applied sciences

MDPI

Article

Transmission Tower Re-Identification Algorithm Based on Machine Vision

Lei Chen, Zuowei Yang, Fengyun Huang *, Yiwei Dai, Rui Liu and Jiajia Li

School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China; chen_lei_jd@whut.edu.cn (L.C.); yangzuowei@whut.edu.cn (Z.Y.); wvsdcc@whut.edu.cn (Y.D.); 283256@whut.edu.cn (R.L.); jarjar@whut.edu.cn (J.L.)

* Correspondence: hawkfly@whut.edu.cn

Featured Application: This work can be potentially applied to the recognition of traffic signs in intelligent driving vehicles and automatic inspection of power systems.

Abstract: Transmission tower re-identification refers to the recognition of the location and identity of transmission towers, facilitating the rapid localization of transmission towers during power system inspection. Although there are established methods for the defect detection of transmission towers and accessories (such as crossarms and insulators), there is a lack of automated methods for transmission tower identity matching. This paper proposes an identity-matching method for transmission towers that integrates machine vision and deep learning. Initially, the method requires the creation of a template library. Firstly, the YOLOv8 object detection algorithm is employed to extract the transmission tower images, which are then mapped into a d-dimensional feature vector through a matching network. During the training process of the matching network, a strategy for the online generation of triplet samples is introduced. Secondly, a template library is built upon these d-dimensional feature vectors, which forms the basis of transmission tower re-identification. Subsequently, our method re-identifies the input images. Firstly, we propose that the YOLOv5n-conv head detects and crops the transmission towers in images. Secondly, images without transmission towers are skipped; for those with transmission towers, The matching network maps transmission tower instances into feature vectors. Ultimately, transmission tower re-identification is realized by comparing feature vectors with those in the template library using Euclidean distance. Concurrently, it can be combined with GPS information to narrow down the comparison range. Experiments show that the YOLOv5n-conv head model achieved a mean Average Precision at an Intersection Over Union threshold of 0.5 (mAP@0.5) score of 0.974 in transmission tower detection, reducing the detection speed by 2.4 ms compared to the original YOLOv5n. Integrating the online triplet sample generation into the matching network training with Inception-ResNet-v1 (d = 128) as the backbone enhanced the network's rank-1 performance by 3.86%.

Keywords: transmission tower re-identification; transmission tower detection; YOLO; triplet loss

Citation: Chen, L.; Yang, Z.; Huang, F.; Dai, Y.; Liu, R.; Li, J. Transmission Tower Re-Identification Algorithm Based on Machine Vision. *Appl. Sci.* **2024**, *14*, 539. https://doi.org/10.3390/app14020539

Academic Editor: João M. F. Rodrigues

Received: 10 November 2023
Revised: 28 December 2023
Accepted: 3 January 2024
Published: 8 January 2024

1. Introduction

Transmission towers are used to support power transmission lines. Due to prolonged exposure to outdoor environments, power systems are susceptible to various natural risks such as floods, snowstorms, and heatwaves, as well as physical defects like missing insulator pieces, loss of damper heads, and infrastructure damage [1]. All these factors can affect the normal operation of power transmission lines, leading to disruptions in power transmission, economic losses, and even threats to public safety. In recent years, with the rapid advancement of machine learning and deep learning, these technologies are increasingly being employed to predict and detect faults in power grids [2,3]. Atrigna et al. [4] propose an approach based on machine learning techniques to predict power grid

outages caused by heatwaves. Typically, power grids commonly utilize risk assessments to mitigate the detrimental effects of natural disasters like heatwaves and floods, thereby reducing potential damage. Simultaneously, due to the prolonged exposure of power systems to outdoor environments, various physical defects can emerge. Such defects can lead to interruptions in power transmission. The literature [5] utilizes variations in voltage and current to localize defects within the power transmission lines, but it is possible that these defects have not yet manifested as changes in the electrical parameters of the power transmission lines. If the grid system could detect these defects promptly, it would enable preemptive interventions before any deviations in voltage and current become evident, thereby reinforcing the stability and reliability of the power transmission lines. Thus, it is essential to detect and replace the defective parts. Typically, the power grid conducts inspections of its power transmission lines to detect faults [6]. Currently, the methods for inspections include manual inspection, helicopter patrols, and Unmanned Aerial Vehicle (UAV) inspections [7]. The inspections of transmission towers and their accessories primarily include: foreign object detection [8,9], the detection of insulator and other infrastructural damage [10,11], and checks for transmission tower tilting and damage [12]. When defects located in transmission towers and their accessories are detected during inspection, the power system should implement maintenance measures immediately. However, prior to maintenance measures, it is imperative to first determine the identity and location of the transmission tower where the damage is situated. Transmission tower re-identification refers to recognizing the transmission tower's location and specific identification, which is pivotal for their timely maintenance. Transmission tower re-identification needs to achieve object detection and identity matching for transmission towers.

The primary task of transmission tower re-identification is the target recognition of the transmission towers. Traditional object detection algorithms tend to rely on hand-crafted features, such as Sobel edge detection, Haar, and HOG features. These features exhibit limited generalization capabilities, resulting in a subpar performance in complex scenarios. Object detection algorithms based on deep learning fall into two categories. One is two-stage detection algorithms reliant on region proposals, such as Fast R-CNN [13] and Faster R-CNN [14]. The other is based on regression analysis, exemplified by the YOLO [15–17] series and EfficientDet [18]. In the context of transmission tower detection, Zhang et al. [19] introduced DSA-YOLOv3 to enhance detection performance in aerial images of transmission towers. Experiments showed that DSA-YOLOv3 achieved a higher average precision (AP) than both YOLOv3 and the two-stage Fast-RCNN. Bian et al. [20] optimized Faster R-CNN by reducing its convolutional layers, catering to the speed and accuracy requirements for transmission tower detection on mobile devices. Sheng et al. [21] took into account the compositional relationship of transmission tower components and enhanced YOLOX for defects detection of the towers and their components. Post-modification, there was a 10.13% improvement in mean average precision (mAP) compared to the original YOLOX. Additionally, algorithms from the YOLO series, such as YOLOv5 and YOLOv7, have been applied in the detection of electrical system equipment and in defect detection [22,23].

Currently, there are three main methods for transmission tower identity matching, that is, manual comparison, nameplate recognition and precise positioning using position and orientation system (POS).

For manual comparison, the identification of transmission towers primarily depends on inspection staff using information from the transmission tower nameplates and their environmental surroundings. Nevertheless, this methodology is characterized by considerable labor intensity and suboptimal efficiency in the inspection processes.

For transmission tower identity matching, the recognition of the nameplate represents a prevalent approach. Kong et al. [24] detected the transmission tower nameplate using Faster R-CNN and subsequently proposed, in her undergraduate thesis, an adversarial-learning-based Super-Resolution Feature Generation Network (SFGNet) for enhancing the accuracy of small object detection. Xia et al. [25] applied affine distortion correction to digital text images and recognized numbers on the transmission tower nameplates using

the CRNN algorithm. Li et al. [26] constructed the F3RNet for the recognition of electrical equipment, attaining an average precision (AP) of 90.1% in the detection of transmission tower nameplates. In practice, relying solely on the transmission tower nameplate for transmission tower re-identification can introduce an array of challenges. For instance, the placement of transmission tower nameplate varies across different transmission towers. In the photographic process targeting the transmission towers, the transmission tower nameplate may reside in the camera's occluded region, becoming imperceptible (Figure 1a); Additionally, the transmission tower nameplate might have fallen off or been damaged due to harsh weather conditions like snowstorms (Figure 1b,c); Due to prolonged outdoor exposure, the transmission tower nameplate has become illegible (see Figure 1d).

Figure 1. Challenges in nameplate identification: (**a**) the nameplate is situated in an occluded zone; (**b**) nameplate detachment; (**c**) nameplate damage; (**d**) nameplate indistinctness.

Precise positioning methods employing the Position and Orientation System (POS) require the use of high-accuracy GPS data. Wang et al. [27] chose to use the Position and Orientation System (POS) to address the transmission line tower matching. Qin et al. [28] utilized POS to acquire the coordinates of the scene's point cloud and the precise movement trajectory of the cable inspection robot. Typically, to obtain high-accuracy GPS data, it is necessary to deploy Differential GPS (DGPS), which enhances the accuracy of standard GPS using ground-based reference stations. When transmission towers are located in remote areas, the system needs to establish multiple stations for comprehensive coverage. Additionally, during the inspection process, the GPS receiver is typically integrated into the inspection device, and it is necessary to deploy sensors such as Inertial Measurement Units (IMUs) and magnetometers to deduce the GPS positions of objects captured in the imagery. Consequently, the deployment of the POS incurs significant expenditures and poses considerable challenges in terms of maintenance. Furthermore, GPS signals can be affected or even lost in dense urban areas, tunnels, mountainous regions, and high-humidity weather conditions.

The advantages and disadvantages of the three methods are delineated in Table 1. In recent years, researchers have detected and recognized transmission tower nameplates for matching towers, while others have utilized POS for the positioning of transmission towers and their components. Their research is summarized in Table 2.

Table 1. The advantages and disadvantages of the three methods.

Reference	Method Category	Advantages	Disadvantages
-	Manual comparison	High reliability	High labor intensity and low work efficiency
[24–26]	Nameplate detection and recognition	High precision	The nameplate may fall off and become invisible
[27,28]	POS	High precision and real-time positioning	High setup and maintenance costs; positioning failure under weak GPS signals.

Table 2. Literature review of nameplate and POS applications in power transmission line inspection.

Reference	Dataset	Task	Utilized Item	Result
[24]	Their own	Nameplate detection	Nameplate	AP: 73.2%
[25]	Their own	Nameplate recognition	Nameplate	Accuracy: 96.4%
[26]	Their own	Nameplate detection	Nameplate	AP: 90.1%
[27]	-	Positioning transmission line tower	POS	Positioning accuracy within 5 m
[28]	-	Point cloud positioning	POS	Build up the cable inspection robot motion trajectory model

Image matching techniques have been widely adopted in various subdisciplines. In the realm of facial recognition, matching algorithms are designed to precisely identify specific individuals from extensive facial image databases. For pedestrian re-identification, matching algorithms strive to accurately recognize the same individual from different camera perspectives and at different time intervals. Moreover, in landmark recognition, matching algorithms have proven their value, adeptly identifying specific structures or locations from a plethora of scene images, significantly underpinning tourism landmark identification and autonomous driving technologies.

At present, the features employed in image matching algorithms primarily encompass local and global characteristics. Local feature matching predominantly relies on feature extraction and descriptor matching. These techniques initially extract salient point or line features from images, compute descriptors for these features, and then match images pairs based on descriptor similarity. Traditional local image feature matching approaches include ORB [29] and LBD [30], among others. ORB integrates FAST key point detection with BRIEF descriptors, incorporates rotational invariance, conserves computational resources, and is frequently employed in visual localization and map reconstruction. LBD is a local line feature matching algorithm that accumulates edge responses to generate surrounding regions of line segments and formulates a one-dimensional descriptor, and is commonly utilized for architectural and urban landscape image matching. Bian et al. [20] enhanced the ORB point matching and LBD line matching algorithms to address the pose estimation of UAV relative to transmission towers. Guo et al. [31] refined the ORB algorithm to extract uniform key points and, in conjunction with the LSD [32] algorithm, matched regions of transmission towers and backgrounds in images to estimate the UAV pose. Deep-learning-based local feature matching algorithms encompass methods like SuperPoint [33] and SuperGlue [34]. SuperPoint, grounded on convolutional neural networks, is designed for key point detection and descriptor generation. SuperPoint is typically employed in image registration and 3D reconstruction tasks, utilizing Euclidean distance and cosine similarity as matching assessment metrics. SuperGlue, a graph neural-network-based matching method, necessitates key points and descriptors as inputs. Through its graph neural network and optimized matching layer, SuperGlue facilitates inter-image matching and is prevalently used in visual SLAM and robotic navigation tasks.

Local feature matching techniques can achieve image local feature matching, subsequently enabling the tracking of dynamic object movement trajectories and pose estimation. However, the task of transmission tower identity matching is to distinguish between different individual transmission towers. In practice, multiple similar transmission towers may

exist within a small vicinity. When matching the identity of transmission towers, certain features that are not successfully matched might be the critical indicators differentiating transmission towers' identities. Consequently, utilizing local feature matching methods may result in subpar matching outcomes.

Global image features represent the overall content of an image. Convolutional Neural Networks (CNNs) progressively extract and integrate local features into a global context through their hierarchically increasing receptive fields and stacked structures, capturing both the fine details and overall structure of images. In recent years, due to their significant feature extraction capabilities, numerous global feature matching algorithms have begun to utilize CNN to extract global image features. Facenet [35] employs a CNN-based architecture and introduces a triplet loss training mechanism for facial matching tasks. Chen et al. [36] employ Convolutional Neural Networks for feature extraction and append a verification sub-network based on a classification sub-network to address the pedestrian re-identification problem. Arcface [37] uses CNN for input image feature extraction, considers angles in the feature space as classification boundaries, and incorporates the Arcface loss for model training, further enhancing the accuracy of face classification. The networks mentioned previously utilize CNNs to extract global image features, and corresponding experiments have demonstrated the reliability of their global feature extraction methods.

The reviewed studies offer vital perspectives on the identification of transmission towers

1. Defect inspection in electrical power systems, which is vital, tends to favor the use of single-stage identification algorithms from the YOLO series. These algorithms not only excel in terms of accuracy but also demonstrate a remarkable recognition speed, meeting the practical demands and standards of electrical inspections;
2. The identification matching of transmission towers is crucial, facilitating the maintenance of the towers and their components. Manual comparison methods are time-consuming and can potentially be influenced by human factors; nameplate recognition and POS positioning represent effective and mature methods for matching transmission towers in the automated inspections. However, nameplate recognition is incapable of handling situations where images lack nameplates. Additionally, POS have high installation and maintenance costs, and their matching effectiveness diminishes when GPS data are unavailable.
3. Local feature matching algorithms have limitations, as models tend to match similar features but often overlook unmatched critical features;
4. Convolutional Neural Networks (CNNs) can extract and integrate global image features. When combined with different head networks, they have been applied in pedestrian re-identification and face matching.

Transmission towers are typically located in vast open areas and surrounded by the same type of transmission towers, which exhibit similar external shapes. In the task of transmission tower re-identification, it is possible to encounter multiple instances of the same type of transmission towers within a single image. Consequently, the spatial relationship of key points among these towers might be identical. Compared to transmission tower re-identification, facial recognition exhibits a more pronounced difference for distinct faces. Additionally, pedestrian re-identification can rely on some characteristics of different individuals for identification, such as facial features and the relative positioning of human body key points [38,39]. Landmarks, due to their specific geographical locations, often come with discernible reference objects, making their identification process relatively stable. Therefore, transmission tower re-identification is a challenging issue. Considering the limitations of nameplate recognition and POS, this paper proposes a system framework for transmission tower re-identification, which contains two parts. These are the construction of the transmission tower identity matching template library and the identity matching stage (the identity matching of the transmission tower in the input images). Both parts require the implementation of transmission tower detection and the generation of feature vectors. Given our aspiration to deploy this algorithm in edge devices such as UAVs, we prioritize accuracy during the establishment of the template library. During the identity matching

stage, we intend to comprehensively consider both speed and accuracy. Consequently, optimizations are made for the transmission tower detection and matching networks. Within the framework of our newly proposed transmission tower re-identification system, the principal contributions are enumerated as follows:

1. During the identity matching stage, we propose using the YOLOv5-conv head network to detect transmission towers. While maintaining detection accuracy, the speed of transmission tower detection is enhanced;
2. During the training of the transmission tower matching network, we introduce an online triplet sample generation strategy. During the training process, we fix the anchor and positive samples in the triplet and employ the Hungarian algorithm to optimize the selection of negative samples in all triplets. The online triplet generation for triplet sampling strengthened model convergence stability, accelerated the convergence speed, and improved the rank-1 accuracy of transmission tower identity matching.
3. We propose a method to establish a transmission tower identity matching template library. The matching template library for transmission towers is constructed based on the feature vectors generated from transmission tower images. Additionally, GPS information can be included in the database, which can be neglected in the absence of GPS signals.
4. Our proposed method does not rely on GPS information. During the process of matching transmission towers, on the one hand, if the input image is equipped with GPS information, the matching accuracy and speed can be improved by narrowing the template library. On the other hand, the method is still capable of performing transmission tower identity matching even in the absence of GPS information in the input images.

2. The Proposed Method

Transmission tower re-identification, aiming to distinguish between different transmission tower instances, is addressed in this paper based on a template library algorithm. This method encompasses two stages: firstly, establishing a transmission tower identity matching template library, and secondly, performing identity matching of transmission towers in input images. Key technologies in both stages include a transmission tower detection network and a matching network.

2.1. The Framework of the Proposed Method

The simplified system framework is shown in Figure 2. Overall, three sets of models need to be trained, which are the YOLOv8s (transmission tower detection), YOLOv5n-conv head (transmission tower detection), and the matching network (transmission tower matching). In Figure 2, 'Backbone: Inception-ResNet-v1 (d = 128)' indicates that our chosen backbone network is the Inception-ResNet-v1 (d = 128). Sections 2.4 and 2.5 consider more complexities beyond those depicted in Figure 2, such as instances where the images lack GPS information. "The construction of the transmission tower identity matching template library" will be elaborated in detail in Section 2.4. The "transmission tower identity matching framework" will be elaborated in detail in Section 2.5.

The first part is the construction of the transmission tower matching template library. The specific steps are as follows:

1. The YOLOv8s target detection network is used to crop the transmission tower from the images;
2. The matching network, utilizing Inception-ResNet-v1 (d = 128) as its backbone, is employed to obtain the feature vector of the cropped transmission tower image;
3. Positional information (GPS) is added to the transmission tower image; this item can be set to "null" in the absence of GPS;
4. Based on the above information, a transmission tower identity matching template library is established.

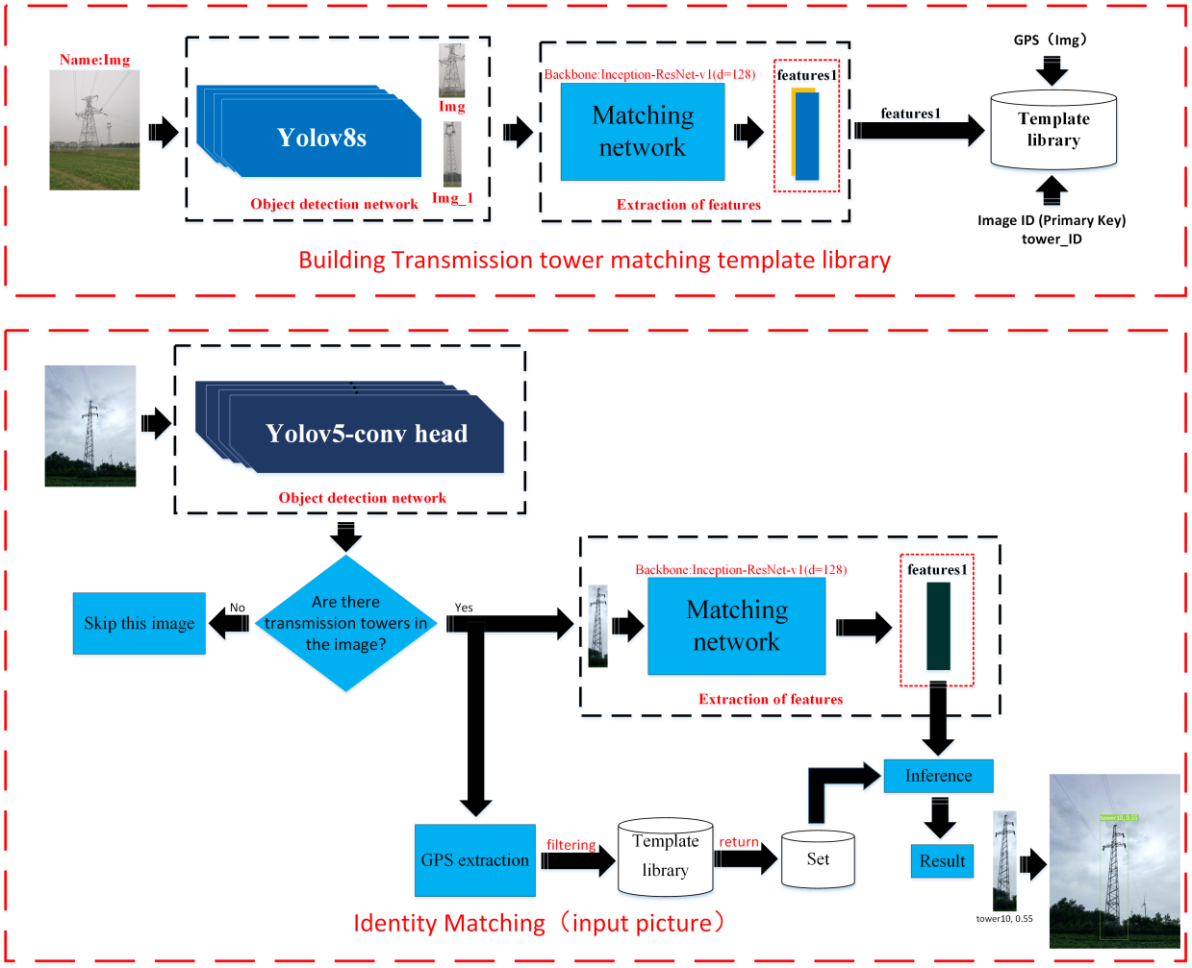

Figure 2. Framework of the transmission tower identity matching system.

The second part is the identity matching of the transmission tower in the input images (the identity matching stage), detailed as follows:

1. The YOLOv5n-conv head network is employed to identify whether there is a transmission tower in the input images. If the input image contains transmission towers, the transmission towers are cropped from the image and processed through steps 2 and 3. If there is no tower, the image is skipped;
2. The cropped transmission tower image is processed through the matching network, which employs Inception-ResNet-v1 (d = 128) as its backbone, to obtain the feature vector for matching;
3. If the image designated for matching contains GPS data, images are filtered from the template library based on the GPS latitude and longitude. The feature vector awaiting matching is then matched with the images in this collection. If the original image lacks GPS information, the feature vector for matching is compared with all images in the template library for identity matching.

Since our matching network is based on metric learning, it can adapt to new data without the need for frequent retraining [35]. We plan to train the matching network annually. In practical applications, the initial step entails the creation of a template library for matching transmission towers. This library is then employed in the identity matching stage when inspecting transmission lines. During the identity matching stage, if transmission towers are present in the input image, the identity matching algorithm is applied to align these towers with the corresponding entries in the template library. In Figure 2, the processed input image yields 'tower10, 0.549', where '0.549' denotes the minimum distance to the images in the template library. 'tower10' indicates that the cropped input image is

matched to the image instance with the identifier 'tower10' in the database, signifying that the model has matched this cropped image to the tower10 instance. Moreover, when GPS information is accessible within the input images, it is considered for integration into the template library. This integration may involve replacing certain tower images to augment the accuracy of the matching process.

In Section 2, this paper elaborates on the methods and models we utilized, while Section 3 is dedicated to the experiments and analyses conducted on these models and methods. The key technical points of this paper are illustrated in Figure 3.

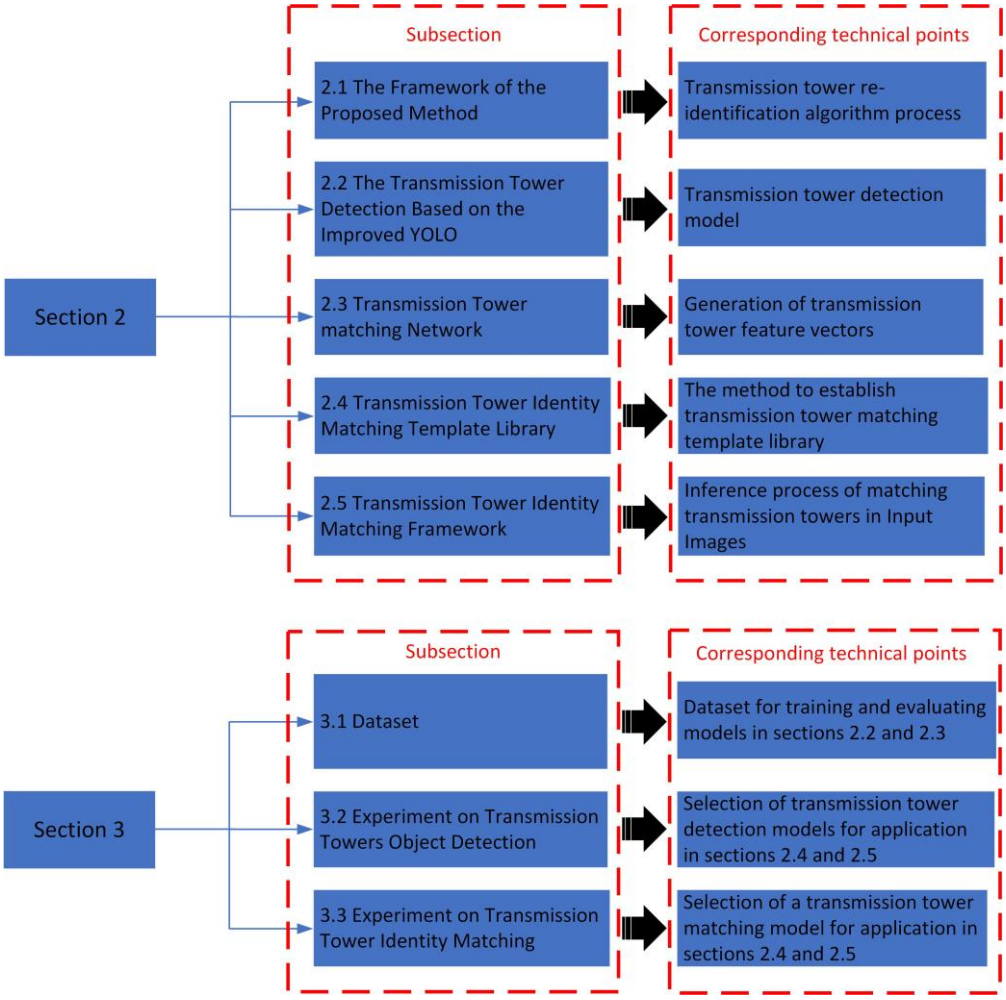

Figure 3. The key technical points in this paper.

2.2. The Transmission Tower Detection Based on the Improved YOLO

The YOLO series, as a classic single-stage detection model, has spawned numerous object detection algorithms [40]. To date, the YOLO series has evolved up to YOLOv8. Zhao et al. [22] indicated that YOLOv7, when applied to anti-vibration hammer corrosion detection, not only offered rapid detection speed but also a higher accuracy, even surpassing two-stage detection models like Faster R-CNN. In their study on insulator defect detection tasks, Souza et al. [10] revealed that YOLOv5l achieves a mAP@0.5 detection accuracy surpassing that of both YOLOv6l and YOLOv7. Currently, YOLOv8's detection accuracy on the COCO dataset surpasses that of YOLOv7 and the YOLOv5 [17]. Given the characteristics of the YOLO network and its applications in the transmission tower defect detection, this study utilizes the YOLOv8 network in establishing the transmission tower identity matching template library. To further accelerate the inference speed of the transmission tower object detection, the improved YOLOv5n-conv head network is employed during the identity matching stage.

2.2.1. Improvement of YOLOv5n

YOLOv5 [15], a detection model introduced by Ultralytics in 2020, has evolved from its initial release of version 1.0 to version 7.0. In the YOLOv5 architecture, CBS consists of Convolution (Conv), BatchNormlization (BN), Sigmoid Linear Unit (SiLU). The CSP Bottleneck with three convolutions (C3) module is utilized in the backbone and the head networks. Within the backbone network, this module is denoted as C3_n, where 'n' represents the number of Bottleneck1 units. In contrast, in the head network, it is denoted as C3_n_F, with 'n' indicating the number of Bottleneck2 units. This differentiation underscores the adaptive use of the CSP Bottleneck structure in various segments of the YOLOv5 model to optimize performance. The backbone feature extraction network of YOLOv5 consists of CBS, C3_n, Spatial Pyramid Pooling—Fast (SPPF). The head network of YOLOv5 consists of CBS, upsampling, C3_n_F and Conv, constructing three sub-detection modules of varying dimensions. The loss function of YOLOv5 combines categorical probability cross-entropy loss, binary cross-entropy loss for confidence, and bounding box regression loss, aiming for a comprehensive optimization of the target detection model. The categorical probability loss adjusts the target category of the classification prediction box, the confidence loss determines whether the prediction box genuinely contains the target object, and the bounding box regression loss finetunes the exact position of the prediction box. The total loss is determined by the cumulative weighted sum of these three losses, as illustrated in Equation (1):

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{box} \quad (1)$$

where L_{cls} represents classification loss, L_{obj} represents confidence loss, L_{box} represents bounding box regression loss, λ_1 is the weight of the classification loss, λ_2 is the weight of the confidence loss, λ_3 is the weight of the bounding box regression loss.

This study introduces two derivative models based on the YOLOv5n architecture: YOLOv5n-c2f head and YOLOv5n-conv head. Glenn Jocher et al. [17] suggested that the CSP Bottleneck with two convolutions (C2f) can introduce gradient flow to enhance the feature extraction capability of the head network. In the YOLOv8 architecture, the structure of the C2f module differs between the backbone and head networks. Within the backbone network, this module is denoted as C2f_n, where 'n' represents the number of Bottleneck1 units. Conversely, in the head network, it is identified as C2f_n_F. Here, 'n' indicates the number of Bottleneck2 units. In our research, the YOLOv5n underwent a structural modification where the original C3_n_F module was replaced with the C2f_n_F module. This alteration resulted in the creation of a new variant, designated as the YOLOv5n-c2f head. Furthermore, we replaced the C3_n_F module in YOLOv5n with a two-layer CBS network, resulting in the formation of the YOLOv5n-conv head. All the previously discussed modules, as well as the refined YOLOv5n-conv head network, are depicted in Figure 4. This substitution aims to streamline the network, decrease the inference time for the transmission tower detection, and facilitate real-time monitoring on mobile devices. The overall network structure of YOLOv5n-conv head is illustrated in Figure 4. Since our alterations were limited to specific layers within the head network of YOLOv5n, the loss calculation for both YOLOv5n-c2f head and YOLOv5n-conv head remains as outlined in Equation (1).

2.2.2. YOLOv8

YOLOv8 [17], an improved iteration based on the network structure of YOLOv5, is the latest model introduced by Ultralytics in 2023. Firstly, when processing image data, the initial convolution kernel size is adjusted to 3×3. Secondly, YOLOv8 improves network performance by incorporating the C2f module, which encompasses the C2f_n_F and C2f_n configurations. This replaces the C3 module used in YOLOv5, consisting of the C3_n_F and C3_n. Thirdly, YOLOv8 adopts the decoupled head (anchor-free head). The decoupled head approach may accelerate training speed and enhance model accuracy. YOLOv8 network architecture and its modules are illustrated in Figure 4. However, on the flip side, it may encounter issues with regression task misalignment. Hence, the Distributed

Focal Loss (DFL), an enhanced version of the focal loss, is employed as a loss function for regressing the distance from the points to the bounding boxes. Consequently, the loss function of YOLOv8 comprises classification loss, regression loss, and DFL loss. Thus, the total loss for YOLOv8 is delineated as depicted in Equation (2):

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{box} + \lambda_3 L_{dfl} \quad (2)$$

where L_{cls} represents classification loss, L_{box} represents bounding box regression loss, L_{dfl} represents DFL loss, λ_1 is the weight of the classification loss, λ_2 is the weight of the bounding box regression loss, λ_3 is the weight of the DFL loss.

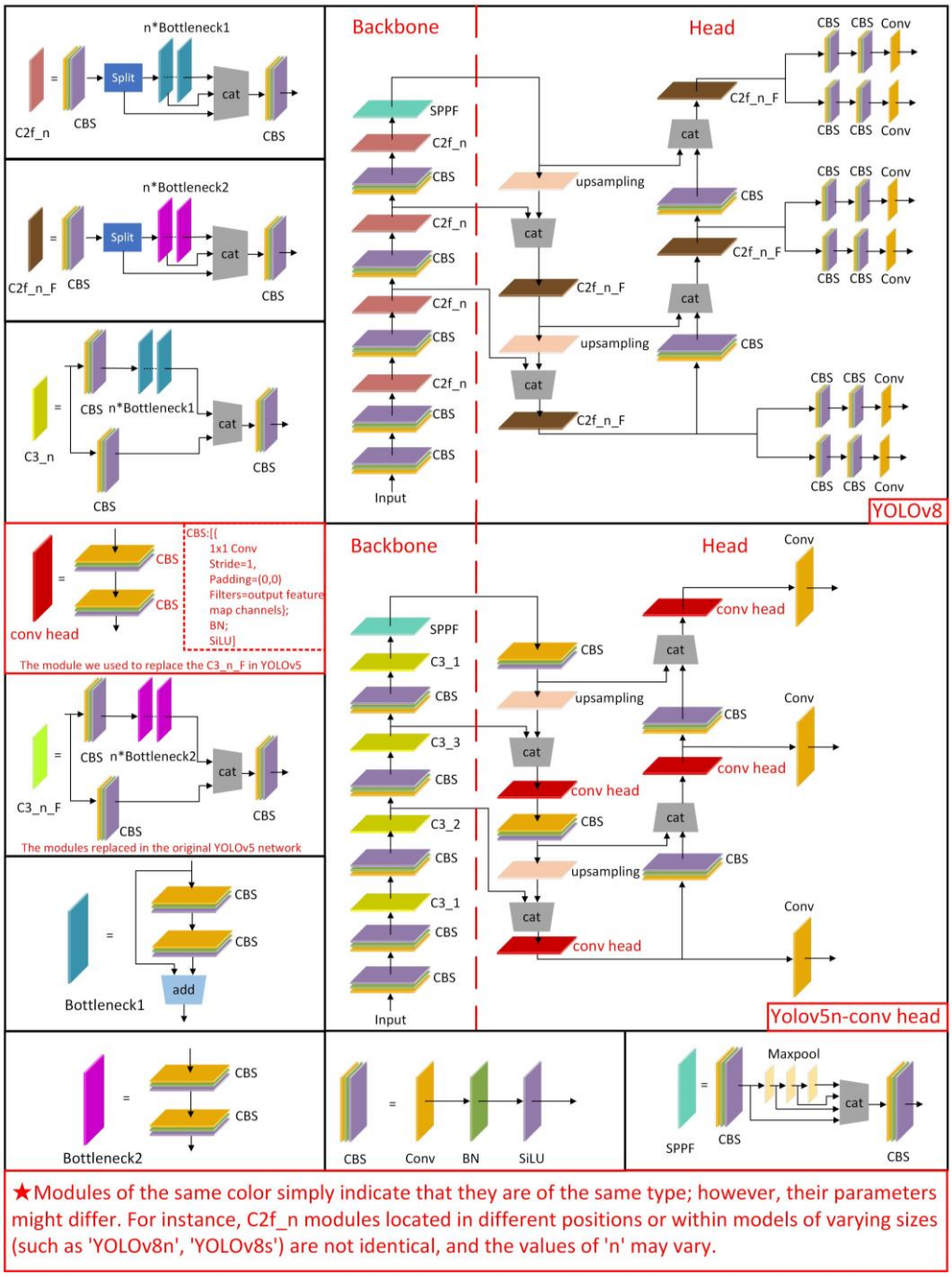

Figure 4. YOLOv5n-conv head and YOLOv8 architecture.

2.2.3. Transmission Tower Detection Model Evaluation

Object detection models are typically evaluated based on their inference time and accuracy. In this study, we chose mean Average Precision (mAP) to assess the accuracy of transmission tower detection. The mAP is derived from the average AP of all categories, indicating the overall detection performance of the model. The calculation formula for mAP is presented by Equation (6):

$$P = \frac{TP}{TP + FP} \tag{3}$$

$$R = \frac{TP}{TP + FN} \tag{4}$$

$$AP = \int_0^1 P(R)dR \tag{5}$$

$$mAP = \sum_{i=1}^{N} \frac{AP_i}{N} \tag{6}$$

where TP represents the number of true positives in the category, FP denotes the number of false positives in the category, FN is the number of false negatives in the category, AP_i represents the Average Precision for the ith category, N is the total number of categories, i represents the ith category, mAP represents the mean Average Precision.

For object detection, IoU is the intersection ratio of two bounding boxes. In this paper, $mAP@IoU$ is a comprehensive metric that considers the precision and recall of all detection categories under a given IoU value. Before calculating $mAP@IoU$, it is necessary to first compute the $AP@IoU$. $AP@IoU$ represents the area under the PR curve formed by precision (P) and recall (R) under a given IoU value, reflecting the accuracy of a single category.

2.3. Transmission Tower matching Network

In this Section, we provide a thorough explanation of the matching network's structure. The network is divided into training and inference stages for application. Key technologies in the training phase include image augmentation, the triplet generation strategy, and construction of the loss function for the network.

2.3.1. Matching Network Architecture

The matching network framework in this paper is based on the architecture of the FaceNet [35]. When two images are inputted, the matching network maps them into feature vectors, as illustrated in Figure 5. Each image, upon processing through the backbone, yields a feature map. The feature map undergoes average pooling (AVG) and L2 normalization (L2) to obtain a k-dimensional feature vector. 'k' represents the number of channels in the feature map obtained after the image passes through backbone. This k-dimensional feature vector is mapped to a d-dimensional feature vector (features1) through the fully connected layer FC1, and then mapped to an m-dimensional feature vector (features2) via the fully connected layer FC2. In this paper, we implemented three distinct networks as alternatives for the backbone of the matching network. These networks are MobileNet [41], MobileViT [42], and Inception-ResNet-v1 [43]. MobileNet (k = 1024) is a convolutional neural network devoid of residual structures. MobileViT (k = 320) is founded on the Vision Transformer architecture, and Inception-ResNet-v1 (k = 1792) is a convolutional neural network equipped with residual structures.

During the matching process, the network directly maps the input image to features1, facilitating subsequent identity matching for the transmission tower. Our approach to train the matching network involved the utilization of triplet loss, in conjunction with cross-entropy loss. Furthermore, we introduced a method for optimizing the generation

sequence of triplets to enhance the training efficiency of the network. Detailed explanations of these training techniques are comprehensively outlined in Section 2.3.5.

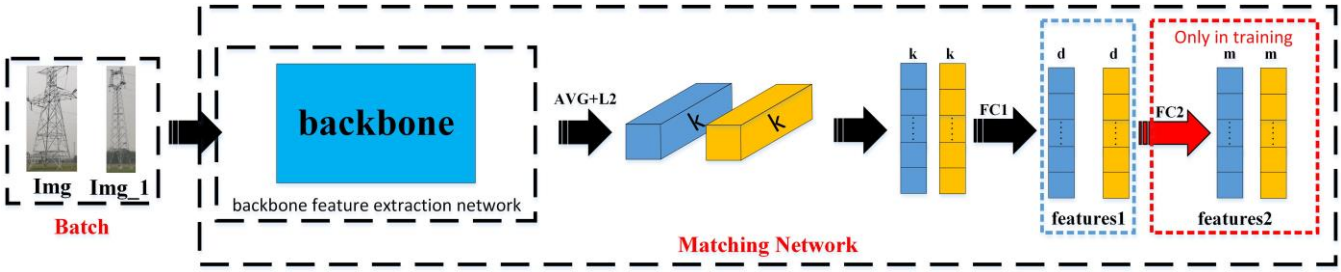

Figure 5. Framework of the matching network.

2.3.2. Training Dataset Image Augmentation

Images of transmission towers that are captured might originate from various collection points, under diverse lighting and weather conditions. Owing to these extrinsic factors, the color, brightness, contrast, and other visual characteristics of the images may be affected to varying extents. While specific environmental backgrounds might momentarily assist in identifying the transmission towers, considering the variability of the external environment, models should refrain from an over-reliance on such information. Consequently, this study introduces an image augmentation technique involving the random bottom occlusion of transmission towers, combined with enhancements in the HSV channel and random image flipping, to enrich the dataset.

By employing random horizontal flipping augmentation on images, this technique can effectively enlarge the transmission tower matching dataset, thereby augmenting its diversity and robustness in deep learning applications. The image augmentation technique in the Hue, Saturation, Value (HSV) channel can adjust hue, saturation, and value to enhance visual effects. This method aims to simulate varying weather conditions, thereby enabling the model to adapt to diverse meteorological scenarios. The HSV enhancement values chosen in this study were referenced from the YOLOv8's augmentation values, with gain coefficients of {0.015, 0.7, 0.4}. While local environments can facilitate the matching of images with the template library, it is pertinent to note that transmission towers are situated outdoors, where environmental conditions might vary. In practice, the vicinity of the base of transmission towers may undergo substantial alterations. To address this, this study implements a random bottom occlusion strategy for image augmentation. The height of the introduced random white block, denoted H_a^z, falls within the range $\left[H_a^6, H_a^4\right]$. Concurrently, its width, W_a^z lies in the range $\left[W_a^8, W_a^2\right]$. When the block is superimposed onto the image at coordinates $\left(H_p, W_p\right)$, H_p is confined to the range $\left[H - H_a^4, H - H_a^z\right]$; W_p is restricted to $(0, W - W_a^z]$. This augmentation is strategically implemented to ensure that the images are adaptive to localized environmental changes. The formulas for calculating the width and height of the added white block are respectively presented in Equations (7) and (8).

$$H_a^z = int\left(\frac{H}{z}\right) \tag{7}$$

$$W_a^z = int\left(\frac{W}{z}\right) \tag{8}$$

where H and W denote the height and width of the image, respectively. z represents the enhancement coefficient. Post enhancement, the height and width of the required white block are expressed as H_a^z and W_a^z.

The results obtained from applying the three image augmentation techniques are depicted in Figure 6.

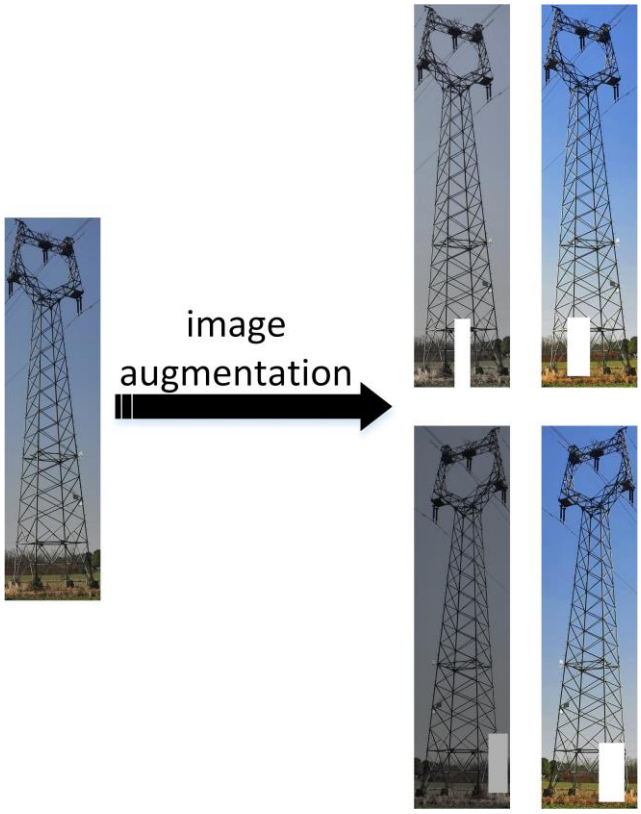

Figure 6. Results after applying the image augmentation.

2.3.3. Triplets Generation Strategy Based on Dynamic Negative Allocation

Triplet loss [35] is a loss function utilized for similarity metric learning. It aims to map images of the same identity in Euclidean space to proximal positions. Simultaneously, it maps images of the different identity to more distant positions in the space. When the network embeds image x into a d-dimensional Euclidean space, the embedding is represented as $f(x) \in R^d$. The objective of the triplet loss function is to reduce the distance between images of the same identity (anchor x_i^a and positive x_i^p), while enlarging the distance from images of a different identity (negative x_i^n). Consequently, the model can adeptly cluster features of transmission towers with identical identities. The formula for calculating the triplet loss is presented as Equation (9):

$$L_{\text{triplet}} = \sum_{i=1}^{N} \left[\| f(x_i^a) - f\left(x_i^p\right) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + a \right]_+ \tag{9}$$

where L_{triplet} represents triplet loss. N represents the number of triplets. x_i^a, x_i^p, x_i^n respectively represent the anchor, positive, and negative, in the *ith* triplet. $f\left(x_i^a\right)$, $f\left(x_i^p\right)$ and $f\left(x_i^n\right)$, respectively, represent the feature vectors after x_i^a, x_i^p and x_i^n have been processed through the network. α represents the enforced margin between positive and negative sample pairs. $[\,]_+$ indicates that when the value $[\,]$ is greater than 0, it is taken as the loss; when it is less than or equal to 0, the loss is considered as 0.

However, simple triplets can lead to a triplet loss of 0, which is not conducive for model training [44]. For transmission tower matching, we think that examples of the same type of transmission tower are more challenging to distinguish than those of different types. Therefore, we believe it is necessary to dynamically adjust triplet samples. Based on the literature [45], this paper introduces an online triplet generation method that fixes the anchor and positive within a batch, optimizing the maximum triplet loss, and employs the Hungarian algorithm to allocate the negatives in the batch. The generation process of

triplets is shown in Figure 7. During the training process, the batchsize is set as a multiple of 3. The batchsize is divided into multiple items (s_1, s_2, . . ., s_m), with the total number of items being one-third of the batchsize. Each item (s_i) consists of an anchor (x_i^a), a positive (x_i^p), and a negative (x_i^n). Following the application of the Hungarian algorithm, the positions of the negative samples in each batch were dynamically optimized, leading to the creation of new items (i_1, i_2, . . ., i_m).

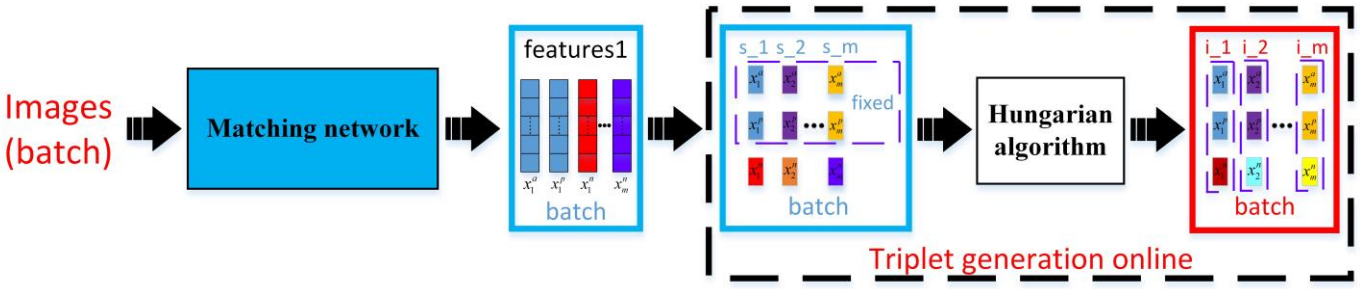

Figure 7. Flowchart of the new triplet generation process.

During the training process, the Hungarian algorithm is employed to allocate the negative samples from all items in the batch. To facilitate optimization, we ensured that the identities of the transmission tower in different items were distinct. Therefore, the maximum batch size can be set to half the total number of transmission tower instances. This strategy seeks to minimize the scoring metric S between the anchor and its corresponding positive relative to the unassigned negative for each batch. The optimization target S is computed as Equation (14). Utilizing the Hungarian algorithm minimizes the value of S, yielding the current allocation matrix $x_{i,j}$ and completing the distribution of negative samples among the items. Each row and column of $x_{i,j}$ contains only one '1', ensuring a unique matching scheme.

$$value_i^j = \left\| f(x_i^a) - f(x_i^p) \right\|_2^2 - \left\| f(x_i^a) - f\left(x_j^n\right) \right\|_2^2 + a \tag{10}$$

$$score_i^j = \begin{cases} 0 & value_i^j \le 0 \\ value_i^j & 0 < value_i^j \end{cases} \tag{11}$$

$$s_{i,j} = -score_i^j, i, j = 1, 2, \ldots m, m \le q/2 \tag{12}$$

$$x_{i,j} = \begin{cases} 1 & Assignment\ from\ jth\ \ negative\ to\ ith\ anchor \\ 0 & others \end{cases} \tag{13}$$

$$S = \sum_{i=1}^{m} \sum_{j=1}^{m} x_{i,j} s_{i,j} \tag{14}$$

where $value_i^j$ represents the triplet value computed from the anchor and positive in the ith item and the negative in the jth item; $score_i^j$ represents the triplet score for the anchor and positive in the ith item and the negative in the jth item; $s_{i,j}$ represents the element in the ith column of the allocation matrix, and s is a m × m matrix. Each row represents the score of the anchor and positive in the ith item corresponding to the negative in the jth item; m represents the total number of items in the batch; q represents the total number of transmission tower instances in dataset; $x_{i,j}$ represents the allocation matrix; S represents the objective function to be optimized.

Under the assumption that feature vectors of the same type of transmission towers are closer compared to those of different types, and that these distances ensure non-zero triplet loss, we developed a triplet mining strategy. These newly formulated triplets, in accordance with our hypothesis, are delineated in Figure 8. In Figure 8, each transmission tower instance is labeled as tower 'i', indicating its sequence as the ith instance (tower_ID).

Detailed information about its data type, and other relevant details are provided in Table 3. Table 3's structure will be detailed in Section 2.4. Additionally, item '*i*' refers to the *ith* triplet sample. Within an item, 'A' represents the anchor, 'P' denotes the positive, and 'N' signifies the negative.

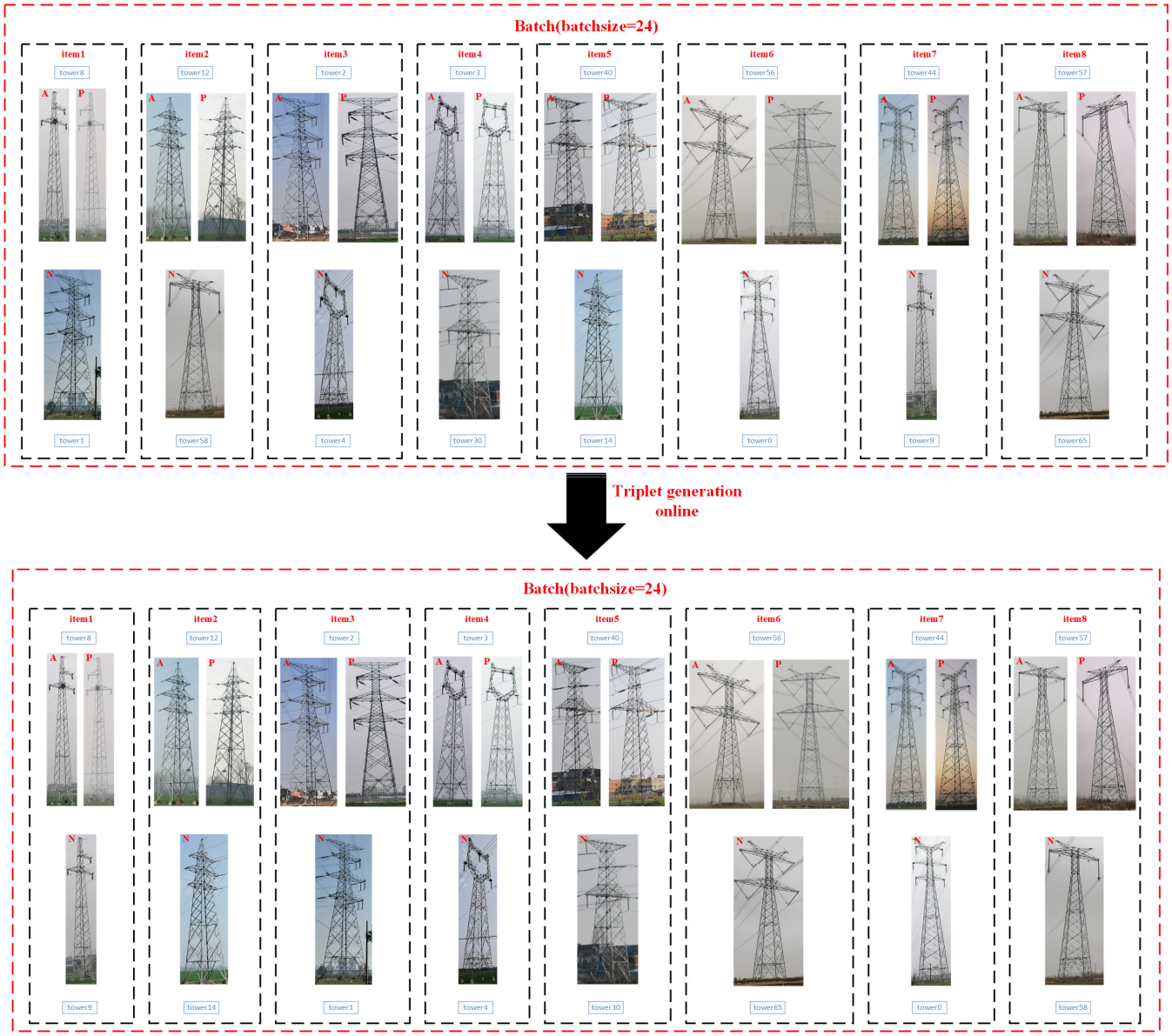

Figure 8. Schematic diagram of online triplet generation.

Table 3. Template library structure(features1 is a 128-dimensional feature vector).

Field	Data Type	Description	Source
ID	int	Primary key	Auto-increment
features1	varchar(4000)	Image feature vector	Matching network result
tower_ID	int	Transmission tower identifier	Manual entry
GPS_longitude	float(20,10)	Image longitude info	Image info or "null"
GPS_latitude	float(20,10)	Image latitude info	Image info or "null"

2.3.4. Matching Network Loss Function

During the early stages of model training, using only triplet loss might lead to difficulties in network convergence. Thus, a common approach is to integrate multi-class

cross-entropy loss to facilitate network convergence [35]. In this study, we utilize a softmax activation function in the network's final layer to map the feature vector to a probability distribution across different transmission tower instances (categories), compute the cross-entropy loss, and subsequently optimize the model in conjunction with the triplet loss. As a result, the composite loss function of the model is shown in Equation (15).

$$S = \frac{1}{N}\sum_{i=1}^{N}\left[\| f(x_i^a) - f\left(x_i^p\right) \|_2^2 - \| f(x_i^a) - f(x_i^n) \|_2^2 + a\right]_+ + \frac{1}{m}\sum_{i=1}^{m}\sum_{c=1}^{C} y_{i,c}\log(p_{i,c}) \quad (15)$$

where N represents the number of online-generated triplet samples, and $f\left(x_i^a\right)$ represents the embedding vector of the anchor image obtained through the network. $f\left(x_i^p\right)$ represents the embedding vector of the positive sample obtained from the network. $f\left(x_i^n\right)$ represents the embedding vector of the negative sample procured from the network. α represents the enforced margin between positive and negative sample pairs. m represents the number of images in the online-generated triplets. C represents the number of categories. $y_{i,c}$ represents the actual label of the ith sample in the cth category. $p_{i,c}$ represents the predicted probability of the ith sample in the cth category.

2.3.5. Matching Network Training Method

The training process of the matching network is illustrated in Figure 9. During the training process, the images first undergo data augmentation. Data augmentation techniques include random flipping, HSV color enhancement, and random bottom occlusion. Augmented images are fed into the matching network to generate corresponding feature vectors (features1 and features2). Specifically, features1 is utilized to construct triplet samples. The detailed description of online triplet generation is provided in Section 2.3.3. Furthermore, features2 is employed to compute the cross-entropy loss. The total loss, described in detail in Section 2.3.4, is optimized using the Adam optimizer during the training of the matching network.

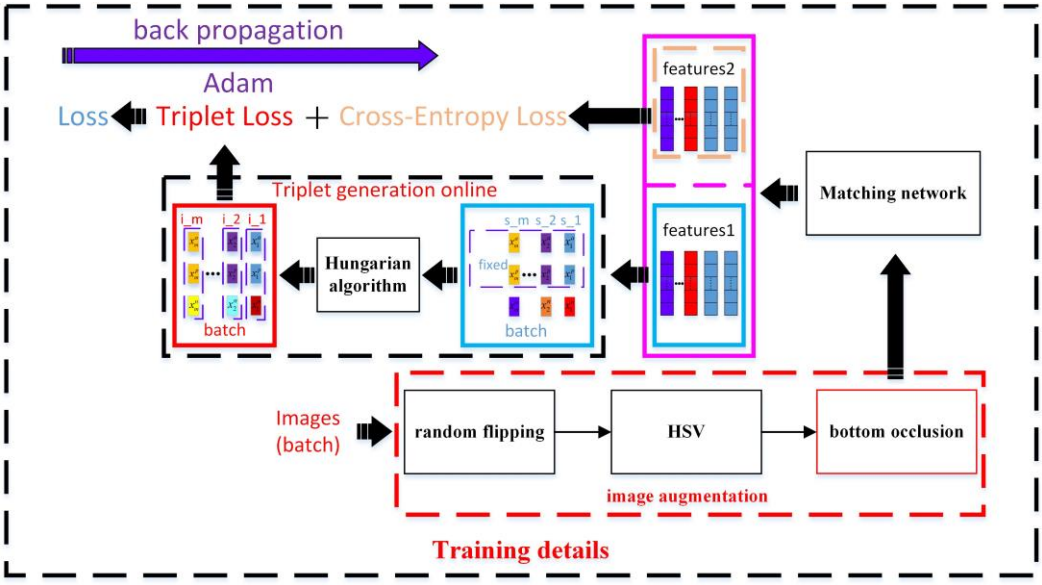

Figure 9. Training procedure of the matching network.

2.3.6. Matching Similarity Evaluation

Each cropped transmission tower image, when passed through the matching network, produces a d-dimensional feature vector. This study opted to employ the Euclidean

distance to evaluate the matching results, using similarity assessment criteria illustrated in Equation (16). A smaller Euclidean distance indicates higher similarity.

$$sim(A, B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2} \tag{16}$$

2.3.7. Matching Accuracy Evaluation

Given the current absence of established evaluation standards for transmission tower matching, the evaluation criteria for the identity matching results of transmission towers refer to common metrics in the image retrieval domain, namely *rank*-1, Mean Average Precision (mAP). Given that the calculation of mAP and Average Precision (AP) in image retrieval differs from that in object detection, in this context, we denote mAP and AP in image retrieval as mAP_r and AP_r, respectively. The *rank*-1 denotes the ratio of the number of times all query images successfully match the total number of queries. Typically, a higher rank-1 metric indicates a better model matching performance. To compute mAP_r, one first needs to determine AP_r. The AP_r value in image retrieval serves as a measure of the re-identification performance, and is computed as delineated in Equation (18).

$$\Delta\ recall = \frac{1}{N_t} \tag{17}$$

$$AP_r = \sum_i \frac{TP_r(i)}{TP_r(i) + FP_r(i)} \cdot \Delta\ recall,\ i \in \Omega \tag{18}$$

where N_t represents the number of images in the gallery that share the same ID with the query image; Ω represents the set of samples with the same identity as the query sample and ranking in the kth position in the query results (for example, the notation Ω = {1, 3, 4} signifies that the identity at the first, third, and fourth ranks within the retrieval outcomes correspond to the identity of the target image under query); $TP_r(i)$ represents the number of positive samples in the top i retrieval results; $FP_r(i)$ represents the number of negative samples in the top i retrieval results.

mAP_r is a more comprehensive metric that not only considers the most similar image but also reflects the similarity situation of all other images in the template library with the image to be retrieved. Hence, it is more indicative of the merits of the algorithm. The computation for mAP_r is shown in Equation (19)

$$mAP_r = \frac{1}{N}\sum_{m=1}^{N} AP_r(m) \tag{19}$$

where mAP_r represents the Mean Average Precision for matching network; N represents the number of query images; $AP_r(m)$ represents the average precision for the mth query images presented in Equation (19).

This study posits that the accuracy of the top few images in the retrieval results is crucial for evaluating the model. Consequently, we introduce a tailored evaluation metric, termed Mean Average Precision, based on the top three most similar retrieved images (mAP_rank-3). Within this metric, the AP_r for each query image is restricted to the top three retrieval results in the gallery, denoted as the AP_rank-3. The calculation methods for AP_rank-3 and mAP_rank-3 refer to Equations (18) and (19), with the specific computational approaches illustrated in Equations (20) and (21)

$$AP\_rank\text{-}3 = \frac{1}{3}\sum_i \frac{TP_r(i)}{TP_r(i) + FP_r(i)},\ i \in \Omega_1 \tag{20}$$

$$mAP\_rank\text{-}3 = \frac{1}{N}\sum_{m=1}^{N} AP\_rank\text{-}3(m) \tag{21}$$

where Ω_1 denotes the set of samples that have the same identity as the query sample and rank at the jth position in the query results, under the condition that j is less than or equal to 3; N represents the number of query images.

2.4. Transmission Tower Identity Matching Template Library

Establishing the transmission tower identity matching template library lays the foundation for identifying transmission towers. The process for setting up this library is illustrated in Figure 10. Input images are cropped to isolate transmission tower images using the YOLOv8s module. Then, cropped transmission tower images are passed through a matching network to obtain the feature vectors, denoted as "features1", which are written into the template library. If the input image contains GPS information, the GPS information is added to the transmission tower template library. If not, the GPS information value is set to "null". Additionally, the "tower_ID" is manually inputted. When features1 is a 128-dimensional feature vector, the data structure of the template library is presented in Table 3. However, when features1 is a 256-dimensional feature vector, varchar(4000) is insufficient to store this vector. In such cases, we employ varchar(8000) to accommodate features1.

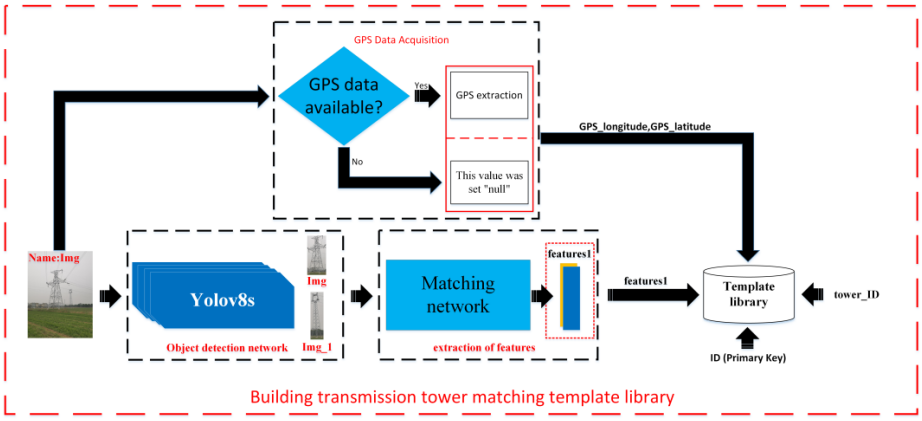

Figure 10. Template library construction flowchart.

2.5. Transmission Tower Identity Matching Framework

The pseudocode for the identity matching process of the transmission tower is presented as Algorithm 1. To facilitate real-time detection on mobile devices, the YOLOv5n-conv head is employed in this study for efficient transmission tower detection during the model's inference phase. Firstly, input images are processed through the YOLOv5n-conv head for transmission tower detection. In scenarios where the input images do not contain transmission tower instances, these images are skipped without further processing. However, when transmission tower instances are detected, they are cropped from the input images for subsequent matching processes. The focus of this section is on cases where transmission tower instances are present in the input images. For images with transmission tower instances, the matching algorithm is shown in Figure 11. In our designed matching algorithm, we utilize different databases for matching based on the presence or absence of standard GPS data. Consequently, this paper employs arrows of three different colors—yellow, black, and red—to guide the reader through the entire matching procedure. In Figure 11, the yellow arrows indicate the necessary steps for the matching algorithm. The black arrows serve as indicators, illustrating the algorithm's operational flow when GPS data are absent in the input image. Conversely, the red arrows depict the algorithm's process when GPS data are present in the input image. Feature vectors, denoted as features1, are obtained through the matching network discussed in Section 2.3.1. The system determines whether the input image contains GPS information. If the input image lacks GPS data, the model compares the feature vectors generated by the matching network with those stored in the template library using the Euclidean distance, a process detailed in Section 2.3.6. If the input image contains GPS data, the model first filters the template

library using this GPS information to form a candidate set (Subset). The feature vectors from the input image are then compared with those vectors in the candidate set using the Euclidean distance to identify the best match. When the Euclidean distance between the most similar image and the original exceeds 1, the model determines that the transmission tower is not present in the template library.

Algorithm 1 Identifying the transmission tower in the input image

```
input: input_image(A)
output: result
# Step 1: Detect and Crop the input image to get transmission tower instances
tower_instances=YOLOv5_conv_head(input_image)
if tower_instances=[]:
    return "null"
# Step 2: Extract feature vector using matching network
input_features1= Matching_Network(tower_instances)
# Step 3: Check if input image has GPS information
if input_image. GPS():
    # Use GPS information to filter template library
    candidate_set = FilterDatabaseUsingGPS(input_image.GPS_data)
else:
    #Use the entire template library
    candidate_set = EntireTemplateLibrary
result=[]
# Step 4: Compare feature vector with candidate set using Euclidean distance
For tower_feature in input_features1:
    min_distance = INFINITY
    best_match = NULL
    for template in candidate_set:
        distance = EuclideanDistance(tower_feature, template. features1)
        if distance < min_distance:
            min_distance = distance
            best_match = template.tower_ID
# Step 5: Determine if the transmission tower is in the template library or not
    if min_distance > 1:
        answer="null"
    else:
        answers= best_match
    result.append(answer)
return result
```

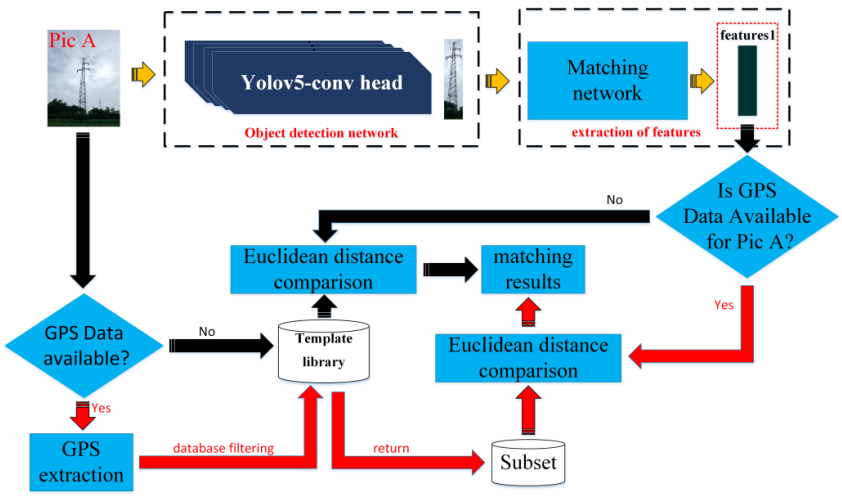

Figure 11. Framework for identity matching of transmission towers from input images.

3. Experiment and Analysis

3.1. Dataset

3.1.1. Object Detection Dataset

Given the scarcity of diverse transmission tower types and instances in public datasets, this study captured 5552 images of transmission towers from various angles, different shooting distances, and under diverse lighting conditions. All collected images are in JPG format with a resolution of 2736 × 3648 pixels. The dataset encompasses eight types of transmission tower structures, as illustrated in Figure 12. To meet the requirements for training and evaluating the transmission tower identification algorithm, all images were annotated with bounding rectangles using the LabelImg tool.

Figure 12. Different types of transmission towers.

The image dataset of transmission towers is split into training, testing, and validation subsets at a ratio of 7:2:1. Specifically, the training subset contains 3965 images, the testing subset includes 1036 images, and the validation subset has 551 images. A detailed distribution of the dataset can be seen in Table 4.

Table 4. Object detection dataset.

Category	Train	Test	Val	Total
butterfly_shape	520	156	76	752
cat_shape	911	291	131	1333
four	716	181	94	991
gan	752	205	108	1065
goat_shape	619	148	75	842
line	567	154	78	799
shang	654	183	101	938
six	958	223	117	1298
total	5697	1541	780	8018

3.1.2. Matching Dataset

Environmental factors might influence the matching of transmission towers. Data collection was conducted on 82 transmission tower examples under varying weather conditions and timepoints. We created a dataset of transmission tower images captured using mobile phones. A total of 1687 images were amassed for the training set,

421 for the validation set, and 347 for the test set. Environmental information aids in the successful identification of transmission towers. However, over time, these environmental characteristics may evolve. Consequently, an over-reliance on such environmental data should be avoided. For this reason, each instance of the transmission towers in the images was cropped. The cropped images were then used to construct the matching dataset. Figure 13 depicts a cropped image of a particular transmission tower instance from the dataset, captured at varied distances and orientations.

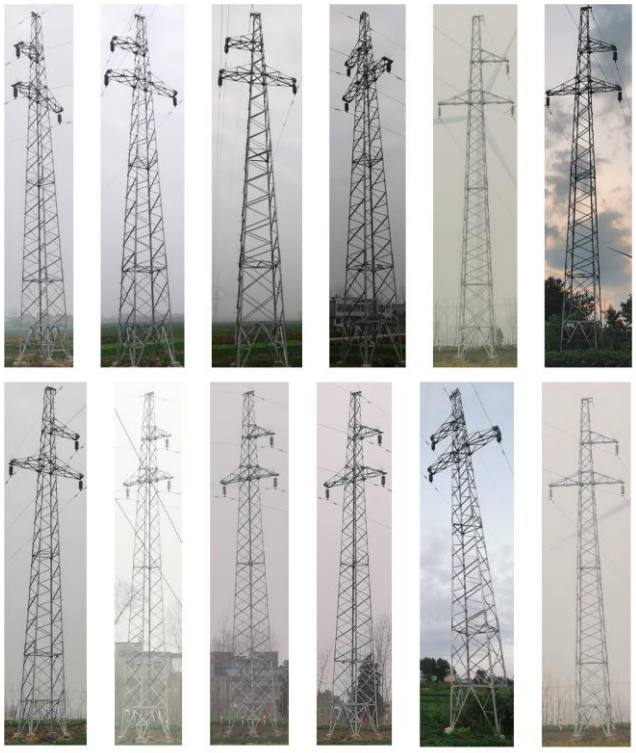

Figure 13. Sample images of a transmission tower from the matching dataset.

3.2. Experiment on Transmission Tower Object Detection

3.2.1. Model Training

Experiments were conducted on Dell Server T7920 (Server 1) and Lenovo Server (Server 2). Server 1 is equipped with two Intel Xeon Gold 6248R CPUs and two NVIDIA A4000 16G GPUs. Server 2 is configured with a 12th Gen Intel(R) Core(TM) i7-12700H CPU and a GeForce RTX 3060 6G GPU. Ubuntu 18.04 LTS was installed on Server 1. Both Server 1 and Server 2 were set up with Python 3.8 and configured with environments including PyTorch 1.13.1 and Ultralytics 8.0.81. YOLOv5n, YOLOv5s, YOLOv8n and YOLOv8s adopted the official COCO dataset weights for weight transfer, while the YOLOv5n-c2f head and YOLOv5n-conv head utilized the pre-trained weights of the YOLOv5n model, specifically trained for transmission tower detection, for the purpose of transfer learning. The input image resolution was set to 640 $\times$ 640, with the Adam optimizer utilized for model optimization, and betas set to (0.937, 0.999). YOLOv5 models necessitate predefined anchor boxes. Through K-means clustering and genetic algorithms, the priors for the first prediction head were determined as [9, 19], [15, 43], [25, 69]; the priors for the second prediction head were [36, 110], [57, 158], [77, 265], while those for the third prediction head were [98, 355], [130, 418], [195, 461]; The binary cross-entropy loss weight for confidence was set to 1, the mean squared error loss weight for positional offset was set to 0.05, and the cross-entropy loss weight for class probability was set to 0.5. Since YOLOv8 employs an anchor-free head, it was appropriate to assign a relatively larger weight to the bounding box loss to expedite convergence. The weight distribution was as follows: bounding box loss at 7.5, classification loss at 0.5, and DFL loss at 1.5. The initial learning rate was set to

0.01, with a final learning rate of 0.0001. The training was configured for 200 epochs with a linear decay based on the number of epochs. Model training was conducted on Server 1. It was observed that YOLOv8n converged around the 190th epoch. A comparison was made between YOLOv5n, YOLOv5s, YOLOv8s, YOLOv5n-c2f head, and YOLOv5n-conv head for convergence patterns from epochs 0 to 190. The experimental results are illustrated in Figure 14.

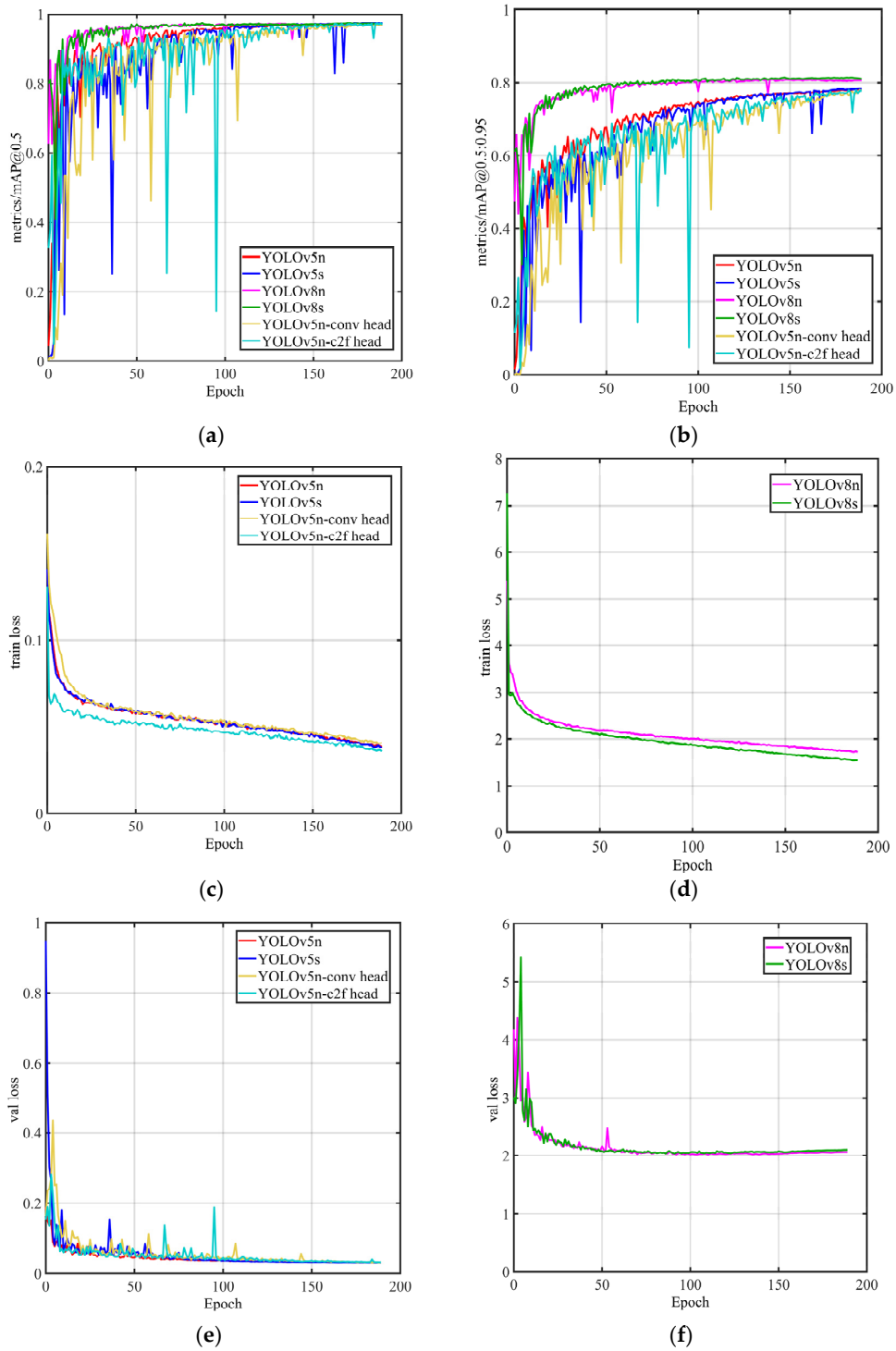

Figure 14. Object detection experimental results (**a**) metrics/mAP@0.5 curve; (**b**) metrics/mAP@0.5 : 0.95 curve; (**c**) training loss curve of the YOLOv5; (**d**) validation loss curve of the YOLOv5; (**e**) training loss curve of the YOLOv8; (**f**) validation loss curve of the YOLOv8.

Figure 14 indicates that the convergence speed and accuracy of the YOLOv8s and YOLOv8n models are superior to those of YOLOv5s and YOLOv5n. The YOLOv5n-c2f head adopted parameters from YOLOv5n (transmission tower detection), showing a faster decline in training loss compared to YOLOv5s and YOLOv5n. However, possibly due to the model not being trained on the COCO dataset and a limited collection of dataset images being available, no accuracy improvement was observed on the validation dataset. The YOLOv5n-conv head also utilized transfer learning with YOLOv5n parameters (transmission tower detection). The results suggest a minor decline in model convergence accuracy.

3.2.2. Model Evaluation and Result Analysis

The models were evaluated on Server 1 and Server 2 using the test dataset, with the results detailed in Table 5. The results indicate that the YOLOv8s achieved the highest accuracy on the test dataset, while the YOLOv5n-conv head demonstrated the fastest inference speed on the dataset. The results show that, on Server 1, the inference speed of YOLOv5n-conv head improved by 1.7 ms compared to the YOLOv5n. On Server 2, compared to the YOLOv5n, its speed improved by 3.3 ms. Despite a reduction of approximately 0.7 in GFLOPs, there was no significant decrease in mAP@0.5 and mAP@0.5:0.95. In comparison to YOLOv8s and YOLOv8n, the mAP@0.5 for YOLOv5n-conv head only decreased by 0.003 and 0.006, respectively, while mAP@0.5:0.95 declined by 0.028 and 0.03. Moreover, when the batchsize was set to 1, there was a notable reduction in inference time. Therefore, on the current dataset, the YOLOv5n-conv head proposed in this study excels in both speed and accuracy, offering a viable option for practical applications in mobile transmission tower detection. YOLOv8s exhibits the highest accuracy and is suitable for the establishment of a template library.

Table 5. Performance results of object detection algorithms on the test dataset.

Model	mAP@0.5	mAP@0.5:0.95	FLOPs/G	Inference (Batchsize = 1 A4000 Ubuntu)/ms	Inference (Batchsize = 1 3060 Windows)/ms
YOLOv5n-conv head (ours)	0.974	0.791	3.6	**8.1**	**20.5**
YOLOv5n-c2f head (ours)	0.974	0.792	4.8	10.5	24.6
YOLOv5n [15] (in 2022)	0.977	0.792	4.3	9.8	23.8
YOLOv5s [15] (in 2022)	0.975	0.789	15.8	10	51.8
YOLOv8n [17] (in 2023)	0.977	0.819	8.2	10.1	35.2
YOLOv8s [17] (in 2023)	**0.98**	**0.821**	28.7	10.9	43.5

3.3. Experiment on Transmission Tower Identity Matching

3.3.1. Model Training

To enhance the performance of the trained model under the constraints of a limited dataset, this study employed transfer learning during model training. The backbone feature extraction networks of all architectures adopted parameters pretrained on the ImageNet dataset for model initialization. In terms of model construction, this research evaluated three distinct backbone network architectures: MobileNet [41], MobileViT [42], and Inception-ResNet-v1 [43]. The Adam optimizer was employed, with betas set to (0.9, 0.999). A cosine learning rate schedule was used, initializing the learning rate at 0.001 with a minimum rate of 10^{-5}. The model's batchsize was set to 27, with the triplet loss hyperparameter α set to 0.15. The training results are illustrated in Figure 15. As shown in Figure 15, models employing an online triplet generation strategy demonstrate an accelerated curve convergence speed when the backbone network generates 128-dimensional feature vectors, in comparison to using basic triplet. It is noteworthy that in large networks such as Inception-ResNet-v1, minor variations during the initial training phase can lead to substantial differences in output. However, the introduction of online triplet generation has been observed to decrease fluctuations in the convergence curve. Furthermore, for the Inception-ResNet-v1 backbone network, using a 256-dimensional feature vector under the online triplet generation strategy results in more pronounced fluctuations in curve convergence compared to using a 128-dimensional feature vector.

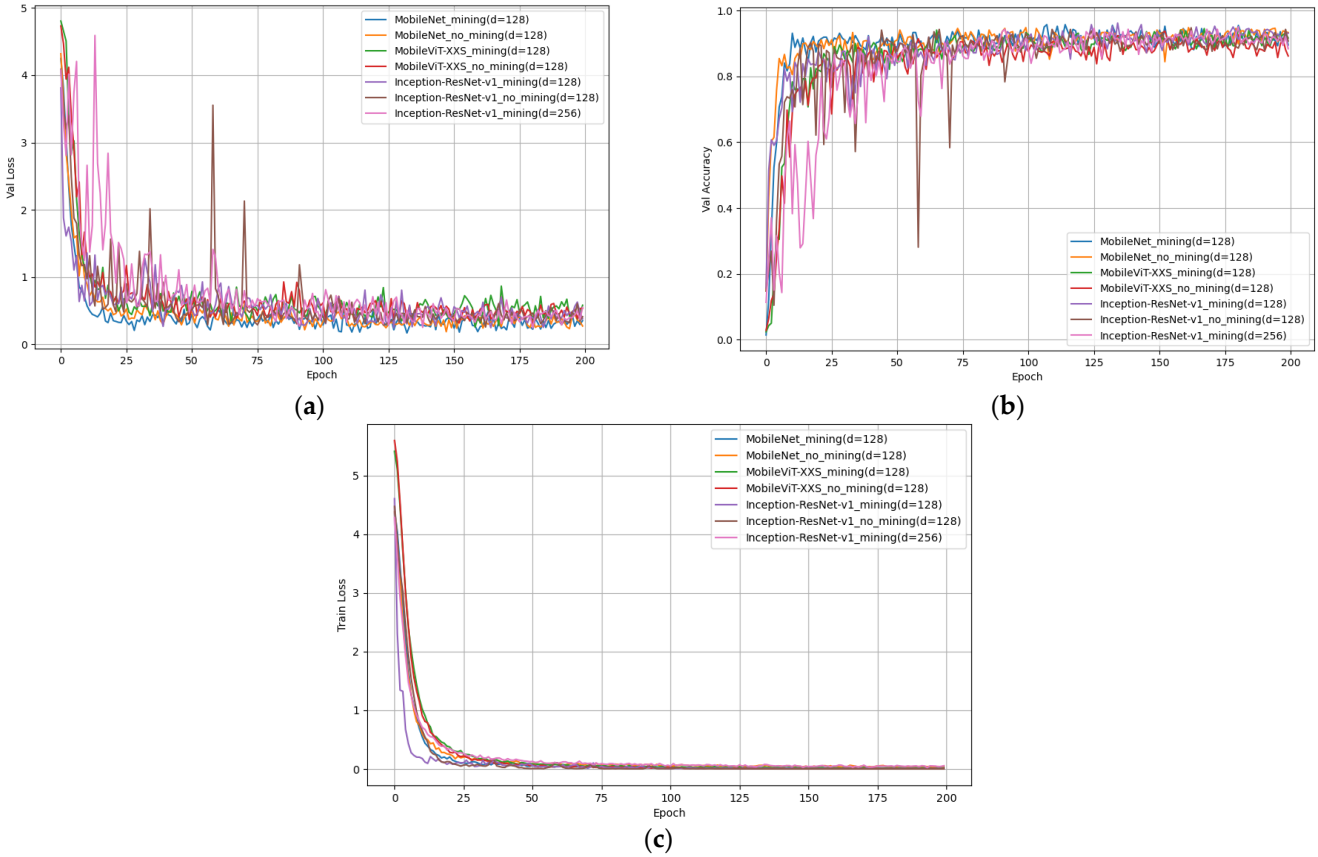

Figure 15. Comparison of training effects in the matching network: (**a**) validation loss curve; (**b**) validation accuracy curve. (**c**) train loss curve.

3.3.2. Model Evaluation and Result Analysis

Utilizing 337 images from the test dataset, for the evaluation of the model, we proceed in two ways. On one hand, we compare the performance of different networks under the same feature dimension (d), and on the other hand, we compare the performance of the same network under different feature dimensions (d). The evaluation of the matching model is conducted using Server 2.

This paper utilizes the training set from the transmission tower matching dataset to establish a template library for transmission tower matching. For matching networks constructed with different backbone networks, when the mapping dimension d is set to 128, the accuracy of the models on the test set is shown in Table 6. Three distinct backbone architectures were integrated into the matching network. Upon adopting the proposed triplet online generation strategy combined with the GPS preliminary filtering, enhancements were observed in both rank-1 and mAP_rank-3. With the exclusive integration of the triplet online generation strategy, Inception-Resnet-v1 exhibited the highest matching precision, registering a rank-1 score of 89.32 and a mAP_rank-3 of 90.31. When solely employing the GPS preliminary filtering, MobileNet demonstrated superior matching precision, achieving a rank-1 score of 88.72 and a mAP_rank-3 of 90.01. Incorporating both the triplet online generation strategy and GPS preliminary filtering, Inception-Resnet-v1 sustained its leading performance, with a rank-1 score of 89.32 and a mAP_rank-3 of 90.31. Furthermore, after deploying GPS preliminary filtering, the inference speed of all models was improved.

Table 6. Comparative results in the test set for matching algorithms with varying backbone networks (d = 128).

Backbone	Triplet Generation Online (Mining)	GPS	Rank-1 (%)	mAP_Rank-3 (%)	Inference (Rank-1)/ms	mAP_r (%)
MobileViT-XXS (d = 128)	√	√	**87.24**	**87.81**	35.1	**88.22**
	√		86.65	87.36	71.2	87.59
		√	86.35	87.56	**34.8**	86.40
			85.16	86.47	71.9	84.86
MobileNet (d = 128)	√	√	**89.32**	**90.08**	**29.5**	89.69
	√		88.43	89.14	68.0	88.89
		√	88.72	90.01	29.9	**90.53**
			88.13	89.37	68.4	89.89
Inception-Resnet-v1 (d = 128)	√	√	**89.32**	**90.31**	**33.3**	85.98
	√		89.32	90.21	70.3	85.04
		√	86.35	87.66	34.5	**88.24**
			85.46	86.92	69.6	87.37

The aforementioned experimental findings elucidate that the application of triplet mining markedly elevates the rank-1 accuracy of models, a fact most prominently observed in the Inception-ResNet-v1 based matching network, which attains the peak rank-1 accuracy subsequent to the mining implementation. Delving into the effects of the feature dimension d, as generated by the matching network, this investigation assesses the impact of varying d values on the network's efficacy, predicated based on the integration of online triplet generation. As delineated in Table 7, a comparative analysis of network performance, utilizing Inception-ResNet-v1 as the backbone for d values of 128 and 256, reveals enhancements in rank-1, mAP_rank-3, and mAP metrics upon increasing d from 128 to 256. Nevertheless, in instances where GPS information is incorporated within the input images, the model experiences a processing deceleration of 16.4 ms per image when the feature dimension d is increased to 256, compared to d = 128. In the absence of GPS data, this delay extends to 40.5 ms per image. Therefore, setting d to 256 is suitable for processing inspection images on local servers, while setting d to 128 is more appropriate for deployment on edge devices such as UAVs.

Table 7. Comparative results on the test set for matching algorithms under different feature dimensions (d = 128, d = 256).

Backbone	Triplet Generation Online (Mining)	GPS	Rank-1 (%)	mAP_Rank-3 (%)	Inference (Rank-1)/ms	mAP_r (%)
Inception-Resnet-v1 (d = 128)	√	√	**89.32**	**90.31**	**33.3**	**85.98**
	√		89.32	90.21	70.3	85.04
Inception-Resnet-v1 (d = 256)	√	√	**91.39**	**91.62**	**49.7**	**90.70**
	√		90.50	91.15	110.8	89.98

3.3.3. Visualization of Matching Network Results

When the matching network generates 'features1' as a 128-dimensional feature vector, the use of a matching network with Inception-ResNet-v1 (d = 128) as its backbone demonstrated a robust overall performance on the dataset used in this study. Additionally, although the use of a matching network with Inception-ResNet-v1 (d = 256) as its backbone slightly surpasses its counterpart with d = 128 in terms of the accuracy metrics, the longer inference time of the former makes it less suitable for optimization and deployment on edge devices. Given our aim to deploy the algorithm directly on edge devices in the future, we chose Inception-ResNet-v1 (d = 128) as the backbone network for our transmission tower re-identification system. For a more in-depth performance analysis, this paper considered the impact of online triplet generation strategy. In Figure 16, the numbers above each

image (e.g., 0.4383) show the Euclidean distance between its 'features1' and the query's 'features1'. Smaller values denote higher similarity. "tower'*i*'" indicates its sequence as the *i*th transmission tower instance. The search results from left to right represent the decrease in image similarity (top-1, top-2, top-3). As illustrated in Figure 16a,b, the online triplet generation strategy can notably reduce distances between images of the same tower, thereby enhancing the rank-1 accuracy of the matching network. In this study, a subsequent data collection was conducted six months later for a subset of transmission towers. As Figure 16c illustrates, changes in the environment led to some mismatches between the newly collected images of these towers and those in the database. However, the metric learning matching algorithm employed in this research demonstrated its robustness; even without retraining the model, the Euclidean distance between the newly collected image of a specific tower and its predicted counterpart remained small. This finding strongly supports the argument that regularly updating the database can maintain and enhance the model's predictive accuracy.

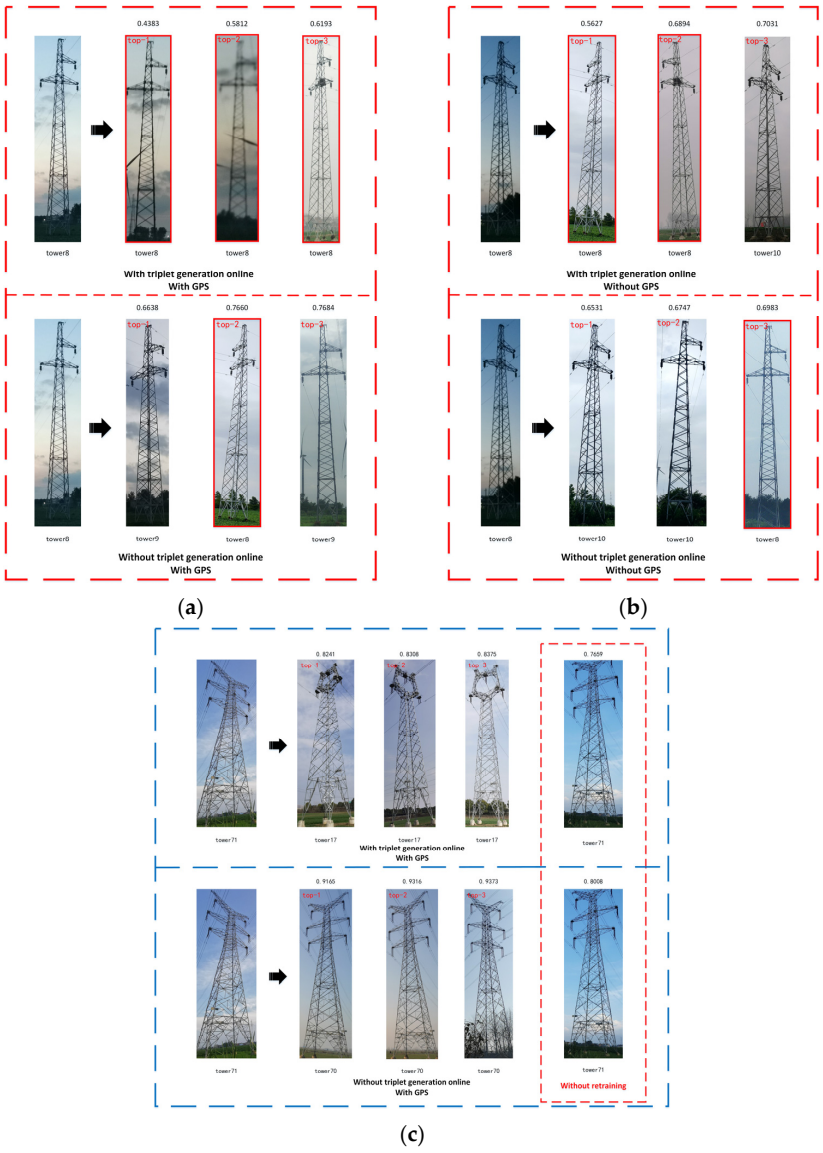

Figure 16. Illustrative examples of retrieval outcomes derived from the matching network utilizing Inception-Resnet-v1 (d = 128) as the backbone: (**a**) the impact of online triplet generation with GPS; (**b**) the impact of online triplet generation without GPS; (**c**) impact of environmental variability on the model.

4. Conclusions

We introduced a novel system tailored for transmission tower re-identification, delineated by the ensuing attributes:

1. For transmission tower detection, we devised two derivative architectures grounded on YOLOv5n: YOLOv5n-C2f head and YOLOv5n-conv head. Empirical evidence underscores the superior efficacy of YOLOv5n-conv head, achieving a 1.7 ms reduction in detection time compared to YOLOv5n on Server 1;
2. Within the matching network, three disparate backbone architectures—MobileNet (d = 128), MobileViT (d = 128), and Inception-Resnet-v1 (d = 128)—witnessed enhancements in both convergence velocity and rank-1 matching precision upon the assimilation of an online triplet sample generation strategy. Notably, leveraging Inception-Resnet-v1 (d = 128) as the backbone culminated in a pinnacle rank-1 matching precision of 89.32%;
3. Harnessing GPS to constrict the matching ambit augments both matching accuracy and efficiency. Instituting a GPS preliminary filtering scope of $[-0.05, +0.05]$ yields a superior outcome. Employing Inception-Resnet-v1 (d = 128) as the backbone elevates matching precision, yet trims the matching time by approximately 37 ms;
4. In the absence of GPS signals, this matching network can also achieve identity matching for transmission towers with a success rate of 89.32%, but the matching time will increase by 37 ms.

After a comprehensive assessment of our research findings and their implications, we acknowledge the following limitations in the transmission tower re-identification algorithm that we propose:

1. The datasets leveraged for object detection and matching in this study are constrained in size. Amassing a dataset could fortify the model's adaptive capacity;
2. The system's matching accuracy falls below that of nameplate recognition and POS localization. Furthermore, its inference time is longer compared to POS localization;
3. Without GPS data, the model experiences an increase in inference time. Additionally, the inference speed of the model proportionally increases with the number of images in the template database. For mobile applications, reducing the system's inference time during the inference phase is crucial.

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

Funding: This research was funded by Guangxi Science and Technology Major Special Fund (Guike AA23062024).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. They are not publicly disclosed to protect the extensive intellectual and proprietary investments made during the dataset's meticulous curation.

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

References

1. Liu, Z.Y.; Wu, G.P.; He, W.S.; Fan, F.; Ye, X.H. Key target and defect detection of high-voltage power transmission lines with deep learning. *Int. J. Electr. Power Energy Syst.* **2022**, *142*, 14. [CrossRef]
2. Graditi, G.; Buonanno, A.; Caliano, M.; Di Somma, M.; Valenti, M. Machine Learning Applications for Renewable-Based Energy Systems. In *Advances in Artificial Intelligence for Renewable Energy Systems and Energy Autonomy*; Manshahia, M.S., Kharchenko, V., Weber, G.-W., Vasant, P., Eds.; Springer International Publishing: Cham, Switzerland, 2023; pp. 177–198.

3. Markus, S. Machine Learning for Energy Transmission. Available online: https://www.datarevenue.com/en-blog/machine-learning-for-energy-transmission (accessed on 30 November 2023).
4. Atrigna, M.; Buonanno, A.; Carli, R.; Cavone, G.; Scarabaggio, P.; Valenti, M.; Graditi, G.; Dotoli, M. A Machine Learning Approach to Fault Prediction of Power Distribution Grids Under Heatwaves. *IEEE Trans. Ind. Appl.* **2023**, *59*, 4835–4845. [CrossRef]
5. Khan, M.A.; Asad, B.; Vaimann, T.; Kallaste, A.; Pomarnacki, R.; Hyunh, V. Improved Fault Classification and Localization in Power Transmission Networks Using VAE-Generated Synthetic Data and Machine Learning Algorithms. *Machines* **2023**, *11*, 963. [CrossRef]
6. 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]
7. Wang, H.; Yang, G.; Li, E.; Tian, Y.; Zhao, M.; Liang, Z. High-Voltage Power Transmission Tower Detection Based on Faster R-CNN and YOLO-V3. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 8750–8755.
8. Liao, J.; Xu, H.; Fang, X.; Zhang, D.; Zhu, G. Quantitative Assessment Framework for Non-Structural Bird's Nest Risk Information of Transmission Tower in High-Resolution UAV Panoramic Images. In Proceedings of the 2023 IEEE International Conference on Power Science and Technology (ICPST), Kunming, China, 5–7 May 2023; pp. 974–979.
9. Tang, C.; Dong, H.; Huang, Y.; Han, T.; Fang, M.; Fu, J. Foreign object detection for transmission lines based on Swin Transformer V2 and YOLOX. In *The Visual Computer*; Springer: Berlin/Heidelberg, Germany, 2023. [CrossRef]
10. Souza, B.J.; Stefenon, S.F.; Singh, G.; Freire, R.Z. Hybrid-YOLO for classification of insulators defects in transmission lines based on UAV. *Int. J. Electr. Power Energy Syst.* **2023**, *148*, 108982. [CrossRef]
11. Wang, B.; Dong, M.; Ren, M.; Wu, Z.; Guo, C.; Zhuang, T.; Pischler, O.; Xie, J. Automatic Fault Diagnosis of Infrared Insulator Images Based on Image Instance Segmentation and Temperature Analysis. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 5345–5355. [CrossRef]
12. Yang, Y.; Wang, M.; Wang, X.; Li, C.; Shang, Z.; Zhao, L. A Novel Monocular Vision Technique for the Detection of Electric Transmission Tower Tilting Trend. *Appl. Sci.* **2023**, *13*, 407. [CrossRef]
13. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
14. 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]
15. Ultralytics.YOLOv5. Available online: https://github.com/ultralytics/yolov5 (accessed on 27 November 2023).
16. 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 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023; pp. 7464–7475.
17. Ultralytics. YOLOv8. Available online: https://github.com/ultralytics/ultralytics (accessed on 18 April 2023).
18. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 10778–10787.
19. Zhang, Z.; Xie, X.; Song, C.; Dai, D.; Bu, L. Transmission Tower Detection Algorithm Based on Feature-Enhanced Convolutional Network in Remote Sensing Image. In Proceedings of the Pattern Recognition and Computer Vision, Cham, Switzerland, 18 June 2022; pp. 551–564.
20. Bian, J.; Hui, X.; Zhao, X.; Tan, M. A monocular vision–based perception approach for unmanned aerial vehicle close proximity transmission tower inspection. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 172988141882022. [CrossRef]
21. Sheng, Y.; Dai, Y.; Luo, Z.; Jin, C.; Jiang, C.; Xue, L.; Cui, H. A YOLOX-Based Detection Method of Triple-Cascade Feature Level Fusion for Power System External Defects. In Proceedings of the 2022 7th International Conference on Communication, Image and Signal Processing (CCISP), Chengdu, China, 18–20 November 2022; pp. 452–456.
22. Zhao, Z.; Guo, G.; Zhang, L.; Li, Y. A new anti-vibration hammer rust detection algorithm based on improved YOLOv7. *Energy Rep.* **2023**, *9*, 345–351. [CrossRef]
23. Zhang, J.; Lei, J.; Qin, X.; Li, B.; Li, Z.; Li, H.; Zeng, Y.; Song, J. A Fitting Recognition Approach Combining Depth-Attention YOLOv5 and Prior Synthetic Dataset. *Appl. Sci.* **2022**, *12*, 11122. [CrossRef]
24. Kong, L.; Zhu, X.; Wang, G. Context Semantics for Small Target Detection in Large-Field Images with Two Cascaded Faster R-CNNs. *J. Phys. Conf. Ser.* **2018**, *1069*, 012138. [CrossRef]
25. Xia, Y.; Wang, G.; Wang, R.; Zhou, F. A cascaded method for transmission tower number recognition in large scenes. In Proceedings of the International Symposium on Multispectral Image Processing and Pattern Recognition, Abu Dhabi, United Arab Emirates, 25 October 2020.
26. Li, B.; Li, Y.; Zhu, X.; Qu, L.; Wang, S.; Tian, Y.; Xu, D. Substation rotational object detection based on multi-scale feature fusion and refinement. *Energy AI* **2023**, *14*, 100294. [CrossRef]
27. Gang, W.; Qingmin, C.; Lin, Y.I.; Wenqing, P.; Jinju, Q.; Feng, Z. Location technology of transmission line tower based on image. *J. Terahertz Sci. Electron. Inf. Technol.* **2018**, *16*, 796–801.
28. Qin, X.Y.; Wu, G.P.; Lei, J.; Fan, F.; Ye, X.H.; Mei, Q.J. A Novel Method of Autonomous Inspection for Transmission Line based on Cable Inspection Robot LiDAR Data. *Sensors* **2018**, *18*, 22. [CrossRef]

29. Rublee, E.; Rabaud, V.; Konolige, K.; Bradski, G. ORB: An efficient alternative to SIFT or SURF. In Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011; pp. 2564–2571.
30. Zhang, L.; Koch, R. An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency. *J. Vis. Commun. Image Represent.* **2013**, *24*, 794–805. [CrossRef]
31. Guo, K.; Cao, R.; Wan, N.; Wang, X.; Yin, Y.; Tang, X.; Xiong, J. Image matching algorithm based on transmission tower area extraction. *J. Comput. Appl.* **2022**, *42*, 1591–1597. [CrossRef]
32. Jérémie, G.G.; Morel, J.-M.; Gregory, R. LSD: A Line Segment Detector. *Image Process. Line* **2012**, *2*, 35–55. [CrossRef]
33. DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperPoint: Self-Supervised Interest Point Detection and Description. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 337–33712.
34. Sarlin, P.E.; DeTone, D.; Malisiewicz, T.; Rabinovich, A. SuperGlue: Learning Feature Matching With Graph Neural Networks. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 4937–4946.
35. Schroff, F.; Kalenichenko, D.; Philbin, J. FaceNet: A unified embedding for face recognition and clustering. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 815–823.
36. Chen, H.; Wang, Y.; Shi, Y.; Yan, K.; Geng, M.; Tian, Y.; Xiang, T. Deep Transfer Learning for Person Re-Identification. In Proceedings of the 2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM), Xi'an, China, 13–16 September 2018; pp. 1–5.
37. Deng, J.; Guo, J.; Xue, N.; Zafeiriou, S. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019; pp. 4685–4694.
38. Zhang, H.; Liu, M.; Li, Y.; Yan, M.; Gao, Z.; Chang, X.; Nie, L. Attribute-Guided Collaborative Learning for Partial Person Re-Identification. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 14144–14160. [CrossRef]
39. Yang, J.R.; Zhang, J.W.; Yu, F.F.; Jiang, X.Y.; Zhang, M.D.; Sun, X.; Chen, Y.C.; Zheng, W.S. Learning to Know Where to See: A Visibility-Aware Approach for Occluded Person Re-identification. In Proceedings of the 18th IEEE/CVF International Conference on Computer Vision (ICCV), Electr Network, Montreal, BC, Canada, 11–17 October 2021; pp. 11865–11874.
40. Diwan, T.; Anirudh, G.; Tembhurne, J.V. Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimed. Tools Appl.* **2023**, *82*, 9243–9275. [CrossRef]
41. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:abs/1704.04861.
42. Mehta, S.; Rastegari, M. MobileViT: Light-weight, General-purpose, and Mobile-friendly Vision Transformer. *arXiv* **2021**, arXiv:abs/2110.02178.
43. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 31. [CrossRef]
44. Li, W.; Qi, K.; Chen, W.; Zhou, Y. Unified Batch All Triplet Loss for Visible-Infrared Person Re-identification. In Proceedings of the 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 18–22 July 2021; pp. 1–8.
45. Wu, F.; Smith, J.S.; Lu, W.; Pang, C.; Zhang, B. Attentive Prototype Few-Shot Learning with Capsule Network-Based Embedding. In Proceedings of the 16th European Conference on Computer Vision, ECCV 2020, Glasgow, UK, 23–28 August 2020; pp. 237–253.