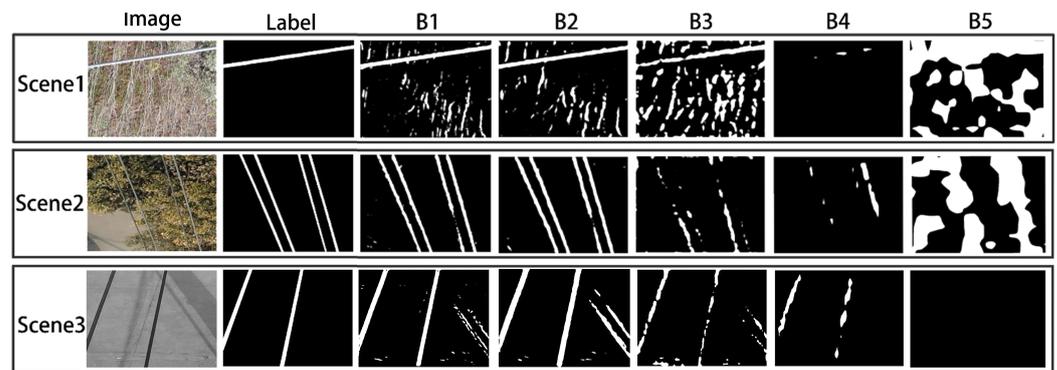
### 3.2.2. Improvement Guidelines

In order to design a network more suitable for power line segmentation, we propose three guidelines specifically tailored for power line segmentation as the primary scene. We have redesigned BA-Net according to these guidelines and achieved significant performance improvement in the power line segmentation scenario.

**Guideline 1.** Larger capacity of the network is needed to deal with diverse backgrounds. The original BA-Net is used for crop segmentation. The input images of BA-Net are mainly composed of plants and soil in a relatively uniform scene. However, the power line inspection images have different scenes with diverse backgrounds containing a large amount of information. Therefore, it is necessary to improve the network capacity. This can be done by increasing the width or depth of the network.

**Guideline 2.** Maintaining high-resolution feature map input can effectively preserve the morphological characteristics of power lines.

As illustrated in Figure 4, we present the output results of each parallel branch (total of five parallel branches) in BA-Net for three different scenarios. From B1 to B5, we upsampled the final output results of the five parallel branches, whose original resolutions have downsampling rates of 2, 4, 8, 16, and 32, respectively, to the input image size and performed visualization.



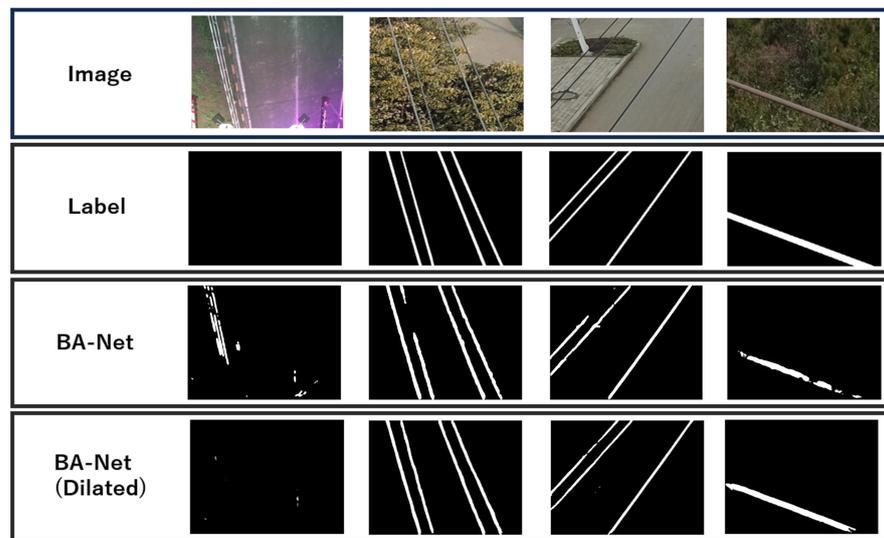
**Figure 4.** Visualization of the segmentation results from different branches of BA-Net in various scenarios.

Based on the visualization results, it can be observed that, although there are numerous false positive predictions in the segmentation results of branches B1, B2, and B3, the segmentation of power line contours is relatively accurate. Conversely, branches B4 and B5 suffer from significant loss of detailed information related to power line morphology. However, for larger-sized object segmentation, each branch of BA-Net is capable of achieving relatively accurate segmentation. Therefore, we believe that the contour information of power lines is composed of fine image details, which are often better represented in higher-resolution feature maps. Conversely, in low-resolution feature maps with higher downsampling rates, this spatial semantic information may be compromised.

**Guideline 3.** The network needs long-range semantic relation capturing ability when segmenting the elongated power lines.

Since power lines are elongated objects, there can be a long distance between different parts of the same power line. Therefore, the power line segmentation network needs to capture long-range semantic relations in the power line images. As shown in Figure 5, when observing the final segmentation results of BA-Net, we can observe the phenomena of discontinuity and misidentification in the segmentation of power lines. The regions that should have been identified as continuous power lines appear to be disconnected in the middle. Additionally, some pseudo-targets in the background that resemble the morphology of power lines are mistakenly identified as power lines by BA-Net.

We believe these phenomena are partially caused by the weakness of the network in capturing long-range semantic relations in the feature maps. When dilated convolutions are introduced to the branches of BA-Net, the occurrence of disconnections in wire segmentation results are reduced in most cases, as shown in Figure 5. As dilated convolution can enlarge the receptive field of convolution kernels, each kernel can learn the relation between pixels with longer distance in the feature maps. Moreover, adopting the dilated convolution also reduces the missegmentation of interfering objects such as tree branches and linear structures in the background.

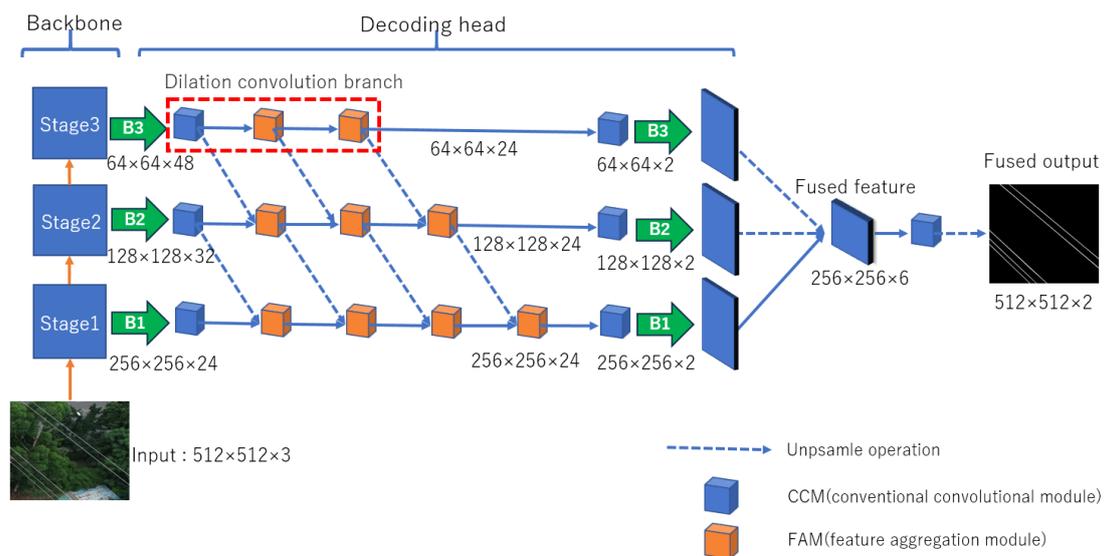


**Figure 5.** Visualization of the prediction results of BA-Net with or without dilated convolution.

### 3.2.3. BA-NetV2

As mentioned earlier, the first stage of the proposed strand breakage detector primarily relies on a power line segmentation network to obtain the basic information about the power lines and generate local images using a sliding window approach. Therefore, the accurate cropping of relevant local images is highly dependent on the accuracy of the power line segmentation network.

In this paper, we introduce a new light-weight power line segmentation model named BA-NetV2, as shown in Figure 6, which is based on the BA-Net architecture and specifically designed for the UAV-based power line inspection scenario. BA-NetV2 inherits the advantages of the BA-Net architecture, which combines efficient multi-scale feature extraction and fusion. Compared to the original design of BA-Net, the design of BA-NetV2 has the following improvements:



**Figure 6.** Illustration of the network architecture of BA-NetV2.  $w \times h \times c$  means the image or the feature map has a resolution of  $w \times h$  and a channel number of  $c$ .

- Expansion of neural network base channels. Considering the power line segmentation task has more diverse and complex backgrounds, we set larger channel numbers in the

backbone and the branches for BA-NetV2 compared to BA-Net. Therefore, BA-NetV2 obtains a larger capacity for complex feature extraction and representation.

- Reduction of parallel branches. As we observed that the low-resolution branches in BA-Net suffer from significant loss of detailed information and can have a negative impact on the overall prediction accuracy, thus, in BA-NetV2, we reduced the fourth and fifth branches. Such a design maintains high resolution of feature maps in the branches, which benefits the extraction and representation of the detailed features of power lines and reduces the computational complexity.
- Adoption of the dilated convolutions. We observed that the segmentation network needs long-range semantic relation capturing ability when segmenting the elongated power lines. Thus, we introduced dilated convolutions with different dilation factors into the third branch to build a feature extraction branch with a larger receptive field while enhancing the scale invariance of the feature maps.

The backbone network of BA-NetV2 is illustrated in Figure 6, which is a three-stage feature extraction network. The outputs of each stage are denoted as F1, F2, and F3, respectively. It is worth noting that each stage of the network is composed of stacked IRBs that are used in MobileNetV2 [29]. The design of the backbone network is similar to BA-Net, using the light-weight IRB module as the basic block and setting relatively small output channel numbers. Since BA-NetV2 is focused on extracting semantic information from more diverse and complex scenes while preserving the shape and spatial details relevant to the power lines, it has two main improvements compared to BA-Net. Firstly, unlike BA-Net with a 32-fold downsampling backbone network, BA-NetV2 adopts a three-stage backbone feature extraction network with an 8-fold downsampling rate to avoid the loss of spatial semantic information caused by excessive downsampling. Secondly, in BA-NetV2, the first, second, and third stages of the backbone network consist of 1, 2, and 3 IRBs. Additionally, to enhance the feature representation ability of the backbone network for power line images with diverse backgrounds, the channel number settings of the backbone network have been increased, with the channel numbers in stages 1 to 3 set to 24, 32, and 48, respectively.

The decoding head of BA-NetV2, as shown in Figure 6, consists of three parallel branches with a base channel number of 24 for each branch. These branches correspond to the feature maps outputted by the three stages of the backbone network, from bottom to top. Among them, branch B1 serves as the main branch of the BA-NetV2 decoding head and corresponds to the high-resolution feature map from the first stage of the backbone network, to extract spatial information about small objects from high-resolution feature maps. Simultaneously, the remaining branches (B2, B3) perform feature extraction and upsampling on low-resolution feature maps that contain rich semantic information.

Similarly to BA-Net, the FAMs are used for feature fusion between adjacent branches, while a CCM is used at the beginning and the end of each branch. The output channel number of the FAMs and CCMs of each branch is 24, except for the CCM at the end of each branch, which has an output channel number of 2. To enable the network to better extract long-range semantic relations in feature maps, dilated convolution is adopted in the first CCM and the two FAMs of the B3 branch. The dilation factors of the convolution kernel in these three modules were empirically set to 2, 3, and 5, respectively. Keeping the three dilation factors relatively prime can avoid the “gridding issue” which would be caused by successive dilated convolution.

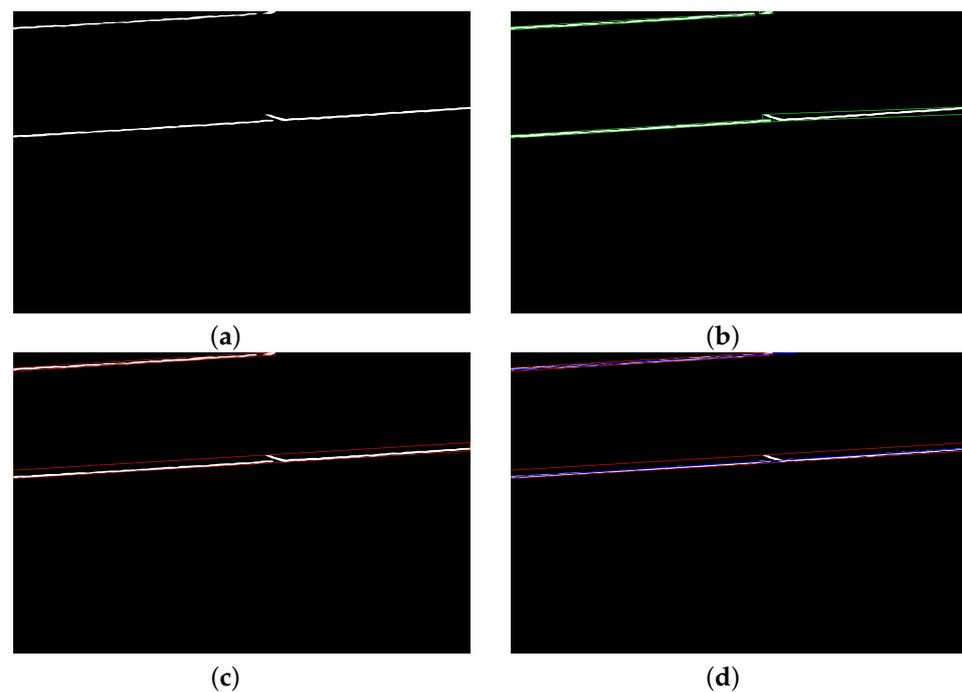
The outputs of the three branches are fused with a CCM with a  $1 \times 1$  convolutional kernel to produce the final prediction. When training the BA-NetV2, cross-entropy losses are calculated on the prediction of each branch and the final prediction, forming the multi-scale supervision.

#### 3.2.4. Postprocessing Method

After segmenting the images using BA-NetV2, it is necessary to extract the position and width information of the power lines in the images, specifically the endpoint coordinates, for subsequent image patch cropping. While most of the power lines in the images can be

well segmented, in some complex environments, such as when certain power line parts are occluded or when the background is intricate, a complete power line can be fragmented into multiple segments or pseudo-targets in the background can be misclassified as power line segments. Moreover, when multiple power lines are present in the same image, it is crucial to determine which line segments belong to the same power line. Therefore, we employ a hierarchical clustering approach to identify line segments belonging to the same power line, and subsequently obtain the position and width of the power line in the image.

Firstly, we perform connected component analysis on the segmented image, where each connected component represents an independent line segment. Since there may be a large number of connected components in an image, it is necessary to apply filtering based on the area and shape (aspect ratio of the bounding rectangles) of the connected components before clustering to speed up the clustering process. Secondly, the filtered connected components undergo hierarchical clustering, where the clustering distance is determined by the angles between the major axes of two different bounding rectangles and the distance from the center point of one bounding rectangle to the major axis of another bounding rectangle between the connected components. Subsequently, based on the clustering results of the connected components, we employ the least squares method to regress the slope and intercept of the center line of each power line, thus obtaining the position information of the power lines in the image. Finally, the width of each power line can be obtained by averaging the width values obtained by scanning along the normal vector direction of the fitted lines at a preset interval. The output of each key step of the postprocessing procedure is shown in Figure 7.

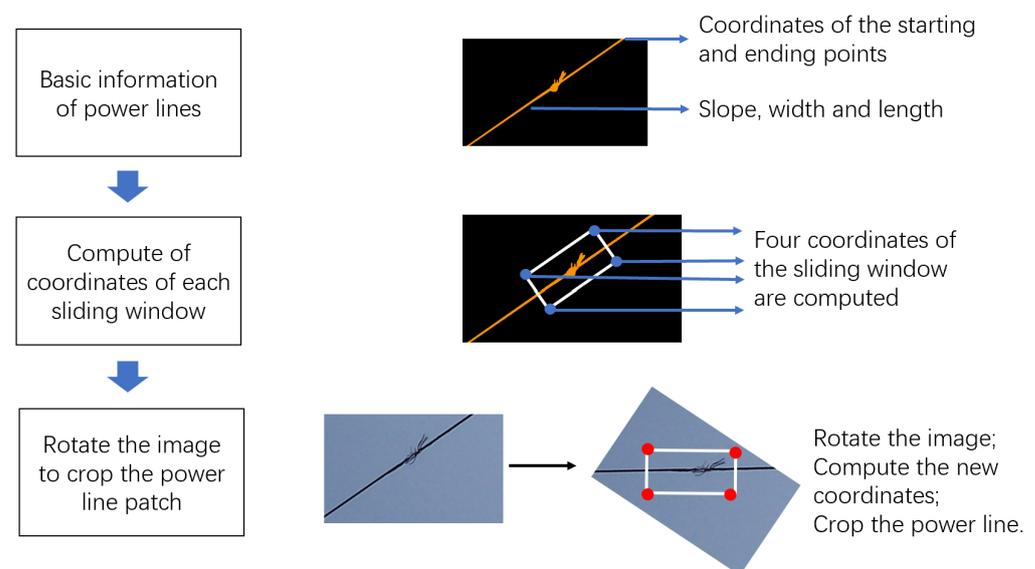


**Figure 7.** Output of each key step of the postprocessing procedure. (a) Segmented image. (b) Finding and filtering the minimum bounding rectangle of the obtained connected components, indicated by a green box. (c) Minimum bounding rectangles of connected components belonging to the same power line after hierarchical clustering, indicated by a red box. (d) Straight lines fitted using the least squares method for connected components belonging to the same power line, indicated by a blue line segment.

### 3.3. Power Line Patch Cropping Method

As previously stated, the cropping method involves the utilization of a sliding window approach subsequent to obtaining candidate regions along the power lines. This approach

ensures comprehensive coverage of the power lines and potential defects. To align the sliding window with each power line, adjustments are made to ensure parallelism. Additionally, the dimensions of the sliding window are adjustable to accommodate complex real-world scenarios effectively. The cropping process involves a two-step operation based on analytical geometry, as shown in Figure 8. Firstly, the coordinates of the four vertices of a specific sliding window are computed with information from the previous procedure. Subsequently, the image is rotated to achieve parallel alignment for the cropping process.



**Figure 8.** Illustration of the power line image patch cropping process. The left part shows the flow path of the cropping process. The right part visualizes the information corresponding to each block in the left part.

### 3.3.1. Coordinates of the Sliding Window

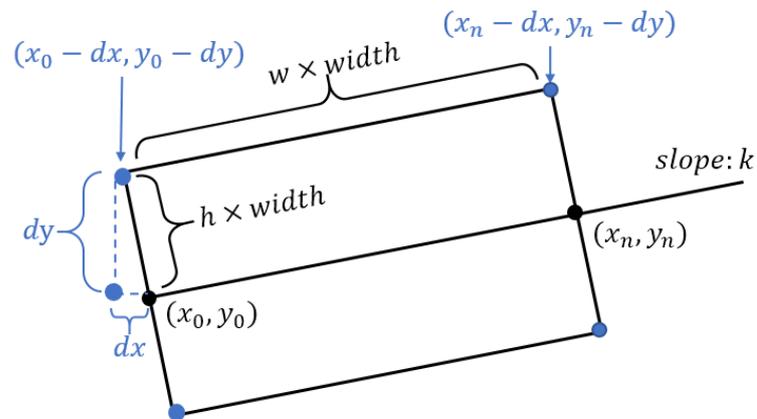
As mentioned earlier, we first compute four coordinates for each sliding window. As shown in Figure 9, the first sliding window is located at the starting point of the power line. Based on analytical geometry knowledge, the coordinates of the four vertices are  $(x_0 - dx, y_0 - dy)$ ,  $(x_n - dx, y_n - dy)$ ,  $(x_0 + dx, y_0 + dy)$ , and  $(x_n - dx, y_n - dy)$  (clockwise). In particular,  $dx$  is computed as  $dx = h \times width \times k \times \sqrt{1/(1+k^2)}$  and  $dy$  is computed as  $dy = h \times width \times \sqrt{1/(1+k^2)}$ , where  $k$  and  $width$  are the slope and width of the power line, respectively. It is worth noting that the two parameters  $w$  and  $h$  could be manually changed to control the width and height of the image patch. Empirically, we set  $w$  and  $h$  to 6 and 24, respectively.

### 3.3.2. Rotation of the Image for Cropping

To crop a power line patch from the original image, a frequently used tool is the OpenCV package. The regular cropping method can only cut an image fragment that is parallel to the vertical and horizontal axes. Thus, we modify the method by first rotating the image so that the power line is parallel to the horizontal axis and then cropping the local power line patch.

The rotation of the image is achieved by a method called “perspective transformation”. Perspective transformation is useful in aligning an image properly. After the application of perspective transformation, the image undergoes a transformation process that rectifies its perspective, resulting in a straightened representation. In our rotation process, we first compute the transformation matrix using the coordinates of the four vertices of the sliding window. The transformation matrix is then used to apply a perspective transformation to the original image, which is then properly rotated for the cropping operation. This rotation

process ensures that the cropping operation is along the power line, covering detailed information and any defects along the power line in the local image cropping.



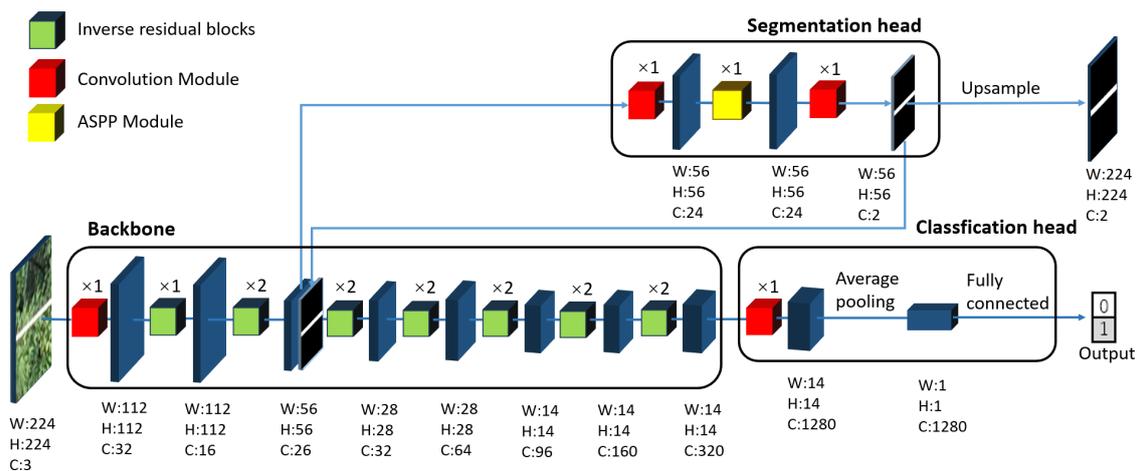
**Figure 9.** Illustration of the key information involved in determining the coordinates of a sliding window.  $(x_0, y_0)$ ,  $(x_n, y_n)$  are the coordinates of the starting point and end point of the current sliding window, respectively;  $k$  and  $width$  are the slope and width of the power line, respectively.  $w$  and  $h$  are adjustable parameters to control the width and height of the image patch. Empirically, we set  $w$  and  $h$  to 6 and 24, respectively.

### 3.4. Patch Classification Network

After obtaining the power line local image patches, the patch classification network performs broken strand recognition in the image patches. Therefore, the ability of the patch classification network directly impacts the precision of the final strand breakage detection performance. Due to the difficulty in obtaining strand breakage samples, we exploit a multi-task learning strategy for the patch classification network, inspired by [31], to make the full use of the limited training data. Specifically, we use a modified MobileNetV2 as the backbone of the classification network and construct an additional segmentation head to achieve multi-task learning along with the classification head of the backbone, as shown in Figure 10. In other words, the proposed patch classification network has a primary head for classification and an auxiliary head for segmentation. The detailed design of the patch classification network is elaborated in the following subsections.

#### 3.4.1. Overall Architecture of the Patch Classification Network

Since the power line image patches generated with our cropping method have high consistency, and the patch classification network only needs to predict whether an input image patch includes a broken strand or not, a relatively small network can be competent at this binary classification task. Furthermore, since tens of image patches can be cropped from a single power line image, the patch classification network should be extremely efficient so that the overall time consumption can be maintained in an acceptable range. Based on the above considerations, we have restructured the original MobileNetV2 to make its network structure more light-weight. This network consists of a backbone network, a segmentation head, and a classification head. The backbone network consists of a total of eight stages. The first stage of the network consists of a convolutional module with a kernel size of 3 and a stride of 2. The second stage is composed of 1 IRB module, while each of the third to eighth stages of the backbone network is composed of 2 stacked IRB modules.



**Figure 10.** The architecture of the patch classification network. It consists of a modified MobileNetV2 backbone network, along with a segmentation head and a classification head. W, H, and C indicate the width, height, and channel number of the feature maps. “×2” means stacking of two same blocks. At the output of the classification head, “0” and “1” corresponds to the prediction of “normal” and “defective”, respectively.

As shown in Figure 10, the segmentation head is connected to the end of the third stage of the backbone and consists of a  $3 \times 3$  convolutional layer, an Atrous Spatial Pyramid Pooling (ASPP) module [32] and a  $1 \times 1$  convolutional layer at the end. The prediction of the segmentation head, which is a double-channel binary segmentation map, is then concatenated back to the output feature map of the third stage of the backbone. The classification head is composed of a  $1 \times 1$  convolutional layer, an average pooling layer, and a fully connected layer. It is worth noting that each convolutional layer used in the segmentation head and the classification head is followed by a batch normalization operation and a ReLU activation.

The benefit of such a design is two-fold: (1) the segmentation head can get supervision from the segmentation task and facilitates the convergence of the shallow layers in the network during training, so that it makes better use of the limited data; (2) the segmentation head can provide richer spatial semantic information for the classification task network.

The classification head receives feature maps from both the backbone and the auxiliary segmentation head, and outputs a binary prediction for each power line image patch to determine whether there exists a broken strand.

### 3.4.2. Multi-Task Loss of the Patch Classification Network

In the training of multi-task networks, it is crucial to address the issue of unifying the losses generated by different tasks. This is particularly important when simultaneously training the classification and segmentation heads, as the auxiliary head can only provide guidance under such circumstances.

During the dataset creation phase, we generated image-level labels (classification labels) and pixel-level labels (segmentation labels) corresponding to both the classification and segmentation tasks. During training, we adopted an end-to-end learning approach to train the classification head and segmentation head. The classification loss and segmentation loss were combined to form a single loss, allowing for end-to-end learning. The definition of the multi-task loss is as follows:

$$L_{total} = \lambda \cdot \gamma_1 \cdot L_{seg} + \gamma_2 \cdot L_{cls} \quad (1)$$

where  $L_{seg}$  and  $L_{cls}$  represent the losses for semantic segmentation and target classification, respectively. Both the segmentation loss function and the classification loss function are computed using cross-entropy loss. Furthermore, the contribution of semantic seg-

mentation and target classification losses to the multi-task loss value is controlled by hyperparameters  $\gamma_1$  and  $\gamma_2$ .  $\lambda$  is defined as a dynamic parameter to adjust the proportion of the segmentation loss in the overall loss value, which is calculated as

$$\lambda = 1 - \frac{n}{n_{ep}} \quad (2)$$

where  $n$  and  $n_{ep}$  indicate the number of the current iteration and total number of iterations. During training,  $\lambda$  is initialized as 1 and linearly decays with the increase in the number of iterations, eventually reaching 0 before the completion of training iterations.

#### 4. Experiments and Results

In order to evaluate the accuracy and efficiency of the proposed strand breakage detector, three parts of the experiment are conducted. Firstly, we evaluate the performance of BA-NetV2 by comparing it with several counterparts. Also, ablation studies are carried out to demonstrate the rationality of the design of BA-NetV2. Secondly, we compare the patch classification network used in the second stage of the detector with a multi-task learning based defect recognition network and some general classification networks, thereby demonstrating the precision of the patch classification network. Subsequently, we compare the overall strand breakage detector with state-of-the-art general object detection networks to ensure the practicality and advancement of the detector. It is worth mentioning that the experiments are conducted indoors on data collected in advance. However, the hardware used for testing the proposed method and existing methods is suitable to be carried by a UAV for power line inspection. Thus, the test results can reflect the applicability of the tested methods to the online power line inspection scenario.

##### 4.1. Performance Evaluation and Ablation Study of BA-NetV2

We conducted an ablation study on the modifications made to upgrade the original BA-Net to BA-NetV2, i.e., the reduction of parallel branches, the increase in the channel numbers, and the utilization of dilated convolution.

Then, to demonstrate the superiority of BA-NetV2, we compare it with several widely used segmentation networks, some of which have been used as base networks for constructing power line segmentation networks in previous research. The segmentation networks used for comparison include HEDnet [33], Fast SCNN [34], U-Net [17], FastFCN [35], DeepLabV3+ [36], and DDRNet [37]. It is worth mentioning that the power line segmentation network proposed in [12] is basically built upon HEDnet; Fast SCNN has been used as the base network in the power line segmentation network proposed in [38]; and U-Net has been used in [39,40] to develop power line segmentation networks.

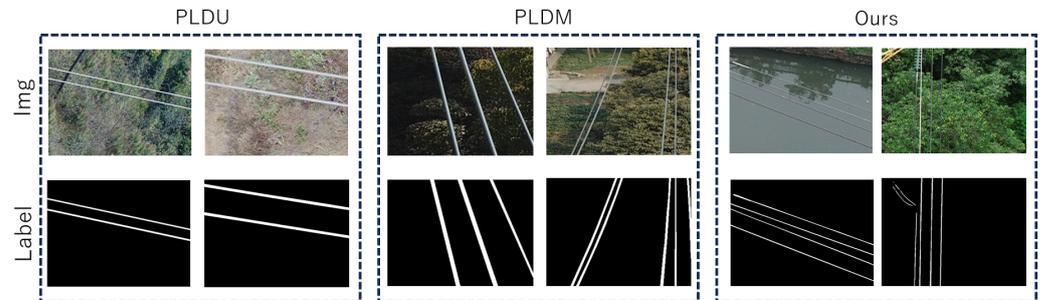
##### 4.1.1. Experimental Settings

**Dataset.** The power line segmentation dataset used in this experiment is a combination of the following two parts:

(1) The relabeled PLDU and PLDM datasets. The PLDU and PLDM datasets created in [12] are open-source power line detection datasets with pixel-wise annotations. PLDU contains 573 power line images of urban scenes, while PLDM contains 287 power line images of mountain scenes. All the images in PLDU and PLDM have the same resolution of  $512 \times 384$ . However, both the PLDU and PLDM datasets are annotated in the boundary detection manner, which is not suitable for our segmentation setting. Thus, we relabeled them in the segmentation manner with power lines set to white pixels and background set to black pixels.

(2) We created an image dataset specifically for power line segmentation tasks, named AIRS-PLS. This dataset consists of 1442 power line images captured under different lighting conditions, backgrounds, and perspectives, exhibiting diverse power line morphologies. The resolution of images in AIRS-PLS varies from  $640 \times 480$  to  $1260 \times 1240$ . For each image in the dataset, high-quality pixel-wise segmentation labels are provided.

In our experimental study, the above datasets are mixed together to form a dataset of 2302 images in total, called Mixed-PLS. The Mixed-PLS dataset is then divided into three disjoint parts of training, validation, and testing sets, with a proportion of about 70%:20%:10%. The training set comprises 1611 images, the validation set contains 460 images, and the testing set consists of 231 images. Some sample images in the dataset are shown in Figure 11.



**Figure 11.** Samples of the Mixed-PLS power line segmentation dataset. This dataset is a combination of three sub-datasets, i.e., the relabeled PLDU dataset and PLDM dataset [12], and the dataset collected by ourselves.

**Evaluation Metrics.** The power line segmentation accuracy of each network in this experiment was evaluated by calculating the Intersection over Union (IoU) of the foreground (power lines) and the ground truth information, as well as the mean IoU (mIoU) that considers both foreground and background in the networks' segmentation output. We also take the inference speed into consideration, which is evaluated by the average number of images processed in a second by each network with the acceleration.

**Implementation Details.** We conducted our experiments using PyTorch. The MM-Segmentation library (<https://github.com/open-mmlab/mmdetection> (accessed on 15 July 2023)) was utilized for implementing the segmentation networks including BA-Net, BA-NetV2, Fast SCNN, U-Net, FastFCN, DeepLabV3+, and DDRNet. As for HEDnet, we used an open-source third-party PyTorch implementation (<https://github.com/meteors-howers/hed> (accessed on 15 July 2023)) of it. All the networks were trained on a server equipped with four NVIDIA RTX 3090 GPUs and tested on an NVIDIA Jetson AGX Orin (32GB version) embedded computer. The hyperparameter settings for training different versions of BA-NetV2 in the ablation study can be found in Table 1. The hyperparameters for training the comparative networks in the comparison experiment are tuned individually. For all the networks, we unified the input image resolution to  $512 \times 512$  and applied the same data augmentation strategy, i.e., horizontal and vertical flips to the input images with a probability of 0.5. The SGD optimizer was used for all the networks during the training.

**Table 1.** The hyperparameter setting for training the BA-NetV2 model.

Hyperparameters	Setting
Initial learning rate	0.001
Minimum learning rate	0.00001
Momentum	0.9
Weight decay	0.0005
Input image size	$512 \times 512$
Batch size	32
Training steps	20,000

#### 4.1.2. Results and Discussion

The experimental results of the ablation study for BA-NetV2 are shown in Table 2. Firstly, reduction of parallel branches enhances both the accuracy and inference speed. This

is probably due to the reduction of the low-resolution branches eliminates their interference to the high-resolution branches and reduced the complexity of the network. Secondly, expansion of the channel number of the network further benefits accuracy of the network by enlarging the capacity of the network, which is necessary for dealing with the diverse backgrounds in different scenes. The additional inference latency brought by the expansion of channel number is limited. Thirdly, the application of dilated convolution branches leads to significant accuracy improvement in the network while keeping the inference time almost unchanged. Moreover, we found that these modifications complement each other when combined in the network, as the network achieved significant improvements in both IoU and mIoU when using them all.

**Table 2.** Results of the ablation study experiment for BA-NetV2. A check mark indicates the modification is applied, while a X brush means the modification is not applied.

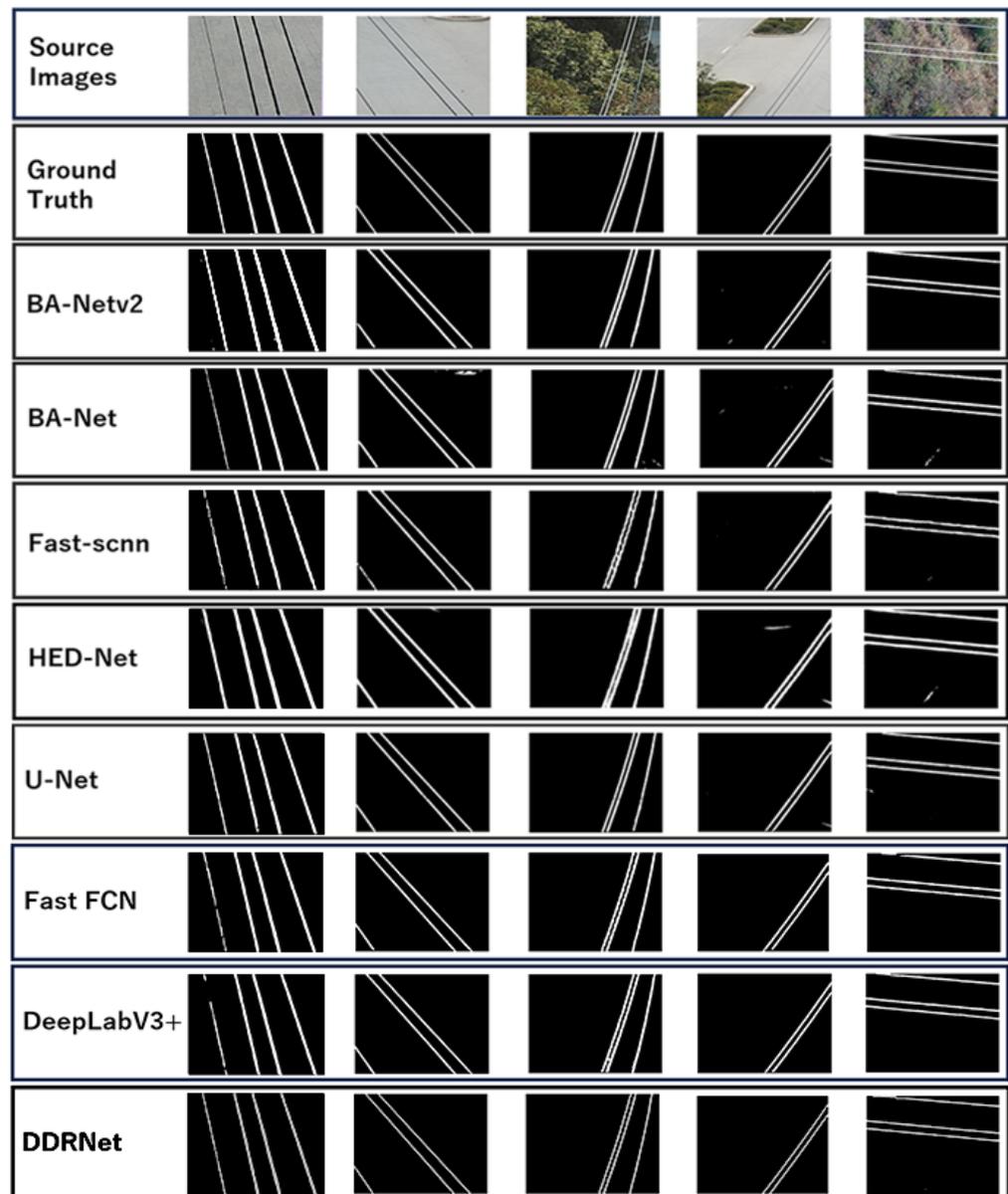
Reduction of Parallel Branches	Expansion of Neural Network Base Channels	Dilated Convolution Branches	IoU (%)	mIoU (%)	Speed (Images/s)
X	X	X	63.6	81.4	30.5
✓	X	X	64.0	81.6	58.3
✓	✓	X	64.6	81.9	48.7
✓	X	✓	64.4	81.8	57.8
✓	✓	✓	66.8	83.0	47.9

Quantitative testing results for different segmentation networks on the power line segmentation task are presented in Table 3. From the table, it is evident that the proposed BA-Net obtains better scores in terms of each accuracy metric compared to all other comparative methods while also the third highest inference speed. These findings indicate that the structural design of the BA-NetV2 enables more effective and efficient extraction and representation of power line features. This is attributed to BA-Net's compact three-branch and multi-scale feature fusion architecture.

**Table 3.** Performances of different methods on the power line segmentation dataset.

Method	IoU (%)	mIoU (%)	Speed (Images/s)
BA-NetV2	66.8	83.0	47.9
BA-Net	63.6	81.4	30.5
Fast SCNN	54.9	76.9	124.0
HEDNet	59.6	78.7	23.0
U-Net	66.0	77.7	9.7
FastFCN	53.7	76.3	11.7
DeepLabV3+	61.0	80.0	44.9
DDRNet	59.3	79.1	51.4

Figure 12 showcases some sample images from the power line segmentation dataset along with the corresponding segmentation results generated by the proposed BA-NetV2 and the compared methods. The results demonstrate that despite the presence of background interference such as linear structure, ground, and plants, BA-NetV2 can generate clearer and more continuous prediction, compared with other segmentation networks.



**Figure 12.** Sample images and segmentation results of BA-NetV2 and the compared networks.

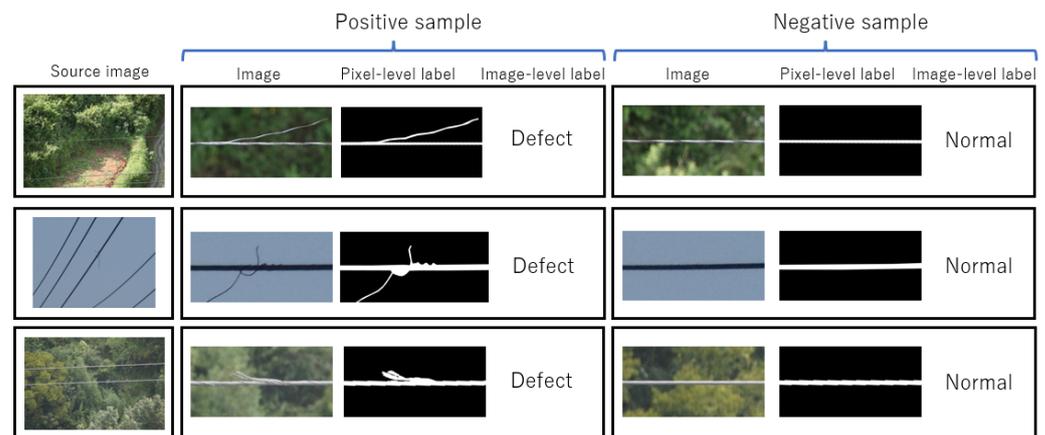
#### 4.2. Performance Evaluation of the Patch Classification Network

In this section, we compare our method with several commonly used general classification models, such as ResNet [41], VGG16 [13], MobileNetV2 [29], and transformer-based methods Swin-Transformer [42] (the light-weight version Swin-Tiny). On the other hand, we also compare it with the defect recognition network proposed in [31] (we name it SegDec) since our patch classification is partially inspired by it.

##### 4.2.1. Experimental Settings

**Dataset.** To validate the effectiveness of the multi-task classification network, we created a power line patch dataset by cropping image patches from our strand breakage dataset, which contains power line images with strand breakage. The power line patch dataset and the strand breakage dataset are named as AIRS-PLIP and AIRS-PLSB, respectively. Specifically, we used BA-NetV2 to segment those power line images of AIRS-PLSB dataset, and cropped out image patches both from the source images and the corresponding segmentation labels, since our patch classification network requires both RGB image patches

and their segmentation labels for multi-task learning. The AIRS-PLSB has 322 power line images, with each image containing at least one broken strand. The resolution of images in AIRS-PLSB varies from  $540 \times 360$  to  $8688 \times 5792$ . It was randomly divided into training, validation and test sets, which contain 222, 50, and 50 images, respectively. The image patches in the training, validation, and test sets of the AIRS-PLIP dataset were cropped from the training, validation and test sets of the AIRS-PLSB dataset, respectively. In order to obtain as many positive samples as possible for the AIRS-PLIP dataset, we set the overlap rate between two adjacent image patches to 0.9 when cropping image patches in the training set of the AIRS-PLSB dataset. The overlap rate was kept 0.2 for cropping image patches from the validation set and the test set of the AIRS-PLSB dataset. As a result, 4205 image patches containing strand breakage were obtained. On the other hand, 9766 image patches without stand breakage were used as negative samples in the training set. (We did not use all the negative samples cropped from the AIRS-PLSB dataset to maintain the balance between positive and negative samples.) Such a dense cropping manner can be regarded as a form of data augmentation. The validation and test sets of the AIRS-PLIP dataset contain 126, and 130 defective image patches, respectively, along with double-size normal image patches. Some samples of the AIRS-PLIP are given in Figure 13.



**Figure 13.** Samples of the AIRS-PLSB dataset and the AIRS-PLIP dataset. The source images of power lines originate from the AIRS-PLSB dataset. The AIRS-PLIP dataset was cropped from the AIRS-PLSB dataset. Each sample of the AIRS-PLIP dataset consists of a RGB image patch, along with its corresponding image-level classification label and pixel-level classification label.

**Evaluation Metrics.** Regarding the performance comparison of the networks, we use three different evaluation metrics to compare the performance of different networks: (a) average precision for positive and negative samples, (b) precision for positive samples, and (c) recall for positive samples. Here, positive samples refer to image patches with broken strand defects, while negative samples refer to images without any defects (normal power lines or background). Additionally, we assess the efficiency of the networks by comparing their inference speed, which is measured by the number of images processed per second by each network.

**Implementation Details.** We implemented the patch classification network, and reimplemented the SegDec network using PyTorch. The TorchVision library (<https://github.com/pytorch/vision> (accessed on 15 July 2023)) was used for training and testing the ResNet-50, VGG16, and MobileNetV2. The Swin-Tiny network was trained and tested with the official implementation (<https://github.com/microsoft/Swin-Transformer> (accessed on 15 July 2023)) of Swin-Transformer. All the models were trained on a server with 8 NVIDIA GeForce RTX 2080TI GPUs and tested on the Jetson AGX Orin embedded computer. Common data augmentation techniques were employed, including random flipping, random scaling, and random cropping.

For the proposed patch classification network, we used the hyperparameters presented in Table 4. We utilized the Adam optimizer to train the involved networks. For the compared methods, the hyperparameters were tuned individually. For all the compared methods, except the SegDec network, pretrained models are used.

**Table 4.** The hyperparameter setting for training the proposed patch classification network and the compared networks.

Hyperparameters	Setting
Initial learning rate	0.001
Minimum learning rate weight decay	0.0001
Input image size	$3 \times 224 \times 224$
Batch size	32
Training steps	9000

#### 4.2.2. Results and Discussion

As shown in Table 5, our model achieves a precision for positive samples that is only 0.01 lower than the highest precision achieved by SegDec. However, compared to SegDec, our model has a much higher average precision by 0.09 and a higher recall rate by 0.03. On the other hand, when compared to ResNet-50, which has the highest average precision, our model still outperforms it by over 0.03 in average precision and has a faster detection speed. It can be observed that our model obtains high scores in all three metrics of precision for positive sample detection, average precision, and recall rate, with recall rate and average precision being the highest among all models. Additionally, the analysis of inference speed further confirms that our proposed model exhibits superior efficiency.

**Table 5.** Performance of different classification methods on AIRS-PLIP dataset.

Method	Precision	Recall	Average Precision	Speed (Images/s)
Ours	0.96	0.90	0.95	257.9
SegDec	0.97	0.87	0.86	45.5
ResNet-50	0.893	0.838	0.914	143.7
VGG16	0.811	0.792	0.863	118.1
MobileNetV2	0.857	0.785	0.855	287.3
Swin-Tiny	0.857	0.877	0.899	125.5

#### 4.3. Overall Performance of the Proposed Strand Breakage Detector

In this section, we evaluate the overall performance of the proposed strand breakage detector, and conduct comparisons between the proposed method and the state-of-the-art object detectors, YOLOv5 [43], YOLOv7 [44], ATSS [45], and EfficientDet [46]. Specifically, we select the smallest model as well as the medium-size model in both YOLOv5 and YOLOv7 series, i.e., the YOLOv5m, YOLOv5s, YOLOv7, and YOLOv7-tiny, as the compared methods. For EfficientDet, we choose the EfficientDet-D3, which is a relatively small model in the EfficientDet series. All the compared models as well as our proposed method were deployed on an NVIDIA Jetson AGX Orin embedded computer to test their inference speed. All the neural networks were converted to ONNX format for on-board inference. Since the embedded computers of the NVIDIA Jetson series are widely used in UAV-based applications, this speed test can help to evaluate the performance of the proposed method and the compared methods in the UAV-based power line inspection scenario.

Considering that strand breakage samples are hard to obtain in real-world scenarios, we further carry out experiments to see the sensitivity of the proposed method to the amount of sample data by reducing the training data by a half.

#### 4.3.1. Experimental Settings

**Dataset.** The power line strand breakage dataset AIRS-PLSB, as mentioned in Section 4.2.1, was used to train and test the proposed strand breakage detector and the compared detectors. The AIRS-PLSB dataset was annotated both pixel-wise for power line segmentation and with bounding box indicating each strand breakage. Since the proposed strand breakage detector is not an end-to-end method, the power line segmentation network and the patch classification network need to be trained separately. The Mixed-PLS dataset and the AIRS-PLIP dataset were also used for training the power line segmentation network BA-NetV2 and the patch segmentation network.

When conducting the experiments for evaluating the models with reduced training data, the image patch classification model of the proposed strand breakage detector was trained with image patches cropped from half of the training set in the AIRS-PLSB dataset. Note that the power line segmentation network was trained with the whole training set of the Mixed-PLS dataset, since it does not contain images from the test set of the AIRS-PLSB dataset. The compared end-to-end object detectors were trained with half of the training set in the AIRS-PLSB dataset.

**Evaluation Metrics.** The strand breakage detection accuracy of the proposed method and the compared methods were evaluated by precision, recall, and F1-score. Since the output form of the proposed method has no confidence score and the bounding boxes have a unified aspect ratio, which is different from those of the compared models, the commonly used evaluation metrics average precision (AP) with confidence or IoU thresholds like AP50, AP75, etc., are not suitable for this experiment. For calculating the precision, recall, and F1-score, we visualized the prediction results and manually counted the correct detections and false detections, and converted them to scores of the three evaluation metrics. The inference speed on the embedded computer of each method was also evaluated by the average number of images processed in a second by each method.

**Implementation Details.** The proposed strand breakage detector was trained part by part. The power line segmentation network BA-NetV2 was firstly trained on the Mixed-PLS dataset, using the hyperparameter setting listed in Table 1. The patch classification network was trained using the AIRS-PLIP dataset, using the setting listed in Table 4. When conducting inference, the local image patches along the segmented power lines were cropped with an overlap rate of 0.2 between two adjacent image patches. Such an overlap setting reduces the redundancy while guaranteeing the full coverage of each power line. It is worth mentioning that we resized the original input image to  $1024 \times 1024$  when its resolution exceeded  $1024 \times 1024$ ; otherwise, the resolution would be unchanged. The compared methods, i.e., YOLOv5, YOLOv7, ATSS, and EfficientDet, were trained in the end-to-end manner with the source image and strand breakage bounding box labels of the training set of the AIRS-PLSB dataset. The input image resolution for all YOLO models was set to  $1024 \times 1024$ . The images in the validation set of the AIRS-PLSB dataset were used for tuning the hyperparameters. The official open-source code of both YOLOv5 (<https://github.com/ultralytics/yolov5> (accessed on 15 July 2023)) and YOLOv7 (<https://github.com/WongKinYiu/yolov7> (accessed on 15 July 2023)) were used in the experiments. The MMDetection library (<https://github.com/open-mmlab/mmdetection> (accessed on 15 July 2023)) was used for implementing ATSS and EfficientDet.

#### 4.3.2. Results and Discussion

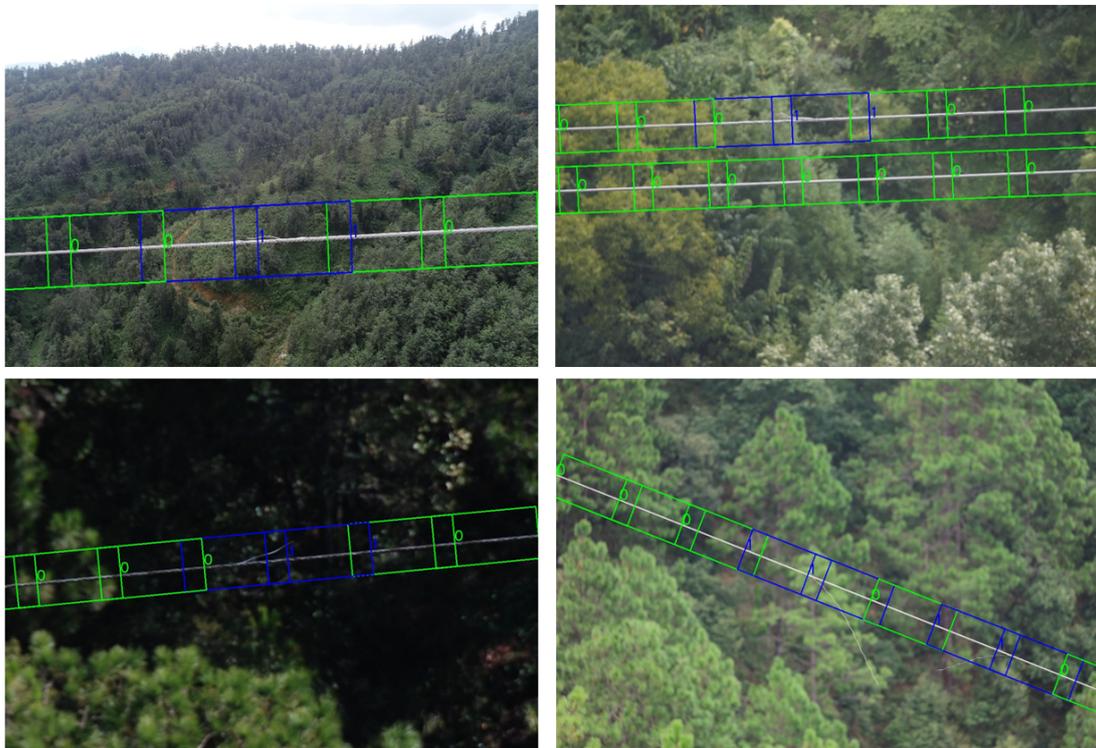
**Detection accuracy.** The quantitative experimental results of the proposed method and the compared methods are listed in Table 6. As can be seen, the proposed method achieved significantly higher scores in all the evaluation metrics related to prediction accuracy. Among the compared methods, YOLOv7 reached the best performance. However, in terms of precision, recall, and F1-score, our method outperformed YOLOv7 by 0.050, 0.077, and 0.065, which are considerable margins. The results demonstrate that our multi-stage pipeline can better make use of the limited broken strand data to learn accurate detection. The sliding window strategy of cropping local image patches also benefits the network in

achieving a high precision score by focusing the patch classification network on the local area of each power line.

**Table 6.** Performance of different detection methods on the test set of the AIRS-PLSB dataset.

Method	Precision	Recall	F1-Score	Speed (Images/s)
Ours	0.833	0.769	0.800	11.5
YOLOv5m	0.769	0.577	0.659	15.4
YOLOv5s	0.775	0.596	0.674	23.4
YOLOv7	0.783	0.692	0.735	9.2
YOLOv7-tiny	0.660	0.596	0.626	24.3
ATSS	0.638	0.712	0.673	9.2
EfficientDet-D3	0.756	0.569	0.667	9.6

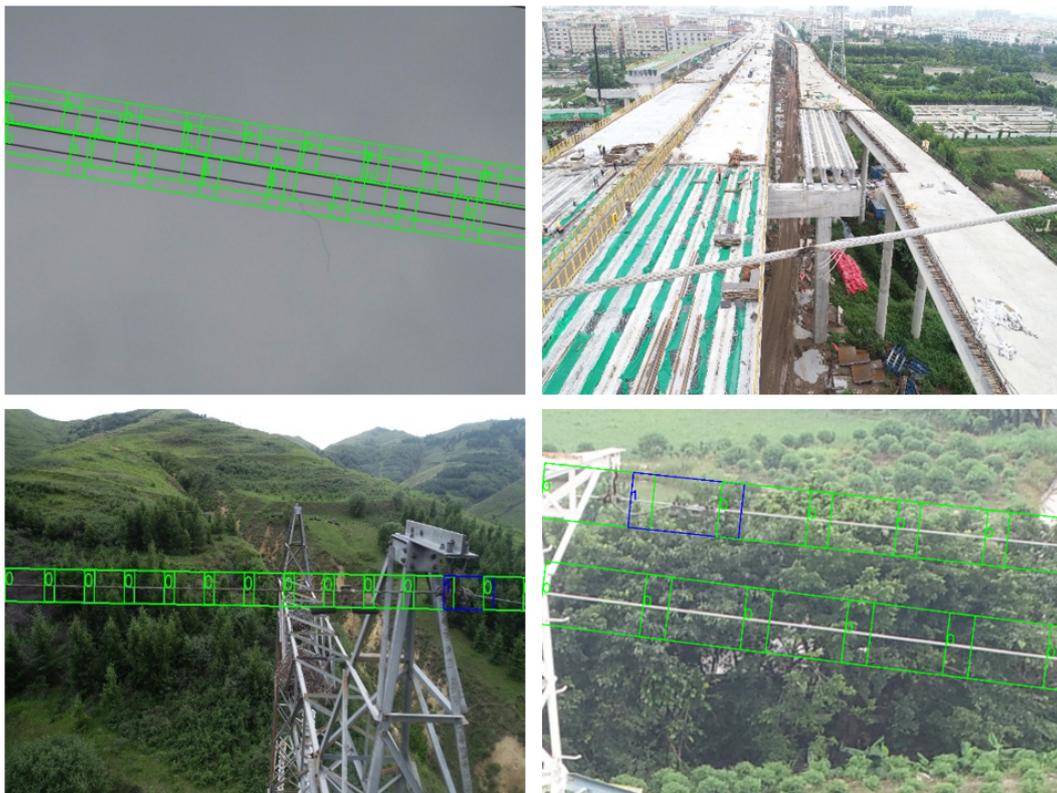
Some visualized prediction results of the proposed methods are shown in Figure 14. In those samples, all the power line image patches generated in the first stage of the proposed method are visualized as colored rectangles. The image patches classified as defective by the patch classification network in the second stage of the proposed method are colored blue, while the image patches classified as normal are colored green. The number 1 or 0 on the right of each patch also indicates the categorical prediction of the proposed method, with 1 for defective and 0 for normal. It can be seen that the image patches are correctly and evenly cropped along each power line with an overlapped area between two adjacent patches. Despite the complex background, varied illumination, and the slender morphological characteristic of broken strands in the images, the proposed method can accurately identify each broken strand.



**Figure 14.** Samples of visualized prediction results of the proposed method. Better viewed in enlarged electronic edition.

Missed detection and false detection are mainly caused by extremely thin broken strands, complex background, and interference of towers and fittings. Figure 15 shows some failure detection cases. The upper left shows the missed detection of an extremely

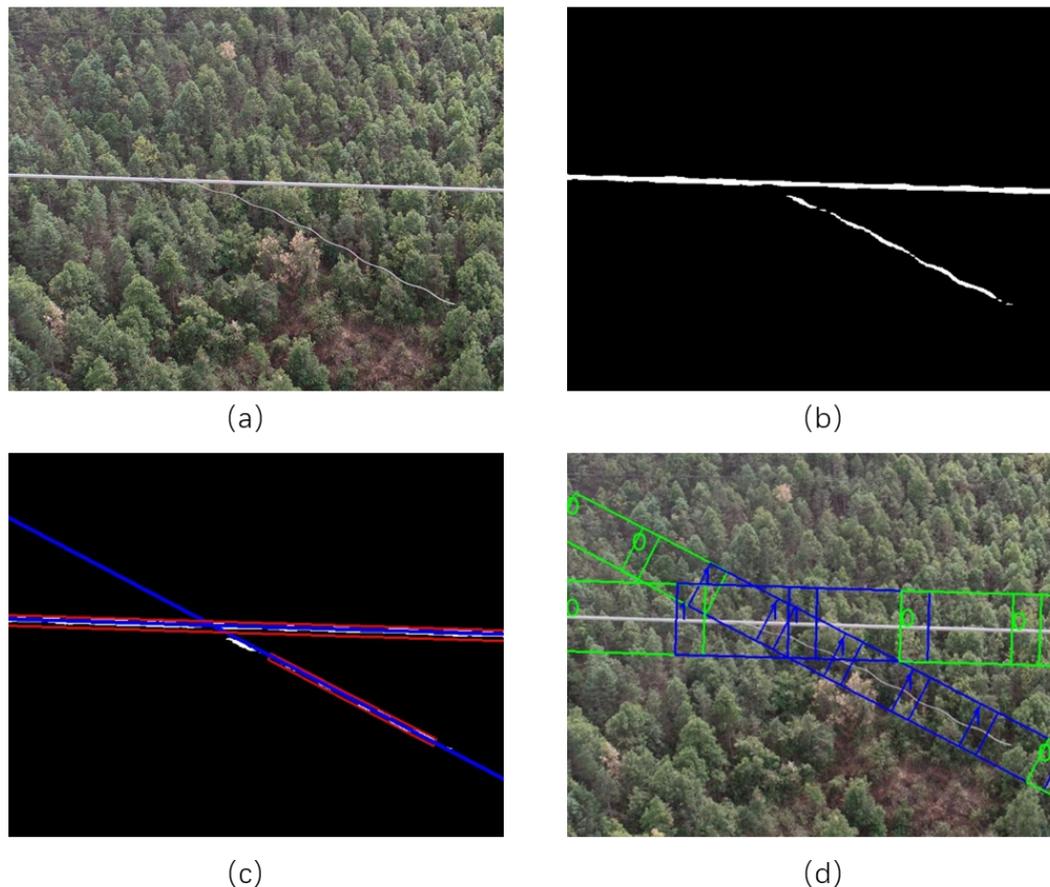
thin strand breakage. This can be improved by further enhancing the resolution of input images but the cost is an increase in inference latency. The upper right shows that, in the extremely complex background, even the power line was not extracted. The lower left and right show the fittings connected to the power lines and the towers causing false detection. In fact, towers have many steel structures that also have linear shape that can interfere with power line segmentation, and the fittings and other components connected to the power lines can affect the image patch classification. To deal with the extremely complex backgrounds and the near tower scenarios, in future work, we will try to collect more data that contain towers and complex backgrounds in urban areas, and further improve the design of our method especially for these challenging scenarios.



**Figure 15.** Examples of missed detection and false detection caused by the extremely thin strand breakage (**upper left**), the extremely complex background (**upper right**), and the fittings connected to the power lines and towers. Better viewed in enlarged electronic edition.

**Detection efficiency.** Due to the multi-stage serial workflow, the inference speed of the proposed method is lower than most of the compared end-to-end detectors. Its speed is about 1/2 of the speed of YOLOv5s or YOLOv5s. According to our test, on the Jetson Orin, the average time consumption of the BA-NetV2 for segmenting a  $512 \times 512$  power line image is 20.9 ms. The patch classification network takes an average of 3.9 ms to process an image patch; the remaining time is consumed in power line fitting and image patch cropping. When there are multiple power lines in an image, the time consumption will increase in the power line fitting, image patch cropping, and patch classification. The inference speed of the proposed method is 11.5 images per second on average, which is still capable of on-site real-time processing in most cases. However, we have to point out that such a speed brings limitations of the speed of UAV flight and image capturing during the inspection process. To further enhance the efficiency of the proposed method in the future, we plan to reimplement the inference stream with C++, improve the parallelism of computation, and conduct compression on the weights of the networks, etc.

It is worth noting that in some cases the broken strand would interfere with the extraction of a power line. Specifically, when the broken strand is thin and long, it may be recognized as a power line and thus lead to image patch cropping along it. This is mainly because such long thin broken strands would be segmented as line segments some of which are long but disconnected to the power line. These long line segments can pass the filtering based on the area and aspect ratio in the postprocessing after power line segmentation and would be misrecognized as a power line. In such a case, it brings redundant computation and risks of generating false positive detection. A typical example is shown in Figure 16.



**Figure 16.** A power line with a thin long broken strand and the processing results of power line segmentation, center line extraction and final strand breakage detection. (a) A sample image with a thin long broken strand snapped from a power line inspection image. (b) Power line segmentation result. (c) Center line extraction result (the blue lines). (d) Final strand breakage detection result.

**Sensitivity to the amount of training data.** Table 7 shows the performance of different classification methods trained with half of the training set in the AIRS-PLIP dataset. It can be seen that, when the training set is reduced by a half, the proposed method has a relatively slight drop in its precision while the recall remains unchanged and finally it has a significantly slighter drop in the F1-score compared to the other detectors. The superiority of the proposed method is further enlarged in terms of the F1-score when the training set is reduced. The F1-scores of the proposed method are 0.065 and 0.168 higher than the best values in the compared methods when trained with the full training set and the half training set, respectively. Therefore, the proposed method shows less sensitivity to the decrease in defect samples, which supports our claim that the proposed strand breakage detection pipeline reduces the need for strand breakage samples.

**Table 7.** Performance of different classification methods trained with half of the training set in the AIRS-PLIP dataset and the performance drop compared with full data training. “Full” indicates the models were trained with the full training set in the AIRS-PLSB dataset; “Half” indicates the models were trained with the half training set in the AIRS-PLSB dataset; “Drop” indicates the performance decrease from “Full” to “Half”. The best performance in each column is highlighted in bold.

Method	Precision			Recall			F1-Score		
	Full	Half	Drop	Full	Half	Drop	Full	Half	Drop
Ours	<b>0.833</b>	<b>0.784</b>	0.049	<b>0.769</b>	<b>0.769</b>	<b>0</b>	<b>0.800</b>	<b>0.777</b>	<b>0.023</b>
YOLOv5m	0.769	0.722	<b>0.047</b>	0.577	0.500	0.077	0.659	0.591	0.068
YOLOv5s	0.775	0.630	0.145	0.596	0.558	0.038	0.674	0.592	0.082
YOLOv7	0.783	0.730	0.053	0.692	0.519	0.173	0.735	0.607	0.128
YOLOv7-tiny	0.660	0.542	0.118	0.596	0.500	0.096	0.626	0.520	0.106
ATSS	0.638	0.579	0.059	0.712	0.423	0.289	0.673	0.489	0.184
EfficientDet-D3	0.756	0.700	0.056	0.596	0.538	0.058	0.667	0.609	0.058

## 5. Conclusions

This paper proposes a real-time broken strand detection method oriented to the UAV-based power line inspection scenario. A multi-stage pipeline is devised, consisting of power line segmentation, image patch cropping, and patch classification. Such a pipeline can make better use of easily obtained normal power line images and the detailed feature information in the local areas of power lines, thus to deal with the challenges caused by the slender morphology of power lines and rareness of strand breakage samples. The key components in the pipeline, i.e., the segmentation network and the patch classification network, are both designed to be light-weight; thus, the overall pipeline is suitable for executing real-time processing on the edge computing resource carried by UAVs. Experimental results show that: (1) The proposed strand breakage method can achieve superior accuracy over state-of-the-art object detection methods and real-time processing on embedded edge computing device. (2) By maintaining high-resolution feature maps, enlarging the network capacity, and enhancing the long-range semantic relation capturing ability, the proposed power line segmentation network BA-NetV2 is better adapted to the elongated feature of power lines and outperforms its counterparts. (3) The patch classification network can reach high accuracy benefiting from the multi-task learning strategy benefits. The proposed strand breakage method provides a promising solution for UAV-based on-site power line defect detection.

Our future works include three aspects: (1) To further enhance the inference efficiency of the proposed method, we plan to reimplement the code in C++, improve the parallelism of computation, and apply model compression techniques. (2) To deal with the extremely complex backgrounds and the near tower scenarios, we will try to collect more data containing towers and complex backgrounds in urban areas, and further improve the design of the proposed method especially for these challenging scenarios. (3) We plan to deploy the proposed strand breakage detection method onto a power line inspection hardware system that we are developing and conduct real-world experiments. We will improve the proposed method based on the experimental results and promote its application in practical power line inspection.

**Author Contributions:** Conceptualization, N.L., X.X., S.W. and N.D.; methodology, J.Y., X.Z., S.S., X.H. and N.L.; software, J.Y., X.Z., S.S. and X.H.; validation, X.X. and N.L.; formal analysis, X.X. and N.L.; investigation, Y.Y. and S.W.; resources, N.D. and N.L.; data curation, S.W., Y.Y., J.Y. and X.Z.; writing—original draft preparation, J.Y., X.Z., S.S. and N.L.; writing—review and editing, J.Y., X.X. and N.L.; supervision, N.D.; project administration, N.D. and N.L.; funding acquisition, N.D. and N.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Key R&D Program of China (2021YFE0204200), Guangdong Basic and Applied Basic Research Foundation (2021A1515110700), Shenzhen Science and Technology Program (JSGG20210802154539015), and funding from Shenzhen Institute of Artificial Intelligence and Robotics for Society.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Our code and part of the data used in this research are available at <https://github.com/AIRS-CSR/StrandBreakageDetection> (accessed on 20 June 2023). (Due to confidentiality regulations, we are unable to make all the data publicly available.)

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

## References

1. Yang, L.; Fan, J.; Liu, Y.; Li, E.; Peng, J.; Liang, Z. A review on state-of-the-art power line inspection techniques. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9350–9365. [[CrossRef](#)]
2. Luo, Y.; Yu, X.; Yang, D.; Zhou, B. A survey of intelligent transmission line inspection based on unmanned aerial vehicle. *Artif. Intell. Rev.* **2023**, *56*, 173–201. [[CrossRef](#)]
3. Liu, M.; Li, Z.; Li, Y.; Liu, Y. A fast and accurate method of power line intelligent inspection based on edge computing. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 3506512. [[CrossRef](#)]
4. Siddiqui, Z.A.; Park, U. A drone based transmission line components inspection system with deep learning technique. *Energies* **2020**, *13*, 3348. [[CrossRef](#)]
5. Chen, Y.; Li, Y.; Zhang, H.; Tong, L.; Cao, Y.; Xue, Z. Automatic power line extraction from high resolution remote sensing imagery based on an improved radon transform. *Pattern Recognit.* **2007**, *49*, 174–186. [[CrossRef](#)]
6. Zhou, G.; Yuan, J.; Yen, I.L.; Bastani, F. Robust real-time UAV based power line detection and tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016.
7. Du, S.; Tu, C. Power line inspection using segment measurement based on HT butterfly. In Proceedings of the 2011 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), Xi'an, China, 25–27 October 2011; pp. 1–4.
8. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)]
9. Alom, M.Z.; Taha, T.M.; Yakopcic, C.; Westberg, S.; Sidike, P.; Nasrin, M.S.; Hasan, M.; Van Essen, B.C.; Awwal, A.A.S.; Asari, V.K. A state-of-the-art survey on deep learning theory and architectures. *Electronics* **2019**, *8*, 292. [[CrossRef](#)]
10. Chang, W.; Yang, G.; Li, E.; Liang, Z. Toward a cluttered environment for learning-based multi-scale overhead ground wire recognition. *Neural Process. Lett.* **2018**, *48*, 1789–1800. [[CrossRef](#)]
11. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A.A. Image-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1125–1134.
12. Zhang, H.; Yang, W.; Yu, H.; Zhang, H.; Xia, G.-S. Detecting power lines in UAV images with convolutional features and structured constraints. *Remote Sens.* **2019**, *11*, 1342. [[CrossRef](#)]
13. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
14. Choi, J.; Lee, S.J. Weakly Supervised Learning for Transmission Line Detection Using Unpaired Image-to-Image Translation. *Remote Sens.* **2019**, *14*, 3421. [[CrossRef](#)]
15. He, M.; Qin, L.; Deng, X.; Zhou, S.; Liu, H.; Liu, K. Transmission Line Segmentation Solutions for UAV Aerial Photography Based on Improved UNet. *Drones* **2023**, *7*, 274. [[CrossRef](#)]
16. Han, K.; Wang, Y.; Tian, Q.; Guo, J.; Xu, C.; Xu, C. Ghostnet: More features from cheap operations. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 1580–1589.
17. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
18. Ishino, R.; Tsutsumi, F. Detection system of damaged cables using video obtained from an aerial inspection of transmission lines. In Proceedings of the IEEE Power Engineering Society General Meeting, Denver, CO, USA, 17–21 July 2004; pp. 1857–1862.
19. Mao, T.; Ren, L.; Yuan, F.; Li, C.; Zhang, L.; Zhang, M.; Chen, Y. Defect recognition method based on HOG and SVM for drone inspection images of power transmission line. In Proceedings of the 2019 International Conference on High Performance Big Data and Intelligent Systems, Shenzhen, China, 5–15 April 2019.
20. Jalil, B.; Leone, G.R.; Martinelli, M.; Moroni, D.; Pascali, M.A.; Berton, A. Fault Detection in Power Equipment via an Unmanned Aerial System Using Multi Modal Data. *Sensors* **2019**, *19*, 3014. [[CrossRef](#)] [[PubMed](#)]
21. Sampedro, C.; Martinez, C.; Chauhan, A.; Campoy, P. A supervised approach to electric tower detection and classification for power line inspection. In Proceedings of the 2014 International Joint Conference on Neural Networks (IJCNN), Beijing, China, 6–11 July 2014.
22. Liu, Y.; Yong, J.; Liu, L.; Zhao, J.; Li, Z. The method of insulator recognition based on deep learning. In Proceedings of the 2016 4th International Conference on Applied Robotics for the Power Industry (CARPI), Jinan, China, 11–13 October 2016; pp. 1–5.

23. Wang, M.; Tong, W.; Liu, S. Fault Detection for Power Line Based on Convolution Neural Network. In Proceedings of the 2017 International Conference on Deep Learning Technologies (ICDLT '17), Association for Computing Machinery, Chengdu, China, 2–4 June 2017.
24. Xu, H.; Pu, Z.; Yv, J.; Bai, J. Recognition algorithm for broken stocks in power grid based on attention mechanism and sliding window detection. *China Energy Environ. Prot.* **2021**, *43*, 211–215.
25. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [[CrossRef](#)]
26. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
27. Yang, X.; Yang, J.; Yan, J.; Zhang, Y.; Zhang, T.; Guo, Z.; Fu, K. Srdet: Towards more robust detection for small, cluttered and rotated objects. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 8232–8241.
28. Li, N.; Chen, Z.; Zhang, X.; Liu, X. An ultra-fast bi-phase advanced network for segmenting crop plants from dense weeds. *Biosyst. Eng.* **2021**, *212*, 160–174. [[CrossRef](#)]
29. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 4510–4520.
30. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018.
31. Božič, J.; Tabernik, D.; Skočaj, D. Mixed supervision for surface-defect detection: From weakly to fully supervised learning. *Comput. Ind.* **2021**, *129*, 103459. [[CrossRef](#)]
32. Che, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848.
33. Xie, S.; Tu, Z. Holistically-nested edge detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1395–1403.
34. Poudel, R.P.K.; Liwicki, S.; Cipolla, R. Fast-SCNN: Fast Semantic Segmentation Network. *arXiv* **2019**, arXiv:1902.04502.
35. Wu, H.; Zhang, J.; Huang, K.; Liang, K.; Yu, Y. FastFCN: Rethinking dilated convolution in the backbone for semantic segmentation. *arXiv* **2019**, arXiv:1903.11816.
36. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
37. Pan, H.; Hong, Y.; Sun, W.; Jia, Y. Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 3448–3460. [[CrossRef](#)]
38. Zhu, K.; Xu, C.; Wei, Y.; Cai, G. Fast-PLDN: Fast power line detection network. *J. Real-Time Image Process.* **2022**, *19*, 3–13. [[CrossRef](#)]
39. Lee, S.J.; Yun, J.P.; Choi, H.; Kwon, W.; Koo, G.; Kim, S.W. Weakly supervised learning with convolutional neural networks for power line localization. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
40. Choi, H.; Yun, J.P.; Kim, B.J.; Jang, H.; Kim, S.W. Attention-based multimodal image feature fusion module for transmission line detection. *IEEE Trans. Ind. Inform.* **2022**, *18*, 7686–7695. [[CrossRef](#)]
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 11–17 October 2021; pp. 9992–10002.
43. Jocher, G. YOLOv5 by Ultralytics. 2020. Available online: <https://github.com/ultralytics/yolov5> (accessed on 13 July 2023).
44. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023.
45. Zhang, S.; Chi, C.; Yao, Y.; Lei, Z.; Li, S.Z. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9756–9765.
46. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and efficient object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10781–10790.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.